Improving the Effectiveness of Name Matching Algorithms with a Portuguese Wordnet

Eduardo Corrêa Gonçalves[1], Thiago Pereira Meirelles[1]

[1]Escola Nacional de Ciências Estatísticas (ENCE/IBGE)
Rio de Janeiro – RJ – Brazil

eduardo.correa@ibge.gov.br, thiagopmeirelles@gmail.com

Abstract. *Name matching is the task of comparing two names to determine whether they denote the same entity or not. The use of a wordnet-like ontology allows name matching algorithms to exploit semantic issues, thereby potentially increasing their effectiveness. In this paper, we propose an algorithm for name matching that takes account of semantic information through the use of Onto.PT, a wordnet-like ontology for Portuguese. The aim of the present study is twofold. First, to evaluate the proposed algorithm on a public dataset that keeps thousands of names of products and services in Portuguese. Second, to discuss the main advantages and pitfalls of using Onto.PT in name matching processes.*

1. Introduction

Name matching is the task of comparing two names to determine if they denote the same entity or not [Anuar et al. 2016, Branting 2003]. Practical applications include the matching of personal names (e.g.: “Jane Sousa” × “Jane Souza”), place names (“UCSD” × “University of California San Diego”), and company/brand names (“Topper” × “Tobber”), among others. In most applications, the names to be compared tend to be very short, composed of no more than six words [Davis Jr. and Salles 2009, Gali et al. 2016, Putnam and Verrinder 2015].

In this work, we address the problem of performing the semantic matching of product and service names in Portuguese by incorporating an external knowledge source into the comparison process. The knowledge source employed in this study is Onto.PT [Oliveira and Gomes 2014], a public domain wordnet-like ontology for Portuguese. The aim of the present paper is twofold. Firstly, to report the results of an experiment performed on a publicly available dataset that stores thousands of names of products and services in Portuguese. The experiment compared the performance of well-known character-based similarity algorithms against a new proposed algorithm that takes semantics into account through the use of Onto.PT. The second goal is to identify the main advantages and pitfalls of applying Onto.PT to name matching processes. The rest of this paper is organized as follows. Section 2 revises related work. Our similarity algorithm that employs Onto.PT is described in Section 3. In Section 4, we report experimental results. Finally, we give concluding remarks in Section 5.

2. Related Work

Over the last decades, several character-based similarity algorithms for name matching have been proposed in the literature [Cristen 2006, Gali et al. 2016, Jaro 1989, Leskovec et al. 2020, Winkler 1994]. These techniques determine the similarity between two names by only evaluating if they share many common characters. In this subsection, we review the character-based algorithms used in this paper. In the definitions throughout the text, we adopted the following notation: *n1* and *n2* are two names whose similarity score is to be computed and $|n1|$ and $|n2|$ represent the lengths of *n1* and *n2*, respectively.

Levenshtein similarity [Cristen 2006], shown in Equation (1), infers how similar two names are based on the number of edit operations (ED) it takes to change one name into the other. The allowed operations are character deletions, insertions, or substitutions.

$$S_L(n1, n2) = 1 - \left(\frac{ED(n1, n2)}{max\ (|n1|, |n2|)}\right) \tag{1}$$

Jaro similarity [Jaro 1989] is another kind of edit-based algorithm which is computed according to Equation (2). In this equation, *c* and *t* represent the number of character matches and transpositions, respectively. A character from *n1* and a character from *n2* match if they are identical and located in the same position or in a range defined by the formula ($max(|n1|, |n2|) / 2) - 1$. The number of transpositions corresponds to the number of matching characters that are in different positions.

$$S_J(n1, n2) = \frac{1}{3}\left(\frac{c}{|n1|} + \frac{c}{|n2|} + \frac{c - t/2}{c}\right) \tag{2}$$

A *q*-gram associated with a string *s* can be defined as any substring of length *q* found within *s* [Leskovec et al. 2020]. Given a name *n*, it is possible to generate a vector containing all its *q*-grams. For instance, the 2-gram vector for the name n = "pepper" can be defined as: v = ['pe', 'ep', 'pp', 'pe', 'er']. Since the substring 'pe' appears twice within *n*, it might be more interesting to store each *q*-gram along with its frequency: *v* = [('pe',2), ('ep',1), ('pp',1), ('er',1)]. The similarity between two *q*-gram vectors can be measured using Cosine, presented in Equation (3). In this formula, *v1.v2* represents the standard dot product whereas $|v1||v2|$ corresponds to the product of the vector norms.

$$S_{q-gram}(n1, n2) = Cosine(v1, v2) = \frac{v1.v2}{|v1||v2|} \tag{3}$$

Table 1 shows examples of pairs of names that denote the same entity (in this case, food products) and thus should be assigned a high similarity score. In the first example, *n2* is misspelled, and it is noticeable that S_L and S_J performed more effectively than the *q*-gram approach. On the other hand, if the words in the names are the same but in different orders, as in the second example, *q*-gram works better. In the third example, we have two names that are synonyms with completely different spelling. In this case, none of the measures is effective (all similarity scores are closer to 0 than to 1).

The character-based similarity algorithms presented in this section offer two advantages: they are simple and language independent. However, a considerable disadvantage lies in that they ignore the possible occurrence of semantic relationships between the names under comparison. In the next section, we discuss how to extend the character-based methods in order to enable them to also exploit semantic issues.

Table 1. Name matching by different character-based algorithms

n1	*n2*	S_L	S_J	$S_{2\text{-}gram}$	$S_{3\text{-}gram}$
pepper	pwpper	0.8333	0.8889	0.6761	0.5000
peas and corn	corn and peas	0.3846	0.5564	0.8333	0.6363
cassava	manioc	0.1429	0.4365	0.0000	0.0000

3. The Proposed Method

3.1. Wordnets

Two names can be considered semantically similar if they carry the same meaning or evoke the same concept [Anuar et al. 2016, Sinoara et al. 2017]. In order to determine the semantic similarity between names, it is necessary to incorporate an external source of knowledge into the matching process. Nowadays, the two most used types of external sources are word embeddings and wordnet-like ontologies [Jurafsky and Martin 2020]. In this work, we opt for a solution based on a wordnet due to its inherent ability to produce decisions that can be easily interpretable.

A wordnet-like ontology is a structure composed of synsets and the semantic relations that connect these synsets [Branco et al., 2020, Fellbaum 1998, Oliveira and Gomes 2014, de Paiva et al. 2016]. Each synset is a set of synonymous word senses associated with its part of speech and a gloss (a dictionary-style definition). Relations between synsets can include hypernymy (links more general concepts to more specifics ones), antonymy (semantic opposition), meronymy (part-whole relation), and others. Therefore, a wordnet can be seen as a graph where nodes are synsets and edges represent their semantic relationships. Figure 1 presents an example of a hypothetical wordnet in which edges represent hypernymy relations.

3.2. Hybrid Similarity Algorithm

We believe that adapting an existing character-based similarity algorithm to allow it to also evaluate semantic closeness would be more suitable for comparing names of products and services. Based on this assumption, we proposed the hybrid similarity function shown in Equation (4). This function simultaneously considers the analysis of character-based similarity (first term), lexical similarity (second term) and semantic similarity (last term).

$$S_H(n1, n2) = \frac{1}{3}\left(S_{char}(n1, n2) + \frac{|T_{n1} \cap T_{n2}|}{|T_{n1} \cup T_{n2}|} + \frac{|R_{n1} \cap T_{n2}|}{max(|T_{n1}|, |T_{n2}|)}\right) \tag{4}$$

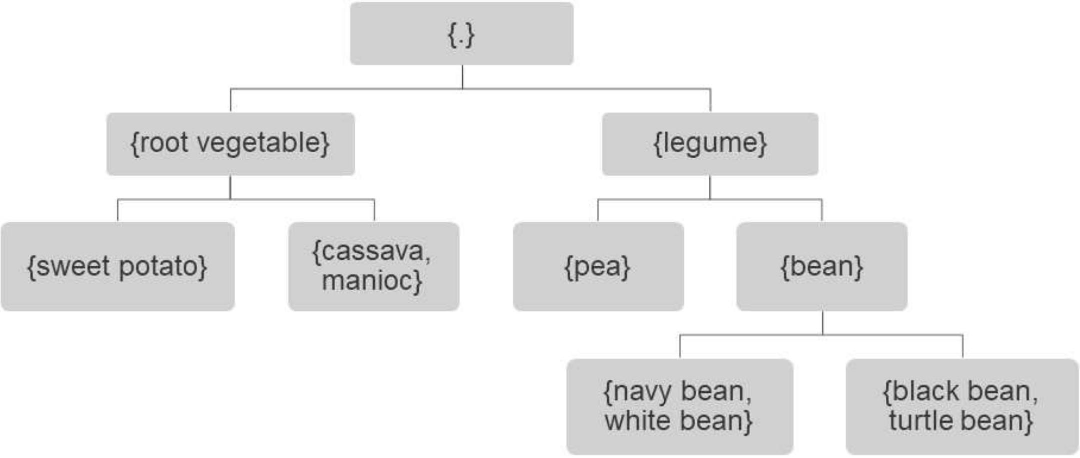

Figure 1. Wordnet-like ontology where edges represent hypernymy relations

In Equation (4):

- The first term, $S_{char}(n1, n2)$, can be any chosen character-based similarity function, such as Levenshtein, Jaro, or Cosine. In the experiment reported in this paper, we used a combination of these functions: $max(S_L, S_J, S_{3\text{-}gram})$.
- T_{n1} and T_{n2} correspond to the set of tokens that compose $n1$ and $n2$, respectively. For instance, $n1$ = “cassava soup” is transformed into T_{n1} = {“cassava”, “soup”}. Hence the second part of the equation computes the ratio of the size of the intersection T_{n1} and T_{n2} to the size of their union. This score reflects similarity at the lexical level [Sinoara et al. 2017].
- R_{n1} corresponds to T_{n1} augmented with the set of tokens that are directly related to each token in T_{n1}. In this work, these are the synonyms and hypernyms obtained from Onto.PT. For example, for T_{n1} = {“cassava”} we have R_{n1} = {“cassava”, “manioc”, “root vegetable”} as “manioc” and “root vegetable” correspond, respectively, to the synonym and hypernym of “cassava”. Thus, the score obtained in the third part of the equation reflects similarity at the semantic level.

Next, we give an example on how to compute the similarity between the product names $n1$ = “sweet potato and manioc” and $n2$ = “cassava + sweet potato” using the proposed function and employing the wordnet presented in Figure 1. Also consider that $S_{3\text{-}gram}$ was chosen to assess character similarity (first part of Equation 4). In this example, we have T_{n1} = {“manioc”, “sweet potato”} and T_{n2} = {“cassava”, “sweet potato”}. Note that the tokens "and" (stop word) and "+" (symbol) are discarded. The set of words directly related to the words in T_{n1} is defined as R_{n1} = {“manioc”, “sweet potato”, “cassava”, “root vegetable”}. Hence, the value of $S_H(n1, n2)$ is computed as follows:

- $S_{3\text{-}gram}(n1, n2) = 0.6048$.
- $|T_{n1} \cap T_{n2}| \,/\, |T_{n1} \cup T_{n2}| = 1 / 3 = 0.3333$.
- $|R_{n1} \cap T_{n2}| \,/\, max(T_{n1}, T_{n2}) = 2 / 2 = 1.000$.

- Final score: $S_H(n1, n2) = 1/3 \times (0.6048 + 0.3333 + 1.000) = 0.6460$.

It is important to mention that our hybrid function S_H is an adaptation of the similarity function first proposed by [Anuar et al. 2016]. Nonetheless, there are two important differences. First, we proposed a similarity function that computes a score based on the combination of character, lexical, and semantic similarity. On the other hand, the method proposed in [Anuar et al. 2016] disregards character-based closeness. The second difference is that the ontology employed in this study is Onto.PT [Oliveira and Gomes 2014] instead of Princeton WordNet [Fellbaum 1998] as our goal is to evaluate names in Portuguese rather than in English.

4. Experiment

4.1. Experimental Methodology

The dataset studied in this work consists of 4,956 pairs of matched names in the Portuguese language [IBGE 2021]. All names in the dataset correspond to descriptions of products and services that can be acquired by families that live in the metropolitan areas of the major Brazilian cities. An excerpt is shown in Table 2.

Table 2. An excerpt from the studied dataset

POF name	SNIPC name
ARROZ POLIDO	Arroz
ARROZ COM CASCA	Arroz
COCO BURITI	Buriti (Coco)
MAIZENA	Amido de Milho

For each pair (*p*, *s*) in the dataset, *p* represents a name used in the questionnaire of the Consumer Expenditure Survey (POF-IBGE) whilst *s* corresponds to a name used by the National System of Consumer Price Indexes (SNIPC-IBGE). It is important to state that in this dataset, the relationship between SNIPC names and POF names is *1* to *N*, which means that one name from SNIPC can be matched with one or more names from POF. Conversely, each POF name matches one and only one SNIPC name. To conduct the experiments, the only preprocessing tasks we carried out in the dataset were the following: converting names to lowercase, removing punctuations and correcting POF names that were not accented.

To compare the algorithms presented in Sections 2 and 3, we decided to treat the name matching problem as an information retrieval (IR) problem [Baeza-Yates and Ribeiro-Neto 2011, Egozi et al. 2011] where the goal was to find the name *s* from SNIPC that best matches a POF name *p*. Although there is only and exactly one correct SNIPC match for each POF name, the evaluated algorithms often return two or more names as the best match (i.e., they may return different names with the same highest similarity score). Due to this fact, we decided to assess the performance of the similarity algorithms using measures capable of taking into consideration results that are partially correct. These are Precision (Pre), Recall (Rec), and F1-Score (F1) [Jurafsky and Martin 2020], respectively shown in Equations (5), (6), and (7). In the formulas, the set with the single relevant SNIPC name for a POF name is denoted as *Relevant* whilst the set of SNIPC names that were identified as the most similar according to the similarity algorithm is denoted as *Retrieved*. In our experiments, the IR task was performed separately for each POF name and the results were averaged.

$$Pre = \frac{|Relevant| \cap |Retrieved|}{|Retrieved|} \tag{5}$$

$$Rec = \frac{|Relevant| \cap |Retrieved|}{|Relevant|} \tag{6}$$

$$F1 = 2 \times \frac{Pr \times Re}{Pr + Re} \tag{7}$$

4.2. Results

We compared the hybrid similarity algorithm S_H proposed in Section 3 against the character-based algorithms presented in Section 2 with the goal of investigating whether the use of Onto.PT increases the effectiveness of the name matching process. We used the implementations of character-based algorithms available at the strsimpy package [Strsimpy 2021], an open-source Python library that implements different string similarity and distance algorithms. Results are shown in Table 3.

Table 3. Performance of the character-based algorithms against the S_H algorithm

algorithm	Pre	Rec	F1
Levenshtein (S_L)	0.4480	0.4702	0.4547
Jaro (S_J)	0.5373	0.5556	0.5432
Cosine 2-gram ($S_{2\text{-}gram}$)	0.6061	0.6204	0.6106
Cosine 3-gram ($S_{3\text{-}gram}$)	0.6153	0.6306	0.6202
$max(S_L, S_J, S_{3\text{-}gram})$	0.5384	0.5567	0.5443
Hybrid Function with Onto.PT (S_H)	**0.6674**	**0.6756**	**0.6700**

The first column indicates the name of the algorithm whilst columns 2, 3, and 4, respectively, show the obtained values for Precision, Recall, and F1-Score. The first five lines of the table present results of purely character-based algorithms whilst the last one the results obtained by the hybrid function with Onto.PT. It is possible to observe that the hybrid similarity algorithm achieved the best results in the three evaluation metrics (with Precision, Recall and F1 superior to 66%). Thus, in the studied dataset, the use of Onto.PT in tandem with the proposed S_H function provided a gain of 5.0% in terms of F1-Score in comparison with the best performing character-based algorithm (Cosine 3-gram).

In what follows, we briefly discuss the pros and cons of using Onto.PT as the external source of knowledge to perform name matching. We consider that Onto.PT has two appealing characteristics. First, it is freely available as a single standard RDF/OWL file that can be easily integrated to any system. Second, it covers a comprehensive number of lexical items. The latest version, Onto.PT v.0.6 [Onto.PT 2013], includes 67,873 nouns and 20,760 adjectives. We found that 75.87% and 77.11% of the *single words* that appear in SNIPC and POF names, respectively, are also present in Onto.PT.

On the other hand, since Onto.PT was built by a fully automated process, it is prone to errors and limitations, as pointed out in [Oliveira 2016, Oliveira and Gomes 2014]. First, only 65% of the hypernym connections proved to be perfectly accurate [de Paiva et al. 2016]. Second, most paths from the more specific synsets to the root of the ontology are not more than three edges long [Oliveira and Gomes 2014], hindering us from evaluating path-based algorithms for computing semantic similarity [Anuar et al. 2016, Croft et al. 2013; Li et al. 2006, Wu and Palmer 1994]. Third, although Onto.PT covers most of the single words in the database, we identified that the same is not true for the *open compounds*. This is a relevant disadvantage in the studied problem since product names are often composed of two nouns or a noun and an adjective. For instance, product names like "milho-verde" ("green corn") and "arroz branco" ("white rice") are absent from Onto.PT, although they do exist as lexical items in Princeton WordNet.

5. Conclusions

In this work, we proposed a hybrid similarity function for name matching that employs Onto.PT as external knowledge source and simultaneously accounts for the analysis of three similarity aspects: character, lexical, and semantics. Experiments on a publicly available dataset of product and service names suggest that this approach has led to more effective results. To the best of our knowledge, this is the first time Onto.PT is employed as a tool for enhancing the effectiveness of name matching algorithms. As future research, we plan to construct a domain ontology of products and services to be used by the hybrid similarity function. We consider that the construction of this ontology will be facilitated if we inherit several of the lexical items that are already included in Onto.PT.

References

Anuar, F. M., Setchi, R. and Lai, Y-K. (2016). Semantic retrieval of trademarks based on conceptual similarity. In *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(2), pages 220–233. IEEE.

Baeza-Yates, J. and Ribeiro-Neto, B. (2011). Modern Information Retrieval: The Concepts and Technology Behind Search. Addison-Wesley Professional, 2nd edition.

Branco, A. et al (2020) "The MWN.PT WordNet for Portuguese: Projection, Validation, Cross-lingual Alignment and Distribution", In: Proc. of the 12th Conf. on Language Resources and Evaluation (LREC), ELRA, p. 4859–4866.

Branting, K. L. A. (2003) "A Comparative Evaluation of name matching Algorithms", In: Proc. of the 9th Intl' Conf. on Artificial Intelligence and Law (ICAIL), ACM, p. 224–232.

Christen, P. (2006) "A Comparison of Personal Name Matching: Techniques and Practical Issues", In: Proc. of the IEEE 6th Data Mining Workshop (ICDMW'06), IEEE, p. 290–294

Croft, D. et al. (2013) "A Fast and Efficient Semantic Short Text Measure", In: Proc. of the 13rd UK Workshop on Computational Intelligence (UKCI), IEEE, p. 221–227.

Davis Jr., C. A. and Salles, E. (2009) "Approximate String Matching for Geographic Names and Personal Names", In: Proc. of the IX Brazilian Symposium on GeoInformatics, INPE, p. 49–60.

de Paiva, V. et al. (2016) "An overview of Portuguese WordNets", In: Proc. of the 8th Global WordNet Conference (GWC 2016), GWA, p. 74–81.

Egozi, O., Markovitch, S., and Gabrilovich, E. (2011) Concept-based information retrieval using explicit semantic analysis. In *ACM Trans Inf Syst*, 29(2) pages 8:1–8:34. ACM.

Fellbaum, C. (1998). WordNet: an electronic lexical database, MIT Press, Cambridge.

Gali, N., Mariescu-Istodor, R. and Fränti, P. (2016) "Similarity Measures for Title Matching", In: Proc. of the 23rd Int'l Conf. on Pattern Recognition, IAPR, p. 1549–1554.

Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. In *Journal of the American Statistical Association*, 84 (406), pages 414–420. Taylor & Francis.

Jurafsky, D. and Martin, J. H. (2020), Speech and Language Processing, Stanford, 3rd edition (draft).

IBGE (2021). IPCA - Índice Nacional de Preços ao Consumidor Amplo (Downloads). https://www.ibge.gov.br/estatisticas/economicas/precos-e-custos/9256-indice-nacional-de-precos-ao-consumidor-amplo.html?=&t=downloads.

Leskovec, J., Rajaraman, A. and Ullman, J. (2020), Mining of Massive Datasets Cambridge University Press, 3rd edition.

Li, Y., et al. (2006). Sentence similarity based on semantic nets and corpus statistics. In *IEEE Transactions on Knowledge and Data Engineering*, 18(8), pages 1138–1150. IEEE.

Oliveira, H. G. (2016). "CONTO.PT: Groundwork for the Automatic Creation of a Fuzzy Portuguese Wordnet". In: Proc. of the 12th Intl' Conf. on the Computational Processing of Portuguese (PROPOR), ACL, p. 283–295.

Oliveira, H. G. and Gomes, P. (2014). Onto.PT: A flexible approach for creating a Portuguese wordnet automatically. In *Language Resources and Evaluation*, 48(2), pages 373–393. Springer.

Onto.PT v.0.6 (2013). http://ontopt.dei.uc.pt/.

Putnam, K. and Verrinder, S. (2015) "What's in a Name? Fast Fuzzy String Matching", In: Midwest.io. https://www.youtube.com/watch?v=s0YSKiFdj8Q&t=1340s.

Sinoara, R., Antunes, J. and Rezende, S. O. (2017). Text mining and semantics: A systematic mapping study. In *Journal of the Brazilian Computer Society*, 23(9), pages 1–20. SpringerOpen.

Strsimpy 0.2.1. (2021). https://pypi.org/project/strsimpy/.

Winkler, W. E. (1994) "Advanced Methods for Record Linkage", In: JSM Proceedings, ASA, p. 467–472.

Wu, Z. and Palmer, M. (1994). "Verbs Semantics and Lexical Selection". In: Proc. of the 32nd Annu. Mtg. of Assoc. for Computational Linguistics, ACL. p. 133–138.