Stony Brook University

SpecNFS: A Challenge Dataset Towards Extracting Formal Models from Natural Language Specifications

Sayontan Ghosh, Amanpreet Singh, Alex Merenstein, Wei Su, Scott Smolka, Erez Zadok, Niranjan Balasubramanian

Agenda

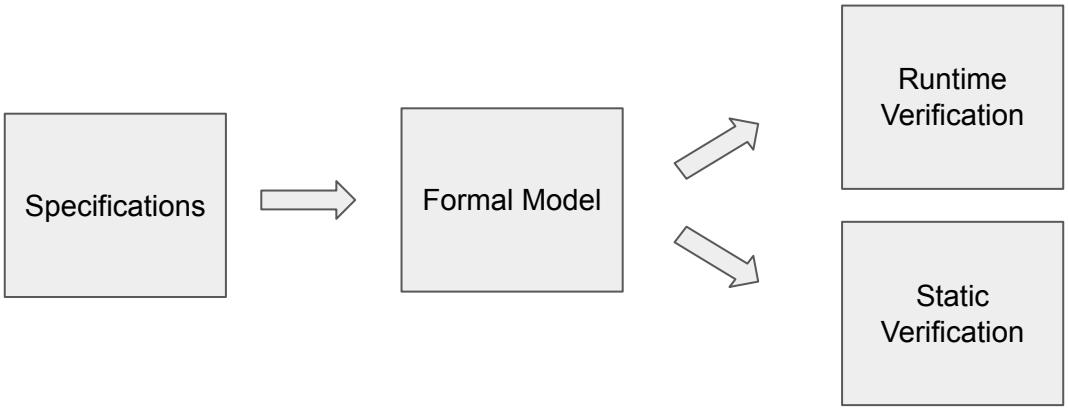
1. Motivation: What can NLP do for formal modeling

2. SpecNFS: Representation, dataset construction and quality

3. Benchmarking: Semantic dependency parsing baselines

4. Challenges and discussion

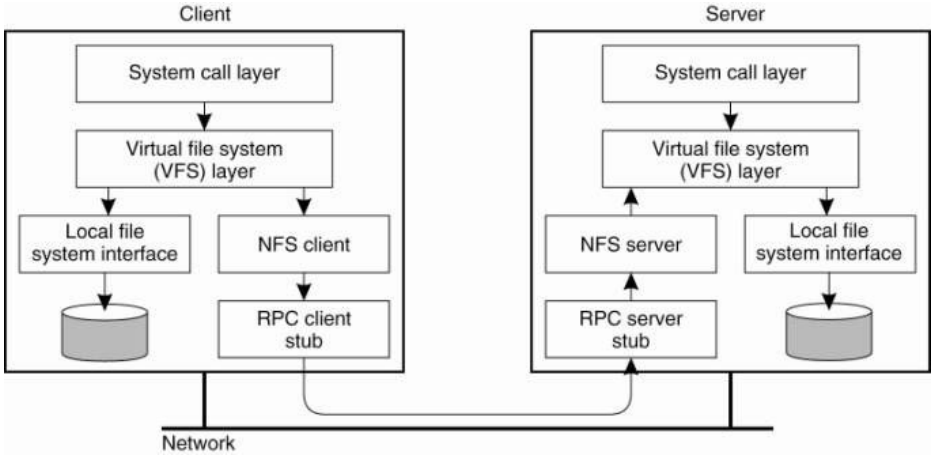
Verifying Software Systems



Network File Systems (NFS)

- Critical infrastructure
- Complex
- Ever-evolving

Network File System Overview

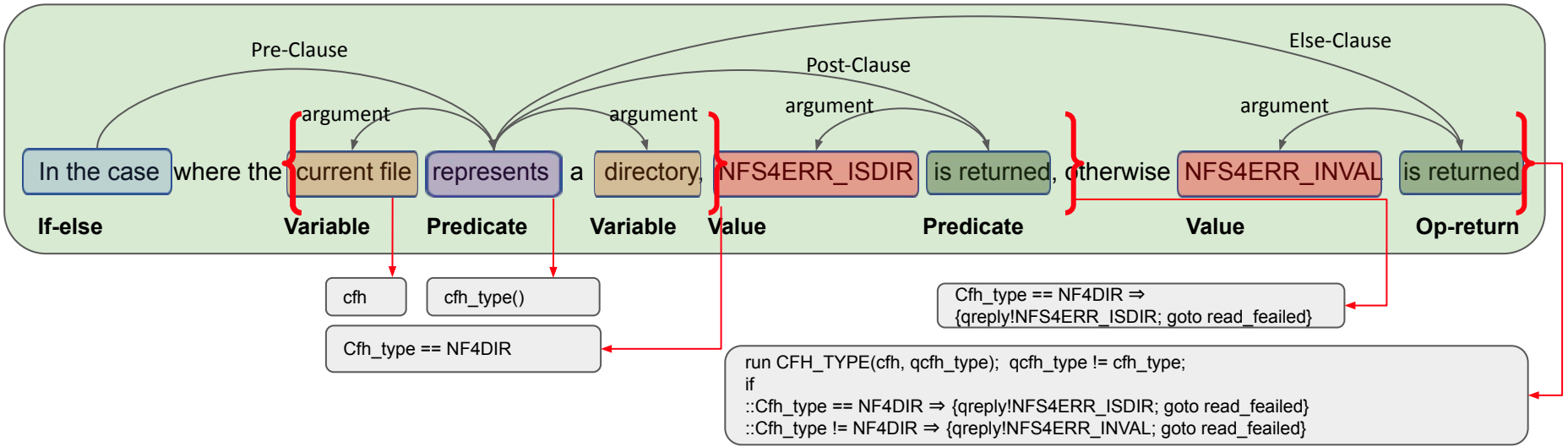


How much can NLP assist in this process?

Abstracting formal representation from specification sentences

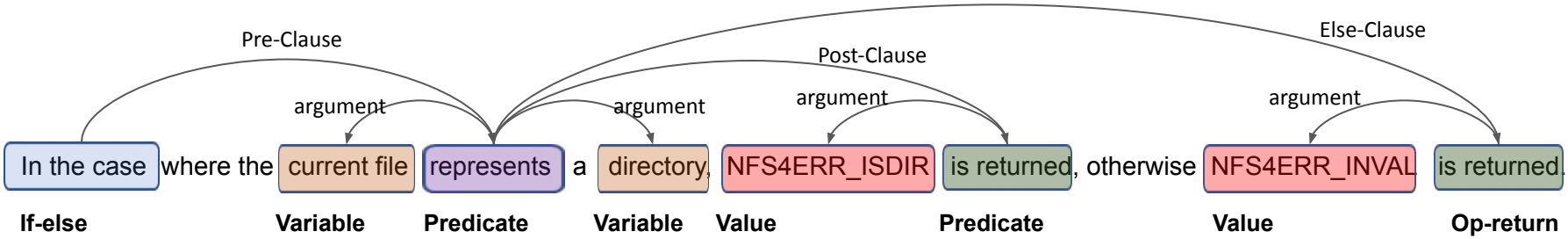
In the case where the current file represents a directory, NFS4ERR_ISDIR is returned, otherwise NFS4ERR_INVAL is returned.

PRE / IF	current file represents a directory
POST / THEN	NFS4ERR_ISDIR is returned
POST / ELSE	NFS4ERR_INVAL is returned

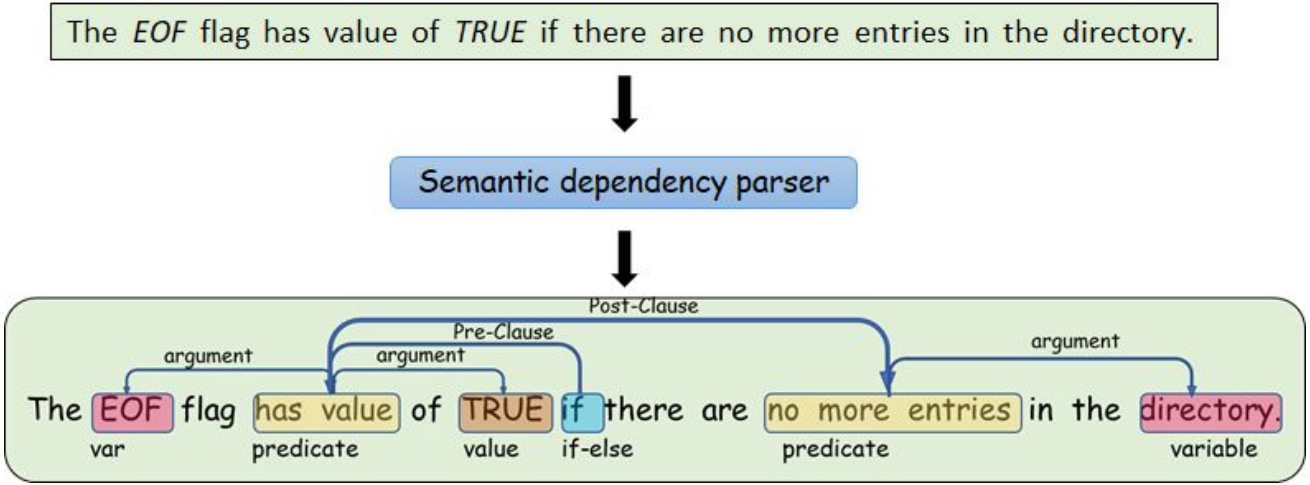


Representing Specification Sentences as Formal Statements

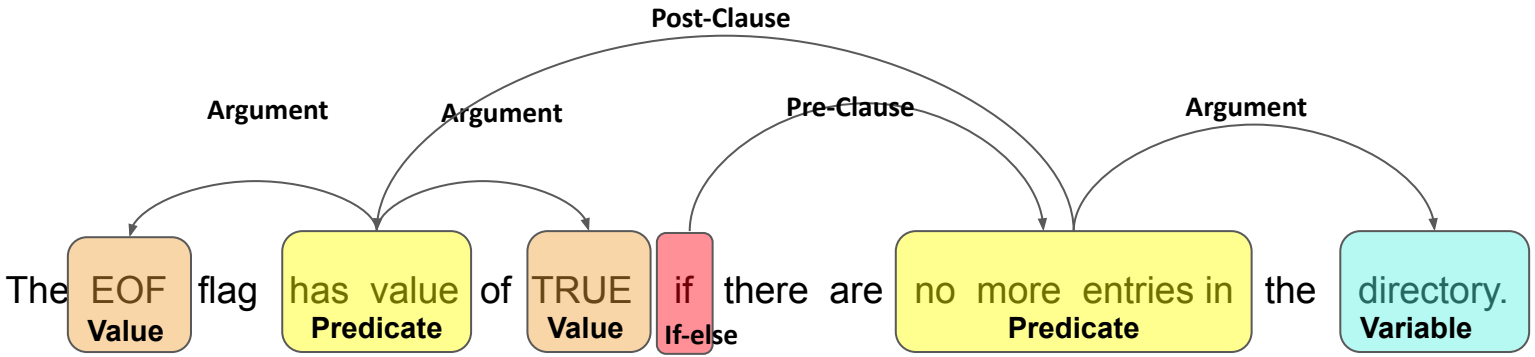
In the case where the current file represents a directory, NFS4ERR_ISDIR is returned, otherwise NFS4ERR_INVAL is returned.



SpecIR: An intermediate representation for specification sentences.



Elements of SpecIR



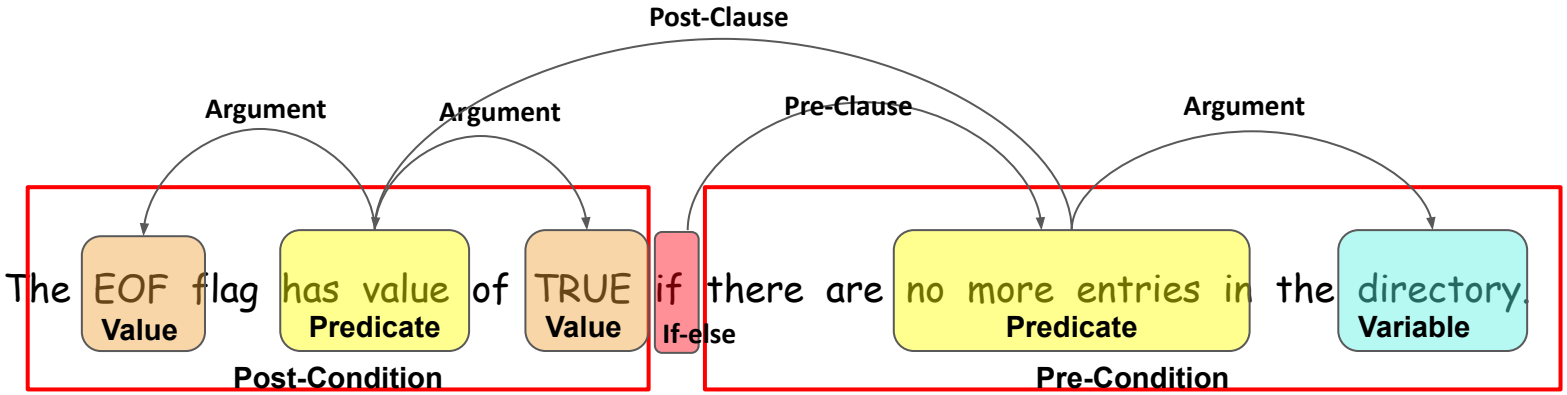
Span Labels

- Objects:
  - Value
  - Variable
- Op_name
- Function
- Op_return
- Predicate
- Connectives:
  - And
  - Or
  - If-Else

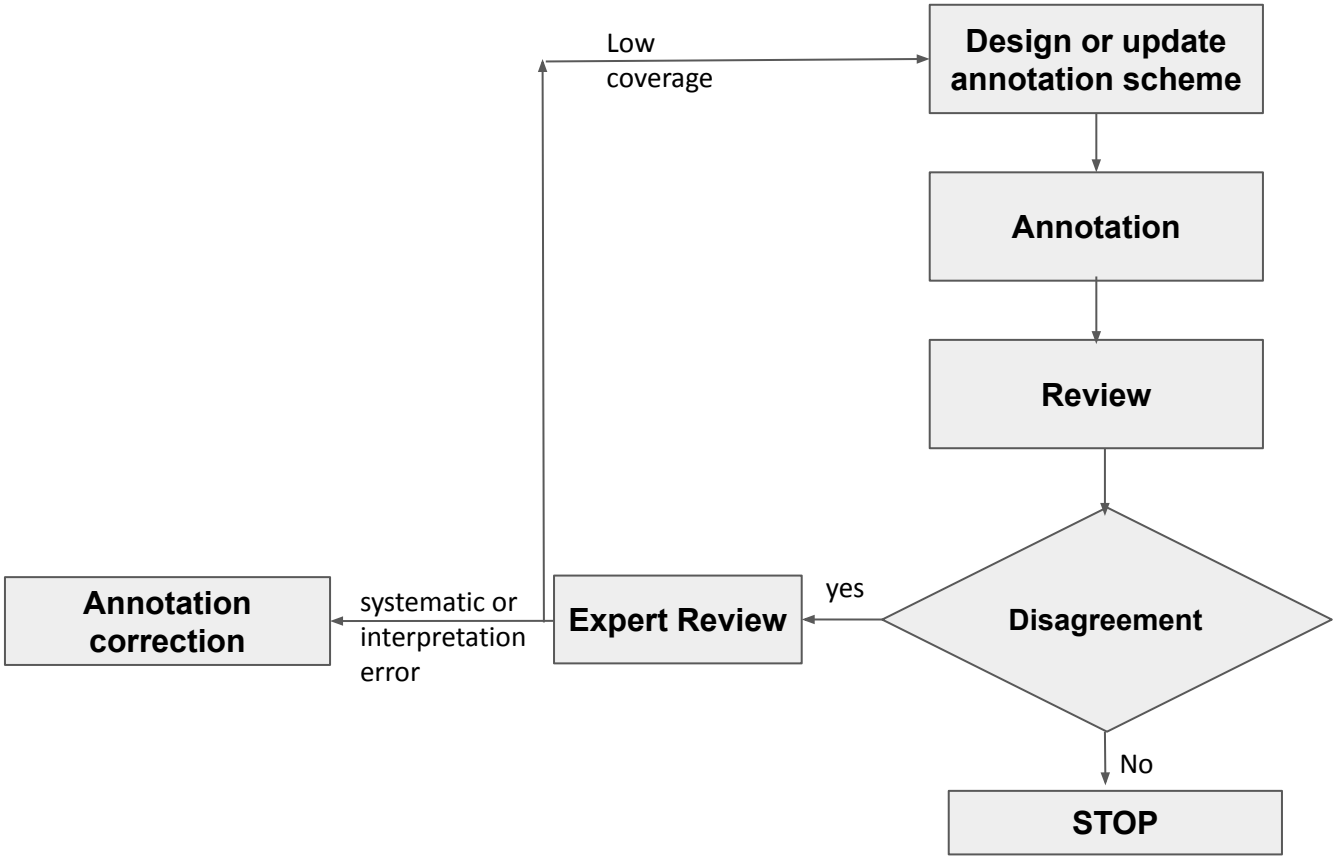
Dependency Link Labels

- Pre-Clause
- Argument
- Return-Val
- Post-Clause
- And-Or-Arg

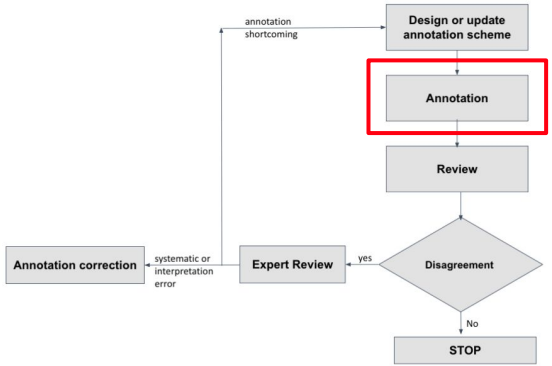
Elements of SpecIR



Annotation Process: Overview

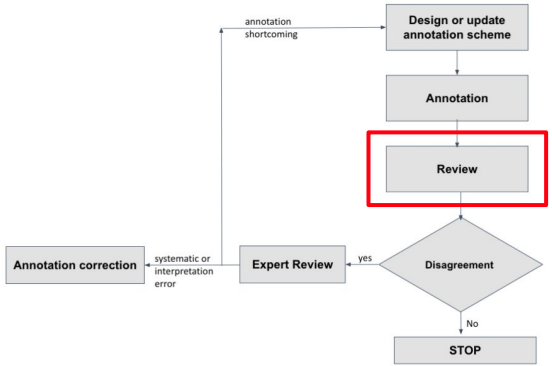


Annotation Process: Annotation



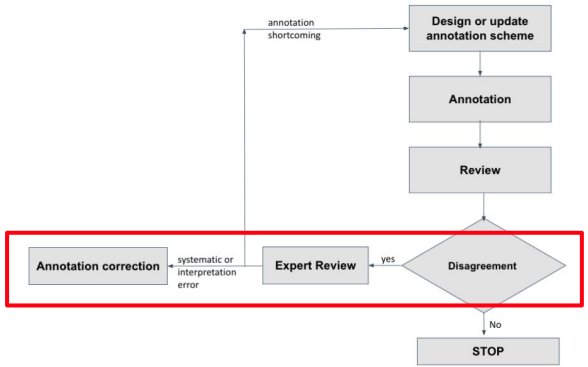
- Iterative approach: Incremental annotation and scheme update
- 5 CS graduate students as annotators and 2 domain experts.
- Annotation scheme designed by the domain experts.
- Annotators trained on using the annotation scheme to annotate specifications.

Annotation Process: Annotation Review



- The annotations were jointly reviewed by annotator and the expert
- Samples of annotations by the annotator were assigned to different reviews for review
- Disagreement between annotator and reviewer was delegator to the expert

Annotation Process: Disagreement Resolution

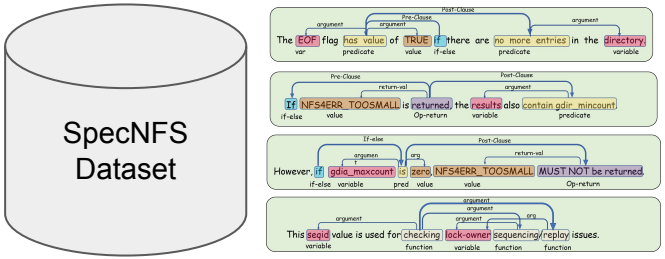


- Expert's decision was final in deciding the correct annotation
- Review of the reported disagreement by the expert also involved detecting
  - systematic errors made by any annotator
  - annotation scheme shortcomings, which led to updating the annotation scheme, if it was possible.
- In-case the annotation scheme was updated, the annotation process was again repeated.

SpecNFS: The annotated dataset

Dataset Statistics

1. No. of sentences : 1198
2. No. of operations covered : 40
3. No. of spans: 9358
4. No. of links: 6872



SpecNFS: Estimating Annotation Quality

- Compare against expert judgment

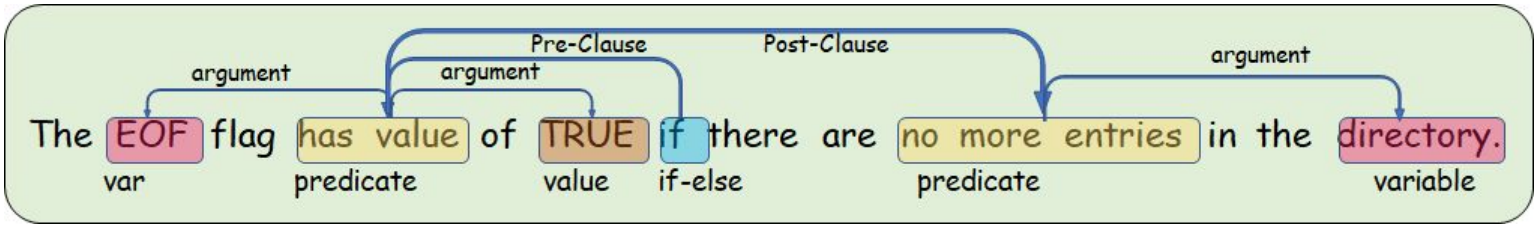
Type	#Analyzed	#Edits	Percentage
Entities	497	20	4%
Links	460	25	5.4%

- For each sentence
  - expert judged whether they agree with the annotation
  - for each discrepancy count edits for fix

Semantic Dependency Parsing: A two-stage pipeline

1. Span Labeling : Identify and label text spans referring to the entities

2. Dependency Link Prediction : Identify semantic dependency links between these entity spans



Benchmarking

1. How effective are LM based semantic dependency parsing tools

2. Does adapting to the NFS domain via domain and task transfer help

3. Do explicit relationship constraint help?

Span Labelling

- This is a standard sequence tagging task
- We benchmark sequence tagging solutions using four large language models:
- We also explore domain adaptation strategies that can utilize other unlabeled domain-relevant texts.

Model	**base**	**NFS-DAPT**	**NFS-TAPT**
BERT	59.5±0.3	59.5±0.3	59.5±0.3
DistilBERT	58.6±0.3	58.9±0.3	59.4±0.3
CodeBERT	59.1±0.3	59.3±0.4	60.1±0.4
CS RoBERTa	59.8±0.5	60.2±0.3	**60.5**±0.3

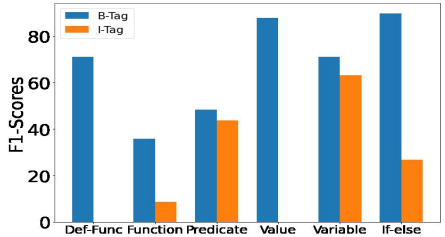
Benchmarking Dependency Link Prediction

Model class	Language Model	Base	Base + NER	Base + NER + DAPT	Base + NER + TAPT	Base + NER + TAPT + Link Constraints
Arc factored	BERT	16.1±1.7	32.0±1.1	31.7 ± 0.7	32.8 ± 2.0	**33.3 ± 2.1**
	DistilBERT	14.2±0.7	15.1±1.2	14.2 ± 0.7	15.5 ± 1.2	15.7 ± 1.2
	CS RoBERTa	6.3 ± 2.0	31.6±1.1	32.8 ± 0.08	32.1 ± 1.0	32.4 ± 0.01
	CodeBERT	18.0±2.1	32.0±0.8	30.8 ± 1.4	31.1 ± 1.3	31.7 ± 1.1

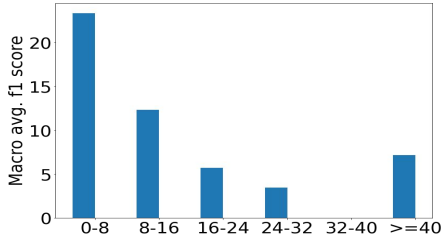
- **Arc factored** based dependency link prediction approach
- **Transferring knowledge** from the Span labelling task leads to significant performance gains.
- Using **DAPT** and **TAPT** trained models do not lead to any significant performance gain.
- Adding **link constraint** during inference leads to negligible performance gain (< 1%)

Error Analysis

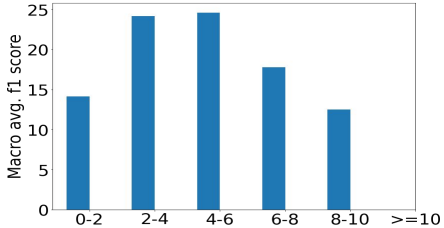
- Span Labeling
  - Functions vs Variables - *If the size specified is larger than the current **size of the file**, the file is "zero extended"*
  - Data Imbalance



- Dependency Link Prediction
  - Link Terminal Information Smoothing

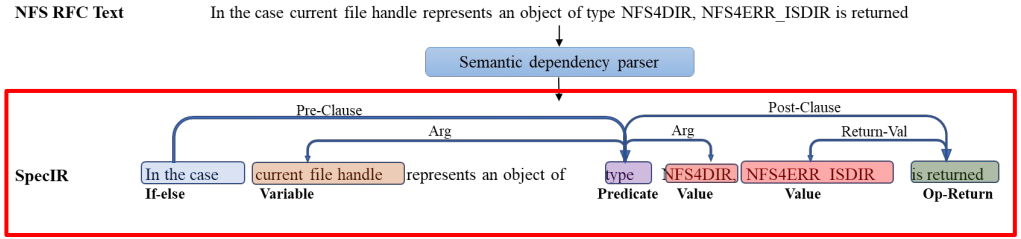


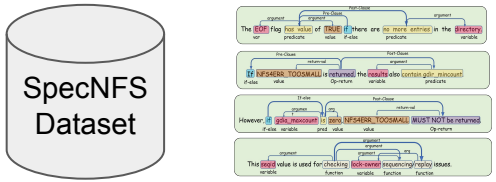
  - Long Term Dependency



Conclusion

- We introduced the problem of extracting formal models from NFS RFCs (towards formal model from NFS-RFC)
- We designed an intermediate representation, SpecIR
- We introduced a challenge dataset, SpecNFS
- Benchmarking and analysis (show challenges for NLP community)





Model	base	NFS-DAPT	NFS-TAPT
BERT			
DistilBER			
CodeBER			
CS RoBERTa			

Model class	Language Model	Base	Base + NER	Base + NER + DAPT	Base + NER + TAPT	Base + NER + TAPT + Link Constraints
Arc factored	BERT	16.1±1.7	32.0±1.1	31.7 ± 0.7	32.8 ± 2.0	**33.3 ± 2.1**
	DistilBERT	14.2±0.7	15.1±1.2	14.2 ± 0.7	15.5 ± 1.2	15.7 ± 1.2
	CS RoBERTa	6.3 ± 2.0	31.6±1.1	32.8 ± 0.08	32.1 ± 1.0	32.4 ± 0.01
	CodeBERT	18.0±2.1	32.0±0.8	30.8 ± 1.4	31.1 ± 1.3	31.7 ± 1.1