Towards a Relation-Based Argument Extraction Model for Argumentation Mining

Gil Rocha(✉) and Henrique Lopes Cardoso

LIACC/DEI, Faculdade de Engenharia, Universidade do Porto,
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
{gil.rocha,hlc}@fe.up.pt

Abstract. Argumentation mining aims to detect and identify the argumentative content expressed in text. In this paper we present a relation-based approach that aims to capture the relation of inference between the premise and conclusion. We follow a supervised machine learning approach and explore features at different levels of abstraction. Then, we apply this system for the task of argumentative sentence detection and compare the performance of the system with a competitive baseline approach. The corpus used in our experiments was annotated with arguments from textual resources written in Portuguese, namely opinion articles. The proposed system outperforms the baseline system, achieving 0.75 of f1-score on the test set.

Keywords: Information extraction · Argumentation mining · Machine learning · Natural language processing

1 Introduction

Argumentation is the process whereby arguments are constructed, presented and evaluated. An argument is composed by a set of propositions, where some of them (the premises) are pieces of evidence offered in support of a conclusion. The conclusion is a proposition that has truth-value (which is either true or false), put forward by somebody as true on the basis of the premises. As an example of an argument, consider the following two sentences: "All men are mortal and Socrates is a man. Therefore, Socrates is mortal.". In this simple example, the conclusion is "Socrates is mortal." and the premises are "All men are mortal" and "Socrates is a man". Each piece of text that constitutes an argument component (*i.e.* premise or conclusion) is known as an *Argumentative Discourse Unit* (ADU) [16]. The aim of *Argumentation Mining* (AM) from text, a sub-domain of text mining, is the automatic detection and identification of the argumentative structure contained within a piece of natural language text. As input, this process receives a piece of natural language text. If the text under analysis contains argumentative content, AM aims to detect all the arguments that are present in the text document, the relations between them and the internal structure of each individual argument. In the end, this process should

N. Camelin et al. (Eds.): SLSP 2017, LNAI 10583, pp. 94–105, 2017.
DOI: 10.1007/978-3-319-68456-7_8

be able to output the corresponding argument diagram: the visual representation of the arguments presented in the text. The full task of AM can be decomposed into several subtasks [17], namely: text segmentation, identification of ADUs, ADU type classification (*i.e.* premise or conclusion), relation identification and relation type classification (*i.e.* support or conclusion).

In this paper we address the task of *Argumentative Sentence Detection* (ASD) following a supervised machine learning approach and employing different formulations to address this task. We explore several machine learning (ML) and natural language processing (NLP) techniques and features at different levels of abstraction: lexical, syntactic, structural and semantic. Some of these features were constructed using external resources, such as: a part-of-speech tagger, fuzzy wordnet and a model developed to recognize textual entailment and paraphrases.

Recognizing Textual Entailment (RTE) [4], a NLP task closely related to AM, aims to find entailment relations between text fragments. Given two text fragments, typically denoted as 'Text' (T) and 'Hypothesis' (H), RTE is the task of determining whether the meaning of the Hypothesis (H, *e.g.* "Joe Smith contributes to academia") is entailed (can be inferred) from the Text (T, *e.g.* "Joe Smith offers a generous gift to the university") [21]. In other words, a sentence T entails another sentence H if after reading and knowing that T is true, a human would infer that H must also be true. We may think of textual entailment and paraphrasing in terms of logical entailment ($\models$) (see [2] for more details).

This paper is structured as follows: Sect. 2 presents related work on argumentation mining. Section 3 introduces the corpus that was used in our experiments. Section 4 describe some of the external resources that were used to performed some of the NLP tasks and employ some of features described in this paper. Section 5 describes the methods that were used to address the task of ASD using supervised ML algorithms. Section 6 presents the results obtained by the system described in this paper. Finally, Sect. 7 concludes and points to directions of future work.

2 Related Work

Most argumentation mining approaches follow a machine learning paradigm, relying on heavily engineered NLP pipelines, extensive manual creation of features and making several simplifying assumptions for each subtask of the process.

Identifying arguments and their components are the first steps of an argumentation mining system. The former is typically formulated as a binary classification problem. Most existing systems make the simplifying assumption that ADUs are sentence level and employ wide variety of ML algorithms, including SVM [20,23], Logistic regression [18], Naïve Bayes [5], Maximum Entropy [14] and Decision Trees [5,23]. Employed features can be divided into lexical, syntactic, structural and semantic. Performance of state-of-the-art systems ranges from 0.55 to 0.77 of F1-score. Fine-grained approaches to determine the exact boundaries of ADUs usually apply state-of-the-art sequential models, such as

HMM and CRF [10,11,22], with performance ranging from 0.2 to 0.42 F1-score. An exception [22] reports F1-score of 0.867, though limited to a specific text genre (persuasive essays).

ADU classification aims to classify each ADU according to its argumentative role. Approaches vary mainly in the adopted argumentation theory, leading to different sets of labels. Typically, systems employ supervised ML algorithms and specialized features: lexical, syntactic, structural, topic, sentiment and semantic [14,18,22]. Performance varies from 0.17 to 0.83 F1-score, depending on the type of texts and assumptions made.

The last two steps of the process comprise the identification and classification of rhetorical relations between ADUs, aiming to obtain an argument diagram. Few state-of-the-art argumentation mining systems address these subtasks: [3] uses textual entailment; [14] uses a context-free grammar; [17] uses a minimum spanning tree algorithm; [12] combines methods from discourse analysis, topic modeling and supervised ML; [22] employs SVM using lexical, syntactic, discourse and structural features combined with a stance recognition model. Performance ranges from 0.51 to 0.83 F1-score, relying on simplifying assumptions regarding previous steps of the process and differing on the target argumentative relations and structure.

3 Corpus

The corpus used in the experiments reported in this paper, the *ArgMine* corpus[1], consists of a news articles collection, namely opinion articles, crawled from *SAPO* (a portal that aggregates news from several news providers in Portugal, amongst other services) and annotated with arguments by human annotators. An opinion article is an article published in a newspaper that reflects the author's opinion about a specific subject. One of the advantages of working with opinion articles is the richer argumentative content that is typically present, as compared to other types of news articles. On the other hand, authors tend to use refined vocabulary which can make the interpretation of the text more challenging. In addition, different authors tend to use different writing styles, which create some variability in the analyzed texts, and in turn complicate the task of machine learning algorithms. Another characteristic of opinion articles is their typical length: they are typically longer than other types of news articles.

Since longer text documents are more difficult and very time-consuming to annotate, each opinion article was divided into paragraphs. Consequently, for each annotation task a paragraph is presented to annotators instead of the complete opinion article. Providing paragraphs to annotators instead of the complete article can have some drawbacks, namely: when an argument is spread through several paragraphs, it is impossible to annotate it because each part of the argument will be presented in different annotation tasks; moreover, in some situations it can happen that some information in the remaining parts of the document could be useful and/or necessary to detect the arguments presented

[1] http://corpora.aifdb.org/ArgMine.

in one of the parts of the document. In the first case, we assume that this situation will not occur too often. A paragraph corresponds to a distinct section in a document, usually dealing with a single topic and terminated by a new line. Since arguments have to be about some topic and changes in topic can indicate that different arguments are being expressed, as explored in [12], then this assumption seems reasonable. In some situations where they are spread through several paragraphs, arguments require complex reasoning and knowledge about the world that are beyond the scope of the approaches presented in this paper.

In each annotation task, the annotators were asked to annotate all the arguments that are explicitly stated in the corresponding paragraph. These annotations consist of argument diagrams (*i.e.* a graph structure, where each node corresponds to an ADU and arrows indicate relations of support or conflict between ADUs) following the premise-conclusion argumentation model.

More details regarding the characteristics of the *ArgMine* corpus are presented in the following sections.

4 Resources

Here we introduce external resources used as auxiliary tools by the methods employed in this paper.

4.1 Data Preparation

To transform each sentence into the corresponding set of tokens and to obtain for each token the corresponding lemma and part-of-speech information (including syntactic function, person, number, tense, amongst others) we used the *CitiusTagger* [8] NLP tool. This tool includes a named entity recognizer trained in natural language text written in Portuguese.

Several experiments were made using different NLP techniques to process the sentences received as input: removing stop-words and auxiliary words (*i.e.* words relevant for the discourse structure but not domain specific, such as: prepositions, determiners, conjunctions, interjections, numbers and some adverbial groups) and lemmatization. Transforming each token in the corresponding lemma is a promising approach because it will make explicit that some of the words are repeated in both sentences, even if small variations of these words are used (*e.g.* different verb tenses). After this step, each sentence was represented in a structured format (set of tokens) and annotated with some additional information regarding the content of the text (*e.g.* part-of-speech tags).

4.2 Semantic Resources

Knowledge about the words of a language and their semantic relations with other words can be exploited with large-scale lexical databases. To enrich the feature set shown in Tables 1 and 2 with semantic knowledge, we explored external semantic resources. By exploiting these resources we aim to enable the system to deal better with the diversity and ambiguity of natural language text.

Similarly to WordNet [6] for the English language, CONTO.PT [9] is a fuzzy wordnet for Portuguese, which groups words into sets of cognitive synonyms (called *synsets*), each expressing a distinct concept. In addition, synsets are interlinked by means of conceptual and semantic relations (*e.g.* "hyperonym" and "part-of"). Synsets included in CONTO.PT were automatically extracted from several linguistic resources. All the relations represented in CONTO.PT (i.e. relations between words and synsets, as well as relations between synsets) include degrees of membership. Two tokens (obtained after tokenization and lemmatization) are considered synonyms if they occur in the same synset. One token T_i is considered hyperonym of T_j if there exists a hyperonym relation ("hyperonym_of") between the synset of T_i and the synset of T_j. Similarly, T_i is considered meronym of T_j if there exists a meronym relation ("part_of" or "member_of") between the synset of T_i and the synset of T_j.

Finally, we exploit a distributed representation of words (word embeddings). These distributions map a word from a dictionary to a feature vector in high-dimensional space in an unsupervised setting (without human intervention). This real-valued vector representation tries to arrange words with similar meanings close to each other based on the co-occurrences of these words in large-scale (non-annotated) corpora. Then, from these representations, interesting features can be explored, such as semantic and syntactic similarities. In our experiments, we used a pre-trained model provided by the *Polyglot*[2] tool [1], in which a neural network architecture was trained with Portuguese *Wikipedia* articles.

In order to obtain a score indicating the similarity between two text fragments T_i and T_j, we compute the cosine similarity between the vectors representing each of the text fragments in the embedding space. Each text fragment is projected into the embedding space as $\vec{T_i} = \sum_{k=1}^{n} \vec{e}(w_k)n^{-1}$, where $\vec{e}(w_k)$ represents the embedding vector of the word w_k and n corresponds to the number of words contained in the text fragment T_i. Then, we compute the final value of the cosine similarity $\delta_{\vec{T_i},\vec{T_j}} = \cos(\vec{T_i}, \vec{T_j})$, $\delta_{\vec{T_i},\vec{T_j}} \in [-1, 1]$ followed by the following rescaling and normalization: $(1.0 - \delta_{\vec{T_i},\vec{T_j}})/2.0$. The entailment versor $(\hat{d})$ corresponds to the normalized direction vector obtained by subtracting the projection of T in the embedding space, $\vec{e}(T)$, from the projection of $H, \vec{e}(H)$.

Additionally, we made use of an external system for recognizing textual entailment and paraphrases in text written in the Portuguese language [19]. This system receives as input a pair of sentences $\langle T, H \rangle$, where T corresponds to the *Text* sentence and H to the *Hypothesis* sentence. Given that the problem was formulated as a multi-class classification problem, the system classifies each $\langle T, H \rangle$ with one of the labels *Entailment* (if $T \models H$), *Paraphrase* (if $T \models H$ and $H \models T$, *i.e.*, if T is paraphrase of H), or *None* (if T and H are not related with one of the previous labels). The system was trained in the ASSIN corpus [7], which corresponds, to the best of our knowledge, to the first corpus annotated with pairs of sentences written in Portuguese that is suitable for this task. It contains 5000 pairs of sentences extracted from news articles written

[2] http://polyglot.readthedocs.io/en/latest/index.html.

in European-Portuguese (EP) and 5000 pairs of sentences written in Brazilian-Portuguese (BP), obtained from *Google News* Portugal and Brazil, respectively. The model for recognizing textual entailment and paraphrases used in this paper was trained and evaluated in the EP partition of the corpus using a maximum entropy model. This model achieved an overall 0.83 of accuracy on the test set.

5 Methods

We here describe the approach we followed to address the task of argumentative sentence detection from natural language Portuguese text. We formulate the problem as a binary classification problem, following two distinct settings, as described in Sects. 5.1 and 5.2.

5.1 Sentence-Based Approach

In the first setting, each learning instance corresponds to a sentence and we aim to classify each sentence as *Argumentative* (Arg), if it contains one complete argument or at least one argumentative discourse unit (ADU), or *Non-argumentative* (NArg) otherwise. Following this setting, we make the simplifying assumption that an ADU or complete argument (*i.e.* containing at least two ADUs, the conclusion and one premise) corresponds to a single sentence. This is a strong assumption because some of the ADUs that can be found in the corpus have intra-sentence boundaries. However, learning intra-sentence boundaries to retrieve the exact boundaries of each ADU requires a corpus containing a considerable amount of intra-sentence annotations, something that the *ArgMine* corpus is lacking at this moment. We argue that making this assumption is the most adequate approach (given the corpus) to the problem.

This experimental setting can be seen as our baseline approach since it corresponds to the simplest way of formulating the problem.

Data Preparation. For each news article a_i, where $a_i \in C^{argmine}$, we divided a_i into sentences using the *Citius Tagger* tool [8], which offers the functionality of dividing a given text in different sentences as part of the process of part-of-speech tagging. Concatenating all the sentences obtained from each article $a_i \in C^{argmine}$, we obtain dataset X, which will be used for the task of argumentative sentence detection. For each sentence $x_j \in X$, we determine the corresponding target value $y_j \in Y$, where Y represents the set of target values, by performing the following procedure: consider news article a_i, where $x_j \in a_i$, and let Z be the set of ADUs annotated for news article a_i. We consider that sentence x_j has argumentative content ($y_j = 1$) if $\exists z_i \in Z : (z_i \subseteq x_j)$ *or* $(x_j \subseteq z_i)$. Otherwise, we consider that sentence x_j has no argumentative content ($y_j = 0$).

Features. As listed in Table 1, we employ features at different levels of abstraction, namely: lexical, syntactic, structural and semantic-level.

Table 1. Feature set for Sentence-based approach

Feature	Description
Lexical	
Bag-of-words	Contiguous sequence of 1 to N tokens from a given sentence. We encode the presence of unigrams ($N = 1$), bigrams ($N = 2$), and trigrams ($N = 3$) in the sentence. Experiments were made with one-hot encoding and TF-IDF encoding;
Clue words	If contains words typically found in argumentative content;
Word couples	All possible combinations of word pairs within a sentence. Experiments were made constraining the pair of words to include one or two clue words. Experiments were made with one-hot encoding and TF-IDF encoding;
Syntactic	
Stats	Statistics regarding some of the part-of-speech tags occurring in the sentence, namely: adverbs, modal auxiliary, verbs and punctuation marks. Experiments were made with normalized counters and one-hot encoding;
Verb tense	Verb tense changes between sentence and surrounding sentences.
Structural	
Sentence_length	Number of tokens in the sentence;
Avg. word length	Averaged number of letters in each word in the sentence;
Relative position	Sentence relative position in the document.
Semantic	
Domain words overlap	Overlap of domain words (nouns, adjectives, verbs) between the sentence and the surrounding sentences. Each pair of words is considered an overlap if they have the same lemma or one of the following relations: synonym, hypernym and meronym.
RTE prediction	If RTE system predicts that the sentence entails or is entailed by any other sentence in the same document
Cosine_similarity	Cosine similarity between the embedding vector $\vec{e}(s_i)$ and the embedding vector $\vec{e}(s_j)$, with $j \in \{i-1; i+1\}$.
Entail_versor	Entailment versor ($\vec{d}$) in the word embeddings space.

5.2 Relation-Based Approach

In this setting the problem is formulated in two steps: (a) a binary classifier is trained to distinguish whether a pair of sentences constitutes a simple argument or not (binary classifier). Here we assume that each sentence is an ADU and that one of the sentences plays the role of premise and the other plays the role of conclusion, composing a simple argument; (b) each sentence is classified as an argumentative sentence (Arg) if the classifier described in (a) predicts that the sentence is part of an argument (premise or conclusion) when paired with any other sentence within a given document. Otherwise, the sentence is classified as non-argumentative (NArg).

We hypothesize that the second formulation yields better predictions for ASD since it encapsulates and focuses on the notion that an argument is made

of the least two components: one conclusion and at least one premise. In Sect. 6, experiments made to validate this hypothesis are presented.

Data Preparation. Similarly to the procedure performed with the method described in Sect. 5.1, we divided each news article a_i into sentences using the *Citius Tagger* tool [8]. For each sentence $s_j \in a_i$, a pair of sentences is created with each of the remaining sentences s_k, with $k \in [1, |a_i|] \wedge k \neq j$. A positive (argumentative) pair is created with the first sentence (P) playing the role of premise and the second sentence (C) playing the role of conclusion in the corresponding annotated argument diagram. Otherwise, the pair of sentences is considered a negative (non-argumentative) pair. We followed this setup for the following reasons: (a) this approach follows the formulation used by the system for RTE and paraphrases. Consequently, predictions made by this system can be directly applied as a feature; (b) this is a consistent way of creating the learning instances (*i.e.* the premise is always the first sentence and the conclusion always the second sentence), which is an important requirement for the learning process when employing machine learning algorithms.

Table 2. Feature set for relation-based approach

Feature	Description
Lexical	
Word couples	All possible combinations of word pairs between the sentences (one word in P and other word in C). Experiments were made constraining the pair of words to include one or two clue words. Experiments were made with one-hot and TF-IDF encoding
Clue Words	If exists premise keyword in P and conclusion keyword in C;
Syntactic	
Stats	Statistics regarding some of the part-of-speech tags occurring in P and C, namely: adverbs, modal auxiliary, verbs and punctuation marks. Experiments were made with normalized counters and one-hot encoding;
Verb tense	Changes in the verb tense between P and C.
Structural	
Sentence_length	Number of tokens in P and C;
Avg. word length	Averaged number of letters in each word in P and C;
Relative position	Absolute distance in number of sentences between P and C.
Semantic	
Domain words overlap	Overlap of domain words between P and C. An overlap occurs when two words have the same lemma or are synonyms
Hyperonym	% of tokens in T hyperonyms of tokens in H. And vice-versa.
Meronym	% of tokens in T meronyms of tokens in H. And vice-versa.
Antonym	% of tokens in T antonyms of tokens in H. And vice-versa.
RTE prediction	RTE system predicts that P entails C
Cosine_similarity	Cosine similarity between the embedding vector $\vec{e}(P)$ and $\vec{e}(C)$
Entail_versor	Entailment versor ($\hat{d}$) in the word embeddings space.

Due the characteristics of the corpus, where the number of sentences containing ADUs is lower than the number of sentences that do not contain any ADU, the number of non-argumentative pairs generated with this approach is much larger than the number of argumentative sentence pairs. Consequently, we obtained a dataset that is extremely unbalanced. To overcome this problem, we performed *random undersampling* [13] to generate a balanced dataset, by randomly removing some of the learning instances (non-argumentative pairs).

Features. As listed in Table 2, we employ features at different levels of abstraction, namely: lexical, syntactic, structural and semantic-level.

Resolution Step. After training the model to classify each pair of sentences as argumentative or non-argumentative we have to translate these predictions to classify each sentence as argumentative or not (ASD), which corresponds to the task we aim to address in this paper.

First, for all possible pairs of sentences in a given document the model previously described predicts if the sentences constitute an argument or not. Then, for all sentences within a document we retrieve all the predictions where the target sentence was used (as P or C). If at least one of these predictions indicates that the target sentence forms an argument with any other sentence, then we indicate that the target sentence is an argumentative sentence. This procedure can be seen as a resolution step where we retrieve all pair-wise predictions and transform them into sentence-level predictions.

6 Experiments

The results presented in this section were obtained using the methods described in Sect. 5 and exploring the corpus described in Sect. 3.

For each classification task, we have run several experiments exploring some well known state-of-the-art algorithms, namely: *Support Vector Machine* (SVM) using linear and polynomial kernels, *Maximum Entropy model* (MaxEnt), *Adaptive Boosting* algorithm (AdaBoost) using *Decision Trees* as weak classifiers, *Random Forest Classifier* using *Decision Trees* as weak classifiers, and *Multilayer Perceptron Classifier* (Neural Net) with one hidden layer. All the ML algorithms previously mentioned were employed using the *scikit-learn* library [15] for the *Python* programming language. Since the MaxEnt model performed better for all the experiments presented in this paper, the results depicted in this section were all obtained using this model.

First, we report on 5-fold cross validation results over all the training examples available in the corpus described in Sect. 3 and using the model described in Sect. 5.1. The system obtained using this experimental setup is our baseline. Results are shown in Table 3.

In the second evaluation scenario we report results obtained using the method presented in Sect. 5.2. Since the number of non-argumentative (NArg) sentence pairs is substantially higher than the number of argumentative (Arg) sentence pairs, we employed methods to generate balanced datasets. To obtain the dataset

Table 3. Sentence-based approach scores

	Prec.	*Rec.*	*F1*	*# Sentences*
NArg	0.55	0.57	0.56	291
Arg	0.49	0.47	0.48	255

presented in Table 4, we used the random undersampling technique [13] by randomly removing some of the NArg examples until the number of NArg examples is the same as the number of Arg examples. The results shown in Table 4 were obtained in a 5-fold cross validation scenario.

Table 4. Relation-based approach scores

	Prec.	*Rec.*	*F1*	*# Sentence Pairs*
NArg	0.94	0.81	0.87	114
Arg	0.83	0.95	0.89	114

Finally, the results depicted in Table 5 were obtained using the test set partition from the corpus described in Sect. 3. The test set consists of 50 sentences: 37 non-argumentative sentences (NArg) and 13 argumentative sentences (Arg). From the analysis of the results, we conclude that the Relation-based approach yields the best overall results and, therefore, corresponds to the model that generalizes better to unseen data. This results confirm the hypothesis formulated in this paper: the Relation-based approach seems to provide a better formulation for the Argumentative Sentence Detection task.

Table 5. ASD test set scores

	Sentence-based approach			Relation-based approach		
	Prec.	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*
NArg	0.81	0.57	0.67	0.88	0.76	0.81
Arg	0.33	0.62	0.43	0.5	0.69	0.58

7 Conclusions

In this paper we address the task of argumentative sentence detection from text written in the Portuguese language. We aim to classify each sentence as containing argumentative content (*i.e.* containing a premise, conclusion or complete argument) or not. We formulate the task following two different approaches: sentence-based and relation-based approach. Validating our hypothesis, the

relation based approach outperformed the sentence-based approach in the test set, demonstrating that the relation-based system generalizes better to unseen data for the task of ASD. In future work, we aim to replicate these experiments in a different corpus to validate the conclusions reported in this paper for texts written in other languages and with a corpora containing more annotated data. Furthermore, we aim to improve the quality of the semantic-based features. Even though semantic-based features were shown to have a positive impact in the predictions made by the system, we noticed some problems regarding coverage and propagation of errors caused by the external tools employed in this paper. Better computations (*e.g.* metrics to evaluate semantic similarity in the embeddings space and fuzzy wordnet), different sentence-level representations (*e.g.* exploring tree and dependency parsers) and approaches to deal with problems of coverage that were experienced when employing external resources are promising directions to improve the results presented in this paper that we aim to pursue.

Acknowledgments. The first author is partially supported by a doctoral grant from Doctoral Program in Informatics Engineering (ProDEI) from the Faculty of Engineering of the University of Porto (FEUP).

References

1. Al-Rfou, R., Perozzi, B., Skiena, S.: Polyglot: distributed word representations for multilingual NLP. In: Proceedings of the 17th Conference on Computational Natural Language Learning, pp. 183–192. ACL, Sofia, August 2013
2. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. J. Artif. Int. Res. **38**(1), 135–187 (2010)
3. Cabrio, E., Villata, S.: Natural language arguments: a combined approach. In: ECAI, vol. 242, pp. 205–210 (2012)
4. Dagan, I., Roth, D., Sammons, M., Zanzotto, F.M.: Recognizing Textual Entailment: Models and Applications. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, San Rafael (2013)
5. Eckle-Kohler, J., Kluge, R., Gurevych, I.: On the role of discourse markers for discriminating claims and premises in argumentative discourse. In: Proceedings of the Conference on Empirical Methods in NLP, Lisbon, Portugal, 17–21 September 2015, pp. 2236–2242 (2015)
6. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database Language, Speech, and Communication. MIT Press, Cambridge (1998)
7. Fonseca, E., Santos, L., Criscuolo, M., Aluisio, S.: ASSIN: Avaliacao de similaridade semantica e inferencia textual. In: Computational Processing of the Portuguese Language - 12th International Conference, Tomar, Portugal, 13–15 July 2016 (2016)
8. Garcia, M., Gamallo, P.: Yet another suite of multilingual NLP tools. In: Sierra-Rodríguez, J.-L., Leal, J.P., Simões, A. (eds.) SLATE 2015. CCIS, vol. 563, pp. 65–75. Springer, Cham (2015). doi:10.1007/978-3-319-27653-3_7
9. Oliveira, H.G.: CONTO.PT: groundwork for the automatic creation of a fuzzy Portuguese wordnet. In: Silva, J., Ribeiro, R., Quaresma, P., Adami, A., Branco, A. (eds.) PROPOR 2016. LNCS, vol. 9727, pp. 283–295. Springer, Cham (2016). doi:10.1007/978-3-319-41552-9_29

10. Goudas, T., Louizos, C., Petasis, G., Karkaletsis, V.: Argument extraction from news, blogs, and social media. In: Likas, A., Blekas, K., Kalles, D. (eds.) SETN 2014. LNCS, vol. 8445, pp. 287–299. Springer, Cham (2014). doi:10.1007/978-3-319-07064-3_23
11. Habernal, I., Gurevych, I.: Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse. In: Proceedings of the Conference on Empirical Methods in NLP, pp. 2127–2137. Association for Computational Linguistics, Lisbon (2015)
12. Lawrence, J., Reed, C.: Combining argument mining techniques. In: Proceedings of the 2nd Workshop on Argumentation Mining, pp. 127–136. ACL (2015)
13. More, A.: Survey of resampling techniques for improving classification performance in unbalanced datasets. Computing Research Repository (CoRR) (2016)
14. Palau, R.M., Moens, M.F.: Argumentation mining: the detection, classification and structure of arguments in text. In: Proceedings of the 12th International Conference on Artificial Intelligence and Law, pp. 98–107. ACM, New York (2009)
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
16. Peldszus, A., Stede, M.: From argument diagrams to argumentation mining in texts: a survey. Int. J. Cogn. Inf. Nat. Intell. (IJCINI) **7**(1), 1–31 (2013)
17. Peldszus, A., Stede, M.: Joint prediction in MST-style discourse parsing for argumentation mining. In: Proceedings of the Conference on Empirical Methods in NLP, pp. 938–948. ACL, Lisbon, September 2015
18. Rinott, R., Dankin, L., Perez, C.A., Khapra, M.M., Aharoni, E., Slonim, N.: Show me your evidence - an automatic method for context dependent evidence detection. In: Proceedings of the Conference on Empirical Methods in NLP, Lisbon, Portugal, 17–21 September 2015, pp. 440–450 (2015)
19. Rocha, G., Lopes Cardoso, H.: Recognizing textual entailment and paraphrases in Portuguese. In: Oliveira, E., Gama, J., Vale, Z., Lopes Cardoso, H. (eds.) EPIA 2017. LNCS, vol. 10423, pp. 868–879. Springer, Cham (2017). doi:10.1007/978-3-319-65340-2_70
20. Rocha, G., Lopes Cardoso, H., Teixeira, J.: ArgMine: a framework for argumentation mining. In: Computational Processing of the Portuguese Language - 12th International Conference, PROPOR 2016, Student Research Workshop, Tomar, Portugal, 13–15 July 2016 (2016)
21. Sammons, M., Vydiswaran, V., Roth, D.: Recognizing textual entailment. In: Bikel, D.M., Zitouni, I. (eds.) Multilingual Natural Language Applications: From Theory to Practice, pp. 209–258. Prentice Hall, Upper Saddle River (2012)
22. Stab, C.: Argumentative writing support by means of natural language processing. Ph.D. thesis, Technische Universität Darmstadt, Darmstadt (2017)
23. Stab, C., Gurevych, I.: Identifying argumentative discourse structures in persuasive essays. In: Proceedings of the Conference on Empirical Methods in NLP, Doha, Qatar, 25–29 October 2014, pp. 46–56 (2014)