A Survey for Fusing Structured Knowledge Priors for NER

CSCI 699 Intro2IE Course

Yuchen Lin
University of Southern California
Los Angeles, CA
yuchen.lin@usc.edu

Woojeong Jin
University of Southern California
Los Angeles, CA
woojeong.jin@usc.edu

1 Our Goal

Named Entity Recognition (NER) is the most fundamental task in information extraction (IE). However, in many domains like social media text (e.g., tweets) and online user-generated content (e.g., web forum, reviews), only a limited amount of labeled data are available for training neural models (and are expensive and slow to create).

Such neural models require a significant amount of training examples to learn simple rules and facts about real world (e.g., actor is artist and artist is person), which is cost-inefficient in consuming human annotation efforts. Moreover, the existing approaches can hardly deal with cases that need deep understanding of hidden semantics.

Our goal is thus to explicitly impose the simple rules and patterns from domain experts as a part of model architecture, and construct neural networks for fusing model structures with prior structures. In this way, simple domain-specific rules and patterns need no data to learn from data, and massive knowledge facts can directly provide additional signals for hidden semantics. Thus, the data can be used more efficiently and tracking the priors in the model can interpret predictions. In the following sections, we will present some previous related works.

2 Previous works

2.1 Input Stage

Recently, there are a few existing works investigating how to impose structured priors at the input stage or output stage. For the input stage, one of the most notable works is the data programming paradigm (Ratner et al., 2016) and its data-creation system Snorkel, which consider structured priors as “labeling functions” and use them to heuristically annotate unlabeled data for future training. For example, a labeling function can label a sentences mentioning chemical A and disease B as relation triple (A, CAUSE, B) if the sentence matches the regular expression “.*cause.*” or the triple is already existent in knowledge bases.

Our prior research also investigated how to capture the expertise subsets of different labeling functions (i.e. heterogeneity of labeling functions) (Liu et al., 2017). However, these generated pseudo labels from structured priors can be very noisy, and the noise can hardly be eliminated in following model-training procedures. Moreover, it does not help models learn or interpret which priors are more useful for each instance.

2.2 Output Stage

Focusing on the output stage, recent research investigates using integer linear programming to constrain model training such that the predictions must follow a set of predefined logical rules (Wang et al., 2015; Guo et al., 2018). For instance, in knowledge graph completion, implication rule between relations (e.g., HasWife → HasSpouse) are used as constraints in the objective functions (x_i^{r_1} ≤ x_i^{r_2}, \forall r_1 \rightarrow r_2, \forall i, \forall j). However, such methods
cannot apply to other tasks as they are hard to adapt to other types of structured priors. Moreover, solely constraining loss functions serves as a post-processing step and is not adequate for fusing structured priors at the model level.

2.3 State-of-the-art BLSTM-CRF family model for NER

The state-of-the-art neural architectures for NER are in the LSTM-CRF family, such as BLSTMs-CRFs (Lample et al., 2016), CNN-BLSTM-CRFs (Ma and Hovy, 2016), and LM-LSTM-CRF (Liu et al., 2018). Furthermore, BLSTMs-CNNs was proposed (Chiu and Nichols, 2015). Lample et al. (Lample et al., 2016) used two new neural architectures—one based on bidirectional LSTMs and conditional random fields, and the other that constructs and labels segments using a transition-based approach inspired by shift-reduce parsers. Their models rely on two sources of information about words: character-based word representations learned from the supervised corpus and unsupervised word representations learned from unannotated corpora. Ma and Hovy (Ma and Hovy, 2016) proposed an end-to-end neural network architecture that benefits from both word- and character-level representations automatically, by using combination of bidirectional LSTM, CNN and CRF. Liu et al. (Liu et al., 2018) proposed a sequence labeling framework, LM-LSTM-CRF, which extracts knowledge from raw texts and empower the sequence labeling task. Besides word-level knowledge contained in pretrained word embeddings, character-aware neural language models are incorporated to extract character-level knowledge. Chiu and Nichols (Chiu and Nichols, 2015) presented a neural network architecture that automatically detects word- and character-level features using a hybrid bidirectional LSTM and CNN architecture. These end-to-end models do eliminate the need of hand-crafted features. However, these models require additional labeled data annotated by human experts when adapting to a new domain. Lin and Lu (Lin and Lu, 2018) found their performance typically degrades dramatically in domains with weak supervision (noisy and less data) or on new relation types. Moreover, it is hard to incorporate the aforementioned structured priors, especially for the knowledge graph facts, into CRF layers. Thus, these architectures cannot provide explainable predictions.

2.4 Limitations and Challenges.

We propose a principled model programming approach, called ReKnow, that can encode multiple types of structured priors as part of the neural model structures using graph neural networks, and apply it to two important IE tasks: named entity recognition and relation extraction. The ReKnow aims to address the following challenges:
a) how to represent various structured priors in a unified way for different sequence encoding layers; b) how to dynamically compose graph neural networks to encode knowledge structures; c) how to softly aligning priors with input sequence as well as assembling multiple priors. We propose to represent heuristic RuleEs and structured KNOWledge as nodes and edges in a relation graph neural network (GNN) layer, and attach the GNN layer over sequence encoding layer to fuse knowledge priors at the model level. Our study is relevant to recent work on modifying neural model architectures to integrate linguistic priors for sentence representation learning (Tai et al., 2015; Marcheggiani and Titov, 2017; Peng et al., 2017). By contrast, ReKnow models knowledge priors as sub-structures in the GNN layer with multi-relational nodes and edges, instead of simply connecting the hidden states in sequence encoders following linguistic structures.

2.5 Model Stage: Relational Inductive Biases

To formally study the problem, we propose to consider the model constraints including structured knowledge as a form of inductive bias. In deep neural networks, we further regard the structured knowledge as relational inductive biases.

To introduce the concept of inductive bias, let’s start with a simple example of induction. Imagine that the first apple we saw is red. It does not follow necessarily that the next apple we will see must be red too, but “red” seems to be a better guess given the limited training data. Behind this guess, there is an assumption: “all apples should be the same color”, which is an instance of inductive bias in the learning process. An inductive bias can be considered as a set of assumptions that a machine learning model leverages to predict the outputs of given unseen inputs (Mitchell, 1980). The assumptions are usually about data-generating process or solution spaces, and they either become regularization terms to avoid over-fitting (e.g. $L_2$ regularization) or directly influence the design of model architectures (e.g. SVMs assume distinct classes tend to be separated by wide boundaries).

Relational inductive biases in building neural networks As for deep learning models, there are many standard building-blocks such as fully-connected layers in MLPs, convolutional layers in CNNs, and recurrent layers in RNNs, and we can find that they often carry implicit relational assumptions (Battaglia et al., 2018). To formulate these relational inductive biases, we have to clarify several key concepts first: entities, relations, and rules.

Entities are distributed representations in networks, such as units in fully-connected layers, grid elements in convolutional layers, and timesteps in recurrent layers. Relations are the way entities interact with each other for further computation in networks (e.g. the relation of fully-connected layers is all-to-all; the relation of convolutional layers is local; the relation of recurrent layers is sequential). Rules are neural network functions that take entities and relations as arguments; the rules are often specified by the weights and bias terms.

A relational inductive bias of a deep learning model is basically a set of assumptions formed by the rule functions with entities as well as relations. To explore relational inductive biases of different neural architectures, we should pay attention to: 1) the entities and relations in specific networks, 2) how the rules are shared/reused across different parts in networks, and 3) how the network applies rules to process the entities. For convolutional layers, the shareable rule functions with grid-element entities offer the important relational inductive biases: “locality” and “translation invariance”; for recurrent layers, the reused rules over sequential timesteps also carry a Markovian locality bias.

To sum up, relational inductive biases provide a way of applying prior assumptions about the relational structures in developing neural networks. These assumptions may be some model design principles, while they can also come from some prior hand-crafted rules and structured knowledge. Relational inductive biases are essential for relational reasoning in complex understanding and inference. Incorporating suitable relation inductive biases is thus a promising bridge between “end-to-end deep learning” and “hand-engineering” with external structured knowledge.

3 Conclusion

To better impose such structured priors in neural IE models, we propose a principled approach called ReKnow that can model the structured priors directly at the model level. Specifically, we fuse the heuristic RuleEs and structured KNOWledge into neural network architectures with graph networks.
References


Liyuan Liu, Xiang Ren, Qi Zhu, Shi Zhi, Huan Gui, Heng Ji, and Jiawei Han. 2017. Heterogeneous supervision for relation extraction: A representation learning approach. In EMNLP.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. AAAI.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In ACL.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In EMNLP.


Quan Wang, Bin Wang, and Li Guo. 2015. Knowledge base completion using embeddings and rules. In IJCAI.