MARKS-mech: A Mask-based Prior Knowledge Dissemination Mechanism for including Discourse Relations for Sentiment Classification

2024 International Joint Conference on Neural Networks (IJCNN) | 979-8-3503-5931-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/IJCNN60899.2024.10651532

Shashank Gupta*, Antonio Robles-Kelly*†, Mohamed Reda Bouadjenek*, Asef Nazari*, and Dhananjay Thiruvady*
*School of IT, Deakin University, Geelong, VIC 3216, Australia
Email: guptashas@deakin.edu.au
†Defence Science and Technology Group, Edinburg, SA 5111, Australia

***Abstract*—Disseminating prior knowledge about a pattern recognition task in Deep Neural Networks (DNNs) is desirable, to enable them to learn some complex patterns or representations, that are otherwise difficult to learn via usual data-driven training. Several methods have been proposed for that purpose, but creating an end-to-end trainable DNN model, while keeping it informed with prior knowledge, remains a challenging task. In this paper, we propose a method to disseminate prior knowledge in DNN models. Specifically, we created a novel MAsk-based pRior Knowledge diSsemination mechanism (MARKS-mech), that transfers logical prior knowledge in DNN models via input data transformation. We utilize a recently constructed Twitter-based dataset to perform our experiments, which is specifically designed to test the logical prior knowledge dissemination ability of methods like ours. We find that our method provides superior knowledge dissemination performance compared to the baselines.**

1. Introduction

Deep Neural Networks (DNNs) exhibit remarkable performance on pattern recognition tasks, mainly due to their ability to learn hierarchical feature representations in unstructured data [8]. Generally, DNNs learn these representations through purely data-driven approaches in which learning is performed automatically through the input training data [17], without any other external supervision.

Previous works have also shown that purely data-driven training, is not only insufficient for learning some complex and desirable representations, but may also lead to learning spurious representations [21], [11]. Moreover, representations learned from purely data-driven learning are highly complex and humanly uninterpretable, due to which, no meaningful relationship in terms of *how?* and *why?* can be established between the input and DNN output. Thus, DNNs are essentially treated as *Black-box models* [28].

For example, on the sentence-level sentiment classification task, it is challenging for simple DNN models like Convolutional Neural Networks (CNNs) [23] and Recurrent Neural Networks (RNNs) [13], [3] to capture complex-linguistic patterns called *Contrastive Discourse Relations* (CDRs) like "a-but-b" in input sentences, via purely data-driven training [14], [16]. A sentence contains the "a-but-b" CDR, if it has an "a-but-b" syntactic structure and the conjuncts - "a" and "b" - contain contrastive sentiment-polarities. In such case, the sentence-sentiment is determined as per the "b" conjunct, and utilizing the opposing sentiment information in "a" conjunct will lead to incorrect sentence sentiment prediction [18], [20], [30], [32].

To counter the drawbacks of purely data-driven training, a group of algorithms and methods called *Informed Machine Learning* (IML) [4], [29] have been proposed, that incorporate some external supervision during the DNN training. This external supervision is generally the *prior knowledge* [17] about the task, which consists of some complex desirable patterns required to be learned, and are difficult for the model to learn via purely data-driven training. These methods usually model this knowledge in a symbolic form and disseminate it in the DNN, to influence its decision to be consistent with the symbolic knowledge.

In this paper, we propose an IML method called MAsk-based pRior Knowledge diSsemination mechanism (MARKS-mech), that captures logical prior knowledge like the "a-but-b" CDR [20] on input sequence, and disseminates the information about dominant conjunct in the DNN model (called *CDR dissemination* task). While providing predictions, our method employs *Feature Manipulation* to pass only the features consistent with prior knowledge to the DNN model. Our method is agnostic to the choice of DNN; does not require any complicated ad-hoc changes to the DNN construction pipeline which consists of: i)training data preparation, ii)defining hypothesis, and iii)optimization; and is jointly optimized with the DNN to create an end-to-end trainable model. The key contributions of this work are summarized as follows:

1) We introduce an IML method called *MARKS-mech*, that helps a DNN model to effectively recognize complex logical structures in input data, and incorporate them while providing prediction on a pattern recognition task.
2) We conduct a thorough analysis of our method on a Twitter-based dataset, which is specifically designed to test the CDR dissemination ability of IML methods. Our analysis includes: -

a) Testing multiple configurations of our method with multiple DNN models, to provide a thorough comparison against the baselines and propose the best configuration for each DNN model.
b) Calculating multiple metrics, which quantify both the sentiment classification and CDR dissemination performances of our method.

2. Related Work

2.1. Informed Machine Learning

IML consists of methods or algorithms proposed to combine prior knowledge [17] about the pattern recognition task with a machine learning model [4], [29]. The combination process usually involves transforming either one, or a combination of multiple steps in the data-driven construction pipeline of machine learning models. This pipeline generally involves the following steps: i) training data preparation, ii) defining the hypothesis (architecture), or iii) defining the learning algorithm (training objective or the loss function). Note that more well-known and similar fields like "Neural-Symbolic AI" [7], [2], [31] can be classified as the sub-field of IML [4].

2.2. Implicit IML methods

While not originally proposed as an IML method, some works show that certain existing DNN models can implicitly capture linguistic structures like "a-but-b", without any explicit modifications to their construction pipeline. For example, Krishna et al. [16] claimed that, creating *Contextualized Word Embeddings* from input sequence can inherently capture the dominant-conjunct information, when fine-tuned with the DNN model on downstream sentiment analysis task. They proposed to create these embeddings using a large pre-trained language model called ELMo [24]. We instead use implicit learning to learn a *rule-mask* by a sequence model which represents the applicable structure and utilize it to transfer information about dominant conjunct on input features via *Feature Manipulation*.

2.3. Explicit IML Methods

These methods incorporate prior knowledge explicitly by encoding information into the trainable weights of a neural network. This is done by modifying either its input training data, architecture, or its objective function. Hu et al. [14] proposed *Iterative Knowledge Distillation* where CDRs modelled as first-order logic rules are incorporated with DNNs via a combination of Knowledge Distillation [12] and Posterior Regularization [6]. The authors proposed an upgraded version of this method called *Mutual Distillation* [15], where some learnable parameters ϕ are introduced with logic rules when constructing the constrained posterior [6], which are learned from the input data. Instead of formulating constraints as regularization terms, Li and Srikumar [19] build *Constrained Neural Layers*, where logical constraints govern the forward computation operations in each neuron. In contrast to these methods, our approach does not encode the CDR information into the trainable parameters of the model, but instead uses *Feature Manipulation* to represent CDR information on input features and pass it to the DNN model while providing predictions. Thus, our method can incorporate such structures without any such complicated ad-hoc changes to either input data, architectures, or training procedures.

3. Methodology

3.1. Contrastive Discourse Relations for Sentiment Classification

Previous works [14], [16] have shown that *Contrastive Discourse Relations* like "a-but-b" are hard to capture by general DNNs like CNNs [23] and RNNs [3] for sentence-level sentiment classification. Sentences containing a CDR have a syntactic structure like *a-keyword-b* where two conjuncts - "a" and "b" - are connected through a discourse marker (*keyword*) and have contrastive sentiment polarities [25]. These relations can be further classified into (i) CDR_{Fol}, where the dominant clause is following (*b* conjunct), or (ii) CDR_{Prev}, where the dominant clause is preceding (*a* conjunct). In Table 1, we provide a list of CDRs used in this study.

Given an input sentence, our task is to identify if it contains a CDR, pass the information about the dominant conjunct to the DNN model, and influence the model decision to be based on the dominant conjunct. Whereas, if the sentence just contains a CDR-syntactic structure (conjuncts do not contain contrastive sentiment senses), we pass the entire sentence to the DNN model to prevent any loss of sentiment-sensitive information. This is called **CDR dissemination task**.

For example, given a sentence "The movie is good *but* the casting was terrible" containing the "a-but-b" CDR (it contains the "a-but-b" syntactic structure and "a" & "b" conjuncts contain contrastive sentiments), we pass only the "b" conjunct information to the DNN model to predict sentence sentiment. This is because the entire sentence sentiment is consistent with the "b" conjunct, and utilizing the opposing sentiment information in "a" conjunct will lead to incorrect sentence sentiment prediction [18], [20], [30], [32]. For a sentence like "John is good at math *but* he is best in Physics", we want the DNN to base its decision on the entire sentence as it contains the "a-but-b" syntactic structure, but "a" & "b" conjuncts do not contain contrastive polarities of sentiment. Hence, removing the "a" part will result in the loss of sentiment-sensitive information. For simple sentences like "I like this movie", we again pass the entire sequence information to the DNN model.

We achieve the task of CDR dissemination in DNNs by dividing it into the following sub-tasks:

TABLE 1: List of Contrastive Discourse Relations (CDRs) used in our analysis.

CDR	Keyword	Dominant conjunct	Sentence
"a-*but*-b"	"but"	"b" (CDR_{Fol}) structure [20]	The movie is good **but** the casting is terrible
"a-*yet*-b"	"yet"	"b" (CDR_{Fol}) structure [20]	Even though we can't travel **yet** we can enjoy each other and what we have
"a-*though*-b"	"though"	"a" (CDR_{Prev}) structure [20]	You are having an amazing time **though** we are having this awful pandemic
"a-*while*-b"	"while"	"a" (CDR_{Prev}) structure [1]	Stupid people are not social distancing **while** there's a global pandemic

1) Recognize the CDR in the input sentence
2) Encode the information about dominant-conjunct on the input sentence
3) Pass the information to the DNN model

Our method contains individual components, each of which achieves one of these sub-tasks and shares the output with others to disseminate the information about the dominant conjunct in the DNN model.

3.2. Recognizing the CDR in Input Sentence via Rule-mask

Given a sentence s as an ordered sequence of n tokens $[t_1, t_2, \cdots, t_n]$, we want to determine whether it contains any of the relations listed in Table 1. This is done by recognizing whether it contains a syntactic structure like "a-keyword-b", and whether the "a" & "b" conjuncts contain contrastive sentiment polarities. The $keyword$ denotes the discourse marker corresponding to a CDR listed in Table 1.

To achieve this objective, we feed the sentence s to a Seq2seq DNN model called **Rule-mask Block** (shown in Figure 1a) which outputs a **rule-mask** value (m). This rule-mask m is an ordered sequence of n binary values, where each value corresponds to a token in the sentence s. The following denotes the rule-mask output for each case of input sentence

1) For a sentence ["The", "movie", "is", "good", "but", "casting", "is", "bad"], which contains a CDR_{Fol} relation ("a-but-b"), the rule-mask output is [0, 0, 0, 0, 0, 1, 1, 1] in which the values are 0 for tokens corresponding to "a" and $keyword$ parts in sentence s. In this case, the rule-mask is defined to have a syntactic structure of $\mathbf{0} - 0 - \mathbf{1}$ denoting that the tokens corresponding "a" and "keyword" parts will not be considered (or masked out) when the DNN performs sentiment classification on s.
2) For a sentence ["You", "are", "having", "an", "amazing", "time", "though", "we", "are", "having", "this", "awful", "pandemic] which contains a CDR_{Prev} relation ("a-though-b"), the rule-mask output is [1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0] in which the values are 0 for tokens corresponding to "b" and $keyword$ parts in sentence s. In this case, the rule-mask is defined to have a syntactic structure of $\mathbf{1} - 0 - \mathbf{0}$ denoting that the tokens corresponding "b" and "keyword" parts will be masked out when the DNN performs sentiment classification on s.
3) For sentences with no relation like ["Titanic", "is", "good", "movie"] or just contains the CDR syntactic structure ["John", "is", "good", "at", "math", "but", "he", "is", "best", "in", "Physics"], the rule-mask output is [1, 1, 1, 1] for the former case, and [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1] for the latter case. The rule-mask in this case is defined to have a syntactic structure of $1 - 1 - 1$ denoting that the entire sentence s will be considered for sentiment classification.

The Rule-mask Block contains a many-to-many output sequence layer followed by a sigmoid-activation layer as shown in Figure 1a. The sequence layer consists of a recurrence-based DNN model like GRU [3] or LSTM [13]. Essentially, we treat the task of rule-mask prediction as a token-level binary classification problem (akin to NER tagging), where we predict either 0 or 1 tags corresponding to every token in the input sentence. Note that we use predictive modeling to construct the rule mask instead of manual construction otherwise, it will not be possible to determine the contrastive sentiment senses between the conjuncts "a" and "b". Moreover, we utilize probabilistic values $[p_{\theta_1}(y_1|t_1), p_{\theta_1}(y_2|t_2), \cdots, p_{\theta_1}(y_n|t_n)]$ of the rule-mask vector instead of absolute values to preserve more information.

In Table 6, we show how accurately this mechanism can predict the rule-mask when using different many-to-many sequence layers like GRU [3] and LSTM [13]. We compare the rule-mask accuracy of our method with another IML method in the literature called Rule-Mask Mechanism [9] which also uses feature manipulation to transfer prior knowledge via a rule-mask vector. Although it performs well, it often confuses sentences containing a CDR and just CDR-syntactic structures. This leads to inaccurate rule-mask outputs on sentences containing just the CDR-syntactic structures, where the sentiment-sensitive information in "a" conjunct is masked out. Our method contains a *Rule-mask Correction* module as described in Section 3.3 which provides more accurate rule-masks demonstrated by rule-mask accuracy results in Table 6. In Section 5.3, we provide a more detailed description of the rule-mask accuracy and analyze its results.

3.3. Rule-mask Correction

The Rule-mask block as shown in Figure 1a can effectively determine the CDR-syntactic structure in the input sentence, however, it often outputs incorrect rule-mask vectors for sentences that just contain the CDR-syntactic structure. For example, for the sentence "The movie is good *but* the casting was terrible" containing the "a-but-b" CDR, it predicts a rule-mask of syntactic structure $0 - 0 - 1$. Whereas, for the sentence "John is good at math *but* he is

best in Physics", it also often predicts the rule-mask of the structure $0-0-1$. To correct the rule-mask values, especially for CDR-syntactic structure sentences, we incorporate the *Rule-mask Correction* module in our method which consists of the *Contrast Block* and the *Rule-mask Update Block*.

3.3.1. Contrast Block. We create a component called *Contrast Block* to specifically determine the contrastive sentiment senses between the conjuncts, as shown in Figure 1b. It consists of a many-to-one output sequence layer followed by a sigmoid activation layer. It takes the input sentence s of a structure *a-keyword-b* and provides a prediction $c = p_{\theta_2}(y_c|s)$, which determines whether the conjuncts "a" and "b" contain contrastive sentiment polarities. If they do, it outputs a value of 1, otherwise it outputs 0. For sentences with no syntactic structure, it outputs 0, since the applicability of CDR requires the presence of a corresponding syntactic structure. Figure 2 denotes the outputs of the contrast block for example cases.

3.3.2. Rule-mask Update Block. After calculating the contrast output c (which determines contrastive sentiments between conjuncts), we utilize it to correct the rule-mask output by re-calculating it as m_{CDR} as shown in Eq. (1).

$$m_{CDR} = [1_1, 1_2, \cdots, 1_n] - p_{\theta_2}(y_c|s) + p_{\theta_2}(y_c|s)[p_{\theta_1}(y_1|t_1), p_{\theta_1}(y_2|t_2), \cdots, p_{\theta_1}(y_n|t_n)] \quad (1)$$

where $[p_{\theta_1}(y_1|t_1), p_{\theta_1}(y_2|t_2), \cdots, p_{\theta_1}(y_n|t_n)]$ represents the rule-mask prediction from the Rule-mask Block and $p_{\theta_2}(y_c|s)$ represents the contrast prediction from the Contrast Block. Based on the contrast prediction value, the outputs of Eq. (1) are either:

1) **A sequence of ones containing n values if $p_{\theta_2}(y_c|s) = 0$:** Denotes that the sentence does not contain conjuncts with contrastive sentiment polarities and the rule-mask should be modified as a "list of ones containing n values" (i.e. of syntactic structure $1-1-1$). This means that DNN should output its decision based on all the tokens of the sentence since it contains just the CDR-syntactic structure, not the actual CDR. Thus, using just a part of the sentence can lead to a loss of sentiment-sensitive information.
2) **The rule-mask prediction if $p_{\theta_2}(y_c|s) = 1$:** Denotes that the sentence contains conjuncts with contrastive sentiment polarities and the CDR corresponding to the CDR-structure is applicable. Hence, the rule-mask should remain as it was predicted from the *Rule-mask Block*, and the DNN should output its decision based on the dominant conjunct of the sentence.

In Figure 2, we show the outputs of *Rule-mask Update* for possible cases of sentences.

3.4. Encoding and Disseminating CDR information

To encode and disseminate information about the dominant conjunct in the DNN model, we compute a product between s and m_{CDR} to calculate a post-processed instance s_c as shown in Eq. (2), which only contains tokens corresponding to the dominant-conjunct. We term it as *Feature Manipulation*.

$$s_c = m_{CDR} * s \quad (2)$$

After computing s_c, we pass it to the downstream DNN model, which now determines the sentiment on the basis of the dominant-conjunct as $p_{\theta_3}(y_s|s_c)$. We optimize the outputs of all the components as well as the sentiment prediction of DNN jointly to create an end-to-end trainable model as shown in Eq. (3)

$$\min_{\theta_1,\theta_2,\theta_3 \in \Theta} L(y_s, p_{\theta_3}(y_s|s_c)) + \Sigma_{i=1}^{n} L(y_i, p_{\theta_2}(y_i|t_i)) + L(y_c, p_{\theta_2}(y_c|s)) \quad (3)$$

where L is the *Binary Cross-Entropy* loss function, $(y_s, p_{\theta_3}(y_s|s_c))$ are the sentiment value and sentiment prediction pairs, $(y_c, p_{\theta_2}(y_c|s))$ are the contrast value and contrast prediction pairs, and $(y_i, p_{\theta_2}(y_i|t_i))$ are the mask value and mask prediction pairs for i^{th} a token in the input sequence $s = [t_1, t_2, \cdots, t_n]$. In Figure 2, we provide a diagrammatic representation of our method and show how each component interacts with others to achieve CDR dissemination in the DNN model.

Our method is completely agnostic to the choice of DNN model, does not require any ad-hoc changes to the DNN construction process, and is jointly optimized with the DNN to create an end-to-end trainable model. While it can also be coupled with the popular transformer-based DNNs like BERT [5] and GPT-2 [26], the focus of our paper is limited to helping simple DNN models based on CNNs [23] and RNNs [3], [13] to capture the complex discourse information as they are unable to do so themselves [14], [16].

4. Experimental Setup

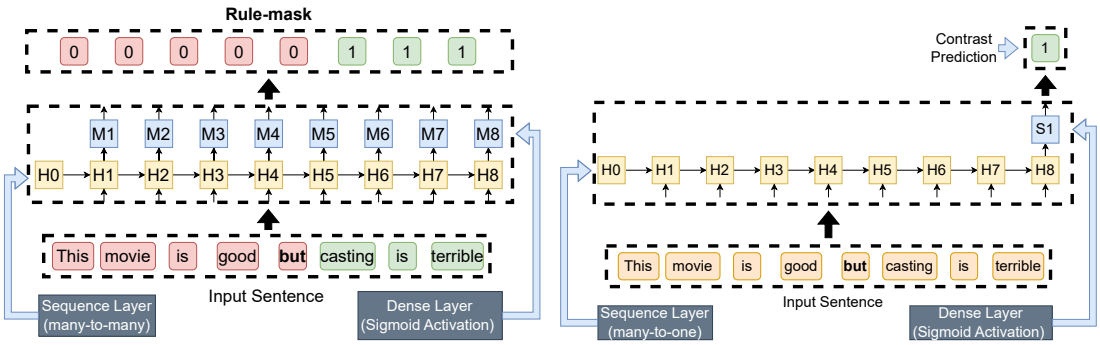
This section describes our experimental setting for testing the CDR dissemination performance of our method.[1]

4.1. Dataset

In order to carry out a comprehensive and impartial analysis, we utilize a new dataset sourced from Twitter called *Covid19-twitter* [9]. This dataset comprises sentences as tweets related to the COVID-19 topic and was specifically designed to test the CDR dissemination performance of IML methods. In Figure 3, we provide a brief description of this dataset.

We use a random division to split this dataset into train, validation, and test sets each containing 80%, 10%, and 10%

1. The source code of this work is available at: https://github.com/shashgpt/CDR-Mechanism.



(a) Rule-mask Block (b) Contrast Block

Figure 1: Diagrams of the Rule-mask and the Contrast blocks used in our method.

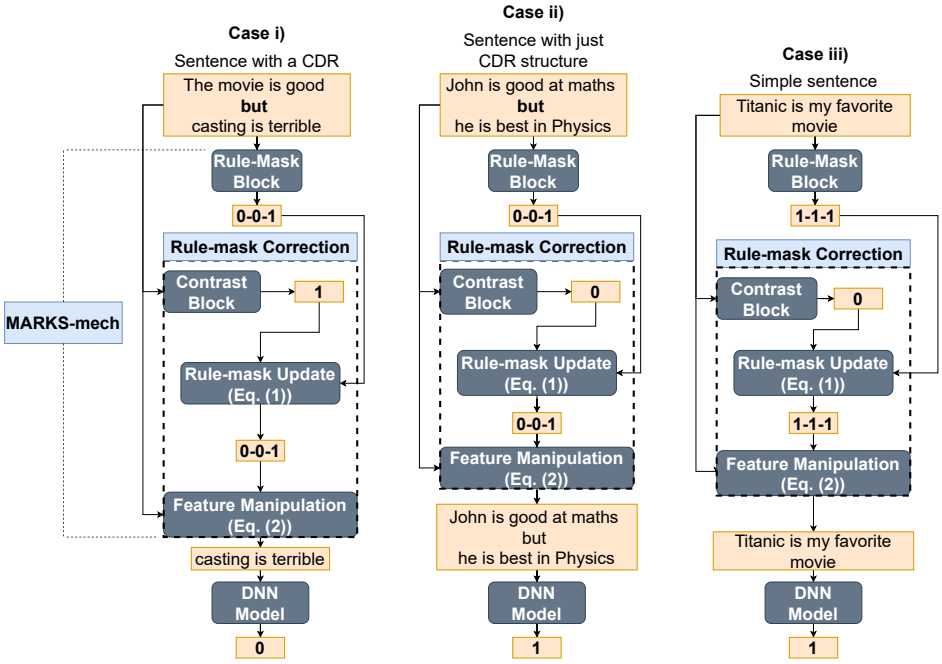


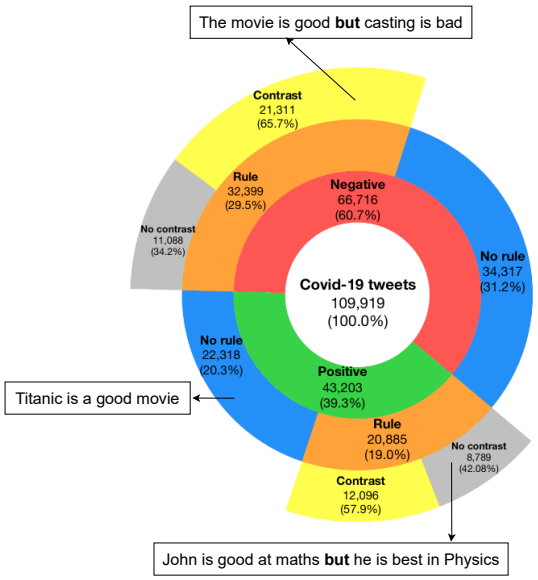
Figure 2: MARKS-mech architecture and its outputs for input sentences containing: i) a CDR, case ii) just the CDR syntactic structure (conjuncts do not contain contrastive sentiment senses), iii) no structure. Note that in the case of a CDR_{Prev} relation, the rule-mask output will be $1 - 0 - 0$.

proportion of tweets respectively. Each set also contains a similar symmetric distribution of tweets as present in the entire dataset (shown in Figure 3a). We show all our results (in Tables 2, 3, and 6) on the *Rule* subset of the test set, which contains an equal proportion of *Rule-contrast* (sentences containing CDRs) and *Rule-no-contrast* (sentences containing just CDR syntactic structures) tweets. This is done to demonstrate the CDR dissemination performance of the classifiers.

4.2. Sentiment Classifiers

To conduct an exhaustive empirical analysis, we train a total of *44 sentiment classifiers* divided into the following categories as listed below: -

1) **Flat classifiers:** We use recurrent-based DNN models like GRU [3] and LSTM [13] to construct the *Flat classifiers* as follows: GRU, BiGRU, LSTM, and BiLSTM. Each model contains one hidden layer with 128 units and provides a measure of a general DNN model performance on the sentiment classification task for our dataset.
2) **Baseline classifiers:** We then create *Baseline classifiers* by coupling an IML method like Rule-Mask Mechanism [9] with each of the DNN model. We train a total of 16 classifiers by creating multiple configurations as follows {GRU, BiGRU, LSTM, and BiLSTM} Rule-masks $\times$ {GRU, BiGRU, LSTM, and BiLSTM} DNN models. This is done to provide the best configuration of the



(a) Distribution of *Covid19-twitter* dataset.

Rules	Positive Contrast	Positive No-contrast	Negative Contrast	Negative No-contrast
$a-but-b$	9135	7091	17665	9002
$a-yet-b$	490	441	1072	761
$a-though-b$	962	443	625	268
$a-while-b$	1509	814	1949	1057

(b) Individual CDR distributions in the dataset.

Figure 3: Description of Covid-19 tweets dataset. The 1st layer distribution denotes the tweets containing negative and positive sentiment labels. In the 2nd layer, *Rule* distribution denotes tweets having either one of the CDR-syntactic structures as shown in Table 1, and *No-rule* distribution contains tweets with no syntactic structures. In the last layer, *Contrast* distribution in *Rule* subsets denotes tweets containing a CDR, and *No-contrast* distribution denotes tweets without a CDR.

RMM method for each DNN model. Krishna et al. [16] hypothesized that constructing contextual word embeddings [24] from input sentences and feeding them to the DNN model can inherently capture complex linguistic structures like CDRs. Hence, we also use Large Language Models like BERTweet [22] and GPT-2 [26] to construct contextual word embeddings that are given input to the DNN models.

3) **MARKS-mech classifiers:** Similar to the RMM method, we create a total of 16 classifiers for our MARKS-mech to provide an exhaustive empirical analysis. We create multiple configurations like *GRU-mask+GRU-contrast*, where we use a GRU sequence layer in Rule-mask Blocks and Contrast Blocks. Again, this is done to propose the best configuration of our method for each DNN model.

TABLE 2: **Sentiment Accuracy** results for the sentiment classifiers on the *Rule* subset of dataset under-study.

Rule subset				
Sentiment Classifiers	**DNN models**			
	GRU	**BiGRU**	**LSTM**	**BiLSTM**
Flat classifiers	0.95	0.955	0.925	0.93
Baseline classifiers				
BERTweet [22]	0.963	0.959	0.959	0.962
GPT-2 [26]	**0.965**	**0.965**	**0.969**	**0.966**
RMM classifiers [9]				
GRU-mask	0.944	0.946	0.94	0.937
BiGRU-mask	0.93	0.938	0.94	0.941
LSTM-mask	0.945	0.948	0.936	0.94
BiLSTM-mask	0.95	0.935	0.939	0.941
MARKS-mech classifiers				
GRU-mask+GRU-contrast	0.946	0.944	0.943	0.945
BiGRU-mask+BiGRU-contrast	0.95	0.951	0.941	0.944
LSTM-mask+LSTM-contrast	0.948	0.945	0.943	0.939
BiLSTM-mask+BiLSTM-contrast	0.946	0.947	0.937	0.94

4.3. Metrics

We use the **Sentiment Accuracy** to quantify the sentiment classification performance of classifiers and a recently developed metric called **Post-hoc Explanation based Rule ConsistencY (PERCY)** score [10] to assess their CDR dissemination performance. Often, Sentiment Accuracy is used in the literature to assess both performances, but Gupta et al. [10] show that it is misleading to assess the CDR dissemination performance. For example, a classifier may correctly predict the sentiment of the sentence "the casting was not bad *but* the movie was awful" as negative but may base its decision on *individual negative words* like "not" in the "a" conjunct instead of using the "b" conjunct. Thus, we use the PERCY score to specifically quantify the CDR dissemination performance of sentiment classifiers. PERCY uses *feature attribution-based* explanation frameworks like LIME [27] to calculate the contribution of each conjunct to the classifier prediction. The conjunct with the highest contribution score determines the basis of classifier prediction.

5. Performance Analysis

5.1. Sentiment Classification Performance

In Table 2, we show the sentiment accuracy results and find that our method performs better compared to flat classifiers and provides comparable performance to the baseline classifiers. We further note that the bidirectional configurations of our method provide the best performance among all other configurations, which means that bidirectional models can learn the CDRs better than uni-directional ones. In particular, the performance improvement over flat classifiers confirms our hypothesis that CDRs like "a-but-b" need to be learned by a model in order to provide better empirical performance.

5.2. CDR Dissemination Performance

In Table 3, we show the PERCY score results and find that our method outperforms all the classifiers. This implies that classifiers constructed from our

TABLE 3: **PERCY scores** for the sentiment classifiers *Rule* subset of dataset under-study.

Rule subset				
Sentiment Classifiers	**DNN models**			
	GRU	**BiGRU**	**LSTM**	**BiLSTM**
Flat classifiers	0.114	0.104	0.093	0.092
Baseline classifiers				
BERTweet [22]	0.108	0.109	0.110	0.109
GPT-2 [26]	0.098	0.094	0.095	0.097
RMM classifiers [9]				
GRU-mask	0.099	0.091	0.095	0.088
BiGRU-mask	0.113	0.103	0.114	0.102
LSTM-mask	0.101	0.097	0.099	0.091
BiLSTM-mask	0.115	0.106	0.111	0.109
MARKS-mech classifiers				
GRU-mask+GRU-contrast	0.093	0.109	0.118	0.093
BiGRU-mask+BiGRU-contrast	0.114	0.127	**0.129**	**0.13**
LSTM-mask+LSTM-contrast	0.115	0.113	0.121	0.116
BiLSTM-mask+BiLSTM-contrast	**0.13**	**0.128**	0.118	0.124

TABLE 4: Anecdotal example to demonstrate the CDR dissemination performance of our method. In each conjunct, we highlight tokens based on their contribution to the sentiment prediction.

Classifier	Sentence
BERTweet GRU	me and my summer camp job that wouldve started this week **but** got cancelled in march
BiLSTM-mask+BiLSTM-contrast GRU	me and my summer camp job that wouldve started this week **but** got cancelled in march

TABLE 5: Anecdotal example to demonstrate that Sentiment Accuracy and PERCY scores are not correlated. In each conjunct, we highlight tokens based on their contribution to the sentiment prediction.

Sentences	Ground truth sentiment
absolutely right sad sad loss **but** the gentleman died of pneumonia another statistic for the covid regime its a joke	Negative

method can better identify the CDR on input sentences and pass the dominant-conjunct information to the DNN model. Further, we observe that the bidirectional configurations - BiGRU-mask+BiGRU-contrast and BiLSTM-mask+BiLSTM-contrast - of our method perform the best which implies that bidirectional models can learn the CDRs better than the unidirectional ones.

In Table 4, we visualize the feature attribution scores of an example predicted by the BERTweet GRU classifier and the BiLSTM-mask+BiLSTM-contrast GRU classifier. As we observe, our method better enables the GRU model to weigh the decision on "b" conjunct.

Note that the classifiers providing a high sentiment accuracy may not provide high PERCY score values as both metrics assess different tasks - Sentiment Accuracy assesses the performance on sentiment classification task while the PERCY scores provide an assessment of the knowledge dissemination task i.e. how effectively the classifiers are able to identify CDRs and base their decisions correctly about sentence sentiment as per the dominant conjunct. We show an anecdotal example in Table 5 where the classifier can provide a *correct sentiment decision* but based on the *wrong conjunct*. We observe that it is using some individual negative words in the "a" conjunct to base its decision.

TABLE 6: Rule-mask m accuracy results for different configurations of the RMM method and our method.

Rule subset				
Sentiment Classifiers	**DNN models**			
	GRU	**BiGRU**	**LSTM**	**BiLSTM**
RMM classifiers [9]				
GRU-mask	0.353	0.347	0.331	0.319
BiGRU-mask	0.479	0.511	0.487	0.476
LSTM-mask	0.361	0.345	0.351	0.352
BiLSTM-mask	0.5	0.524	0.539	0.529
MARKS-mech classifiers				
GRU-mask+GRU-contrast	0.617	0.612	0.616	0.627
BiGRU-mask+BiGRU-contrast	**0.721**	**0.719**	**0.704**	**0.722**
LSTM-mask+LSTM-contrast	0.603	0.592	0.609	0.586
BiLSTM-mask+BiLSTM-contrast	0.699	0.68	0.692	0.686

5.3. Rule-mask prediction performance

Our method predicts a rule-mask output, which is applied to the input sentence to extract features consistent with the logic rule. For a sentence $s = [t_1, t_2, \cdots, t_n]$ as an ordered sequence of n tokens, given the actual rule-mask value $y = [y_1, y_2, \cdots, y_n]$ as an ordered sequence of n values, and its prediction value as $p(y|s) = [p_{\theta_1}(y_1|t_1), p_{\theta_1}(y_2|t_2), \cdots, p_{\theta_1}(y_n|t_n)]$, we calculate the rule-mask prediction accuracy as:

$$o(\mathbf{s}) = \begin{cases} 1, & \text{if } (p(y|s) = y) \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

For a collection of sentences $\mathbb{S}$, rule-mask accuracy is calculated as follows:

$$o(\mathbb{S}) = \frac{1}{|\mathbb{S}|} \sum_{i=1} o(\mathbf{s}_i) \tag{5}$$

In Table 6, we show the rule-mask accuracy of the RMM method and our MARKS-mech method. We find that our method significantly outperforms the RMM method, which implies that the RMM method confuses between sentences containing a CDR and just a CDR-syntactic structure, hence outputs the incorrect rule-masks. The rule-mask correction module in our method works as intended and thus, our method can better recognize the CDRs in input sentences and can correctly pass the dominant conjunct to the DNN.

6. Conclusion

We presented a novel IML method called MARKS-mech to disseminate logical prior knowledge in DNN models. Our method employs *Feature Manipulation* which transforms input data to represent the prior knowledge on its features. We test our method on the sentence-level sentiment classification task and disseminate complex linguistic relations called Contrastive Discourse Relations (CDRs) in DNNs. For our experiments, we utilize a recently constructed Twitter-based dataset specifically designed to test the CDR dissemination performance of IML methods. We conduct a thorough analysis of our method by creating its multiple configurations and calculating multiple metrics, to assess both its sentiment classification and CDR dissemination performances. Our

results demonstrate that it provides superior CDR dissemination performance compared to existing methods in the literature. As our method is simple, intuitive, and model-agnostic, it can be easily utilized by ML practitioners to create IML models.

7. Limitations and Future work

Currently, our method is only applicable to logical rules, which are based on structures like *a-but-b*, and does not generalize to other syntactic structures or linguistic phenomena. In the future, we plan to upgrade it to accommodate more complex representations of linguistic knowledge.

References

[1] Agarwal, R., Prabhakar, T.V., Chakrabarty, S.: "i know what you feel": Analyzing the role of conjunctions in automatic sentiment analysis. In: Proceedings of the 6th International Conference on Advances in Natural Language Processing (2008)

[2] Besold, T.R., d'Avila Garcez, A., Bader, S., Bowman, H., Domingos, P., Hitzler, P., Kuehnberger, K.U., Lamb, L.C., Lowd, D., Lima, P.M.V., de Penning, L., Pinkas, G., Poon, H., Zaverucha, G.: Neural-symbolic learning and reasoning: A survey and interpretation (2017)

[3] Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: NIPS (2014)

[4] Dash, T., Chitlangia, S., Ahuja, A., Srinivasan, A.: A review of some techniques for inclusion of domain-knowledge into deep neural networks. Scientific Reports (2022)

[5] Devlin, J., Chang, M.W., Lee, K., 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 (2019)

[6] Ganchev, K., Graça, J., Gillenwater, J., Taskar, B.: Posterior regularization for structured latent variable models. Journal of Machine Learning Research (2010)

[7] Garcez, A.S.d., Broda, K., Gabbay, D.M., et al.: Neural-symbolic learning systems: foundations and applications. Springer Science & Business Media (2002)

[8] Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)

[9] Gupta, S., Bouadjenek, M.R., Robles-Kelly, A.: A mask-based logic rules dissemination method for sentiment classifiers. In: ECIR (2023)

[10] Gupta, S., Bouadjenek, M.R., Robles-Kelly, A.: Percy: A post-hoc explanation-based score for logic rule dissemination consistency assessment in sentiment classification. Knowledge-Based Systems (2023)

[11] Gürel, N.M., Qi, X., Rimanic, L., Zhang, C., Li, B.: Knowledge enhanced machine learning pipeline against diverse adversarial attacks. In: Proceedings of the 38th International Conference on Machine Learning (2021)

[12] Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NIPS Deep Learning and Representation Learning Workshop (2015)

[13] Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9**(8), 1735–1780 (1997)

[14] Hu, Z., Ma, X., Liu, Z., Hovy, E., Xing, E.: Harnessing deep neural networks with logic rules. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (2016)

[15] Hu, Z., Yang, Z., Salakhutdinov, R., Xing, E.: Deep neural networks with massive learned knowledge. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (2016)

[16] Krishna, K., Jyothi, P., Iyyer, M.: Revisiting the importance of encoding logic rules in sentiment classification. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (2018)

[17] Krupka, E., Tishby, N.: Incorporating prior knowledge on features into learning. In: Meila, M., Shen, X. (eds.) Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 2, pp. 227–234. PMLR, San Juan, Puerto Rico (21–24 Mar 2007)

[18] Lakoff, R.: If's, and's and but's about conjunction. In: Studies in Linguistic Semantics (1971)

[19] Li, T., Srikumar, V.: Augmenting neural networks with first-order logic. In: ACL (2019)

[20] Mukherjee, S., Bhattacharyya, P.: Sentiment analysis in twitter with lightweight discourse analysis. In: COLING (2012)

[21] Nguyen, A.M., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. CoRR (2014)

[22] Nguyen, D.Q., Vu, T., Tuan Nguyen, A.: BERTweet: A pre-trained language model for English tweets. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (2020)

[23] O'Shea, K., Nash, R.: An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458 (2015)

[24] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., 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) (2018)

[25] Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., Webber, B.: The Penn Discourse TreeBank 2.0. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08) (2008)

[26] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. Tech. rep., OpenAI (2018)

[27] Ribeiro, M.T., Singh, S., Guestrin, C.: "why should i trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016)

[28] Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence (2019)

[29] von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., Walczak, M., Garcke, J., Bauckhage, C., Schuecker, J.: Informed machine learning – a taxonomy and survey of integrating prior knowledge into learning systems. IEEE Transactions on Knowledge and Data Engineering (2023)

[30] Tang, D.: Sentiment-specific representation learning for document-level sentiment analysis. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (2015)

[31] Yu, D., Yang, B., Liu, D., Wang, H., Pan, S.: A survey on neural-symbolic learning systems (2023)

[32] Zirn, C., Niepert, M., Stuckenschmidt, H., Strube, M.: Fine-grained sentiment analysis with structural features. In: Proceedings of 5th International Joint Conference on Natural Language Processing (2011)