FUNDAÇÃO GETULIO VARGAS
ESCOLA DE MATEMÁTICA APLICADA

FREDSON SILVA DE SOUZA AGUIAR

IMPLEMENTAÇÃO DE UM *WORKFLOW* DE ATUALIZAÇÃO E EXPERIMENTOS UTILIZANDO *WORD EMBEDDING* E ANALOGIA PARA O ENRIQUECIMENTO DA OPENWORDNET-PT

Rio de Janeiro
2021

FREDSON SILVA DE SOUZA AGUIAR

IMPLEMENTAÇÃO DE UM *WORKFLOW* DE ATUALIZAÇÃO E EXPERIMENTOS UTILIZANDO *WORD EMBEDDING* E ANALOGIA PARA O ENRIQUECIMENTO DA OPENWORDNET-PT

Trabalho de conclusão de curso apresentada para a Escola de Matemática Aplicada (FGV/EMAp) como requisito para o grau de bacharel em Matemática Aplicada.

Área de estudo: processamento de linguagem natural e representação de conhecimento.

Orientador: Dr. Alexandre Rademaker

Rio de Janeiro

2021

Ficha catalográfica elaborada pela BMHS/FGV

Aguiar, Fredson Silva de Souza

Implementação de um *workflow* de atualização e experimentos utilizando *word embedding* e analogia para o enriquecimento da OpenWordnet-PT/ Fredson Silva de Souza Aguiar. – 2021.

34f.

Trabalho de Conclusão de Curso – Escola de Matemática Aplicada.

Advisor: Dr. Alexandre Rademaker.

Includes bibliography.

1. Wordnets 2. Processamento de linguagem 3. Representação de conhecimento I. Rademaker, Alexandre II. Escola de Matemática Aplicada III. Implementação de um *workflow* de atualização e experimentos utilizando *word embedding* e analogia para o enriquecimento da OpenWordnet-PT

FREDSON SILVA DE SOUZA AGUIAR

IMPLEMENTAÇÃO DE UM *WORKFLOW* DE ATUALIZAÇÃO E EXPERIMENTOS UTILIZANDO *WORD EMBEDDING* E ANALOGIA PARA O ENRIQUECIMENTO DA OPENWORDNET-PT

Trabalho de conclusão de curso apresentada para a Escola de Matemática Aplicada (FGV/EMAp) como requisito para o grau de bacharel em Matemática Aplicada.

Área de estudo: processamento de linguagem natural e representação de conhecimento.

E aprovado em 08/12/2021
Pela comissão organizadora

Dr. Alexandre Rademaker
Escola de Matemática Aplicada - FGV/EMAp

Profa. Asla Medeiros e Sá
Escola de Matemática Aplicada - FGV/EMAp

Prof. Jefferson de Barros Santos
Escola Brasileira de Administração Pública e de Empresas - FGV/EBAPE

Dedico essa dissertação a todas que lutaram para que eu estivesse aqui.

Agradecimentos

Agradeço a Deus pela por iniciar e concluir essa jornada. A minha família, por ter me apoiado em todas as decisões, especialmente quando decidi me mudar para o outro lado do país para estudar; sem vocês eu não teria saído para descobrir o mundo. A Alana, pelo carinho e atenção eternamente inabaláveis. Ao CDMC, especialmente a Cássia, Luziel e Camacho, que primeiro me receberam de braços abertos, propiciando um mundo novo de oportunidades maravilhosas. Finalmente, agradeço ao meu orientador, Alexandre Rademaker, pela vastidão de desafios, ensinamentos e paciência.

"As fronteiras da minha linguagem são as fronteiras do meu universo"

- Ludwig Wittgenstein

Resumo

A OpenWordnet-PT é atualmente referência em wordnets para língua portuguesa. No entanto, à medida em que o recurso cresce, a manutenção puramente manual torna-se inviável. Dessa forma, discutimos atualização de infra-estrutura junto a algumas iniciativas em que OWN-PT é empregada, com ênfase no uso de representações vetoriais para o aprendizado de novas relações semânticas.

Palavras-chave: Wordnets. Processamento de Linguagem. Representação de Conhecimento.

Abstract

OpenWordnet-PT is reference in wordnets for the portuguese language. However, as the resource grows, purely manual maintenance becomes impractical. Thus, we discuss infrastructure update along with some initiatives in which OWN-PT is applied, with emphasis on the use of vector representations in learning new semantic relations.

Keywords: Wordnets. Language Processing. Knowledge Representation.

Lista de abreviaturas e siglas

KB	Knowledge Base
IRI	Internationalized Resource Identifier
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
LMF	Lexical Markup Framework
RDF	Resource Description Framework
PWN	Princeton Wordnet
OWN	Open Wordnets
OWN-PT	Open Wordnet for Portuguese
OWN-EN	Open Wordnet for English
PyOWN	Python for OpenWordnets

Sumário

1 Introdução

Wordnets são como dicionários que documentam sentidos de palavras, organizando-as em grupos de sinônimos chamados synsets. Tais grupos representam conceitos abstratos e são vinculados através de relações semânticas como hiperonímia ou antonímia. O termo foi cunhado inicialmente em Princeton, 1985, e atualmente é de grande importância na área de processamento de linguagem natural (MILLER et al., 1990).

Nesse contexto, a OpenWordnet-PT é atualmente principal referência em wordnets para língua portuguesa, e encontra-se em constante evolução (PAIVA; RADEMAKER; MELO, 2012; PAIVA; REAL et al., 2016). No entanto, na medida em que o recurso cresce, a manutenção puramente manual torna-se inviável. Ao mesmo tempo, existe ainda carência de esforços voltados ao desenvolvimento do cenário linguístico em torno da língua portuguesa, especialmente no que refere ao desenvolvimento e aplicação das wordnets no processamento de linguagem natural.

1.1 Motivação

Nos últimos anos, sistemas capazes de processar linguagem humana tem se tornado comuns, e com isso, a demanda por dados linguísticos abrangentes e de qualidade tem sido crescente. Tarefas como tradução, *question answering*, desambiguação e inferência textual podem, em muitos casos, aproveitar do uso de wordnets para tais finalidades. No entanto esse tipo de recurso é ainda pouco discutido no âmbito da língua portuguesa. Dessa forma, nos dedicamos à validação e publicação a OWN-PT e OWN-EN, bem como à revisão da importância destas para a área de estudo em torno do processamento de linguagem.

1.2 Objetivos

Diante disso, nossos objetivos incluem: validação dos dados (palavras e synsets) e estrutura (relações, redundâncias, incompletudes) presentes atualmente na OWN-PT; publicação do dado de modo compatível com bibliotecas acessíveis; implementação de *workflow* para revisão e publicação. Finalmente, exploramos um experimento realizado sobre o uso de vetorização de palavras para enriquecimento da OWN-PT.

De forma precisa, nas seções 2 e 3, apresentamos conceitos fundamentais. Na seção 2, introduzimos definições sobre wordnets através de uma abordagem ampla, focada em conceitos propostos originalmente por (MILLER et al., 1990). Já na seção 3, introduzimos conceitos específicos inerentes à OWN-PT, como representações em RDF, URIs e vocabulários.

Na seção 4, apresentamos iniciativas para a manutenção e publicação das OWNs. Como resultado, publicamos um *workflow* na forma do pacote PyOWN, responsável pelas etapas de reparação, atualização e preparação para publicação de OWN-PT e OWN-EN.

Finalmente, na seção 5, apresentamos experimentos em torno do uso de representações vetoriais (*word embeddings*) e analogia para a sugestão automática de novas relações semânticas para a OpenWordnet-PT. Tais iniciativas voltados ao desenvolvimento de wordnets, com ênfase na aplicação voltada para a língua portuguesa, culminaram com a publicação de um trabalho no LDK[1] 2021.

[1] veja: 2021.ldk-conf.org

2 Wordnets

O conceito de Wordnet foi inicialmente cunhado em Princeton, como um sistema de referência lexical, inspirado por estudos em psicolinguística relacionados a memória humana (MILLER et al., 1990); sendo a sua estrutura similar a um dicionário de conceitos abstratos, ou, mais precisamente, um tesauro indexando tais conceitos.

O grande diferencial das wordnets em relação a dicionários tradicionais é a consideração de aspectos organizacionais ou estruturais do léxico mental humano. Estas contemplam bases cognitivas e lexicais fundamentais no processo de raciocínio linguístico. Tais aspectos foram inicialmente estabelecidos através de associações de palavras e, posteriormente, empregando hipóteses provenientes da psicolinguística moderna.

Originalmente, quatro classes de conceitos foram propostas para ser indexadas: Substantivos (Noun); Verbos (Verb); Adjetivos (Adjective); e Advérbios (Adverb). Estas classes são organizadas de maneiras distintas através de relações semânticas baseadas em propriedades específicas. Ainda, as formas escritas das palavras empregam entre si relações lexicais.

2.1 Conceitos Básicos

O princípio fundamental ao qual devemos nos atentar é a distinção entre *forma* e *significado*. O termo *palavra* é comumente empregado para se referir tanto à forma escrita como ao sentido empregado. Diante disso, a seguinte distinção é proposta a fim de evitar ambiguidades: *word form*, ou *forma*, se refere a expressão ou forma escrita de uma palavra; *word meaning*, ou *significado*, se refere ao conceito abstrato a que uma *forma* se refere.

Naturalmente, o mapeamento entre formas e significados possui cardinalidade $N : N$, isto é, uma mesma forma pode estar associada a vários significados, assim como um único significado pode ser expresso através de várias formas sinonímicas. Este fato implica que tal mapeamento pode ser descrito como uma tabela ou matriz, chamada Matriz Lexical (Figura 1).

Vale ressaltar que, através desta representação, tornam-se evidentes dois problemas com os quais os falantes devem lidar na interlocução: a polissemia e sinonímia. Com efeito, um falante parte de um significado e deve decidir qual sinônimo a ser empregado. Por outro lado, o ouvinte recebe uma forma e deve desambiguar o significado pretendido.

Outro questionamento que surge diz respeito a representação dos próprios significados, um vez que mostra-se impraticável descrever computacionalmente os conceitos abstratos de forma construtiva. A alternativa imediata é escolher uma abordagem diferen-

Word Meanings	Word Forms				
	F_1	F_2	F_3	. . .	F_n
M_1	$E_{1,1}$	$E_{1,2}$			
M_2		$E_{2,2}$			
M_3			$E_{3,3}$		
⋮				. . .	
M_m					$E_{m,n}$

Figura 1 – Conceito de Matriz Lexical. Fonte: (MILLER et al., 1990)

cial, em que os conceitos são representados por um família arbitrária de símbolos que os determinem: com frequência, o conjunto de sinônimos usados para se referir ao significado basta para identificá-lo. Dessa forma, usamos *synsets*, do inglês *synonym sets* ou conjunto de sinônimos, para designar os significados a que as formas se referem. Nos casos em que os sinônimos são insuficientes, significados podem ser desambiguados através de glosas.

Atualmente, iniciativas independentes de (MILLER et al., 1990) preveem a descrição de *synsets* incluindo informações de glosas, exemplos e definições. Ainda, introduz-se o conceito de *word sense* ou *sense*, que denomina o aspecto de sentido em que uma forma pode ser empregada. De maneira relaxada, *sense* pode ser entendido como a relação entre forma e *synset* (GOODMAN, 2021; GLOBAL WORDNET ASSOCIATION, 2021).

2.2 Relações em Wordnets

Wordnets são organizadas em relações semânticas e lexicais, que desempenham papéis fundamentais, definindo estruturas de classes e, essencialmente, introduzindo conhecimento através da coesão entre formas e conceitos. Efetivamente, estas estruturas têm sido comumente empregadas como bases de conhecimento.

Quanto a relações semânticas, em se tratando de mapeamentos entre conceitos, é natural apresentar estas como relações entre *synsets*. Já relações lexicais ocorrem como mapeamentos entre formas, como são a sinonímia e antonímia. Abaixo, descrevemos brevemente algumas destas relações:

- Sinonímia - define que dois termos são intercambiáveis sem alterar o valor-verdade da expressão. Essa, no entanto, é uma definição restritiva. Uma definição enfraquecida é que a substituição pode ser feita restrita a certos contextos. É uma relação simétrica transitiva.

- Hiponímia - define uma relação entre *synsets* e pode ser descrita sob a regra segundo a qual *x é hipônimo de y* quando *um x é um (tipo de) y*, assim como *cão é um animal* ou *macieira é um tipo de árvore*. Tal relação constrói uma hierarquia, em

que um hipônimo herda todas as características de seus hiperônimos. É uma relação assimétrica transitiva

- Meronímia - é uma relação semântica que descreve a relação parte-todo através da regra segundo a qual *x é merônimo de y* quando *x é parte de y*. Esta é uma relação assimétrica transitiva.

Note que mapeamentos como os acima permitem responder perguntas de forma mais precisa introduzindo informações do mundo real, ou senso comum, que não ocorrem em dicionários usuais. Vale ressaltar que a lista acima não esgota as relações presentes em wordnets, apenas representa o tipo de informação disponível.

Além de relações semânticas e lexicais, é possível destacar ainda a importância da morfologia na interlocução. Um falante deve se referir a forma *animais* tendo por conhecido que está é uma flexão do substantivo *animal*. Isso nos leva ao problema de lematização, que em muitos casos pode ser resolvido através de heurísticas, ou dados publicados, como (FIGUEIREDO DE ALENCAR; CUCONATO; RADEMAKER, 2018).

2.3 Estruturas Lexicais

Através do experimento de associação de palavras, notou-se que as respostas para certas classes de palavras era pertencente à mesma classe, com significância: substantivos 79%; adjetivos 65%; verbos 43%; e advérbios 43%. Diante disso, além de relações semânticas, (MILLER et al., 1990) propõe a divisão do léxico em categorias, *parts-of-speech* ou *pos*: Substantivos (Noun); Verbos (Verb); Adjetivos (Adjective); e Advérbios (Adverb). No entanto, tentar impôr uma única estrutura a cada uma das classes seria pouco preciso, e representaria mal as classes.

Dessa forma, cada classe é internamente definida seguindo uma estrutura lexical única; substantivos são organizados através de hierarquias de tópico; verbos são organizados com base em relações de implicação, por exemplo. A imposição dessas diferentes categorias introduz certa redundância no modelo, mas apresenta a vantagem de diferentes categorias sintáticas serem mais facilmente diferenciadas, e podem ser sistematicamente exploradas.

3 OpenWordnets

A OpenWordnet-PT é atualmente a principal referência em wordnets para língua portuguesa. Com início em 2012, a iniciativa propõe ser disponibilizada de maneira aberta, e encontra aplicações na BabelNet[1], Freeling[2] e Google Translate[3], por exemplo. Dessa forma, nota-se a relevância desta, especialmente para o cenário em torno da língua portuguesa. Tal iniciativa publica ainda a OpenWordnet-EN, que disponibiliza o mesmo tipo de dado para a língua inglesa. Ambas versões são originalmente baseados na Wordnet de Princeton (PAIVA; RADEMAKER; MELO, 2012; PAIVA; REAL et al., 2016). Frequentemente mencionaremos OpenWordnets para nos referir a ambos publicações realizadas pela organização.

Figura 2 – Logo da OpenWordnet-PT

Aqui, destacamos a importância da discussão em torno desse tipo de dado, quanto ao seu desenvolvimento e aplicações. De fato, a manutenção de dados linguísticos de qualidade como este tem impactos práticos em motores de busca, tradutores automáticos, atendimento a perguntas e problemas de inferência textual, por exemplo.

Nas próximas seções, descrevemos a arquitetura segundo a qual as OWNs são construídas e publicadas. Posteriormente, abordamos os esforços realizados no sentido da sua validação, correção e publicação.

3.1 Resource Description Framework

RDF é uma família de linguagens formais da W3C projetada para interoperabilidade de descrições de dados na web. Tal modelo se baseia em predicações sobre recursos na forma de triplas *sujeito-predicado-objeto*, em que *sujeito* é um recurso, e *predicado* expressa alguma afirmativa sobre este, estabelecendo uma relação entre *sujeito* e *objeto* (Figura 3.1). Serviços capazes de indexar e operar sobre tais triplas são chamados *triple stores* (KOUYLEKOV; OEPEN, 2014).

[1] veja: babelnet.org
[2] veja: nlp.lsi.upc.edu/freeling
[3] veja: translate.google.com/intl/en/about/license

RDF estende a estrutura de relações da web para usar URIs como representantes de instâncias na web, bem como as relações entre estas, usualmente referidas como triplas. Tal estrutura descreve um grafo direcionado e etiquetado, em que recursos representam nós, e predicados representam arestas (WORLD WIDE WEB CONSORTIUM, 2014, 2021a).

Aqui, entenda-se URIs e IRIs como identificadores de recursos na web. Idealmente tais URIs são universalmente únicos, a fim de identificar uma entidade. Naturalmente, a utilização de padrões de URI e IRI facilita a mesclagem de diferentes ferramentas bem como o compartilhamento entre diferentes plataformas (GREGORIO et al., 2012; BERNERS-LEE; FIELDING; MASINTER, 2005).

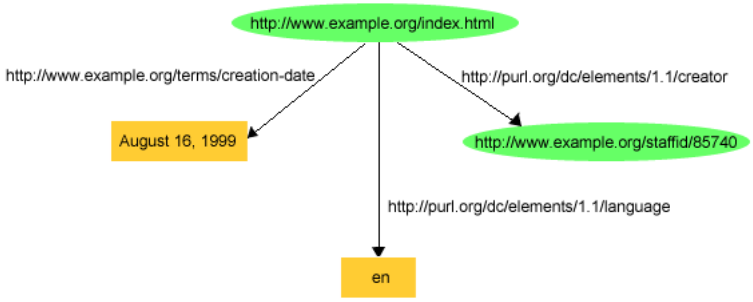

Figura 3 – Representação de dados em RDF

Tais propriedades tornam RDF uma opção vantajosa para representação de wordnets em RDF. Efetivamente, atualmente, existem diversas conversões de wordnets para este formato, muitas dessas baseadas em (WORLD WIDE WEB CONSORTIUM, 2021b), incluindo a própria iniciativa OWN-PT.

3.2 Classes Básicas e Predicados

RDF permite descrever conjuntos de classes, propriedades e restrições entre seus recursos e predicações; a tais conjuntos de regras chamamos vocabulários ou esquemas. Estes preveem um sistema de tipos e classes de maneira similar às linguagens orientadas a objeto, isto é, definem categorias que compartilham características em comum.

O vocabulário[4] de OWN-PT prevê três classes fundamentais: Synset, WordSense e Word (forma), incluindo subclasses para cada estrutura lexical na seção 2.3, como NounSynset ou AdjectiveWordSense. As instancias dessas classes devem implementar certas propriedades que descrevem tal entidade através de predicações. De maneira similar, podemos ainda definir relações semânticas e lexicais entre os recursos através de predicados.

Na tabela 1, listamos alguns predicados definidos no vocabulário que descrevem ou instanciam Synsets, WordSenses e Words. O predicado *wn30:synsetId*, por exemplo, descreve a relação entre um Synset e seu identificador, do tipo Literal. De maneira similar,

[4] veja: `github.com/own-pt/openWordnet-PT/blob/master/own.ttl`

o predicado *wn30:lemma* relaciona uma Word a sua forma escrita, ou lema, do tipo Literal. Nesse contexto, tais predicações são usadas para atribuir propriedades e definir relações entre recursos. A exemplo, no apêndice .1 definimos um Synset adjetivo satélite que expressa o conceito que nos referimos através da forma "moribundo", e que se relaciona a outro conceito por similaridade.

Predicado	**Domínio**	**Imagem**
wn30:synsetId	wn30:Synset	rdfs:Literal
wn30:gloss	wn30:Synset	rdfs:Literal
wn30:example	wn30:Synset	rdfs:Literal
wn30:containsWordSense	wn30:Synset	wn30:WordSense
rdfs:label	wn30:WordSense	rdfs:Literal
wn30:word	wn30:WordSense	wn30:Word
wn30:pos	wn30:Word	rdfs:Literal
wn30:lemma	wn30:Word	rdfs:Literal

Tabela 1 – Algumas propriedades definindo Synsets, WordSenses e Words

Nesse ponto, vale ressaltar algumas redundâncias no modelo: WordSenses possuem *label* repetindo a forma da Word a que está ligado; e Synsets possuem seus IDs repetidos na própria URI, por exemplo. Tais redundâncias diminuem as chances de perda de informação, e foram fundamentais na reparação de algumas falhas encontradas no dado através de inspeção. Veremos a seguir que a estrutura definida desta maneira define uma base de conhecimento, e que encontra diversas aplicações.

3.3 Bases de Conhecimento

Representação de Conhecimento é o campo da inteligência artificial dedicado a representar informações sobre o mundo de uma forma que um sistema computacional possa utilizar para resolver tarefas complexas, como manter diálogo com um ser humano. A uma conjunto de representações desse tipo chamamos Bases de Conhecimento, do inglês Knowledge Bases (RONALD BRACHMAN, 2004). Exemplos de KBs incluem (WIKIPEDIA, 2021; DBPEDIA ASSOCIATION, 2021; PEASE, 2021).

Assim como demais wordnets, as OWNs formam bases de conhecimento especialmente relevantes, pois representam a estrutura do léxico mental humano. Esse tipo de estrutura encontra aplicações em uma gama de problemas linguísticos, incluindo os problemas de inferência textual e raciocínio, como (PAIVA; RADEMAKER; MELO, 2012; LIEN; KOUYLEKOV, 2014, 2015; KALOULI; BUIS et al., 2019).

Diante disso, destacamos a seguir o trabalho em (KALOULI; REAL; PAIVA, 2018), em que mostramos que a estrutura de semântica da PWN basta para resolver uma classe de problemas de inferência textual simples.

3.3.1 Inferência Textual Simples

O Problema de Inferência Textual, pode ser descrito como um problema de dedutibilidade: um texto tomado por premissa permite deduzir um texto tomado por conclusão do ponto de vista das interpretações semânticas? Isto é, dado um par de textos (T_1, T_2), estamos interessados em avaliar se: T_1 implica T_2; T_1 contradiz T_2; ou T_1 independe de T_2. Observamos que para um processo de raciocínio sobre a relação de inferência é fundamental o acesso a conhecimentos que estão além das próprias sentenças.

Para um experimento de inferência simples, (KALOULI; REAL; PAIVA, 2018) considera pares de sentenças diferindo por zero ou uma palavra, então propõe uma série de heurísticas baseadas em relações semânticas extraídas da PWN (sinonímia, hiperonímia, hiponímia e antonímia) a fim de calcular a relação entre os pares. Os resultados do experimento mostram que o método é confiável e apresenta uma acurácia de quase 100% nos casos diferindo por uma única palavra.

Em (KALOULI; BUIS et al., 2019), como em outros, discute-se a performance de sistemas capazes de detectar inferência como uma boa medida para compreensão verdadeira da linguagem humana. Conclui-se que a incorporação de conhecimento através de wordnets é verdadeiramente vantajosa.

4 Py-OpenWordnet

A OpenWordnet-PT é originalmente publicada em um repositório[1] público da organização OWN-PT, e podia ser mantida através de uma aplicação[2] online. Através da aplicação, os mantenedores deveriam ser capazes de propôr e votar sugestões de alteração sobre os Synsets, como a inclusão ou remoção de glosas, exemplos e formas, que poderiam ser aplicadas a cada ciclo de atualização caso a quantidade dos votos fosse favorável.

Atualmente, a atualização com base nos votos não tem sido possível devido a problemas internos. Porém, ainda é possível descarregar os relatórios incluindo estado da wordnet, sugestões e votos na forma de arquivos brutos. Tais arquivos de *dump* serviriam como entrada para um novo fluxo de atualização, que substituiria a aplicação original quanto a atualização e publicação.

No entanto, um fato que tornou-se conhecido através de inspeção e atribuições no repositório era o acúmulo de erros possivelmente devidos ao mal funcionamento de servidores, casos de uso imprevistos, ou anotações inadequadas ao longo dos anos. Os dados mostraram-se inválidos com relação ao vocabulário de OWN-PT: instâncias mal tipadas; objetos de relações indefinidos; e URIs inválidas, são exemplos de casos que exigiam atenção. Dessa forma, destacou-se ainda a necessidade de reparação das OWNs anterior a sua atualização e publicação.

Como parte do aparato necessário para a reparação e publicação, implementamos o pacote Python for OpenWordnets. PyOWN[3] é um pacote python que contém uma série de funcionalidades para correção, atualização e publicação das OWNs. Junto ao pacote, publicamos uma aplicação *shell* responsável pelo processo de *release* do dado. No apêndice .2, detalhamos a estrutura e referência em que o pacote foi baseado.

O core da funcionalidade foi desenvolvido em python visando acessibilidade, fácil integração com interfaces de linha de comando, e pela disponibilidade de pacotes já implementados. Destacamos as bibliotecas RDFLib[4], para manipulação de RDF, e LXML[5], para manipulação de XML.

4.1 Manutenção e Atualização

O desenvolvimento de PyOWN foi originalmente motivado pela necessidade de publicação da OWN, o que exigiria a sua atualização com relação a sugestões manualmente

[1] veja: github.com/own-pt/openWordnet-PT
[2] veja: openwordnet-pt.org
[3] veja: github.com/own-pt/py-ownpt
[4] veja: pypi.org/project/rdflib
[5] veja: pypi.org/project/lxml

anotadas pelos autores. No entanto, durante inspeção, notou-se a necessidade de manutenção. Dessa forma, a implementação de PyOWN seguiu-se em duas etapas: reparação e atualização.

4.1.1 Etapa de Reparação

Alguns demandas a serem revisadas eram conhecidas previamente e já haviam sido documentadas no repositório da OWN-PT, enquanto outras foram levantadas durante o processo, de maneira iterativa: novos erros eram levantados à medida que antigos eram corrigidos. Dessa forma, implementamos a ações de reparação sob demanda, de modo a serem aplicadas sequencialmente (Tabela 2).

Funcionalidade	**Descrição**
add_word_types	grant well typed Words
remove_blank_words	grant well formed URIs for Words
remove_void_words	remove Words without lemma
remove_double_words	remove Words with multiple lemmas
add_sense_types	grant well typed Senses
replace_blank_senses	grant well formed URIs for Senses
expand_sense_words	recover lemma by Senses label
add_sense_labels	recover label by Word lemma
add_sense_number	add sense number
format_lexicals	grant well defined lemmas
replace_word_uris	grant unique URIs for Words
replace_sense_labels	match labels to lemmas
remove_word_duplicates	unify Words with same lemma
remove_sense_duplicates	unify Senses with same label
remove_desconex_sense_nodes	remove desconex Senses
remove_desconex_word_nodes	remove desconex Words

Tabela 2 – Algumas das ações de reparação implementadas por PyOWN

Algumas ações não listadas são integrados a rotinas fundamentais: lemmas, glosas e exemplos equivalentes a menos de espaçamentos não podem ser definidas novamente; as URIs geradas são validadas e para evitar colapso de identificadores. Além disso, ações demasiado específicas não foram listadas acima, como a atualização de nomes de propriedades, ou reformulação de identificadores. Atualmente, acreditamos que grande parte dessas ações não serão reutilizadas, mas continuam implementadas para documentar as alterações realizadas.

4.1.2 Etapa de Atualização

As etapas de atualização atualmente supõem as wordnets já corrigido de acordo com as ações descritas acima. Assim, a atualização ocorre também em dois passos: aplicação de sugestões dadas anotações manuais e regras; e reparação do dado atualizado.

As sugestões são aplicadas seguindo regras específicas. Por exemplo, exige-se uma pontuação mínima para uma sugestão feita ser acatada, diferenciando usuários tipo sênior e júnior. Além disso, aspectos como ordem de aplicação devem ser considerados. A reparação subsequente à atualização é uma redundância visando evitar os erros mais comuns observados anteriormente, dando preferência a segurança sobre performance da aplicação.

Novamente, algumas funcionalidades fundamentais são implementadas como parte do core da aplicação, garantindo URIs bem formadas, exemplos e glosas formatados, e impedem a declaração de recursos e relações previamente existentes, por exemplo. Demais detalhamentos sobre implementação não serão apresentados, visto que não fazem parte do escopo desejado para esta apresentação.

4.2 Publicação e Compatibilidade

Finalmente, subsequente à manutenção e atualização, está a publicação das OWNs, incluindo a documentação do estado atual, estatísticas e levantamentos sobre compatibilidade e acessibilidade do formato publicado. Partindo da versão 1.0.0, próximas publicações periódicas devem seguir os seguintes critérios:

- MAJOR: alterações em larga escala de Synsets, esquemas ou formato de arquivos;
- MINOR: revisões periódicas em um número limitado de Synsets e Words;
- PATCH: correções de typos, códigos ou outras alterações de menor escala.

A primeira publicação oficial inclui trabalhos realizados desde 2012, e próximas publicações seguirão o formato *MAJOR.MINOR.PATCH* (Tabela 3).

Alguns usuários podem ter preferência por fazer uso da wordnet em formato RDF através de *triples stores*, enquanto outros podem preferir lançar mão de bibliotecas para manipulação. Dessa forma, foi decidida a publicação nos seguintes formatos: Turle, formato para serialização de RDF, em partições logicamente bem definidas; e LMF, formato compatível com a biblioteca WN.

Wordnet	**Lang**	**Words**	**Senses**	**Synsets**
OWN-PT	pt	59211	83762	52670
OWN-EN	en	156584	206978	117659

Tabela 3 – Algumas estatísticas OWNs 1.0.0

4.2.1 Partição Lógica de Arquivos

A OWN-PT foi publicada em formato Turtle, para serialização compacta de RDF, amigável para a leitura humana. O grafo RDF foi particionado em arquivos individuais, cada um dos quais contém definições sobre certos aspectos das OWNs:

- synsets: definições de Synsets, incluindo glosas e exemplos;
- wordsenses: definições de WordSenses e pertencimento a Synsets;
- words: definições de Words e pertencimento a WordSenses;
- relations: relações originais da PWN;
- morpholinks: demais relações morfossemânticas.
- same-as: mapeamento de Synsets equivalentes entre OWN-PT e OWN-EN.

Tal particionamento permite acessar apenas as informações necessárias em aquivos menores, além de facilitar o acompanhamento de alterações no dado ao longo do tempo.

4.2.2 Lexical Markup Framework

LMF é um formato de publicação de wordnets proposto pela Global WordNet Association, e compatível com a biblioteca WN, um pacote python que provê uma interface para wordnets, incluindo pesquisas no grafo, métricas de similaridade, desanbiguação de Senses, e modelagem precisa de Synsets, Senses e Words. Tal pacote se assemelha ao submódulo para wordnets de NLTK[6].

Acreditamos que a publicação de OWN-PT com compatibilidade para uso através da WN possa atrair novos usuários ao simplificar o acesso ao dado e disponibilizar ferramental desenvolvido pela própria comunidade internacional, com maior liberdade para diferentes linguagens (Figura 4).

```
$ python3 -m wn download --index index.toml own
Download [##############################] (14925293/14925293 bytes) Complete
Added own-en:1.0.0 (OpenWordnet-EN)
Added own-pt:1.0.0 (OpenWordnet-PT)
```

Figura 4 – Baixando OWNs através da biblioteca WN

[6] veja: nltk.org/ e nltk.org/howto/wordnet.html

5 Enriquecimento por Analogia

Neste capítulo, descrevemos experimentos realizados no uso de vetorização de palavras e aprendizado de máquinas na obtenção de novas sugestões aplicadas ao enriquecimento da OWN-PT. Focamos no uso de analogia para aprendizado de novas relações semânticas através de vetores de deslocamento voltado à língua portuguesa (GONÇALO OLIVEIRA; AGUIAR; RADEMAKER, 2021).

A seguir, apresentamos fundamentos dos métodos de vetorização e analogia, seguindo com detalhamentos e resultados dos experimentos realizados. Alegamos que o uso de vetorização e aprendizado computacional em conjunto com métodos baseados em anotação manual tem grande potencial não apenas no enriquecimento de wordnets, como em demais problemas linguísticos incluindo o problema de inferência textual.

5.1 Vetorização e Analogias

Quando lidamos com representação de conhecimento, destacam-se duas abordagens principais: bases de conhecimento baseadas em anotação manual, e métodos distribucionais, como vetorização de palavras (word embeddings). Embora não fortemente formalizados, as formas vetorizadas podem ser obtidas automaticamente a partir de grandes corpora de texto; verdadeiramente, algumas iniciativas se dedicam a publicação desse tipo de dado como (GLOBAL WORDNET ASSOCIATION, 2021; HARTMANN et al., 2017).

Idealmente, *embeddings* são capazes de capturar certas semânticas do espaço representado, e, pela regularidade das linguagens naturais, estes se mostram capazes de aprender certas relações semânticas. Como mostra (DROZD; GLADKOVA; MATSUOKA, 2016), algumas relações semânticas podem ser capturadas por métodos simples, como através dos vetores deslocamento. Dessa forma, consideramos os *offsets* entre formas vetorizadas e analogia para obter sugestões automáticas de novas relações.

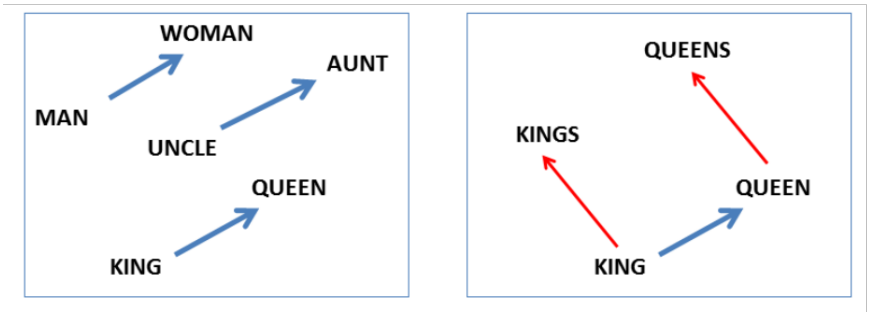

Figura 5 – *Offsets* capturando relações semânticas (MIKOLOV; YIH; ZWEIG, 2013)

A analogia consiste em responder *o que está para b como a∗ está para a*. No

exemplo da figura 5, *Rainhas está para Rainha como Reis está para Rei*, dessa forma o vetor deslocamento evidencia a relação *singular-plural*. Através de vários exemplos, devemos ser capazes de aprender relações dessa forma. Portanto, nosso objetivo seria aprender tais analogias através de uma quantidade de exemplos retirados da OWN-PT.

O método mais comum para cálculo de analogias é o 3CosAdd, embora métodos mais sofisticados demonstrem resultados melhores, como o 3CosAvg e LRCos:

$$\textbf{3CosAdd: } b* = \underset{w \in V}{\operatorname{argmax}} \cos(w, b + a * -a) \tag{5.1}$$

$$\textbf{3CosAvg: } b* = \underset{w \in V}{\operatorname{argmax}} \cos(w, b + f_{avg}) \tag{5.2}$$

$$\textbf{LRCos: } b* = \underset{w \in V}{\operatorname{argmax}} P(w \in C^*) \cos(w, b) \tag{5.3}$$

em que f_{avg} é o vetor deslocamento médio, e C^* a *classe objetivo*. Nos experimentos, demos preferência aos métodos 5.2 e 5.3.

O método LRCos introduz o conceito de classes de pertencimento: dado um conjunto pares de analogias, consideramos as formas pertencentes ao domínio da relação avaliada como instancias da *classe fonte*, e formas da imagem como instancias da *classe objetivo*. Aprendemos a relação de pertencimento $w \in C^*$ através de regressão logística: palavras-objetivo são tomadas como um exemplos de resultados *positivos* para a regressão, enquanto as palavras-fonte junto a amostras aleatórias do dicionário são exemplos *negativos*. Tal método se beneficia ao garantir similaridade com a classe objetivo, o que evita a captura de características irrelevantes (DROZD; GLADKOVA; MATSUOKA, 2016).

5.2 Experimentos e Discussões

A fim de analisar a performance dos diferentes métodos acima, optamos pela utilização de Vecto[1], que implementa 3CosAvg e LRCos, e suporta testes de analogia no formato BATS[2]. Os testes foram retirados da OWN-PT, e tomando vetores de dimensão 300 de NILC[3]. Consideramos 12 relações (Tabela 5) que acreditamos poder ser mais facilmente aprendidas, e que possuíssem uma quantidade considerável de instâncias para serem treinadas. Finalmente, geramos um teste de analogia para cada relação.

Os testes de analogia são organizados em arquivos de texto de várias linhas consistindo de duas colunas. Cada linha descreve uma pergunta, uma palavra-pergunta na primeira coluna, e uma lista de possíveis respostas para a relação a ser aprendida na segunda (Tabela 4). Para a criação dos testes, cada instância das relações entre Synsets é expandida no produto cartesiano entre as possíveis formas, então agrupamos as relações entre formas pelas palavras-pergunta na primeira coluna, e listamos as formas relacionadas

[1] veja: `vecto.space`
[2] veja: `vecto.space/projects/BATS`
[3] veja: `nilc.icmc.usp.br`

Source	**Target**
atividade	ativo/agencioso/inativo
bem-estar	doentio/adoentado/insalubre/salubre/doente/são/saudável
bondade	boa/bom
pálpebra	celha/conjuntiva/pestana/cílio
pássaro	plumagem/ala/bico/pluma/asa/pena/culatra/fúrcula/garupa/...
pé	calcanhar/alfândega/artelho/polegada/dedo/calcâneo/hálux/sola

Tabela 4 – Excertos de datasets para as relações: attribute; e partMeronymOf

na segunda. Para este trabalho, decidimos considerar apenas as linhas em que as palavras-pergunta ocorrem em Synsets do tipo Core, que descrevem conceitos mais frequentemente usados. Como resultado, obtivemos quantidades variadas de questões para cada relação (Tabela 5).

Relation	**Questions**	**SimToB**	**3CosAvg**	**LRCos**
agent	75	1.3%	1.3%	29.3%
antonymOf	68	19.1%	19.1%	7.4%
attribute	88	2.3%	9.1%	21.6%
byMeansOf	41	14.6%	26.8%	46.3%
causes	60	5.0%	6.7%	8.3%
entails	123	5.7%	5.7%	6.5%
memberHolonymOf	157	5.1%	5.1%	4.5%
memberMeronymOf	77	10.4%	7.8%	3.9%
partHolonymOf	417	2.2%	3.1%	7.2%
partMeronymOf	569	1.2%	1.2%	3.0%
substanceHolonymOf	33	6.1%	6.1%	0.0%
substanceMeronymOf	84	1.2%	2.4%	7.1%

Tabela 5 – Acurácia de testes de analogia para diferentes relações

A acurácia do método é calculada treinando-se o modelo em todas as linhas do conjunto de dados completo, exceto uma, então tentamos prever uma das relações entre as colunas fonte e objetivo (Tabela 4), isto é, data a palavra na coluna fonte, desejamos obter como resposta uma das palavras na coluna objetivo. Ao final, Vecto calcula a acurácia média repetindo-se o processo acima para cada linha da entrada.

Como resultados, obtivemos melhor performance através de LRCos. No entanto, a acurácia do método foi ainda baixa; apenas três relações (agent, attribute, byMeansOf) obtendo resultados acima de 20% e nenhuma acima de 50%. Por outro lado, deve-se considerar que a OWN-PT é uma iniciativa ainda em evolução, e certamente incompleta, o que pode refletir tais resultados.

Além disso, apenas a acurácia não basta para medir a utilidade do experimento. De fato, embora as sugestões não sejam 100% corretas no dados, ainda é possível avaliar as sugestões de relações dadas por Vecto para cada relação. Através de anotação manual, observamos que boa parte das sugestões obtidas servem ainda como referência para novas

relações na OWN-PT, isto é, tal abordagem serve como método para sugestão automática de novas relações.

Relation	**Examples**	**Already In**	**Invalid**	**Correct**	**Other**
antonymOf	90	0 (0%)	39 (43%)	3 (3%)	26 (29%)
attribute	94	3 (3%)	35 (37%)	27 (29%)	23 (24%)
causes	95	2 (2%)	39 (41%)	10 (11%)	4 (4%)
partHolonym	97	6 (6%)	22 (23%)	26 (27%)	17 (18%)

Tabela 6 – Anotação manual de 376 pares de palavras entre 4 relações

Os resultados na tabela 6 foram anotados pelo especialista mantenedor da OWN-PT para 4 relações de exemplo. As anotações incluem, além das estatísticas de sugestões validas e inválidas, as informações sobre relações já existentes na OWN-PT (coluna **Already In**), e sugestões que poderiam servir para outra relação (coluna **Other**). Dessa forma, não podemos aplicar sugestões do método diretamente, mas este agiliza a anotação por parte dos especialistas.

Tal discussão rendeu a participação na publicação de (GONÇALO OLIVEIRA; AGUIAR; RADEMAKER, 2021) na conferência internacional LDK 2021, sobre dados linguísticos no contexto das ciências de dados e aplicações baseadas em conhecimento.

6 Conclusões

A OpenWordnet-PT é uma iniciativa de grande importância para as comunidades falantes das línguas portuguesa e inglesa, com diversas aplicações. Os primeiros trabalhos em torno desse tipo de dado remontam a década de 1980 e, no entanto, ainda hoje, esta é uma comunidade crescente.

Em nossas atividades, pudemos trabalhar em torno de um dado de grande relevância para os cenários das línguas portuguesa e inglesa, além ser introduzidos à interação junto comunidade internacional em torno das wordnets, através de discussões e questionamentos. Como resultado, fomos capazes de implementar um *workflow* para a reparação, atualização e publicação das OpenWordnets para português e inglês, através de PyOWN. Atualmente, por meio de Python for OpenWordnets, as OWN-PT e OWN-EN publicam sua primeira *release* oficial no repositório da organização OpenWordnet-PT.

Existe ainda espaço para discussão e trabalhos futuros. OpenWordnet-PT carece de uma aplicação online mais completa e robusta que a atual, incluindo ferramental para navegação, visualização e análises. Há nicho para indagação sobre a estrutura topológica definida pelas wordnets e como essas podem ser visualizadas. A avaliação aprofundada dos grafos que buscam descrever o léxico mental humano pode ter resultados interessantes, como a existência de conceitos desconexos, ou a caracterização puramente topológica de Synsets. Paralelamente, a explicação visual destes conceitos e como estes se relacionam poderia facilitar a compreensão da comunidade sobre as wordnets e sua importância.

Paralelamente, fomos capazes de explorar a estrutura semântica da OWN-PT associado ao uso de *word embeddings* para discutir a sugestão de novas relações com resultados positivos: embora o método não apresente acurácia perfeita, este ainda pode ser usado como ferramenta auxiliar iterativa. Tal investigação culminou na publicação de um trabalho em conjunto com Alexandre Rademaker, professor pela FGV EMAp e pesquisador pela IDM Brasil, e Hugo Gonçalo Oliveira, pesquisador pela Universidade de Coimbra, na conferência LDK, de importância no cenário internacional sobre ciências de dados e aplicações baseadas em conhecimento.

Investigações futuras sobre o trabalho em torno de *word embeddings* podem incluir o uso de técnicas robustas no aprendizado de relações igualmente complexas. Novamente, o estudo sobre características inerentes ao grafo que descreve o léxico mental humano nesse tipo de questionamento podem ser de grande valor. Analogamente, há ainda espaço para estudo na visualização das relações semânticas capturadas pelos métodos vetoriais, especialmente considerando que lidamos com espaços semânticos abstratos e vetores de alta dimensão.

Referências

BERNERS-LEE, Tim; FIELDING, R; MASINTER, L. **Uniform Resource Identifier (URI): Generic Syntax**. [S.l.: s.n.], 2005.

DBPEDIA ASSOCIATION. **Global and Unified Access to Knowledge Graphs**. Disponível em: <https://wiki.dbpedia.org/>. Acesso em: 24 nov. 2021.

DROZD, Aleksandr; GLADKOVA, Anna; MATSUOKA, Satoshi. Word Embeddings, Analogies, and Machine Learning: Beyond king - man + woman = queen. In: PROCEEDINGS the 26th International Conference on Computational Linguistics: Technical papers COLING 2016. [S.l.: s.n.], 2016. (COLING 2016), p. 3519–3530.

FIGUEIREDO DE ALENCAR, Leonel; CUCONATO, Bruno; RADEMAKER, Alexandre. MorphoBr: an open source large-coverage full-form lexicon for morphological analysis of Portuguese. **Texto Livre: Linguagem e Tecnologia**, v. 11, n. 3, p. 1–25, 2018. DOI: 10.17851/1983-3652.11.3.1-25. Disponível em: <https://periodicos.ufmg.br/index.php/textolivre/article/view/16809>.

GLOBAL WORDNET ASSOCIATION. **Global Wordnet Formats**. Disponível em: <globalwordnet.github.io/schemas>. Acesso em: 24 nov. 2021.

GONÇALO OLIVEIRA, Hugo; AGUIAR, Fredson Silva de Souza; RADEMAKER, Alexandre. On the Utility of Word Embeddings for Enriching OpenWordNet-PT. In_____. **3rd Conference on Language, Data and Knowledge (LDK 2021)**. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. v. 93. (Open Access Series in Informatics (OASIcs)), 21:1–21:13. ISBN 978-3-95977-199-3. DOI: 10.4230/OASIcs.LDK.2021.21. Disponível em: <https://drops.dagstuhl.de/opus/volltexte/2021/14557>.

GOODMAN, Michael Wayne. **Wn Documentation**. Disponível em: <wn.readthedocs.io/en/latest>. Acesso em: 24 nov. 2021.

GREGORIO, J et al. **URI Template**. [S.l.: s.n.], 2012.

HARTMANN, Nathan et al. **Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks**. [S.l.: s.n.], 2017. arXiv: 1708.06025 [cs.CL].

KALOULI, Aikaterini-Lida; BUIS, Annebeth et al. Explaining Simple Natural Language Inference. In: PROCEEDINGS of the 13th Linguistic Annotation Workshop. Florence, Italy: Association for Computational Linguistics, ago. 2019. P. 76–84.

KALOULI, Aikaterini-Lida; REAL, Livy; PAIVA, Valeria de. WordNet for "Easy" Textual Inferences. In: PROCEEDINGS of the Globalex Workshop, associated with LREC 2018. Miyazaki, Japan: [s.n.], 2018.

KOUYLEKOV, Milen; OEPEN, Stephan. RDF Triple Stores and a Custom SPARQL Front-End for Indexing and Searching (Very) Large Semantic Networks. **Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations**, p. 90–94, 2014.

LIEN, Elisabeth; KOUYLEKOV, Milen. Semantic Parsing for Textual Entailment. In: PROCEEDINGS of the 14th International Conference on Parsing Technologies. Bilbao, Spain: Association for Computational Linguistics, jul. 2015. P. 40–49. DOI: 10.18653/v1/W15-2205. Disponível em: <https://www.aclweb.org/anthology/W15-2205>.

_____. UIO-Lien: Entailment Recognition using Minimal Recursion Semantics. **Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)**, p. 99–703, 2014.

MIKOLOV, Tomas; YIH, Wen-tau; ZWEIG, Geoffrey. Linguistic Regularities in Continuous Space Word Representations. In: PROCEEDINGS of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Atlanta, Georgia: Association for Computational Linguistics, jun. 2013. P. 746–751. Disponível em: <https://aclanthology.org/N13-1090>.

MILLER, George A. et al. Introduction to WordNet: An On-line Lexical Database. Versão inglesa. **Journal of Lexicography**, Oxford University Press, v. 3, n. 4, p. 235–244, 1990. Disponível em: <ftp://ftp.cogsci.princeton.edu/pub/wordnet/5papers.pdf>.

PAIVA, Valeria de; RADEMAKER, Alexandre; MELO, Gerard de. OpenWordNet-PT: An Open Brazilian Wordnet for Reasoning. In: PROCEEDINGS of COLING 2012: Demonstration Papers. Mumbai, India: The COLING 2012 Organizing Committee, dez. 2012. P. 353–360. Published also as Techreport http://hdl.handle.net/10438/10274. Disponível em: <http://www.aclweb.org/anthology/C12-3044>.

PAIVA, Valeria de; REAL, Livy et al. An overview of Portuguese Wordnets. In: PROCEEDINGS of 8th Global WordNet Conference. Bucharest, Romania: [s.n.], 2016. (GWC'16), p. 74–81.

PEASE, Adam. **Suggested Upper Merged Ontology (SUMO)**. Disponível em: <adampease.org/OP>. Acesso em: 24 nov. 2021.

RONALD BRACHMAN, Hector Levesque. **Knowledge Representation and Reasoning**. 1. ed. [S.l.]: Morgan Kaufmann, 2004. (The Morgan Kaufmann Series in Artificial Intelligence). ISBN 1558609326.

WIKIPEDIA. **Wikidata**. Disponível em: <wikidata.org/wiki/Wikidata:Main_Page>. Acesso em: 24 nov. 2021.

WORLD WIDE WEB CONSORTIUM. **RDF**. Disponível em: <w3.org/RDF>. Acesso em: 24 nov. 2021.

WORLD WIDE WEB CONSORTIUM. **RDF Schema 1.1**. 2014. Disponível em: <w3.org/TR/rdf-schema>. Acesso em: 24 nov. 2021.

_____. **RDF/OWL Representation of WordNet**. Disponível em: <w3.org/TR/wordnet-rdf>. Acesso em: 24 nov. 2021.

Apêndices

.1 Exemplo de Definição RDF em Turtle

Apresentamos um exemplo de definição em RDF codificado em Turtle, um formato para serialização de grafos RDF compacto e legível para humanos. A linguagem permite definir prefixos para representar URIs longas através da concatenação *prefix:suffix*, e sequencias de predicações sobre um mesmo recurso separadas por ";".

Abaixo segue um exemplo em que definimos um Synset **synset-00004171-s** da subclasse AdjectiveSatelliteSynset possuindo exemplo e glosa, além de um WordSense **wordsense-00004171-s-1** para a Word **word-moribundo-a**. Assim, definimos o conceito a que nos referimos através da forma "moribundo". Tal Synset se relaciona por similaridade ao Synset **synset-00003939-a**.

```
@prefix own-pt: <https://w3id.org/own/own-pt/instances/> .
@prefix owns: <https://w3id.org/own/schema/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

# define synset "00004171-s"
own-pt:synset-00004171-s
    rdf:type owns:AdjectiveSatelliteSynset ;
    owns:example "um paciente moribundo"@pt ;
    owns:gloss "estar à beira da morte"@pt ;
    owns:synsetId "00004171-s" ;
    owns:containsWordSense own-pt:wordsense-00004171-s-1 .

# define WordSense for "moribundo"
own-pt:wordsense-00004171-s-1
    rdf:type owns:WordSense ;
    rdfs:label "moribundo"@pt ;
    owns:word own-pt:word-moribundo-a .

# define Word "moribundo"
own-pt:word-moribundo-a
    rdf:type owns:Word ;
    owns:lemma "moribundo"@pt ;
    owns:pos "a" .

# define relations Synset "00004171-s"
own-pt:synset-00004171-s owns:similarTo own-pt:synset-00003939-a .
own-pt:synset-00003939-a owns:similarTo own-pt:synset-00004171-s .
```

.2 Estrutura de Diretórios da PyOWN

A estrutura do pacote PyOWN segue em alguns aspectos a proposta de PyDelphin[1]. Abaixo, descrevemos a estrutura de diretórios, bem como alguns arquivos fundamentais. A biblioteca pode ser instalada localmente através de Pip[2].

- **pyown (dir)**
  - **cli (dir)**: contêiner de interfaces de linha de comando incluindo: conversão para LMF; separação de RDF em arquivos por categorias lógicas; gerador de estatísticas; e atualização.
  - own.py: descreve a classe básica OWN, em que definimos funções fundamentais: adição e remoção segura de Words, WordSenses e triplas; definição de prefixos globais; ponteiros; formatação de strings e URIs, etc. Demais classes devem estender OWN.
  - util.py: funcionalidades básicas mas que independem de OWN, como a normalização formatos de arquivos e unificação ações a serem aplicadas.
  - *.py: arquivos de código fonte, incluindo classes, funções e códigos auxiliares, exceto interfaces de linha de comando.
- LICENSE: atualmente licenciado sob a **MIT License**.
- README.md: descrição de objetivos, recursos e utilização.
- mapping.org: descreve os mapeamentos de relações para **LMF**.
- requirements: lista de pacotes python requeridos pelo **PyOWN**.
- setup.py: arquivo de configuração para instalação do pacote através de **Pip**.
- release.sh: *script* responsável por processo de publicação, incluindo atualização, estatísticas, arquivos LMF e separação do RDF em arquivos por categorias lógicas.

[1] veja: github.com/delph-in/pydelphin
[2] veja: pip.pypa.io/en/stable