Syntactic Knowledge for Natural Language Inference in Portuguese

Erick Fonseca(✉) and Sandra M. Aluísio

Institute of Mathematics and Computer Science, University of São Paulo,
Av. Trabalhador São-carlense, 400, São Carlos, SP, Brazil
erickrfonseca@gmail.com, sandra@icmc.usp.br

Abstract. Natural Language Inference (NLI) is the task of detecting relations such as entailment, contradiction and paraphrase in pairs of sentences. With the recent release of the ASSIN corpus, NLI in Portuguese is now getting more attention. However, published results on ASSIN have not explored syntactic structure, neither combined word embedding metrics with other types of features. In this work, we sought to remedy this gap, proposing a new model for NLI that achieves 0.72 F_1 score on ASSIN, setting a new state of the art. Our feature analysis shows that word embeddings and syntactic knowledge are both important to achieve such results.

Keywords: Natural Language Inference
Recognizing Textual Entailment · Feature engineering · Syntax

1 Introduction

Natural Language Inference (NLI), also known as Recognizing Textual Entailment (RTE), is the NLP task of determining whether a hypothesis H can be inferred from a premise P [7] (usually, P and H are sentences). Other semantic relations are also possible, such as contradiction [4,13] and paraphrase [10].

Datasets for NLI on English exist since 2005, with the RTE Challenges [6], and later with the SICK [13] and SNLI [4] corpora. For Portuguese, only recently the ASSIN [10] corpus was released, containing 10,000 sentence pairs annotated for NLI (entailment, paraphrase and neutral) and for semantic similarity [1].

Still, published results on the ASSIN dataset are worse than a word overlap baseline or only slightly better than it [2,3,8,9]. We hypothesize this is because these models focused on lexical overlap and similarity, without any attention to syntactic structure. On top of that, lexical similarity methods could be improved with the use of word embeddings, which have already been shown to be very effective in the semantic similarity task [11].

In this work, we sought to remedy this limitation by exploring richer representations for the input pairs. We extracted features including syntactic knowledge and embedding-based similarity, besides well established ones dealing with word alignments. Our model, named Infernal (*INFERence in NAural Language*),

A. Villavicencio et al. (Eds.): PROPOR 2018, LNAI 11122, pp. 242–252, 2018.
https://doi.org/10.1007/978-3-319-99722-3_25

achieved new state-of-the-art results, with 0.72 macro F_1 score on the complete ASSIN dataset (i.e., both Brazilian and European variants).

Moreover, we analyzed a sample of misclassified pairs in order to understand the difficulties of the task. We found that most of them pose significant difficulties, and that richer NLP resources (such as repositories of equivalent phrases) are necessary to improve performance on NLI and related tasks.

This paper is organized as follows. We first summarize the ASSIN dataset in Sect. 2, and briefly discuss previous related work on it in Sect. 3. We present our model in Sect. 4, and our experiments and findings in Sect. 5. We bring our conclusions in Sect. 6.

2 ASSIN Dataset

ASSIN [10] has 10,000 sentence pairs annotated for NLI (with entailment, paraphrase and neutral labels), half in Brazilian Portuguese (PT-BR) and half in European Portuguese (PT-PT)[1]. It is an unbalanced dataset: the neutral relation has 7,316 pairs, entailment has 2,080 while the paraphrases are a small portion of the set (604 pairs). Either language variant has 2,500 pairs for training, 500 for validation and 2,000 for testing, the three of them with the same proportions among classes.

Its sentences come from news articles, making the corpus more complex and with more varied topics than SNLI or SICK, which were produced from image captions. Thus, ASSIN presents challenges such as world knowledge, idiomatic expressions and named entities. Its difficulty is reflected in the fact that three out of four participants in the ASSIN shared task had performance below a word overlap baseline (a logistic regression classifier trained with the ratio of words in P that appear in H and the ration of words in H that appear in P).

3 Related Work

While works for NLI in English currently take advantage of large scale datasets to train deep neural networks [5,18], the lack of such a corpus for Portuguese has limited NLI strategies to shallow models dependent on feature engineering. Thus, we restrict our revision to published work with the ASSIN dataset.

The current state-of-the-art in ASSIN for PT-BR was achieved by Fialho et al. [9]. They consider different views of the input sentences: the original words, a lowercase version, stemmed words, among others. From each view, they extract metrics like string edit distance, word overlap, BLEU, ROUGE, and others. In total, 96 features are fed to an SVM, achieving 0.71 F_1 for PT-BR and 0.66 for PT-PT. Feitosa and Pinheiro [8] tried to improve on these results, adding eight new wordnet-based features to capture lexical similarity. However, they did not gain any significant improvement.

[1] Available at nilc.icmc.usp.br/assin/.

Rocha and Cardoso [17] reached the state of the art for PT-PT. They used a relatively small set of features, including counts of overlapping or semantically related tokens (such as synonyms and meronyms), named entities, word embedding similarity and whether both sentences have the same verb tense and voice. While the last one employs some syntactic knowledge, it is rather limited, and it is not clear how they deal with sentences with more than one verb, which are common in ASSIN. They only present results for PT-PT, with their best setup having 0.73 F_1. Curiously, while Fialho et al. [9] reports better results when combining PT-BR and PT-PT training data, Rocha and Cardoso [17] had a slight performance decrease when they did the same.

Reciclagem [2] did not use any machine learning technique; instead, it relied solely on lexical similarity metrics extracted from various resources. ASAPP, from the same authors, improved on this base by using an automatic classifier and included features such as counts of tokens, nominal clauses and named entities.

The Blue Man Group [3] extracted many word embedding-based similarity features from the pairs. They compare words of one sentence with the other, grouping similarity values in histograms, which are then fed to an SVM classifier. They also report negative results with deep neural networks, although they do not mention any performance value.

4 Data Modeling

4.1 Pre-processing

We perform some steps in an NLP pipeline in order to extract features. We run a syntactic parser, a named entity detector (NER), lemmatize words and find lexical alignments.

We used the Stanford CoreNLP dependency parser [12] trained on the Brazilian Portuguese corpus of the Universal Dependencies (UD) project[2], version 1.3. We used the spaCy pre-trained NER model for Portuguese[3], version 2.0. SpaCy also has a pre-trained syntactic parser for Portuguese, but we found that its performance is worse than the CoreNLP model.

For lemmatization, we checked the Unitex DELAF-PB dictionary[4] with POS tags produced by CoreNLP. The DELAF-PB is a Brazilian Portuguese resource; however, word forms in European Portuguese orthography can easily be checked against it after replacing some characters. If a word is not found in the dictionary, it is searched again after replacing some consonant clusters (ct and pt for t, cç and pç for ç, and mn for n).

Once we have word lemmas, we align words in the two sentences if they have the same lemma or share a synset in OpenWordNet-PT [14]. Using the same

[2] More information at http://universaldependencies.org.

[3] More information at http://spacy.io.

[4] The DELAF-PB dictionary maps inflected word forms to lemmas, according to their part-of-speech tag. More information at http://www.nilc.icmc.usp.br/nilc/projects/unitex-pb/web/dicionarios.html.

resource, we also align verbs with nominalizations (such as *correr* and *corrida*). Named entities are aligned if they are exactly the same or if one is an acronym composed of the initial letters of the words of the other one.

4.2 Feature Extraction

Once we have preprocessed sentence pairs, we can extract features from them. We also depended on two other resources to compute features: a stopword list and a word embedding model. The former is the one available in NLTK[5], expanded with the punctuation signs and the words *é*, *ser* and *estar*, which were lacking from it. The embedding model was trained with Glove [16] over a collection of news texts, literary books and Wikipedia, with over 500 million tokens.

We also used the concept of Tree Edit Distance (TED) in some features, which measures how different two trees are from each other and has already been successfully used for NLI [19]. The idea of TED is to apply a sequence of edit operations to a tree that transforms it into another one. The possible operations are the insertion of a node, removal of a node, or replacement of a node for another. The cost for each operation must be defined individually *a priori* (possibly, costs may depend on the involved words or dependency labels). Given two trees and the cost of each possible operation, the minimal TED can be computed in polynomial time. We used the Zhang-Shasha algorithm [20] in our implementation.

The complete list of features is described as follows. Note that, while we list 14 features, some of them can be computed when aligning P to H or vice-versa, and others can be normalized by the length of either sentence. In total, we extract 28 feature values.

1. **BLEU.** (BiLingual Evaluation Understudy) is a common metric in Machine Translation. It computes how many n-grams with $1 \leq n \leq 4$ from one sentence appears in the other. It has two values: using P as reference (denoted here as $P \rightarrow H$) and using H ($H \rightarrow P$).
2. **Dependency overlap.** The proportion of overlapping dependency tuples. A dependency tuple is composed by the dependency label, parent node and child node; two tuples are considered as overlapping when they have the same label and aligned parent and child nodes. Additionally, the passive subject label (*nsubjpass*) is considered equivalent to the direct object label (*dobj*). This feature has two values: the ratio of overlapping tuples with respect to the length of P and to the length of H.
3. **Nominalization.** This feature checks whether one sentence has a verb aligned with a nominalized form used as direct object. It has two values, depending on which sentence has the verb and which has the nominalization.
4. **Length ratio.** Length ratio between the number of tokens in P and H, excluding *stopwords*.

[5] More information at http://www.nltk.org.

5. **Verb arguments.** This feature has two values that check whether verbs in the two sentences also have the aligned subject and direct object. If the objects differ, it has value $(0, 0)$; if they are aligned, it is $(1, 1)$. If only H has an object, it has value $(0, 1)$, if only in P, $(1, 0)$. The values are denoted, respectively, by *verb arguments* $P \rightarrow H$ and *verb arguments* $H \rightarrow P$.
6. **Negation.** This feature checks if an aligned verb is negated in one of the sentences. This happens when the verb has a modifier with the label *neg*.
7. **Quantities.** This feature has two values that check for quantities describing aligned words (indicated by the dependency label *num*). The value of the modifier is computed both for digits and fully written forms. The first feature value is 1 if any two words have the same quantity, 0 otherwise; the second one is 1 if there is a quantity mismatch. In case of no aligned words with quantity modifier, it is $(0, 0)$.
8. **Sentence cosine.** The cosine similarity between the two sentence vectors. Vectors are obtained as the elementwise average of all token vectors.
9. **Simple TED.** The TED between the two sentences, considering insert, removal and update costs as 1. Two nodes are matched when they have the same lemma and dependency label. This feature has three values: the TED value itself, TED divided by the length of P and by the length of H.
10. **TED with cosine distance.** The TED like the one above, except that update costs are equal to the cosine distance between embeddings.
11. **Word overlap.** The ratio of words in each sentence for which there is another word with the same lemma in the other sentence, excluding stopwords. The ratio to the length of P is denoted *overlap P*, while the ratio to H is *overlap H*.
12. **Synonym overlap.** Like the one above, but considering any aligned word, not only with the same lemma.
13. **Soft overlap.** This feature measures word embedding similarity instead of a binary match. For each word in a sentence, except stopwords, we take its highest cosine similarity with words from the other sentence, then we average all similarities. It has one value for each sentence.
14. **Named entities.** This feature checks for the presence of named entities, and has three binary values. The first indicates whether there is named entity in P without an equivalent in H; the second one indicates the opposite; and the third one indicates the presence of an aligned pair. All combinations of values are possible, depending on the number of named entities in the pair.

Our features contemplate different levels of knowledge: simple count statistics (1, 4, 11), resource-based lexical semantics (3, 12, 14), syntax (2, 3, 5, 6, 7, 9, 10) and embedding-based semantics (8, 10, 13). To the best of our knowledge, features 9, 10 and 13 have not been used before for NLI. Our implementation is available at https://github.com/erickrf/infernal.

5 Experiments

We trained different classifiers in our experiments, using the scikit-learn library [15]: Logistic Regression (LR), Support Vector Machines (SVM), Random Forest (RF) and Gradient Boosting (GB).

We combined PT-BR and PT-PT training data, like in the best results reported by [9]. Before training classifiers, we normalize feature values: given a training data matrix $X \in \mathbb{R}^{n \times d}$, with n training examples and d features (28 with our full set), we normalize each column to have mean 0 and variance 1.

We did a 10-fold cross-validation in the training set in order to select the most relevant hyperparameters for some algorithms. For SVM, we used a penalty c of value 10, and an RBF kernel with γ coefficient 0.01. For RF, we used 500 trees which could use up to 6 features each, and expandable to the maximum. For GB, we used 500 trees with a maximum depth of 3 and learning rate η of 0.01. All other hyperparameters had default values of scikit-learn version 0.18.

Additionally, for LR and SVM, it is also possible to weight training examples to the inverse proportion of their class (in order to give more importance to paraphrase and entailment examples), and we also experimented that. Table 1 shows the results of our classifiers, as well as the previous state-of-the-art and the word overlap baseline.

Table 1. Infernal performance on ASSIN. The F1 values are the macro F1 (mean for all classes). The bottom part of the table shows previous state-of-the-art results and the word overlap baseline. RC refers to Rocha and Cardoso [17]

Model	Validation		PT-BR		PT-PT		All	
	Accuracy	F_1	Accuracy	F_1	Accuracy	F_1	Accuracy	F_1
RL	85.50%	0.72	**87.30%**	**0.71**	85.75%	0.72	**86.52%**	**0.72**
RL, weighted	85.20%	**0.74**	85.00%	0.69	84.60%	**0.74**	84.80%	**0.72**
Random Forest	85.20%	0.72	86.20%	0.67	**86.20%**	**0.74**	86.20%	0.71
GB	**85.80%**	0.73	86.35%	0.67	86.10%	**0.74**	86.22%	0.71
SVM	85.60%	0.73	86.90%	0.70	85.75%	0.73	86.33%	**0.72**
SVM, weighted	80.20%	0.69	79.20%	0.64	80.95%	0.71	80.08%	0.68
L2F/INESC-ID [9]	—	—	85.85%	0.66	84.90%	0.71	—	—
RC [17]	—	—	—	—	83.5%	0.73	—	—
Baseline	81.40%	0.69	82.80%	0.64	81.75%	0.7	82.27%	0.67

Almost our models achieved higher accuracy and F_1 than the previous state of the art, showing that our features provide a good representation of the data for this problem. This is more evident when we consider that we used 28 features, while [9] used 96. No single algorithm stood out as best, but Logistic Regression seems interesting for coupling good performance with low computational cost and low sensibility to hyperparameters.

5.1 Feature Analysis

We also analyzed the relative importance of our features. Determining the exact importance of each one in a multidimensional setting where there may be some interdependence is impossible, but we can get reasonable estimates from methods like Random Forest and Gradient Boosting. These methods, which are ensembles of decision trees, can score feature importance based on how well they split the data in different classes.

Thus, we trained 10 instances of RF and GB, varying the random seed, and averaged the relative importance of each one, in order to get a more stable estimate. The importance of the features can be seen in Table 2, ordered according to the average importance for the two algorithms.

As expected, features related to word overlap have bigger weight, evidenced by the good performance of the word overlap baseline. Among them, our newly

Table 2. Features importance. The first and fifth column show the relative ordering of each feature; **%GB** and **%RF** indicate the percentual importance of each feature for each algorithm.

#	Feature	%GB	%RF	#	Feature	%GB	%RF
1	Soft overlap H	12.68%	14.06%	15	TED/length H	1.94%	3.23%
2	Overlap H	11.30%	13.74%	16	BLEU P → H	2.08%	3.07%
3	Synonym overlap H	4.85%	8.73%	17	TED cosine	1.97%	2.95%
4	Soft overlap P	7.56%	4.93%	18	Quantity mismatch	3.97%	0.84%
5	Cosine	6.64%	5.36%	19	TED	2.25%	1.93%
6	Overlap P	6.85%	4.89%	20	Quantity match	2.07%	0.71%
7	Length ratio	6.01%	5.13%	21	Non-aligned NE H	1.96%	0.38%
8	TED cosine/length H	6.63%	3.92%	22	Non-aligned NE P	1.16%	0.44%
9	TED/length P	3.50%	4.90%	23	Verb arguments P → H	0.25%	0.52%
10	Dependency overlap H	2.52%	5.51%	24	Verb arguments H → P	0.25%	0.52%
11	TED cosine/length P	3.06%	4.13%	25	Nominalization in P	0.63%	0.11%
12	Synonym overlap P	3.75%	3.14%	26	Negated verb	0.51%	0.14%
13	Dependency overlap P	2.81%	3.06%	27	Aligned NE	0.09%	0.49%
14	BLEU H → P	2.32%	3.02%	28	Nominalization in H	0.39%	0.13%

proposed soft overlap is one of the most important ones as well as the sentence cosine, showing that the flexible nature of word embeddings can be very useful for NLI.

In the middle positions, we see features related to syntactic structure: dependency overlap, matching quantities and TED. While less informative than lexical overlap, they still bring substantial information, which suggests they were responsible for the good performance of our models beyond lexical similarity.

As the least useful features, we have nominalizations, named entities, negation and verb structure features. While somewhat informative, we found that negated verbs and nominalizations are relatively rare in ASSIN, limiting their impact. The verb structure feature is too specific to be discriminative as well. We conjecture that named entity features had lower usefulness because the same entity may often be described in different ways—such as an omitted first or last name. Retraining our models without the least informative features resulted in a slight performance drop, indicating that they are still good to have.

5.2 Error Analysis

We manually analyzed 65 wrongly classified pairs by our LR model and listed the linguistic phenomena that led to the mistakes. The listing is shown in Table 3 and described as follows.

Table 3. Main sources of errors

Phenomena	Occurrences
Too much overlap	23
Rewrite	21
Contextual synonyms	19
Quantity specifier	5
Qualified named entity	4

Too much overlap is the main cause of neutral pairs misclassified as entailment or paraphrase. Example: *A presidente Dilma Rousseff empossa, nesta segunda-feira (5), os novos ministros, em cerimônia no Palácio do Planalto/Dez ministros tomaram posse nesta segunda-feira (5) numa cerimônia no Palácio do Planalto.*

Rewrite is the opposite, when the same content is described with different words or implicitly. This causes entailment and paraphrases to be classified as neutral. Example: *Os trabalhadores protestam contra a regulamentação da terceirização, a retirada de direitos trabalhistas e o ajuste fiscal/Os trabalhadores protestam contra o projeto de lei que regulamenta a terceirização no país.*

Contextual synonyms are words which have the same meaning only in very specific contexts, and thus not expected to be found in wordnets. Example: *Os demais agentes públicos serão* ***alocados*** *na classe econômica/ Todo o resto dos funcionários públicos terá que* ***embarcar*** *na classe econômica.*

Quantity specifiers are expressions that specify that two quantities may differ and still keep an entailment relation, such as *at least*, *approximately*, etc. Example: *De acordo com a polícia, 56 agentes e 12 manifestantes ficaram feridos/Pelo menos 46 policiais e sete manifestantes ficaram feridos.*

Qualified named entity are named entities appearing in one sentence with a more detailed description, such as a title or profession (*actor*, *president*, etc.). Since this description is subsumed by the entity itself, it should not affect an entailment decision. Example: *Tite, no segundo tempo, trocou Ralf por Mendoza/O atacante Mendoza entrou no lugar do volante Ralf.*

These issues are hard to solve. For quantification, a list of expressions indicating approximate quantities can solve some cases. Concerning rewritten passages, resources containing equivalent expressions and phrases are also useful, although limited in the generalization capacity.

At any rate, a larger NLI corpus would be useful, allowing models to learn more subtleties from language and depend less on word overlap. Also, more data would make more feasible the efficient training of neural models, which have been successful in larger English corpora.

6 Conclusion

We have presented a new feature set for the NLI task on the ASSIN corpus, shown that it sets a new state-of-the-art with different classifiers, and performed a careful analysis of feature importance and sources of error.

The features we proposed encode syntactic knowledge about the pairs, something that, to the best of our knowledge, was missing in all published results on ASSIN to date. Also, we proposed a more flexible lexical similarity measure, the soft overlap, which is a strong indicator for NLI. Our feature set has been shown to be very useful for this task, and might be useful as well for other related tasks involving the semantics of two sentences.

Moreover, we pointed out the current challenges that ASSIN poses to NLI systems. Once we have efficient means to overcome them, even better performances can be expected.

Acknowledgments. This work was supported by FAPESP grant 2013/22973-0.

References

1. Agirre, E., et al.: SemEval-2015 Task 2: semantic textual similarity, English, Spanish and pilot on interpretability. In: Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval 2015, pp. 252–263. Association for Computational Linguistics (2015)
2. Alves, A.O., Oliveira, H.G., Rodrigues, R.: ASAPP e Reciclagem no ASSIN: Alinhamento Semântico Automático de Palavras aplicado ao Português. Linguamática **8**(2), 43–58 (2016)
3. Barbosa, L., Cavalin, P., Martins, B., Guimarães, V., Kormaksson, M.: Blue Man Group no ASSIN: Usando Representações Distribuídas para Similaridade Semântica e Inferência Textual. Linguamática **8**(2), 15–22 (2016)
4. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP, pp. 632–642. The Association for Computational Linguistics (2015)
5. Chen, Q., Zhu, X., Ling, Z.H., Wei, S., Jiang, H., Inkpen, D.: Enhanced LSTM for natural language inference. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Long Papers, vol. 1, pp. 1657–1668. Association for Computational Linguistics (2017)
6. Dagan, I., Glickman, O., Magnini, B.: The PASCAL recognising textual entailment challenge. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS, vol. 3944, pp. 177–190. Springer, Heidelberg (2006). https://doi.org/10.1007/11736790_9
7. Dagan, I., Roth, D., Sammons, M., Zanzotto, F.M.: Recognizing Textual Entailment: Models and Applications. Synthesis Lectures on Human Language Technologies. Morgan & Claypool, San Rafael (2013)
8. Feitosa, D.B., Pinheiro, V.C.: Análise de medidas de similaridade semântica na tarefa de reconhecimento de implicação textual. In: Proceedings of Symposium in Information and Human Language Technology (2017)
9. Fialho, P., Marques, R., Martins, B., Coheur, L., Quaresma, P.: INESC-ID no ASSIN: measuring semantic similarity and recognizing textual entailment. Linguamática **8**(2), 33–42 (2016)
10. Fonseca, E.R., dos Santos, L.B., Criscuolo, M., Aluísio, S.M.: Visão Geral da Avaliação de Similaridade Semântica e Inferência Textual. Linguamática **8**(2), 3–13 (2016)
11. Hartmann, N.S.: Solo queue no ASSIN: mix of a traditional and an emerging approaches. Linguamática **8**(2), 59–64 (2016)
12. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations, pp. 55–60 (2014)
13. Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., Zamparelli, R.: A SICK cure for the evaluation of compositional distributional semantic models. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation, pp. 216–223. European Language Resources Association (ELRA) (2014)
14. de Paiva, V., Rademaker, A., de Melo, G.: OpenWordNet-PT: an open Brazilian WordNet for reasoning. In: Proceedings of the 24th International Conference on Computational Linguistics, COLING 2012, pp. 353–360 (2012)

15. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
16. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Empirical Methods in Natural Language Processing, EMNLP, pp. 1532–1543 (2014). http://www.aclweb.org/anthology/D14-1162
17. Rocha, G., Cardoso, L.H.: Recognizing textual entailment: challenges in the Portuguese language. Information **9**(4), 76 (2018)
18. Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., Zhang, C.: DiSAN: directional self-attention network for RNN/CNN-free language understanding. ArXiv e-prints (2017)
19. Zanoli, R., Colombo, S.: A transformation-driven approach for recognizing textual entailment. Nat. Lang. Eng. **23**(4), 507–534 (2016)
20. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput. **18**, 1245–1262 (1989)