Pontifícia Universidade Católica
do Rio de Janeiro

Daniel Specht Silva Menezes

Reconhecimento de entidades mencionadas para o português

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio.

Orientador: Prof. Ruy Luiz Milidiú

Rio de Janeiro
Setembro de 2018

PUC-Rio - Certificação Digital Nº 1612825/CA

Pontifícia Universidade Católica
do Rio de Janeiro

Daniel Specht Silva Menezes

Reconhecimento de entidades mencionadas para o português

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Ruy Luiz Milidiú
Orientador
Departamento de Informática – PUC-Rio

Prof. Sérgio Colcher
Departamento de Informática – PUC-Rio

Prof. Daniel Schwabe
Departamento de Informática – PUC-Rio

Prof. Marcio da Silveira Carvalho
Coordenador Setorial do Centro Técnico Científico – PUC-Rio

Rio de Janeiro, 27 de Setembro de 2018

PUC-Rio - Certificação Digital Nº 1612825/CA

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Daniel Specht Silva Menezes

Graduou-se em Sistemas de informação pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Participou de diversos grupos de pesquisa, decidindo por continuar a formação acadêmica por meio do estudo do ramo da Inteligência Artificial.

PUC-Rio - Certificação Digital Nº 1612825/CA

Ficha Catalográfica

Menezes, Daniel Specht Silva

Reconhecimento de entidades mencionadas para o português / Daniel Specht Silva Menezes; orientador: Ruy Luiz Milidiú. – Rio de janeiro: PUC-Rio, Departamento de Informática , 2018.

v., 84 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática .

Inclui bibliografia

1. Informática – Teses. 2. Aprendizado de máquina;. 3. Processamento de linguagem natural;. 4. Reconhecimento de entidades mencionadas;. 5. Wikipédia;. 6. Datasets;. 7. Redes neurais. I. Milidiú, Ruy Luiz. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática . III. Título.

CDD: 004

Agradecimentos

Primeiramente a minha família por ter me apoiado e estimulado em todo o processo de desenvolvimento deste trabalho.

A Luísa, por ter sido paciente em minha ausência.

Ao Pedro Savarese, pelos conselhos.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

PUC-Rio - Certificação Digital Nº 1612825/CA

Resumo

Menezes, Daniel Specht Silva; Milidiú, Ruy Luiz. **Reconhecimento de entidades mencionadas para o português**. Rio de Janeiro, 2018. 84p. Dissertação de Mestrado – Departamento de Informática , Pontifícia Universidade Católica do Rio de Janeiro.

A produção e acesso a quantidades imensas dados é um elemento pervasivo da era da informação. O volume de informação disponível é sem precedentes na história da humanidade e está sobre constante processo de expansão. Uma oportunidade que emerge neste ambiente é o desenvolvimento de aplicações que sejam capazes de estruturar conhecimento contido nesses dados. Neste contexto se encaixa a área de Processamento de Linguagem Natural (PLN) — *Natural Language Processing* (NLP) — , ser capaz de extrair informações estruturadas de maneira eficiente de fontes textuais. Um passo fundamental para esse fim é a tarefa de Reconhecimento de Entidades Mencionadas (ou nomeadas) — *Named Entity Recognition* (NER) — que consistem em delimitar e categorizar menções a entidades num texto. A construção de sistemas para NLP deve ser acompanhada de *datasets* que expressem o entendimento humano sobre as estruturas gramaticais de interesse, para que seja possível realizar a comparação dos resultados com o real discernimento humano. Esses *datasets* são recursos escassos, que requerem esforço humano para sua produção. Atualmente, a tarefa de NER vem sendo abordada com sucesso por meio de redes neurais artificiais, que requerem conjuntos de dados anotados tanto para avaliação quanto para treino. A proposta deste trabalho é desenvolver um *dataset* de grandes dimensões para a tarefa de NER em português de maneira automatizada, minimizando a necessidade de intervenção humana. Utilizamos recursos públicos como fonte de dados, nominalmente o DBpedia e Wikipédia. Desenvolvemos uma metodologia para a construção do corpus e realizamos experimentos sobre o mesmo utilizando arquiteturas de redes neurais de melhores performances reportadas atualmente. Exploramos diversas modelos de redes neurais, explorando diversos valores de hiperparâmetros e propondo arquiteturas com o foco específico de incorporar fontes de dados diferentes para treino.

Palavras-chave

Aprendizado de máquina; Processamento de linguagem natural; Reconhecimento de entidades mencionadas; Wikipédia; Datasets; Redes neurais

PUC-Rio - Certificação Digital Nº 1612825/CA

Abstract

Menezes, Daniel Specht Silva; Milidiú, Ruy Luiz (Advisor). **Named Entity Recognition for Portuguese**. Rio de Janeiro, 2018. 84p. Dissertação de Mestrado – Departamento de Informática , Pontifícia Universidade Católica do Rio de Janeiro.

The production and access of huge amounts of data is a pervasive element of the Information Age. The volume of availiable data is without precedents in human history and it's in constant expansion. An oportunity that emerges in this context is the development and usage of applicationos that are capable structuring the knowledge of data. In this context fits the Natural Language Processing, being able to extract information efficiently from textual data. A fundamental step for this goal is the task of Named Entity Recognition (NER) which delimits and categorizes the mentions to entities. The development o systems for NLP tasks must be accompanied by datasets produced by humans in order to compare the system with the human discerniment for the NLP task at hand. These datasets are a scarse resource which the construction is costly in terms of human supervision. Recentlly, the NER task has been approached using artificial network models which needs datsets for both training and evaluation. In this work we propose the construction of a datasets for portuguese NER with an automatic approach using public data sources structured according to the principles of Semantic Web, namely, DBpedia and Wikipédia. A metodology for the construction of this dataset was developed and experiments were performed using both the built dataset and the neural network architectures with the best reported results. Many setups for the experiments were evaluated, we obtained preliminary results for diverse hiperparameters values, also proposing architectures with the specific focus of incorporating diverse data sources for training.

PUC-Rio - Certificação Digital Nº 1612825/CA

Keywords

Machine Learning; Natural Language Processing; Named Entity Recognition; Wikipédia; Datasets; Artificial Neural Networks

Sumário

PUC-Rio - Certificação Digital Nº 1612825/CA

PUC-Rio - Certificação Digital Nº 1612825/CA

Lista de figuras

PUC-Rio - Certificação Digital Nº 1612825/CA

Lista de tabelas

PUC-Rio - Certificação Digital Nº 1612825/CA

PUC-Rio - Certificação Digital Nº 1612825/CA

1
Introdução

A produção e acesso a quantidades imensas dados é um elemento pervasivo da era da informação. O volume de informação disponível é sem precedentes na história da humanidade e está sobre constante processo de expansão. Uma oportunidade que emerge neste contexto é o desenvolvimento de aplicações que sejam capazes de estruturar e utilizar o conhecimento desses dados a fim de elaborar soluções para uma série de problemas e.g. motores de busca, *marketing*, notícias. O desenvolvimento dessas aplicações busca emular e transcender a o discernimento presencial humano, i.e. reproduzir o processo humano de tomada de decisões.

Neste contexto se encaixa a área de Processamento de Linguagem Natural (PLN) — *Natural Language Processing* (NLP) — : ser capaz de extrair informações estruturadas de fontes textuais. Uma das aplicações neste domínio é o Reconhecimento de Entidades Mencionadas (ou nomeadas) — *Named Entity Recognition* (NER) —, onde busca-se delimitar num texto as menções a indivíduos e suas respectivas classes, e.g. pessoas, empresas, lugares . Soluções para este problema foram aplicadas, por exemplo, para biomedicina[1], geologia[2] e mercado de ações [3].

Entretanto, as soluções desenvolvidas sobre modelos de aprendizado supervisionado necessitam de *datasets* que explicitem as entidades mencionadas, seus tipos e outras informações relevantes para sua identificação e delimitação num texto. O desenvolvimento desses conjuntos de dados requer a supervisão humana para serem caracterizados como *Gold Standard*, i.e. qualidade boa e consistente, fruto da avaliação humana. As técnicas utilizadas para a construção de corpus anotados variam em alguns quesitos, por exemplo: volume de exemplos, proporção de cada classe, estilos diferentes de anotação...

Ademais, existem iniciativas de desenvolvimento de corpus anotados automaticamente e sem supervisão humana direta, os chamados de *Silver Standard*. As iniciativas mais populares nesta direção foram tomadas utilizando o Wikipédia como fonte de dados[4][5][6], aproveitando os padrões nas estruturas de seus artigos a fim de identificar diferentes tipos de entidade e os textos que a mencionam. Um ponto forte dessa abordagem é sua flexibilidade para aplicação em outros idiomas. Outras fontes de dados possíveis são os bancos de

PUC-Rio - Certificação Digital Nº 1612825/CA

dados públicos como WikiData[7], GoogleGraph[8] e DBpedia[9] que também apresentam entidades e seus detalhes, referenciando recursos textuais que discursam sobre as mesmas. A finalidade de corpus *Silver Standard* no contexto de aprendizado supervisionado deve ser limitada a fonte de treinamento, afinal, testar sistemas sobre *datasets* construídos automaticamente não reflete uma estimativa de performance confiável quando o objetivo é avaliar a proximidade dos resultados com o entendimento humano.

A exploração desta tarefa teve três grandes momentos em conferências de NLP e uma em especial para o português descritas a seguir.

A sexta e sétima *Message Understanding Conference* (MUC) [10] [11] apresentaram as primeiras competições para o desenvolvimento de sistemas para NER. Na sexta edição, a tarefa de NER foi formulada como um passo fundamental para a tarefa mais abrangente de extração de informação. Além de NER, essa conferência correspondeu também a um primeiro passo para as tarefas de: (1) correferência— *correference* —, (2) extração de estruturas de predicado-argumento — *predicate-argument structure* — e (3) desambiguação de palavras — *word sense disambiguation* —. Um corpus anotado foi produzido com entidades dos tipos pessoa, lugar, organização, tempo e expressões numéricas para inglês sobre textos de diferentes domínios. A sétima edição se diferenciou da sexta em dois aspectos: (1) *datasets* construído sobre um mesmo domínio e (2) relações entre entidades — *Template Relation* —.

A conferência *Conference of Natural Language Learning* (CoNLL) em suas edições de 2002[12] e 2003[13]. Nelas corpus de diferentes idiomas foram produzidos com o intuito de abordar a tarefa por meio de sistemas extensíveis a diversos idiomas (multi-lingual). A edição de 2002 lidou com os idiomas holandês e espanhol. Já edição de 2003 lidou com inglês e alemão e adicionou informações gramaticais adicionais no corpus produzido (*pos-tags* e *chunks*).

No programa *The Automatic Content Extraction Program* (ACE) [14] foram produzidos corpus anotados para os idiomas árabe, chinês e inglês. Introduziu alguns novos tipos de entidades, totalizando: pessoa, local, organização, instalação, veículo, armamento e ponto geográfico. Houve também a organização das classes das entidades de maneira hierárquica, por exemplo, a classe "organização" pode receber os subtipos "governo", "comercial", "educacional", "sem fins lucrativos" e "outro".

O avanço da tarefa no escopo da língua portuguesa se deu primeiramente em virtude do projeto Linguateca[15], que visa fomentar a evolução das atividades em NLP para o idioma. Por meio do projeto Avaliação de Reconhecimento de Entidades Mencionadas (HAREM) corpus anotados para entidades mencionadas para o português foram desenvolvidos, especificamente o Primeiro

PUC-Rio - Certificação Digital Nº 1612825/CA

HAREM [16] e Segundo HAREM[17]. Para o projeto foi aplicada a metodologia de, "Avaliação Conjunta"[18], um modelo de avaliação em que vários grupos comparam, com base num conjunto de tarefas consensuais, o progresso dos seus sistemas numa dada área usando para isso um conjunto de recursos comum e uma métrica consensual.

Nos quatro eventos mencionados, houve o desenvolvimento de metodologias diferentes para a avaliação da performance dos sistemas, cada uma visando valorizar os sistemas por diferentes características. Essas métricas são mencionadas em Capítulo 3.

Dada a contextualização da tarefa, traçamos como objetivo central deste trabalho o desenvolvimento de sistemas para NER por meio de uma abordagem que visa contornar a escassez de corpus anotados para a tarefa. Para tal, elaboramos um *dataset* para NER em português a partir de bases de dados públicas, DBpedia e Wikipédia especificamente. Em seguida, sobre o *dataset* desenvolvemos modelos de redes neurais explorando variações nas arquiteturas e configurações de hiperparâmetros.

As principais contribuições deste trabalho são:

1. **Construção do *dataset***. Desenvolvemos uma metodologia para a construção dum *dataset* para NER em português, de maneira automatizada, a partir das fontes de dados públicas DBpedia e Wikipédia. Aplicamos diversas técnicas para mitigar o ruído presente no *dataset* e destacamos quais suas possíveis divergências com corpus desenvolvidos por humanos.

2. **Exploração de modelos de redes neurais**. Exploramos as arquiteturas de redes neurais com as melhores performances publicadas, criando variações com o objetivo de utilizar o *dataset* produzido de maneira eficaz. Reportamos os resultados preliminares sobre diversos modelos de redes neurais, apontado um caminho para a calibração dos hiperparâmetros

PUC-Rio - Certificação Digital Nº 1612825/CA

2
Datasets

Neste capítulo são apresentados os *datasets* utilizados neste trabalho.

– Em Seção 2.1 detalhamos as estruturas mais comuns utilizadas para modelagem de *datasets* para NER. O entendimento desses formatos é necessário para se compreender a sua compatibilidade com os modelos desenvolvidos neste trabalho e como essas diferenças afetam o posicionamento dos resultados na literatura.

– Em Seção 2.2 detalhamos o processo de construção do *dataset* a partir de recursos públicos estruturados utilizando a arquitetura *Linked Data* e o pré-processamento utilizado.

– Em Seção 2.7 são descritos os conjuntos de calibração e teste utilizados no que diz respeito a compatibilidade de sua estrutura com os modelos explorados e suas divergências com o *dataset* de treino criado.

PUC-Rio - Certificação Digital Nº 1612825/CA

2.1
Estilos de anotação

O progresso da tarefa de reconhecimento de entidades mencionadas foi sustentada pelo desenvolvimento de *datasets* que diversas vezes adotaram formatos divergentes. Dependendo dos sistemas propostos para a tarefa, as peculiaridades de cada formato podem impactar a performance dos sistemas e em alguns casos, exigir pré-processamentos específicos.

A importância desta secção se deve ao fato de que este trabalho combina o uso de dois formatos distintos de *dataset* descritos a seguir.

A partir da seção é utilizado o termo "*token*". Este termo se refere a um trecho dum texto, uma sequência de caracteres, correspondendo a um a delimitação que explicita um papel gramatical, e.g. palavras, frases, entidades... Ademais, o verbo "tokenizar" será definida como o processo de "Dividir os *tokens*".

2.1.1
Anotação por linguagem de marcação

Uma maneira de delimitar a entidade em texto é por meio da linguagem de marcação — *markup* —. Esta metodologia foi utilizado, nas mencionadas conferências MUC, ACE e no projeto HAREM, sendo o formato precursor. Um exemplo real desta metodologia extraído do corpus HAREM I é exibido em Exemplo 2.1:

```
... meu irmão nasceu em
<LOCAL TIPO="ADMINISTRATIVO" MORF="M,S">
    Tortosendo
</LOCAL>
comigo ...
```

Exemplo 2.1: Exemplo da linguagem de marcação utilizada na estrutura do Primeiro Harem

Uma propriedade crucial deste formato é a delimitação em texto feita a nível de carácteres, não há informações sobre *tokens* de frases, palavras ou outros componentes da estrutura gramatical. Entretanto existem sistemas que necessitam da estrutura do texto a nível de palavras e frases. Portanto, esses sistemas devem realizar o pré-processamento do *dataset* para elicitar as estruturas gramaticais necessárias.

Por exemplo, ao se realizar a elaboração de uma nova rede neural que recebe como entrada os *tokens* de palavras e frases, para se utilizar um *dataset* que não determine essas informações será requerido o pré-processamento do *dataset*. Deve-se dissociar o impacto do modelo de rede neural ao do pré-processamento para compará-lo as abordagens presentes na literatura. Não é incomum que publicações sobre NER não entrem em detalhes quanto as técnicas de pré-processamento envolvidas.

Na literatura no que diz respeito ao desenvolvimento de modelos de redes neurais para a tarefa de NER, são utilizados *datasets* onde se explicitam os limites das frases e palavras, nominalmente os *datasets* das conferências CoNLL de 2002 e 2003. Isso facilita a atribuição dos méritos dos modelos sem estar sujeito a perturbações oriundas de diferentes escolhas de pré-processamento.

2.1.2
Anotação a nível de frases e palavras

Outra metodologia de anotação é realizada a nível de palavras e frases. Essa metodologia foi adotada pela conferência CoNLL de 2002 por meio da anotação tipo BIO (*begin*, *inside*, *outside*).

PUC-Rio - Certificação Digital Nº 1612825/CA

O	O
jornalista	O
João	B-PER
Neves	I-PER
se	O
encontra	O
na	O
América	B-LOC
do	I-LOC
Sul	I-LOC
.	O

Exemplo 2.1: Exemplo de anotação utilizando o modelo BIO.

Na estrutura de anotação BIO (essa metodologia também já foi referenciada por IOB2 ou até mesmo IOB) para cada linha é informada uma palavra e sua classificação. Uma classe tem, por exemplo, o sufixo PER e LOC, para pessoa e organização, respectivamente. A classe apresenta o prefixo "B" — *begin* — se for a primeira palavra de uma entidade ou "I" — *inside* — para as demais palavras da entidade. Se uma palavra não faz parte de uma entidade mencionada ela é classificada com "O" — *outside* —. No total, para o caso ilustrado, são 5 classes. Linhas em branco marcam o final da frase, como ilustrado em Exemplo 2.1. Essa é a metodologia adotada pelos *datasets* mais utilizados na literatura para a exploração de abordagens para NER.

Nesta abordagem existe um maior isolamento do *dataset* de possíveis variações de pré-processamento para os modelos que necessitam das frases e palavras do texto. Afinal, existe a padronização das delimitações de palavras e frases para os experimentos publicados e isso agrega no sentido de atribuir o crédito da performance ao modelo, posicionando firmemente a performance obtida na literatura.

Variações surgiram sobre o BIO visando valorizar algumas características adicionais da delimitação da entidade no texto. Por exemplo, o BIOES (*begin*,*inside*,*end*,*outside*,*single*) onde existe um rótulo adicional para entidades compostas por apenas uma palavra (S - *single*) e o final da entidade é marcado por E (*end*). Na literatura existem exemplos de experimentos que avaliam uma melhoria significante nos resultados usando IOBES[19][20] e outros não presenciam nenhuma melhoria [21]. Para fins de experimentação, realizamos experimentos sobre IOBES e BIO.

PUC-Rio - Certificação Digital Nº 1612825/CA

2.1.3
Impacto dos estilos de anotação neste trabalho

Neste projeto utilizaremos uma combinação de *datasets* de ambos os formatos descritos. Isso se deve a dois fatores principais:

(1) Os modelos de aprendizado supervisionado explorados neste projeto necessitam de entradas que delimitem palavras e frases, para isso a metodologia de anotação a nível de frases e palavras se mostram mais adequadas. Mais especificamente o *dataset* desenvolvido a partir do Wikipédia é modelado em IOB e IOBES.

(2) Os *datasets* de avaliação e teste são os *datasets* do Primeiro HAREM e miniHAREM divididos em diferentes proporções para treino e teste. Entretanto, diferentemente do *dataset* de treino, eles não destacam em texto as palavras e frases, são anotados por meio da metodologia de *markup*, sendo necessário realizar uma etapa de pré-processamento para realizar a transição para o formato BIO. Afinal, exploraremos modelos de redes neurais que na atualidade apresentam suas melhores performances por meio de entradas correspondendo a sentenças e palavras já destacadas. O pré-processamento é descrito em detalhes em Subseção 2.3.8.

2.2
Construção do *dataset* de treino

Nesta secção são descritas as motivações e metodologias adotadas para a construção do *dataset* de treino a partir de recursos públicos.

2.3
Motivação

Nesta seção dissertamos sobre a motivação por trás da construção de corpus de maneira automatizada, desenvolvemos mais detalhadamente alguns dos pontos já mencionados em Seção 2.1.

2.3.1
Esforço humano

Invariavelmente, são necessários corpus anotados por humanos para a avaliação da performance dos sistemas voltados para NER. Afinal, se a proposta dos sistemas é identificar entidades mencionadas da com um mesmo grau de qualidade que um ser humano, é necessário comparar as saídas dos sistema com o entendimento humano.

O esforço necessário para a produção de corpus pode adotar grandes proporções dados os objetivos traçados para a tarefa. Sistemas que buscam

PUC-Rio - Certificação Digital Nº 1612825/CA

ser agnósticos a idioma necessitam de corpus de diferentes idiomas anotados seguindo uma estrutura em comum para que possam ser utilizados pelo mesmo sistema maximizando a compatibilidade. Uma iniciativa de destaque para a produção de corpus anotados com a mesmas estruturas para diversas tarefas de NLP em diversos idiomas é o projeto *Universal Dependencies* (UD)[22].

Domínios diferentes necessitam de anotadores que conheçam o domínio em questão bem o suficiente para que o corpus não apresente uma grande quantidade de discordâncias entre anotadores. Ou seja, o conhecimento necessário para produzir um corpus de qualidade pode fugir do campo de expertise dos anotadores e isso pode afetar negativamente a qualidade final do *dataset* desenvolvido.

Ressalto que a mesmo quando o corpus é produzido por humanos existe um grau de discordância entre os anotadores. Afinal, o discernimento humano sobre a gramática e semântica é um fator sujeito a compreensão particular de cada anotador. Essas diferenças podem ser encaradas como ruído do corpus. Tendo em vista as as preocupações referentes ao impacto dessas discordâncias, esses conflitos podem ser formalmente modelados por meio de métricas de "Concordância entre Múltiplos Anotadores" — *Inter-Annotator Agreement* — [23].

Neste trabalho buscamos capturar uma grande quantidade de exemplos aplicando uma abordagem de construção de exemplos de maneira automatizada. Entretanto, este conjunto está sujeito a uma presença de ruído muito superior a um corpus anotado por humanos.

PUC-Rio - Certificação Digital Nº 1612825/CA

2.3.2 Abrangência de domínios

Independente da performance de um sistema sobre um *dataset* específico, não existem garantias da consistência dessa performance sobre domínios diferentes. Os *datasets* mais utilizados na literatura (CoNLL, MUC) são produzidos sobre notícias, ou seja, a performance reportada pelo modelo em teste para um *dataset* de notícias pode não se traduzir para aplicações sobre textos sobre domínios específicos, e.g. petróleo, financeira, publicações em redes sociais...

Uma abordagem para aprimorar a projeção de qualidade entre domínios foi apresentada conferência ACE, mencionada em Capítulo 1. Onde os *datasets* estruturaram as entidades em estruturas hierárquicas para as classes das entidades, construindo níveis de especificidade e abstração, e.g. a classe "organização" pode receber os subtipos "governo", "comercial", "educacional", "sem fins lucrativos" e "outro".

As ações da $\underbrace{\text{Petrobras}}_{\text{Empresa de petróleo}}$ e da $\underbrace{\text{Microsoft}}_{\text{Empresa de tecnologia}}$ terminaram o dia em alta.

Exemplo 2.1: Um exemplo de sentença contendo organizações de diferentes tipos, classificadas em seu maior nível de especificidade.

As ações da $\underbrace{\text{Petrobras}}_{\text{Empresa}}$ e da $\underbrace{\text{Microsoft}}_{\text{Empresa}}$ terminaram o dia em alta.

Exemplo 2.2: Um exemplo de sentença contendo organizações de maneira geral.

As ações da Petrobras e da $\underbrace{\text{Microsoft}}_{\text{Empresa de te tecnologia}}$ terminaram o dia em alta.

Exemplo 2.3: Sentença anotada para identificação de empresas de tecnologia.

Por meio da construção de uma ontologia sobre as classes das entidades é possível implementar um corpus flexível em relação ao domínio abordado. Isso é alcançado ao se anotar o corpus respeitando a hierarquia dessa ontologia. Para uma entidade em texto é atribuída a classe com o maior nível de especificidade como ilustrado em Figura 2.1.

Se for necessário ajustar o domínio para empresas no geral, o *datset* pode ser articulado para tal, como ilustrado no Exemplo 2.2.

Outro exemplo seria quando se deseja desenvolver um sistema para identificar apenas as entidades de empresas de tecnologia. Neste caso o corpus seria estruturado como ilustrado no Exemplo 2.3.

Uma abordagem de destaque em especial para redes neurais é o uso de *word embeddings* pré-treinadas, como descrito em Subsubseção 4.2.1.2. Essa técnica permite a extração de relações semânticas e sintáticas entre palavras a partir de corpos de texto não anotados. Os textos utilizados para treino podem ser selecionados os arbitrariamente, ou seja podem ser escolhidos conforme o domínio de aplicação do sistema.

Neste trabalho, endereçamos a projeção da performance entre domínios de três maneiras:

1. Construímos o *dataset* a partir de uma ontologia de entidades.

2. O *dataset* é elaborado a partir de textos de variados domínios, correspondendo aos artigos do Wikipédia.

3. Uso de *word-embeddings* pré-treinadas.

PUC-Rio - Certificação Digital Nº 1612825/CA

2.3.3 Coleta de dados

Existem fontes de dados publicas que visam detalhar objetos de acordo com seus domínios e associá-los a recursos textuais. Exemplos dessas fontes de dados são os bancos de dados públicos DBpedia[9], Wikidata[7] e Google Graph [8] e a enciclopédia eletrônica Wikipédia[24]. A abordagem utilizada neste projeto envolve ligar as informações captadas pelo banco de dados DBpedia com os recursos textuais do Wikipédia. Abordagens similares foram utilizadas em [25] [26].

2.3.4 Wikipédia

O Wikipédia é uma enciclopédia aberta, cooperativa e multilíngue que busca registrar em formato eletrônico conhecimento sobre diversas domínios. A produção de conteúdo dos artigo é realizada de maneira cooperativa no sentido de que correções e aprimoramentos são realizados pelos usuários nos artigos com o intuito de aprimoramento da qualidade.

Embora existam ações de usuários mal intencionados visando vandalizar o conteúdo da página, isso não deve ser considerado um grande empecilho para seu uso para a produção de um corpus de treinamento devido a baixa quantidade de ocorrência desses casos e aos protocolos de detecção e correção de vandalismo. Por exemplo, existe a automatização do processo de detecção de vandalismo [27] e usuários com histórico de edições maliciosas acabam por ter todas as suas edições desfeitas, retornando os artigos a uma versão anterior as modificações.

Uma prática de automatização da melhoria da qualidade é realizada por meio da notificação dos editores de erros gramaticais e inconsistência em padrões e.g. sinais de pontuação, de aspas sem seu par.

Os elementos da estrutura de artigos do Wikipédia que o torna um bom candidato para o uso em construção de corpus anotados são:

1. Volume de recursos textuais construídos por humanos.

2. Variedade de domínios abordados.

3. Sua estrutura de caixas informativas[28], que visa estruturar as informações dos artigos uniformemente de acordo com o tema do artigo.

4. Links internos — *interlinks* — [29] que visam ligar as páginas relacionadas quando mencionadas entre si.

PUC-Rio - Certificação Digital Nº 1612825/CA

Esses elementos relevantes sobre a estrutura do Wikipédia são descritos a seguir em termos de suas particularidades interessantes para a construção do *dataset*.

2.3.4.1 Caixa informativa

As caixas informativas — *infobox* — do Wikipédia são páginas construídas com o objetivo de estruturar informações relevantes sobre o artigo conforme ao tipo atribuído para o mesmo, visando reutilizar a mesma estrutura para artigos semelhantes, e.g. estabelecer um conjunto de propriedades relevantes sobre artigos sobre pessoas como data de nascimento, obra, filhos... Por meio deste elemento é possível ter acesso a seguinte conexão: (1) quais as categorias das páginas e (2) quais os termos pelos quais a categoria é identificada i.e. quais os seus nomes. Um exemplo deste elemento é ilustrado em Figura 2.4.

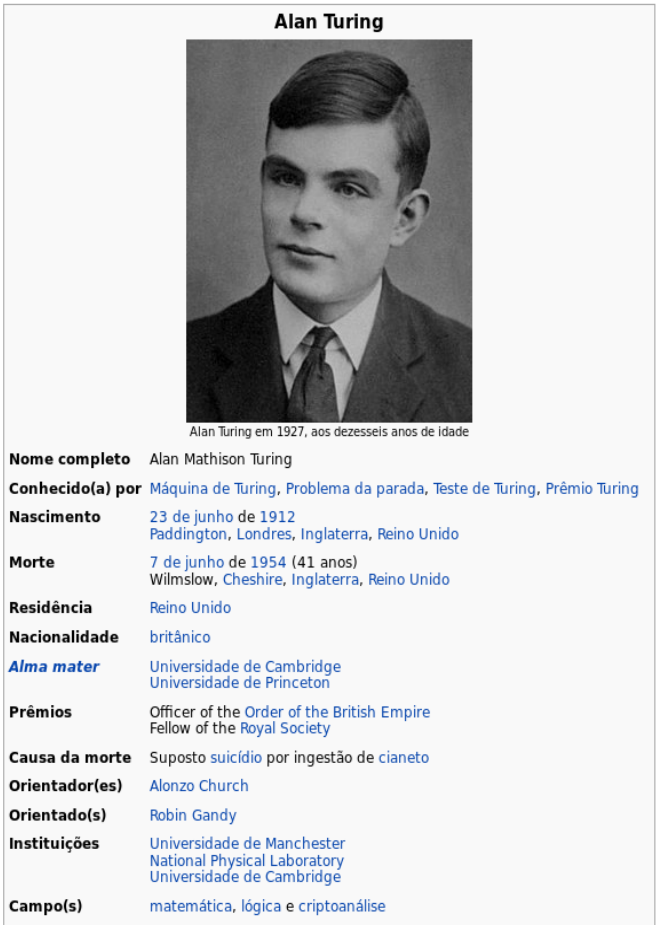


Figura 2.4: Um exemplo de *infobox* para artigos sobre matemáticos. Nela obtemos as informações do matemático Alan Turing.

Ou seja, os artigos do Wikipédia que podem ser explorados para conhe-

PUC-Rio - Certificação Digital Nº 1612825/CA

cer quais as entidades sendo referenciadas em corpos de texto e alguns detalhes relevantes sobre a mesma. Entretanto, é necessário aplicar um processo de exploração manual sobre os tipos de *infobox* registradas pelo Wikipédia e classificá-las dentro do escopo dos tipos de entidades de interesse, recuperar seus artigos, encontrar inconsistências ... Identificar e aplicar técnicas de pré-processamento. Uma versão desta extração foi realizada pelo projeto DBpedia, onde um grupo de pesquisadores extraiu essa estrutura identificando e consertando as inconsistências encontradas, como é descrito em Subseção 2.3.5

2.3.4.2 Ligações internas

Os artigos do Wikipédia são formatados seguindo uma linguagem de marcação chamada wikitexto — *wikitext* — [30]. É por meio desta linguagem que os *interlinks* são definidos no texto do artigo. Para criar um link para outra página, é necessário associar o trecho do texto ao título, obedecendo ao formato:

$$[[\boldsymbol{A} \,|\boldsymbol{B}]]$$

- ***A***: O texto a ser exibido no *link*, no corpo do texto esse segmento será formatado com a cor azul.
- ***B*** : O título da página que será referenciada.

Por exemplo, o trecho em *wikitext* a seguir é exibido como ilustrado na Figura 2.5

"... o matemático [[[Alan Mathison Turing|Alan Turing]] foi um dos pioneiros da ciência da computação ..."

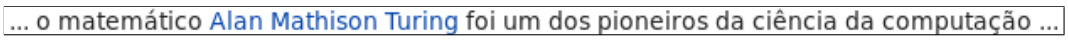


Figura 2.5: Um exemplo que ilustra como ocorrem as ligações entre páginas no Wikipédia. O texto em azul corresponde ao texto do link

.

As ligações internas são úteis neste trabalho para encontrar os artigos que mencionam a entidade, mesmo quando não são o artigo da entidade em si.

PUC-Rio - Certificação Digital Nº 1612825/CA

2.3.5 DBpedia

DBpedia é um projeto cujo objetivo é extrair e estruturar das informações da Wikipédia. Neste projeto é construída uma base de dados que foi modelada a partir dos fundamentos da Web semântica [31] aplicando as especificações RDF (*Resource Description Framework*).

A estruturação das informações do Wikipédia foi realizada sobre a organização dos artigos na forma de uma ontologia [32]. Esse mapeamento em grande escala somente foi possível devido a já mencionada estrutura de caixas informativas, por meio do conhecimento estruturado nelas foi possível, com limitada intervenção humana, extrair uma estrutura hierárquica dos artigos de maneira automatizada.

Embora as caixas informativas sejam o recurso mais relevante no processo de estruturação de informação, existem outros padrões estabelecidos para a estrutura dos artigos que foram levados em conta. Por exemplo, a primeira frase do artigo deve constituir uma definição para o tópico do artigo, formatado com letras em negrito.

A construção da ontologia do DBpedia teve como foco o idioma inglês e as relações extraídas foram projetadas para os demais idiomas. Em suma, a ontologia foi extraída e pré-processada a partir do Wikipédia em inglês e, por meio das ligações interlinguísticas [33], ela foi projetada para outros idiomas. No caso de um artigo presente na ontologia em inglês não em outro idioma e vice-versa, o referido artigo é ignorado.

A relevância da estrutura definida pelo DBpedia neste trabalho está em possibilitar identificação das páginas cujo tópico se encaixa em uma das classes de entidade de interesse, e.g. a página do mencionado matemático Allan Turing caracteriza um texto sobre pessoa. Ademais, são especificados os possíveis nomes das entidades (e.g. nome de solteiro ou apelido), aumentando a quantidade de termos que podem ser buscados no texto a fim de identificar uma entidade.

Outra vantagem desse recurso é o fato de que pré-processamento manual foi realizado pelos membros do projeto (DBpedia) a fim de encontrar todas as conexões pertinentes, redundâncias, sinônimos... Melhorias de qualidade no geral que devem ser realizadas manualmente,

Em suma, é possível extrair um conjunto de entidades a partir do DBpedia onde conhecemos:

- A classe da entidade
- Os termos pelos quais a entidade é referenciada

PUC-Rio - Certificação Digital Nº 1612825/CA

– Qual o artigo do Wikipédia da entidade

2.3.6 Ligando entidades e artigos

Além dos artigos do Wikipédia das entidades conhecidas, é possível, por meio do uso das ligações internas, buscar nos demais artigos, menções adicionais as entidades.

Se um artigo qualquer tiver uma ligação interna para um artigo de outra entidade de interesse, é possível buscar menções a entidade no texto. Além do interesse em aumentar a quantidade de exemplos, por meio desta metodologia é possível diminuir o ruído do *dataset* no que diz respeito as entidades mencionadas presentes no texto não capturadas.

Ademais, é importante mencionar que na estrutura do Wikipédia as recomendações de edição explicitam que apenas a primeira menção a um artigo interno deve constituir uma ligação, independente da existência de outras menções pelo texto. Ou seja, deve existir apenas uma ligação interna para cada artigo referenciado.

Portanto, a fim de maximizar a quantidade de conteúdo extraído do Wikipédia e a qualidade dos exemplos, consideraremos não somente as páginas do Wikipédia que dissertam sobre uma entidade, mas também as páginas que as referenciam.

PUC-Rio - Certificação Digital Nº 1612825/CA

2.3.7 Coleta e estruturação da informação

Nesta secção é descrito o processo de coleta dos dados e sua estruturação em um banco de dados integrado.

2.3.7.1 Coleta das entidades do DBpedia

A coleta dos dados do DBpedia foi realizada utilizando o serviço público de acesso a uma versão da base de dados [34]. A busca foi realizada sobre o escopo de classes de entidade selecionado: pessoas, lacais e organizações. As buscas utilizadas foram definidas utilizando a linguagem SPARQL. A última versão da base de dados foi construída sobre uma versão do Wikipédia de outubro de 2016, ou seja, ass entidades foram estruturadas a partir da versão do Wikipédia do mesmo período.

Nas buscas sobre o banco de dados foram filtradas as seguintes informações sobre as entidades:

– A classe da entidade

– O id da página, também chamado de *wiki id*

– O título da página

– Os nomes da entidade. Neste caso a ontologia varia, de acordo com a classe, em quantas propriedades podem corresponder a nomes. Por exemplo, entidades do tipo lugar não possuem a propriedade de sobrenome - *foaf:surname*.

2.3.7.2 Coleta dos dados do Wikipédia

A partir da versão do Wikipédia em que foi utilizada para a construção do DBpedia (outubro de 2016). Os dados do Wikipédia são disponibilizados por meio de arquivos para *download*, os chamados *dumps*, eles são disponibilizados em um arquivo XML com algumas informações relevantes para este trabalho no que diz respeito aos artigos:

– Título do artigo

– Id do artigo, um número identificador único

– Texto do artigo (no formato wikitexto)

2.3.7.3 Estruturação dos dados

Os dados coletados foram organizados em um banco de dados MongoDB onde as ligações entre entidades e artigos foram modeladas. Na Figura 2.6, a estruturação da base de dados é ilustrado em um diagrama de relações de entidade

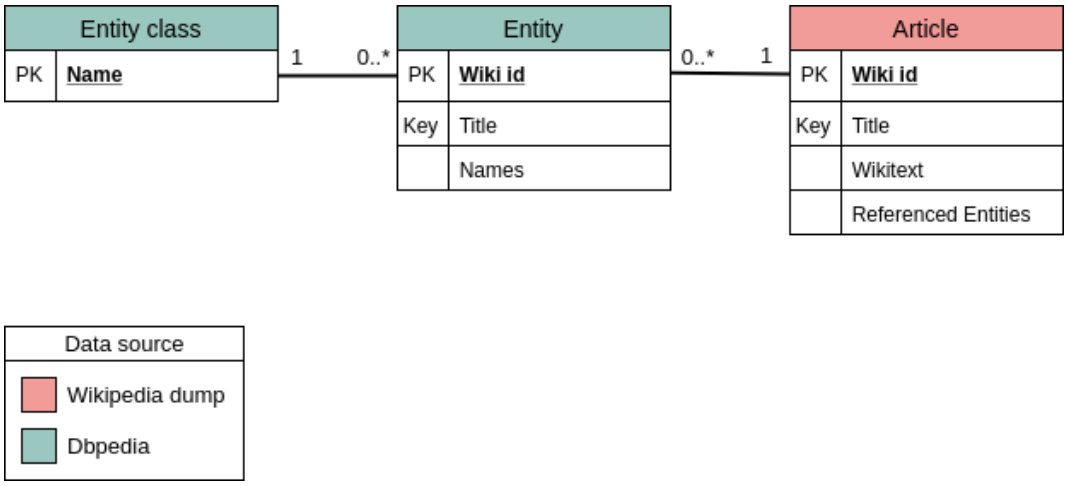


Figura 2.6: Diagrama de relações entre entidades dos dados coletados.

PUC-Rio - Certificação Digital Nº 1612825/CA

2.3.8 Pré-processamento

É necessário pré-processar o texto dos artigos do Wikipédia a fim de estruturar o texto como entrada para modelos de redes neurais. Serão construídas entradas estruturadas para o formato BIO, descrito em 2.1.

As seções seguintes detalham os passos necessários para a estruturação do texto de cada artigo. Em ordem os passos são:

- Remoção dos elementos do texto do artigo, a partir de seu código em Wikitexto, que são desinteressantes para a tarefa a fim de remover o ruído.
- Transformação do Wikitexto em texto bruto.
- Exact matching das menções a entidades. A partir do texto bruto, destacar como menção a entidade as partes do texto iguais a um dos nomes conhecidos das entidades mencionadas no texto.
- Busca por entidades mencionadas que não encontradas.
- Tokenização das frases e palavras do texto.
- Encaixe das menções a entidades nas sentenças e palavras tokenizadas.
- Transformação para IOB.

PUC-Rio - Certificação Digital Nº 1612825/CA

2.3.8.1 Filtro dos elementos do Wikitexto

As partes de interesse do artigo são aquelas que dissertam sobre o tema por meio de um texto estruturado utilizando os elementos gramaticais tradicionais, e.g. parágrafos, frases, palavras... Entretanto, existem instâncias nos textos dos artigos que não se adéquam a uma estrutura padronizada, por exemplo, listas, tabelas, imagens, elementos em notação química, física, matemática... Esses elementso são definidos por meio da linguagem Wikitexto referenciada anteriormente. De uma maneira abrangente, os seguintes elementos do Wikitexto do artigo foram removidos:

- **Listas**, (e.g. elementos *unbulleted list*, *flatlist*,*bulleted list*...)
- **Tabelas** (e.g. *infobox*, *table*, *categorytree*)
- **Arquivos no geral** (e.g. *media*,*arquivo*,*audio*,*video*...)
- **Formatações cientificas** (e.g. *chem*, *graph*, *hiero*, *math*, *score*...)
- Trechos com **identação irregular** (e.g. *outdent*, *outdent2*)

2.3.8.2 Filtro seções

Outra oportunidade para a remoção do ruído é a remoção de secções dos artigos que são padronizadas pelo guida de edição do Wikipédia[35]. Por exemplo, existem secções que se referem as ligações externas, geralmente uma lista de *links* para outras páginas. As secções removidas foram:

- **Referências**: lista das referências realizados em texto.
- **Ver também**: lista de referências que não foram explicitas em texto, correspondendo a páginas com tópicos relacionados.
- **Bibliografia**: lista de fontes relevantes ao tópico.
- **Ligações externas**: *links* para páginas fora do Wikipédia.

2.3.8.3 Transformação do Wikitexto em texto bruto

Após a remoção dos elementos ruidosos deve-se realizar a conversão dos Wikitexto tratado para o formato de texto bruto. Isso é realizado de forma automática pela ferramenta MWparser [36].

2.3.8.4 *Exact matching* das entidades mencionadas

A partir do texto bruto do artigo é possível realizar a busca por menções as entidades uma vez que uma coleção de nomes dessas entidades foi coletada do DBpedia. A identificação das menções é realizada a partir da busca pelos nomes das entidades referenciadas no texto. São marcados apenas os segmentos de texto que são exatamente iguais a um dos nomes conhecidos das entidades.

Para fins ilustrativos, considere o texto T representado em2.7 anotado a partir das entidades $e1$ e $e2$, definidas em Tabela 2.2 e Tabela 2.3.

Entidade 1 (e1)	
Classe	LOC
Nomes	Rio, Rio de Janeiro, Cidade Maravilhosa

Tabela 2.2: Exemplo de entidade do tipo lugar.

PUC-Rio - Certificação Digital Nº 1612825/CA

Entidade 2 (e2)	
Classe	PER
Nomes	João, João Castro, João Castro Almeida

Tabela 2.3: Exemplo de entidade do tipo pessoa.

$\underbrace{\text{João Castro}}_{PER}$ foi ao $\underbrace{\text{Rio de Janeiro}}_{LOC}$. Visitou o Pão de Açúcar.

Exemplo 2.7: Um exemplo de sentença com as menções as entidades conhecidas destacadas.

2.3.8.5 Busca pelas demais entidades mencionadas

Embora tenhamos uma lista de entidades mencionadas no artigo, existe a possibilidade de existirem entidades que não são referenciadas diretamente pela página. No exemplo do texto T, Figura 2.7, o trecho "Pão de Açúcar" é uma entidade do tipo "lugar". Uma maneira de lidar com as entidades não conhecidas, é utilizar um sistema auxiliar de NER para realizar a extração dessas entidades para encontrar esses casos.

O sistema para NER auxiliar utilizado foi o Polyglot [37]. Se trata de um sistema treinado em cima de um *dataset* também construído a partir do Wikipédia.

Para este trabalho adotaremos a seguinte terminologia para referenciar e diferenciar as entidades presentes no *dataset* de treino construído

- **Anotadas** - Oriundas do *exact matching* com o um dos nomes de entidade presentes no DBpedia.
- **Preditas** - Extraídas pelo *Polyglot*.

A partir do texto T com as menções anotadas a por *exact matching* buscamos pelas demais menções a entidades. Um exemplo desta busca sobre o texto T é ilustrado no Exemplo 2.8.

$\underbrace{\text{João Castro}}_{anot-PER}$ foi ao $\underbrace{\underbrace{\text{Rio de Janeiro}}_{anot-LOC}}_{pred-ORG}$. Visitou o $\underbrace{\text{Pão de Açúcar}}_{pred-LOC}$.

Exemplo 2.8: Um exemplo de sentença com tanto as menções de entidades anotadas quento preditas. Neste exemplo existe um conflito entre a entidades anotadas e preditas.

PUC-Rio - Certificação Digital Nº 1612825/CA

Como ilustrado no Exemplo 2.8 existem possíveis conflitos entre os segmentos de entidades preditas e anotadas. Para desambiguar esses conflitos priorizam-se as entidades anotadas. Afinal, é atribuído um grau de confiança maior para as entidades anotadas em virtude de serem estimadas a partir das entidades do DBpedia e ligações internas do artigo, ambas produzidas com supervisão humana. Após a desambiguação, temos que o texto T adota o formato ilustrado no Exemplo 2.9.

$$\underbrace{\text{João Castro}}_{anot-PER} \text{ foi ao } \underbrace{\text{Rio de Janeiro}}_{anot-LOC}\text{. Visitou o } \underbrace{\text{Pão de Açúcar}}_{pred-LOC}\text{.}$$

Exemplo 2.9: O texto T após a desambiguação dos conflitos.

2.3.8.6 Tokenização das palavras e sentenças

Para estruturar o texto no formato BIO é necessário conhecer as delimitações das palavras e frases. Esse processo de tokenização foi realizado no texto a partir do conjunto de ferramentas NLTK (*The Natural Language Toolkit*)[38], especificamente utilizamos a ferramenta "Punkt" do pacote para tokenização das sentenças e palavras. A ferramenta Punkt implementa um algoritmo multilíngue, não supervisionado[39].

O texto T com os *tokens* de entidades palavras e frases é ilustrado no Exemplo 2.10, onde w_i correspondem aos *tokens* de palavras e s_i correspondem aos *tokens* de sentenças.

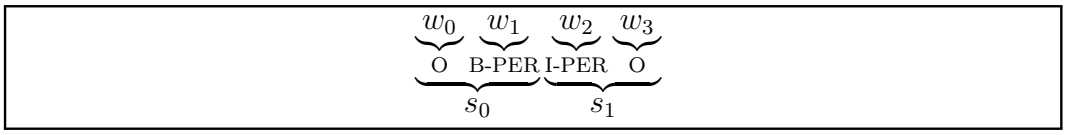
João	Castro	foi	ao	Rio	de	Janeiro	.	Visitou	o	Pão	de	Acúcar	.
B-PER	I-PER			B-LOC	I-LOC	I-LOC				B-LOC	I-LOC	I-LOC	
w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}
							s_0						s_1

Exemplo 2.10: O texto T e seus *tokens* de entidades, palavras e sentenças.

Em alguns casos pode haver incompatibilidade entre os *tokens* de entidade e de palavras e frases. A seguir descrevemos os dois casos em que essas incompatibilidades ocorrem e quais as medidas tomadas para a resolução.

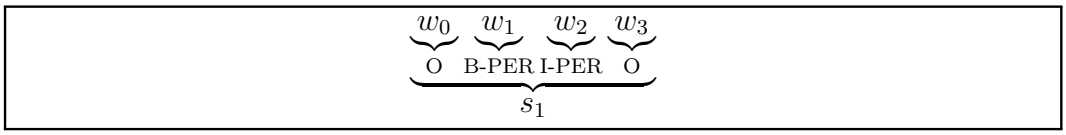
(1) *Tokens* de sentenças que "cortam" uma entidade mencionada. Não é correto que uma entidade mencionada se encontre entre duas sentenças diferentes. Um exemplo deste caso é ilustrado no Exemplo 2.11.

PUC-Rio - Certificação Digital Nº 1612825/CA



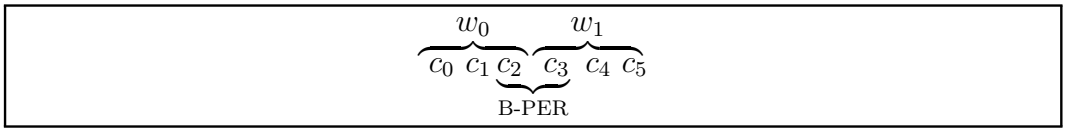
Exemplo 2.11: Conflito entre os *tokens* de entidade e sentenças.

Neste exemplo, temos que w_0 e w_1 compõem uma menção a entidade que se encontra entre as sentenças $s_0 e s_1$. Nessas circunstâncias realizamos a concatenação das sentenças que contêm a entidade até que a menção se encontre em apenas uma sentença. O resultado desta concatenação para o exemplo atual é o exibido em Exemplo 2.12.



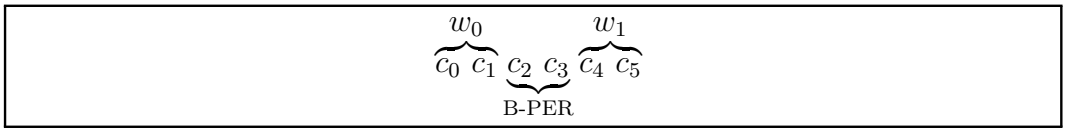
Exemplo 2.12: Resolução do conflito entre os *tokens* de entidade e sentenças por meio da concatenação de sentenças.

(2) *Tokens* de palavras que "cortam" uma entidade mencionada. Não é possível que *tokens* de palavras e *tokens* de entidades ocupem a mesma posição na sequências de caracteres. Um exemplo deste caso é ilustrado no Exemplo 2.13.



Exemplo 2.13: Conflito entre os *tokens* de entidade e palavras.

Nesses casos, as palavras cujos caracteres conflitam com os caracteres das entidades encontradas são removidos do escopo da palavra. No exemplo em questão, obtém-se:



Exemplo 2.14: Resolução do conflito entre os *tokens* de entidade e palavras por meio do "corte" das palavras nos limites da entidade.

PUC-Rio - Certificação Digital Nº 1612825/CA

2.3.8.7 Estruturação em BIO

Por fim, ao se considerar as palavras que não estejam dentro dos limites de uma entidade como "O" (*outside*) teremos um arquivo no formato BIO como ilustrado no Exemplo 2.4.

João	B-PER
Castro	I-PER
foi	O
ao	O
Rio	B-LOC
de	I-LOC
Janeiro	I-LOC
.	O
Visitou	O
o	O
Pão	B-LOC
de	I-LOC
Açúcar	I-LOC
.	O

Exemplo 2.4: Estrutura BIO final do texto T.

2.3.9 Resultados da extração

Realizado o pré processamento dos artigos, avaliaremos as características do corpus resultante. A finalidade é aplicar uma inspeção que auxilie, ao mesmo tempo: (1) acusar inconsistências no corpus, e.g. tamanhos de sentenças, (2) informações relevantes para a calibração e avaliação da performance dos modelos e.g. proporções das quantidades de tipos de entidade, tamanho das entidades e origem anotadas a partir das entidades extraídas do DBpedia ou preditas a partir do parser utilizado.

A seguir, as seções exploram métricas importantes sobre os dados coletados. O propósito desta análise é embasar e compreender a performance dos modelos explorados em Seção 4.3. Serão consideradas apenas as sentenças que possuem entidades anotadas. Afinal, considerar sentenças que possuem apenas entidades oriundas da extração do parser não representa uma metodologia que aproveita o discernimento humano investido na estruturação dos dados do DBpedia.

PUC-Rio - Certificação Digital Nº 1612825/CA

Após a apresentação das métricas, são discutidas medidas pertinentes para mitigar possíveis anomalias e desbalanceamentos.1

2.4 Sentenças

A primeira métrica relevante corresponde ao número total de sentenças extraídas e as suas dimensões em termos da quantidade de *tokens*.

A quantidade total de sentenças extraídas é **3.650.909**

O tamanho das sentenças é uma métrica importante para o processo de modelagem. As camadas de *input* dos modelos explorados que correspondem a redes neurais recorrentes estão sujeitas a impactos negativos em sua performance ao se considerar sequências de grandes dimensões os chamados *vanishing and exploding gradient*, como será descrito na seção Subseção 4.1.1. O tamanho das sentenças se apresenta como ilustrado em Figura 2.15. Um detalhamento de métricas estatísticas relevantes é ilustrado em Tabela 2.5.

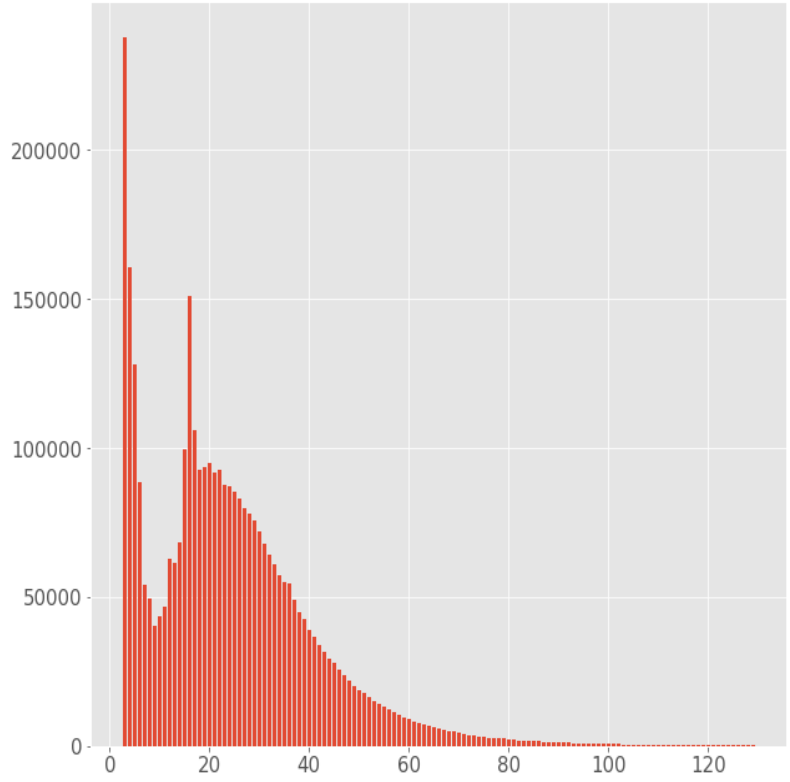


Figura 2.15: Tamanho das sentenças. Para fins de legibilidade consideram-se apenas aquelas até 120 palavras.

A primeira inconsistência acusada sobre o tamanho das sentenças é o tamanho da maior sentença que possui 8437 palavras, uma valor extremamente superior ao tamanho médio das sentenças. Para lidar com este problema,

PUC-Rio - Certificação Digital Nº 1612825/CA

Métrica	
Total	3650909
μ Média	24.04
σ Desvio padrão	20.41
Mínimo	1
Máximo	8437
Q_1 Percentil 25%	11
Q_2 Percentil 50%	21
Q_3 Percentil 75%	33

Tabela 2.5: Métricas sobre a quantidade de *tokens* das sentenças.

aplicamos um filtro sobre as sentenças, excluíndo todas aquelas com um tamanho superior a 200 *tokens*. Esse valor foi selecionado por ser próximo ao tamanho da maior sentença extraída dos *dataset* das edições do HAREM utilizadas neste projeto, Seção 2.7, que possui 183 *tokens*. As seções a seguir deste capítulo, apresentam estatísticas sobre o conjunto de sentenças após a aplicação do filtro.

PUC-Rio - Certificação Digital Nº 1612825/CA

2.5 Classes de *tokens*

A proporção das classes dos *tokens* encontrados é uma métrica relevante. Para compreender o impacto de um possível desbalanceamento na qualidade dos modelos. As proporções entre os tipos seguem o gráfico definido em Figura 2.16 e Tabela 2.6.

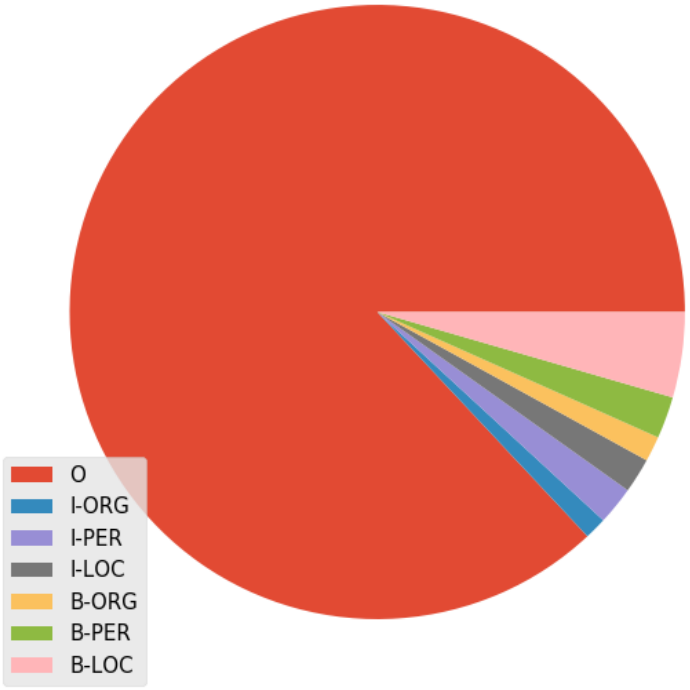


Figura 2.16: Proporção entre os tipos de *token*.

Classe	Total
O	86.97%
I-ORG	1.16%
I-PER	2.01%
I-LOC	1.84%
B-ORG	1.33%
B-PER	2.20%
B-LOC	4.49%

Tabela 2.6: Proporções entre todas as classes de *token*.

A quantidade de palavras que não compõem entidades é muito superior, mesmo após a remoção das sentenças que não contêm nenhuma entidade. O desbalanceamento dos *tokens* que compõem menções a entidades é ilustrado em Figura 2.17, e detalhado em Tabela 2.7.

Os gráficos ilustrados em Figura 2.17 e Tabela 2.7 expressam a relação entre a quantidade de *tokens* de entidades.

PUC-Rio - Certificação Digital Nº 1612825/CA

Classe	%
I-ORG	8.88%
I-PER	15.42%
I-LOC	14.12%
B-ORG	10.21%
B-PER	16.93%
B-LOC	34.44%

Tabela 2.7: Proporções entre as classes de *tokens* considerando apenas aquelas que pertencem a entidades.

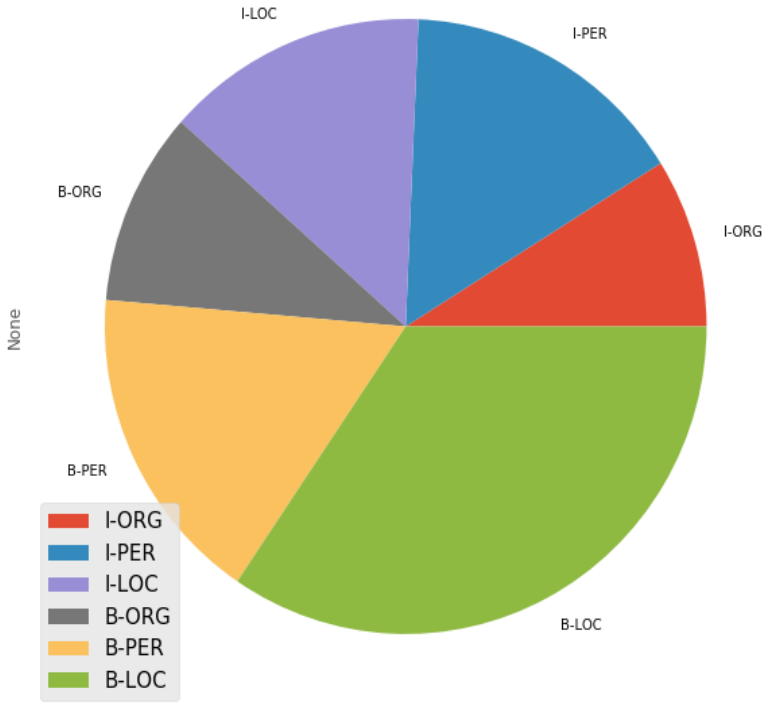


Figura 2.17: Proporção entre os tipos de *token* de entidades.

A quantidade de *tokens* de cada classe está bastante desbalanceada. Em especial os *tokens* referentes a classe "LOCAL" é muito superior aos das demais classes. Esse tipo de desbalanceamento pode apresentar repercussões negativas sobre a performance do modelo em diferentes classes. Para averiguar o impacto deste desbalanceamento, construímos uma versão do *dataset* balanceada em termos da proporção entre *tokens* de diferentes tipos. como descrito na seção a seguir.

PUC-Rio - Certificação Digital Nº 1612825/CA

2.6 Balanceamento

Nesta seção descrevemos como realizamos o processo de balanceamento dos *tokens* referentes a menções a entidades resultando em uma segunda versão do *dataset*.

Iremos adotar uma metodologia onde filtraremos sentenças até que a quantidade total de menções a entidade de todos os tipos estejam próximas. Em outras palavras, igualando a quantidade de menções de todas as classes com a de menor ocorrência, "ORG".

Uma menção a entidade pode ser identificada por meio da presença de um *token* com o prefixo "B" (*begin*), portanto buscaremos balancear a quantidade de *tokens* de cada classe com este prefixo.

Buscamos entender quantos dos *tokens* são oriundos das entidades do DBpedia (*tokens* anotados) e quantos são fruto da predição realizada pelo parser auxiliar (*tokens* preditos). Isso se deve ao fato de que buscamos focar em remover sentenças em que existem mais *tokens* preditos pelo parser auxiliar. Essa proporção entre *tokens* preditos e anotados é ilustrada em Figura 2.18.

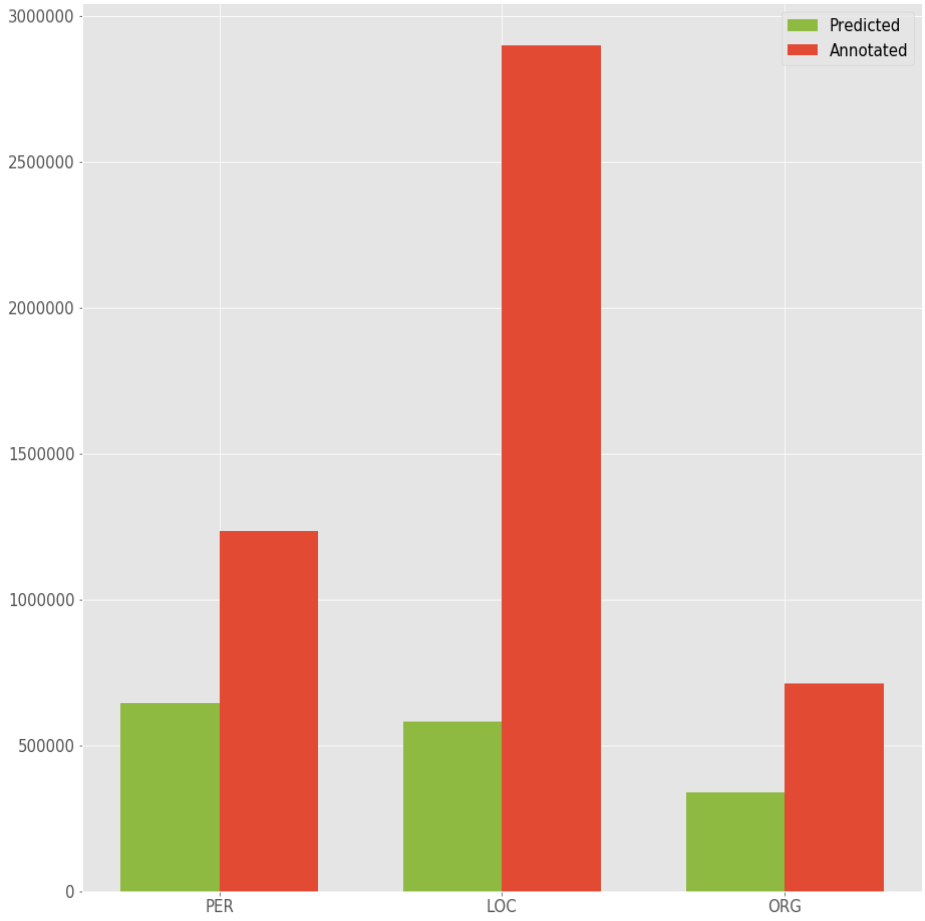


Figura 2.18: Proporção entre as origens dos *tokens*.

PUC-Rio - Certificação Digital Nº 1612825/CA

Atingimos uma aproximação satisfatória por meio da aplicação do mesmo filtro para "PER" e "LOC" individualmente. Basicamente:

1. Realizamos uma busca pelas sentenças cujas entidades anotadas são apenas do tipo de interesse, "PER" ou "LOC".

2. Priorizamos o filtro de sentenças onde há uma maior quantidade de exemplos preditos dessa mesma classe de interesse. Em outras palavras, ao se filtrar sentenças com exemplos anotados do tipo "PER", por exemplo iremos priorizar aquelas com a maior quantidade de entidades "PER' preditas pelo parser auxiliar.

3. Removemos as sentenças resultado da busca até a quantidade de *tokens* totais da classe de interesse se aproximar a quantidade da classe de menor proporção.

O *dataset* resultante desta extração é substancialmente menor. Houve uma redução de 66% nas sentenças, totalizando **1.216.976**. As proporções entre as classes de entidades e suas origens são ilustradas em Figura 2.19 e Figura 2.20. Um detalhamento adicional sobre as proporções dos tipos de *token* presentes em menções a entidades é apresentado em Tabela 2.8.

PUC-Rio - Certificação Digital Nº 1612825/CA

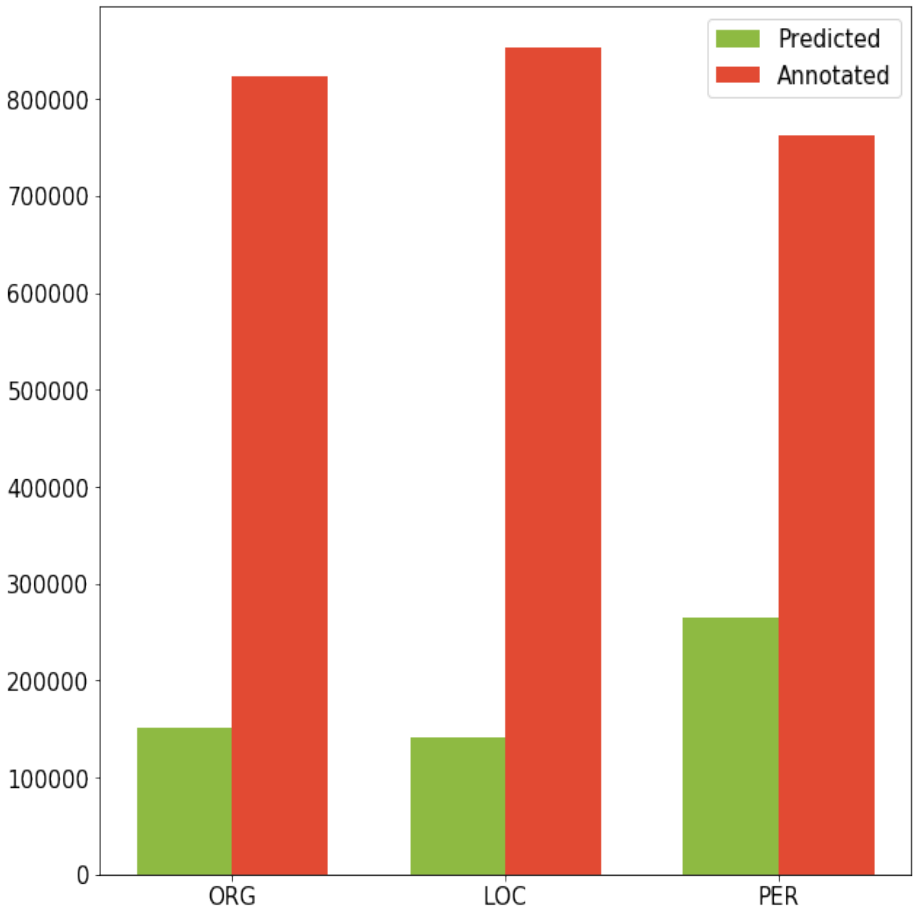


Figura 2.19: Proporção entre as origens dos *tokens* no *dataset* filtrado.

PUC-Rio - Certificação Digital Nº 1612825/CA

Classe	%
I-ORG	15.76%
I-PER	17.39%
I-LOC	14.60%
B-ORG	17.00%
B-PER	17.93%
B-LOC	17.33%

Tabela 2.8: Proporções entre as classes de *tokens* considerando apenas aquelas que pertencem a entidades.

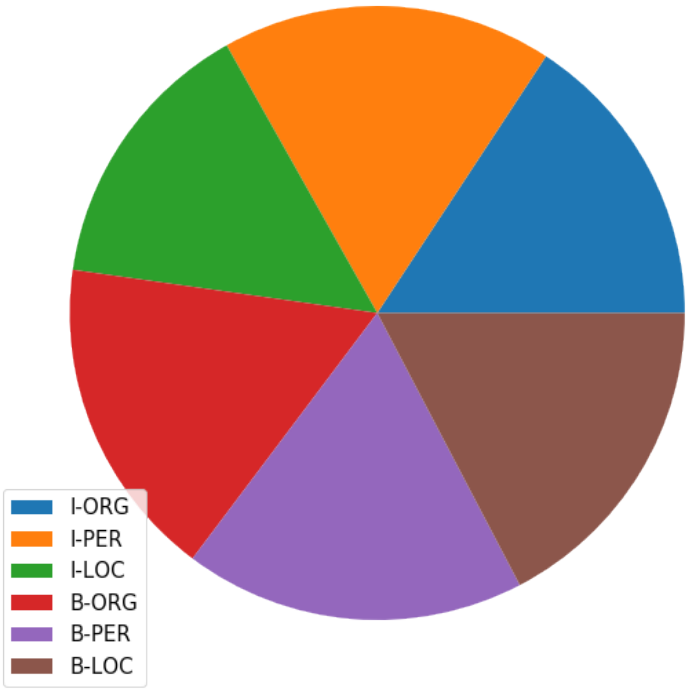


Figura 2.20: Proporção entre os tipos de *token* de entidades no *dataset* filtrado.

PUC-Rio - Certificação Digital Nº 1612825/CA

2.7 Conjuntos de calibração e teste

Os corpus HAREM e miniHarem produzidos pelo projeto Linguateca[15] são utilizados como *gold standard* para teste e calibração. As proporções de cada um dos corpus reservadas para calibração e teste são:

- Calibração: 20% do corpus HAREM
- Teste: 80% do corpus HAREM e 100% do mini HAREM

Entretanto, existem questões de compatibilidade com relação as estruturas dos corpus do projeto HAREM e o corpus produzido neste trabalho.

Essas incompatibilidades devem ser detectadas e entendidas a fim de entender os impactos na performance do uso destes recursos possa ser realizado sob o contexto da estrutura utilizada para treino.

2.7.1 Classes e categorias

Os *datasets* do projeto HAREM possuem uma quantidade de classes de entidade superior ao escopo delimitado para este trabalho. As classes também são categorizadas com tipos. A classe "ORGANIZACAO" pode receber, por exemplo, os tipos o "EMPRESA", "INSTITUICAO" e "ADMINISTRACAO".

Selecionaremos apenas as entidades das classes desejadas (organização, pessoa e local). Entretanto existam casos, mesmo dentro das classes de interesse, que não são pertinentes para o experimento. Isso se deve ao fato que nem todos os tipos dentro do escopo de uma classe de interesse foram abordados pelo corpus construído.

A seleção das entidades mencionadas para cada classe seguiu os seguintes critérios:

PUC-Rio - Certificação Digital Nº 1612825/CA

2.7.2 Entidades da classe local

Para a classe "LOCAL" são considerados todos os tipos exceto "VIRTUAL" que diz respeito a endereços de sites na WEB e.g. "http:www.inf.puc-rio.br". E exclusão desse tipo levou em conta três questões principais: (1) a estrutura gramatical destas ocorrências não são compatíveis com as presentes no corpus de treinamento, afinal, foram anotados locais físicos, (2) a estrutura de links não foi considerada em nenhum caso do corpus de treino produzido. (3) O tipo "VIRTUAL" deixaram de ser considerados EM a partir do evento do miniHAREM.

2.7.3 Entidades da classe pessoa

Para as classes do tipo "PESSOA", nem todos os tipo são compatíveis com a estrutura do *dataset* construído. No exemplo a seguir, extraído do corpus miniHAREM, "Conselho de Mafra" tem o tipo "PESSOA", entretanto, nossas diretivas de anotação para a construção do *dataset* buscam apenas uma equivalência exata entre o nome da entidade e o segmento do texto. Ou seja, neste caso se tivéssemos conhecimento de que existe uma entidade com o nome "Concelho de Mafra" a classe atribuída seria "ORGANIZACAO" e não "PESSOA'.

```
... gostaria de ouvir o
<PESSOA TIPO="GRUPOCARGO" MORF="M,S">
␣␣␣␣␣␣␣␣Concelho␣de␣Mafra
</PESSOA>
sobre um assunto ...
```

Isso se deve ao fato de que as anotações do HAREM consideram o papel semântico das menções. Para a classe "PESSOA", apenas o tipo "INDIVIDUAL" é compatível com o corpus produzido. Nos experimentos (4.3) averiguamos qual o impacto de se utilizar apenas o tipo "INDIVIDUAL" e ao se utilizar todos os tipos da classe.

2.7.4 Entidades da classe organização

Para a classe "ORGANIZACAO" serão consideradas todos os tipos.

2.7.5 Possíveis inconsistências entre os corpus teste e avaliação

Mesmo com os filtros aplicados sobre as classes, existem outras fontes de inconsistência entre o corpus construído a partir do Wikipédia e o HAREM. Nesta secção serão elicitados alguns exemplos.

No caso das classes "PESSOA" os títulos devem ser incluídos na menção, e.g. "tio João", "presidente Vargas". Nestes casos, não existe uma equivalência com o estilo de anotação para o corpus construído, afinal, o corpus de treino é construído sobre menções exatas a nomes. Um exemplo deste caso extraído do *dataset* Primeiro HAREM é o Exemplo 2.2.

```
... instantes largos,
<PESSOA TIPO="INDIVIDUAL" MORF="F,S">
␣␣␣␣␣␣␣␣tio␣Cosme
</PESSOA>
enfeixava todas ...
```

Exemplo 2.2: Exemplo de entidade da classe "PESSOA" incluindo o título. "tio".

Para classes do tipo "ABSTRACCAO" são incluídos menções a nomes. Entretanto, novamente, como a construção do corpus é realizada sobre menções exatas aos nomes da entidade, exemplos da classe "ABSTRACCAO" que fazem menção a nomes seriam classes de outras entidades no corpus produzido. Um exemplo extraído do Primeiro HAREM é o Exemplo 2.3.

PUC-Rio - Certificação Digital Nº 1612825/CA

```
... Hoje, o nome
<ABSTRACCAO TIPO="NOME" MORF="F,S">
␣␣␣␣␣␣␣␣Intelbras
</ABSTRACCAO>
é uma marca de qualidade ...
```

Exemplo 2.3: Exemplo de uma entidade da classe “ABSTRACAO” que faz referência a um nome.

Há casos especiais em que existem diferentes opções de classes possíveis para uma entidade. Esses casos ocorrem de duas maneiras diferentes, descritas a seguir.

Primeira maneira. No caso em que entidades opções de classe diferentes individualmente. No 2.4 é ilustrado um exemplo deste caso. No exemplo, a entidade “Bombeiros” pode ser de dois tipos: (1) classe "PESSOA"do tipo “GRUPO” ou (2) classe “ORGANIZACAO” com categoria “INSTITUICAO”.

```
<PESSOA|ORGANIZACAO TIPO="GRUPO|INSTITUICAO">
␣␣␣␣Bombeiros
</PESSOA|ORGANIZACAO>
```

Exemplo 2.4: Exemplo de diversas opções de classe para uma entidade.

Segunda maneira No caso de marcações do tipo “ALT” o mesmo trecho pode conter diferentes opções de entidades em diferentes locais do mesmo trecho. No 2.5 o trecho “Monárquica” pode ser da classe “PESSOA” com a categoria “GRUPOMEMBRO” ou pode ser um trecho sem classe nenhuma.

```
    <ALT>
␣␣␣␣␣␣␣␣Monárquica|
␣␣␣␣␣␣␣␣<PESSOA TIPO="GRUPOMEMBRO" MORF="F,S">
␣␣␣␣␣␣␣␣␣␣␣␣Monárquica
␣␣␣␣␣␣␣␣</PESSOA>
␣␣␣␣</ALT>
```

Exemplo 2.5: Exemplo da linguagem de marcação utilizada na estrutura do Primeiro Harem.

PUC-Rio - Certificação Digital Nº 1612825/CA

3
Avaliação de performance

Avaliação de performance é um elemento crucial para se construir a compreensão sobre a qualidade dos sistemas desenvolvidos. O desenvolvimento cientifico deve ser sujeito a métricas objetivas para análise de resultados para que seja possível comparar os sistemas desenvolvidos em suas características relevantes.

Para NER, exitem três medidas principais, precisão (*Precision*), revocação/abrangência (*Recall*) e mediada F *F-measure*. Elas são computadas sobre da quantidade de resultados que se encaixam nas seguintes categorias:

- (**P**) **Positivo**: classificação de um caso positivo como tal, e.g. delimitou e classificou corretamente a entidade em texto.
- (**FP**) **Falso Positivo**: classificação incorreta de um caso positivo como negativo.
- (**N**) **Negativo**: classificação correta de um caso negativo, e.g. não detectou entidades em um segmento de texto que, de fato, não as contêm.
- (**FN**) **Falso Negativo**: classificação incorreta de um caso negativo como positivo.

PUC-Rio - Certificação Digital Nº 1612825/CA

Precision avalia quantos dos resultados avaliados como positivos são verdadeiramente positivos. É uma medida que busca definir a confiança sobre a qualidade dos resultados positivos identificados.

$$Precision = \frac{P}{P + FP}$$

Recall avalia a cobertura dos resultados, i.e. quantos positivos (P) do total de casos existentes no *dataset* foram encontrados.

$$Recall = \frac{P}{P + FN}$$

F-measure é uma métrica que avalia *Precision* e *Recall* em conjunto. É realizada a média harmônica ponderada sobre ambas as mediadas.

$$F_\beta = \frac{(\beta^2 + 1) * Precision * Recall}{\beta^2 * Precision + Recall}$$

Na literatura, com a finalidade de considerar os efeitos de *Precision* e *Recall* igualmente, a convenção a é utilizar $\beta = 1$. Resultando em:

$$F_{\beta=1} = \frac{2 * Precision * Recall}{Precision + Recall}$$

Para $\beta = 1$ temos $0 \leqslant F_{\beta=1} \leqslant 1 | F_{\beta=1} \in \mathbb{R}$.

Existem divergências quanto as técnicas utilizadas para avaliação de performance. Por exemplo, deve-se considerar uma entidade destacada corretamente em texto mas classificada incorretamente como positivo? E se a delimitação for correta parcialmente?

A seguir serão descritas algumas características das metodologias de avaliação relevantes para este trabalho.

3.1 Avaliação CoNLL 2002

Na conferência CoNLL em 2002 foi introduzida a abordagem por meio de "*exact matching*"("correspondência exata"). Se trata de uma abordagem direta onde um resultado somente é considerado como positivo se houver a correspondência exata da entidade, tanto em termos de tokenização como de classificação. Esta corresponde a técnica mais utilizada pela literatura. Em suma aplica a medida $F_{\beta=1}$ diretamente.

3.2 Avaliação HAREM

Nos eventos de avaliação conjunta do HAREM a metodologia de avaliação foi utilizada considerando os detalhes do corpus produzido.

1. Devido a estrutura do corpus HAREM quanto a possibilidade de uma entidade com várias classes, ao realizar a avaliação, o resultado deve se adequar a qualquer uma das possíveis opções, ou seja, a mesma parte do texto pode considerar diferentes opções de predição.

2. Além da classe da entidade, adiciona informações sobre o tipo da classe

3. Adiciona informações sobre a morfologia da entidade, quando apropriado, como gênero (masculino, feminino ou indefinido) e numero (singular, plural ou indefinido)

4. Os artigos são classificados quanto as suas fontes e o dialeto do português (e.g. Portugal, Brasil..)

PUC-Rio - Certificação Digital Nº 1612825/CA

Esse nível de complexidade deu origem a uma metodologia de avaliação que é bastante ajustável para avaliar apenas os elementos do corpus que o sistema se propõe e usar. É possível calibrar o avaliador dentro de uma grande gama de possibilidades, por exemplo, avaliando o sistema apenas para os textos de português de Portugal considerando a classe "PESSOA" do tipo "CARGO". Neste trabalho, simplesmente selecionamos o modo de avaliação semântico da ferramenta de avaliação do Primeiro HAREM que considera apenas a classe das entidades.

Ademais, a metodologia de pontuação do HAREM é sensível a acertos parciais, incorporando esses casos na medida $F_{\beta=1}$.

PUC-Rio - Certificação Digital Nº 1612825/CA

4
Experimentos

Nesta seção são apresentados os experimentos realizados.

– Em Seção 4.1 é descrita a evolução das técnicas utilizadas para o desenvolvimento de sistemas para NER com ênfase nos artigos e eventos mais relevantes para este trabalho.
– EmSeção 4.2 são descritos os componentes fundamentais das arquiteturas de redes neurais com os melhores resultados atuais para NER.
– Em Seção 4.3 são apresentados os experimentos realizados e o processo de calibração e modificação das arquiteturas.

PUC-Rio - Certificação Digital Nº 1612825/CA

4.1
História das abordagens voltadas para NER

Nesta seção será descrita a história dos sistemas voltados para NER , com ênfase nas conferências e publicações de maior importância para este trabalho. Mais especificamente, será descrita a história recente de abordagens para NER pelo uso de redes neurais e, por fim, quais os sistemas e resultados relevantes para o português.

4.1.1
Redes neurais

As abordagens iniciais para as tarefas de NLP são muito dependentes de recursos *ad-hoc* , extraídos por experts, e.g. dicionários contendo informações morfológicas sobre palavras, seus valores gramaticais, casos específicos para figuras de linguagem... Tendo em vista a aplicação para a qual o sistema é destinado, (e.g. aplicações para documentos da área de petróleo na língua finlandesa) é necessária a alocação de recursos na construção do ferramental necessário.

Algumas técnicas utilizadas para NLP nesse contexto são:

– Naïve Bayes
– Conditional Random Field (CRF)
– Support Vector Machine (SVM)

– Hidden Markov Model (HMM)

– Maximum Entropy (MaxEnt)

Dada a motivação de se evitar a dependência de intervenção humana , abordagens como *Entropy Guided Transformation Learning* [40] e *word embeddings* foram elaboradas.

4.1.1.1 Language Processing Almost From Scratch

A ênfase atual para modelos de redes neurais em NLP se deu inicialmente devido ao trabalho *Natural Language Processing Almost From Scratch* [41] no qual o elemento chave foi ganho de performance acompanhada da redução da necessidade da produção *features* específicas sobre o texto. Isso se deve ao uso de *word embeddings*, que basicamente trama as relações semânticas e sintáticas entre palavras em um espaço vetorial de dimensão arbitrária a partir da disposição destas em corpos de texto. Em suma, as relações entre palavras são mapeadas em vetores de dimensões arbitrárias, possibilitado simultaneamente a estruturação das palavras como entrada para as redes em representações menos esparsas e mais expressivas.

Ao se utilizar as *word embeddings* que foram construídas sobre corpos de textos dissertando sobre variados domínios é possível, além do ganho de performance sobre *datasets* do experimento, expandir a qualidade dos resultados para domínios contemplados pelas *embeddings* e não pelos *datasets* de treino.

Entretanto, o modelo de Collobert possui dois pontos fracos significativos:

1. O uso de uma arquitetura *feed foward neural network* que restringe o contexto das frases a uma janela em torno de cada palavra. Ou seja, ignoram-se as relações entre palavras fora do contexto da janela.

2. Não realiza aprendizado sobre os caracteres das palavras, desconsiderando importantes informações morfológicas, e.g. lista de prefixos e sufixos, o que é essencial para tarefas como *Pos-tagging*. Essas *features* morfológicas devem ser injetadas manualmente.

4.1.1.2 CharRNN

A rede de Collobert foi adaptada para utilizar no aprendizado informações morfológicas das palavras por meio da arquitetura CharWNN[42]. Nesta adaptação, uma rede convolucional foi utilizada para extrair características

PUC-Rio - Certificação Digital Nº 1612825/CA

morfológicas das palavras automaticamente. Apresentou um grandes ganho de performance para a tarefa de *part of speech tagging* que depende fortemente dessas características morfológicas. Quando aplicado para NER em português apresentou um novo estado da arte sobre o corpus HAREM I [20]. Entretanto, a distância entre as dependências se mantiveram dependentes de uma janela em torno de cada palavras.

4.1.1.3 Bi-directional LSTMs

Redes neurais recorrentes —*Recurrent Neural Networks*— [43] podem ser utilizadas para o fim de detectar dependências na sequência como um todo (e.g. sequência de palavras ou caracteres). O princípio por trás deste modelo é a transmissão do estado interno de um elemento da sequência ao adjacente, em uma direção definida. Portanto, essa transmissão de estado pode ser utilizada para fins de detectar relações entre palavras em qualquer posição da sentença. Entretanto, as versões iniciais deste modelo estão suscetíveis a dois problemas fundamentais quando aplicado para tarefas de NLP:

1. A grandes sequências estão predispostas a apresentar o *vanishing gradient problem* e *exploding gradient problem*[44], dificultando o aprendizado de dependências de longas distâncias.

2. O contexto da sequência está limitado a uma direção, (i.e. um elemento da sequência recebe o estado de elementos anteriores e não posteriores ou vice-versa). Para tarefas de NLP isso não é apropriado, afinal, as relações gramaticais das palavras podem estar sujeitas a informações em ambas as direções da sentença.

Ambos problemas tiveram abordagens desenvolvidas por meio de modificações na estrutura da LSTM, como descrito a seguir.

Tendo em mente o primeiro problema, magnitude dos gradientes transmitidos, foram desenvolvidas as LSTM —*long-short term units*— [45], unidades de redes neurais recorrentes que possuem uma estrutura de *gates* que possibilitam o aprendizado de dependências de grandes distâncias. Basicamente existe um estado comum transmitido para todas as unidades da rede, esse estado é manipulado a cada passo a fim de ser enfatizado apropriadamente.

Para satisfazer a necessidade de uso da sequencia em ambas as direções foram desenvolvidas RNNs capazes de compartilhar a transmissão do estado em ambas as direções [46]. Em suma duas LSTMs são aplicadas para a sequência, cada um em uma direção diferente. Essa abordagem foi introduzida

PUC-Rio - Certificação Digital Nº 1612825/CA

em NLP [47] por justamente expressar o conteúdo da sentença em ambas as direções.

4.1.1.4 Bi-directional LSTM-CNNs

É possível combinar os seguintes elementos já mencionados elementos: (1)LSTMs bidirecionais e (2)inserção de informações morfológicas tal como realizado para a arquitetura CharWNN. Essa arquitetura foi proposta e aplicada para NER em [19].

4.1.1.5 Bi-directional LSTM-CNN-CRF

Apesar do compartilhamento do estado da LSTM bidirecionalmente, NER é um problema de rotulamento de sequências — *sequence labeling* — que, apresenta uma estrutura sintática. Por exemplo, como uma entidade é uma unidade atômica presente em uma sentença, temos que na notação BIO, nenhuma entidade classificada deve iniciar com um *token* do tipo *inside* (I), o tipo correto seria *begin* (B). Portanto é pertinente injetar a informação referente as classificações dos *tokens* adjacentes para que essa estrutura sintática seja destacada.

No artigo *Bidirectional LSTM-CRF Models for Sequence Tagging* [48] foi a proposta uma camada para a rede neural incorporando *Conditional Random Fields* (CRF) . Essa última camada conecta consecutivamente os *outputs* com a finalidade de realizar a classificação incorporando as predições sobre os elementos adjacentes, destacando a presença de estruturas sintáticas.

4.1.1.6 Variações

Atualmente, as arquiteturas de maior notoriedade seguem os fundamentos das arquiteturas descritas nesta seção. Destaco as seguintes variações sobre as arquiteturas fundamentais descritas:

- Uso de *residual connections* [49], para disponibilizar a entrada da rede em todos os níveis,i.e. todas as camadas conhecem o *input* da rede.
- Alterações na técnica de transmissão de estado entre as unidades da LSTM[50] a fim de aproveitar características da estrutura de uma sentença.
- Arquitetura genérica que visa ser calibrada de forma a se o mais independe possível de quaisquer recurso além de estrutura inalterada da sentença [51].

PUC-Rio - Certificação Digital Nº 1612825/CA

4.1.2 Performance HAREM

Nesta seção são explorados os sistemas que obtiveram as melhores performances no HAREM I seguindo a metodologia de avaliação descrita em . Também serão exploradas as abordagens que buscaram utilizar uma combinação entre corpus criados via Wikipédia e os corpus do projeto HAREM.

Como a metodologia de avaliação do Primeiro HAREM pode adotar diversas configurações diferentes, iremos adotar mais pertine para a tarefa atual e situar nossos resultados entre os outros sistemas que foram avaliados da mesma maneira. Em suma a configuração em questão lida com sistemas avaliados no quesito semântico (apenas a classe da entidade) e que incluem as entidades "PESSOA", "LOCAL" e "ORGANIZAÇÃO".

4.1.2.1 Sistema CORTEX

O primeiro estado da arte no quesito semântico foi estabelecido na primeira edição do evento HAREM pelo sistema CORTEX[52]. Este sistema foi desenvolvido a partir regras gramaticais e bases de dados externas. Duas configurações de avaliação para este sistema se mostram relevantes:

1. **Cenário total**: todas as categorias, Tabela 4.1.
2. **Cenário seletivo**: categorias : "PESSOA", "ORGANIZACAO", "LOCAL", "TEMPO", "ACONTECIMENTO" e "VALOR", Tabela 4.2.

CORTEX - Cenário total		
Precisão(%)	Abrangência(%)	$F_{B=1}$
77,85	50,92	61,57

Tabela 4.1: Resultados do sistema CORTEX para o cenário total.

CORTEX - Cenário seletivo		
Precisão(%)	Abrangência(%)	$F_{B=1}$
77,86	60,97	68,39

Tabela 4.2: Resultados do sistema CORTEX para o cenário seletivo.

PUC-Rio - Certificação Digital Nº 1612825/CA

4.1.2.2 Comitê ETL ETL_{CMT}

A técnica *Entropy Guided Transformation Learning* (ETL)[40] é um aprimoramento sobre o *Transformation Based Learning* (TBL) [53]. O TBL busca a partir de um classificador base aprender as regras que adicionadas ao sistema melhoram a performance. Entretanto, essas regras precisam ser construídas por meio do uso de esforço humano e agrupadas de acordo com a tarefa em questão. ETL busca minimizar o esforço humano envolvido em agrupar as regras por meio da realização automática desta etapa via em árvores de decisão.

Os *datasets* utilizados para o experimento foram:

- Treino: 100% Primeiro HAREM
- Teste: 100% miniHarem

O ETL superou o estado-da-arte definido pelo CORTEX sobre o corpus miniHAREM. A configuração utilizada corresponde a um comitê de modelos ETL —ETL_{CMT}— e o sistema final realiza classificação com base em votação da maioria. Foram considerados dois cenários de avaliação:

1. Cenário total: todas as categorias, Tabela 4.3.
2. Cenário seletivo: categorias : "PESSOA", "ORGANIZACAO", "LOCAL", "TEMPO" e "VALOR". Tabela 4.4.

ETL_{CMT} - Cenário total		
Precisão(%)	Abrangência(%)	FB=1
77,52	53,86	63,56

Tabela 4.3: Resultados do sistema ETL_{CMT} para o cenário total.

ETL_{CMT} - Cenário seletivo		
Precisão(%)	Abrangência(%)	FB=1
77,27	65,20	70,72

Tabela 4.4: Resultados do sistema ETL_{CMT} para o cenário seletivo.

PUC-Rio - Certificação Digital Nº 1612825/CA

4.1.2.3 CharWNN

A rede *CharWNN* mencionada em Subseção 4.1.1 obteve um novo aprimoramento. Um *setup* diferente para os corpus de utilizados foi aplicado:

- Treino: 95% Primeiro HAREM
- Calibração: 5% Primeiro HAREM
- Teste: 100% miniHarem

Avaliações realizadas:

1. Cenário total: todas as categorias, Tabela 4.5.
2. Cenário seletivo: categorias : "PESSOA", "ORGANIZACAO", "LOCAL", "VALOR" e "DATA", Tabela 4.6.

CharWNN - Cenário total		
Precisão(%)	Abrangência(%)	FB=1
74,54	68,53	71.41

Tabela 4.5: Resultados do sistema CharWNN para o cenário total.

CharWNN - Cenário seletivo		
Precisão(%)	Abrangência(%)	FB=1
78.38	77.49	77.93

Tabela 4.6: Resultados do sistema CharWNN para o cenário seletivo.

PUC-Rio - Certificação Digital Nº 1612825/CA

4.1.2.4 Abordagens combinando Wikipédia e HAREM

Um trabalho similar este foi realizado justamente por meio da construção de um corpus em português para treino de sistemas voltados para NER utilizando o Wikipédia e DBpedia[25]. Sobre a mencionada abordagem são executados experimentos com diferentes configurações para os corpus a fim de averiguar os impactos de performance. Os experimentos são realizados por meio do treino e avaliação do sistema Stanford NER [54] . Os resultados são os representados em Tabela 4.7:

Entretanto, existe um diferença entre a metodologia destacada e a metodologia deste trabalho: neste trabalho, buscamos diminuir o ruído do *dataset* identificando as entidades mencionadas que não são contempladas no DBpedia.

	Teste sobre o segundo HAREM		
Dataset treino	Precisão	Abrangência	F1
Primeiro HAREM	71%	58%	64%
Wikipédia, 2%	84%	22%	35%
Wikipédia, 20%	71%	23%	36%

Tabela 4.7: Resultados para diversas configurações de corpus de treino.

4.2 Exploração de hiperparâmetros e arquiteturas

As arquiteturas de redes neurais dependem muito da escolha correta dos hiperparâmetros para alcançarem a melhor performance possível dentro de seu potêncial. Nesta seção serão exploradas arquiteturas de redes neurais e combinações de parâmetros para a tarefa. O espaço de busca para as melhores combinações de parâmetros é muito extenso e um ponto de partida adequado é averiguar na literatura as convergências e divergências sobre as escolhas de parâmetros.

No trabalho *Pratical Bayesian Optimization Of Machine Learning Algorithms* [55] uma descrição pertinente desta etapa de seleção de hiperparâmetros é formulada : " frenquentemente uma magia negra que requer experiência de experts". Articulando o processo de calibração de hiperparâmetros como um processo regado de incertezas e dependente de conhecimento *ad-hoc* dos desenvolvedores.

Um trabalho de destaque para NER sobre as arquiteturas atuais mais relevantes é o trabalho *Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks* no qual uma extensa combinação de parâmetros foi examinada, inclusive mensurando as distribuições de performance considerando *seeds* diferentes. Foram avaliadas questões referentes a sensibilidade do modelo no que diz respeito a diferentes *seeds* i.e. qual a variância da performance dos resultados dada a mudança da *seed* utilizada. Realizamos menções aos resultados obtidos no referido artigo como maneira de justificar as escolhas de hiperparâmetros realizadas.

4.2.1 Modelo Base

Os trabalhos de maior impacto envolvendo LSTMs e descritos em Subseção 4.1.1 são ilustrados de maneira genérica em Subseção 4.2.1 onde a figura representa os elementos principais das arquiteturas de maior notoriedade para a tarefa maneira abstrata. Os componentes ilustrados na figura são descritos a seguir, expondo quais os hiperparâmetros envolvidos e qual o grau de impacto

PUC-Rio - Certificação Digital Nº 1612825/CA

reportado na literatura para a performance do modelo.

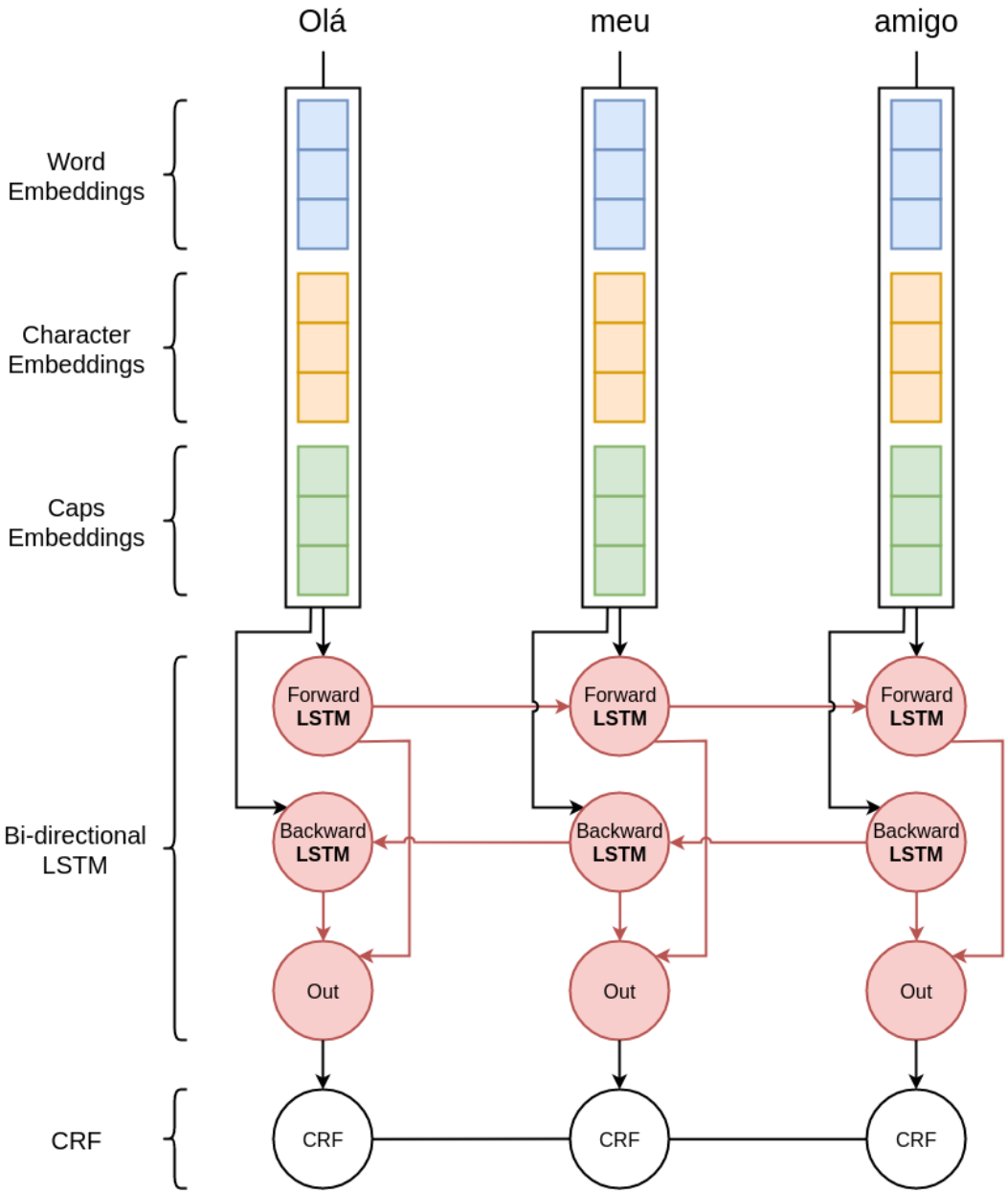


Figura 4.1: Representação genérica dos elementos comuns de dos modelos de melhor performance na literatura.

4.2.1.1 Formalização preliminar das entradas

Apresento nesta seção uma formalização inicial e elementar que será usada como base para as seções a seguir.

Dadas as sentenças S dos *datasets*:

$$S = \{s_0, s_1 ... s_{|S|}\} \tag{4-1}$$

PUC-Rio - Certificação Digital Nº 1612825/CA

Temos que cada sentença é composta por um conjunto de palavras.

$$\forall s \in S, \quad s = \{w_0, w_1 ... w_{|s|}\} \tag{4-2}$$

Cada palavra é composta por um conjunto de caracteres.

$$\forall w \in s, \quad w = \{c_0, c_1 ... c_{|w|}\} \tag{4-3}$$

4.2.1.2 *Word embeddings*

A técnica de *word embeddings* mapeia palavras para vetores de dimensões predefinidas. A construção deste recurso pode ser realizada sobre corpos de texto não anotados e utiliza a disposição das palavras a fim de elicitar uma representação vetorial das relações semânticas e sintáticas nos texto utilizado. Ademais, a redução de dimensionalidade é vantajosa para a aplicação de redes neurais, uma vez que: (1) evita que a rede tenha de comportar representações esparsas e (2) destila as relações de maior importância. Em suma, permite o mapeamento de relações semânticas em um espaço vetorial de dimensões arbitrárias.

Por meio da revisão da literatura sobre o uso de *word embeddings* em modelos para NER envolvendo redes neurais, destacam-se alguns elementos que se mostraram relevantes para a seleção deste recurso de maneira apropriada:

1. **Modelo de *embeddings***. Técnicas diferentes foram desenvolvidas para construir os modelos de *word embeddings*. Cada uma com suas propriedades e heurísticas particulares. Exemplos: GloVe[56] (semântica e sintaxe), Fast Text[57](morfologia) e Word2Vec[58](semântica).

2. **A dimensão do mapeamento**. A dimensão geralmente é selecionada por meio de tentativa e erro com valores entre 50 e 1000. Ademais, existem algumas tentativas de se estabelecer um *lower bound* para a sua seleção [59], mas sem grande destaque.

3. **Os gêneros de corpos de texto utilizados para o treino dos vetores**. Na literatura os *datasets* utilizados são compostos principalmente de textos de notícias anotados e as *embeddings* normalmente contemplam este gênero textual. Ademais, em [21] foi reportado uma crescimento na precisão da performance de 4.97% ao se utilizar *embeddings* treinadas sobre corpus que contemplam o gênero textual dos *datasets* do experimento.

PUC-Rio - Certificação Digital Nº 1612825/CA

O uso das *word embeddings* pela rede neural se dá como descrito em [41]. Basicamente, definimos uma *embedding layer* como representado em Subseção 4.2.1 que segue o procedimento:

1. É selecionado um conjunto de palavras para constituir o vocabulário reconhecido, $V = \{w_0, w_1 ... w_{|V|}\}$

2. Define-se uma tabela de busca T , que mapeia para cada palavra em V um vetor $v_w \in \mathbb{R}^{dim}$ onde dim é a dimensão de mapeamento utilizada.

3. Formalizo T como uma matriz onde cada linha é entendida como um vetor para cada palavra em V. Ou seja, $T \in \mathbb{R}^{|V| \times dim}$

4. Para se obter o vetor de $w_i \in V$ definimos, $v_{w_i} = T_{i,*}$

Neste trabalho utilizamos como base as *embeddings* geradas por [60], de 50, 100 e 300 dimensões. Esta escolha se deve a compatibilidade com os três elementos de impacto descritos anteriormente. Mais especificamente:

1. Utiliza o modelo de maior impacto na literatura para NER, GloVe.

2. Foram treinados para grande gama de dimensões, e.g. 50, 100, 300...

3. Foram construídos sobre corpos de texto oriundos de notícias e do Wikipédia e sobre sobre o português do Brasil e de Portugal tal como os *datasets* utilizados neste trabalho.

Nos casos excepcionais onde se lida com palavras fora do vocabulário D — *Out-of-Vocabulary words* (OOV) — , usamos a metodologia adotada em [51], um vetor inicializado com $[+\sqrt{\frac{3}{dim}}, +\sqrt{\frac{3}{dim}}]$ para $dim = 30$. Essa metodologia de inicialização é inspirada em [61].

As *word embeddings* podem ter seus parâmetros atualizados durante o treino ou não. No caso de se realizar o treinamento, o custo computacional aumenta consideravelmente, são introduzidos $dim * |D|$ parâmetros a mais na rede para treino.

Neste trabalho as *word embeddings* são palavras onde todas as letras são minúsculas. Durante o treino das *word embeddings* todas as letras foram convertidas para seu formato minúsculo. Isso ajuda a evitar que palavras capitalizadas por serem a primeira da frase sejam consideradas distintas de suas versões minúsculas.

PUC-Rio - Certificação Digital Nº 1612825/CA

4.2.1.3 *Character Representation*

Para realizar a extração de elementos morfológicos das palavras iremos utilizar a arquitetura proposta em [20]. Essa arquitetura faz uso de uma rede convolucional para a elicitação de elementos morfológicos das palavras. O funcionamento dos *character embeddings* ocorre de maneira similar aos *word embeddings*:

1. É selecionado um conjunto de palavras para constituir o alfabeto de caracteres reconhecidos, $A = \{c_0, c_1 ... w_{|A|}\}$

2. Define-se uma tabela de busca TA , que mapeia para cada caractere em A um vetor $v_c \in \mathbb{R}^{|A|}$.

3. Para inicialização dos vetores seguimos a abordagem utilizada por [48] em que a inicialização é realizada por meio de valores uniformemente em $[+\sqrt{\frac{3}{dim}}, +\sqrt{\frac{3}{dim}}]$ para $dim = 30$. Tal como realizado para palavras não presentes no dicionário, como descrito na seção anterior.

4. Formalizo TA como uma matriz onde cada linha é entendida como um vetor para cada caractere em A. Ou seja, $TA \in \mathbb{R}^{|A| \times |A|}$

5. Para se obter o vetor de $c_i \in A$ definimos, $v_{c_i} = TA_{i,*}$

O componente para a representação dos caracteres é a rede neural ilustrada em Figura 4.2. Os caracteres são mapeados em sequência para seus vetores, em seguida é aplicada uma camada convolucional e por fim é realizado *max pooling* para a extração das características morfológicas.

PUC-Rio - Certificação Digital Nº 1612825/CA

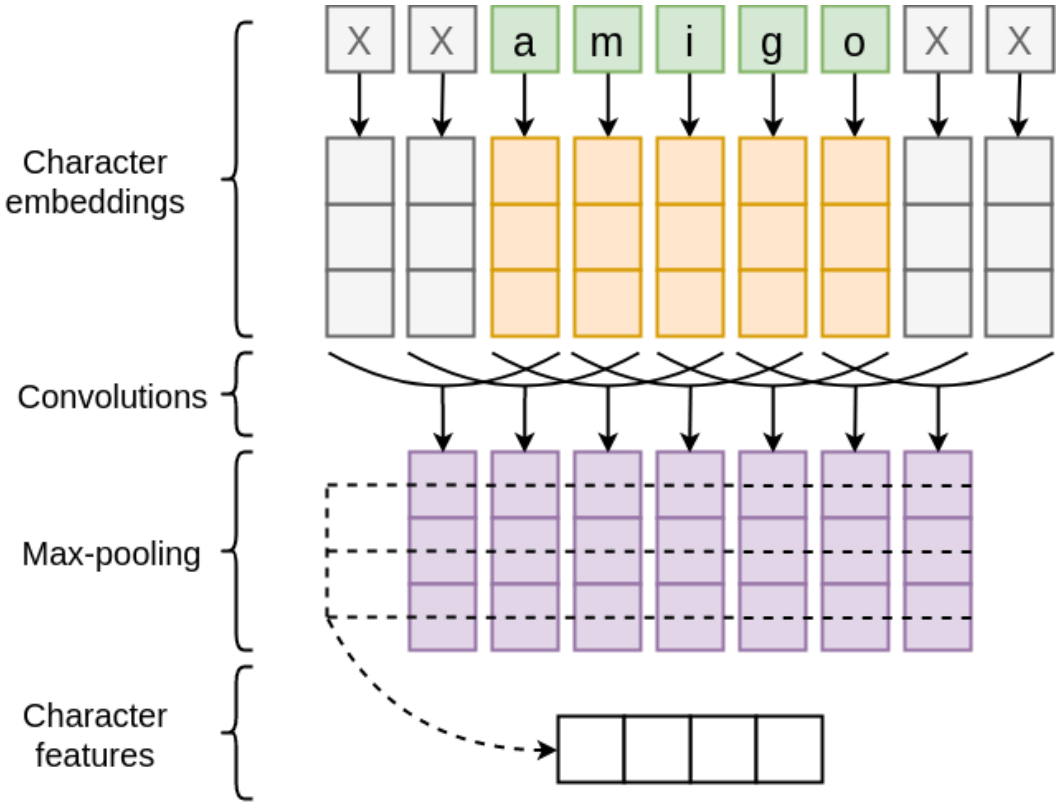


Figura 4.2: O componente da rede responsável pela extração de elementos da estrutura morfológica das palavras.

PUC-Rio - Certificação Digital Nº 1612825/CA

Os hiperparâmetros relevantes envolvidos nesta etapa são:

1. Tamanho da janela — *window size* — utilizada sobre os caracteres
2. A quantidade de *features* capturadas
3. O alfabeto utilizado
4. *Dropout* pode ser utilizado sobre os *character embeddings* e sobre *Character Features*.

Existe outra abordagem envolvendo o uso LSTMs bi-direcionais [50]. Entretanto, não há diferenças de performance significativas reportadas para NER entre as duas abordagens [21]. As diferenças de performance entre essas duas técnicas são significativas para algumas outras tarefas de NLP como *pos-tagging* e *chunking*.

Neste trabalho, nosso dicionário é composto por dígitos, acentos e as possíveis letras do alfabeto para o português, todas as letras são convertidas para seu formato minúsculo. E, no caso de letras desconhecidas, utilizamos a mesma técnica usada para *word embeddings* Subsubseção 4.2.1.2.

4.2.1.4 Local Features

É possível inserir informações adicionais sobre características relevantes das palavras a fim de conduzir o modelo a considerar essas características explicitamente. Exemplos dessas características são: capitalização das palavras (maiúscula, minúscula), sufixos, prefixos, presença de pontuação...

Neste trabalho exploramos os impactos na performance dos modelos ao se inserir as seguintes informações sobre as palavras, utilizadas em [21]:

- Começa com letra maiúscula
- Todas as letras são maiúsculas
- Todas as letras são minúsculas
- Se contêm digito
- Se a maioria dos caracteres são dígitos ($> 50\%$)
- Se todos os caracteres são dígitos

PUC-Rio - Certificação Digital Nº 1612825/CA

4.2.1.5 Bi-LSTM

A camada de Bi-LSTM recebe os vetores mapeados dos *word embeddings*, *char embeddings* e *local features*. Cada unidade de *input* da rede recebe as entradas referentes a uma palavra.

Os hiperparâmetros envolvidos na calibração desta camada são:

- **Unidades da LSTM**. Para qual a dimensão as entradas são mapeadas pela LSTM.
- **Droput Variacional**. Uma técnica para aplicar *dropout* nas *gated recurrent units* da LSTM [62].
- ***Dropout***. *Dropout* pode ser aplicado tanto no *input* como no *output* da LSTM

4.2.1.6
Treino e *gradient descent optimization*

Os otimizadores buscam lidar com os problemas associados aos possíveis problemas do uso de *mini-batch gradient descent*, como:

- Tempo para convergência do modelo
- Ajuste da taxa de aprendizado — *learning rate* — para evitar mínimos subótimos e pontos de "sela" — *hinge* —.
- Ajuste da *learning rate* para cada parâmetro da rede, enfatizando situações que ocorrem em menor proporção e vice-versa.

Para acelerar o processo de treino da LSTM, agrupamos as sentenças em *mini-batches* de acordo com seus tamanhos. Por exemplo, ao se utilizar *mini-batches* de tamanho 10, iremos agrupar 10 sentenças do mesmo tamanho. Isso se deve ao fato de que a implementação de LSTM utilizada recebe invariavelmente um conjunto de entradas de mesmo tamanho. Para contornar essa limitação, e adicionar sentenças de tamanhos diferentes no mesmo *mini-batch*, seria necessário fazer com que todas as sentenças adotassem o tamanho da maior do grupo, colocando um "preenchimento" nas sentenças menores para que adotassem esse tamanho, aumentando o uso de recursos computacionais.

PUC-Rio - Certificação Digital Nº 1612825/CA

4.3
Experimentos

LSTMs são caras em termos de custo de processamento. Este fato aliado ao tamanho de nosso *dataset* tornaria um processo automático de busca por hiperparâmetros muito demorado. Nossa abordagem consiste em partir de um modelo base para então realizar alterações nos parâmetros e avaliar a sua melhoria incrementalmente.

Durante o treinamento, realizaremos avaliações em intervalos regulares para avaliar a performance do modelo. Recuperaremos o estado do modelo em que sua $F_{\beta=1}$ reportada foi a maior para a realização da etapa de teste. Como o *dataset* construído é muito extenso, iremos realizar a avaliação do modelo em intervalos definidos arbitrariamente, em função da quantidade de exemplos utilizados para treino.

Os resultados de maior importância no processo de exploração de modelos e sua posição com os trabalhos de maior relevância na literatura são os apresentados em Tabela 4.8. Os resultados são detalhados, analisados e comparados coma mais detalhes em Subsubseção 4.3.1.5.

Comparação entre os sistemas			
	Precisão(%)	Abrangência(%)	$F_{B=1}$
Cortex	77,86	60,97	68,39
ETL_{CMT}	77,27	65,20	70,72
CharWNN	78.38	77.49	77.93
Wikipédia 20%	71.00	23.00	36.00
Modelo preliminar	67.90	53.90	60.10
1 Wiki 1 HAREM	70.09	65.89	67.92
1 Wiki 1 HAREM + Diff	71.38	70.44	70.91

Tabela 4.8: Compilação dos resultados de maior relevãncia e suas posições entre os trabalhos relacionados.

4.3.1 Modelo preliminar

O modelo inicial para a exploração de hiperparâmetros aplica a configuração definida em Tabela 4.9 sobre o modelo genérico descrito em Subseção 4.2.1. Realizamos experimentos para avaliar as mudanças de performance ao adotar diferentes hiperparâmetros.

- Neste modelo base, não serão utilizadas as *local features* descritas em Subsubseção 4.2.1.4.
- É utilizada a mesma *seed* para a geração de números aleatórios em todos os experimentos preliminares. Isso destaca as mudanças de performance entre modelos por não estarem sujeitas a aleatoriedades adicionais em virtude de diferentes *seeds*. Podemos, por exemplo, avaliar a performance de um modelo inicializado da mesma maneira para diferentes taxas de aprendizado.
- O corpus utilizado para treino é o composto por todas as sentenças extraídas do Wikipédia.
- A avaliação de performance do modelo é realizada a cada 1000 sentenças de treino

4.3.1.1 Estudo preliminar sobre a metodologia de calibração

Como destacado em Seção 2.7, existem algumas divergências quanto as estruturas do corpus do primeiro HAREM e o corpus produzido automaticamente a partir do Wikipédia. A diferença de maior impacto, como descrito anteriormente, se trata da classe "PESSOA", cujos tipos exceto "INDIVIDUAL" não são compatíveis com a metodologia de construção do *dataset*.

PUC-Rio - Certificação Digital Nº 1612825/CA

	LSTM
LSTM units	200
Stacked LSTMs	1
Variational Dropout	0.0
Dropout	0.5, no input da LSTM
	Character representation
Char window size	3
Char filters	30
Dropout	0.5, Aplicado sobre a saída da camada de *char embeddings*
	Word Embeddings
Word embeddings dim	100
Word embeddings model	GloVe
	Gradient optimizarion
Otimizador	SGD
Decay	Calculado por $nt = n_0/(1 + pt)$ t é o número do batch (quantidade de exemplos) completado. n_0 é a lr original p é calculado para ficar entre 0.016 e terminar com 0.003.
Momentum	0.9
Gradient clipping	5.0
Mini-batch size	10

Tabela 4.9: Configuração utilizada para o modelo preliminar.

Portanto, hipotetizamos: realizar a calibração avaliando somente os casos do tipo "INDIVIDUAL" para a classe "PESSOA" é a metodologia mais pertinente. Afinal, devemos avaliar a performance do modelo considerando as estruturas dos exemplos introduzidos no conjunto de treino, averiguando se o modelo consegue generalizá-los corretamente.

Realizamos um experimento inicial sobre a hipótese. Treinamos o modelo base Subseção 4.2.1 sobre um *dataset* de calibração executando a avaliação sobre todos os tipos de "PESSOA" para um e, para o outro, apenas pessoas do tipo "INDIVIDUAL".

A princípio os resultados confirmam a hipótese de que existe ganho de performance ao se considerar apenas pessoas do tipo individual, como demonstrado em Tabela 4.10. Entretanto, deve-se levar em consideração o fato de que: a quantidades de exemplos, ao se eliminar as entidades cujo tipo não é "INDIVIDUAL", diminui. Ou seja, a diferença de performance entre os casos pode ser atribuído a esse desbalanceamento na quantidade de exemplos.

Para os experimentos preliminares adotaremos a metodologia de ignorar os casos do tipo individual. Afinal, como esses casos não estão presentes no *dataset* de treino podemos atribuir as eventuais classificações corretas sobre esses como fruto de uma generalização incorreta por parte do modelo e o acerto seria coincidental.

PUC-Rio - Certificação Digital Nº 1612825/CA

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
Apenas "INDIVIDUAL"	**60.76%**	**66.20%**	56.15%
Sem filtro	60.65%	63.87%	**57.74%**

Tabela 4.10: Resultados do modelo preliminar ao se calibrar o modelo preliminar utilizando *datsets* com todos os tipos da classe "PESSOA" e com apenas o tipo "INDIVIDUAL" da classe.

4.3.1.2 Estudo preliminar sobre os hiperparâmetros do modelo base

Nesta seção são realizados experimentos sobre os hiperparâmetros do modelo base, para entender seus impactos sobre a performance e elaborar a configuração da rede buscando ganho de performance incrementalmente.

Realizamos experimentos sobre o impacto da quantidade de ***recurrent units*** na LSTM. Uma possível intuição sobre o impacto deste hiperparâmetro é: como a quantidade "regras" aprendidas pela rede é positivamente proporcional a sua largura e o *dataset* é bastante ruidoso, temos que aumentam as chances do modelo incorporar a estrutura do ruído com o aumento da largura. Em Tabela 4.11 podemos perceber que, preliminarmente, nenhuma modificação dobre o o valor padrão adotado pelo modelo preliminar (200) obteve ganho de performance.

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
100 *rec units*	60.45% (-0.31%)	65.56% (-0.64%)	56.07% (-0.07%)
150 *rec units*	60.45% (-0.31%)	62.04% (-4.16%)	58.93% **(+2.79%)**
300 *rec units*	60.74% (-0.02%)	65.70% (-0.50%)	56.48% (+0.33%)

Tabela 4.11: Impacto sobre a performance do modelo preliminar ao se utilizar diferentes quantidades de *recurrent units*.

Uma possível abordagem para estimular a detecção de dependências de grandes distâncias entre as palavras da sentença é por meio do uso de **LSTMs em sequência**. Isso se deve ao fato de que os *outputs* das LSTMs inferiores serão os *inputs* das LSTMs superiores, promovendo a busca de dependências de longas distâncias em sequencia.

Em Tabela 4.12 podemos repara que ao se utilizar 3 camadas de LSTM em sequência temos um ganho de performance considerável em comparação com as alternativas, como demonstrado em Tabela 4.12

No trabalho [49] foram utilizadas redes neurais com **conexões residuais**, nas quais o *input* da rede é acessível por todas as LSTMs em sequência

PUC-Rio - Certificação Digital Nº 1612825/CA

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
2 LSTMs	60.48% (-0.28%)	69.19% **(+2.99%)**	53.73% (-2.42%)
3 LSTMs	61.38% **(+0.62%)**	62.52% (-3.68%)	60.28% **(+4.13%)**
4 LSTMs	59.09% (-1.67%)	67.32% (+1.12%)	52.66% (-3.49%)

Tabela 4.12: Impacto sobre a performance do modelo preliminar ao se utilizar diferentes quantidades de *recurrent units*.

da rede, a operação utilizada para a criação desta conexão residual é concatenação. Em Tabela 4.13 podemos averiguar que conexões residuais melhoram a performance do *baseline*, mas não mais do que 3 LSTMs em sequência como demonstrado em Tabela 4.12.

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
2 LSTMs	61.19% **(+0.43%)**	64.66% (-1.54%)	58.08% (+1.93%)
3 LSTMs	60.78% (+0.02%)	63.12% (-3.08%)	58.60% **(+2.45%)**

Tabela 4.13: Impacto sobre a performance do modelo preliminar ao se utilizar conexões residuais sobre camadas LSTM em sequência.

PUC-Rio - Certificação Digital Nº 1612825/CA

Realizamos experimentos sobre diferentes valores para o ***dropout* variacional** aplicado sobre as unidades recorrentes da LSTM. A princípio, o *dropout* de melhor performance é o utilizado pelo modelo base, 25, como demonstrado em Tabela 4.14

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
25	61.98% **(+1.22%)**	70.39% **(+4.19%)**	55.36% (-0.78%)
30	61.91% (+1.14%)	65.25% (-0.95%)	58.89% (+2.74%)
35	61.77% (+1.01%)	62.22% (-3.98%)	61.33% **(+5.18%)**

Tabela 4.14: Impacto sobre a performance do modelo preliminar ao se utilizar diferentes taxas de *variational dropout*.

Avaliamos o impacto de diferentes valores para taxa de ***dropout*** aplicado no input da LSTM e nos *embeddings* da camada convolucional do modelo base. Entretanto, os valores para esse hiperparâmetro devem ser calibrados a fim de regularizar as redes de acordo com sua quantidade de parâmetros, e neste experimento especifico, a rede base não adota grandes dimensões.

Avaliamos algumas técnicas para realizar a **otimização dos gradientes**. Como descrito em Tabela 4.16. Testamos a performance ao se fixar uma taxa

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
Dropout 60	60.21% (-0.55%)	63.78% (-2.42%)	57.02% **(+0.87%)**
Dropout 70	58.47% (-2.29%)	61.47% (-4.73%)	55.75% (-0.40%)

Tabela 4.15: Impacto sobre a performance do modelo preliminar ao se utilizar diferentes taxas de *Dropout*.

de aprendizado no SGD e utilizamos o otimizador NADAM, como sugerido no paper [21]. Os resultado reportados não apresentam ganho de qualidade.

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
mini-batch 32	60.08% (-0.68%)	65.58% (-0.62%)	55.43% (-0.71%)
Lr 0.005	59.90% (-0.87%)	58.78% (-7.42%)	61.05% **(+4.91%)**
NADAM	59.66% (-1.10%)	63.96% (-2.24%)	55.90% (-0.25%)

Tabela 4.16: Impacto sobre a performance do modelo preliminar ao se utilizar diferentes configurações para otimização sobre o *stochastic gradient descent*.

A variação sobre a performance do modelo com diferentes dimensões de mapeamento para ***word embeddings*** pode ser observada em. Na literatura examinada para este trabalho, os melhores valores para hiperparâmetro estão entre 50 e 100. Em Tabela 4.17 temos que a dimensão de mapeamento com a melhor performance é 100.

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
Dimensão 50	60.58% (-0.18%)	63.74% (-2.46%)	57.72% **(+1.57%)**
Dimensão 300	60.35% (-0.41%)	63.72% (-2.48%)	57.33% (+1.18%)

Tabela 4.17: Impacto sobre a performance do modelo preliminar ao se utilizar *word embeddings* de diferentes dimensões.

Averiguamos quais os impactos da metodologia do **formato de anotação** utilizado par aos testes. Começamos os experimentos com atécnica IOB, entretanto, existem evidências na literatura de que o esquema IOBES promove melhores performances [63] e, em outros trabalhos não é reportado nenhum ganho de performance aparente[21]. Realizamos um experimento convertendo os *datasets* para o formato IOBES por ser mais expressivo sobre as características das entidades. Como pode ser observado em Tabela 4.18, há um ganho de performance ao se utilizar essa técnica.

Como o dicionário e alfabeto utilizados não possuem letras maiúsculas, mas essas são informações relevantes para o idioma português, especialmente

PUC-Rio - Certificação Digital Nº 1612825/CA

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
IOBES	62.08% (+1.32%)	65.05% (-1.15%)	59.36% (+3.22%)

Tabela 4.18: Impacto sobre a performance do modelo preliminar ao se utilizar o esquema de anotação IOBES.

para a tarefa em questão já que frequentemente entidades nomeadas apresentam a primeira letra como maiúscula. Realizamos um experimento onde foram inseridas as ***local features*** descritas em Subsubseção 4.2.1.4 que incluem informações das palavras referentes a capitalização das letras das palavras e a presença de numerais. Os resultados apresentam grande ganho de performance como apresentado em Tabela 4.19.

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
Local features	66.90% (+6.13%)	74.34% (+8.13%)	60.81% (+4.66%)

Tabela 4.19: Performance sobre o modelo ao se realizar o treino apenas com o Segundo HAREM.

Outra arquitetura, proposta em [50], consiste em aplicar uma camada densamente conectada entre as camadas LSTM e CRF. Realizados os experimentos apresentados em Tabela 4.20 e confirmamos um aumento na qualidade da performance para uma configuração explorada para a rede.

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
Ativação *tan*	60.96% **(+0.20%)**	59.15% (-7.05%)	62.88% **(+6.73%)**
Ativação *relu*	60.70% (-0.07%)	66.37% (**+0.17**%)	55.91% (-0.24%)

Tabela 4.20: Performance do modelo preliminar ao se adicionar uma camada densamente conectada após a primeira LSTM.

4.3.1.3 Ajustes sobre os experimentos preliminares

A partir dos resultados preliminares, estimamos novas configurações para atingir um ganho de performance. Para esse fim, construímos três configurações. Ademais todas as configurações possuem os seguintes pontos em comum:

– Uso das *local features* descritas em Subsubseção 4.2.1.4

PUC-Rio - Certificação Digital Nº 1612825/CA

– Utilizar o esquema de anotação IOBES

– Treino sobre o *dataset* balanceado

Primeira configuração. Utiliza hiperparâmetros que apresentaram maior ganho de $F_{\beta=1}$. As modificações sobre o modelo preliminar foram:

– *Dropout variacional* de 0.25

– Três LSTMs em sequência

Para a segunda e terceira configuração traçamos como objetivo a otimização do *recall*. Isso se deve ao fato de que na maior parte dos experimentos em que ocorrem melhorias sobre a medida $F_{\beta=1}$ devem esses ganhos de qualidade par ao aumento da precisão. Ou seja, é pertinente buscar modelos que visem a melhoria da abrangência do modelo.

Segunda configuração. Sobre esta configuração consideramos apenas os hiperparâmetros que apresentaram ganho sobre $F_{\beta=1}$ entretanto, sobre este conjunto, priorizaremos os hiperparâmetros que mais obtiveram ganho em *recall*:

– *Dropout* variacional: 0.35

– Duas camadas densamente conectadas após a LSTM. Ativação *tan*.

Terceira configuração. Utilizamos os parâmetros onde houve os maiores ganho de *abrangência* independente das outras métricas:

– *Dropout*: 0.6

– *Dropout* variacional: 0.35

– Quantidade de unidades recorrentes: 150

– Duas camadas densamente conectadas após a LSTM. Ativação *tan*.

– Taxa de aprendizado: 0.005

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
1ª Configuração	65.97% (+5.21%)	68.57% (+2.37%)	63.56% (+7.41%)
2ª Configuração	67.28% (+6.52%)	69.99% (+3.79%)	64.77% **(+8.63%)**
3ª Configuração	67.49% **(+6.73%)**	77.53% **(+11.33%)**	59.76% (+3.61%)

Tabela 4.21: Performance das novas configurações desenvolvidas após a interpretação dos resultados.

PUC-Rio - Certificação Digital Nº 1612825/CA

4.3.1.4 Adicionando semântica ao modelo

Como descrito na Seção 2.7, o *dataset* construído a partir do Wikipédia não apresenta informações semânticas anotadas por humanos. Para que essas informações sejam introduzidas no modelo é necessário adicionar no conjunto de treino exemplos as expressem.

Portanto, iremos realizar testes sobre modelos que são treinados sobre uma combinação de *datasets* do Wikipédia e do Segundo HAREM. Para tal, introduzimos o *dataset* do Segundo HAREM ao conjunto de treino. Este corpus foi construído utilizando as mesmas diretrizes de anotação do Primeiro HAREM e miniHAREM.

Ademais, para os experimentos a seguir, utilizaremos o conjunto de calibração utilizando todos os tipos da classe "PESSOA" e não apenas os do tipo "INDIVIDUAL".

Como maneira de reduzir o custo computacional de realização dos experimentos, utilizamos um modelo que apresenta uma distribuição de resultados com menor variância. Mais especificamente, utilizaremos o otimizador NADAM, por convergir mais rapidamente [21] do que SGD. Em suma, por estarmos sujeitos a uma menor variância nos resultados, podemos realizar menos experimentos para reportar a performance do modelo aproximada para a sua melhor qualidade. Utilizaremos também duas camadas de LSTM para reduzir po tempo de treino.

Nos experimentos a seguir apresentamos a performance do modelo ao se utilizar diferentes combinações de proporções entre sentenças oriundas do Segundo HAREM e do Wikipédia.

Primeiro experimento: Utilizar apenas o corpus do Segundo HAREM para treino. Ao se utilizar apenas o corpus do Segundo HAREM para treino, podemos avaliar se há melhoria ao se adicionar sentenças do Wikipédia no conjunto de avaliação. Basicamente, estabelecemos um *baseline*. Os resultados dessa execução são expressos na Tabela 4.22.

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
Segundo HAREM	72.72%	75.28%	70.32%

Tabela 4.22: Performance sobre o modelo ao se realizar o treino apenas com o Segundo HAREM.

Em seguida realizamos dois experimentos com proporções diferentes:

1. Uma sentença do Wikipédia para cada sentença do HAREM.

PUC-Rio - Certificação Digital Nº 1612825/CA

2. Duas sentenças do Wikipédia para cada sentença do HAREM.

3. Uma sentença do Wikipédia para cada duas sentença do HAREM.

Os resultados para as diferentes proporções é expresso na Tabela 4.23.

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
1 Wiki, 1 HAREM	74.22% (+1.50%)	77.58% (+2.30%)	71.14% (+0.81%)
2 Wiki, 1 HAREM	73.72% (+1.00%)	78.16% (+2.87%)	69.76% (-0.56%)
1 Wiki, 2 HAREM	73.65% (+0.93%)	76.13% (+0.84%)	71.33% (+1.01%)

Tabela 4.23: Impacto na performance para diferentes proporções de sentenças do Segundo HAREM e Wikipédia.

Observa-se uma melhoria na performance ao se utilizar uma combinação de sentenças do Wikipédia e Segundo HAREM.

Segundo experimento: Adicionamos ao modelo informações sobre a fonte da sentença. Basicamente, expressamos na rede se a sentença é oriunda do Wikipédia ou do Segundo HAREM. A intuição por trás deste experimento consiste em estimular a rede a incorporar as divergências e similaridades na estrutura de ambos os corpus, melhorando a precisão do modelo. Em suma, introduzimos uma "*flag*" na sentença acusando sua origem. O resultado do experimento realizado é apresentado na Tabela 4.24.

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
Diff Wiki1 HAR 1	74.93% (+2.22%)	78.72% (+3.43%)	71.50% (+1.18%)

Tabela 4.24: Performance do modelo ao se introduzir a informação referente a fonte da sentença.

Ao se introduzir a informação da origem da sentença no modelo pode-se observar uma melhora mais acentuada na precisão do modelo. O modelo incorpora as diferenças entre sentenças oriundas do Segundo HAREM, que possuem informações semânticas, e das sentenças do Wikipédia, resultando em melhorias na precisão e e abrangência.

Sobre este modelo realizamos experimentos adicionais visando promover melhorias na qualidade dos resultados. Os experimentos em questão, cujos resultados são expressos em Tabela 4.25, são:

1. "Embaralhamento" (*shuffle*) do *dataset*

2. Aplicação de resíduo na segunda camada da rede

PUC-Rio - Certificação Digital Nº 1612825/CA

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
Shuffle	75.64% (+2.92%)	77.34% (+2.06%)	74.02% (+3.69%)
Residual	75.43% (+2.71%)	79.04% (+3.75%)	72.13% (+1.81%)

Tabela 4.25: Performance para modificações no corpus.

Observa-se que as abordagens exploradas promoveram um ganho de performance sobre o modelo.

Terceiro experimento: Por fim, desenvolvemos uma arquitetura a partir dos resultados do experimento anterior. Basicamente incorporando resíduo e "embaralhando" o *dataset*. A performance deste experimento está descrita em Tabela 4.26.

Experimento	Calibração $F_{\beta=1}$	Calibração P	Calibração R
Experimento Compilado	75.47% (+2.75%)	78.07% (+2.79%)	73.04% (+2.72%)

Tabela 4.26: Resultados para o terceiro experimento.

PUC-Rio - Certificação Digital Nº 1612825/CA

4.3.1.5 Análise dos resultados

Nestas secção analisamos os resultados obtidos.

A seguir, listamos os experimentos de maior relevância para este trabalho, analisando a qualidade dos resultados e quais suas características marcantes. A compilação destes resultados em comparação com os resultados publicados de maior relevância para este trabalho estão em Tabela 4.28 e Tabela 4.29.

Os três modelos de maior relevância abordados neste trabalho são:

(A) Preliminar - Por meio da calibração do modelo preliminar que utiliza apenas o *dataset* do Wikipédia para treino, podemos averiguar:

- Como a performance do modelo construído é impactado por diferentes configurações de hiperparâmetros, estimando configurações relevantes embasadas em trabalhos anteriores.
- Qual a performance obtida por meio do treino em um *dataset* construído automaticamente, *Silver standard*, e calibração e teste sobre *datasets Golden Standard*.

Neste modelo constatamos um ganho de performance considerável sobre o trabalho [25]. Atribuímos este ganho de performance aos seguintes fatores:

Comparação entre os sistemas relevantes deste trabalho			
	Precisão(%)	Abrangência(%)	$F_{B=1}$
(A) Modelo preliminar	67.90	53.90	60.10
(B) 1 Wiki 1 HAREM	70.09	65.89	67.92
(C) 1 Wiki 1 HAREM + Diff	71.38	70.44	70.91

Tabela 4.27: Compilação dos resultados de maior relevância neste trabalho.

– Os modelos utilizados neste trabalho são os modelos mais performáticos atualmente, constituindo abordagens mais modernas.

– Neste trabalho extraímos uma quantidade de sentenças superior, introduzindo uma maior gama de casos de treino.

– Utilizamos um parser auxiliar para a extração de entidades não destacadas pelo DBpedia.

(B) Combinando Segundo HAREM e Wikipédia - Por meio da introdução de sentenças do Segundo HAREM, cujas diretivas de anotação são compatíveis com a estrutura do *dataset* de calibração e teste utilizado, podemos averiguar qual o impacto de se adotar diferentes proporções para a quantidade de sentenças das duas origens no conjunto de treino.

Nestes experimentos observamos uma melhoria consistente ao se combinar sentenças de ambas as origens. O resultado de melhor performance foi o que combinamos sentenças do Wikipédia e Segundo HAREM em proporções iguais.

(C) Explicitando a origem das sentenças - Ao se incluir no modelo a informação referente a qual a origem da sentença, observa-se uma melhoria na performance. Analisamos que isto se deve ao fato de que o modelo compreende quais as diferenças e similaridades de ambas as origens de sentença.

Por fim:

– Em Tabela 4.27 comparamos as abordagens mais relevantes neste trabalho, os modelos A, B e C destacados.

– Em Tabela 4.28 comparamos os resultados do modelo de melhor performance, o modelo C, com os resultados das publicações de maior relevância.

– Em Tabela 4.29 comparamos os resultado do modelo de melhor performance, modelo C, com os resultados das publicações de maior relevância a nível de classe de entidade.

Destacamos também que existe uma grande diferença entre a performance no conjunto de calibração e teste em nossos modelos. Tome por exemplo

PUC-Rio - Certificação Digital Nº 1612825/CA

Comparação entre os sistemas			
	Precisão(%)	Abrangência(%)	$F_{B=1}$
Cortex	77,86	60,97	68,39
ETL_{CMT}	77,27	65,20	70,72
CharWNN	78.38	77.49	77.93
Wikipédia 20%	71.00	23.00	36.00
(C) 1 Wiki 1 HAREM + Diff	71.38	70.44	70.91

Tabela 4.28: Posicionamento do modelo de melhor qualidade neste trabalho com as performances de melhor performance publicadas.

Performance por classe de entidade									
	PER			LOC			ORG		
	P	*R*	$F_{B=1}$	P	R	$F_{B=1}$	P	R	$F_{B=1}$
ETL_{CMT}	81.49	61.14	69.87	76.18	68.16	71.95	65.34	50.29	56.84
CharWNN	81.35	77.07	79.15	76.91	78.55	77.72	70.65	71.56	71.10
Modelo **(B)**	74.27	73.21	73.73	73.68	74.67	74.17	63.32	60.41	61.83

Tabela 4.29: Comparação do resultado de melhor performance a nível de entidade com os trabalhos relacionados na literatura.

PUC-Rio - Certificação Digital Nº 1612825/CA

o modelo **(C)**, cuja performance de treino e teste se encontra em Tabela 4.30. Pode-se averiguar uma grande diferença de $4,83$ pontos percentuais entre o conjunto de calibração e teste. Atribuímos essa discrepância o fato de O conjunto de teste utilizado neste trabalho é muito superior em dimensão aos demais sistemas relevantes publicados na literatura. Neste trabalho o conjunto de teste possui uma quantidade de artigos anotados 80% superior.

Para posicionar os resultados na literatura de maneira consistente é necessário destacar as diferenças nos experimentos realizados neste trabalho com os publicados na literatura, i.e. buscar divergências nas metodologias publicadas a fim de elicitar possíveis causas de discrepâncias entre os resultados. Destacamos as diferenças entre o experimento de melhor performance neste trabalho e o estado da arte, o sistema CharWNN:

1. **Distribuição dos conjuntos de teste e calibração**:
   - Como descito em Seção 2.7 e Subseção 4.1.1, as proporções entre os *datasets* utilizados para calibração e testes são diferentes. O conjunto de teste para este trabalho é 80% maior, e o de calibração 4 vezes maior. Ou seja, neste trabalho a etapa de avaliação aborda uma variedade exemplos maior.
   - Não é informado como o conjunto de calibração foi selecionado, quais as sentenças que o compõem. Ou se esse seria o conjunto utilizado para aplicar a técnica de *k-fold*.

	Calibração			Teste		
	$F_{\beta=1}$	P	R	$F_{\beta=1}$	P	R
Experimento Compilado	75.47%	78.07%	73.04%	70.91%	71.38%	70.44%

Tabela 4.30: Comparação entre a performance de avaliação e teste.

  - Não é informado se as sentenças utilizadas para o conjunto de calibração são obtidas por artigo, i.e. se são selecionados 5% dos artigos ou das sentenças, podendo as sentenças de um artigo estar presentes tanto no conjunto de calibração quanto de teste.

2. **Modelo de *word embeddings***

  - O modelo utilizado para o Char WNN corresponde ao modelo utilizado na publicação [42]. Diferente do utilizado neste trabalho.

3. **Pré-processamento**

  - Não são informados quais os passos utilizados para a conversão do HAREM para IOB2.

PUC-Rio - Certificação Digital Nº 1612825/CA

5
Conclusão

Neste trabalho construímos sem supervisão humana um *dataset* voltado para o reconhecimento de entidades mencionadas e exploramos as arquiteturas mais atuais para a tarefa realizando o treino sobre os dados coletados.

Para o *dataset*, descrevemos a metodologia desenvolvida em detalhes, promovendo a replicabilidade de nossos resultados. O *corpus* produzido apresentou uma grande quantidade de exemplos anotados e teve seu ruído mitigado por meio de diversas abordagens de filtragem. O enfoque de nossa abordagem é o português, de acordo com nosso conhecimento sobre a literatura, as técnicas publicadas não descrevem abordagens específicas para o idioma.

Experimentamos sobre as arquiteturas de redes neurais com as melhores performances reportadas atualmente para a tarefa. Realizamos o treino com diferentes configurações de corpus de treino, mais especificamente: (1) apenas sentenças oriundas do Wikipédia e (2) combinando com sentenças anotadas por humanos. Exploramos diversas configurações de arquiteturas visando o melhor aproveitamento do *dataset* construído. Reportamos os resultados preliminares sobre diversos modelos de redes neurais, apontado um caminho para a calibração dos hiperparâmetros.

Posicionamos os resultado obtidos neste trabalho entre as publicações com as melhores performances reportadas. Destacamos quais as diferenças entre as técnicas aplicadas e seus possíveis impactos para a qualidade dos resultados.

Elaboramos um modelo que busca utilizar sentenças anotadas com diferentes diretivas e incorporar as características dessas e, de acordo com os experimentos realizados neste trabalho, a qualidade dos resultados preliminares é promissora.

PUC-Rio - Certificação Digital Nº 1612825/CA

6
Trabalhos futuros

Este trabalho pode ser continuado de duas maneiras principais: (1) melhorias no *dataset* e (2) melhorias nos modelos.

6.1
Melhorias no *dataset*

– Neste trabalho utilizamos um sistema de NER auxiliar para identificar as entidades não presentes no DBpedia com a finalidade de reduzir o ruído do *dataset*. Uma abordagem para melhorar essa etapa é o uso de um sistema de NER treinado sobre este *dataset*, possibilitando uma abordagem com várias iterações. Em outras palavras podemos usar o modelo produzido neste trabalho para identificar as entidades não anotadas para então realizar novamente o treino e assim ciclicamente.

– Podemos buscar utilizar o parser padrão do Wikipédia ao invés de uma ferramenta alternativa para manusear o wikitexto dos artigos. Afinal, neste trabalho utilizamos a ferramenta alternativa MWparser e não a ferramenta original do Wikipédia.

– Na literatura existem variações de técnicas para detectar quais as sentenças do *dataset* produzido que possuem a maior qualidade [4]. Correspondendo uma possível abordagem para mitigar ainda mais o ruído do *dataset*.

6.2
Melhorias nos modelos

– O *dataset* produzido é muito ruidoso, existem maneiras de modelar este ruído e introduzir essa informação em modelos de redes neurais [64]. A modelagem do ruído para NER pode ser entendido como a probabilidade de inversão entre as classes abordadas pelo modelo, e.g. qual a probabilidade entre a classe "B-PER" se encontrar invertida com "B-LOC". Uma abordagem realizada especificamente para o Wikipédia pode ser observada em [65].

PUC-Rio - Certificação Digital Nº 1612825/CA

– Existem iniciativas para substituir as camadas LSTM por CNNs, visando tanto ganho de qualidade quanto diminuir o custo computacional do treino [66].

– Em um trabalho recente, foram construídas *embeddings* para gerar a representação vetorial genérica sobre entidades de diferentes classes. Em outras palavras, criam-se *embeddings* especificamente para cada classe de entidade. Essa abordagem obteve grande sucesso e utilizando o Wikipédia especificamente para sua construção [67]. O *dataset* produzido neste trabalho pode ser utilizado para a criação destes *embeddings* em português.

– Foi aplicado com sucesso a inclusão de *embeddings* sobre os afixos das palavras [68].

PUC-Rio - Certificação Digital Nº 1612825/CA

7
Referências bibliográficas

[1] CAMPOS, D.; MATOS, S. ; OLIVEIRA, J. L.. **Biomedical named entity recognition: a survey of machine-learning tools**. In: THEORY AND APPLICATIONS FOR ADVANCED TEXT MINING. InTech, 2012.

[2] SOBHANA, N.; MITRA, P. ; GHOSH, S.. **Conditional random field based named entity recognition in geological text**. International Journal of Computer Applications, 1(3):143–147, 2010.

[3] SCHUMAKER, R. P.; CHEN, H.. **Textual analysis of stock market prediction using breaking financial news: The azfin text system**. ACM Trans. Inf. Syst., 27(2):12:1–12:19, Mar. 2009.

[4] NOTHMAN, J.; RINGLAND, N.; RADFORD, W.; MURPHY, T. ; CURRAN, J. R.. **Learning multilingual named entity recognition from wikipedia**. Artificial Intelligence, 194:151–175, 2013.

[5] NOTHMAN, J.; CURRAN, J. R. ; MURPHY, T.. **Transforming wikipedia into named entity training data**. In: PROCEEDINGS OF THE AUSTRALASIAN LANGUAGE TECHNOLOGY ASSOCIATION WORKSHOP 2008, p. 124–132, 2008.

[6] GHADDAR, A.; LANGLAIS, P.. **Winer: A wikipedia annotated corpus for named entity recognition**. In: PROCEEDINGS OF THE EIGHTH INTERNATIONAL JOINT CONFERENCE ON NATURAL LANGUAGE PROCESSING (VOLUME 1: LONG PAPERS), volumen 1, p. 413–422, 2017.

[7] VRANDEČIĆ, D.. **Wikidata: A new platform for collaborative data collection**. In: PROCEEDINGS OF THE 21ST INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, WWW '12 Companion, p. 1063–1064, New York, NY, USA, 2012. ACM.

[8] **Google knowledge graph**. `https://www.google.com/intl/es419/insidesearch/features/search/knowledge.html`. Acessado: 21/7/2018.

PUC-Rio - Certificação Digital Nº 1612825/CA

[9] AUER, S.; BIZER, C.; KOBILAROV, G.; LEHMANN, J.; CYGANIAK, R. ; IVES, Z.. **Dbpedia: A nucleus for a web of open data**. In: Aberer, K.; Choi, K.-S.; Noy, N.; Allemang, D.; Lee, K.-I.; Nixon, L.; Golbeck, J.; Mika, P.; Maynard, D.; Mizoguchi, R.; Schreiber, G. ; Cudré-Mauroux, P., editors, THE SEMANTIC WEB, p. 722–735, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[10] GRISHMAN, R.; SUNDHEIM, B.. **Message understanding conference-6: A brief history**. In: COLING 1996 VOLUME 1: THE 16TH INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS, volumen 1, 1996.

[11] CHINCHOR, N. A.. **Overview of muc-7/met-2**. Technical report, SCIENCE APPLICATIONS INTERNATIONAL CORP SAN DIEGO CA, 1998.

[12] TJONG KIM SANG, E. F.. **Introduction to the conll-2002 shared task: Language-independent named entity recognition**. In: PROCEEDINGS OF THE 6TH CONFERENCE ON NATURAL LANGUAGE LEARNING - VOLUME 20, COLING-02, p. 1–4, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[13] TJONG KIM SANG, E. F.; DE MEULDER, F.. **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, p. 142–147. Association for Computational Linguistics, 2003.

[14] DODDINGTON, G. R.; MITCHELL, A.; PRZYBOCKI, M. A.; RAMSHAW, L. A.; STRASSEL, S. ; WEISCHEDEL, R. M.. **The automatic content extraction (ace) program-tasks, data, and evaluation.** In: LREC, volumen 2, p. 1, 2004.

[15] SANTOS, D.. **Caminhos percorridos no mapa da portuguesificação: A linguateca em perspectiva**. quot; Linguamática 11 (2009), 2009.

[16] SANTOS, D.; SECO, N.; CARDOSO, N. ; VILELA, R.. **Harem: An advanced ner evaluation contest for portuguese**. In: QUOT; IN NICOLETTA CALZOLARI; KHALID CHOUKRI; ALDO GANGEMI; BENTE MAEGAARD; JOSEPH MARIANI; JAN ODJIK; DANIEL TAPIAS (ED) PROCEEDINGS OF THE 5 TH INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC'2006)(GENOA ITALY 22-28 MAY 2006), 2006.

PUC-Rio - Certificação Digital Nº 1612825/CA

[17] FREITAS, C.; MOTA, C.; SANTOS, D.; OLIVEIRA, H. G. ; CARVALHO, P.. **Second harem: Advancing the state of the art of named entity recognition in portuguese.** In: LREC. Citeseer, 2010.

[18] SANTOS, D.. **Avaliação conjunta: um novo paradigma no processamento computacional da língua portuguesa**. No prelo, 2007.

[19] CHIU, J. P.; NICHOLS, E.. **Named entity recognition with bidirectional lstm-cnns**. arXiv preprint arXiv:1511.08308, 2015.

[20] SANTOS, C. N. D.; GUIMARAES, V.. **Boosting named entity recognition with neural character embeddings**. arXiv preprint arXiv:1505.05008, 2015.

[21] REIMERS, N.; GUREVYCH, I.. **Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging**. arXiv preprint arXiv:1707.09861, 2017.

[22] NIVRE, J.; DE MARNEFFE, M.-C.; GINTER, F.; GOLDBERG, Y.; HAJIC, J.; MANNING, C. D.; MCDONALD, R. T.; PETROV, S.; PYYSALO, S.; SILVEIRA, N. ; OTHERS. **Universal dependencies v1: A multilingual treebank collection.** In: LREC, 2016.

[23] ARTSTEIN, R.; POESIO, M.. **Inter-coder agreement for computational linguistics**. Computational Linguistics, 34(4):555–596, 2008.

[24] **Wikipedia**. `https://pt.wikipedia.org/`. Acessado: 11/8/2018.

[25] WEBER, C.; VIEIRA, R.. **Building a corpus for named entity recognition using portuguese wikipedia and dbpedia**. In: I WORKSHOP ON TOOLS AND RESOURCES FOR AUTOMATICALLY PROCESSING PORTUGUESE AND SPANISH, p. 9–15, 2014.

[26] HAHM, Y.; PARK, J.; LIM, K.; KIM, Y.; HWANG, D. ; CHOI, K.-S.. **Named entity corpus construction using wikipedia and dbpedia ontology.** In: LREC, p. 2565–2569, 2014.

[27] NASAW, D.. **Meet the 'bots' that edit wikipedia**. `https://www.bbc.co.uk/news/magazine-18892510`, Julho 2012. Acessado: 21/7/2018.

[28] **Infocaixa**. `https://pt.wikipedia.org/wiki/Ajuda:Infocaixa`. Acessado: 11/8/2018.

[29] **interlink**. `https://pt.wikipedia.org/wiki/Ajuda:Tutorial/Liga%C3%A7%C3%B5es_internas`. Acessado: 11/8/2018.

PUC-Rio - Certificação Digital Nº 1612825/CA

[30] **wikitexto**. https://pt.wikipedia.org/wiki/Wikitexto. Acessado: 11/8/2018.

[31] BERNERS-LEE, T.; HENDLER, J. ; LASSILA, O.. **The semantic web**. Scientific american, 284(5):34–43, 2001.

[32] **Dbpedia datasets**. http://dbpedia.org/ontology/. Acessado: 21/7/2018.

[33] **Interwikis**. https://pt.wikipedia.org/wiki/Ajuda:Guia_de_edi%C3%A7%C3%A3o/Interwikis. Acessado: 11/8/2018.

[34] **Dbpedia datasets**. https://dbpedia.org/sparql. Acessado: 21/7/2018.

[35] **livro de estilo**. https://pt.wikipedia.org/wiki/Wikip%C3%A9dia:Livro_de_estilo. Acessado: 11/8/2018.

[36] **mwparserfromhell**. https://mwparserfromhell.readthedocs.io/en/latest/. Acessado: 11/8/2018.

[37] AL-RFOU, R.; KULKARNI, V.; PEROZZI, B. ; SKIENA, S.. **Polyglot-ner: Massive multilingual named entity recognition**. In: PROCEEDINGS OF THE 2015 SIAM INTERNATIONAL CONFERENCE ON DATA MINING, p. 586–594. SIAM, 2015.

[38] BIRD, S.; LOPER, E.. **Nltk: the natural language toolkit**. In: PROCEEDINGS OF THE ACL 2004 ON INTERACTIVE POSTER AND DEMONSTRATION SESSIONS, p. 31. Association for Computational Linguistics, 2004.

[39] KISS, T.; STRUNK, J.. **Unsupervised multilingual sentence boundary detection**. Comput. Linguist., 32(4):485–525, Dec. 2006.

[40] DOS SANTOS, C. N.; MILIDIÚ, R. L.. **Entropy guided transformation learning**. In: FOUNDATIONS OF COMPUTATIONAL, INTELLIGENCE VOLUME 1, p. 159–184. Springer, 2009.

[41] COLLOBERT, R.; WESTON, J.; BOTTOU, L.; KARLEN, M.; KAVUKCUOGLU, K. ; KUKSA, P.. **Natural language processing (almost) from scratch**. Journal of Machine Learning Research, 12(Aug):2493–2537, 2011.

[42] SANTOS, C. D.; ZADROZNY, B.. **Learning character-level representations for part-of-speech tagging**. In: PROCEEDINGS OF THE 31ST INTERNATIONAL CONFERENCE ON MACHINE LEARNING (ICML-14), p. 1818–1826, 2014.

PUC-Rio - Certificação Digital Nº 1612825/CA

[43] GOLLER, C.; KUCHLER, A.. **Learning task-dependent distributed representations by backpropagation through structure**. Neural Networks, 1:347–352, 1996.

[44] BENGIO, Y.; SIMARD, P. ; FRASCONI, P.. **Learning long-term dependencies with gradient descent is difficult**. IEEE transactions on neural networks, 5(2):157–166, 1994.

[45] GERS, F. A.; SCHMIDHUBER, J. ; CUMMINS, F.. **Learning to forget: Continual prediction with lstm**. 1999.

[46] GRAVES, A.; SCHMIDHUBER, J.. **Framewise phoneme classification with bidirectional lstm and other neural network architectures**. Neural Networks, 18(5-6):602–610, 2005.

[47] GRAVES, A.; MOHAMED, A.-R. ; HINTON, G.. **Speech recognition with deep recurrent neural networks**. In: ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP), 2013 IEEE INTERNATIONAL CONFERENCE ON, p. 6645–6649. IEEE, 2013.

[48] HUANG, Z.; XU, W. ; YU, K.. **Bidirectional LSTM-CRF models for sequence tagging**. CoRR, abs/1508.01991, 2015.

[49] TRAN, Q.; MACKINLAY, A. ; YEPES, A. J.. **Named entity recognition with stack residual lstm and trainable bias decoding**. arXiv preprint arXiv:1706.07598, 2017.

[50] LAMPLE, G.; BALLESTEROS, M.; SUBRAMANIAN, S.; KAWAKAMI, K. ; DYER, C.. **Neural architectures for named entity recognition**. arXiv preprint arXiv:1603.01360, 2016.

[51] MA, X.; HOVY, E.. **End-to-end sequence labeling via bi-directional lstm-cnns-crf**. arXiv preprint arXiv:1603.01354, 2016.

[52] ARANHA, C. N.. **O cortex e a sua participaçao no harem**. Reconhecimento de entidades mencionadas em português: Documentaçao e actas do HAREM, a primeira avaliaçao conjunta na área. Linguateca, p. 113–122, 2007.

[53] BRILL, E.. **Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging**. Computational linguistics, 21(4):543–565, 1995.

PUC-Rio - Certificação Digital Nº 1612825/CA

[54] FINKEL, J. R.; GRENAGER, T. ; MANNING, C.. **Incorporating non-local information into information extraction systems by gibbs sampling**. In: PROCEEDINGS OF THE 43RD ANNUAL MEETING ON ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, p. 363–370. Association for Computational Linguistics, 2005.

[55] SNOEK, J.; LAROCHELLE, H. ; ADAMS, R. P.. **Practical bayesian optimization of machine learning algorithms**. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 2951–2959, 2012.

[56] PENNINGTON, J.; SOCHER, R. ; MANNING, C.. **Glove: Global vectors for word representation**. In: PROCEEDINGS OF THE 2014 CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING (EMNLP), p. 1532–1543, 2014.

[57] JOULIN, A.; GRAVE, E.; BOJANOWSKI, P. ; MIKOLOV, T.. **Bag of tricks for efficient text classification**. In: PROCEEDINGS OF THE 15TH CONFERENCE OF THE EUROPEAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: VOLUME 2, SHORT PAPERS, p. 427–431. Association for Computational Linguistics, April 2017.

[58] MIKOLOV, T.; CHEN, K.; CORRADO, G. ; DEAN, J.. **Efficient estimation of word representations in vector space**. arXiv preprint arXiv:1301.3781, 2013.

[59] PATEL, K.; BHATTACHARYYA, P.. **Towards lower bounds on number of dimensions for word embeddings**. In: PROCEEDINGS OF THE EIGHTH INTERNATIONAL JOINT CONFERENCE ON NATURAL LANGUAGE PROCESSING (VOLUME 2: SHORT PAPERS), volumen 2, p. 31–36, 2017.

[60] HARTMANN, N.; FONSECA, E. R.; SHULBY, C.; TREVISO, M. V.; RODRIGUES, J. ; ALUÍSIO, S. M.. **Portuguese word embeddings: Evaluating on word analogies and natural language tasks**. CoRR, abs/1708.06025, 2017.

[61] HE, K.; ZHANG, X.; REN, S. ; SUN, J.. **Delving deep into rectifiers: Surpassing human-level performance on imagenet classification**. CoRR, abs/1502.01852, 2015.

[62] GAL, Y.; GHAHRAMANI, Z.. **A theoretically grounded application of dropout in recurrent neural networks**. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 1019–1027, 2016.

PUC-Rio - Certificação Digital Nº 1612825/CA

[63] DAI, H.-J.; LAI, P.-T.; CHANG, Y.-C. ; TSAI, R. T.-H.. **Enhancing of chemical compound and drug name recognition using representative tag scheme and fine-grained tokenization**. Journal of cheminformatics, 7(S1):S14, 2015.

[64] GOLDBERGER, J.; BEN-REUVEN, E.. **Training deep neural-networks using a noise adaptation layer**.

[65] VAN DEN BERG, E. M.. **Noisy Label Neural Network Approach to Named Entity Recognition**. PhD thesis, University of Groningen - University of Saarlandes, 2016.

[66] BAI, S.; KOLTER, J. Z. ; KOLTUN, V.. **An empirical evaluation of generic convolutional and recurrent networks for sequence modeling**. arXiv preprint arXiv:1803.01271, 2018.

[67] GHADDAR, A.; LANGLAIS, P.. **Robust lexical features for improved neural network named-entity recognition**. arXiv preprint arXiv:1806.03489, 2018.

[68] YADAV, V.; SHARP, R. ; BETHARD, S.. **Deep affix features improve neural named entity recognizers**. In: PROCEEDINGS OF THE SEVENTH JOINT CONFERENCE ON LEXICAL AND COMPUTATIONAL SEMANTICS, p. 167–172, 2018.

PUC-Rio - Certificação Digital Nº 1612825/CA