Neural-Symbolic Modeling for Natural Language Discourse

Anonymous Authors[1]

Abstract

Building models for realistic natural language tasks requires dealing with long texts and accounting for complicated structural dependencies. Neural-symbolic representations have emerged as a way to combine the reasoning capabilities of symbolic methods, with the expressiveness of neural networks. In this paper, we present DRAIL a declarative framework for specifying deep relational models, designed to support a variety of NLP scenarios. Our framework supports easy integration with expressive language encoders, and provides an interface to study the interactions between representation, inference and learning.

1. Introduction

Understanding natural language interactions in realistic settings requires models that can deal with noisy textual inputs, reason about the dependencies between different textual elements and leverage the dependencies between textual content and the context from which it emerges. We propose a neural-symbolic approach that allows us to introduce domain knowledge, constrain the learning problem, and use expressive neural models better equipped to deal with text.

As a motivating example, consider the interactions in the debate network described in Fig. 1. Given a debate claim (t_1), and two consecutive posts debating it (p_1, p_2), we define a textual inference task, determining whether a pair of text elements hold the same stance in the debate ($\texttt{Agree}(\texttt{X}, \texttt{Y})$). This task is similar to other textual inference tasks (Bowman et al., 2015) which have been successfully approached using complex neural representations (Peters et al., 2018; Devlin et al., 2019). Exploiting the dependency between these decisions can be done using symbolic probabilistic inference. For example, assuming that one post agrees with the debate claim $\texttt{Agree}(\texttt{t}_1, \texttt{p}_2)$, and the other one does not $\neg\texttt{Agree}(\texttt{t}_1, \texttt{p}_1)$, the disagreement between the two posts can be inferred: $\neg\texttt{Agree}(\texttt{t}_1, \texttt{p}_1) \wedge \texttt{Agree}(\texttt{t}_1, \texttt{p}_2) \rightarrow \neg\texttt{Agree}(\texttt{p}_1, \texttt{p}_2)$. Finally, we consider the *social context* of the text. The disagreement between the posts can reflect a difference in the perspectives their authors hold on the issue.

Motivated by these challenges, we introduce DRAIL[1], a Deep Relational Learning framework, which uses a combined neuro-symbolic representation for modeling the interaction between multiple decisions in relational domains. Our main design goal in DRAIL is to provide a generalized tool, specifically designed for NLP tasks. Existing approaches designed for classic relational learning tasks (Cohen et al., 2020), such as knowledge graph completion, are not equipped to deal with the complex linguistic input. While others deal with very specific NLP settings such as word-based quantitative reasoning problems (Manhaeve et al., 2018) or aligning images with text (Mao et al., 2019). While the example in this paper focuses on modelings argumentation and its social and political context, the same principles can be applied to wide array of NLP tasks with different contextualizing information, such as images that appear next to the text, or prosody when analyzing transcribed speech, to name a few examples.

DRAIL uses a declarative language for defining deep relational models. Similar to other declarative languages (Richardson & Domingos, 2006; Bach et al., 2017), it allows users to inject their knowledge by specifying dependencies between decisions using first-order logic rules, which are later compiled into a factor graph with neural potentials. In addition to probabilistic inference, DRAIL also models dependencies using a *distributed knowledge representation*, denoted RELNETS, which provides a shared representation space for entities and their relations, trained using a relational multi-task learning approach. This provides a mechanism for explaining symbols, and aligning representations from different modalities. Following our running example, ideological standpoints, such as `Liberal` or `Conservative`, are discrete entities embedded in the same space as textual entities and social entities. These entities are initially associated with users, however using RELNETS this information will propagate to texts reflecting these ideologies, by exploiting the relations that bridge

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]System and code will be released to the community

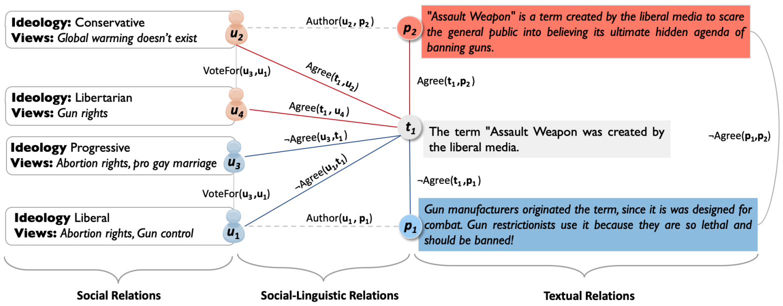


Figure 1: Example debate

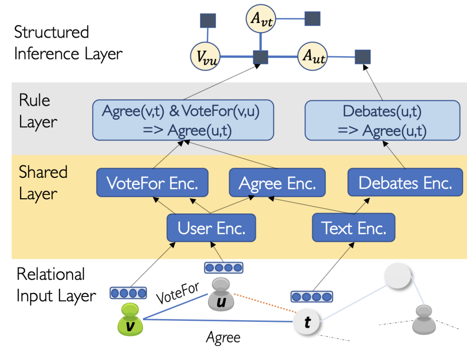


Figure 2: General DRAIL overview

social and linguistic information (see Fig. 1).

To demonstrate DRAIL's modeling approach, we introduce the task of *open-domain stance prediction with social context*[2], which combines social networks analysis and textual inference over complex opinionated texts

2. DRAIL Overview

DRAIL was designed for supporting complex NLP tasks. Problems can be broken down into domain-specific atomic components (which could be words, sentences, paragraphs or full documents, depending on the task), and dependencies between them, their properties and contextualizing information about them can be explicitly modeled. In DRAIL dependencies can be modeled over the predicted output variables (similar to other probabilistic graphical models), as well as over the neural representation of the atoms and their relationships in a shared embedding space.

A DRAIL task is defined by specifying a finite set of *entities* and *relations*. Entities are either discrete symbols (e.g., an ideology or issue stances), or attributed elements with complex internal information (e.g., documents, users). Decisions are defined using rule *templates*, formatted as horn clauses: $t_{LH} \Rightarrow t_{RH}$, where t_{LH} (*body*) is a conjunction of observed and predicted predicates, and t_{RH} (*head*) is the output predicate to be learned. Consider the debate prediction task in Fig. 1, it consists of several sub-tasks, involving textual inference, social relations and their combination.

Each rule template is associated with a neural architecture and a feature function, mapping the initial observations to an input vector for each neural net. We use a shared *relational* embedding space, denoted RELNETS, to represent entities and relations over them. As described in Fig. 2 ("Shared Layer"), each entity and relation type is associated with an encoder, trained jointly across all prediction rules. This is a form of relational multi-task learning, as the same entities and relations are reused in multiple rules and their representation is updated accordingly. Each rule defines a neural net, learned over the relations defined on their LHS. They they take a composition of the vectors generated by the relations encoders as an input (Fig. 2, "Rule Layer"). DRAIL is architecture-agnostic, and neural modules for entities, relations and rules can be specified using Pytorch.

The relations in the horn clauses can correspond to hidden or observed information, and a specific input is defined by the instantiations -or *groundings*- of these elements. The collection of all rule groundings results in a factor graph representing our global decision, taking into account the consistency and dependencies between the rules. This way, the final assignments can be obtained by performing MAP inference. For example, the dependency between the users' views on the debate topic ($\mathtt{Agree}(\mathtt{u}, \mathtt{t})$) and agreement between them on the topic ($\mathtt{VoteFor}(\mathtt{u}, \mathtt{v})$), is modeled as a factor graph in Fig. 2 ("Structured Inference Layer")).

2.1. RELNETS

Our goal when using RELNETS is to learn entity representations that capture properties unique to their types (e.g. users, issues), as well as relational patterns that contextualize entities, allowing them to generalize better. We make the distinction between *raw* (or *attributed*) entities and *symbolic* entities. Raw entities are associated with rich, yet unstructured information and attributes, such as text or user profiles. On the other hand, symbolic entities are well defined concepts, and are not associated with additional information, such as political ideologies (e.g. *liberal*) and issues (e.g. *gun-control*). With this consideration, we identify two types of representation learning objectives:

Embed Symbol / Explain Data: aligns the embedding of symbolic entities and raw entities, grounding the symbol in the raw data, and using the symbol embedding to explain properties of previously unseen raw-entity instances. For example, learning an ideology embedding that is closest to the statements made by people with that ideology.

[2] Annotated dataset will be released to the community

t0: $\texttt{InThread}(\texttt{T},\texttt{P}) \wedge \texttt{Claim}(\texttt{T},\texttt{C}) \Rightarrow \texttt{Agree}(\texttt{P},\texttt{C})$
t1: $\texttt{Debates}(\texttt{T},\texttt{U}) \wedge \texttt{Claim}(\texttt{T},\texttt{C}) \Rightarrow \texttt{Agree}(\texttt{U},\texttt{C})$
t2: $\texttt{Debates}(\texttt{T},\texttt{U}) \wedge \texttt{Votes}(\texttt{T},\texttt{V}) \Rightarrow \texttt{VoteFor}(\texttt{V},\texttt{U})$

t3: $\texttt{InThread}(\texttt{T},\texttt{P}_1) \Rightarrow \texttt{HasIdeology}(\texttt{P}_1,\texttt{I})$
t4: $\texttt{Claim}(\texttt{T},\texttt{C}) \Rightarrow \texttt{HasIdeology}(\texttt{C},\texttt{I})$
t5: $\texttt{Debates}(\texttt{T},\texttt{U}) \Rightarrow \texttt{HasIdeology}(\texttt{U},\texttt{I})$

c0: $\texttt{Agree}(\texttt{P},\texttt{C}) \wedge \texttt{Author}(\texttt{P},\texttt{U}) \Rightarrow \texttt{Agree}(\texttt{U},\texttt{C})$
c1: $\texttt{Agree}(\texttt{P}_1,\texttt{C}) \wedge \texttt{Respond}(\texttt{P}_1,\texttt{P}_2) \Rightarrow \neg\texttt{Agree}(\texttt{P}_2,\texttt{C})$
c2: $\texttt{Agree}(\texttt{P},\texttt{C}) \wedge \texttt{VoteFor}(\texttt{V},\texttt{P}) \Rightarrow \texttt{Agree}(\texttt{V},\texttt{C})$
c3: $\texttt{HasIdeology}(\texttt{C},\texttt{I}) \wedge \texttt{HasIdeology}(\texttt{U},\texttt{I})$
$\Rightarrow \texttt{Agree}(\texttt{U},\texttt{C})$

Table 1: DRAIL Program for Open Domain.
*Negated, converse and VoteSame rules omitted to save space

	Model	Random			Hard		
		P	**U**	**V**	**P**	**U**	**V**
Local	INDNETS	63.9	61.3	54.4	62.2	53.0	51.3
	E2E	66.3	71.2	54.4	63.4	68.1	51.3
Reln. Emb.	TransE	58.5	54.1	52.6	57.2	53.1	51.2
	ComplEx	61.0	63.3	58.1	57.3	55.0	55.4
	RotatE	59.6	58.3	54.2	57.9	54.6	51.0
Prob. Logic.	PSL	78.7	77.5	55.4	72.6	71.8	52.6
	TensorLog	72.7	71.9	56.2	70.0	67.4	55.8
DRaiL	E2E +Inf	80.2	79.2	54.4	76.9	75.5	51.3
	JOINTINF	80.7	79.5	55.6	75.2	74.0	52.5
	GLOBAL	81.0	79.5	55.8	75.3	74.0	53.0
	RELNETS	81.9	80.4	57.0	78.0	77.2	53.7

Table 2: General Results

Translate / Correlate: aligns the representation of pairs of symbolic or raw entities. For example, aligning user representations with text, to move between social and textual information, or capturing the correlation between symbolic judgements like agreement and matching ideologies.

3. Learning

The scoring function used for comparing output assignments can be learned *locally* for each rule separately, or *globally*, by considering the dependencies between rules.

Global Learning: This approach uses inference to ensure that the networks' parameters for all weighted rule templates are consistent across all decisions. Let Ψ be a factor graph with potentials $\{\psi_r\} \in \Psi$ over the all possible structures Y. Let $\boldsymbol{\theta} = \{\theta^t\}$ be a set of parameter vectors, and $\Phi_t(\boldsymbol{x_r}, \boldsymbol{y_r}; \theta^t)$ be the scoring function defined for potential $\psi_r(\boldsymbol{x_r}, \boldsymbol{y_r})$. Here $\mathbf{y} \in Y$ corresponds to the current prediction resulting from the MAP inference procedure and $\hat{\mathbf{y}} \in Y$ corresponds to the gold structure. We can learn using the structured hinge loss:

$$\max_{\mathbf{y} \in Y} (\Delta(\mathbf{y}, \hat{\mathbf{y}}) + \sum_{\psi_r \in \Psi} \Phi_t(\boldsymbol{x_r}, \boldsymbol{y_r}; \theta^t)) - \sum_{\psi_r \in \Psi} \Phi_t(\boldsymbol{x_r}, \boldsymbol{y_r}; \theta^t)$$

Joint Inference: Each weighted rule template optimized independently, and joint inference performed at prediction.

4. Experimental Evaluation

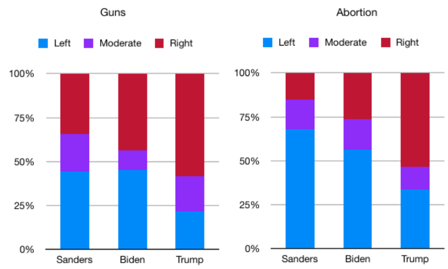
Traditionally, stance prediction tasks have focused on predicting stances on a specific topic, such as abortion. Predicting stances for a different topic, such as gun control would require learning a new model from scratch. In this task, we would like to leverage the fact that stances in different domains are correlated. Instead of using a pre-defined set of debate topics (i.e., *symbolic* entities) we define the prediction task over claims, expressed in text, specific to each debate. Concretely, each debate will have a different clam (i.e., different value for $\texttt{C}$ in the relation $\texttt{Claim}(\texttt{T},\texttt{C})$, where $\texttt{T}$ corresponds to a debate thread). We refer to these settings as *Open-Domain* and write down the task in Tab. 1. In addition to the textual stance prediction problem (t0), where $\texttt{T}$ corresponds to a post, we represent users ($\texttt{U}$) and define a user-level stance prediction problem (t1). We assume that additional users read the posts and vote for content that supports their views, resulting in another prediction problem (t2). Then, we define representation learning tasks, which align symbolic (ideology, defined as $\texttt{I}$) and raw (users and text) entities (t3,t4,t5). Finally, we write down all dependencies and constrain the final prediction (c0,c1,c2,c3).

Dataset: We collected a set of 7,555 debates from *debate.org*, containing a total of 42,245 posts across 10 broader political issues. For a given issue, the debate topics are nuanced and vary according to the debate question expressed in text (e.g. *Should semi-automatic guns be banned*, *Conceal handgun laws reduce violent crime*). Debates have at least two posts, containing up to 25 sentences each. In addition to debates and posts, we collected the user profiles of all users participating in the debates, as well as all users that cast votes for the debate participants. Profiles consist of attributes (e.g. gender, ideology). User data is considerably sparse. We create two evaluation scenarios. In the *random* split, debates are randomly divided into ten folds of equal size. In the *hard* split, debates are separated by political issue. This results in a harder problem, as the test and training data do not share topically related debates. We perform 10-fold cross validation and report accuracy.

Entity and Relation Encoders: We represent posts and titles using a pre-trained BERT-small encoder (Turc et al., 2019), a compact version of BERT (Devlin et al., 2019). For users, we use feed-forward computations with ReLU activations over profile features and pre-trained node embedding (Grover & Leskovec, 2016) over the friendship graph. All relation and rule encoders are represented as feed-forward networks with one hidden layer and ReLU activations. A softmax is used over the rule embedding to obtain a score. Note that all of these modules are updated during learning.

Issue	Ideology	Closest statements in the embedding space
LGBT	Libl	gay marriage ought be legalized, gay marriage should be legalized, same-sex marriage should be federally legal homosexuals have a right to marriage, gay marriage
	Con	homosexuality is immoral, gay marriage is not bad, homophobia is justified, Leviticus 18:22 and 20:13 prove the anti-gay position, homosexuality is not a sin nor taboo

Table 3: Embedding symbols (ideology)



Politician	Issue	Statement	Label	Score
Sanders	Guns	For background checks, and closing loopholes	Pro	.87
Sanders	Guns	Intervene with mental illness, to prevent mass shootings	Mod	.42
Sanders	Guns	Mixed approach to gun control vs. gun rights	Mod	.32
Sanders	Abortion	Advocate for family planning and funding for contraceptives	Libl	.62
Biden	Guns	Guns need to have trigger locks	Pro	.81
Biden	Guns	I go skeet-shooting, badly, and my sons go bird-hunting	Con	.51
Biden	Abortion	Accepts catholic church view that life begins at conception	Con	.86
Biden	Abortion	Ensure access to and funding for contraception	Libl	.63
Trump	Guns	No limits on guns; they save lives	Con	.85
Trump	Guns	Buying lots of ammunition and body armor should be a red flag	Pro	.74
Trump	Abortion	I am pro-life; fight ObamaCare abortion funding	Con	.52
Trump	Abortion	Planned Parenthood does great work on women's health	Libl	.52

Figure 3: Statements made by politicians explained using our model trained on *debate.org.*

Tab. 2 shows results for different types of approaches for capturing dependencies. Relational embeddings (Bordes et al., 2013; Trouillon et al., 2016; Sun et al., 2019) and probabilistic logics, a purely symbolic one (PSL) (Bach et al., 2017) and a neuro-symbolic one (TensorLog) (Cohen et al., 2020). We contrast these approaches with local models and a set of DRAIL models. For all methods, the same underlying encoders were used. In E2E models, post and user information is collapsed into a single module (rule), whereas in INDNETS, JOINTINF, GLOBAL and RELNETS they are modeled separately. We can appreciate the advantage of relational embeddings in contrast to INDNETS for user and voter stances, particularly in the case of ComplEx and RotatE. We can attribute this to the fact that all objectives are trained jointly and entity encoders are shared. However, approaches that explicitly model inference, like PSL, TensorLog and DRAIL outperform relational embeddings and end-to-end neural networks. This is because they enforce domain constraints. On the other hand, the main difference between DRAIL and the other probabilistic logics is that our GLOBAL and RELNETS models back-propagate to the base classifiers and fine-tune parameters using a structured objective. Whereas PSL and TensorLog learn rule weights over the scores of the base classifiers. We observe the advantage of having a global learning objective, sharing information with RELNETS and breaking down the decision into modules, instead of learning an end-to-end model.

Then, we perform a qualitative evaluation to illustrate our ability to move between symbolic and raw information. In Tab. 3, we *embed* ideologies and find the five statements closest in the embedding space. This experiment shows that our model is easy to interpret, and provides an explanation for the decision made.

Finally, we evaluate our learned model over entities that have not been observed during training. To do this, we extract statements made by three prominent politicians from *ontheissues.org*. Then, we try to *explain* the politicians by looking at their predicted ideology. Results for this evaluation can be seen in Fig. 3. This figure shows the proportion of statements that were identified for each ideology: left (liberal or progressive), moderate, and right (conservative). We find that we are able to recover the relative positions in the political spectrum for the evaluated politicians: Bernie Sanders, Joe Biden and Donald Trump. For the two evaluated issues, we find that Sanders is the most left leaning, followed by Biden. In contrast, Donald Trump stands mostly on the right. We also include some examples of the classified statements. We show that we are able to identify cases in which the statement does not necessarily align with the known ideology for each politician.

5. Conclusions

In this paper, we motivate the need for a declarative neural-symbolic approach that can be applied to NLP tasks involving long texts and contextualizing information. We introduce a general framework to support this, and tackle a problem with diverse relations and rich representations, resulting in a model that is easy to interpret and expand. The code and documentation for DRAIL will be released to the community, to help promote this modeling approach for other applications.

References

Bach, S. H., Broecheler, M., Huang, B., and Getoor, L. Hinge-loss markov random fields and probabilistic soft logic. *Journal of Machine Learning Research (JMLR)*, 18:1–67, 2017.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 2787–2795. Curran Associates, Inc., 2013.

Bowman, S. R., Angeli, G., Potts, C., and 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*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075.

Cohen, W. W., Yang, F., and Mazaitis, K. Tensorlog: A probabilistic database implemented using deep-learning infrastructure. *J. Artif. Intell. Res.*, 67:285–325, 2020. doi: 10.1613/jair.1.11944.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 855–864, 2016. doi: 10.1145/2939672.2939754.

Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., and De Raedt, L. Deepproblog: Neural probabilistic logic programming. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 3749–3759. Curran Associates, Inc., 2018.

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., and Wu, J. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. 2019.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202.

Richardson, M. and Domingos, P. M. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006. doi: 10.1007/s10994-006-5833-1.

Sun, Z., Deng, Z., Nie, J., and Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., and Bouchard, G. Complex embeddings for simple link prediction. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2071–2080, New York, New York, USA, 20–22 Jun 2016. PMLR.

Turc, I., Chang, M.-W., Lee, K., and Toutanova, K. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*, 2019.