Similaridade Semântica de Nomes de Produtos Alimentícios Utilizando Wordnets do Português

Leonardo S. G. de Lima [1], Eduardo C. Gonçalves[1]

[1] *Escola Nacional de Ciências Estatísticas (ENCE-IBGE), Rua André Cavalcanti 106, Rio de Janeiro, RJ, 202031-050, Brazil*

Abstract

Different modern applications allow the user to search for a name of a certain entity and get the desired result even though an alternative spelling is used instead of the exact one that is recorded in the database. Such alternative spellings may consist of abbreviations, synonyms, hypernyms etc. This paper evaluates a technique based on Portuguese wordnets to compute the similarity between food product names. Two different similarity levels are simultaneously taken into account: lexical and semantic. The technique was validated through an experiment that compared the performance of two different wordnets (OpenWordNet-PT and a lexical knowledge base about food products) in a dataset that contains 1,000 names of food products sold in supermarkets matched with product names from the IBGE's National System of Consumer Price Indexes database.

Keywords

Semantic similarity, wordnet, short text, approximate string matching

Resumo

Diferentes aplicações modernas permitem que o usuário realize buscas pelo nome de uma entidade e obtenha o resultado desejado mesmo se utilizar uma grafia alternativa que não corresponda exatamente a que está cadastrada na base de dados. Tais grafias alternativas podem ser resultado de abreviações, uso de sinônimos, hiperônimos etc. O presente artigo avalia uma técnica baseada no uso de wordnets do Português para determinar a similaridade entre nomes de produtos alimentícios. Dois diferentes níveis de similaridade são simultaneamente avaliados: léxico e semântico. A técnica foi validada através de um experimento que comparou o desempenho de duas diferentes wordnets (OpenWordNet-PT e uma base de conhecimento lexical sobre produtos alimentícios) em uma base de dados contendo nomes de 1.000 produtos alimentícios comercializados em supermercados casados com nomes de produtos da base de dados do Sistema Nacional de Índices de Preços ao Consumidor do IBGE.

Palavras-chave

Similaridade semântica, wordnet, texto curto, casamento aproximado de strings

1. Introdução

Muitas aplicações modernas permitem que os usuários realizem buscas pelo nome de uma entidade – como produto, local, empresa etc. – utilizando grafias alternativas que são similares, porém não idênticas ao nome real da entidade no banco de dados [1]. Tais grafias podem ser resultado de

Proceedings of the 15th Seminar on Ontology Research in Brazil (ONTOBRAS) and 6th Doctoral and Masters Consortium on Ontologies (WTDO), November 22-25, 2022
EMAIL: leosglima@gmail.com (L. S. G. de Lima); eduardo.correa@ibge.gov.br (E. C. Gonçalves)

abreviações, uso de sinônimos, hiperônimos ou hipônimos, entre outros motivos. Ainda com a grafia alternativa da entidade, tais aplicações conseguem obter o resultado desejado. Na literatura, este tipo de problema é conhecido como problema do casamento aproximado de strings [2] ou problema do casamento de nomes [3].

As técnicas básicas para realizar o casamento aproximado entre strings – como análise de caracteres em comum e proporção de termos iguais [2, 3] – possuem a vantagem de serem independentes de linguagem. No entanto, uma importante desvantagem é que elas não são capazes de analisar a similaridade das palavras em nível semântico, sendo por isso ineficazes para lidar com termos sinônimos (ex.: "batata-baroa" e "mandioquinha") ou termos que possuam relação hierárquica (ex.: "mandioquinha" e "batata-inglesa" são tipos de "batata").

Desta forma, o presente artigo avalia um algoritmo baseado no uso de wordnets para realizar o casamento semântico de nomes de produtos alimentícios. Mais precisamente, o objetivo é realizar o casamento semântico de nomes de produtos alimentícios vendidos em supermercados com os nomes utilizados pelo Sistema Nacional de Índices de Preços ao Consumidor (SNIPC) do IBGE. O SNIPC é o sistema responsável por produzir o Índice Nacional de Preços ao Consumidor Amplo (IPCA), que tem por objetivo medir a inflação de um conjunto de produtos e serviços comercializados no varejo, referentes a consumo pessoal das famílias brasileiras [4]. Entre outras aplicações, o algoritmo descrito neste trabalho pode ser utilizado por um sistema para coleta automática de preços baseado em web scraping [5].

Os experimentos foram realizados utilizando duas diferentes wordnets, a fim de comparar os resultados e verificar qual delas possui o melhor desempenho para o objetivo proposto. A primeira wordnet utilizada foi a OpenWordNet-PT [6], que é implementada na biblioteca NLTK [7] do Python. A segunda é uma base de conhecimento lexical sobre produtos alimentícios criada de forma semiautomática a partir de classificações de produtos utilizadas no IBGE [8]. A base de dados utilizada nos experimentos contém 1.000 pares de nomes de produtos casados corretamente com sua descrição no SNIPC. Os resultados apontam que o algoritmo semântico combinado com a base de conhecimento lexical sobre produtos alimentícios representa a mais promissora dentre as alternativas avaliadas.

O restante do artigo está dividido da seguinte forma. A Seção 2 apresenta o referencial teórico e trabalhos relacionados. A Seção 3 descreve a metodologia utilizada no trabalho. A Seção 4 reporta e comenta os resultados do experimento. Por fim, a Seção 5 contém as conclusões do estudo.

2. Referencial Teórico

2.1. Similaridade de Jaccard

Uma das mais simples medidas de similaridade entre strings é a similaridade de Jaccard [9]. Dadas duas strings *S1* e *S2*, teremos dois conjuntos de *tokens* gerados por essas duas strings, que são as palavras que as compõem. Sejam esses dois conjuntos *tok*(*S1*) e *tok*(*S2*). O cálculo considera a proporção de tokens em comum entre as duas strings, conforme apresentado na Equação (1).

$$SJ(S1, S2) = \frac{|\ tok(S1) \cap tok(S2)|}{|\ tok(S1) \cup tok(S2)|}, \quad (1)$$

A Tabela 1 apresenta três exemplos de casos práticos de cálculo da similaridade de Jaccard. O primeiro envolve duas strings com três tokens cada, todos iguais entre si, resultando no valor similaridade 1,00 (valor máximo). O exemplo mostra que a similaridade de Jaccard oferece a vantagem de não ser afetada em situações em que as palavras são as mesmas nos dois nomes, mas aparecem em ordem trocada. No segundo exemplo, ambas as strings possuem dois tokens sendo apenas um comum a elas, resultando em uma similaridade de valor 0,33 (veja ainda que, neste exemplo, "arroz cateto" e "arroz branco" são dois diferentes tipos de "arroz"). Já no terceiro exemplo, os dois nomes comparados são sinônimos, porém não possuem tokens em comum, o que resulta em um valor de similaridade igual a zero (valor mínimo).

Tabela 1
Exemplos da Similaridade de Jaccard

S1	*S2*	*SJ(S1, S2)*
arroz com feijão	feijão com arroz	1,00
arroz branco	arroz cateto	0,33
aipim	mandioca	0,00

Os dois últimos exemplos demonstram a principal desvantagem da medida de Jaccard: desconsiderar a questão semântica. Mesmo que dois nomes descrevam produtos muito próximos (como "arroz branco" e "arroz cateto") ou produtos que denotem uma mesma entidade (como é o caso de "aipim" e "mandioca"), a similaridade de Jaccard entre esses nomes terá valor baixo caso possuam poucos tokens em comum.

2.2. Wordnets

Para avaliar a similaridade semântica entre nomes, é necessário incorporar uma fonte externa de conhecimento ao algoritmo de comparação [10]. Um dos tipos de fontes externas mais comumente utilizadas são as wordnets [1, 6, 11, 12].

Wordnet é um modelo para representar e organizar bases de dados de itens lexicais [11]. Neste modelo, grupos de itens lexicais sinônimos são agrupados em conjuntos denominados *synsets*. Estes formam o bloco básico de construção de uma wordnet, podendo ser vistos como as possíveis lexicalizações para um determinado conceito de uma língua [12]. Os synsets deverão ser conectados de acordo com diferentes tipos de relação semântica que existam entre eles, tais como a relação de hiponímia (um synset é hipônimo de outro synset quando o primeiro é mais específico, denotando uma subclasse do segundo) e a relação de meronímia (o conceito representado por um synset é uma parte de outro), apenas para citar dois exemplos.

Desta forma, uma wordnet pode ser vista como um grafo em que os nós são os synsets (representando diferentes conceitos ou entidades) e as arestas são as relações semânticas entre eles. A Figura 1 mostra um exemplo de wordnet de produtos alimentícios. Nela, os synsets são conjuntos de termos sinônimos que denotam um determinado produto alimentício ou uma categoria de produtos, enquanto as arestas representam a relação semântica hiperônimo/hipônimos entre eles.

As wordnets podem ser utilizadas para resolver diferentes tipos de problemas práticos da área de processamento de linguagem natural [13], incluindo o cálculo da similaridade entre textos curtos. Na wordnet da Figura 1, é possível observar, por exemplo, que "aipim", "mandioca" e "macaxeira" são denominações distintas para um mesmo produto, uma vez que estão em um mesmo nó (mesmo synset). Sendo assim, basta atribuir o valor máximo de similaridade (valor 1,00) ao comparar estes nomes.

Para nomes que não façam parte de um mesmo synset, é possível explorar a topologia do grafo e determinar o quanto estes nomes são similares de acordo com a distância entre tais synsets [1]. A medida de Wu & Palmer [14], apresentada na Equação (2), calcula a similaridade entre dois nós através de uma fórmula que leva em consideração essa distância e a profundidade dos nós no grafo. Na fórmula, *d1* representa a distância do nó 1 ao ancestral em comum com o nó 2, *d2* representa a distância do nó 2 a esse ancestral em comum e *d3* representa a distância da raiz do grafo ao ancestral em comum.

$$WP(nó1, nó2) = \frac{2 \times d3}{(d1 + d2 + 2 \times d3)}, \quad (2)$$

Por exemplo, suponha que desejamos utilizar a wordnet da Figura 1 para calcular a similaridade entre os nomes "rúcula" (que faz parte do synset {rúcula}) e "aipo" (que faz parte do synset {aipo, salsão}). O ancestral comum aos synsets {rúcula} e {aipo, salsão} é o synset {verduras}, que está uma aresta distante de ambos os synsets e a duas arestas de distância da raiz do grafo. Sendo assim, tem-se que: WP("rúcula", "aipo") = WP({rúcula}, {aipo, salsão}) = (2 × 2) ÷ (1 + 1 + 2 × 2) = 0.67.

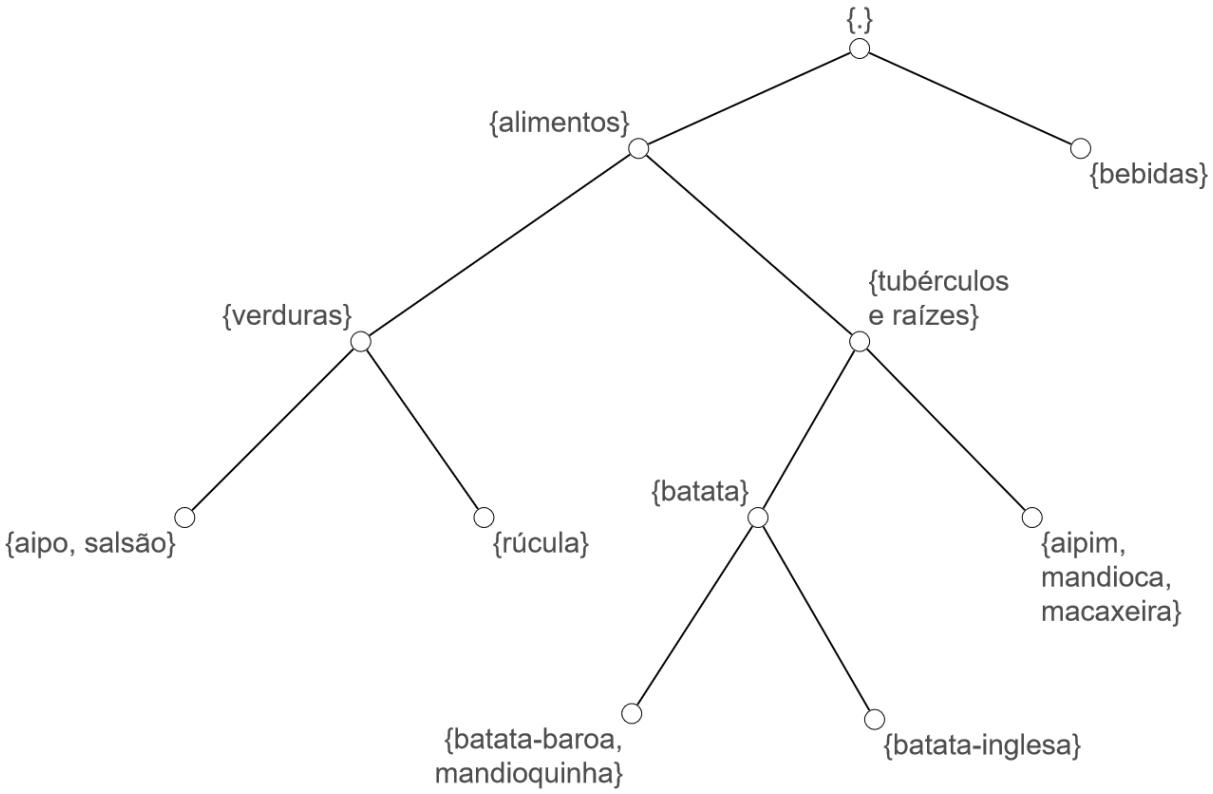

Figura 1: Exemplo de wordnet em que os synsets denotam categorias e nomes de produtos alimentícios e as arestas representam a relação semântica hiperônimo/hipônimo entre os synsets.

2.3. Similaridade Semântica

A similaridade de Jaccard trabalha no nível léxico [10], em que somente as palavras originalmente presentes nos nomes são levadas em consideração para determinar a similaridade. Porém, existe um nível ainda mais elementar de similaridade, o nível alfabético, onde o algoritmo considera que dois nomes são similares caso compartilhem muitos caracteres em comum [15]. Um conhecido algoritmo que trabalha no nível alfabético é Levenshtein [16] que infere a similaridade entre duas strings considerando apenas a quantidade de operações de inserção, remoção e substituição de caracteres necessárias para transformar a primeira na segunda. Outro conhecido algoritmo que trabalha no nível alfabético é Jaro-Winkler [17], que leva em consideração a quantidade de caracteres iguais das duas strings que estejam localizados em posições próximas e também a semelhança entre o prefixo (caracteres iniciais) das duas strings.

Entretanto, a literatura aponta que os algoritmos de Levenshtein, Jaro-Winkler e Jaccard não costumam ter desempenho eficaz em problemas onde a semântica dos nomes é um aspecto importante para a determinação da similaridade [1, 15] – como é o caso do casamento de nomes de produtos, em que o uso de sinônimos e hiperônimos precisa ser levado em conta.

Sendo assim, este trabalho realiza uma comparação entre Levenshtein, Jaro-Winkler (dois algoritmos com abordagem alfabética), Jaccard (algoritmo com abordagem léxica) e uma abordagem semântica baseada em wordnet, proposta em [1]. O algoritmo descrito em [1] foi originalmente idealizado para resolver o problema do casamento semântico de nomes de marcas em inglês (ex.: “Red Bull” × “Blue Ox”). Ele foi implementado em um sistema de buscas de um banco de dados de marcas registradas no Reino Unido, com o objetivo de impedir o registro de marcas que confundissem os consumidores. Em testes realizados em um banco de dados de casos provenientes de disputas judiciais, a técnica obteve, em média, acurácia 20% superior a obtida por algoritmos baseados apenas na comparação de caracteres e/ou tokens.

2.3.1 Descrição do Algoritmo Semântico

Esta seção descreve o algoritmo avaliado no presente trabalho, que foi originalmente proposto em [1] e é baseado no modelo de contraste de Tversky [18]. De acordo com esse modelo, a semelhança entre dois objetos pode ser definida como uma combinação linear de suas características comuns e distintas.

Considere duas strings *q* e *t*, cujos conjuntos de tokens sejam denotados por *Q* e *T*, respectivamente. A similaridade entre *q* e *t* é computada a partir de uma fórmula composta por três partes, que leva em conta tanto a similaridade léxica como a semântica:

- PARTE 1: baseada na proporção de termos em comum entre *T* e *Q* (Jaccard).
- PARTE 2: baseada na proporção de sinônimos e hiperônimos dos termos de *Q* que são comuns a *T*.
- PARTE 3: baseada na topologia da wordnet, onde calcula-se o Wu & Palmer médio entre todos os termos de *Q* que não estejam contidos em *T*.

Por exemplo, considere os nomes de produtos *q* = "aipo" e *t* = "salsão (verdura)", cujos respectivos conjuntos de *tokens* são *Q* = {"aipo"} e *T* = {"salsão", "verdura"}. Suponha que a wordnet apresentada na Figura 1 será empregada como fonte externa de conhecimento. A partir desta wordnet, torna-se possível obter o conjunto *R* de sinônimos e hiperônimos dos termos de *Q*, dado por *R* = {"aipo", "salsão", "verdura"}. A similaridade entre *q* e *t* poderá então ser computada da seguinte forma:

- PARTE 1: $|Q \cap T| / |Q \cup T| = 0 / 3 = 0{,}00$.
- PARTE 2: $|R \cap T| / \max(|T|, |Q|) = 2 / 2 = 1{,}00$.
- PARTE 3: (WP("aipo", "verdura") + WP("aipo", "salsão")) / 2 = 0,70.

O valor final de similaridade é dado por: (0,00 + 1,00 + 0,70) / 3 = 0,57.

Neste exemplo, *q* = "aipo" e *t* = "salsão (verdura)" são grafias alternativas para denotar um mesmo produto. Entretanto, se apenas Jaccard fosse utilizada para computar a similaridade entre esses nomes (PARTE 1 da equação), o valor resultante seria igual a 0,00 (valor mínimo). Com a incorporação da wordnet da Figura 1 ao processo de comparação, torna-se possível levar em consideração também a similaridade semântica (PARTES 2 e 3 da equação). Com isto, um valor ajustado (maior) para a similaridade entre *q* e *t* é computado como valor final de similaridade.

3. Metodologia

O algoritmo de [1] foi originalmente proposto para avaliar a similaridade entre nomes de marcas em inglês e utiliza a WordNet de Princeton [11] como fonte externa de conhecimento. Nesse trabalho, o algoritmo será utilizado para avaliar a similaridade semântica de nomes de produtos alimentícios em português, utilizando duas diferentes wordnets para o português brasileiro. A primeira delas é a OpenWordNet-PT, disponível na biblioteca NLTK do Python. A segunda, uma wordnet específica contendo apenas nomes de produtor alimentícios que foi batizada como OntoSNIPC.

A OpenWordNet-PT ou OpenWN-PT [6], é uma wordnet livre para a língua Portuguesa que vem sendo desenvolvida desde o ano de 2010. Sua versão inicial foi gerada combinando dados da WordNet de Princeton, da UWN / MENTA [19] e dos conceitos base da EuroWordNet [20]. Desde então, tem sido constantemente melhorada, seja manualmente ou através do uso de outras fontes e de técnicas de aprendizado de máquina [12]. Ao longo dos últimos anos, diversos projetos escolheram a OpenWN-PT para servir como base léxica do Português, dentre os quais a conhecida biblioteca NLTK [7] para processamento de linguagem natural no ambiente Python.

A segunda wordnet utilizada no presente trabalho é a OntoSNIPC. Ela foi construída de forma semiautomática, a partir de dois arquivos públicos do IBGE que contêm descrições e classificações de produtos, ambos disponibilizados em [8]. O primeiro arquivo consiste na estrutura IPCA-INPC. Este arquivo organiza mais de seiscentos produtos e serviços de maneira hierárquica em grupos, subgrupos, itens e subitens (ou seja, em 4 diferentes níveis, sendo o nível de grupo o mais agregado e o de subitem o menos agregado). Um recorte do arquivo é mostrado na Figura 2.

```
0         Índice Geral
1000000     Alimentação e Bebidas
1100000       Alimentação no Domicílio
1101000         Cereais, Legum. e Oleaginosas
1101002           Arroz
1101051           Feijão-Mulatinho
1101052           Feijão-Preto
```

Figura 2: Trecho do arquivo com a estrutura IPCA-INPC.

No recorte da Figura 2, "Alimentação e Bebidas" representa o grupo, "Alimentação no Domicílio" é o subgrupo, "Cereais, Legum. e Oleaginosas" representa o item e "Arroz", "Feijão-Mulatinho" e "Feijão-Preto" são subitens. Observe que a estrutura IPCA-INPC já representa, por si só, um tipo bem simples de wordnet.

O segundo arquivo empregado na construção da OntoSNIPC é um arquivo que contém dois mil nomes de produtos alimentícios usados na Pesquisa de Orçamentos Familiares (POF) do IBGE [21]. Basicamente, o processo de construção da OntoSNIPC consistiu em acrescentar cada nome de produto da POF um nível abaixo de seu subitem correspondente na estrutura IPCA-INPC. A partir do arquivo da POF também foi possível identificar sinônimos para diversos produtos que fazem parte do IPCA-INPC. Neste caso, cada nome do IPCA-INPC e seus respectivos sinônimos provenientes da POF foram agrupados em um mesmo synset.

4. Experimento

4.1. Base de Dados e Medidas de Avaliação

A base de dados utilizada neste trabalho consiste em 1.000 pares de nomes de produtos alimentícios vendidos em supermercados corretamente casados com os nomes de produtos utilizados na base de dados do SNIPC do IBGE. Alguns exemplos são apresentados na Tabela 2. A base pode ser obtida em https://raw.githubusercontent.com/edubd/bd/main/bd_1000.csv.

Para comparar as abordagens, o problema foi tratado como um problema de recuperação da informação [22] cujo objetivo é encontrar o nome SNIPC correspondente para cada nome de produto de supermercado. As medidas de Precisão (*precision*), Revocação (*recall*) e F1, respectivamente definidas nas equações (3), (4) e (5), foram empregadas para medir o desempenho dos algoritmos. Nas fórmulas, o conjunto com o único nome da SNIPC que representa o par correto do nome do produto vendido em supermercado é denotado por *Relevante* enquanto o conjunto de nomes da SNIPC com maior valor de similaridade atribuída pelo algoritmo de casamento de nomes é denotado por *Recuperados*.

$$Precisão = \frac{|\,Relevante\,| \cap |Recuperados|}{|\,Recuperados\,|}, \tag{3}$$

$$Revocação = \frac{|\,Relevante\,| \cap |Recuperados|}{|\,Relevante\,|}, \tag{4}$$

$$F1 = 2 \times \frac{Precisão \times Revocação}{Precisão + Revocação}, \tag{5}$$

Tabela 2
Exemplos de pares de nomes de produtos da base de dados utilizada no experimento

Nome do produto no supermercado	Nome no SNIPC
1 Kg Levedo (levedura) De Cerveja 100% Puro	Levedo de cerveja
Jambu Unidade	Jambuaçu
Macaxeira Regional Kg	Mandioca (aipim)
Maminha Bovina Extra Limpa Resfriada BASSI 1,2kg	Alcatra
Ovos Brancos Grandes Bandeja com 20 Unidades	Ovo de galinha
Wafer Parati Minueto Mega Morango 105g	Biscoito

4.2. Experimento

O experimento comparou o desempenho de quatro diferentes algoritmos de similaridade: Levenshtein, Jaro-Winkler, Jaccard e o algoritmo Semântico, descrito na Subseção 2.3.1. Este último foi avaliado com o uso de duas diferentes wordnets: OntoSNIPC e OpenWordNet-PT. A implementação do algoritmo Semântico foi feita na linguagem Python. Para os demais algoritmos, utilizou-se as implementações disponibilizadas no pacote strsimpy [23]. Os resultados são apresentados na Tabela 3.

Em consonância com o observado na literatura [1, 15], o algoritmo de Levenshtein (puramente baseado na similaridade de caracteres) obteve um fraco desempenho para o problema do casamento de nomes de produtos. Além de não avaliar a semântica, este algoritmo é afetado quando os nomes que são comparados possuem comprimentos diferentes [3, 16], situação mais comum na base de dados do experimento (como mostrado nos exemplos da Tabela 2).

O algoritmo Jaro-Winkler, também baseado na similaridade de caracteres, obteve desempenho superior ao de Levenshtein. Isto pode ser justificado pelas suas seguintes características: (i) Jaro-Winkler leva em consideração caracteres que estão localizados em posições diferentes, porém dentro de uma vizinhança próxima nas strings em comparação; (ii) o algoritmo atribui peso maior para strings que possuem prefixos parecidos. Por estas razões, acaba por ser menos afetado quando as strings que estão sendo comparadas possuem comprimento muito diferente.

A Tabela 3 mostra ainda que o algoritmo de Jaccard, baseado na proporção de tokens em comum entre os nomes, obteve desempenho superior a Levenshtein e Jaro-Winkler: Precisão de 56,19%, Revocação de 67,80% e F1 de 57,67%.

Por sua vez, o algoritmo Semântico utilizando a OntoSNIPC como fonte externa de conhecimento obteve resultados bem superiores aos de todas as demais técnicas avaliadas: Precisão de 72,07%, Revocação de 87,00% e F1 de 78,83%. Note que o valor de F1 obtido pelo algoritmo Semântico com a OntoSNIPC é mais de 20% superior ao obtido por Jaccard e mais de 25% superior ao obtido por Jaro-Winkler. Estes resultados são similares aos que foram reportados em [1].

Tabela 3
Resultados dos diferentes algoritmos para avaliar a similaridade entre strings

Algoritmo	Wordnet	*Precisão*	*Revocação*	*F1*
Levenshtein	-	0,2160	0,2490	0,2239
Jaro-Winkler	-	0,5108	0,5130	0,5115
Jaccard	-	0,5619	0,6780	0,5767
Semântico	OntoSNIPC	**0,7207**	**0,8700**	**0,7883**
Semântico	OpenWordNet-PT	0,5575	0,6270	0,5705

Entretanto, o algoritmo Semântico combinado com a OpenWordNet-PT não foi mais eficaz do que Jaccard de acordo com as três medidas de desempenho consideradas (conforme apresentado na última linha da Tabela 3). Duas possíveis justificativas são apresentadas a seguir.

Primeiro, tem-se que menos de 60% dos nomes dos produtos presentes na base de dados utilizada no experimento constam na OpenWordNet-PT. Ou seja: a cobertura dessa wordnet é relativamente baixa para a base de dados do experimento. De fato, foi constatado que diversos nomes de produtos comumente vendidos em supermercados das cidades brasileiras, como "cupuaçu" (fruta), "cherne" (tipo de peixe), "alcatra" e "chã" (tipos de corte de carne bovina), estão ausentes da OpenWordNet-PT. Por outro lado, esses nomes aparecem nos arquivos de classificações do IBGE que foram empregados na construção da OntoSNIPC.

A segunda justificativa é o efeito da ambiguidade. Enquanto a OntoSNIPC é uma base léxica que contém apenas nomes de produtos, a OpenWordNet-PT é uma wordnet de fato. Por isso, é comum que uma palavra esteja presente em mais de um synset da OpenWordNet-PT, ao passo que essa situação é rara na OntoSNIPC. Infelizmente, o algoritmo Semântico proposto em [1] não realiza o tratamento de ambiguidades, uma vez que, ao processar cada termo *t* de uma string, ele utiliza todos os synsets em que *t* ocorre para montar os conjuntos de sinônimos e hiperônimos deste termo.

5. Conclusões

Este trabalho abordou a tarefa de casamento de nomes de produtos em Português (textos curtos), através da avaliação de um algoritmo Semântico baseado em wordnets. O algoritmo foi testado com o uso de duas diferentes wordnets: OntoSNIPC (base léxica criada de forma semiautomática a partir de classificações de produtos do IBGE) e OpenWordNet-PT (wordnet livre para o Português brasileiro). A base de dados utilizada no experimento contém 1.000 nomes de produtos alimentícios vendidos em supermercados corretamente casados com os nomes de produtos utilizados na base de dados do SNIPC do IBGE.

O algoritmo Semântico combinado com a OntoSNIPC obteve resultados 27% e 21% superiores aos obtidos pelos conhecidos algoritmos Jaro-Winkler e Jaccard, respectivamente, de acordo com a medida de F1. Por outro lado, ao combinar este mesmo algoritmo Semântico com a OpenWordNet-PT, o desempenho não foi superior ao de Jaccard.

Tendo em vista os resultados promissores da OntoSNIPC, para futuros trabalhos pretende-se inicialmente expandir esta base léxica, acrescentando produtos que não são alimentícios, mas que também fazem parte do SNIPC (ex.: produtos de limpeza, utensílios domésticos etc.). Adicionalmente, pretende-se incorporar técnicas de análise sintática [13] ao processo de casamento, para que seja possível atribuir maior peso às palavras que são mais discriminativas no nome de um produto. Por exemplo, ao comparar "feijão branco" e "feijão preto", o substantivo "feijão" deve ser considerado mais discriminativo do que os adjetivos "branco" e "preto". Por fim, pretende-se avaliar outras medidas de similaridade diferentes de Wu & Palmer na parte 3 da fórmula do algoritmo semântico, como, por exemplo, a medida de Resnik [24].

6. Referências

[1] F. M. Anuar, R. Setchi, and Y-K Lai, Semantic retrieval of trademarks based on conceptual similarity, IEEE trans. on system, man, and cybernetics: Systems 46.2 (2016) 220–233. doi: 10.1109/TSMC.2015.2421878.

[2] L. Gravano, et al. Using q-grams in a DBMS for approximate string processing. IEEE data engineering bulletin 24.4 (2001) 28–34.

[3] K. L. A. Branting, A comparative evaluation of name matching algorithms, in: Proceedings of the 9th International Conference on Artificial Intelligence and Law, ICAIL'03, ACM Press, Edinburgh, Scotland, 2003, pp. 224–232. doi: 10.1145/1047788.1047837.

[4] IBGE, Para Compreender o INPC (Um Texto Simplificado), 7a. ed., IBGE, Rio de Janeiro, RJ, 2016.

[5] J. Hillen, Web scraping for food price research, Britsh Food Journal 121.12 (2019) 3350–3361. doi: 10.1108/BFJ-02-2019-0081.
[6] V. de Paiva, A. Rademaker, and G. de Melo, OpenWordNet-PT: An open Brazilian wordnet for reasoning, in: Proceedings of COLING 2012: Demonstration Papers, COOLING Organizing Committee, Mumbai, India, 2012, pp. 353–360.
[7] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, 1st ed., O'Reilly, Sebastopol, CA, 2009.
[8] CONCLA, Comissão Nacional de Classificações, 2022. URL: https://concla.ibge.gov.br/classificacoes/download-concla.html.
[9] J. Leskovec, A. Rajaraman, and J. Ullman, Mining of Massive Datasets, 3rd ed., Cambridge University Press, Palo Alto, CA, 2020.
[10] R. Sinoara, J. Antunes, and S. O. Rezende, Text mining and semantics: A systematic mapping study, Journal of the brazilian computer society, 23.9 (2017), 1–20. doi: 10.1186/s13173-017-0058-7
[11] C. Fellbaum, WordNet: An Electronic Lexical Database, 3rd ed., Cambridge University Press, New York, NY, 1998.
[12] H. G. Oliveira, V. de Paiva, C. Freitas, A. Rademaker, L. Real, and A. Simões, As wordnets do português, Oslo studies in languages, 7.1 (2015), 397–424. doi: 10.5617/osla.1445
[13] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. (draft), Palo Alto, CA, 2022. URL: https://web.stanford.edu/~jurafsky/slp3/
[14] Z. Wu and M. Palmer, Verbs semantics and lexical selection, in: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, ACL'94, ACM Press, New Mexico, USA, 1994, pp. 133–138. doi: 10.3115/981732.981751
[15] T. P. Meirelles, E. C. Gonçalves, and D. T. Gomes, Pareamento de nomes de produtos e serviços utilizando medidas de similaridade textual nos níveis alfabético, léxico e semântico, Cadernos do ime – série informática, 129 (2021), 104–117. doi: 10.12957/cadinf.2021.68557.
[16] P. Christen, A comparison of personal name matching: Techniques and practical issues, in: Proceedings of the IEEE 6th Data Mining Workshop, ICDMW'06, IEEE, Hong Kong, China, 2006, pp. 290–294. doi: 10.1109/ICDMW.2006.2
[17] W. E. Winkler, Advanced methods for record linkage, in: JSM Proceedings, Survey Research Methods Section, ASA, Toronto, Canada, 1994, pp. 467–472.
[18] A. Tversky, Features of similarity, Psychological Review, 84 (1977) , 327–352.
[19] UWN / MENTA – Towards a Universal Multilingual Wordnet, 2022: URL: https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/uwn
[20] P. Vossen, et al., The EuroWordNet Base Concepts and Top Ontology, 1st ed., Centre National de la Recherche Scientifique, Paris, 1998.
[21] IBGE, POF – Pesquisa de Orçamentos Familiares, 2022. URL: https://www.ibge.gov.br/estatisticas/sociais/saude/24786-pesquisa-de-orcamentos-familiares-2.html?=&t=o-que-e.
[22] J. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval: The Concepts and Technology Behind Search, 2nd ed., Addison-Wesley Professional, New York, NY, 2011.
[23] strsimpy, python-string-similarity, 2022. URL: https://pypi.org/project/strsimpy/.
[24] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in: Proceedings of the 14th Joint Conference on Artificial Intelligence, IEEE, Quebec, Canada, 1995, pp. 448–453. doi: 10.5555/1625855.1625914.