MIPRO 2021, September 27 - October 01, 2021, Opatija, Croatia

Negation Detection Using NooJ

Gaurish Thakkar, Nives Mikelić Preradović and Marko Tadić
Faculty of Humanities and Social Sciences, University of Zagreb, Zagreb, Croatia
gthakkar@m.ffzg.hr, nmikelic@ffzg.hr, marko.tadic@ffzg.hr

***Abstract* - The availability of extensive annotated data for natural language processing tasks is an unsolved problem. Transfer learning techniques usually mitigate these issues by relying on existing models in another language. If no such models exist, the whole transfer learning setup becomes an implausible option. This paper presents a simple approach to use grammar rule as a noisy labelling function to train a classic generative-discriminative classification setup. The approach relies on a simple NooJ grammar along with a series of other data labelling functions. We evaluate the approach on the Conan-Doyle dataset for the task of explicit negation detection with a low-resource setting and report an improvement of 2% over the baseline.**

Keywords - Negation; noisy labels; labelling functions; unsupervised learning;

I. INTRODUCTION

Negation is an occurrence of semantic opposition. This occurs when two expressions related to each other invert the meaning expressed by one of the constituents. The negation phenomenon is visible in Sentiment Analysis tasks [2], altering the sentiment's polarity. It has an impact on the search results in Information Retrieval [23]. When performing Machine Reading [18] and Information Extraction in various domains, particularly in the biomedical domain [8], the sentence's meaning may be inverted [8]. Negations can be either implicit, explicit, or morphological.

- **Implicit**, as in "*He is yet to win his first award*," carries the negative sentiment, but no negative words are present.
- **Explicit**, as in "*this is **not** good*.".
- **Morphological**, where it is either denoted by a prefix (for example "*dis-*", "*non-*") or a suffix("*-less*")", as in "*It seems a singularly **useless** thing to steal*" said Sherlock Holmes.

The most common setting for solving negation detection involves detection of negation cues and negation scopes:

- **Input**: "*Come, come, we are **not** so far wrong, after all*," said Holmes.
- **Output**: (" : O), (Come : O), (, : O), (come : O), (, : O), (we :B-scope), (are :I-scope),(not :B-cue), (so :B-scope), (far :I-scope), (wrong :I-scope), (, : O), (after : O), (all :O), (, : O), (" : O), (said : O), (Holmes : O), (. : O).

This example depicts output from a typical negation detection system, which requires a minimal amount of annotated data to train a machine learning model. Unfortunately, not all languages have such a resource available. Therefore, we will simulate a low-resourced setting for this task and build upon the techniques of NooJ grammar and data labelling functions. In this experimental setup, we concentrate our attention on the explicit and morphological variant of the problem. The rest of the paper is organised as follows. In section 3, we describe our methodology followed by data described in section 4. Section 5 discusses the experiments. Results and error analysis are discussed in sections 6 and 7, respectively, followed by a conclusion and future work in section 8.

II. RELATED WORK

Rule-based systems for negation detection proved to be commonly used and effective indifferent NLP systems [16], although their performance compared to machine learning systems is debated. However, in clinical natural language processing, as opposed to machine learning models, rule-based models apply rules that are comprehensible from the human perspective, which is essential in the context of clinical findings. According to a recent study [13], the negated finding in medical reports can be improved using lexical and syntactic rules utilising the formal grammars created in NooJ. Previous studies [28] have reported techniques that used features like negation cue phrases (NegEx [4], PyConTextNLP [3]) and dependency features (SynNeg). NegEx is used for negation detection in clinical texts. Rules in the form of regular expressions are utilised to reach decisions derived from tokens' occurrence and location in a sentence [16]. The scope of negation is captured using a window of five tokens [10]. Transformer based techniques [30,15] have shown the state of the art results in negation detection using the representations obtained from the language representation models.

Furthermore, Ratner et al. [22] introduced the system called Snorkel, which uses the collection of labelling functions based on data programming [1] to curate a large amount of supervised data without access to the ground truth. Also, Safranchik et al. [24] introduced the algorithm to link predictions from previous or next labelling to improve predictability. These previous methods [26] have been successfully used for solving named entity recognition and classification (NERC) tasks. NooJ system has been used to automatically detect verbal phrasemes in the culinary field [29], to solve the problem of verbal polysemy in automatic translation [6], to grammatically model ellipsis in a sentence and produce non-elliptic paraphrase [11] and to automatically detect and extract non-finite clauses from an English corpus of business-related texts [20]. Also, NooJ was utilised to implement the required linguistic resources for corpus study, term acquisition, and conceptualisation in the building of the legal domain-specific ontology for the Legal Information Retrieval System [12], to extract sentiment terms from a corpus of product reviews from Amazon and to compute their polarities [7] and to enhance question-answering systems based on Fuzzy Logic [17]. There is no evidence of the application of NooJ for negation detection tasks.

III. NooJ Grammars and Data Programming

Our overall setup is like [24] in terms of the algorithm,

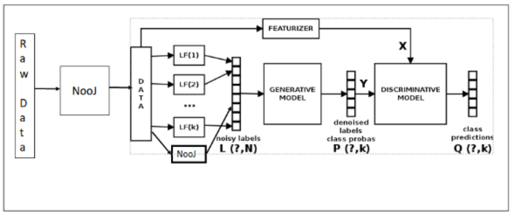

Figure 1. Overall System

but we solve explicit negation detection rather than Named Entity Recognition and Classification (NERC). Figure 1 depicts the overall system setup. The main components are as follows.

A. NooJ Grammar

NooJ [27] grammars are representations of various linguistics phenomena. The NooJ environment provides easy means to create graph-based grammar creation functionality, capturing short and long-term dependencies. We devise NooJ grammar to solve the task of negation-cue and scope detection in our experiments. The graph of the grammar is presented below in Figure 2. It is a simple graph that captures the negation-cues, everything to the cue's left and everything to the cue's right, which ends in punctuation.

B. Data Programming

Rather than relying only on a single rule or single gold annotation, this paradigm utilises multiple noisy rules in the form of labelling functions to annotate a single instance of data. These noisy rules can be a simple lookup, or simple regex or complex logic depending on the task and a user's choice. Labelling functions are advantageous when the availability of non-annotated data set is in abundance. For example, a simple labelling function for NERC could be a Wikipedia query to tag the named entity. An instance tagged by the labelling function can be a sentence or a single token, or a span of tokens. Another simple yet powerful feature of labelling functions is that they can abstain (do not assign any label) on confusing instances. Each of the labelling functions perform one task better than another and can make mistakes. To mitigate the labelling function's noisy nature, we passed the tagging through a generative model that tries to model the tagging's underlying distributions. The trained generative model is used the model to tag the corpus's untagged version. This dataset which the generative model has tagged, is used to train a discriminative model.

C. Models

The following is the list of models used in our experiments.

- **Classical HMM** [5] - The probabilistic model captures dependencies between two consecutive annotations tagged by labelling functions.
- **Linked HMM** [25] – A linking rule associates two annotations. This probabilistic model jointly models estimations using individual tagging and the linking rules.
- **Naive Bayes** [14] - Models the individual tagging as an identical and independent observation.

D. Labelling Functions

We tagged each of the tokens in the sentence using the labelling functions. The tokens are either tagged as other, cue, or a scope. The labelling functions used are as follows.

- **BigramNegationCueNT**- Tags a primary cue, which is the word not, and a secondary cue which co-occurs with the primary cue, which are the words like *can, did, would, have*. Abstains if there are no matches.
- **TrigramNegationCue**- These are the trigrams that define a negation scope. For example, by *no means, nothing at all*.
- **CommonFalsePositives**- Words like *And or But*, which are found at the beginning of the sentence. These are tagged as "O".
- **CommonTruePositivesImplicitsCue**- Words from the pre-compiled list of negation cues. For example*, not, no, n 't, never*, etc.
- **CommonTruePositivesImplicitsScope**- These are the words that begin with prefix *un-, im-, in-, dis-(**unknown, disconnected)*** or end with the suffix *–less(**breathless-ness, lifeless**)*.
- **CueConstituencyKeywords**- This is a linking rule that takes the negation cues and the constituency representation of the sentence and extracts the sub-tree, which denotes the cue's negation scope.
- **Non-EntityPunctuation**- These are the tokens from a set of punctuations like a comma, brackets, hyphens, etc. We label all the punctuation as "O".
- **NooJLabels**- This labelling function directly uses tagged output processed using NooJ grammar. It tags for negation cue and the negation scope.

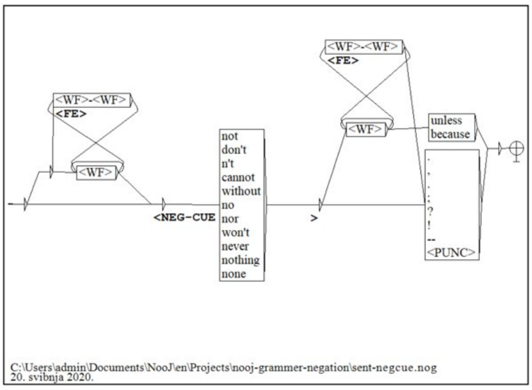

Figure 2. NooJ Grammar

E. Data

We used the Conan-Doyle negation corpus [19], which is a collection of detective stories. This corpus has been annotated for negation cue, negation scope as well as events. The event annotation was not used in this study. Table I shows the exact number of instances. We used only the negated subset of the corpus as we did not want to use any supervision from the dataset. For negation cue detection, we used a pre-compiled list of negation cues from the development set.

TABLE I. DATA DISTRIBUTION IN THE CONAN DOYLE NEGATION CORPUS

	Train	*Dev*	*Test*
Negated	842	144	235
Total	3644	787	1089

F. Experiments

The NooJ software was used for creating grammar. The NooJ grammar shown in Figure 2 is used to detect the negation scope and the cue, respectively. The detected tokens are converted to IO notation. In subsequent experiments, we combined the grammar with labelling functions to create a noisy labelled corpus. This corpus is used to train a generative model. We experimented with various models like linked HMM, HMM, and Naive Bayes. Finally, the generative model's output was used to train the discriminative model, which was BiLSTM in our case. We also performed one experiment removing NooJ grammar as a labelling function to verify the feature's subsequent contribution. Since the experiment was performed in a low-resource setting, the annotated training set was not used for supervised learning. The development set was used for fine-tuning and checking the performance of the labelling functions. We have run a grid search on the generative model hyper-parameters to obtain the best performance. The generative models from 5 epochs were trained. For the discriminative model, i. e. BiLSTM, the features from the BERT (bert-base-uncased)[9] model were used, along with character representations, to represent the tokens. The character representation was passed through a CNN encoder which outputs a 128-

TABLE II. RESULTS OF NEGATION DETECTION IN DISCRIMINATIVE STEP

Method	*Precision*	*Recall*	*F1*
NooJ	0.75	0.73	0.74
NooJ+LinkedHMM+BiLSTM	0.75	0.73	0.74
NooJ+LF+LinkedHMM+ BiLSTM	**0.76**	**0.75**	**0.76**
NooJ+LF+HMM+BiLSTM	0.75	0.73	0.74
NooJ+LF+NB+BiLSTM	0.67	0.74	0.70
LF+LinkedHMM+BiLSTM	0.50	0.74	0.60

TABLE III. RESULTS OF NEGATION DETECTION IN GENERATIVE STEP

Method	*Precision*	*Recall*	*F1*
NooJ	0.75	0.73	0.74
NooJ+LinkedHMM	0.75	0.68	0.71
NooJ+LF+LinkedHMM	**0.76**	**0.73**	**0.74**
NooJ+LF+HMM	0.75	0.73	0.74
NooJ+LF+NB	0.53	0.74	0.62
LF+LinkedHMM	0.55	0.71	0.62

dimension vector. These were concatenated and passed through an encoder. A three-layered BiLSTM with 200 hidden dimensions and 0.5 dropouts is used. The CRF layer was used for predicting the final tags of the token. As the only prepossessing step, we lowercased the tokens. Finally, the test set was used to report the performance matrices.

G. Results

The results of various labelling functions can be seen in Table III. The scores are calculated on the development set, and the column Token Accuracy depicts the accuracy of each of the labelling functions. For the first experiment, which uses NooJ grammar for detecting the negation cue and scope, the F1 score of 0.74 was achieved on the test set. Table II shows the NooJ grammar when combined with various generative models. The best performance is achieved when with LinkedHMM as it outperformed HMM on the precision score. In the last experiment, we removed the NooJ grammar rule and relied on the LinkedHMM and labelling functions only. A considerable drop in performance has been recorded. Table IV presents the same experiments but using the BiLSTM model as the discriminative model. The model was trained using the data generated from the Generative model.

Here we discovered that the best generative model also performed best in the discriminative setup. The best performing discriminative setup performed 2 points higher than the previous best generative counterpart in F1-score.

The error in detecting the negation of lines in the corpus is mainly attributed to a negation cue that is more implicit and more morphological. More missions were

TABLE IV. LABELLING FUNCTIONS AND ACCURACY.

Function	*True Positive*	*False Positive*	*False Negative*	*Token Accuracy*	*Token Votes*
BigramNegationCueNT	21	17	406	1.00	38
CommonFalsePositives	0	0	427	1.00	10
CommonTruePositivesImplicitsCue	140	5	287	0.97	145
CommonTruePositivesImplicitsScope	1	10	426	1.00	11
CueConstituencyKeywords	254	158	173	0.79	1272
Non-EntityPunctuation	0	0	427	0.95	348
NooJLabels	276	88	151	0.80	2278
TrigramNegationCue	1	0	426	1.00	3

observed if words with the prefix **un-*** or **im-*** were not part of the list. Secondly, the current setup fails to capture complex sentences, like the following:

- *Mr Sherlock Holmes, who was usually very late in the mornings, save upon those not infrequent occasions when he was up all night, was seated at the breakfast table*.
- "*Now, tell me, Dr Mortimer – and this is important – the marks which you saw were on the path and not on the grass?*".

The apparent reason for this is found in the simple NooJ grammar that skips detecting relevant cases in the first step.

IV. CONCLUSION

In this paper, we showed that a simple grammar graph does perform decently in a low-resource setting. Also, NooJ can be leveraged as a labelling function for training NLP systems. However, the experiment was done for a language that indeed has simple grammar. This is certainly not the case for low-resourced languages. For the latter languages, such grammars have to be built first. In the future, we plan to tackle more challenging semantic representation which remained unaddressed. Another refinement will be to increase the linking rules in order to improve the low accuracy and coverage. Additionally, we will add more supervision that identifies non negated spans, i.e. more tagging rules that vote on "O" tags will be added. Finally, we plan to repeat the experiment with low-resource Slavic languages (for example, Croatian, Serbian, Czech and Polish) and utilise the setup for other natural language processing tasks. Negation is a complex phenomenon and differs widely from language to language, and the complexity of writing these rules cannot be underestimated.

ACKNOWLEDGMENT

The work presented in this paper has received funding from the European Union's Horizon 2020 research and innovation programme under the *Marie Sklodowska-Curie* grant *agreement no. 812997* and under the name CLEOPATRA (Cross-lingual Event-centric Open Analytics Research Academy).

REFERENCES

[1] Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. Learning the structure of generative models without labelled data. Proceedings of machine learning research, 70:273,2017

[2] Jeremy Barnes, Erik Velldal, and Lilja Øvrelid. Improving sentiment analysis with multi-task learning of negation.arXiv preprint arXiv:1906.07610, 2019.

[3] Brian E Chapman, Sean Lee, Hyunseok Peter Kang, and Wendy W Chapman. Document-level classification of ct pulmonary angiography reports based on an extension of the context algorithm.Journal of biomedical informatics, 44(5):728–737, 2011.

[4] Wendy W Chapman, Will Bridewell, Paul Hanbury, Gregory F Cooper, and Bruce G Buchanan.A simple algorithm for identifying negated findings and diseases in discharge summaries.Journal of biomedical informatics, 34(5):301–310, 2001.

[5] Eugene Charniak.Statistical language learning. MIT Press, 1996.

[6] Hajer Cheikhrouhou. Automatic recognition and translation of polysemous verbs using the platform NooJ. In International Conference on Automatic Processing of Natural-Language Electronic Texts with NooJ, pages 39–51. Springer, 2019.

[7] N. Cirillo. Dealing with producing and consuming expressions in Italian sentiment analysis. InFehri H., Mesfar S., and Silberztein M, editors, Formalising Natural Languages with NooJ 2019 and Its Natural Language Processing Applications. NooJ 2019. Communications in Computer and Information Science, volume 1153. Springer, Cham.doi:10.1007/978-3-030-38833-1_17.

[8] Clément Dalloux, Vincent Claveau, Natalia Grabar, Lucas Emanuel Silva Oliveira, ClaudiaMaria Cabral Moro, Yohan Bonescki Gumiel, and Deborah Ribeiro Carvalho. Supervised learning for the detection of negation and of its scope in french and brazilian portuguese biomedical corpora. Natural Language Engineering, pages 1–21, 2020.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[10] Cyril Grouin, Asma Ben Abacha, Delphine Bernhard, Bruno Cartoni, Louise Deleger, Brigitte Grau, Anne-Laure Ligozat, Anne-Lyse Minard, Sophie Rosset, and Pierre Zweigenbaum.Caramba: concept, assertion, and relation annotation using machine-learning based approaches.Ini2b2 Medication Extraction Challenge Workshop, 2010.

[11] Barahona H. and Koza W. Grammatical modelling of a nominal ellipsis grammar for spanish. In Bekavac B., Kocijan K., Silberztein M., and Šojat K, editors, Formalising Natural Languages: Applications to Natural Language Processing and Digital Humanities. NooJ2020. Communications in Computer and Information Science, volume 1389. Springer, Cham.doi:10.1007/978-3-030-70629-6_8.

[12] Belhoucine K., Mourchid M., Mbarki S., and Mouloudi A. A bottom-up approach for moroccan legal ontology learning from arabic texts. In Bekavac B., Kocijan K., Silberztein M., and Šojat K, editors, Formalising Natural Languages: Ap-plications to

Natural Language Processing and Digital Humanities. NooJ 2020. Communications in Computer and Information Science, volume 1389. Springer, Cham.doi:10.1007/978-3-030-70629-6_20.

[13] Walter Koza, Mirian Muñoz, Natalia Rivas, Ninoska Godoy, Darío Filippo, Viviana Cotik,Vanesa Stricker, and Ricardo Martínez. Automatic detection of negated findings with nooj:First results. InInternational Conference on Applications of Natural Language to InformationSystems, pages 298–302. Springer, 2018.

[14] David D Lewis. Naive (bayes) at forty: The independence assumption in information retrieval.In European conference on machine learning, pages 4–15. Springer, 1998.

[15] Chen Lin, Steven Bethard, Dmitriy Dligach, Farig Sadeque, Guergana Savova, and Timothy AMiller. Does BERT need domain adaptation for clinical negation detection? Journalof the American Medical Informatics Association, 27(4):584–591, 02 2020.arXiv:https://academic.oup.com/jamia/article-pdf/27/4/584/32917050/ocaa001.pdf,doi:10.1093/jamia/ocaa001.

[16] Saeed Mehrabi, Anand Krishnan, Sunghwan Sohn, Alexandra M Roch, Heidi Schmidt, JoeKesterson, Chris Beesley, Paul Dexter, C Max Schmidt, Hongfang Liu, et al. Deepen: A negation detection system for clinical text incorporating dependency relation into negex.Journal of biomedical informatics, 54:213–219, 2015.

[17] M. Monteleone. Nooj for artificial intelligence: An anthropic approach. In Bekavac B., KocijanK., Silberztein M., and Šojat K, editors, Formalising Natural Languages: Applications to Natural Language Processing and Digital Humanities. NooJ 2020. Communications in Computer and Information Science, volume 1389. Springer, Cham.doi:10.1007/978-3-030-70629-6_15.

[18] Roser Morante and Walter Daelemans. Annotating modality and negation for a machine reading evaluation. InCLEF (Online Working Notes/Labs/Workshop), pages 17–20, 2012.

[19] Roser Morante and Walter Daelemans. Conan Doyle-neg: Annotation of negation cues and their scope in conan doyle stories. InLREC, 2012.

[20] Ben Amor O. and Derbel F. The identification of english non-finite structures using nooj platform. In Fehri H., Mesfar S., and Silberztein M, editors, Formalising Natural Languages with NooJ 2019 and Its Natural Language Processing Ap-plications. NooJ 2019.Communications in Computer and Information Science, volume 1153. Springer, Cham.doi:10.1007/978-3-030-38833-1_3.

[21] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, KentonLee, and Luke Zettlemoyer. Deep contextualised word representations. InProceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237, NewOrleans, Louisiana, June 2018. Association for Computational Linguistics. URL:https://www.aclweb.org/anthology/N18-1202,doi:10.18653/v1/N18-1202.

[22] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and ChristopherRé. Snorkel: Rapid training data creation with weak supervision. In Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases, volume 11, page 269. NIHPublic Access, 2017.

[23] Lior Rokach, Roni Romano, and Oded Maimon. Negation recognition in medical narrative reports.Information Retrieval, 11(6):499–538, 2008.

[24] Esteban Safranchik, Shiying Luo, Stephen H Bach, Elaheh Raisi, Stephen H Bach, Stephen HBach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, et al. Weakly supervised sequence tagging from noisy rules. In AAAI, pages 5570–5578, 2020.

[25] Lawrence K Saul and Michael I Jordan. Boltzmann chains and hidden markov models. In Advances in neural information processing systems, pages 435–442, 1995.

[26] Jingbo Shang, Liyuan Liu, Xiang Ren, Xiaotao Gu, Teng Ren, and Jiawei Han. Learningnamed entity tagger using domain-specific dictionary.arXiv preprint arXiv:1809.03599, 2018.

[27] Max Silberztein. Formalising Natural Languages: The NooJ Approach. John Wiley & Sons,2016.

[28] Hideyuki Tanushi, Hercules Dalianis, Martin Duneld, Maria Kvist, Maria Skeppstedt, and Sumitra Velupillai. Negation scope delimitation in clinical text using three approaches:Negex, pycontext nlp and synneg. In19th Nordic Conference of Computational Linguistics(NODALIDA 2013), May 22-24, 2013, Oslo, Norway, pages 387–474. Linköping UniversityElectronic Press, 2013.

[29] Tong Yang. Automatic extraction of verbal phrasemes in the culinary field with nooj. In International Conference on Automatic Processing of Natural-Language Electronic Texts with NooJ, pages 83–94. Springer, 2018.

[30] Yiyun Zhao and Steven Bethard. How does BERT's attention change when you fine-tune? An analysis methodology and a case study in negation scope. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4729–4747, Online, July 2020.