# ASAPP 2.0: Advancing the state-of-the-art of semantic textual similarity for Portuguese

**Ana Alves**
CISUC / ISEC, Polytechnic Institute of Coimbra, Portugal
ana@dei.uc.pt
https://orcid.org/0000-0002-3692-338X

**Hugo Gonçalo Oliveira**
CISUC / Department of Informatics Engineering, University of Coimbra, Portugal
hroliv@dei.uc.pt
https://orcid.org/0000-0002-5779-8645

**Ricardo Rodrigues**
CISUC / ESEC, Polytechnic Institute of Coimbra, Portugal
rmanuel@dei.uc.pt
https://orcid.org/0000-0002-6262-7920

**Rui Encarnação**
CISUC, University of Coimbra, Portugal
race@dei.uc.pt
https://orcid.org/0000-0002-5176-4137

## Abstract

Semantic Textual Similarity (STS) aims at computing the proximity of meaning transmitted by two sentences. In 2016, the ASSIN shared task targeted STS in Portuguese and released training and test collections. This paper describes the development of ASAPP, a system that participated in ASSIN, but has been improved since then, and now achieves the best results in this task. ASAPP learns a STS function from a broad range of lexical, syntactic, semantic and distributional features. This paper describes the features used in the current version of ASAPP, and how they are exploited in a regression algorithm to achieve the best published results for ASSIN to date, in both European and Brazilian Portuguese.

**2012 ACM Subject Classification** Computing methodologies → Natural language processing

**Keywords and phrases** natural language processing, semantic textual similarity, semantic relations, word embeddings, character n-grams, supervised machine learning

**Digital Object Identifier** 10.4230/OASIcs.SLATE.2018.12

**Funding** This work was partially financed by the ERDF European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project REMINDS – UTAP-ICDT/EEI-CTP/0022/2014.

## 1 Introduction

Computing the similarity of words or sentences in terms of their meaning is an active area of research in Natural Language Processing (NLP) and understanding (NLU). This is confirmed by related shared tasks, such as SemEval STS [2, 1, 8], which required the manual compilation of annotated data for benchmarking this specific task. Most successful approaches for English

7th Symposium on Languages, Applications and Technologies (SLATE 2018).
Editors: Pedro Rangel Henriques, José Paulo Leal, António Leitão, and Xavier Gómez Guinovart
Article No. 12; pp. 12:1–12:17
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

learn a similarity function with ensembles of classifiers that combine different metrics, such as n-gram, word or chunk overlap, semantic relations, or distributional similarity [29, 32].

SemEval STS task targets English since 2012 and we can thus say that, for this language, STS is becoming mature. Spanish and Arabic were included in recent editions [1, 8], which have also targeted cross-lingual pairs. For other languages, STS is still in its early days. Until recently, there was not a public dataset for computing semantic similarity between Portuguese sentences. But, in 2016, a collection for STS in Portuguese was released in the scope of the ASSIN shared evaluation [12].

Following the participation of our ASAPP system [4] in ASSIN, we kept working towards the improvement of our results and advancing the state-of-the-art of Portuguese STS. This paper presents a post-evaluation approach to ASSIN, based on supervised machine learning.

The paper describes ASAPPV2.0 and focuses on the features currently extracted, many inspired by related work for English, but adapted for Portuguese. For some, we present the results achieved without supervision which, in some cases, were surprisingly high. Yet, the best results are obtained after learning a regression function, based in a varied set of lexical, syntactic, semantic and distributional features. In the end, we were able to improve not only the previous results of ASAPPv1.5 [14], but also outperform the best official results in ASSIN by two and four points, respectively in the European (PTPT) and Brazilian (PTBR) Portuguese collections.

The remainder of this paper starts by presenting some related work, in section 2, namely a brief overview of the best results for English STS, together with commonly used features, then focusing in Portuguese STS, mostly around the ASSIN task, its collections and best approaches. In section 3, all the exploited features are described and several are illustrated in examples, ending with some results obtained with different feature sets, but without supervision. In section 4, extracted features are exploited to learn a similarity function, this time with supervision, using not only different regression algorithms, but also using the training collections differently, towards the best results in the ASSIN task.

## 2 Related Work

The SemEval shared evaluations include STS tasks since 2012 [2]. Results are typically assessed by the Pearson correlation (hereafter $\rho$, between $-1$ and 1) and the Mean Squared Error (MSE) between values computed by the system and those based on the opinion of several human judges, for the same collection of pairs.

Most successful approaches are supervised. To learn a similarity function, they rely on an ensemble of classifiers and exploit different features, some of which as basic as token or n-gram overlap, but also similarity measures computed in WordNet [10], topics and deep semantic models (see, e.g., [29, 32]). For English, the best $\rho$ has ranged from 0.618, in SemEval 2013, to 0.854 in SemEval 2017. For the adopted baseline – the cosine of the vectors that represent the words in each sentence of the pair – $\rho$ has ranged from 0.311, in 2012, to 0.728, in 2017. For Spanish STS, the best system [32] in SemEval 2017 achieved $\rho = 0.856$, with similar features as the English version.

ASAP [3], which was the starting point of ASAPP, was originally developed for the *Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness* task [23], in SemEval 2014, but participated one year later in SemEval 2015 STS [5], though with modest results.

An earlier approach to Portuguese STS [27] exploited a knowledge base to identify related words in different sentences. The proposed measure was tested in natural language

**Table 1** Examples from the training collections.

| Collection | Id | Pair | Sim |
|---|---|---|---|
| PTPT | 2675 | **t:** *O Chelsea só conseguiu reagir no final da primeira parte.*<br>**h:** *Não podemos aceitar outra primeira parte como essa.* | 1.25 |
| PTPT | 315 | **t:** *Todos que ficaram feridos e os mortos foram levados ao hospital.*<br>**h:** *Além disso, mais de 180 pessoas ficaram feridas.* | 3.00 |
| PTBR | 1282 | **t:** *As multas previstas nos contratos podem atingir, juntas, 23 milhões de reais.*<br>**h:** *Somadas, as multas previstas nos contratos podem chegar a R$ 23 milhões.* | 5.00 |

descriptions of bugs in software engineering projects, which had their similarity annotated by two humans. But it was not until 2015 that a collection was publicly released for computing STS in Portuguese, with the goal of being used in the ASSIN shared task [12], which targeted Semantic Similarity and Textual Entailment in Portuguese. Training data comprised 3,000 sentence pairs for PTBR, and another 3,000 for PTPT. Test data comprised 2,000 PTBR pairs and 2,000 PTPT pairs. While recent editions of English STS have used text from varied sources, sentences in the ASSIN collections were obtained exclusively from Google News. Table 1 shows three pairs in the ASSIN training collections, including ids, sentences ($t$, for text, $h$, for hypothesis), and the average similarity given by four human judges that followed the same guidelines. Similarity values range from 1 (completely different sentences, on different subjects) to 5 (sentences mean essentially the same).

ASSIN had 6 participating teams, which submitted 14 runs for the STS task in PTBR and 17 in PTPT. Distinct systems achieved the best official results for PTPT and PTBR. For PTBR, the best run [17] achieved $\rho = 0.70$ with $MSE = 0.38$, obtained by computing the cosine similarity of a vector representation of each sentence, based on the sum of the TF-IDF scores and word2vec [25] vectors of each word. For PTPT, the best run [11] achieved $\rho = 0.73$ with $MSE = 0.61$, obtained after learning a similarity function with a Kernel Ridge Regression using several similarity metrics as input, computed between the two sentences of each pair, including overlap and set similarity measures on multiple text representations (e.g., lowercase, character trigrams). ASAPP [4], an adaptation of ASAP to Portuguese, also participated, with best runs achieving $\rho = 0.65$ and $MSE = 0.44$, for PTBR, and $\rho = 0.68$ and $MSE = 0.70$ for PTPT.

As the collections of ASSIN are available[1], work on Portuguese STS continued, even after the evaluation, using those collections as benchmarks. This included our previous work [14], where we report on gradual improvements as features and techniques are added, though without outperforming the best results. An important conclusion was that the best test results ($\rho = 0.711$ for PTPT, and $\rho = 0.697$ for PTBR) were obtained after training the model on both the PTPT and PTBR collections. But other recent works tackled Portuguese STS and relied on the ASSIN collection for evaluation [18, 7]. Hartmann et al. [18] tested a broad range of distributional similarity models of Portuguese (word embeddings) for different NLP tasks, including STS on the ASSIN collection. The best results obtained are quite low ($\rho = 0.60$ for PTBR, using Wang2vec skip-gram with 1,000 dimensions; $\rho = 0.55$ for PTPT, using word2vec CBOW with 600 and 1,000 dimensions), but their main goal was to compare the models, also developed by them. Their results suggest that relying on a single feature, even on large quantities of data, or on a small set of features of the same kind is not

[1] http://nilc.icmc.usp.br/assin/

enough to achieve high STS scores. Cavalcanti et al. [7] used a regression algorithm that exploited four features: the cosine similarity between the sentence vectors weighted with TF-IDF; word2vec similarity based on a three-layer sentence representation; word overlap; length of the shortest sentence divided by the length of the longest. They outperformed the best results for PTBR, with $\rho = 0.71$ and $MSE = 0.37$, and achieved $\rho = 0.70$ and $MSE = 0.57$, for PTPT.

## 3 Features for Semantic Textual Similarity in Portuguese

In order to compute their semantic similarity, several features are extracted from the ASSIN sentence pairs. A broad range of features was exploited, including lexical, syntactic, semantic and distributional features. Some were already used in previous versions of ASAPP [4, 14], but new features are new in ASAPPV2.0, namely the dependency-based and distributional features. Although these were later used to learn a model of STS, in the end of this section, we reveal a selection of unsupervised results, obtained with some subsets of related features.

Several features were extracted with the NLPPort [28] tools, developed in our group and freely available[2]. Those include TokPORT, a tokenizer; TagPORT, a part-of-speech tagger; ChkPORT, a syntactic chunker; LemPORT, a lemmatizer; and EntPORT, a named entity recognizer. In addition, PTStemmer[3] was used for obtaining the stems of each token. To acquire the semantic features, we resort to a set of Portuguese lexical knowledge bases (LKBs), enumerated in Section 3.3. Syntactic dependencies and distributional features were extracted with the spaCy toolkit[4].

### 3.1 Lexical Features

The following lexical features, related to words at the surface level, were exploited:

- Number of common tokens, after tokenization with TokPort.
- Number of negation words (*não*, *nada*, *nenhum*, *de modo algum*, . . . ) in each sentence of the pair and their absolute difference.
- Number of common lemmas, obtained with LemPORT.
- Number of common stems, obtained with PTStemmer.

Set similarity metrics were computed to devise their integration in the feature set. Those metrics included the Jaccard, Overlap, Dice coefficient, plus the Cosine Similarity, computed according to equations 1, 2, 3, 4, respectively, for the sets of the tokens, lemmas and stems, in each sentence of the pair ($T$ and $H$).

$$Jaccard(T, H) = \frac{|T \cap H|}{|T \cup H|} \quad (1)$$

$$Overlap(T, H) = \frac{|T \cap H|}{|min(T, H)|} \quad (2)$$

$$Dice(T, H) = \frac{|T \cap H|}{|T| + |H|} \quad (3)$$

$$Cos(T, H) = \frac{|T \cap H|}{\sqrt{|T|}\sqrt{|H|}} \quad (4)$$

[2] NLPPort is available from `https://github.com/rikarudo/`
[3] PTStemmer is available from `https://code.google.com/archive/p/ptstemmer/`
[4] `https://spacy.io/`

**(t)** *Ricky Álvarez voltou hoje, dia 17 de Setembro, a ser associado à equipa do Futebol Clube do Porto.*
**NP:** [Ricky Álvarez], [17 de Setembro], [a equipa], [o Futebol Clube do Porto]
**VP:** [voltou], [dia]*, [ser associado]
**PP:** [a], [a], [de]
**ADVP:** [hoje]

**(h)** *Nas suas três temporadas na equipa de Milão, Ricky participou em 90 jogos e apontou 14 golos.*
**NP:** [as suas três temporadas], [a equipa], [Milão], [Ricky], [90 jogos], [14 golos]
**VP:** [participou], [apontou]
**PP:** [em], [em], [de], [em]

| **Non-Zero Features** | **values** | | |
|---|---|---|---|
| | t | h | \|#t - #h\| |
| # NP | 4 | 6 | 2 |
| # VP | 3 | 2 | 1 |
| # PP | 3 | 4 | 1 |
| # ADVP | 1 | 0 | 1 |

**Figure 1** Extraction of noun, verbal, propositional and adverbial phrases and related features.

## 3.2 Syntactic Features

The set of syntactic features exploited included the number of noun, verb, prepositional and adverbial phrases in each sentence of the pair and their absolute difference. Figure 1 illustrates the chunk-based features, with their computation in a pair of sentences.

In ASAPPv2.0, syntactic dependencies are also exploited, namely the Jaccard coefficient between the dependencies in the first sentence of the pair and those in the second sentence. Syntactic dependencies were computed with spaCy's dependency parser. Each sentence is converted to a list of triples related to the arcs in the dependency tree, ignoring just the punctuation tokens. Each triple – $(token_1, token_2, DEPENDENCY)$ – contains two connected tokens (head and child) and the syntactic dependency name that labels the relation. The Jaccard similarity of the two lists is then computed to measure the similarity of the pair of sentences, as in equation 5. The computation of the previous feature is illustrated in Figure 2.

$$Jaccard\_Dep(T,H) = \frac{|Dep(T) \cap Dep(H)|}{|Dep(T) \cup Dep(H)|} \tag{5}$$

When computed with this feature, alone, semantic similarity is poor, but it captures some relations that are not covered by the other features used in previous versions of ASAPP. Namely, it aims at capturing the dissimilarity between sentences such as: {"The tiger killed the man.", "The main killed the tiger"}. Besides their strong overlapping and exact matches of noun phrases and verbal phrases, their outcome is significantly different.

## 3.3 Semantic Features

Given not only their importance for understanding the meaning of a sentence, but also their frequent presence in the ASSIN collection, named entities in the sentences of the pair were extracted and classified into one of nine types (abstraction, product, event, number, organization, person, place, thing and time). The number of named entities of each type in the sentences is used as features, plus their absolute difference, which makes a total of 27 features. Figure 3 illustrates the computation of those features.

**(t)** *Sebastian Vettel garantiu a pole-position para o Grande Prémio de Singapura de Fórmula 1.*

- (‘Sebastian’, ‘Vettel’, FLAT:NAME), (‘garantiu’, ‘Sebastian’, NSUBJ), (‘garantiu’, ‘pole’, OBJ),
- (‘pole’, ‘a’, DET), (‘pole’, ‘position’, APPOS), ( ‘postion’, ‘Grande’, NMOD)
- (‘Grande’, ‘para’, CASE), **(‘Grande’, ‘o’, DET), (‘Grande’, ‘Prémio’, FLAT:NAME), (‘Grande’, ‘Singapura’, NMOD)**
- **(‘Singapura’, ‘de’, CASE), (‘Singapura’, ‘Fórmula’, NMOD)**
- **(‘Fórmula’, ‘de’, CASE), (‘Fórmula’, ‘1’, FLAT:NAME)**

**(h)** *O Grande Prémio de Singapura de Fórmula 1 tem início marcado para as 13h00 de domingo.*

- **(‘Grande’, ‘o’, DET), (‘Grande’, ‘Prémio’, FLAT:NAME), (‘Grande’, ‘Singapura’, NMOD)**
- **(‘Singapura’, ‘de’, CASE), (‘Singapura’, ‘Fórmula’, NMOD)**
- **(‘Fórmula’, ‘de’, CASE), (‘Fórmula’, ‘1’, FLAT:NAME)**
- (‘tem’, ‘Grande’, NSUBJ), (‘tem’, ’início’, OBJ), (‘início’, ‘marcardo’, ACL), (‘marcardo’, ‘ 13h00’, OBL)
- (‘13h00’, ‘para’, CASE), (‘13h00’, ‘as’, DET), (‘13h00’, ‘domingo’, NMOD, (‘doming’, ‘de’, CASE)

$$Jaccard\_Dep(T,H) = \frac{7}{22} \approx 0.3182$$

**Figure 2** Extraction of syntactic dependencies with the spaCy toolkit and its related feature.

Language is flexible in such a way that the same idea can be transmitted through different words, generally related by well-known semantic relations, such as synonymy or hypernymy. These relations are implicitly mentioned in dictionaries, and explicitly encoded in LKBs, such as WordNet [10]. We decided to use LKBs currently available for Portuguese, namely three wordnets – WordNet.Br [9] (which covers only verbs), OpenWordNet-PT (OWN.PT) [26] and PULO [30]; two synonymy-based thesauri – TeP [24] and OpenThesaurus.PT[5]; three lexical networks extracted from Portuguese dictionaries – PAPEL [15] and relations from Dicionário Aberto [31] and Wiktionary.PT[6]; and the semantic relations in a set of linguistic resources – Port4Nooj [6]. All of these LKBs cover synonymy relations (e.g., *realçar* synonym-of *sublinhar*) , all but OpenThesaurus.PT, WordNet.Br, and Port4Nooj cover antonymy (e.g., *tristeza* antonym-of *alegria*) , all but TeP and OT cover hypernymy relations (e.g., *mover* hypernym-of *tremer*) , in addition to relations of other types, covered only by some LKBs, such as part-of (e.g., *núcleo* part-of *átomo*), causation (e.g., *frio* causation-of *crestar*) , or purpose (e.g., *polir* purpose-of *lixa*) , among others.

The aforementioned LKBs have substantially different sizes and the creation of most involved some degree of automation, which means that they contain noise, including rarely used words and meanings, not so useful relations, and also actual errors. Therefore, we rely on redundancy to build more reliable and useful semantic networks [13], namely *Redun2* and *Redun3*, which include all the relation instances respectively in at least two or three LKBs. They were exploited in different ways (see Section 3.5), but the final model only considered the following features:

- Set similarity metrics considering semantic relations in *Redun3* LKB: after computing the overlap of the similarity of the stems, the metrics were adjusted as in equation 6.

[5] `http://paginas.fe.up.pt/~arocha/AED1/0607/trabalhos/thesaurus.txt`

[6] `http://pt.wiktionary.org`

**(t)** *Ricky Álvarez voltou hoje, dia 17 de Setembro, a ser associado à equipa do Futebol Clube do Porto.*
**Person:** [Ricky Álvarez]
**Time:** [17 de Setembro]
**Org:** [Futebol Clube do Porto]

**(h)** *Nas suas três temporadas na equipa de Milão, Ricky participou em 90 jogos e apontou 14 golos.*
**Person:** [Ricky]
**Place:** [Milão]
**Numeric:** [90], [14]
**Event:** [jogos]

| **Non-Zero Features** | **values** | | |
|---|---|---|---|
| | t | h | \|#t - #h\| |
| # Person | 1 | 1 | 0 |
| # Time | 1 | 0 | 1 |
| # Organization | 1 | 0 | 1 |
| # Place | 0 | 1 | 1 |
| # Numeric | 0 | 2 | 2 |
| # Event | 0 | 1 | 1 |

**Figure 3** Extraction of named entities and related features.

There, $\gamma$ was set according to equation 7 and $Sim$ was computed according to equation 8. Constants were empirically set to $\alpha = 0.75$ and $\beta = 0.05$.

$$Jaccard^{+}(T, H) = \frac{|T \cap H| + \gamma}{|T \cup H|} \tag{6}$$

$$\gamma = \sum_{i=1}^{|T'|} \sum_{j=1}^{|H'|} Max(Sim(T'_i, H'_j)) \quad (7) \qquad Sim(T'_i, H'_j) = \begin{cases} \alpha, & \text{if } dist(T'_i, H'_j) = 1 \\ \beta, & \text{if } dist(T'_i, H'_j) = 2 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

- A feature for each of four relation groups considered (synonymy, hypernymy, antonymy and other) in each LKB, which would be the number of semantic relations of those types held, in the target LKB, between lemmas in one sentence of the pair and lemmas in the other, normalized after division by the sum of the number of open-class words (nouns, verbs, adjectives and adverbs) in sentences $t$ and $h$. Although these features would clearly not be enough for computing similarity all alone, our belief is that they would be a useful complement to the other.

Figure 4 illustrates the computation of the Jaccard$^{+}$ feature with the *Redun3* network.

## 3.4 Distributional Features

In ASAPPv2.0, distributional features are also exploited, namely word and character n-gram distribution, and models of distributional similarity. For convenience reasons, these features were extracted with Python tools, in opposition to all the others, extracted with tools in Java.

Set similarity features already covered the similarity of n-grams of size 1. Yet, the new features considered n-grams of size 2 with additional restrictions. Character n-grams were also considered, as they are known for capturing features at different levels, and can be

Sentences:

***(t)*** *Os Estados Unidos anunciaram oficialmente esta sexta-feira o abandono de um plano que visava treinar e equipar rebeldes na Síria.*

***(h)*** *Os Estados Unidos anunciaram esta sexta-feira que interrompe o projeto de treino de rebeldes sírios.*

Relations in *Redun3* connecting words in $t$ with words in $h$:

- *plano* synonym-of *projeto*
- *Síria* place-of *sírio*

- Synonymy (1): $\frac{1}{|openClassWords(t)+|openClassWords(h)|}$
- Other relations (1): $\frac{1}{|openClassWords(t)+|openClassWords(h)|}$

$$Jaccard^{+}(T,H) = \frac{lemmas(t) \cap lemmas(h) + \alpha + \beta}{lemmas(t) \cup lemmas(h)} = \frac{9 + 0.75 + 0.05}{23} \approx 0.39$$

**Figure 4** Computation of Jaccard$^{+}$ feature with Reund3 as the semantic network.

extremely useful in morphologically-rich languages, such as Portuguese. Among other text classification tasks, character n-grams revealed to be successful in author attribution [21]. This adds to the simplified pre-processing steps, which require no specific tools or detailed linguistic knowledge.

The exploitation of n-grams resulted in three features – NG1, NG2, NG3 – obtained after computing the cosine similarity of two vectors containing the following information:

- **NG1**: vectors with the binary term-frequency (TF) in which the vocabulary corresponds to the set of n-grams of words, with $n \in \{1, 2\}$, in lowercase, stemmed with the Portuguese RSLP stemmer available in the NTLK toolkit[7], after removing stopwords, and considering only n-grams that occur in more than one sentence (document_frequency $> 1$).

- **NG2**: vector with binary TF for character n-grams, with $n \in \{1, 3\}$, within the limits of word boundaries, in lowercase, and considering only n-grams that occur in more than one sentence (document_frequency $> 1$) and a maximum of 50% of the sentences (max_document_frequency $= 0.5 \times \#Dataset\_Sentences$).

- **NG3**: vectors with binary TFs for char n-grams, with $n \in \{1, 3\}$, not considering word boundaries, in lowercase, and considering only n-grams that occur in more than one document (document_frequency $> 1$) and a maximum of 40% of documents (max_document_frequency $= 0.4 \times \#Dataset\_Sentences$).

We also followed the current trend of using word embeddings, learned from a large corpus with a neural network, in semantic similarity tasks. For this purpose, we resorted to the NILC embeddings [18], which offer a wide variety of pre-trained embeddings, learned with different models in a large Portuguese corpus, and freely available[8].

[7] RSLP stemmer available from `http://www.nltk.org/_modules/nltk/stem/rslp.html`

[8] NILC embeddings available from `http://nilc.icmc.usp.br/embeddings`

More precisely, two different features were computed from the embeddings, both after the conversion of each sentence into a vector computed from the vectors of its tokens. The difference occurs on how this vector is created.

- In the first feature, the sentence vector is obtained from the sum of the token vectors;
- In the second feature, it is computed from the weighted sum of the token vectors, using the TF-IDF value of each token as the weight.

In both, the similarity of each pair of sentences is computed as the cosine similarity between their vectors.

The previous features were initially extracted using different embeddings. The results of using only these features were analysed (see Section 3.5), and only one model of embeddings was used in the final set of features.

## 3.5 Unsupervised Results

Before moving to a supervised approach, the correlation of some of the extracted features was computed when used alone to predict STS. This would give valuable hints on the relevance of each feature and, at the same time, set baselines. Three groups of features were tested: (i) set similarity combined with semantic networks; (ii) word embeddings; and (iii) n-grams. While the first group has been used since our first approach to ASSIN [4], the second and third were only recently added to our feature set.

### 3.5.1 Set similarity and LKBs

First, all the combinations of set similarity features were tested in the training collections, with different kinds of normalization (none, stemming and lemmatisation). The previous measures were then tested when combined with the semantic relations in each of the exploited LKBs, as described in Section 3.3. Table 2 shows a selection of the best results at this stage. The best results with Jaccard$^+$ and Cosine$^+$ were obtained with the *Redun3* LKB, and are presented here. Additional results can be found in our previous approach [14], where we also concluded that using all words, instead of just open-class, and not requiring a match of parts-of-speech would improve the correlation $\rho$. Another conclusion was that stemming would lead to better results than lemmatisation and that using the LKBs could lead only to minor improvements.

Based on those conclusions, the selection of approaches to use in the test collections was narrowed to two, Cosine$^+$ and Jaccard$^+$, on *Redun3*, computed after stemming. Their results are presented in table 3, together with a baseline that computes the cosine of the stems in both sentences of the pair. It is worth noticing that Cosine$^+$ would be the fifth and third best run in ASSIN, respectively for PTPT and PTBR, which corresponds to the fourth and second best system.

### 3.5.2 N-grams

All the three n-gram features were tested, first in the training, then on the test collection. Results, presented in table 4, are quite surprising, especially for the character n-grams (features NG2 and NG3). The power of these features, even when used alone, would result in technical ties with the second best run for PTPT and with the best run for PTBR, in the official evaluation. As mentioned earlier, character n-grams carry a mix of lexical, syntactic, and even author style content. Without any normalization, different forms of the same word are considered completely different tokens. This is often solved with stemming, which ends up

**Table 2** Set similarity results in the training collections.

| Normalization | Measure | PTPT | | PTBR | |
|---|---|---|---|---|---|
| | | $\rho$ | MSE | $\rho$ | MSE |
| None | Jaccard | 0.661 | 1.220 | 0.587 | 1.168 |
| None | Cosine | 0.664 | 0.552 | 0.587 | 0.591 |
| Stems | Jaccard | 0.700 | 1.140 | 0.625 | 0.853 |
| Stems | Cosine | **0.706** | **0.443** | **0.626** | **0.467** |
| Lemmas | Jaccard | 0.695 | 1.131 | 0.610 | 0.921 |
| Lemmas | Cosine | 0.698 | 0.446 | 0.610 | 0.484 |
| Stems | Jaccard$^+$ | 0.717 | 1.049 | 0.632 | 0.778 |
| Stems | Cosine$^+$ | **0.721** | **0.388** | **0.631** | **0.453** |
| Lemmas | Jaccard$^+$ | 0.709 | 1.116 | 0.621 | 0.843 |
| Lemmas | Cosine$^+$ | 0.712 | 0.431 | 0.620 | 0.464 |

**Table 3** Test results when semantic networks are exploited, plus the Cosine baseline.

| Normalization | Measure | PTPT | | PT-PBR | |
|---|---|---|---|---|---|
| | | $\rho$ | MSE | $\rho$ | MSE |
| Stems | Jaccard$^+$ | 0.669 | 0.723 | 0.666 | 0.825 |
| Stems | Cosine$^+$ | **0.677** | **0.686** | **0.667** | 0.454 |
| *(baseline)* Stems | Cosine | 0.656 | 0.658 | 0.653 | **0.445** |

**Table 4** Results for n-gram features in the training and test collections.

| Features | Train | | | | Test | | | |
|---|---|---|---|---|---|---|---|---|
| | PTPT | | PTBR | | PTPT | | PTBR | |
| | $\rho$ | MSE | $\rho$ | MSE | $\rho$ | MSE | $\rho$ | MSE |
| NG1 | 0.664 | 0.470 | 0.580 | 0.537 | 0.600 | 0.748 | 0.600 | 0.514 |
| NG2 | **0.743** | **0.395** | 0.685 | 0.429 | **0.696** | 0.608 | **0.696** | **0.425** |
| NG3 | **0.743** | 0.454 | **0.688** | **0.483** | 0.699 | **0.597** | 0.695 | 0.488 |

considering them equal. So, character n-grams provide a more precise representation of word proximity, because the sets of n-grams for forms of the same word have much in common, but are not equal.

### 3.5.3 Word embeddings

Though there are many NILC embeddings, we compared only those with 300-sized vectors, a commonly used dimension. As mentioned earlier, two different features were extracted, one relying on the TF to compute the sentence vectors, and another relying on TF-IDF for the same purpose. Table 5 shows the results when using only these features, with the different tested embeddings.

From those results, we decided to use only the word2vec CBOW model, which got the best $\rho$ in the training collection of PTPT, using TF, and the lowest MSE in all the other collections, with TF and TF-IDF. Another option would have been to select fastText SKIP-GRAM,

**Table 5** Results for embedding features in the training collections.

| Model | Weights | PTPT | | PTBR | |
|---|---|---|---|---|---|
| | | $\rho$ | MSE | $\rho$ | MSE |
| word2vec CBOW | TF | **0.592** | **0.614** | 0.511 | **0.755** |
| word2vec SKIP-GRAM | TF | 0.551 | 1.379 | 0.491 | 2.195 |
| fastText CBOW | TF | 0.408 | 1.400 | 0.350 | 1.570 |
| fastText SKIP-GRAM | TF | 0.566 | 2.647 | **0.521** | 2.830 |
| Wang2vec CBOW | TF | 0.557 | 2.336 | 0.511 | 2.484 |
| Wang2vec SKIP-GRAM | TF | 0.559 | 2.625 | 0.508 | 2.784 |
| GloVe | TF | 0.507 | 2.076 | 0.444 | 2.299 |
| word2vec CBOW | TF-IDF | 0.609 | **0.572** | 0.550 | **0.682** |
| word2vec SKIP-GRAM | TF-IDF | 0.587 | 1.073 | 0.524 | 1.266 |
| fastText CBOW | TF-IDF | 0.451 | 1.649 | 0.402 | 1.772 |
| fastText SKIP-GRAM | TF-IDF | **0.639** | 2.393 | **0.591** | 2.526 |
| Wang2vec CBOW | TF-IDF | 0.626 | 1.951 | 0.578 | 2.054 |
| Wang2vec SKIP-GRAM | TF-IDF | 0.622 | 2.280 | 0.577 | 2.382 |
| GloVe | TF-IDF | 0.528 | 1.871 | 0.502 | 2.036 |

**Table 6** Results for embedding features in the test collections.

| Model | Weights | PTPT | | PTBR | |
|---|---|---|---|---|---|
| | | $\rho$ | MSE | $\rho$ | MSE |
| word2vec CBOW | TF | 0.548 | 1.125 | 0.538 | 0.749 |
| word2vec CBOW | TF-IDF | 0.555 | 1.072 | 0.572 | 0.665 |

but the results of this model has always a very high MSE. Table 6 shows the results of the selected model in the test collections.

These results are not much different from those by Hartmann et al. [18], who used all the NILC embeddings in the test collections of ASSIN. It should also be noted that they are lower than all the other unsupervised results here, which shows that, when it comes to STS, this kind of embeddings alone are definitely not enough for achieving high results. Alternative ways for representing sentences as distributional vectors have to be devised in the future. Nevertheless, these two features were included in our feature set, where, combined with the others, should have a positive impact.

## 4 Learning a model for Portuguese STS

For improving the unsupervised results, the selected features were used together to learn a STS function from each training collection and, later, from both. Here, we describe the learning algorithms used, and report on the training and test results achieved, which beat the best performances in ASSIN to date, thus setting the state-of-the-art of Portuguese STS.

**Table 7** Weka setup for the three learning algorithms used.

| |
|---|
| M5Rules<br>`weka.classifiers.rules.M5Rules -M 4.0` |
| RandomSubspace w/ M5<br>`weka.classifiers.meta.RandomSubSpace -P 0.5 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.M5P - -M 4.0` |
| Gaussian Process w/ RBF Kernel<br>`weka.classifiers.functions.GaussianProcesses -L 1.0 -N 0 -K "weka.classifiers.functions.supportVector.RBFKernel -G 0.01 -C 250007"` |

## 4.1 Regression Algorithms

Several regression algorithms, provided by the Weka [16] machine learning toolkit, were selected to learn a STS function. Table 7 presents the setup of the three best-performing algorithms, after an exhaustive set of runs. The used algorithms are:

- *M5Rules* [20] generates a decision list for regression problems using a separate-and-conquer strategy. In each iteration, it builds a model tree using the M5 algorithm and turns the "best" leaf into a rule.
- *Random Subspace* [19] is an ensemble learning algorithm that builds a decision tree classifier. It consists of random subspacing regression ensembles composed of multiple trees constructed systematically by pseudo-randomly selected subsets of components of the feature vector.
- Regression algorithm based on *Gaussian Processes* [22], with a Radial Basis Function (RBF) Kernel as the Gaussian function. This implementation is simplified in Weka: it does not apply hyper-parameter-tuning and uses normalization to the target class (similarity value), so the features simplify the choice of a noise level.

## 4.2 Training and Testing

Each of the selected regression algorithms was used for learning two STS models, for PTPT and for PTBR, based on the respective training collections. Table 8 shows the average training performance with the current set of features (v2) for the three regression algorithms, in a 10-fold cross validation, for PTPT and PTBR. When compared to our previous results (v1.5) [14], in the same table, there are improvements in training.

The learned models were then used for computing STS in the respective test collections. Table 9 shows the test results of the new models, again side-by-side with our previous results, and also with the systems that achieved the best official results, for PTBR and PTPT, in ASSIN, respectively Solo Queue [17] and L2F/INESC-ID [11]. Our current results are clearly better than our best unsupervised results and also than our previous results, which means that the new features had a positive impact. Furthermore, when compared to the best official ASSIN results, there are also improvements in $\rho$ and MSE. More precisely, for PTPT, $\rho$ is 0.02 points higher and MSE is 0.03 points lower than the best, and, for PTBR, $\rho$ is 0.04 higher and MSE is 0.03 points lower. We can thus see these results as the new state-of-the-art of Portuguese STS.

**Table 8** Performance when training in the PTPT and PTBR collections between previous (v1) and present (v2) ASAPP systems.

| Method | | PTPT | | PTBR | |
|---|---|---|---|---|---|
| | | $\rho$ | MSE | $\rho$ | MSE |
| v1.5 | M5Rules | 0.742 | 0.472 | 0.657 | 0.518 |
| | RandomSubspace | **0.756** | **0.457** | **0.662** | **0.515** |
| | GaussianProcess | 0.739 | 0.479 | 0.658 | 0.520 |
| v2 | M5Rules | 0.778 | 0.440 | 0.723 | 0.480 |
| | RandomSubspace | **0.784** | **0.432** | **0.723** | **0.479** |
| | GaussianProcess | 0.776 | 0.444 | 0.722 | 0.481 |

**Table 9** Test results for models trained in the respective training collection compared with the state-of-the-art systems.

| Method | | PTPT | | PTBR | |
|---|---|---|---|---|---|
| | | $\rho$ | MSE | $\rho$ | MSE |
| v1.5 | M5Rules | 0.703 | 0.714 | 0.678 | 0.411 |
| | RandomSubspace | **0.709** | **0.698** | **0.686** | **0.403** |
| | GaussianProcess | 0.694 | 0.725 | 0.683 | 0.406 |
| v2 | M5Rules | 0.740 | 0.590 | 0.730 | 0.350 |
| | RandomSubspace | **0.750** | **0.580** | **0.740** | **0.350** |
| | GaussianProcess | 0.740 | 0.620 | 0.730 | 0.350 |
| Solo Queue [17] | | 0.700 | 0.660 | **0.700** | **0.380** |
| L2F/INESC-ID [11] | | **0.730** | **0.610** | - | - |

### 4.3 Training on both collections

Since they are just variants of the same language, instead of training independent models for PTPT and PTBR, we concatenated the training collections and learned new (variant-ignoring) models from the resulting larger collection, which comprised 6,000 pairs. Tables 10 and 11 show, respectively, the training performance of the same learning algorithms on a 10-fold cross-validation in the larger collection, and the results of the new models in each test collection. These are compared with the best official results in ASSIN.

Although with our previous feature set (v1.5) training with a single collection lead to improvements, with the current set, $\rho$ was similar to the one obtained with a collection trained for each variant. Only MSE was lower. Still, this shows that a single model could be used for computing STS in the PTPT and PTBR collection.

## 5 Concluding Remarks

We have described the most recent developments on ASAPP. In addition to features used in our previous work, which already considered the presence of negations, token, lemma, stem, chunk and named entity overlap, plus semantic relations, new features were added: syntactic dependencies, word and character n-gram similarity, and distributional similarity.

**Table 10** Training performance in a collection with both PTPT and PTBR training pairs between previous and present ASAPP systems.

| Method | | $\rho$ | MSE |
|---|---|---|---|
| v1.5 | M5Rules | 0.705 | 0.493 |
| | RandomSubspace | **0.713** | **0.486** |
| | GaussianProcess | 0.701 | 0.493 |
| v2 | M5Rules | 0.756 | 0.456 |
| | RandomSubspace | **0.760** | **0.451** |
| | GaussianProcess | 0.754 | 0.459 |

**Table 11** Test results for models trained with both PTPT and PTBR training pairs compared with the state-of-the-art systems.

| Method | | PTPT | | PTBR | |
|---|---|---|---|---|---|
| | | $\rho$ | MSE | $\rho$ | MSE |
| v1.5 | M5Rules | 0.702 | 0.648 | 0.690 | 0.505 |
| | RandomSubspace | **0.711** | **0.657** | **0.697** | **0.499** |
| | GaussianProcess | 0.691 | 0.678 | 0.684 | 0.509 |
| v2 | M5Rules | 0.740 | 0.540 | 0.730 | 0.350 |
| | RandomSubspace | **0.750** | **0.540** | **0.740** | **0.340** |
| | GaussianProcess | 0.740 | 0.560 | 0.730 | 0.350 |
| Solo Queue | | 0.700 | 0.660 | **0.700** | **0.380** |
| L2F/INESC-ID | | **0.730** | **0.610** | - | - |

Interesting results can be achieved with some of the previous features alone, where we highlight the good performance of character n-grams. Yet, the best results were obtained using all the previous features to learn a STS function from the training collections of ASSIN. Three different regression algorithms were tested for this purpose, and all outperformed the best official results of ASSIN – Pearson $\rho$ of 0.75 and 0.74, MSE of 0.54 and 0.34, respectively for European and Brazilian Portuguese. This means that we can see the approach reported here as the current state-of-the-art of Portuguese STS. Moreover, we have confirmed that a single model, learned from training collections in both variants, obtains very similar results than two different models, each trained and tested on a variant-dependent collection.

Given the Pearson $\rho$ of the human annotation of the ASSIN collections [12] – 0.74 – , trying to improve these results further is probably unrealistic, and possibly not very useful. Nevertheless, there is work to do, especially regarding an analysis of feature relevance, and the integration of all features in a single pipeline, which, until this point, was not our main goal. Although some experiments were reported with each feature alone, this analysis is harder when all the features are combined. For this purpose, the correlation between the features and the similarity scores could be computed to analyse feature relevance; a method such as Principal Component Analysis (PCA) could be applied for feature reduction; and, when possible, the STS functions obtained with the regression algorithms, and the included weights, should be analysed. Identifying the most relevant features should be especially useful for learning more about Portuguese STS and would help us on the integration of all feature extraction methods, hopefully only the most relevant, in a single pipeline.

It should also be stressed that the reported results were obtained in the ASSIN collection. As far as we know, there is currently no other collection with the same kind of annotations in Portuguese, at least freely available and with similar size. In the future, it would be important to test our approach in different collections of Portuguese sentences with STS scores.

## References

1 Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *10th Intl. Workshop on Semantic Evaluation (SemEval)*, pages 497–511, 2016.

2 Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. Semeval-2012 task 6: A pilot on semantic textual similarity. In *6th Intl. Workshop on Semantic Evaluation*, pages 385–393, 2012.

3 Ana Alves, Adriana Ferrugento, Mariana Lourenço, and Filipe Rodrigues. ASAP: Automatic semantic alignment for phrases. In *8th Intl. Workshop on Semantic Evaluation (SemEval)*, pages 104–108, 2014.

4 Ana Alves, Ricardo Rodrigues, and Hugo Gonçalo Oliveira. Asapp: Alinhamento semântico automático de palavras aplicado ao português. *Linguamática*, 8(2):43–58, 2016.

5 Ana Alves, David Simões, Hugo Gonçalo Oliveira, and Adriana Ferrugento. ASAP-II: From the alignment of phrases to textual similarity. In *9th Intl. Workshop on Semantic Evaluation (SemEval 2015)*, pages 184–189, 2015.

6 Anabela Barreiro. Port4NooJ: an open source, ontology-driven portuguese linguistic system with applications in machine translation. In *Intl. NooJ Conference (NooJ'08)*, 2010.

7 Anderson Pinheiro Cavalcanti, Rafael Ferreira Leite de Mello, Máverick André Dionísio Ferreira, Vitor Belarmino Rolim, and João Vitor Soares Tenório. Statistical and semantic features to measure sentence similarity in Portuguese. In *Proceedings of 6th Brazilian Conference on Intelligent Systems*, pages 342–347, 2017.

8 Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *11th Intl. Workshop on Semantic Evaluation (SemEval)*, pages 1–14, 2017. doi:10.18653/v1/S17-2001.

9 Bento C. Dias-da-Silva. Wordnet.Br: An exercise of human language technology research. In *3rd Intl. WordNet Conf. (GWC)*, pages 301–303, 2006.

10 Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, 1998.

11 Pedro Fialho, Ricardo Marques, Bruno Martins, Luísa Coheur, and Paulo Quaresma. INESC-ID@ASSIN: Medição de similaridade semântica e reconhecimento de inferência textual. *Linguamática*, 8(2):33–42, 2016.

12 Erick Fonseca, Leandro Santos, Marcelo Criscuolo, and Sandra Aluísio. Visão geral da avaliação de similaridade semântica e inferência textual. *Linguamática*, 8(2):3–13, 2016.

13 Hugo Gonçalo Oliveira. Comparing and combining Portuguese lexical-semantic knowledge bases. In *6$^{th}$ Symposium on Languages, Applications and Technologies (SLATE)*, pages 16:1–16:14, 2017.

14 Hugo Gonçalo Oliveira, Ana Oliveira Alves, and Ricardo Rodrigues. Gradually improving the computation of semantic textual similarity in Portuguese. In *18th EPIA Conference on Artificial Intelligence*, volume 10423, pages 841–854, 2017. doi:10.1007/978-3-319-65340-2_68.

15 Hugo Gonçalo Oliveira, Diana Santos, Paulo Gomes, and Nuno Seco. PAPEL: A dictionary-based lexical ontology for Portuguese. In *8th Intl. Conf. Computational Processing of the Portuguese Language (PROPOR)*, volume 5190, pages 31–40, 2008.

16 Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18, 2009. doi:10.1145/1656274.1656278.

17 Nathan Hartmann. Solo Queue at ASSIN: Combinando abordagens tradicionais e emergentes. *Linguamática*, 8(2):59–64, 2016.

18 Nathan S. Hartmann, Erick R. Fonseca, Christopher D. Shulby, Marcos V. Treviso, Jéssica S. Rodrigues, and Sandra M. Aluísio. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In *11th Brazilian Symposium in Information and Human Language Technology (STIL)*, 2017.

19 Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

20 Geoffrey Holmes, Mark Hall, and Eibe Frank. Generating rule sets from model trees. In *12th Australian Joint Conf. on Artificial Intelligence*, pages 1–12, 1999.

21 Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Authorship attribution in the wild. *Languages Resourses Evaluation*, 45(1):83–94, 2011. doi:10.1007/s10579-009-9111-2.

22 David Mackay. Introduction to Gaussian Processes. In *Neural Networks and Machine Learning*. Springer, 1998.

23 Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *8th Intl. Workshop on Semantic Evaluation (SemEval)*, pages 1–8, 2014.

24 Erick Maziero, Thiago Pardo, Ariani Felippo, and Bento Dias-da-Silva. A Base de Dados Lexical e a Interface Web do TeP 2.0 - Thesaurus Eletrônico para o Português do Brasil. In *VI Workshop em Tecnologia da Informação e da Linguagem Humana (TIL)*, pages 390–392, 2008.

25 Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Workshop track of the Intl. Conf. on Learning Representations (ICLR)*, 2013.

26 Valeria Paiva, Alexandre Rademaker, and Gerard Melo. OpenWordNet-PT: An open Brazilian Wordnet for reasoning. In *24th Intl. Conf. on Computational Linguistics (COLING)*, 2012.

27 Vladia Pinheiro, Vasco Furtado, and Adriano Albuquerque. Semantic textual similarity of portuguese-language texts: An approach based on the semantic inferentialism model. In *11th Conf. on the Computational Processing of the Portuguese Language (PROPOR)*, pages 183–188, 2014. doi:10.1007/978-3-319-09761-9_19.

28 Ricardo Rodrigues, Hugo Gonçalo-Oliveira, and Paulo Gomes. NLPPort: A pipeline for portuguese nlp. In *$7^{th}$ Symposium on Languages, Applications and Technologies (SLATE)*, pages 18:1–18:9, 2018.

29 Barbara Rychalska, Katarzyna Pakulska, Krystyna Chodorowska, Wojciech Walczak, and Piotr Andruszkiewicz. Samsung Poland NLP team at SemEval-2016 task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity. In *10th Intl. Workshop on Semantic Evaluation (SemEval)*, pages 602–608, 2016.

30 Alberto Simões and Xavier Guinovart. Bootstrapping a Portuguese wordnet from Galician, Spanish and English wordnets. In *Advances in Speech and Language Technologies for Iberian Languages*, volume 8854 of *LNCS*, pages 239–248, 2014.

31 Alberto Simões, Álvaro Sanromán, and José Almeida. Dicionário-Aberto: A source of resources for the Portuguese language processing. In *10th Intl. Conf. on the Computational Processing of the Portuguese Language (PROPOR)*, volume 7243, pages 121–127, 2012.

32 Junfeng Tian, Zhiheng Zhou, Man Lan, and Yuanbin Wu. Ecnu at semeval-2017 task 1: Leverage kernel-based traditional nlp features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity. In *11th Intl. Workshop on Semantic Evaluation (SemEval)*, pages 191–197, 2017. doi:10.18653/v1/S17-2028.