XplaiNLI: Explainable Natural Language Inference through Visual Analytics

Aikaterini-Lida Kalouli[1], **Rita Sevastjanova**[1], **Valeria de Paiva**[2], **Richard Crouch**[3], and **Mennatallah El-Assady**[1]
[1] University of Konstanz, firstname.lastname@uni.kn
[2] Topos Institute, valeria.depaiva@gmail.com
[3] Chegg, dick.crouch@gmail.com

Abstract

Advances in Natural Language Inference (NLI) have helped us understand what state-of-the-art models really learn and what their generalization power is. Recent research has revealed some heuristics and biases of these models. However, to date, there is no systematic effort to capitalize on those insights through a system that uses these to *explain* the NLI decisions. To this end, we propose XplaiNLI, an eXplainable, interactive, visualization interface that computes NLI with different methods and provides explanations for the decisions made by the different approaches.

1 Introduction

We present XplaiNLI, an interactive visualization, web-based interface that computes Natural Language Inference (NLI) with three different approaches and provides sketches of explanations for the decision made by each approach.[1] An overview of XplaiNLI is found in Figure 1. The user on the frontend (right) inputs a premise (P) and a hypothesis (H). The pair is passed to the backend (left) where it goes through a symbolic and a deep learning (DL) component, which compute an inference label each. Each component also determines the rules and features that lead to the decision: for the symbolic one, we use Natural Logic (Valencia, 1991) inference rules to explain the inference label, while for the DL approach, we use insights gained from relevant work (Naik et al., 2018; Gururangan et al., 2018; Dasgupta et al., 2018; McCoy et al., 2019) to account for the decision. The complete output enters the hybrid component, which combines the strengths of the symbolic NLI engine and the DL model and determines which approach's label should be trusted based on semantic characteristics of the sentences. All output is forwarded to the frontend, where an intuitive visualization encodes the inference labels of the three approaches as well the corresponding explanations. The user can interact further with the interface by adding her own heuristics and by providing feedback on the inference label, which is used for improving the separate components.

2 Related Work

Work on interpretability for NLI is still at an early stage. One strand of research explains the models by "stress-testing" them and revealing the phenomena that the models cannot handle or by detecting bias in the training data (Gururangan et al., 2018; Dasgupta et al., 2018; McCoy et al., 2019, inter alia). Another strand of research has approached the task by directly learning natural language explanations along with the inference decision (Camburu et al., 2018) or creating distributional representations of syntactic and semantic inference rules (Zanzotto and Ferrone, 2017) and training machine-learning models on them. Although all these approaches shed light on the processes behind the reasoning task, the insights gained have not yet been used in their full potential; XplaiNLI seeks to fill this gap.

3 XplaiNLI Backend Model

The backend outputs the inference relation for a given pair, as well as the features that lead to that decision, based on each of the following three approaches. The exact backend implementation and the performance

[1]Video: dropbox.com/s/mbgn3u6ilngohe1/XplaiNLI.mp4?dl=0 Demo: bit.ly/XplaiNLI Code: https://github.com/kkalouli/XplaiNLI

Proceedings of the 27th International Conference on Computational Linguistics, pages 48–52
Barcelona, Spain (Online), December 12, 2020.

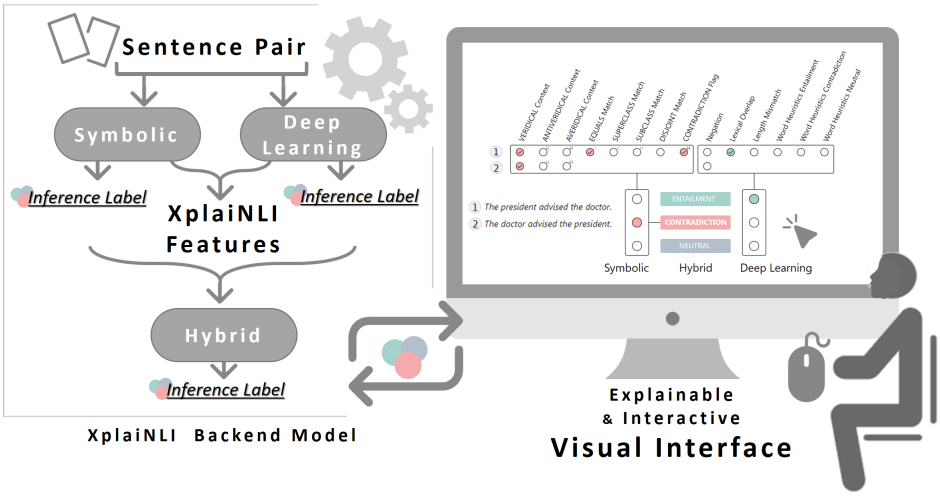

Figure 1: The high-level architecture of XplaiNLI: on the left, the three NLI approaches providing an inference label and explainable features, and on the right, the interactive, explainable, visual frontend.

of each of the approaches is detailed in Kalouli et al. (2020); this paper focuses on explainability.

3.1 The Deep Learning Component

For the DL component we use BERT-base (Devlin et al., 2018), one of the state-of-the-art models for NLI, which we fine-tune for our task. For fine-tuning, we use the SemEval 2014 version of SICK (Marelli et al., 2014). We utilize a corrected version of the corpus (Kalouli et al., 2018)[2] to mitigate some of the shortcomings of the original corpus, e.g., event and entity coreference issues. We do not fine-tune on other commonly-used benchmarks, such as MNLI (Williams et al., 2017), as these corpora suffer from similar problems. For fine-tuning, we use the HuggingFace implementation[3] and we fine-tune the parameters suggested by the authors: batch size, learning rate and number of epochs. Our best performing model uses a batch size of 32, learning rate of 2e-5 and 3 epochs. The trained model classifies an input pair into E(ntailment), C(ontradiction) or N(eutral).

To provide *potential* explanations for the model's decision, we implement the findings of Naik et al. (2018), Gururangan et al. (2018), Dasgupta et al. (2018) and McCoy et al. (2019). Their work has revealed specific heuristics and artifacts that arguably appear in the training sets of these models and can thus explain to some extent the way the models label a pair. Particularly, we implement four kinds of heuristics/explanations. First, the presence of negation. As observed by Naik et al. (2018), Dasgupta et al. (2018) and McCoy et al. (2019), negation words such as *no, not, don't, nobody*, etc. make the model predict C, consistent with the heuristic found in the SNLI training set. Second, we follow Dasgupta et al. (2018), Naik et al. (2018) and McCoy et al. (2019) and compute the lexical overlap of the two sentences. It is argued that whenever H is completely contained in P, the models tend to predict E, no matter the word order or other constraints. The third heuristic of sentence length is similar (Naik et al., 2018; Gururangan et al., 2018): Hs that are much longer than their Ps tend to be neutral, while Hs that are shorter than their Ps tend to be entailed. Last, we add relation-specific word heuristics. According to the findings of Gururangan et al. (2018), specific words being present in H or/and P are characteristic for a specific inference relation. So, generic words like *animal, instrument, outdoors* are mostly found in the Hs of entailments, while modifiers and superlatives like *sad, tall, best, first* are mostly found in neutral pairs.

3.2 The Symbolic Component

The symbolic component implements a version of *Natural Logic* (NL) (Valencia, 1991). NL attempts to explain inferences through monotonicity, i.e., by whether the concepts expressed in a sentence can become "more general" or "more specific" salva veritate. For example, in the sentence *a woman is walking*, *woman*

[2]Available under `github.com/kkalouli/SICK-processing`
[3]Available under `github.com/huggingface/transformers`

can be replaced by the more general *person* while preserving truth. The symbolic component is based on an improved version of the Graphical Knowledge Representation (GKR) by Kalouli and Crouch (2018) – GKR allows for the kind of inference mechanism we require. In the first stage of the process, P and H are parsed to their GKR representations, each producing six default GKR graphs: a dependency graph, a conceptual graph, a contextual graph, a lexical graph, a properties graph and a coreference graph. In the next stage, the lexical graphs, which contain, for each content word, the WordNet (Fellbaum, 1998) senses, synonyms, antonyms, hypernyms, hyponyms and the SUMO (Niles and Pease, 2001) concepts, superconcepts and subconcepts are used to determine matches between H and P and their specificity. For example, *person* in H can be matched to *woman* in P and be assigned the specificity *superclass*: *person* is a hypernym of *woman*. One of the four specificity markers (*equal, subclass, superclass, disjoint*) can be assigned. In the next stage, the determined specificities are updated based on the predicate-argument structure of each sentence, captured in the concept graph. For instance, *woman* is a subclass of *person* but it is not a subclass of *tall person* (not all women are tall). For the two terms of a match, the system considers if both, none or only one of them have dependents (modifiers/arguments) in their respective concept graph. Based on that, different update rules apply. For example, if *person* in H has additional dependents such as *tall* but *woman* in P does not, then the match becomes more specific: since H (*person*) was already more general than P (*woman*) (specificity superclass), then making this match more specific leads to the specificity becoming undetermined (none). After updating all H-P matches, the exact inference relation is determined based on the GKR context graphs, the instantiabilities they contain and the specificities of the matches. For example, if the H-term is instantiated and more or equally specific than the uninstantiated P-term (*a woman – no woman*), there is a contradiction. If the H-term is instantiated and more general (*a person – no woman*) than the P-term, we cannot determine the relation. Similarly for entailments: if the match is equally or more specific and both terms are instantiated, there is an entailment (*a woman - a woman*. See Kalouli et al. (2020) for more details on the symbolic engine.

These rules, i.e. the exact combinations of specificity relations and contexts, can be used straightforwardly to explain the decision made by the symbolic component.

3.3 The Hybrid Component

The hybrid approach is based on the fact that distributional features are suitable for dealing with conceptual aspects of the meanings of words, phrases, and sentences, but struggle with Boolean and contextual phenomena like modals, quantifiers, negation, implicatives, propositional attitudes, conditionals, etc. (Dasgupta et al., 2018; Naik et al., 2018; McCoy et al., 2019, to name only a few). These are phenomena to which more symbolic/structural approaches are well suited. Thus, we expect that "easy" cases which do not involve such phenomena will be best handled by the DL approach, while hard linguistic phenomena like the ones mentioned will be best handled by the symbolic approach. Thus, the hybrid component determines whether to use the symbolic or the DL label as its own inference label, based on specific semantic characteristics of the pair.

During training, the hybrid classifier learns for each pair which of the components delivers the right label (again based on the SICK-train corpus): the symbolic one (S), the DL one (DL) or both of them (B).[4] With this, the classifier indirectly learns whether the pair is "easy" or hard: if S is right, the pair is probably hard; if DL is right, the pair is probably easier; if both are right, we cannot make any claims about the nature of the pair. The learning is based on the implemented rules of the symbolic component (cf. Section 3.2), which are converted to features, e.g., the pair *P: The woman is walking. H: The person is not walking* would be assigned the features *veridical, antiveridical, superclass* because the match *person-woman* has the superclass specificity and the highest match *walk-walk* is instantiated in P and uninstantiated in H. These features (rules) capture the effects of hard linguistic phenomena like modals, negation, quantifiers, implicatives, factives, etc. To target explainability and as decision trees have been shown to be one of the most interpretable models (Guidotti et al., 2018), we train a Random Forest classifier (Gini impurity) with 30 estimators:[5] each pair is classified as one of S, DL or B, and then mapped to the respective label: if

[4] If none of them delivers the right label, then we cannot make any claims about the nature of the pair.

[5] This classifier is different from the one in Kalouli et al. (2020), where the focus is on performance rather than explainability.

classified as S or DL, the symbolic or the DL inference label are used, respectively; if classified as B, then either one of S or DL can be chosen but we use the DL label for higher robustness.

The features used for prediction are also used for explainability purposes.

4 Explainable Visual Interface

The user interface (Figure 1, right) features three main components, all emphasizing the role of the human-in-the-loop. Two text fields (for P and H) allow users to insert the inference pair to be computed.

Visualizing Explanations With the submission of the input pair, the system on the backend computes one inference label for each approach as well as explanations for each label. The results are visualized with an intuitive visualization schema (Figure 1, right): each sentence of the pair is presented along with all features that could lead to a certain inference label. On the left side, the user can find the features (rules) of the symbolic approach and on the right, the features of the DL model. The features that are relevant for this pair are colored and contain ✓, if the feature's value is true, or no ✓, if the value is false. The color of the features encodes the inference relation that each approach predicted: green is for E, red for C and grey for N. Some DL features might have lower opacity: this means that they should – according to the literature – lead to a different label than the one actually predicted by the model. In this way, the user can verify previous literature findings or discover new patterns. The colored features are then linked with the predicted inference label, also encoded by color. No link between the DL features and the label means that the prediction is not based on any of these features. In the middle of the visualization, the user can find the label of the hybrid approach, marked with bold text. Again, links visualize the behavior of the approach: if there is a link between the symbolic decision and the hybrid one, the hybrid approach chose the symbolic label; if the link is between the DL label and the hybrid one, the hybrid approach chose the DL label. If both links exist, then the labels of symbolic and DL were the same and so the hybrid approach just chose one of them. In terms of visualization, all features used for the hybrid decision are marked with a grey *H* in increasing opacity: the darker the color, the more weight this feature had for the decision.

User-defined Heuristics Along with the input pair, users can also input words – also words not found in P or H – that are expected to act as heuristics for a certain inference relation. The option of input words is available for both P and H and for all three inference relations. For instance, the user can insert the word *asleep* in the `Contradiction` field of H to check the artifact that hypotheses containing the word *asleep* are bound to be labeled as C by a DL model. Due to the system's architecture (see Section 3), only the DL model might get explained by additional heuristics; the symbolic approach is based on predefined inference rules and the hybrid approach uses semantic features to make its decision, independently from surface heuristics. The current version of the system only supports the search for specific words as heuristics; future versions will extend to further user-defined heuristics, e.g. Part-Of-Speech tags.

Learning from User Feedback The labels of the hybrid decision are at the same time clickable buttons for users to provide their annotation of the pair. With this annotation, an (offline) learning process is initiated: the pair and the user's annotation are added to the training pool of the DL model so that the model can be re-trained on increasingly large data. Whenever enough data has been collected, the model is re-trained; this re-training also triggers the re-training of the hybrid model, leading to improved results.

5 Conclusion

This paper presented an interactive visualization interface for explainable NLI. The interface uses three different approaches to compute inference and visualizes the features that lead to each decision. In contrast to black-box machine-learning models, this approach enables users to get intuitions of the decision-making process (Spinner et al., 2020), as well as to distill linguistic knowledge about the analyzed phenomena. The options for user-defined heuristics and user-driven learning can help refine the used models and components and optimize them to the users' intuition and domain understanding. To increase explainability and comparability, future work will allow the user to a) choose between different DL models for training, b) choose between hybrid models trained on different datasets, c) define their own rules for the hybrid classifier, and d) display the decision tree of the hybrid classifier for better exploration.

References

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, pages 9539–9549.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J. Gershman, and Noah D. Goodman. 2018. Evaluating Compositionality in Sentence Embeddings. *CoRR*, abs/1802.04302.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language understanding. *CoRR*, abs/1810.04805.

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

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.*, 51(5), August.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation Artifacts in Natural Language Inference Data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics.

Aikaterini-Lida Kalouli and Richard Crouch. 2018. GKR: the Graphical Knowledge Representation for semantic parsing. In *Proceedings of the Workshop on Computational Semantics beyond Events and Roles*, pages 27–37, New Orleans, Louisiana. Association for Computational Linguistics.

Aikaterini-Lida Kalouli, Livy Real, and Valeria de Paiva. 2018. WordNet for "Easy" Textual Inferences. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, may. European Language Resources Association (ELRA).

Aikaterini-Lida Kalouli, Richard Crouch, and Valeria de Paiva. 2020. Hy-NLI: a Hybrid system for Natural Language Inference. In *Proceedings of the 28th International Conference on Computational Linguistics*, COLING '20. Association for Computational Linguistics.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. SemEval-2014 Task 1: Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland, August. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy, July. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress Test Evaluation for Natural Language Inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Ian Niles and Adam Pease. 2001. Toward a Standard Upper Ontology. In Chris Welty and Barry Smith, editors, *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, pages 2–9.

Thilo Spinner, Udo Schlegel, Hannah Schäfer, and Menna El-Assady. 2020. explAIner: A Visual Analytics Framework for Interactive and Explainable Machine Learning. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1064–1074, Jan.

Victor Sánchez Valencia. 1991. *Studies on Natural Logic and Categorial Grammar*. Ph.D. thesis, University of Amsterdam.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. *CoRR*, abs/1704.05426.

F. M. Zanzotto and L. Ferrone. 2017. Can we explain natural language inference decisions taken with neural networks? Inference rules in distributed representations. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3680–3687, May.