Enriquecimento Semi-Automático de uma Ontologia de Domínio em Português Utilizando Redes Neurais Recorrentes

Moniele K. Santos[1], **Larissa A. de Freitas**[1]

[1]Federal University of Pelotas (UFPel)
Pelotas, RS, Brazil

{mksantos,larissa}@inf.ufpel.edu.br

Abstract. *Domain ontologies are useful data structures for tasks that need structured information. However they can become static after some time as there is a need to update them regularly. Ontology enrichment is a form of knowledge acquisition, which proposes to update the concepts and relationships in the domain ontology structure. Thus, the present work stands out for presenting a proposal for semi-automatic enrichment of domain ontologies in Portuguese. We are using HOntology, a Portuguese domain ontology of hotels, as a case study. And we will be enriching their relations of "synonymy", "hyperonymy", "hyponymy" and "hasCategory". In this experiment, an average F-Measure of 0.69 was obtained using two BiLSTMs, which given two candidate terms, classified them in some type of HOntology relationship.*

Resumo. *Ontologias de domínio são estruturas de dados úteis para tarefas que necessitam de informações estruturadas. Contudo elas podem se tornar estáticas depois de certo tempo, visto que há uma necessidade de atualizá-las regularmente. O enriquecimento de ontologias é uma forma de aquisição de conhecimento, que propõe atualizar os conceitos e relações na estrutura da ontologia de domínio. Desta forma, o presente trabalho destaca-se por apresentar uma proposta de enriquecimento semi-automática de ontologias de domínio em português. Estamos utilizando a HOntology, uma ontologia de domínio em português de hotéis, como estudo de caso. E estaremos enriquecendo as suas relações de "sinonímia", "hiperonímia", "hiponímia" e "temCategoria". Neste experimento, obteve-se uma Medida-F média de 0,69 utilizando duas BiLSTMs, que dado dois termos candidatos, os classificou em algum tipo de relação da HOntology.*

1. Introdução

As ontologias de domínio são úteis para tarefas que necessitam de dados estruturados. Entretanto, elas podem se tornar estáticas após certo tempo, visto que há uma necessidade de atualizá-las regularmente. O enriquecimento de ontologias é uma forma de aquisição de conhecimento [Petasis et al. 2011], que propõe atualizar os conceitos e as relações na arquitetura da ontologia de domínio.

Segundo a literatura, existem dois passos para enriquecer uma ontologia de domínio, sendo eles: 1) uma parcela de tempo significativo para analisar e extrair manualmente, dentre um grande volume de dados, relações e conceitos úteis, específicos do domínio que

possam ser utilizados para atualizar a ontologia; 2) um ou mais especialistas de domínio, que terão capacidade de decidir modificações válidas [Velardi et al. 2001].

O propósito principal deste trabalho é facilitar o processo de atualizar e expandir uma ontologia de domínio em português, a partir de um algoritmo que consiga abstrair relações da ontologia e classificar novos conceitos a essas relações existentes, que devem ser aprovados através de uma avaliação manual realizada por especialistas de domínio.

O restante do trabalho está organizado da seguinte forma. No Capítulo 2 é descrito o Referencial Teórico e Tecnológico, onde consta a contextualização das tecnologias utilizadas; no Capítulo 3 são apresentados os Trabalhos Relacionados, os quais possuem alguma semelhança com o assunto deste trabalho; no Capítulo 4 é apresentada a Abordagem Proposta; no Capítulo 5 são apresentados os Resultados Obtidos; e por fim, no Capítulo 6 é discutida a Conclusão da abordagem proposta e trabalhos futuros.

2. Referencial Teórico

2.1. HOntology

HOntology (em que 'H' significa hotel, hostal e hostel) é uma ontologia de domínio, disponível gratuitamente em quatro idiomas: inglês, português, espanhol e francês. Os autores [Chaves et al. 2012] criaram essa ontologia do setor de acomodação a partir de *reviews online* e outros vocabulários como DBPedia[1] e Schema[2]. A estrutura da HOntology contém 282 conceitos categorizados em 16 conceitos *top-level*. A hierarquia de conceitos tem uma profundidade máxima de 5.

2.2. Enriquecimento de Ontologias

Já sabemos que as ontologias de domínio contém conceitos e relações de um determinado campo do conhecimento. Porém, sua estrutura tende a mudar eventualmente com o surgimento de novos termos e regras. O enriquecimento de ontologias visa atualizar a arquitetura da ontologia com a inserção de novos conceitos e/ou relações. Nesse tópico específico da Computação, são realizadas pesquisas sobre métodos e técnicas para o enriquecimento focando na aquisição de uma ontologia atualizada, baseada em informações semânticas, extraídas a partir de dados/textos de um domínio específico. Assim, exitem duas formas de se enriquecer uma ontologia: manual e semi-automática. A forma manual consiste em uma seleção manual de conceitos para a atualização que é enviada para a análise de um especialista de domínio. A forma semi-automática que consiste na automatização de alguma forma na busca por conceitos e relações que possam ser adicionados a ontologia e que depois é enviada para validação por um especialista de domínio [Petasis et al. 2011]. São encontrados na literatura trabalhos com abordagens de enriquecimento dos dois tipos como [Poetsch et al. 2019] que aplica o enriquecimento manual da HOntology e [Mohan Sanagavarapu et al. 2021] que apresenta ser o estado-da-arte no enriquecimento semi-automático de ontologias de domínio.

2.3. Redes Neurais Recorrentes

As Redes Neurais Recorrentes (RNRs), segundo [Goodfellow et al. 2016], são *"a família das redes neurais que processam dados sequenciais"*. O seu nome é usado pra referenciar

[1] www.dbpedia.org
[2] www.schema.org

duas amplas classes de redes com uma estrutura geral semelhante, onde uma é impulso finito e a outra é impulso infinito. O impulso finito é um grafo acíclico direto que pode ser desenrolado e substituído por uma rede neural *feedforward*, enquanto uma rede recorrente de impulso infinito é um grafo cíclico direcionado que não pode ser desenrolado. Dentro da arquitetura da RNR, existe uma memória dos elementos anteriores. Esse é um aspecto muito útil para tarefas de Processamento de Linguagem Natural (PLN) como descrevem [Manning and Schütze 1999] e [Otter et al. 2020].

As RNRs que foram utilizadas no nosso estudo de caso são:

- ***Long Short-Term Memory* (LSTM)** [Hochreiter and Schmidhuber 1997]: é uma RNR com ótimos resultados em tarefas gerais de classificação de texto, como documentado em [Greff et al. 2016, Xingjian et al. 2015, Gers et al. 1999]. A ideia central por trás de sua arquitetura, e a que difere do resto da literatura Deep Learning (DL), é ter uma célula de memória a qual pode manter seu estado continuamente.
- **LSTM Bidirecional (BiLSTM)** [Huang et al. 2015]: LSTM Bidirecional ou BiLSTM, é um modelo de processamento sequencial que possui dois modelos LSTM em sua arquitetura. Usando uma Bidirecional, o modelo executará suas entradas de duas maneiras: uma do passado para o futuro e outra do futuro para o passado. O que difere essa abordagem da unidirecional, é que na LSTM que funciona em sentido contrário, preservando as informações do futuro e usando os dois estados ocultos combinados são capazes de, a qualquer momento, preservar informações do passado e do futuro. BiLSTMs aumentam efetivamente a quantidade de informações disponíveis para a rede, melhorando o contexto disponível para o algoritmo, como argumenta [Siami-Namini et al. 2019].

3. Trabalhos Relacionados

Pesquisas que utilizem Aprendizado de Máquina (AM) para enriquecer ou expandir uma ontologia ainda são escassas. Porém, nos últimos anos, podemos citar alguns trabalhos que propõem formas semi-automáticas de enriquecer ontologias de domínio. Dentre elas, há o trabalho de [Althubaiti et al. 2020], que corresponde a um modelo genérico que enriquece classes e subclasses de uma ontologia de doenças e que se baseia em Word Embedding (WE) e AM. A pesquisadora utiliza uma rede neural simples com apenas uma camada de neurônios, e fomenta no trabalho que essa configuração basta para seus resultados alvo. Na classificação de doença ou não doença (2 classes) e de 17 classes diferentes de doenças infecciosas e anatômicas, e obteve um Medida-F de 95% e 73%, respectivamente.

Em 2015, foi desenvolvida uma aplicação web [Xiang et al. 2015], que tem o objetivo enriquecer ontologias, de modo que é fornecido como entrada, a ontologia e as modificações desejadas. Esta abordagem baseia-se em padrões de projeto de ontologia. É uma ferramenta útil, mas apenas para ontologias que possuem modificações pontuais e conhecidas. Não é uma proposta de construção de ontologia de forma automática, pois ela não aprende sozinha, novas classes ou relações, é preciso do auxílio humano.

Por outro lado, existem métodos que convertem *corpus* em espaço vetorial, que pode ser manipulado para alguma tarefa ou trabalho específico. O trabalho de [Smaili et al. 2018] gera representações vetoriais de entidades biológicas em ontologias,

combinando axiomas formais de ontologia e axiomas de anotação, a partir dos metadados da ontologia. Em uma primeira etapa, os autores usam Word2Vec para gerar o vetor, e em uma segunda etapa, eles avaliam o modelo com uma predição de associações gene-doença com base em similaridade fenotípica gerando representações vetoriais de genes e doenças usando uma ontologia de fenótipo.

O trabalho de [Poetsch et al. 2019] usa WE para enriquecer ontologias do setor hoteleiro. Ele testa vários modelos de WE, dando destaque para o Wang2Vec, que obteve os melhores resultados. Apesar de conseguir extrair novas classes e subclasses para enriquecer a ontologia, a alteração e/ou inserção de novas classes e subclasses não foram feitas, pois requeria a autorização de um especialista de domínio. Além disso, os espaços vetoriais foram analisados manualmente pelo próprio pesquisador, criando um gargalo para o processo.

Em [Mohan Sanagavarapu et al. 2021], os autores enriquecem a ontologia de Segurança da Informação com modelos BiLSTM [Graves and Schmidhuber 2005], a partir de dados extraídos da DBpedia e da Wikipédia. A Acurácia obtida foi de 80%.

Percebe-se que não há nenhum dos trabalhos citados enriquece ontologias de domínio da língua portuguesa de forma semi-automática . Este trabalho destaca-se por apresentar uma estratégia que consegue automatizar o processo de encontrar novas classes e relacioná-las com classes já estabelecidas na ontologia de domínio do setor hoteleiro.

Infelizmente é custoso construir um trabalho que enriqueça uma ontologia de domínio de forma completamente automática, pois ela sempre estará intrinsecamente ligada a um especialista do domínio. Dito isto, a presença de um especialista é imprescindível, esse tipo de profissional terá o papel de validar as saídas processadas pelo modelo probabilístico tornando o processo de atualização da ontologia válida e confiável.

4. Abordagem Proposta

Para o desenvolvimento desse trabalho, criamos um fluxo de passos (Figura 1), baseado em [Mohan Sanagavarapu et al. 2021], que utilizaram dados da DBPedia e alimentaram uma Rede Neural com pares de termos e caminhos de dependência para prever alguma relação entre os termos candidatos. Portanto, como é apresentado na Figura 1, começamos fazendo uma coleta de dados, que é então pré-processada. Logo a seguir dividimos os dados em dois conjuntos, um de treinamento e outro de teste. O conjunto de treinamento é usado em uma RNR, com a função de encontrar padrões a partir da entrada de dados, sendo composta de um par de termos e o tipo de relação entre os dois termos. O conjunto de teste servirá como avaliador dessa RNR já treinada, portanto, nos retornará como saída termos candidatos que expandirão a ontologia HOntology. A seguir, serão mostrados os passos do método proposto de forma detalhada.

4.1. Conjunto de Dados

Para criação do Conjunto de Dados, extraímos informações de 3 recursos diferentes. Primeiro, como mostrado no experimento do trabalho de [Mohan Sanagavarapu et al. 2021], é preciso capturar todos os conceitos da ontologia de domínio, que se tornam uma base de informação confiável e de fundamental importância no aprendizado do classificador. Em seguida, utilizamos essa lista de conceitos para servir como sementes em consultas SPARQL na estrutura ontológica da DBpedia. Além disso, utilizamos um dicio-

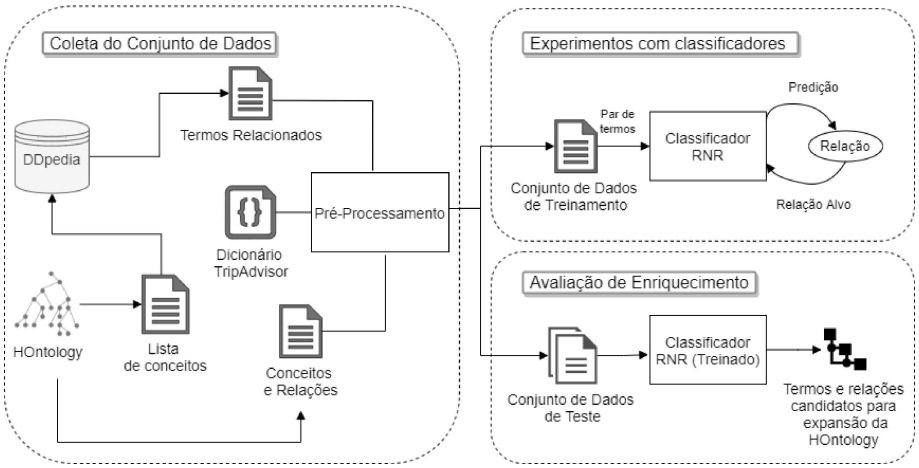

Figura 1. Fluxograma da abordagem proposta. Fonte: Própria.

nário contendo as palavras mais recorrentes nas *reviews* do TripAdvisor, coletadas por [Barbosa 2015] e filtradas por [Poetsch et al. 2019].

A configuração do Conjunto de Dados, é composta por 5.706 triplas *(termo_1, termo_2, relação)*. Onde as relações ontológicas entre "termo_1" e "termo_2" podem ser dos seguintes tipos: "hiponímia", "hiperonímia", "sinonímia", "temCategoria" ou "sem relação".

4.1.1. Conceitos e Relações da HOntology

O primeiro recurso utilizado foi a HOntology. Com o auxílio da biblioteca *pronto*[3] em *Python* extraímos da representação OWL da HOntology 304 conceitos, 516 triplas das relações "hiperonímia" ou "hiponímia" retiradas de todos os níveis de sua taxonomia e 2 triplas da relação "temCategoria" retiradas das propriedades de objeto da HOntology.

4.1.2. Termos Relacionados da DBpedia

O segundo recurso utilizado foi a DBpedia. Utilizamos a lista de conceitos extraída na etapa anterior para consultar termos relacionados na estrutura ontológica da DBpedia. Nela é possível navegar dentro de um termo e encontrar diversos tipos de relações e propriedades. Por exemplo, a propriedade *isPrimaryTopicOf* denota que um conceito é tópico principal dentro de outro conceito.

Nesse sentido, verificamos quais conceitos existiam na DBpedia em português. Dos 304 conceitos, apenas 160 possuem páginas de artigos da Wikipédia em português. Portanto, extraímos todas as propriedades contidas na estrutura desses termos. Porém, notou-se que várias propriedades não demonstravam nenhum tipo de relação, como: número de páginas, resumo do artigo, identificador do artigo, entre outras. Por esse motivo, selecionamos apenas as propriedades com alguma ligação com outros termos e que apresentaram retorno nas consultas. Sendo elas:

[3]https://pypi.org/project/pronto/

- *wikiPageLink*[4] : propriedade que denota que uma página da Wikipédia tem link para outra página da Wikipédia.
- *Subject*[5] : propriedade que denota o tópico do recurso.
- *wikiPageRedirects*[6] : propriedade que denota o redirecionamento da Wikipédia. Onde os redirecionamentos são páginas que redirecionam automaticamente o navegador do leitor para uma página de destino especificada.
- *is_wikiPageLink_of*: propriedade de sentido inverso da propriedade *wikiPageLink*.

A partir das consultas SPARQL, foram geradas triplas com linhas no formato (*conceito_ontologia*, *pagina_wiki*, *relação*), em que "conceito_ontologia" representa o termo que foi consultado, 'pagina_wiki' representa a página da Wikipédia e 'relação' representa a relação (*WikiPageLink*, *Subject*, *wikiPageRedirects* e *wikiPageRedirects*) . Com isso, conseguimos coletar ao todo, 374 propriedades *WikiPageLink*, 19 propriedades *Subject*, 15 propriedades *wikiPageRedirects* e 7 propriedades *isWikiPageLink*.

Ainda, classificamos a relação entre os termos manualmente, e notou-se que 15 das 19 propriedades *Subject*, apontavam ser páginas da Wikipédia do tipo Categoria. Portanto, classificamos esses termos com a relação do tipo "temCategoria". Além disso, encontramos exemplos de "hiperonímia", "hiponímia" e "sinonímia".

Contudo, os pares de termos que não conseguimos estabelecer uma lógica que apontasse para alguma relação específica da HOntology. E portanto essas relações foram classificadas como "sem relação", que embora não tenham uma das 4 relações escolhidas, ainda tem potencial de corresponderem a outro tipo de relação não abordada nesse trabalho, já que os dados são do mesmo domínio.

4.1.3. Dicionário do TripAdvisor

O terceiro recurso utilizado foi o dicionário com classificação POS *Tagging* contendo *reviews* do site TripAdvisor, criado no trabalho [Poetsch et al. 2019] e cordialmente compartilhado conosco. Este dicionário tem em sua composição total 11.161 palavras, sendo 4.052 classificadas como verbos e 7.109 classificadas como substantivos. Dito isso, selecionamos apenas os substantivos e utilizamos um modelo *Wang2Vec* de 100 dimensões, treinado a partir de um corpus em português e específico do setor hoteleiro, para medir a similaridade entre esses termos retirados das *reviews* de hotéis com os conceitos da ontologia, utilizando método "similarity" da biblioteca *gensim*[7]. E apenas os que tivessem grau de similaridade maior que 0,99 são classificados como sinônimos dos conceitos e herdam todas as relações do mesmo. Então se, por exemplo, um substantivo A tem um grau *n* de similaridade com o conceito B da ontologia; e ainda se *n* for maior que 0,99, então A tem relação de "sinonímia" com B e consequentemente, herda as relações de "hiperonímia", "hiponímia" e "temCategoria" que A possui com outros conceitos. Deste modo, se A é subclasse de C, e B é sinônimo de A, logo B é subclasse de C.

[4]https://dbpedia.org/ontology/wikiPageWikiLink
[5]http://purl.org/dc/elements/1.1/subject
[6]https://dbpedia.org/ontology/wikiPageRedirects
[7]https://radimrehurek.com/gensim/

Por fim, foram geradas triplas com linhas no formato (*termo_dic*, *conceito_ ontologia* , *relação*), em que “termo_dic” representa o substantivo similar retirado do dicionário, “conceito_ontologia” representa o conceito da ontologia que o “termo_dic” é similar e a “relação” simboliza o tipo de relação entre às duas primeiras colunas. Com isso, conseguimos coletar 231.690 triplas, sendo 61.010 de relações de “sinonímia” e 170.680 de relações de “hiperonímia” ou “hiponímia”.

4.2. Pré-Processamento dos Dados

Com o objetivo de facilitar a classificação da RNR no treinamento e teste, preparamos o conjunto de dados que será inserido na entrada do modelo. Para isso, executamos alguns pré-processamentos: remoção de *stopwords*, tokenização, remoção de quebras de linhas e caracteres especiais.

A remoção de *stopwords* tem o propósito de remover palavras que aparecem com alta frequência nos exemplos e que carregam pouco significado. Para essa tarefa foi utilizada a biblioteca em Python, *Natural Language Toolkit* (NLTK) [Bird et al. 2009]. Para implementar a tokenização, utilizou-se o componente de processamento de texto da biblioteca Keras[8], com tamanho de vocabulário de 20.000 palavras. Além disso, todas as palavras passam por uma transformação *lower case*, e quebras de linha e caracteres especiais (tais como: ! " $ % ’()*+,-./:;<=>?@[ ] ^ ` { | } ~) são removidos. Quando há bi-gramas ou e tri-gramas, isto é, quando o termo contém duas palavras e três palavras, respectivamente, os espaços são substituídos por ”_”s. Como por exemplo a tri-grama “café da manhã ”, é mapeada para “cafe_da_manha”.

4.3. Experimentos com Classificadores

As duas RNRs utilizadas para a tarefa de classificação de novos conceitos da ontologia de domínio são a LSTM e a BiLSTM. A LSTM é ainda uma das RNRs mais referenciadas na classificação de texto e foi escolhida por ter um alto nível de aceitação dentro da literatura DL. Por outro lado, [Mohan Sanagavarapu et al. 2021] foi o trabalho que obteve os melhores resultados no enriquecimento de novos conceitos de ontologias de domínio, utilizando duas BiLSTM como classificador.

4.3.1. Experimento 1: Classificador RNR com 1 entrada

Logo após a concatenação dos pares de termos em uma única entrada, é feito um processo de *Embedding*, onde utilizamos o modelo Wang2Vec com 50 dimensões do Repositório Word Embedding NILC[9] para a conversão palavra-vetor. Seguindo a estrutura do modelo, a saída dessa etapa é passada para uma sequência de camadas, sendo elas: *dropout* de 0.5, Convolucional de 1 dimensão e *max pooling* de tamanho 4. A partir da computação desse processamento, o seu resultado é inserido em uma RNR. Testando a LSTM simples, utilizamos 100 camadas escondidas. Por outro lado, com a BiLSTM foi testado dois tipos de configurações: a primeira configuração, com uma única camada BiLSTM que contem 200 camadas escondidas; e a segunda configuração, com 2 camadas BiLSTM, uma atrás da outra, contendo respectivamente, 200 e 100 camadas escondidas. E por fim, uma camada de saída de tamanho 5, com função de ativação *softmax*.

[8]https://keras.io/

[9]http://www.nilc.icmc.usp.br/embeddings

4.3.2. Experimento 2: Classificador RNR com 2 entradas

Nessa estratégia, é criado um modelo que divide a camada de entrada em duas. Estas duas entradas, possuem camadas de *Embedding* Wang2Vec com 50 dimensões, *dropout* de 0.5, Convolucional de 1 dimensão e *max pooling* de tamanho 4, vistas no método anterior. Porém, nesse momento, elas convergem numa camada de concatenação, para então, serem direcionadas como entrada da LSTM ou BiLSTM. Testamos as mesmas configurações nas RNRs, citadas na Subseção anterior, que são posteriormente adicionadas na camada de saída de tamanho 5, com função de ativação *softmax*.

4.4. Avaliação de Enriquecimento

Para avaliar o desempenho do modelo em seu treinamento e teste, usamos as métricas de Acurácia, Precisão, Revocação e Medida-F. Segundo [Sokolova et al. 2006], o uso de métricas é muito importante para classificar e analisar o aprendizado de um classificador. O autor ainda frisa no uso de Medida-F com modelos de multi-classificação desbalanceados, pois realiza uma média proporcional entre outras duas medidas, a Precisão e a Revocação.

Além disso, estamos contabilizando o número de termos encontrados, isto é, a soma total de amostras em que o modelo acertou o tipo de relação, sendo elas: "hiponímia", "hiperonímia", "sinonímia" e "temCategoria". Essa medida é uma aproximação do número real de termos, visto que existem algumas triplas com os mesmos termos e relações diferentes. Então dessa forma, comparamos o número de termos que existem na HOntology (304 conceitos) com os conceitos classificados positivamente e validados pelo modelo neural treinado.

5. Resultados Obtidos

No primeiro tipo de experimento, contendo 1 entrada no modelo e utilizando o otimizador "Adadelta", foram calculadas as métricas a partir do desempenho das RNRs na fase de teste, como demonstra a Tabela 1. Como podemos ver, os melhores resultados foram obtidos pelo modelo que contem duas BiLSTMs, concordando com a abordagem do trabalho [Mohan Sanagavarapu et al. 2021].

Métrica	LSTM	BiLSTM	
		1 BiLSTM	2 BiLSTMs
Nº de Termos	366	399	**428**
Acurácia	0,47	0,47	**0,50**
Precisão	0,37	0,44	**0,47**
Revocação	0,43	0,47	**0,50**
Medida-F	0,40	0,45	**0,46**

Tabela 1. Métricas por classificador RNR - Experimento Utilizando 1 entrada. Fonte: Própria.

É importante destacar que o número de termos denota o número de termos preditos como corretos na fase de teste. Portanto, observa-se que o maior número de termos anotado neste experimento, mostra que a rede neural com duas BiLSTMs conseguiu identificar 428 conceitos, 124 a mais que os definidos na HOntology.

Percebeu-se que, ambas LSTM e BiLSTM, não estavam conseguindo distinguir as classes, principalmente entre "hiponímia" e "hiperonímia". É possível visualizar na primeira linha da Matriz de Confusão, os exemplos que são de relação "hiponímia", classificados em sua maioria como relações de "hiperonímia" (43 exemplos) e "sinonímia" (58 exemplos).

Tendo em vista esse comportamento, levantou-se a hipótese do modelo não estar aprendendo a diferença entre a ordem dos termos. Pois, é possível existir, por exemplo, a tripla (termo1, termo2, hiponímia), onde "termo1" é subclasse de "termo2", e a tripla (termo2, termo1, hiperonímia), onde "termo2" é superclasse de "termo1".

Portanto, viu-se razoável testar um novo tipo de abordagem. Criamos um experimento de modelos com duas entradas distintas, uma para cada termo, de modo a melhorar o desempenho, por classe, da rede.

Além da separação dos termos por entrada, no segundo experimento foi testada a diferença no desempenho de dois otimizadores, "RMSprop" e "Adadelta". Observa-se que modelos com o otimizador "RMSprop" obtiveram resultados melhores quando comparados aos do otimizador "Adadelta", como demonstra a Tabela 2.

Métricas	LSTM		BiLSTM			
	Adadelta	RMSprop	1 BiLSTM		2 BiLSTMs	
			Adadelta	RMSprop	Adadelta	RMSprop
Nº de Termos	397	579	500	574	450	**598**
Acurácia	0,41	0,68	0,46	0,68	0,49	**0,70**
Precisão	0,37	0,67	0,30	0,70	0,50	**0,70**
Revocação	0,38	0,68	0,48	0,68	0,39	**0,70**
Medida-F	0,37	0,66	0,40	0,66	0,43	**0,69**

Tabela 2. Métricas por classificador RNR e otimizador - Experimento Utilizando 2 entradas. Fonte: Própria.

Os experimentos com duas entradas apresentaram resultados inferiores quando utilizado o otimizador "Adadelta". Pois como vimos anteriormente (Tabela 1), o modelo que com 2 BiLSTMs desempenhou Medida-F de 0,46, agora com duas entradas apresenta o valor 0,43 na mesma métrica. Embora exista tal piora de resultados na micro-média dos modelos, as métricas *por classe* tem uma distribuição melhor se comparadas com as do modelo de 1 entrada.

No histograma (Figura 2) com métricas por classe da BiLSTM com 1 entrada, nota-se as medidas de Precisão, Revocação e Medida-F da classe "hiponímia" apresentarem valores nulos, expressando que a rede neural não conseguiu acertar nenhum exemplo dessa classe.

Além disso, existem classes mostradas no histograma que tem suas métricas nulas, as classes de "Hiponímia", "temCategoria" e "sem relação". Portanto, viu-se necessário fazer um balanceamento dos pesos de cada classe, ou seja, classes com um número maior de amostras possuem um peso menor e classes com um número menor de amostras possuem um peso maior. Os pesos por classe foram calculados com a função "com-

pute_class_weight" da biblioteca *Sklearn*[10]. Esses pesos são enviados juntamente com o conjunto de dados na etapa de treinamento, para que o modelo leve em consideração a distribuição distorcida das classes e que, em teoria, otimizará a sua generalização.

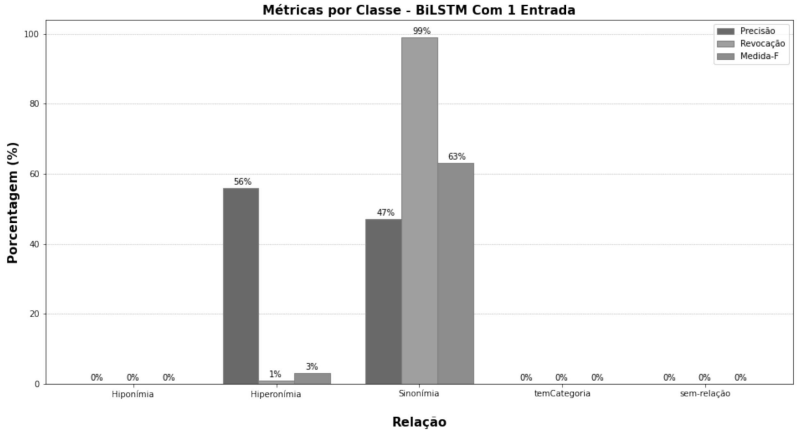

Figura 2. Histograma com métricas por classe da BiLSTM com 1 entrada. Fonte: Própria.

Foi escolhida a rede neural com o melhor resultado dentre todos os experimentos anteriores, ou seja, o modelo com 2 BiLSTMs de 2 entradas (Figura 3) e otimizador "RMSprop", com o objetivo de fazer um último experimento testando o balanceamento de pesos por classe.

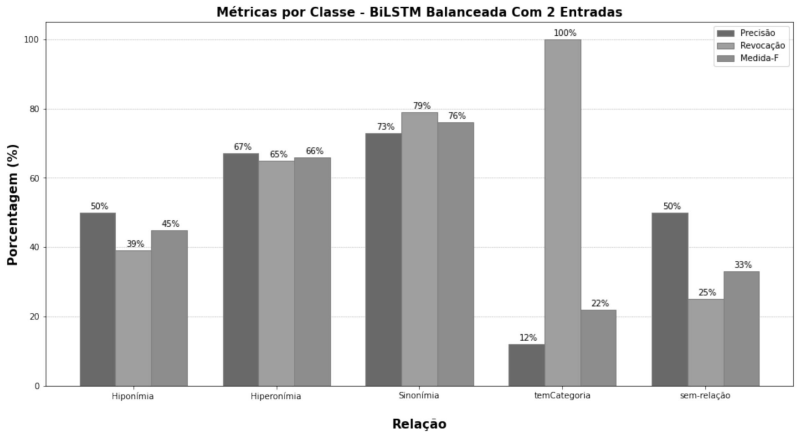

Figura 3. Histograma com métricas por classe da BiLSTM com 2 entradas e Balanceamento de Peso por Classe. Fonte: Própria.

Podemos ressaltar que, em todos os modelos experimentados, o número de termos com relações preditas corretamente na fase de teste, ainda é maior que o número de conceitos da própria HOntology. Isso prova que os modelos construídos conseguiram abstrair padrões de textos vindos do domínio de hotéis. Dando destaque ao modelo com 2 BiLSTMs e balanceamento de pesos por classes, que apresentou uma média de Precisão por classe razoável, tendo em vista a pouca quantidade de dados, principalmente nas relações "temCategoria" e "sem relação".

[10]https://scikit-learn.org/stable/

E quando se obtido os termos e eventualmente até a inserção de novos dados para predição de relação, será então possível notificar algum especialista do domínio para fazer o enriquecimento propriamente dito da HOntology.

6. Conclusões e Trabalhos Futuros

Este trabalho apresentou uma abordagem para enriquecer de forma semi-automática ontologias de domínio em Português, com uso de Redes Neurais Recorrentes. E conclui-se, através de seus resultados, de que a estratégia utilizada mostra-se promissora, visto que apesar de o número de amostras no conjunto de dados ser pequeno, no melhor experimento, nosso modelo demonstrou ter aprendido a generalizar os conceitos referentes aos cinco tipos de relação escolhidos para o enriquecimento da HOntology ("hiponímia", "hiperonímia", "sinonímia", "temCategoria" ou "sem relação").

O conjunto de dados em português[11] demonstrou ser demasiada pequena, quando comparada ao de trabalhos relacionados que enriquecem ontologias na língua inglesa [Mohan Sanagavarapu et al. 2021, Althubaiti et al. 2020], o que limita o desempenho das redes neurais. Ainda, a abordagem proposta gerou modelos que abstraíram conceitos e relações do domínio de hotéis, e que demonstraram classificar conceitos e relações *candidatas* a um enriquecimento, este que apenas acontecerá depois da devida aprovação de um ou vários especialistas de domínio.

Como trabalho futuro, é imprescindível a presença de especialistas do domínio de hotelaria que consigam avaliar de forma manual o conjunto de dados, pois no presente trabalho uma única pessoa verificou os exemplos mostrados ao classificador.

Referências

Althubaiti, S., Kafkas, S., Abdelhakim, M., and Hoehndorf, R. (2020). Combining lexical and context features for automatic ontology extension. *Journal of Biomedical Semantics*, 11.

Barbosa, R. R. L. (2015). *Aplicación del análisis de sentimientos a la evaluación de datos generados en medios sociales*. PhD thesis, Universidad de Alcalá, Espanha.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. "O'Reilly Media, Inc.", Massachusetts.

Chaves, M. S., de Freitas, L. A., and Vieira, R. (2012). Hontology: A multilingual ontology for the accommodation sector in the tourism industry. In Filipe, J. and Dietz, J. L. G., editors, *KEOD*, pages 149–154. SciTePress.

Gers, F. A., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: Continual prediction with lstm. In *International Conference on Artificial Neural Networks*. IET.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press, Massachusetts. `http://www.deeplearningbook.org`.

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

[11] https://github.com/mksantos2/semiautomatic-portuguese-ontology-enrichment/

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2016). Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232.

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

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *CoRR*.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Mohan Sanagavarapu, L., Iyer, V., and Raghu Reddy, Y. (2021). Ontoenricher: A deep learning approach for ontology enrichment from unstructured text. *arXiv e-prints*, pages arXiv–2102.

Otter, D. W., Medina, J. R., and Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*.

Petasis, G., Karkaletsis, V., Paliouras, G., Krithara, A., and Zavitsanos, E. (2011). Ontology population and enrichment: State of the art. In *Knowledge-driven multimedia information extraction and ontology evolution*, pages 134–166. Springer.

Poetsch, M., Corrêa, U., and Freitas, L. (2019). A word embedding analysis towards ontology enrichment. *Research in Computing Science*, 148:153–164.

Siami-Namini, S., Tavakoli, N., and Namin, A. S. (2019). The performance of lstm and bilstm in forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3285–3292. IEEE.

Smaili, F. Z., Gao, X., and Hoehndorf, R. (2018). Opa2vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. *CoRR*, abs/1804.10922.

Sokolova, M., Japkowicz, N., and Szpakowicz, S. (2006). Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, pages 1015–1021. Springer.

Velardi, P., Fabriani, P., and Missikoff, M. (2001). Using text processing techniques to automatically enrich a domain ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 270–284.

Xiang, Z., Zheng, J., Lin, Y., and He, Y. (2015). Ontorat: Automatic generation of new ontology terms, annotations, and axioms based on ontology design patterns. *Journal of Biomedical Semantics*, 6:4.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, NIPS'15, pages 802–810, Cambridge. MIT Press.