dfki-mlst @DialAM 2024

How to win the Shared Task?

Arne Binder and Tatiana Anikina

DFKI Saarbrücken, MLT Meeting 01.10.2024

Outline

DialAM 2024 Shared Task: What is this about?

Main part (Arne): System Description

1. Approach
2. Results

Bonus part (Tatiana):

1. Challenges
2. Additional Experiments
3. Recipe for Success @DialAM-2024

DialAM 2024 - The Data

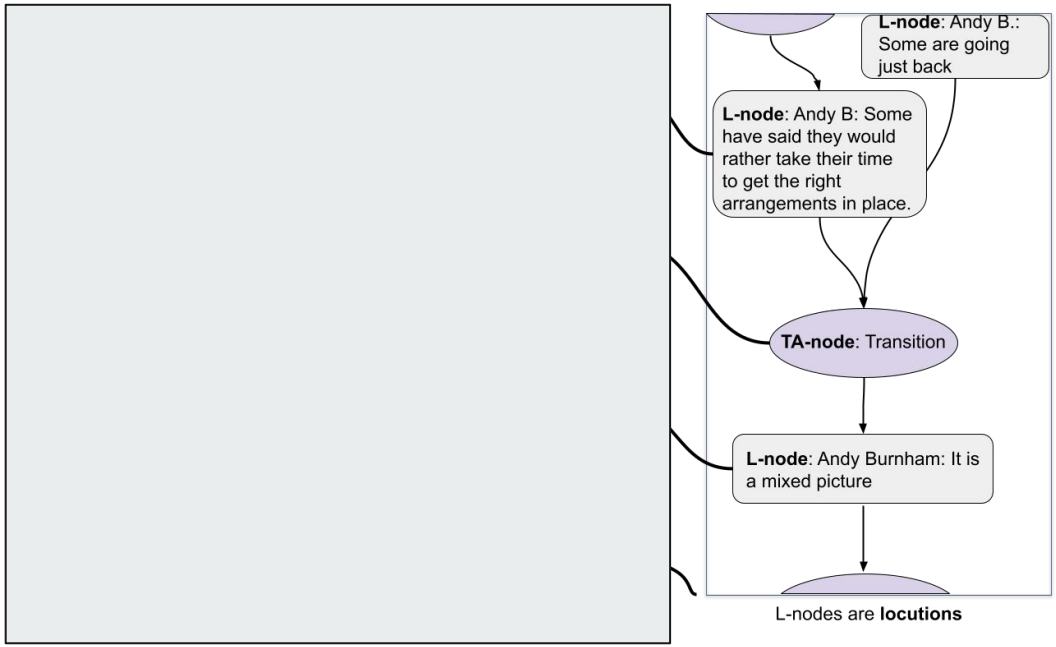
Argument Mining on Dialogue

DialAM 2024 - The Data

Argument Mining on Dialogue

Inference Anchoring Theory (IAT)

- **Dialogue** structure (right)

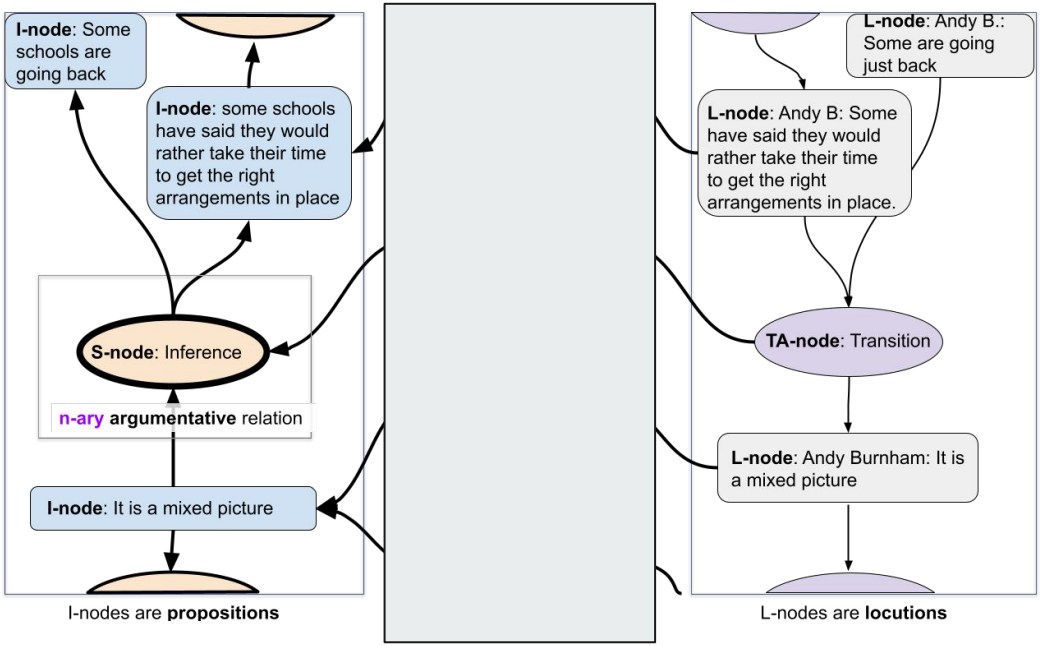


DialAM 2024 - The Data

Argument Mining on Dialogue

Inference Anchoring Theory (IAT)

- **Dialogue** structure (right)
- **Argument** structure (left)

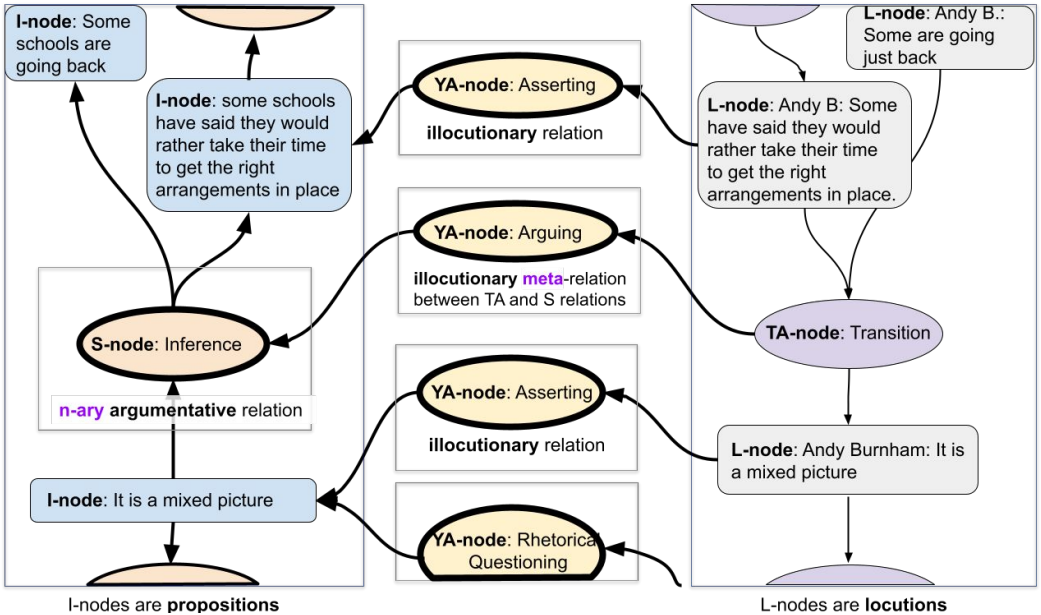


DialAM 2024 - The Data

Argument Mining on Dialogue

Inference Anchoring Theory (IAT)

- **Dialogue** structure (right)
- **Argument** structure (left)
- **Anchoring** (middle)



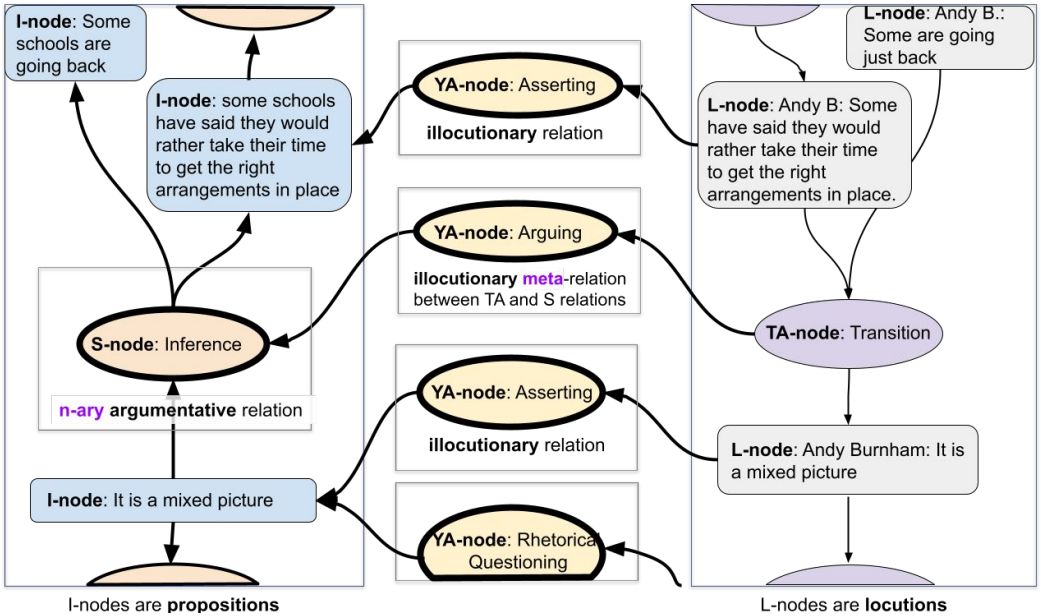
DialAM 2024 - The Task

Argument Mining on Dialogue

Inference Anchoring Theory (IAT)

- **Dialogue** structure (right)
- **Argument** structure (left)
- **Anchoring** (middle)

Focus on **Relation Extraction (RE)**



DialAM 2024 - The Task

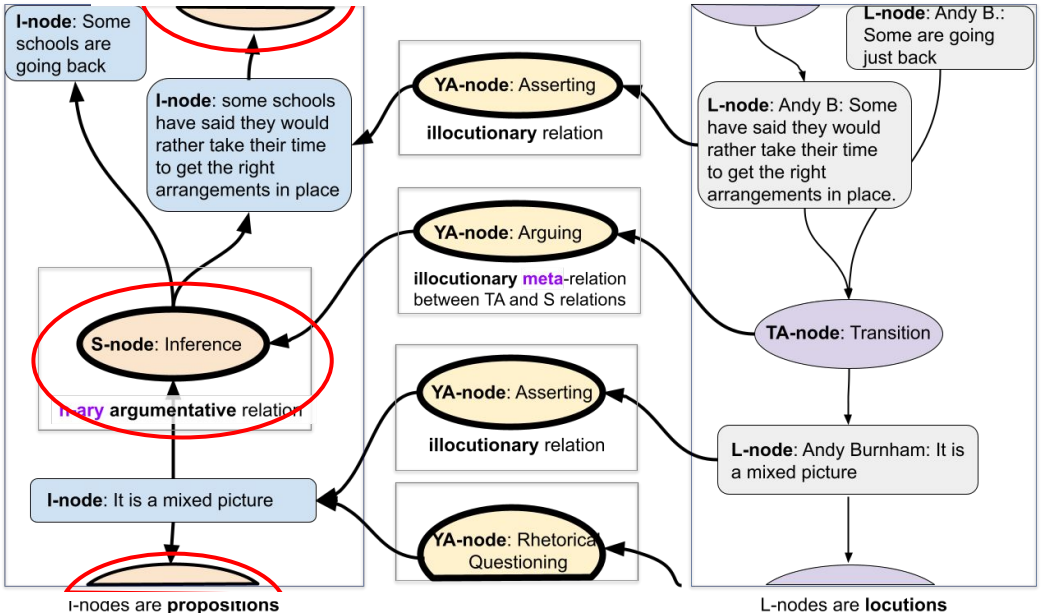
Argument Mining on Dialogue

Inference Anchoring Theory (IAT)

- **Dialogue** structure (right)
- **Argument** structure (left)
- **Anchoring** (middle)

Focus on **Relation Extraction (RE)**

1. **Argumentative** Relations (⬭)



DialAM 2024 - The Task

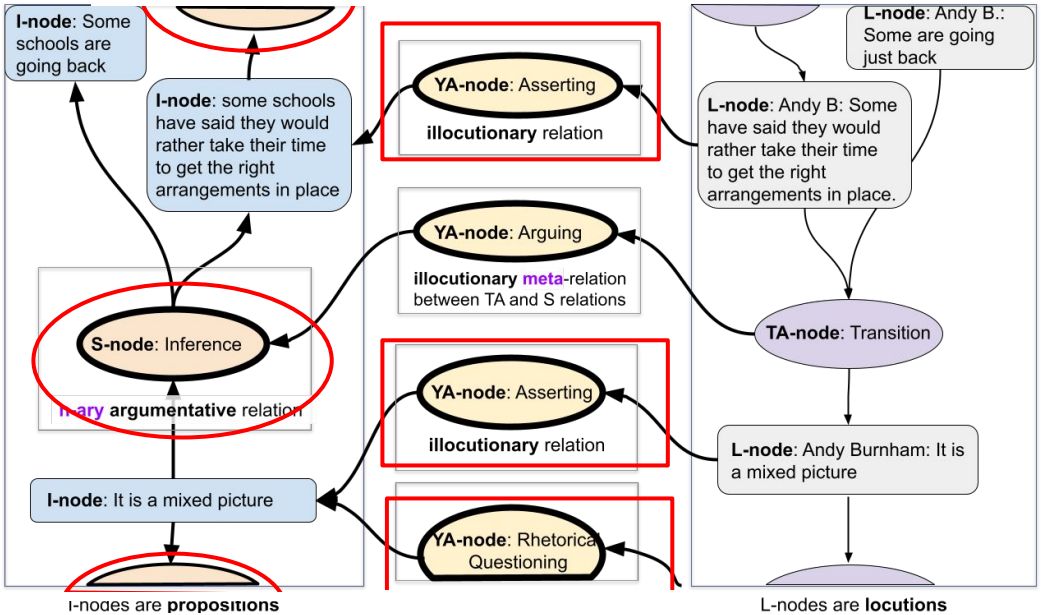
Argument Mining on Dialogue

Inference Anchoring Theory (IAT)

- **Dialogue** structure (right)
- **Argument** structure (left)
- **Anchoring** (middle)

Focus on **Relation Extraction (RE)**

1. **Argumentative** Relations (⬭)
2. **Illocutionary** Relations
   a. Anchoring **propositions** (▭)



DialAM 2024 - The Task

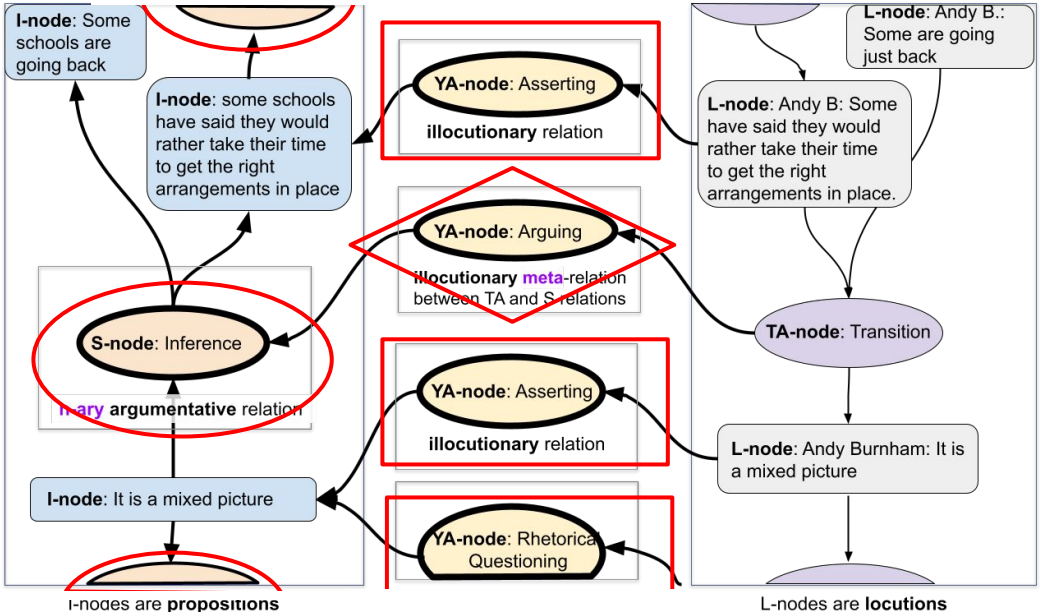
Argument Mining on Dialogue

Inference Anchoring Theory (IAT)

- **Dialogue** structure (right)
- **Argument** structure (left)
- **Anchoring** (middle)

Focus on **Relation Extraction (RE)**

1. **Argumentative** Relations (⬭)
2. **Illocutionary** Relations
   a. Anchoring **propositions** (▭)
   b. Anchoring **argumentative Relations** (◇)



System Description - Introduction

How does “classic” Relation Extraction work?

1. **Select** candidate pairs of entities

   “**Alice** loves **science** and **pizza**” -> (alice, science), (alice, pizza), (science, pizza)

System Description - Introduction

How does “classic” Relation Extraction work?

1. **Select** candidate pairs of entities

   “**Alice** loves **science** and **pizza**” -> (alice, science), (alice, pizza), (science, pizza)

2. **Classify** them

   (alice, science) ➡ “Alice loves science and pizza” ➡

   (alice, pizza) ➡ “Alice loves science and pizza” ➡

   (science, pizza) ➡ “Alice loves science and pizza” ➡

System Description - Introduction

How does “classic” Relation Extraction work?

1. **Select** candidate pairs of entities

   “**Alice** loves **science** and **pizza**” -> (alice, science), (alice, pizza), (science, pizza)

2. **Classify** them

(alice, science) ➡ “Alice loves science and pizza” ➡

(alice, pizza) ➡ “Alice loves science and pizza” ➡

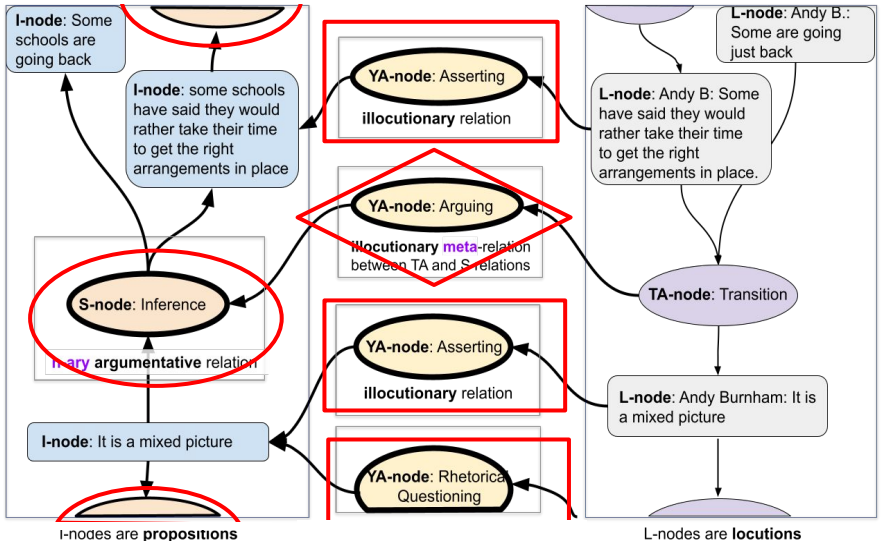
(science, pizza) ➡ “Alice loves science and pizza” ➡

➡ *likes (alice, science)*

➡ *likes (alice, pizza)*

➡ ~~*no relation (science, pizza)*~~

System Description - Selecting Candidate Tuples

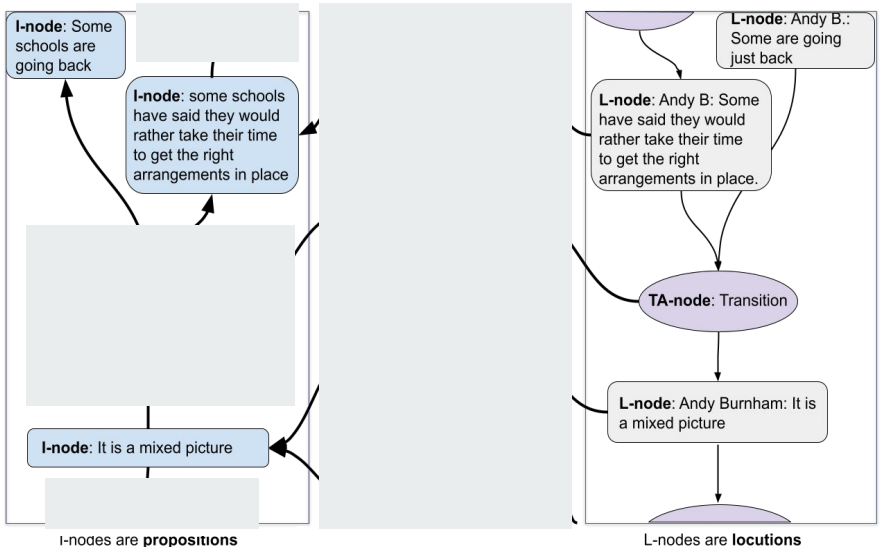


DialAM-2024 Shared Task

Focus on **Relation Extraction (RE)**

1. **Argumentative** Relations (⬭)
2. **Illocutionary** Relations
   a. Anchoring **propositions** (▭)
   b. Anchoring **argumentative Relations** (◇)

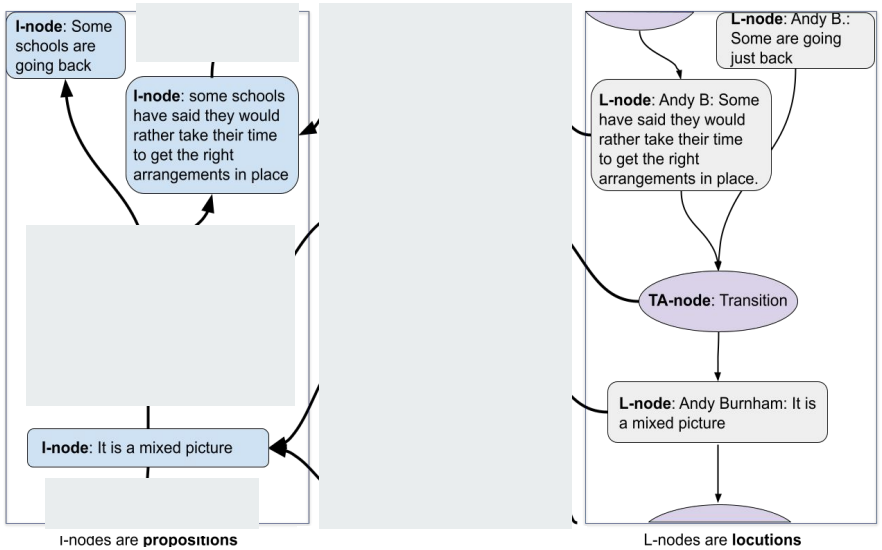
System Description - Selecting Candidate Tuples



I-nodes are **propositions**

L-nodes are **locutions**

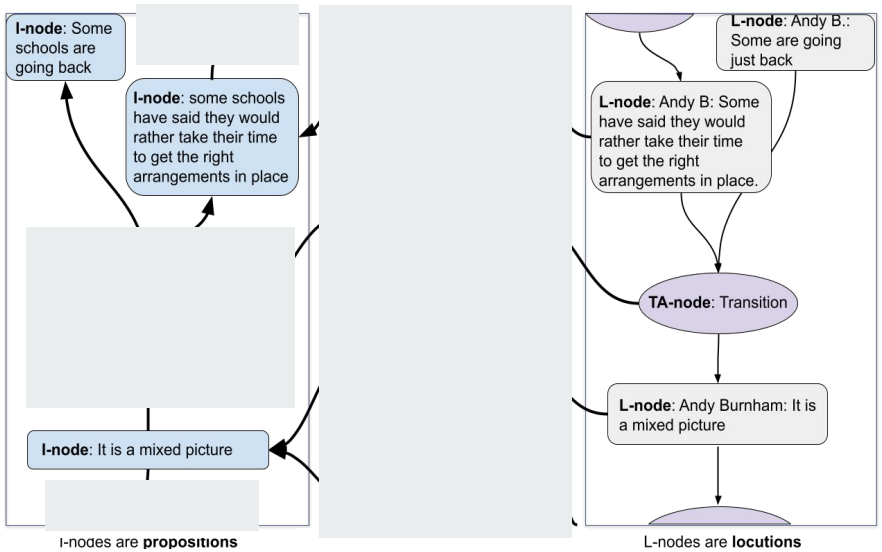
System Description - Selecting Candidate Tuples

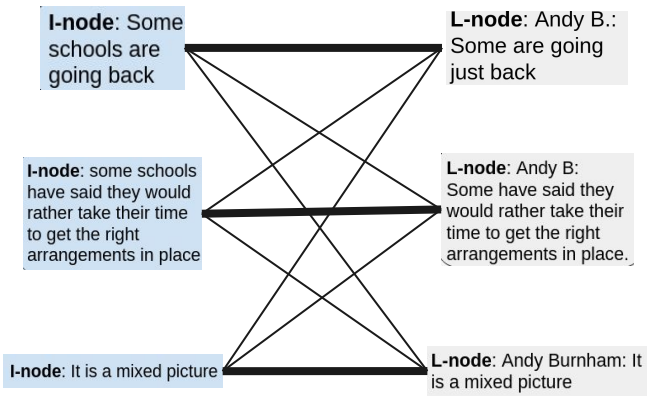


	I-node: Some schools are going back	**I-node**: some schools have said they would rather take their time to get the right arrangements in place	**I-node**: It is a mixed picture
L-node: Andy B.: Some are going just back	.8	.2	.3
L-node: Andy B: Some have said they would rather take their time to get the right arrangements in place.	.2	.9	.1
L-node: Andy Burnham: It is a mixed picture	.3	.1	.85

Similarity Measure: Longest Substring

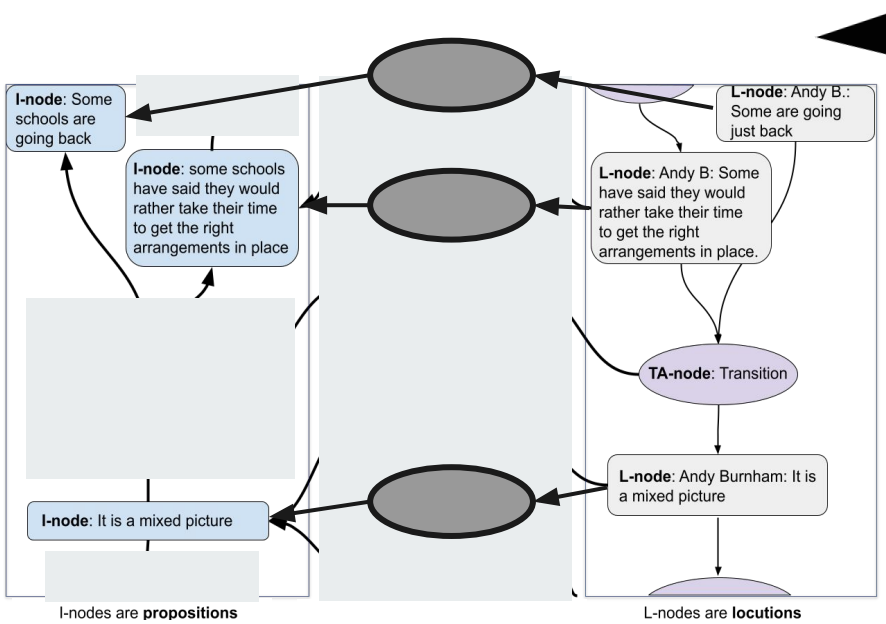
System Description - Selecting Candidate Tuples

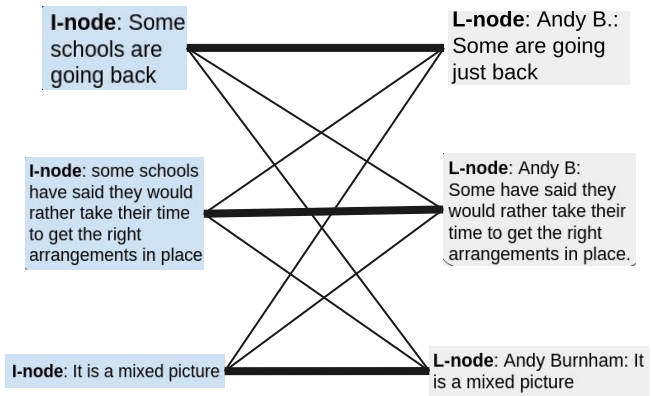




Bi-Partite Matching Algorithm

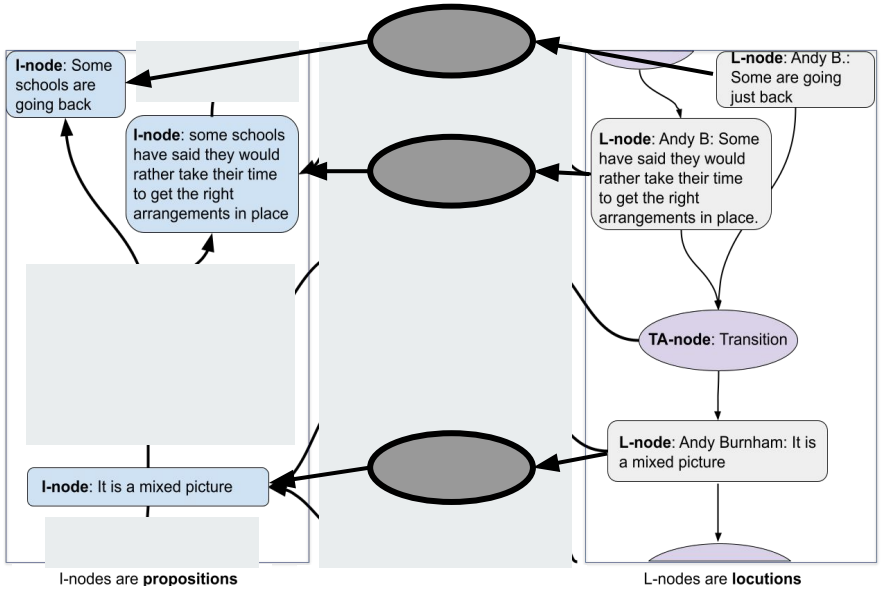
Selecting Candidate Tuples





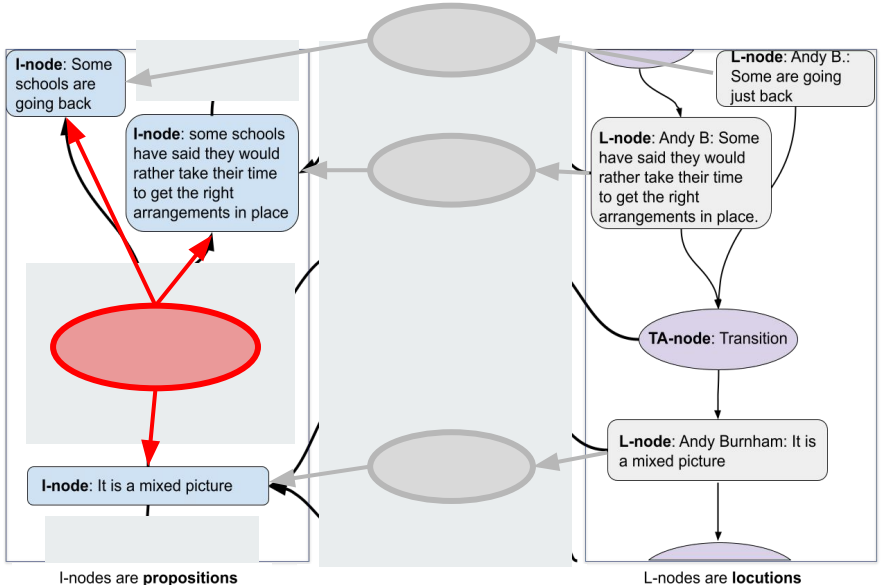
Bi-Partite Matching Algorithm

Selecting Candidate Tuples



- ☐ **Argumentative** Relations
- ☐ **Illocutionary** Relations
  - ☑ Anchoring **propositions**
  - ☐ Anchoring **argumentative Relations**

Selecting Candidate Tuples

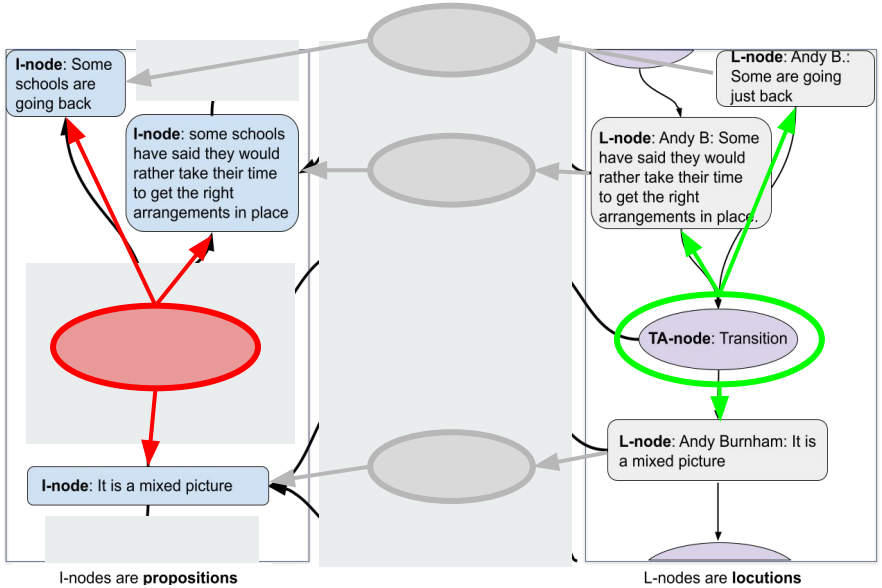


? **Argumentative** Relations

☐ **Illocutionary** Relations

- ☑ Anchoring **propositions**
- ☐ Anchoring **argumentative Relations**

Selecting Candidate Tuples

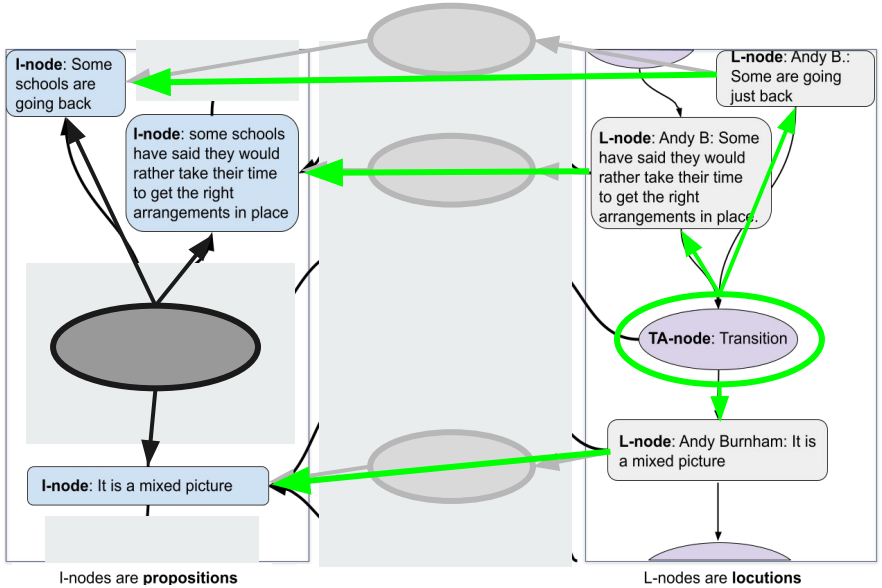


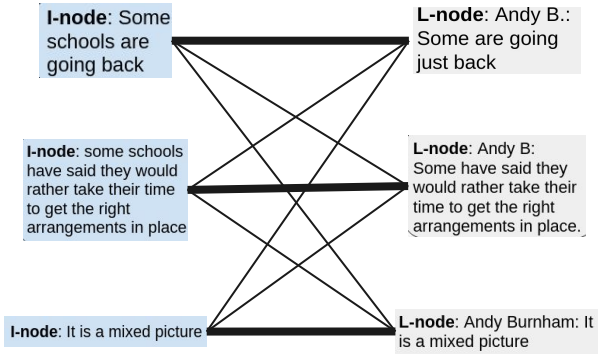
? **Argumentative** Relations

☐ **Illocutionary** Relations

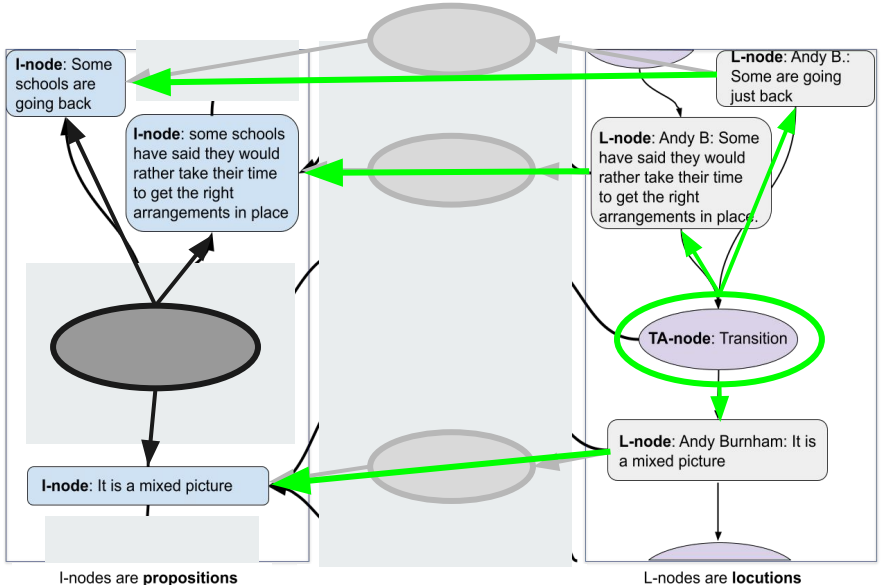
- ☑ Anchoring **propositions**
- ☐ Anchoring **argumentative Relations**

Selecting Candidate Tuples



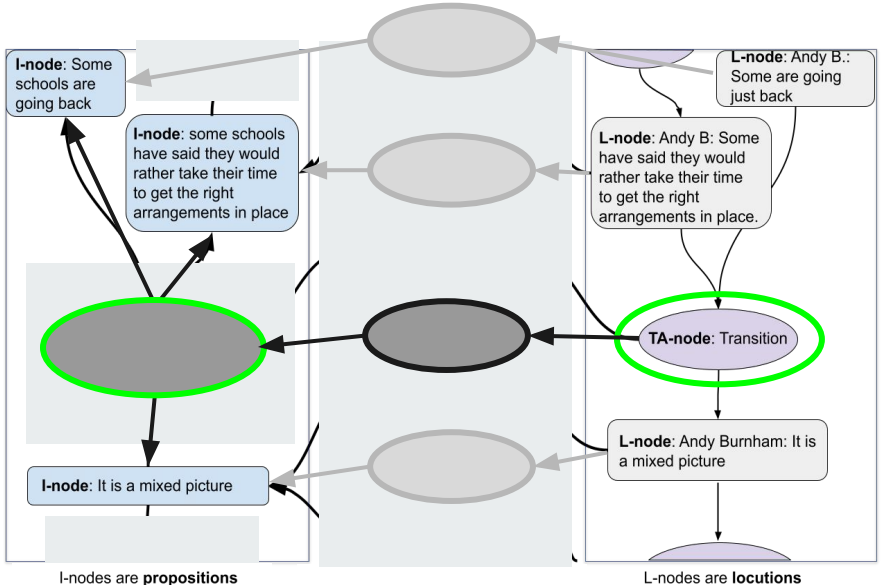


Selecting Candidate Tuples



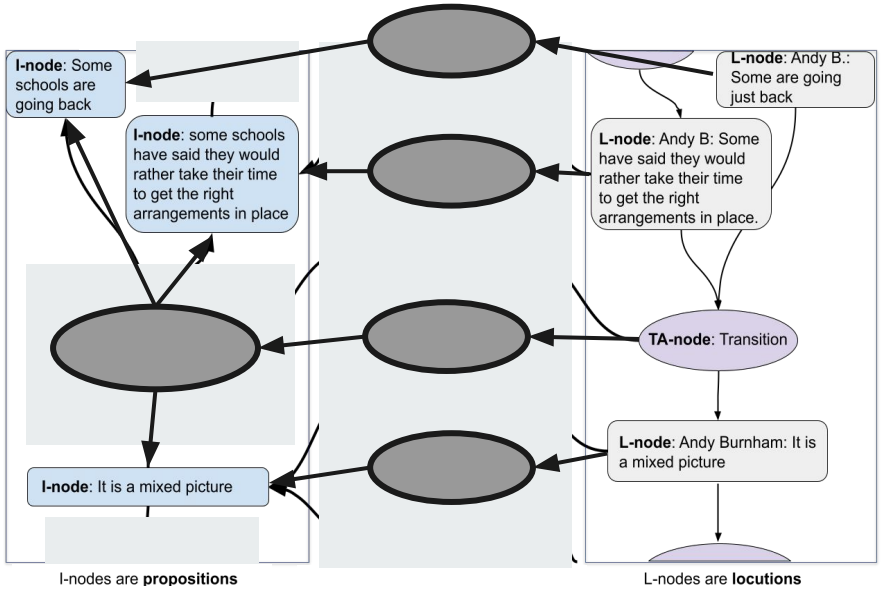
- ☑ **Argumentative** Relations
- ☐ **Illocutionary** Relations
  - ☑ Anchoring **propositions**
  - ☐ Anchoring **argumentative Relations**

Selecting Candidate Tuples



- ☑ **Argumentative** Relations
- ☑ **Illocutionary** Relations
  - ☑ Anchoring **propositions**
  - ☑ Anchoring **argumentative Relations**

Selecting Candidate Tuples



- ☑ **Argumentative** Relations
- ☑ **Illocutionary** Relations
  - ☑ Anchoring **propositions**
  - ☑ Anchoring **argumentative Relations**

System Description - Overview

How does “classic” Relation Extraction work?

1. **Select** candidate pairs of entities

   “**Alice** loves **science** and **pizza**” -> (alice, science), (alice, pizza), (science, pizza)

2. **Classify** candidate pairs

   (alice, science) ➡ “Alice loves science and pizza” ➡

   (alice, pizza) ➡ “Alice loves science and pizza” ➡

   (science, pizza) ➡ “Alice loves science and pizza” ➡

➡ *likes (alice, science)*

➡ *likes (alice, pizza)*

➡ ~~*no relation (science, pizza)*~~

System Description - Overview

How does “classic” Relation Extraction work?

1. **Select** candidate pairs of entities

   “**Alice** loves **science** and **pizza**” -> (alice, science), (alice, pizza), (science, pizza)

2. **Classify** candidate pairs

(alice, science) ➡ “Alice loves science and pizza” ➡

(alice, pizza) ➡ “Alice loves science and pizza” ➡

(science, pizza) ➡ “Alice loves science and pizza” ➡

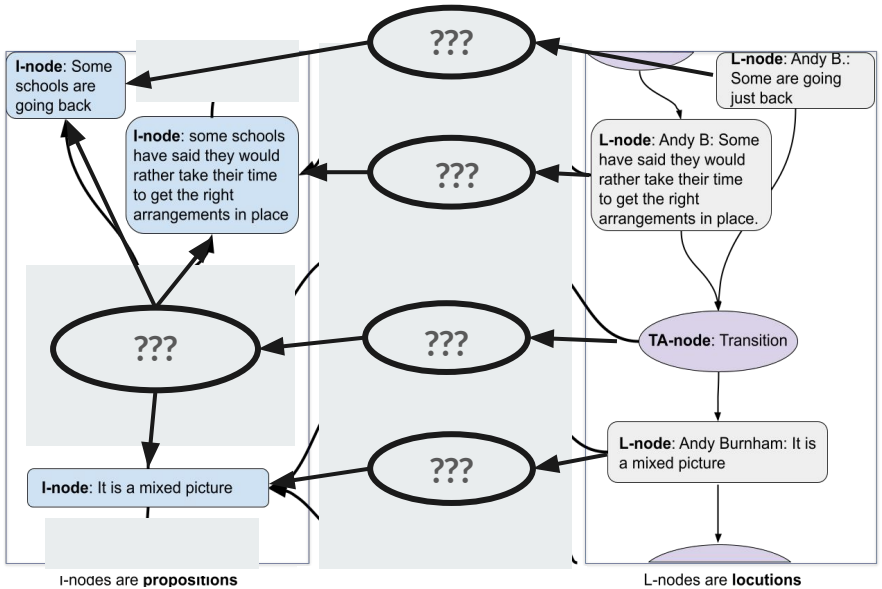
➡ *likes (alice, science)*

➡ *likes (alice, pizza)*

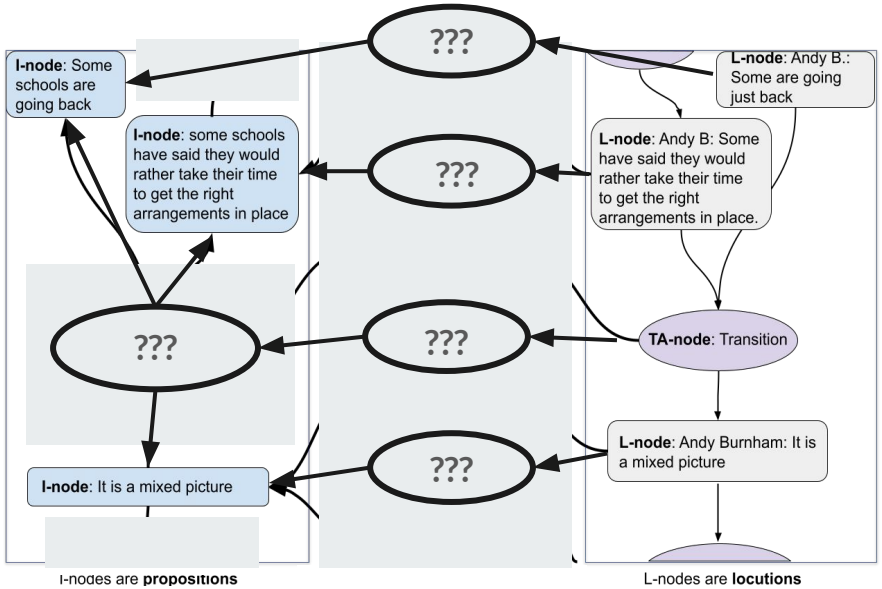
➡ ~~*no relation (science, pizza)*~~

How to encode?

System Description - Classifying Candidate Tuples

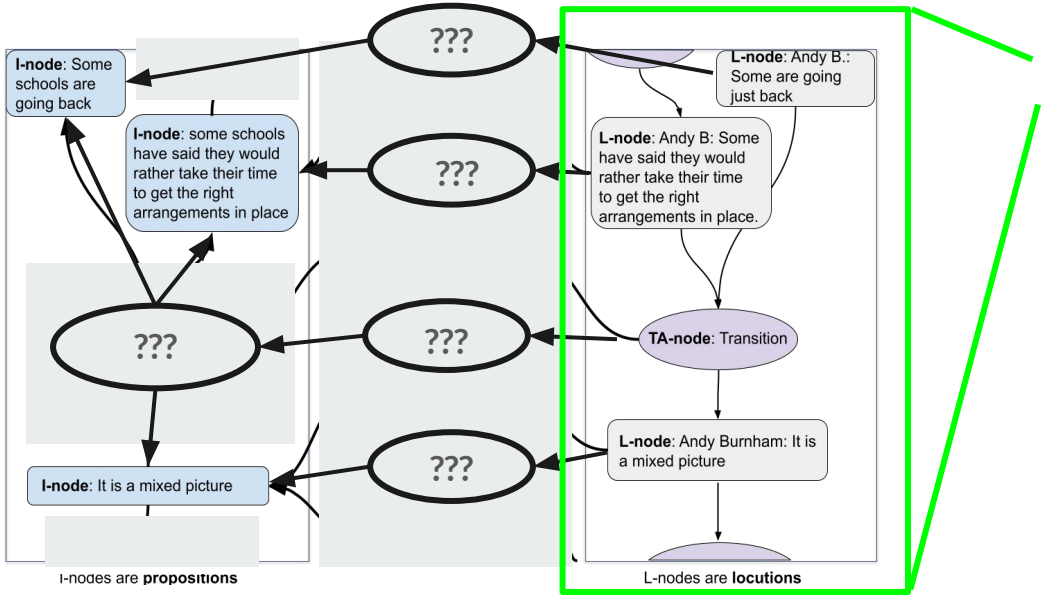


System Description - Classifying Candidate Tuples



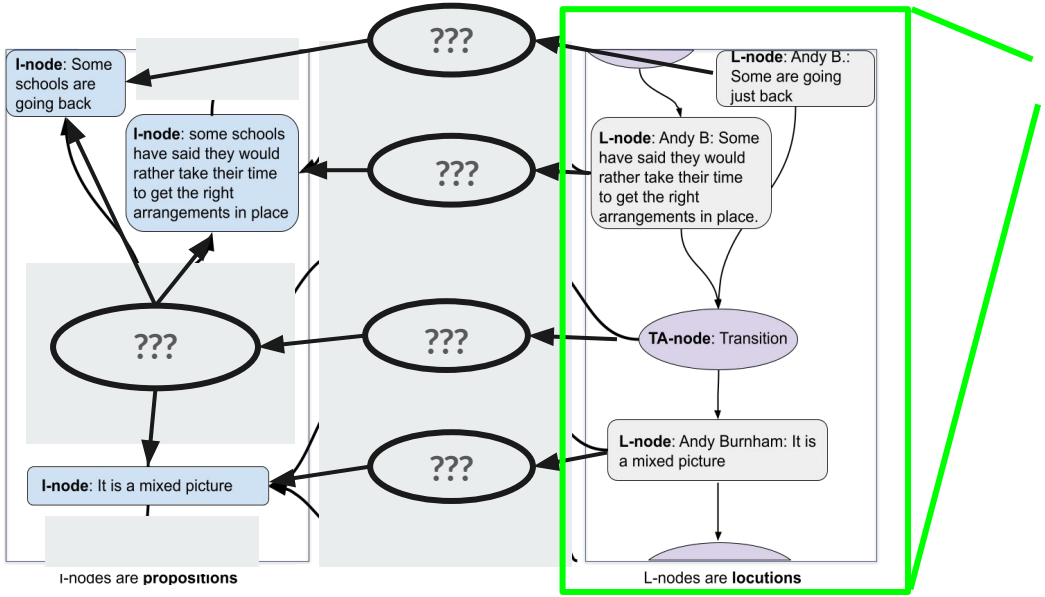
wants text (or a sequence of tokens)

System Description - Encoding Candidate Tuples



close to real world (text) data

System Description - Encoding Candidate Tuples

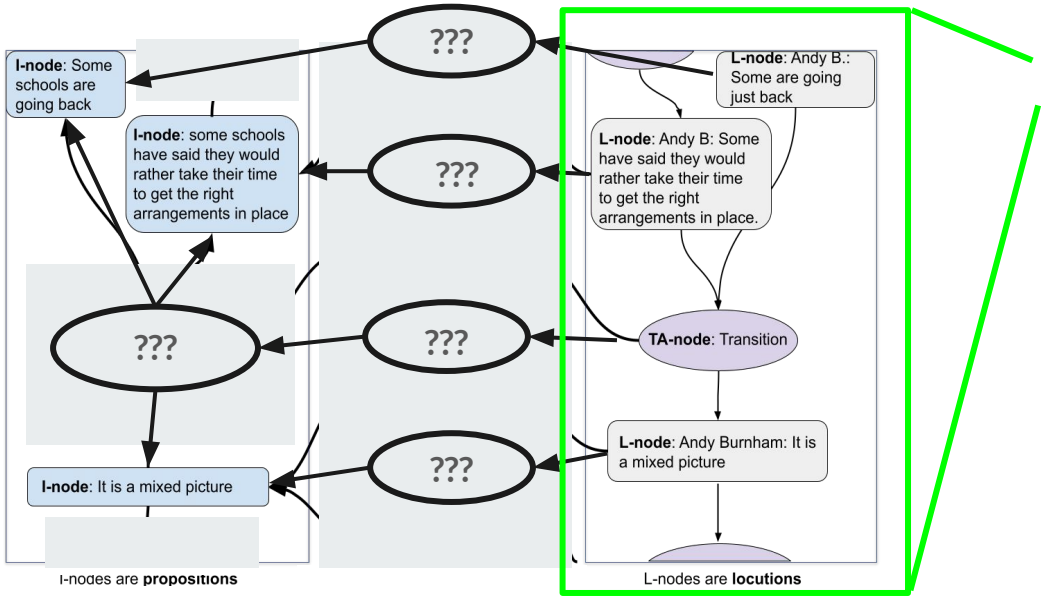


close to real world (text) data

→ **serialize L-nodes**

"Andy B.: Some are going just back</S> Andy B.: Some have said they would rather take their time to get the right arrangements in place.</S>Andy B.: It is a mixed picture."

System Description - Encoding Candidate Tuples



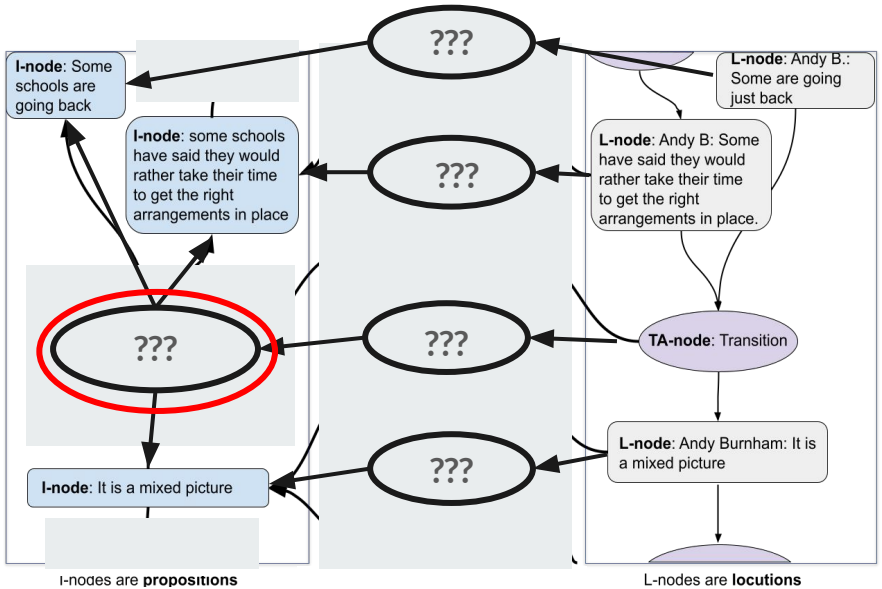
close to real world (text) data

→ **serialize L-nodes**

"Andy B.: Some are going just back</S> Andy B.: Some have said they would rather take their time to get the right arrangements in place.</S>Andy B.: It is a mixed picture."

→ **integrate candidate tuples (?)**

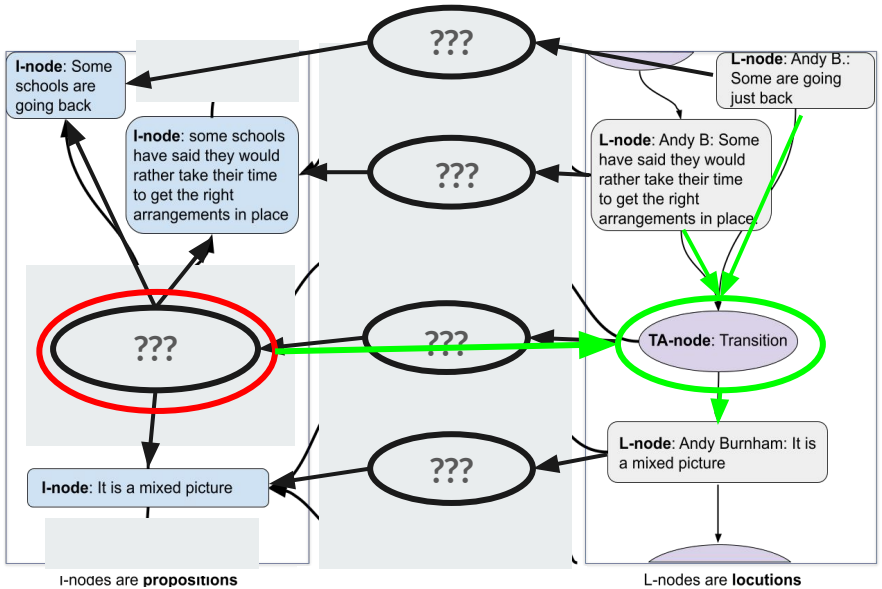
System Description - Encoding Candidate Tuples



integrate candidate tuples:

1. **Argumentative** Relations

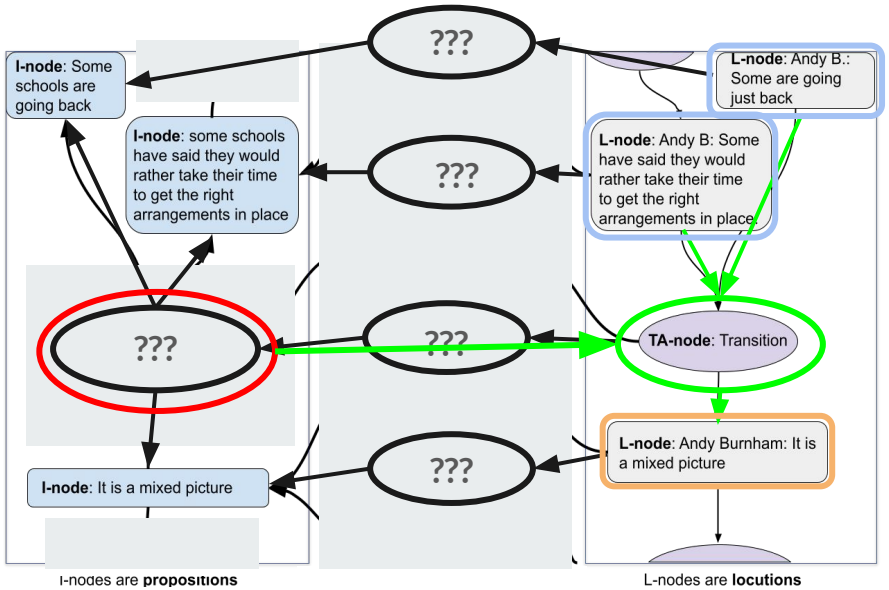
System Description - Encoding Candidate Tuples



integrate candidate tuples:

1. **Argumentative** Relations

System Description - Encoding Candidate Tuples

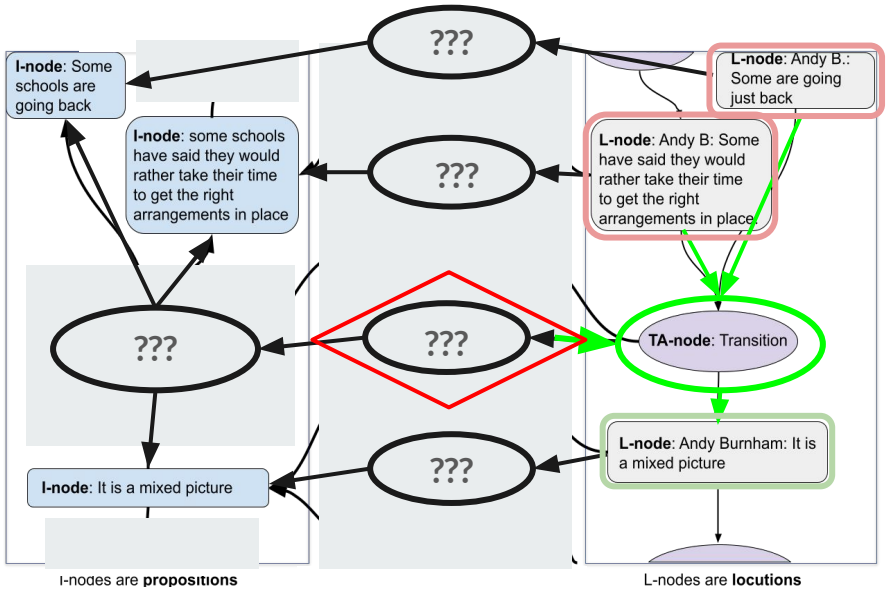


integrate candidate tuples:

1. **Argumentative** Relations

"<arg:in>Andy B.: Some are going just back </arg:in></S><arg:in>Andy B.: Some have said they would rather take their time to get the right arrangements in place.</arg:in></S><arg:out>Andy B.: It is a mixed picture.</arg:out>"

System Description - Encoding Candidate Tuples

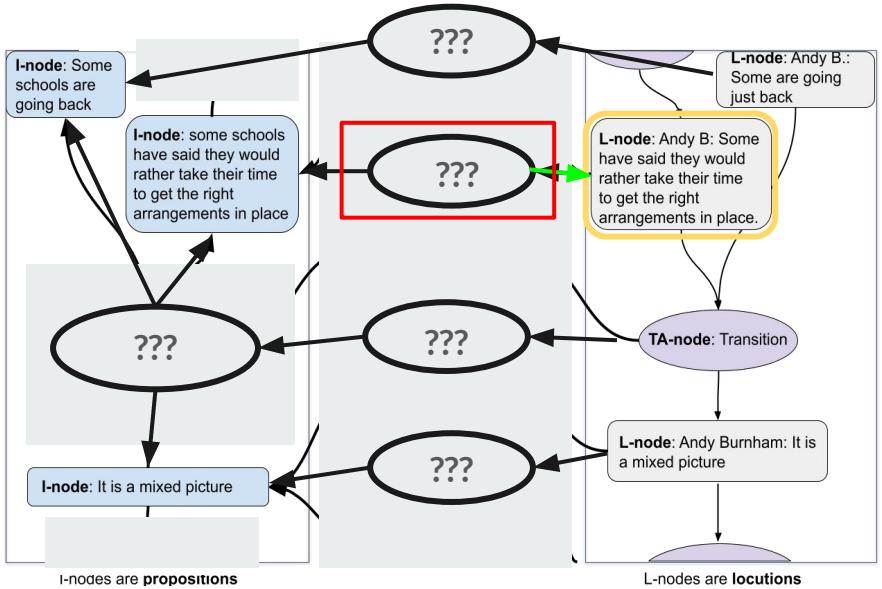


integrate candidate tuples:

2. **Illocutionary** Relations:
anchoring **argumentative relations**

"<illo-arg:in>Andy B.: Some are going just back </illo-arg:in></S><illo-arg:in>Andy B.: Some have said they would rather take their time to get the right arrangements in place.</illo-arg:in></S><illo-arg:out>Andy B.: It is a mixed picture.</illo-arg:out>"

System Description - Encoding Candidate Tuples



integrate candidate tuples:

2. **Illocutionary** Relations:
anchoring **propositions**

"Andy B.: Some are going just back

</S><illo-prop>Andy B.: Some have said they would rather take their time to get the right arrangements in place.</illo-prop>

</S>Andy B.: It is a mixed picture."

Evaluation

Data split sizes: train: 1259, validation: 140, test: 11.

Model	ARI		ILO		GLOBAL	
	Focused	*General*	*Focused*	*General*	*Focused*	*General*
baseline	22.80	26.46	**72.09**	45.75	47.45	36.10
best-competitor	**35.89**	46.22	69.95	**81.17**	45.23	63.70
dfki-mlst (ours)	30.40	**55.33**	66.10	78.78	**48.25**	**67.05**

Table 1: Argumentative (ARI) and illocutionary (ILO) relation detection performance of the *official baseline*, *best competitor model*, and *dfki-mlst (ours)* per task and in means of macro F1.

Evaluation - Impact of Candidate Tuple Selection

Model	ARI		ILO		GLOBAL	
	Focused	*General*	*Focused*	*General*	*Focused*	*General*
candidate selection only	78.61	93.04	83.05	95.61	80.83	94.33
full pipeline	36.33	60.06	70.35	85.11	53.34	72.59
full pipeline, normalized	46.22	64.55	84.71	89.02	65.99	76.95

Table 2: **Impact of candidate selection on the performance, evaluated on the validation data.** *candidate*

1. Candidate selection, 2. Assigning gold labels where possible, 3. Evaluate with gold data

Evaluation - Impact of Candidate Tuple Selection

Model	ARI		ILO		GLOBAL	
	Focused	*General*	*Focused*	*General*	*Focused*	*General*
candidate selection only	78.61	93.04	83.05	95.61	80.83	94.33
full pipeline	36.33	60.06	70.35	85.11	53.34	72.59
full pipeline, normalized	46.22	64.55	84.71	89.02	65.99	76.95

Table 2: **Impact of candidate selection on the performance, evaluated on the validation data.** *candidate*

“full pipeline” / “candidate selection only” → assuming perfect candidate selection

Bonus part

Challenges

1. **Imbalanced** Label Distribution
2. **Meta-relation** Classification
3. **Annotation Bugs**: Loops and Disconnected Parts
4. **Reported Speech**
5. **Unanchored** Nodes

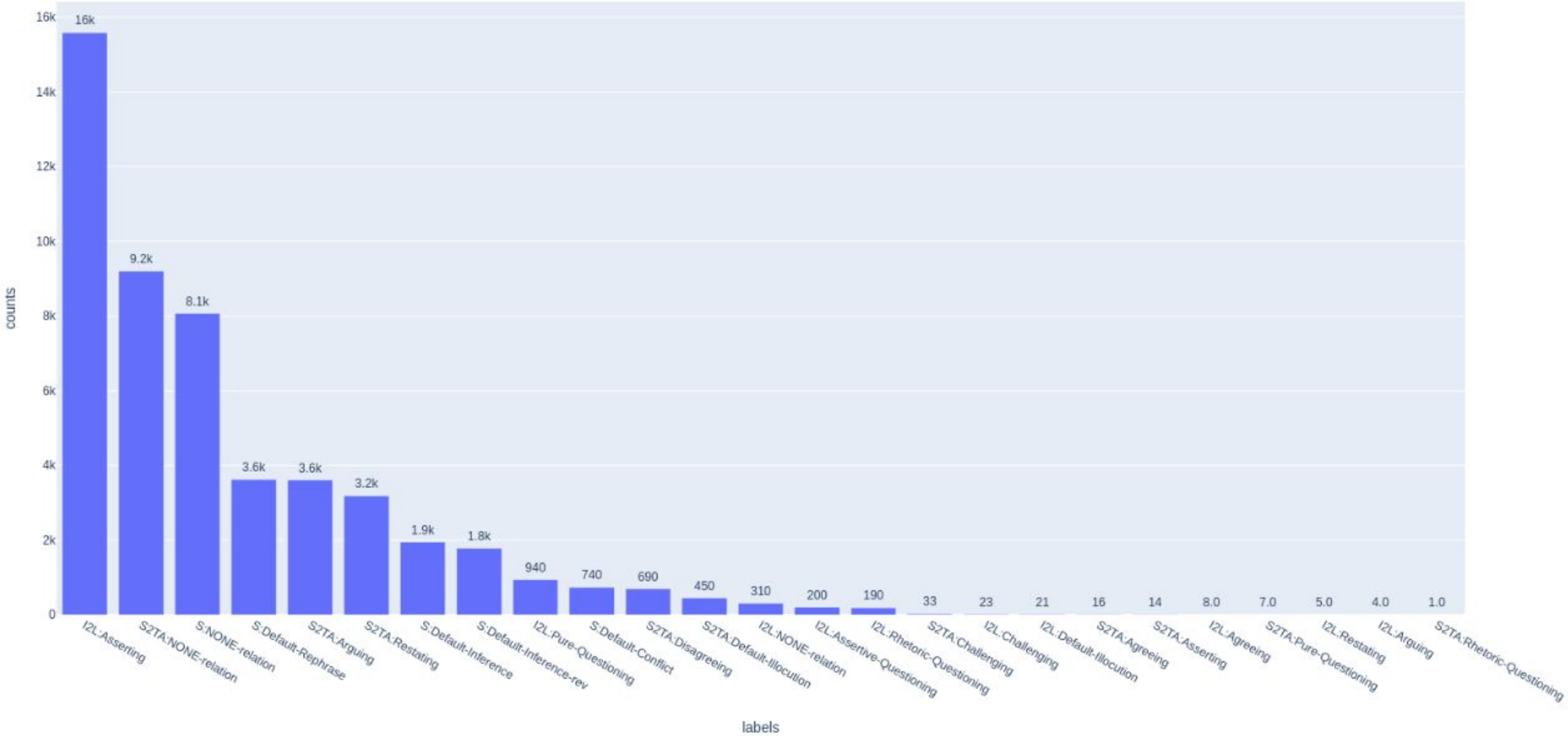
Label Distribution

Our classifier has to distinguish between **25 relation labels** that include argumentative as well as illocutionary and NONE relations.

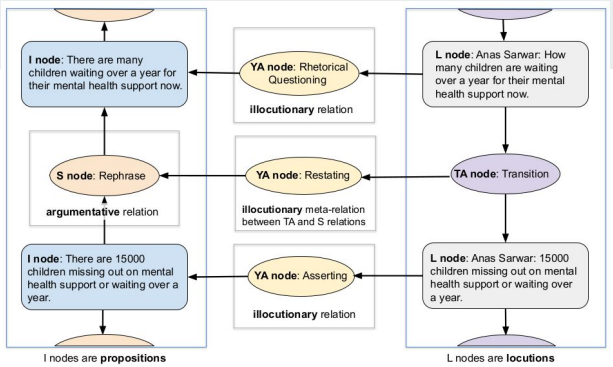
The data distribution is **very imbalanced**! Especially **YA relations** connecting locutions with propositions are very challenging.

E.g. *Asserting* appears in more than 90% of all YA-node annotations, while labels such as *Restating*, *Arguing* and *Agreeing* all together make up less than 1%.

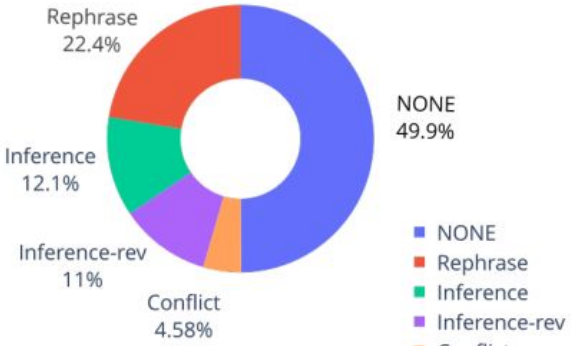
Label Distribution

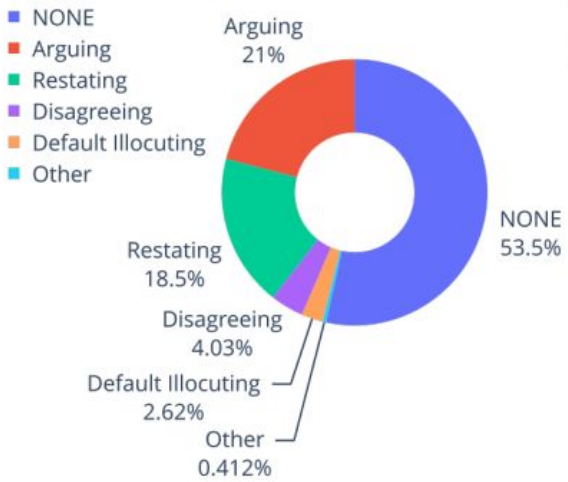


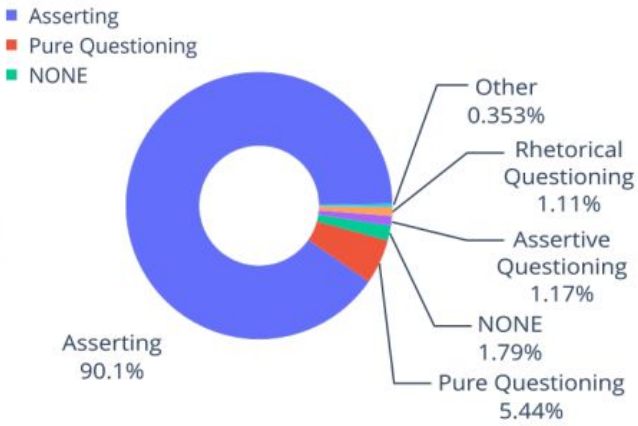
Label Distribution



Label distribution for different types of relations:
S-nodes for argumentative and YA-nodes for illocutionary relations.



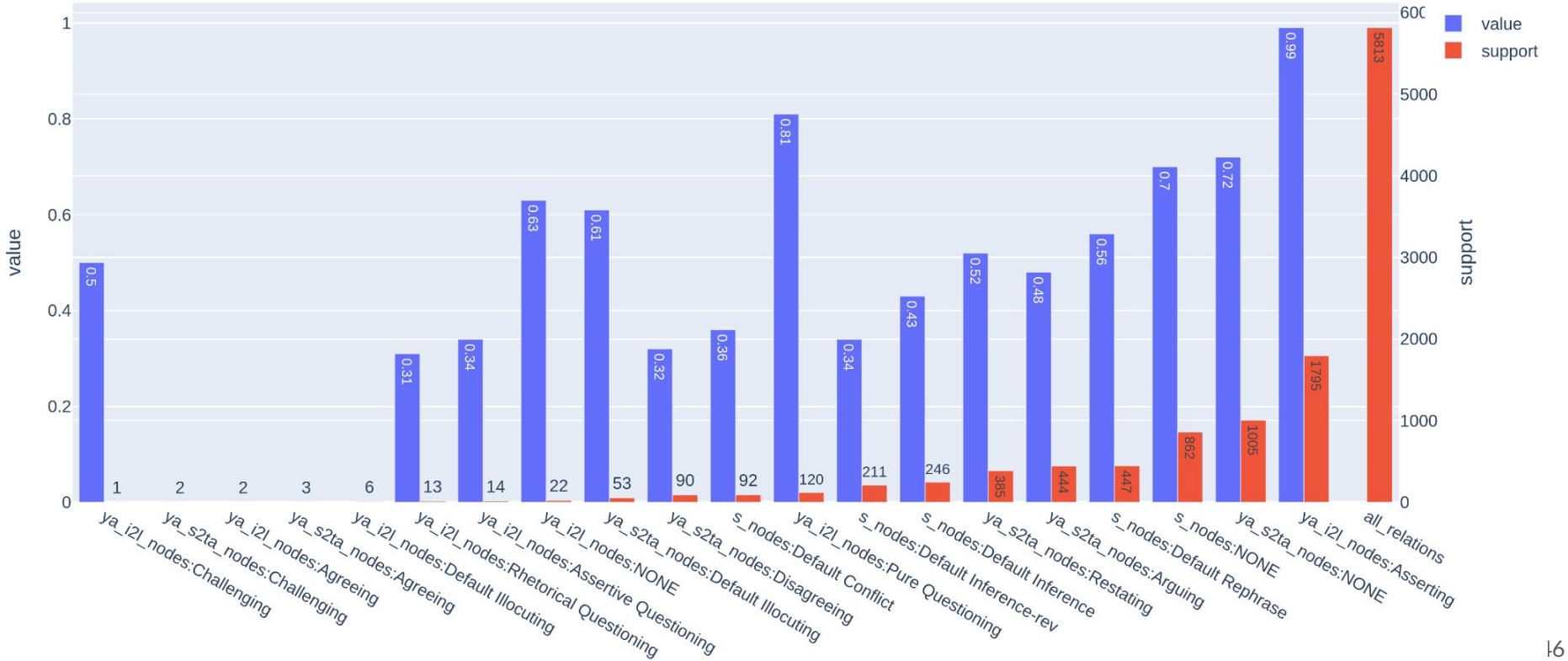




(a) S relations between I nodes. (b) YA relations between TA and S nodes. (c) YA relations between L and I nodes.

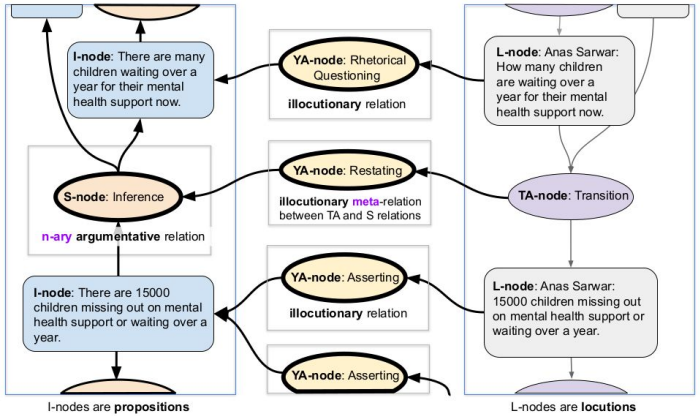
Label Distribution

Performance of dfki-mlst with DeBERTa-v3 on the fixed validation set (140 documents).



Meta-relations

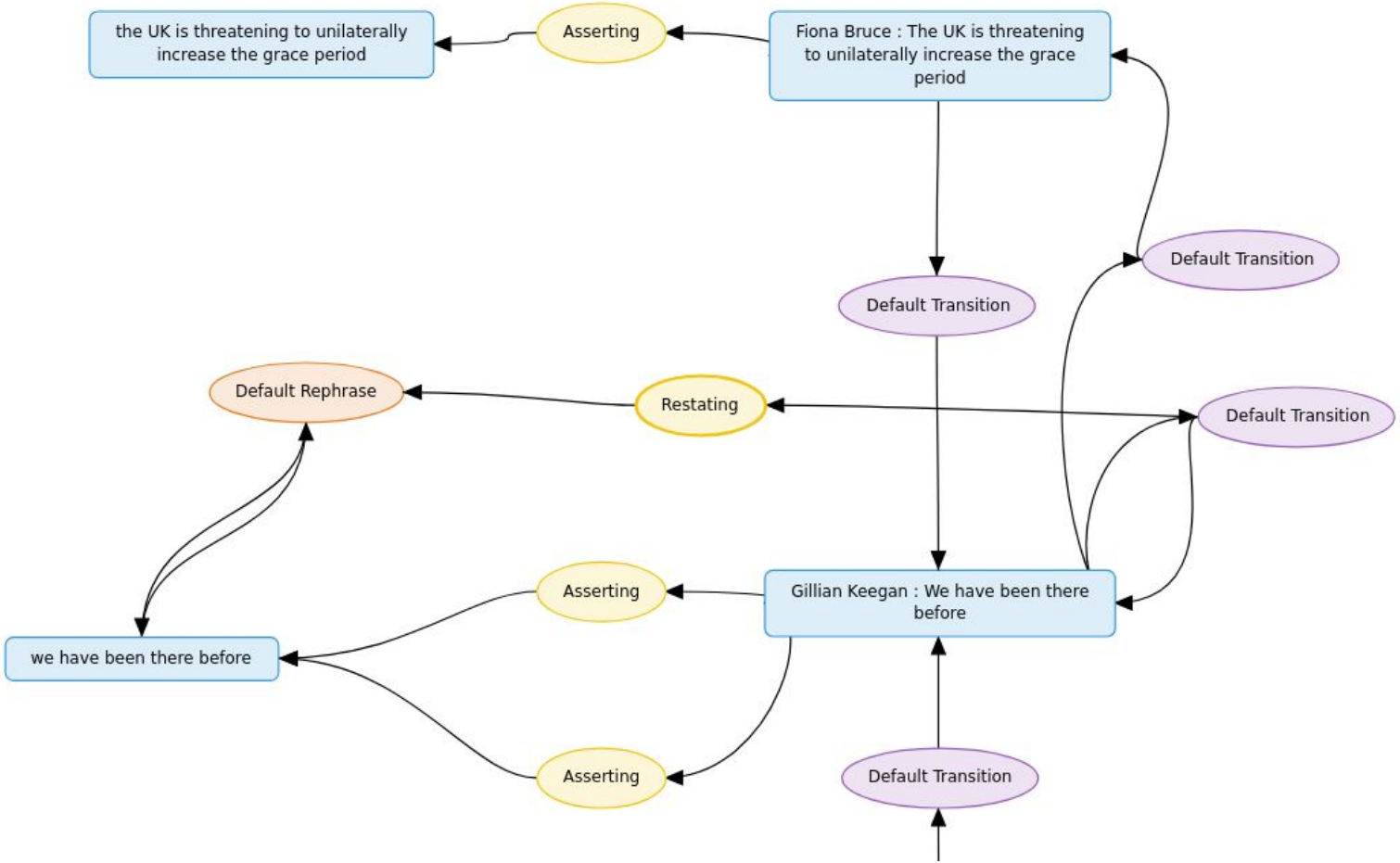
Remember this figure?



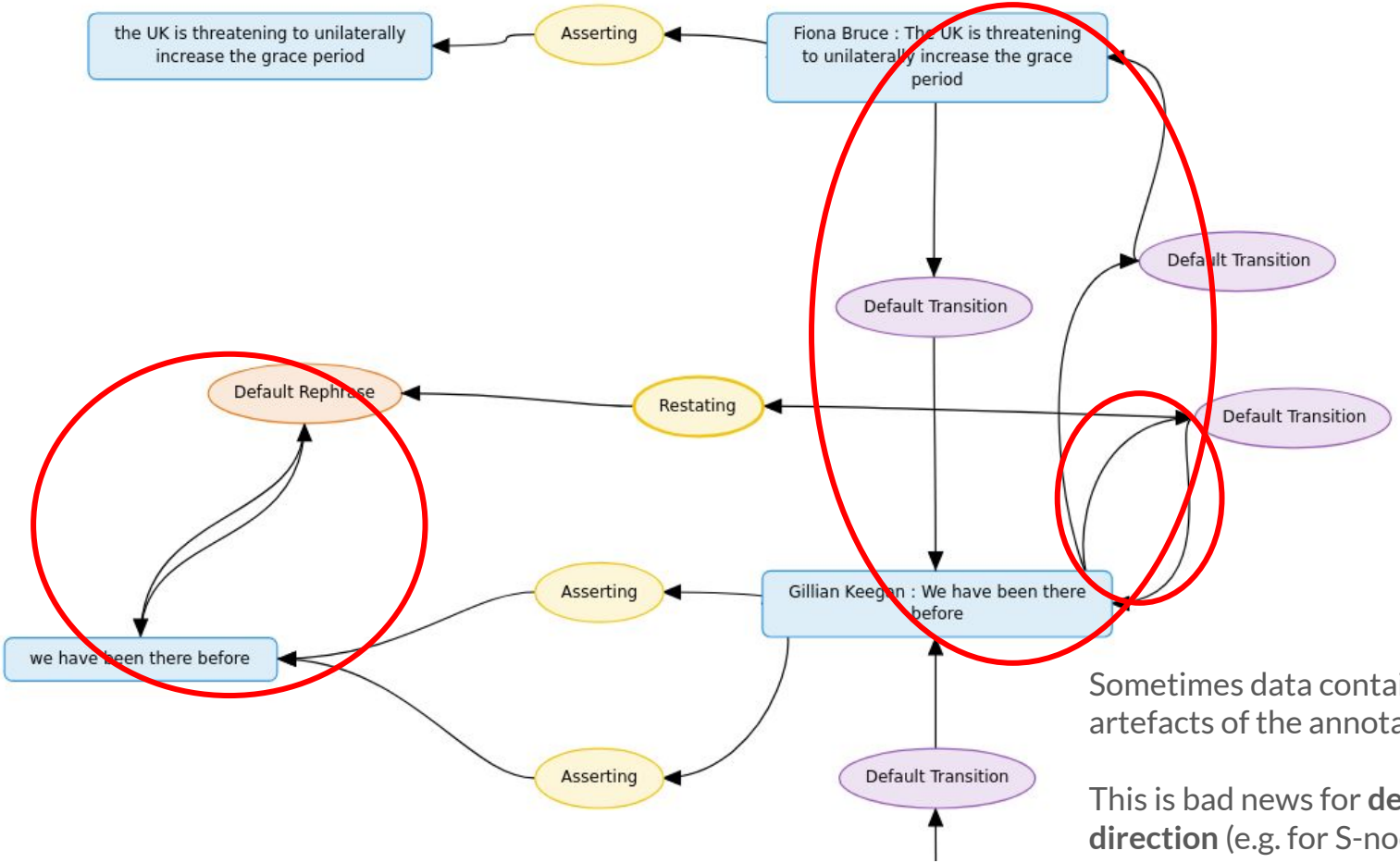
Meta-relations connect two different relation nodes, e.g. YA-nodes between TA and S-nodes.

Why are they challenging? To classify the meta-relations we need to **first identify the correct argumentative relations** (S-nodes). There are 10 types of valid meta-relations: *Agreeing*, *Arguing*, *Asserting*, *Challenging* etc. and they have quite **imbalanced distribution**.

Loops


the UK is threatening to unilaterally increase the grace period
Asserting
Fiona Bruce : The UK is threatening to unilaterally increase the grace period
Default Transition
Default Transition
Default Rephrase
Restating
Default Transition
Asserting
Gillian Keegan : We have been there before
we have been there before
Asserting
Default Transition

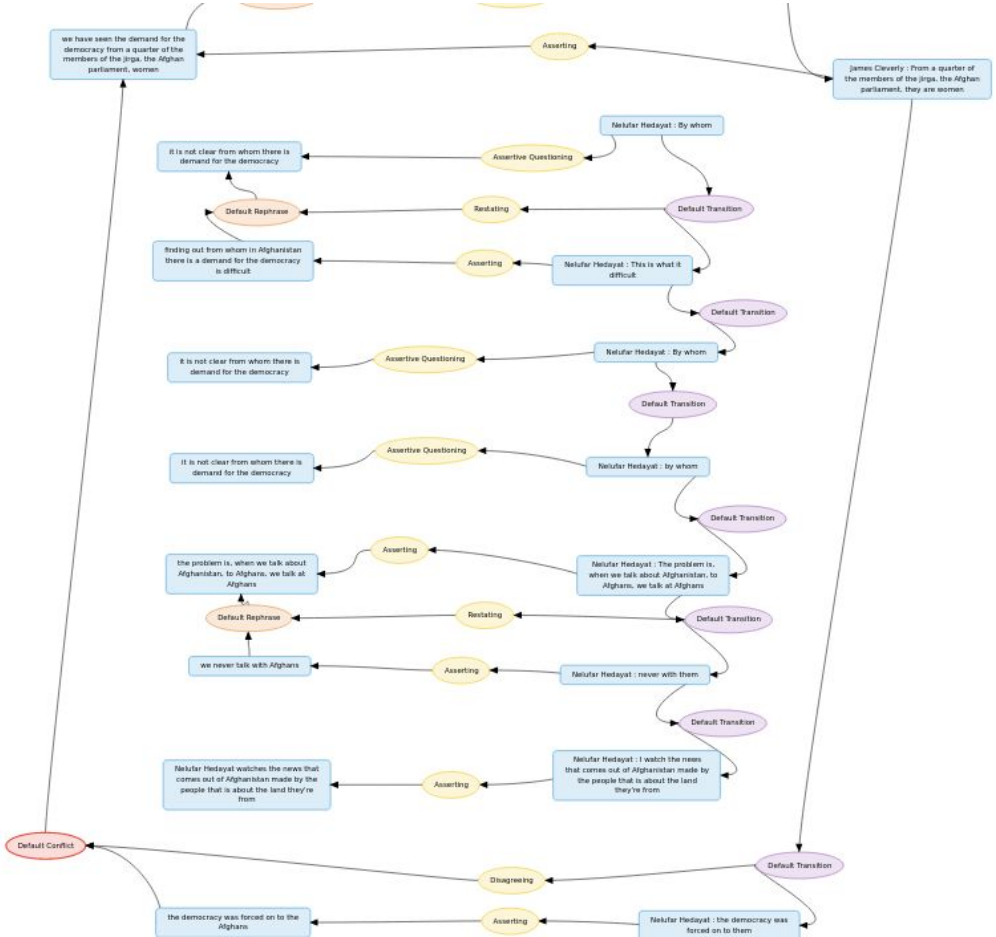
Loops



Sometimes data contains **loops** which are artefacts of the annotation export.

This is bad news for **detecting the relation direction** (e.g. for S-nodes).

Disconnected Parts

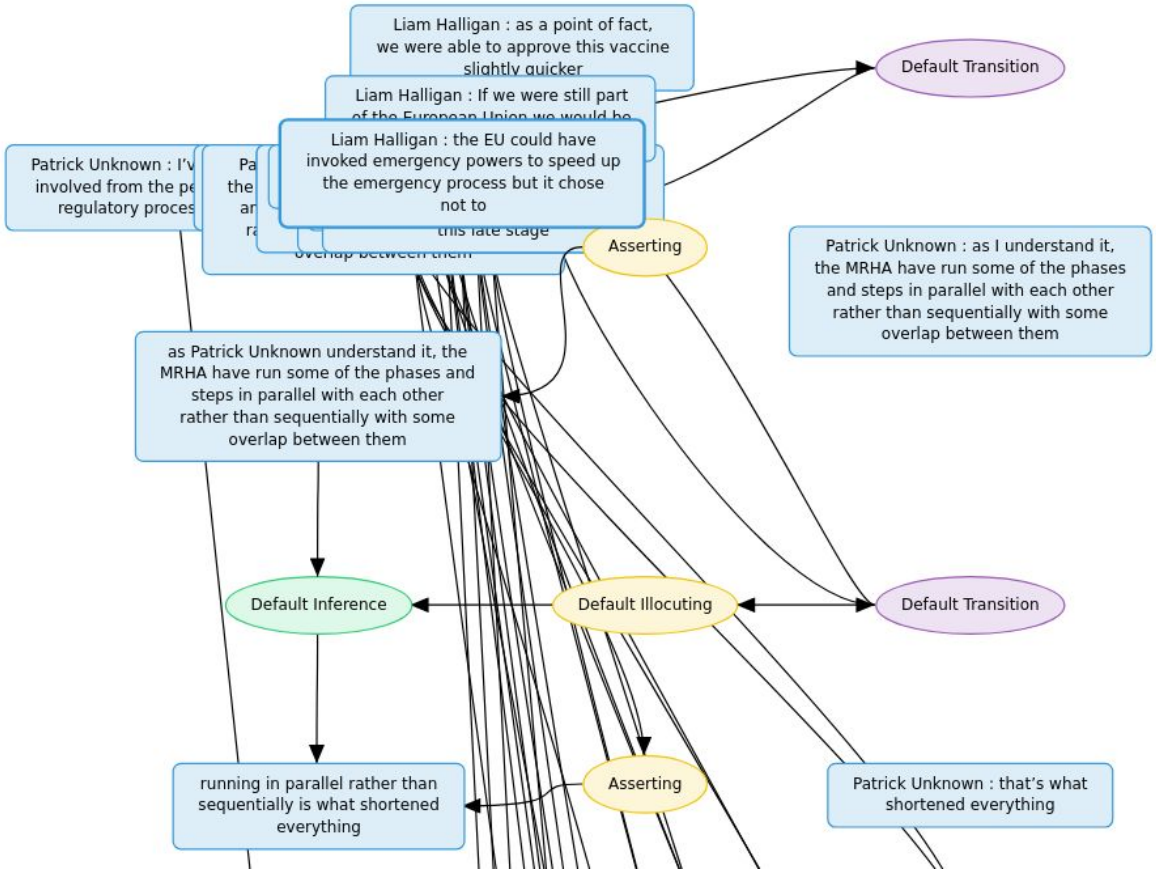


Here we have a single nodeset that consists of **two disconnected graphs**.

This is problematic because we assume that dialogue transitions are continuous and there is **only one L-node w/o any incoming edges**.

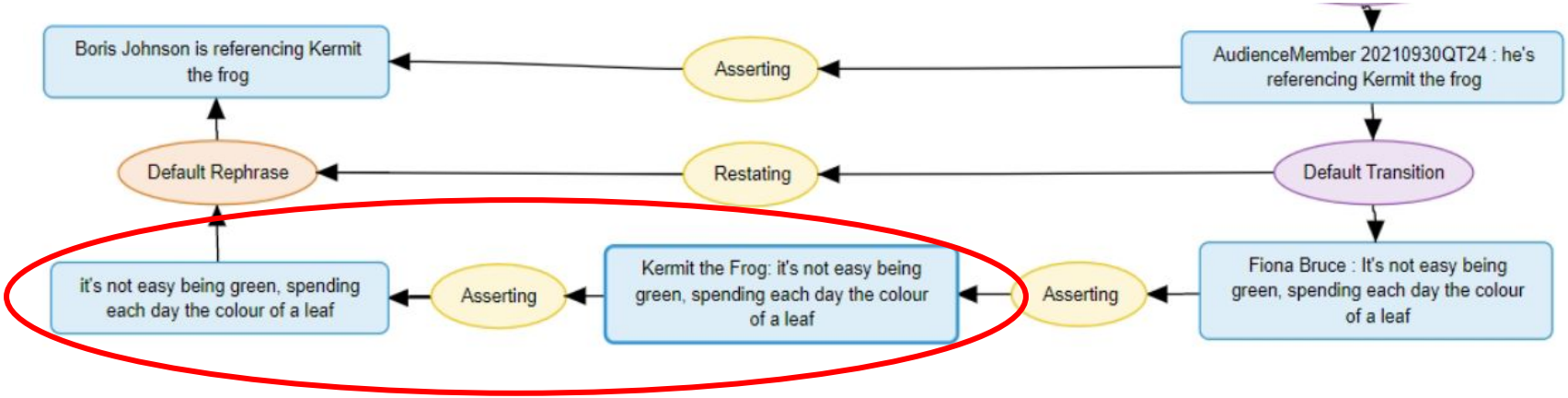
Solution: consider only one of the subgraphs or black-list the whole nodeset. If a single node is disconnected, **remove** it from the graph.

“Broken” nodesets



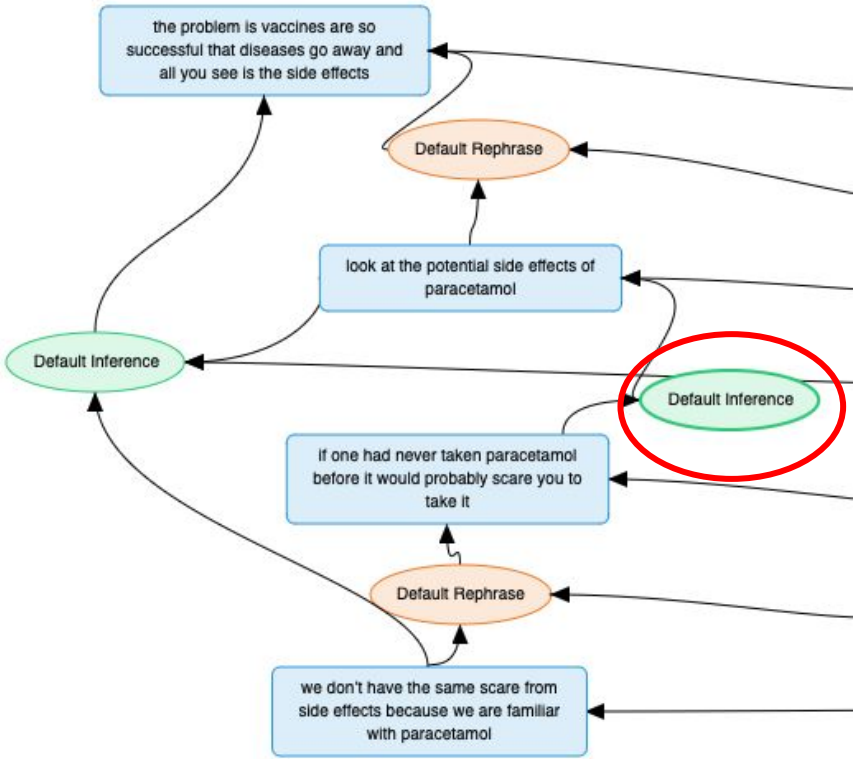
Some nodesets simply look like this 🙄

Reported Speech



Our approach, unfortunately, cannot model reported speech, because we assume that there are no propositions that are not anchored.

Unanchored Nodes



We can have **unanchored YA-nodes** (e.g. *Default Inference* node inside the red box).

This happens because we have a **rephrase between two propositions**, and a linked argument between one of these propositions together with a third.

So, the *Default Inference* YA-node is **not** anchored in this example!

This is problematic because we assume that **all YA-nodes must be anchored** either in the corresponding locutions (L-nodes) or in the transitions between them (TA-nodes).

Additional Experiments

Evaluation Metrics & Results

Model Selection: Classifier Base Models

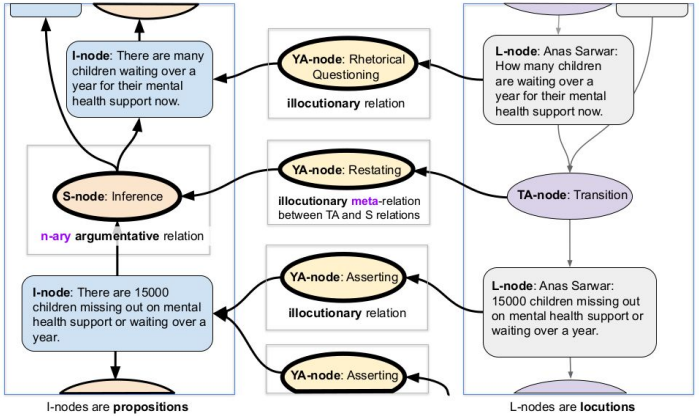
Data Augmentation: EDA & Paraphrases

Balancing Out: Weighted Loss & Re-sampling

Combining Node Texts: L + I-nodes

Training on *all* Nodesets

Evaluation Metrics



Precision, Recall, and F1 score are computed for 3 settings:

	ARI argumentative	**ILO** illocutionary	**GLOBAL**
Relation types	*Inference, Conflict, Rephrase*	*Asserting, Agreeing, Questioning* etc.	all types
Relation nodes	S-nodes connecting I-nodes	YA-nodes connecting: 1) L and I-nodes (locution-to-proposition) 2) TA and S-nodes (meta-relations)	all nodes

Evaluation Metrics

Each setting is further evaluated at 2 levels:

Focused: looking only at the related arguments/locutions (according to the gold standard)

General: evaluating the complete argument maps (including unrelated nodes)

High performance in General but low in Focused represents a **pessimistic approach** that over-relies on the non-related class.

High performance in Focused but low in General represents an **optimistic approach**, relating too many propositions/locutions.

Evaluation @Focused

ARI - Focused	Precision	Recall	F1
MAJORITY-BL	0.00	0.00	0.00
RoBERTa-BL	37.10	18.42	22.80
dfki-mlst	43.87	24.82	30.40
KnowComp	23.47	5.85	9.06
misaka	23.47	5.85	9.06
Pokemon	**46.26**	**32.43**	**35.89**
Pungene	30.18	17.59	20.51
Turiya	18.95	4.21	6.65

ILO - Focused	Precision	Recall	F1
MAJORITY-BL	0.00	0.00	0.00
RoBERTa-BL	73.10	72.55	72.09
dfki-mlst	69.12	66.25	66.10
KnowComp	48.44	41.27	44.33
misaka	48.44	41.27	44.33
Pokemon	54.15	49.87	51.39
Pungene	**71.18**	**69.23**	**69.95**
Turiya	43.81	26.09	30.41

GLOBAL - Focused	Precision	Recall	F1
MAJORITY-BL	0.00	0.00	0.00
RoBERTa-BL	55.10	45.49	47.45
dfki-mlst	**56.50**	**45.53**	**48.25**
KnowComp	35.95	23.56	26.70
misaka	35.95	23.56	26.70
Pokemon	50.20	41.15	43.64
Pungene	50.68	43.41	45.23
Turiya	31.38	15.15	18.53

Evaluation @General

ARI - General	Precision	Recall	F1
MAJORITY-BL	28.79	30.28	29.52
RoBERTa-BL	28.59	34.69	26.46
dfki-mlst	**61.96**	**53.30**	**55.33**
KnowComp	32.43	33.79	32.75
misaka	32.43	33.79	32.75
Pokemon	32.00	46.56	30.64
Pungene	49.21	46.32	46.22
Turiya	30.81	31.52	30.75

ILO - General	Precision	Recall	F1
MAJORITY-BL	34.71	35.90	35.29
RoBERTa-BL	39.11	62.07	45.75
dfki-mlst	81.08	79.25	78.78
KnowComp	82.35	76.26	78.90
misaka	82.35	76.26	78.90
Pokemon	56.41	64.57	59.36
Pungene	**81.99**	**80.79**	**81.17**
Turiya	51.37	57.05	53.31

GLOBAL - General	Precision	Recall	F1
MAJORITY-BL	31.75	33.09	32.40
RoBERTa-BL	33.85	48.38	36.10
dfki-mlst	**71.52**	**66.28**	**67.05**
KnowComp	57.39	55.03	55.82
misaka	57.39	55.03	55.82
Pokemon	44.20	55.57	45.00
Pungene	65.60	63.55	63.70
Turiya	41.09	44.29	42.03

Evaluation Summary

dfki-mlst shows the **best results in the GLOBAL** setting for both General and Focused

we **outperform RoBERTa baseline** by +30.95 F1 points in General and by +0.8 F1 points in Focused

dfki-mlst also achieves the **best results for ARI General** (argumentative relations)

our approach **works better for ARI** than ILO task

Model	ARI		ILO		GLOBAL	
	Focused	*General*	*Focused*	*General*	*Focused*	*General*
baseline	22.80	26.46	**72.09**	45.75	47.45	36.10
best-competitor	**35.89**	46.22	69.95	**81.17**	45.23	63.70
dfki-mlst (ours)	30.40	**55.33**	66.10	78.78	**48.25**	**67.05**

Model Selection: Classifier Base Models

Our approach works with different base models as text encoders and we experiment with several options before selecting **DeBERTa-large-v3** for the final submission.

Originally we evaluate base models on the **custom validation set**. We take ~10% (140 out of 1399) of the shuffled train data as our validation set.

Since the organizers released annotations for the **gold test data** (11 nodesets), we also evaluate the same set of base models on these data (reported on the next slides).

Model Selection

We compare the following models available on HuggingFace (using the *large* version when available):

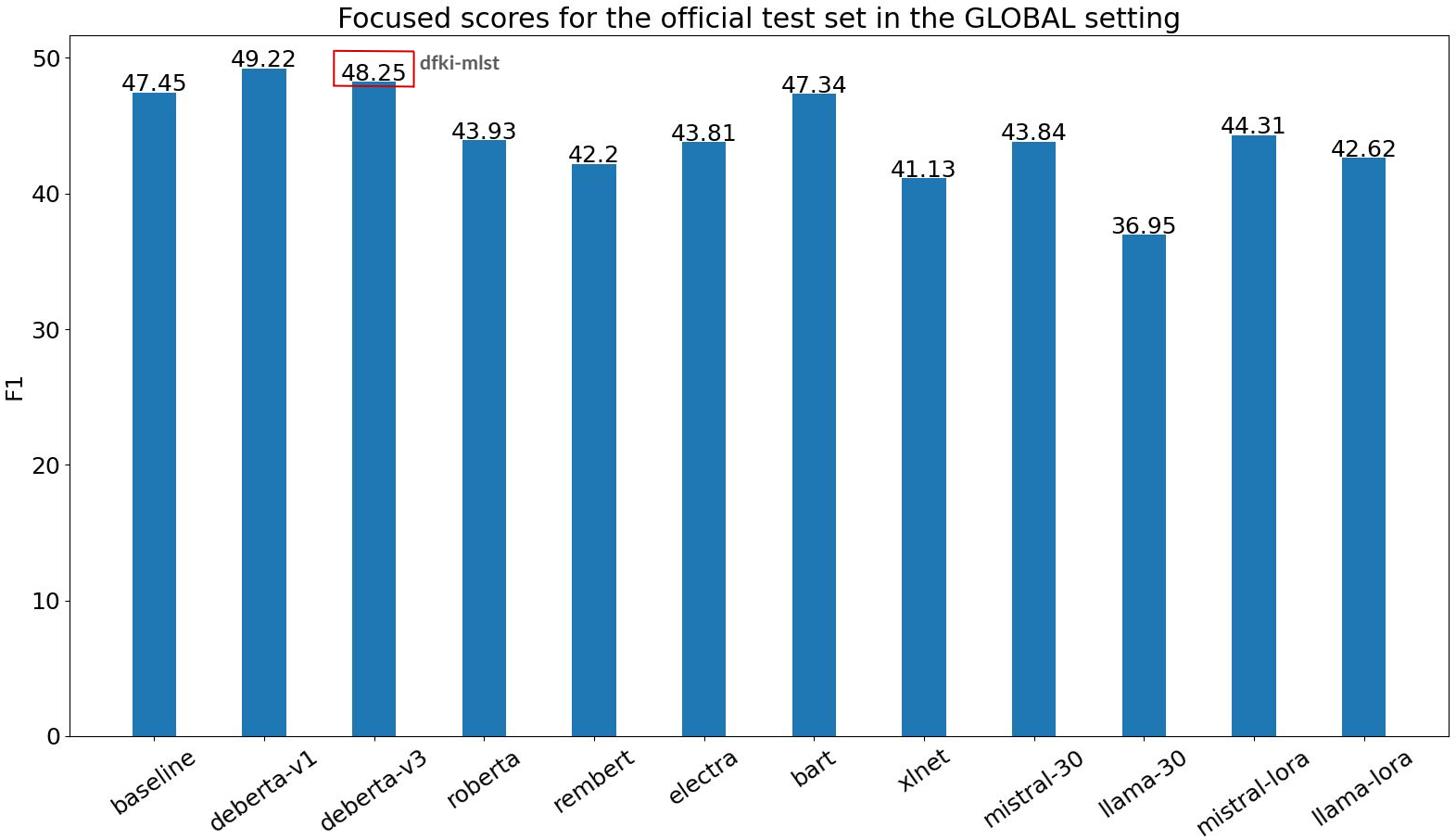
DeBERTa-v1, DeBERTa-v3, RoBERTa, RemBERT, ELECTRA, BART, XLNet

Additionally, we experiment with Llama-2-7B and Mistral-7B-v0.1 but these models are expensive to train, so we either:

(a) Freeze the first 30 layers and train the rest (Llama/Mistral-30)
(b) Use LoRA to train only the adapter weights and keep the base model frozen (Llama/Mistral-LoRA)

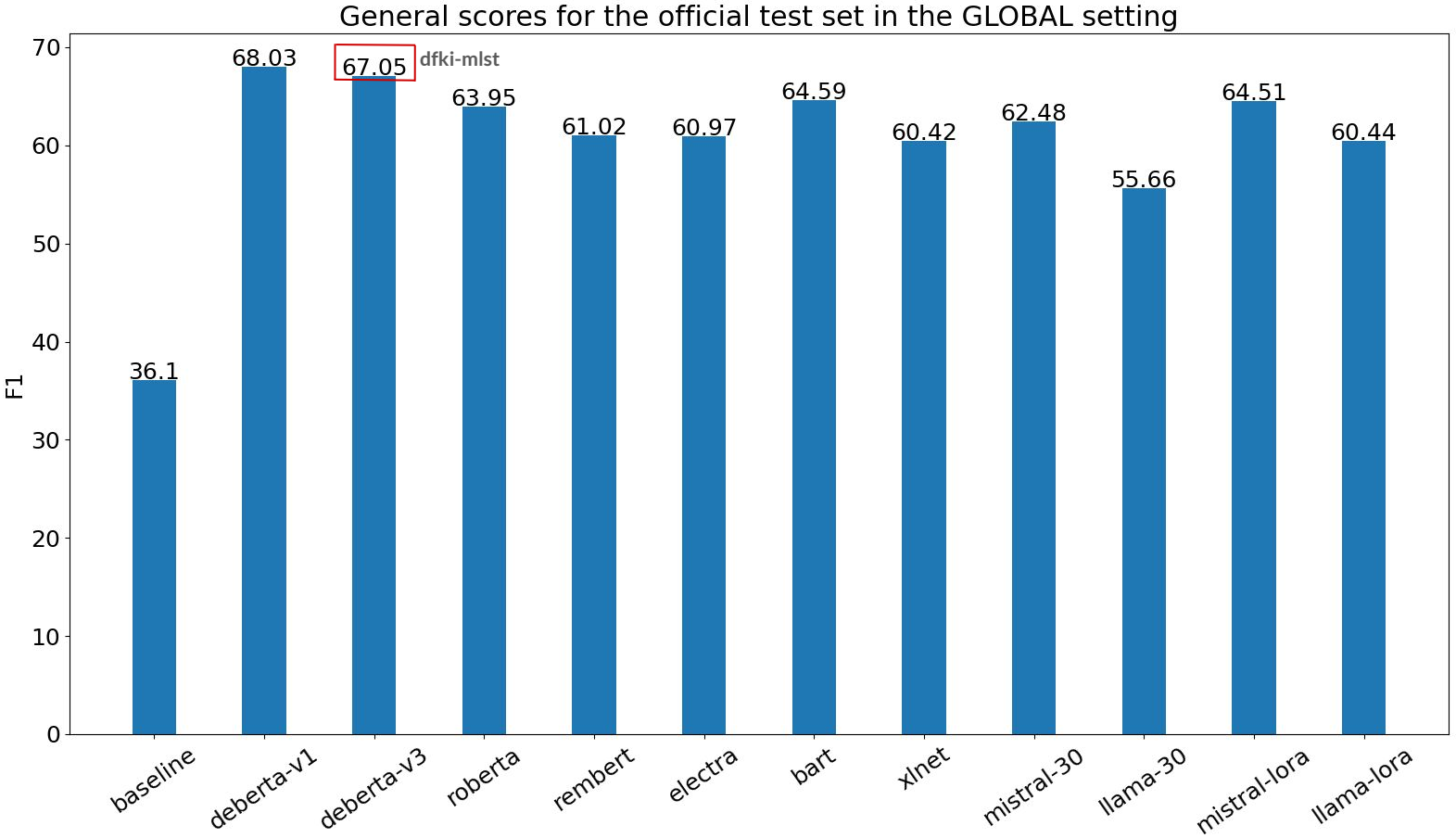
Focused (only gold relations considered), right, there are too many numbers, please skip to the next slide :)

Model	ARI-Focused			ILO-Focused			GLOBAL-Focused		
	Prec	*Rec*	*F1*	*Prec*	*Rec*	*F1*	*Prec*	*Rec*	*F1*
baseline$_{RoBERTa}$	37.10	18.42	22.80	73.10	72.55	**72.09**	55.10	45.49	47.45
best-competitor	46.26	32.43	**35.89**	71.18	69.23	69.95	50.68	43.41	45.23
dfki-mlst$_{DeBERTa\text{-}v3}$	43.87	24.82	30.40	69.12	66.25	66.10	56.50	45.53	**48.25**
DeBERTa-v1	50.98	27.98	**33.82**	66.04	64.32	64.63	58.51	46.15	**49.22**
RoBERTa	48.11	20.45	26.62	63.64	60.66	61.24	55.88	40.55	43.93
RemBERT	41.02	18.35	24.20	62.33	59.49	60.20	51.67	38.92	42.20
ELECTRA	37.46	14.65	20.25	68.76	67.54	67.37	53.11	41.10	43.81
BART	34.09	18.14	22.41	73.50	72.12	**72.28**	53.80	45.13	47.34
XLNet	36.75	19.90	24.04	60.63	58.89	58.22	48.69	39.39	41.13
Mistral-30	33.40	16.50	19.66	67.91	69.08	68.02	50.66	42.79	43.84
Llama-30	21.75	13.25	14.28	60.68	60.10	59.61	41.22	36.67	36.95
Mistral-LoRA	33.62	18.90	23.08	68.36	64.89	65.55	50.99	41.89	44.31
Llama-LoRA	39.07	16.56	22.08	64.68	62.40	63.16	51.88	39.48	42.62


Focused scores for the official test set in the GLOBAL setting
F1
0
10
20
30
40
50
47.45
49.22
48.25
dfki-mlst
43.93
42.2
43.81
47.34
41.13
43.84
36.95
44.31
42.62
baseline
deberta-v1
deberta-v3
roberta
rembert
electra
bart
xlnet
mistral-30
llama-30
mistral-lora
llama-lora

General (all relations considered), the same here, see the next slide for the summary...

Model	ARI-General			ILO-General			GLOBAL-General		
	Prec	*Rec*	*F1*	*Prec*	*Rec*	*F1*	*Prec*	*Rec*	*F1*
baseline$_{RoBERTa}$	28.59	34.69	26.46	39.11	62.07	45.75	33.85	48.38	36.10
best-competitor	49.21	46.32	46.22	81.99	80.79	**81.17**	65.60	63.55	63.70
dfki-mlst$_{DeBERTa\text{-}v3}$	61.96	53.30	**55.33**	81.08	79.25	78.78	71.52	66.28	**67.05**
DeBERTa-v1	64.05	57.14	**57.93**	79.04	78.19	78.12	71.55	67.66	**68.03**
RoBERTa	64.86	49.55	52.73	76.83	75.05	75.17	70.84	62.30	63.95
RemBERT	54.79	46.00	47.56	75.92	74.28	74.49	65.36	60.14	61.02
ELECTRA	46.18	39.37	41.41	81.23	81.07	80.53	63.70	60.22	60.97
BART	49.78	44.3	45.49	84.22	83.97	**83.68**	67.00	64.13	64.59
XLNet	55.51	48.28	48.80	73.88	72.93	72.05	64.69	60.61	60.42
Mistral-30	47.20	43.14	42.33	82.15	83.91	82.62	64.68	63.52	62.48
Llama-30	39.59	38.38	37.10	74.93	75.01	74.22	57.26	56.70	55.66
Mistral-LoRA	54.41	47.06	49.03	82.07	79.78	79.99	68.24	63.42	64.51
Llama-LoRA	51.72	42.95	44.89	77.03	75.55	75.99	64.38	59.25	60.44


General scores for the official test set in the GLOBAL setting
F1
0
10
20
30
40
50
60
70
36.1
68.03
67.05
dfki-mlst
63.95
61.02
60.97
64.59
60.42
62.48
55.66
64.51
60.44
baseline
deberta-v1
deberta-v3
roberta
rembert
electra
bart
xlnet
mistral-30
llama-30
mistral-lora
llama-lora

Model Selection

On the custom validation set DeBERTa-v3 outperforms the second best model DeBERTa-v1 by a small margin. But **DeBERTa-v1 actually works better than DeBERTa-v3 on the gold test data**.

BART demonstrates the best F1 scores in both **ILO-Focused (72.28 F1)** and **ILO-General (83.68 F1)** settings. However, it is much worse on the argument relation identification (-7.98% F1 in Focused and -9.84 % F1 in General).

Mistral and Llama do not show good results on the relation classification task. In both cases adding LoRA works better than tuning the last layers.

Data Augmentation: EDA & Paraphrases

Since we have only 1259 nodesets for training and the task is challenging, we also experiment with different ways to **extend the data** w/o modifying the nodeset structure, i.e. by changing the **node text**.

Easy Data Augmentation (EDA) can generate more data by randomly choosing a token to:

replace with a synonym from WordNet

delete or **insert** at random position

swap one token with another

As an alternative to EDA, we also **paraphrase the original data** with T5 fine-tuned on ChatGPT paraphrases.

Data Augmentation: Examples

Original L-node text:

“Claire Fox: that will show how virtuous I am”

EDA-paraphrased:

“Claire Fox: appearance that will show how virtuous I am”

T5-paraphrased:

“Claire Fox: My goodness will be demonstrated to others through this.”

Data Augmentation: Results

Focused

Model	ARI-Focused			ILO-Focused			GLOBAL-Focused		
	Prec	*Rec*	*F1*	*Prec*	*Rec*	*F1*	*Prec*	*Rec*	*F1*
baseline$_{RoBERTa}$	37.10	18.42	22.80	73.10	72.55	**72.09**	55.10	45.49	47.45
dfki-mlst$_{DeBERTa\text{-}v3}$	43.87	24.82	30.40	69.12	66.25	66.10	56.50	45.53	48.25
DeBERTa-v1$_{+paraphr_data}$	43.99	22.03	27.69	68.35	66.01	66.31	56.17	44.02	47.00
DeBERTa-v3$_{+EDA_sequential}$	48.08	25.26	30.66	65.72	62.21	62.71	56.90	43.73	46.69
DeBERTa-v3$_{+EDA\ \ combined}$	47.73	29.11	**34.16**	65.92	64.53	64.80	56.83	46.82	**49.48**

General

Model	ARI-General			ILO-General			GLOBAL-General		
	Prec	*Rec*	*F1*	*Prec*	*Rec*	*F1*	*Prec*	*Rec*	*F1*
baseline$_{RoBERTa}$	28.59	34.69	26.46	39.11	62.07	45.75	33.85	48.38	36.10
dfki-mlst$_{DeBERTa\text{-}v3}$	61.96	53.30	55.33	81.08	79.25	78.78	71.52	66.28	67.05
DeBERTa-v1$_{+paraphr_data}$	58.42	49.66	51.36	80.44	79.30	**79.16**	69.43	64.48	65.26
DeBERTa-v3$_{+EDA_sequential}$	62.53	52.97	54.88	78.31	75.79	75.92	70.42	64.38	65.40
DeBERTa-v3$_{+EDA_combined}$	60.21	56.75	**56.86**	78.50	78.12	77.95	69.36	67.44	**67.41**

Takeaway: combining the data brings improvements compared to vanilla DeBERTa-v3 in ARI and GLOBAL settings but results in worse scores on the ILO task.

Balancing Out: Weighted Loss & Re-sampling

DialAM training data has **imbalanced distribution** with 12 labels representing <1% of all the data.

Therefore, we experiment with **weighted loss**:

(a) **collect** the statistics on label distribution

(b) **compute** the class weight: $w_c = \frac{|D|}{|D_c| \cdot |C|}$

where *D* is the set of all samples (relation-class-pairs) and *C* is the set of all class labels

(c) **restrict** the range of weights within [1, 20] to avoid over-penalizing the classifier for very rare classes

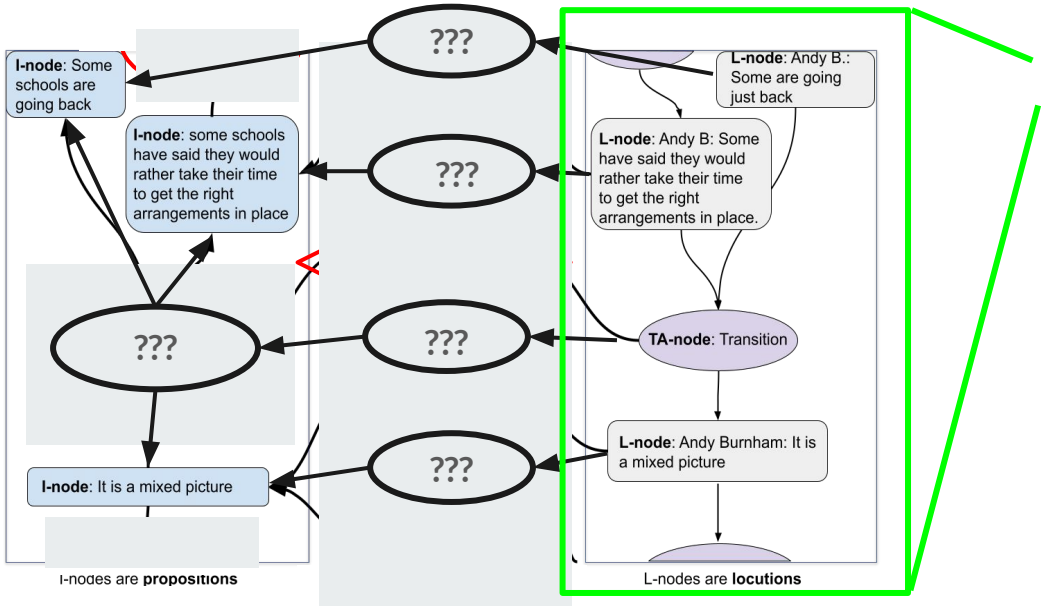
Balancing Out: Weighted Loss & Re-sampling

Training with weighted loss **improves** the results for the **ARI task** (both Focused and General) but has a significant **drop in performance** for the **ILO task**.

When we simply **remove all rare labels**, e.g. if they appear less than 5 times in the training data, we have an **overall drop in performance** for both tasks.

This shows that having even very few examples of rare classes can be beneficial for the classifier.

Recap: Encoding Candidate Tuples

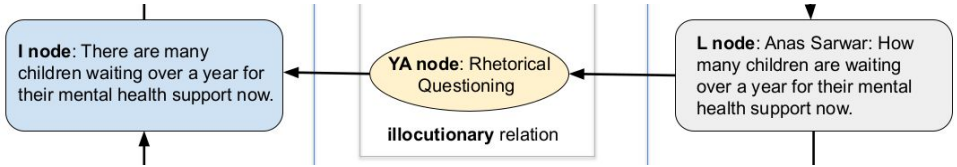


close to real world (text) data

→ **serialize L-nodes**

"Andy B.: Some are going just back</S> Andy B.: Some have said they would rather take their time to get the right arrangements in place.</S>Andy B.: It is a mixed picture."

Combining Node Texts: L + I-nodes



I and L nodes represent arguments and dialogue turns, correspondingly.

I-node text often includes some additional details, “summarizing” the content as an argument, while L node text represents an unedited dialogue turn.

We experiment with **combining both texts** when encoding the relations.

This results in the **best performance on the ILO task** and the combined text approach achieves overall best scores in the GLOBAL setting, outperforming the version that uses only L-node text by 3.4% F1 in Focused and 1.9% F1 in General.

However, this approach under-performs on the ARI task!

Training on *all* Nodesets

We also test whether **adding some of the previously blacklisted nodesets** can help to train the model.

In this setting we ignore the following issues:

- ambiguous direction of the *Inference* S-node
- no TA-node anchor for the *Inference* S-node
- S-node arguments are not unique

This way we increase the size of the training split **from 1259 to 1475** but the added nodesets have "worse" quality. This is reflected in the results because we observe a **drop in performance** compared to the original (filtered) version.

Recipe for Success @DialAM-2024

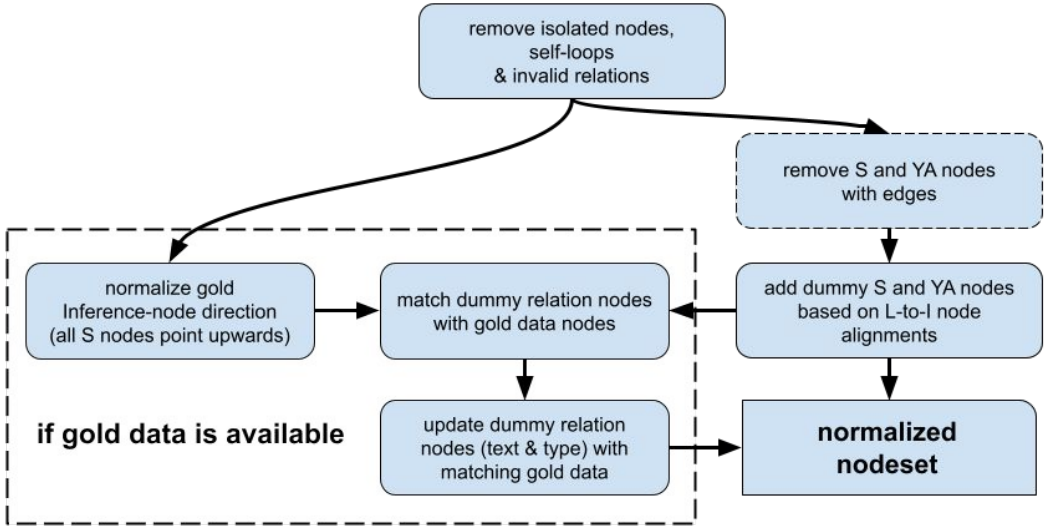
We used the **Pytorch-IE (PIE)** framework to do all the experiments. Our code is available at https://github.com/ArneBinder/dialam-2024-shared-task

So, let's have a look at the whole process once again using the analogy of baking a pie:

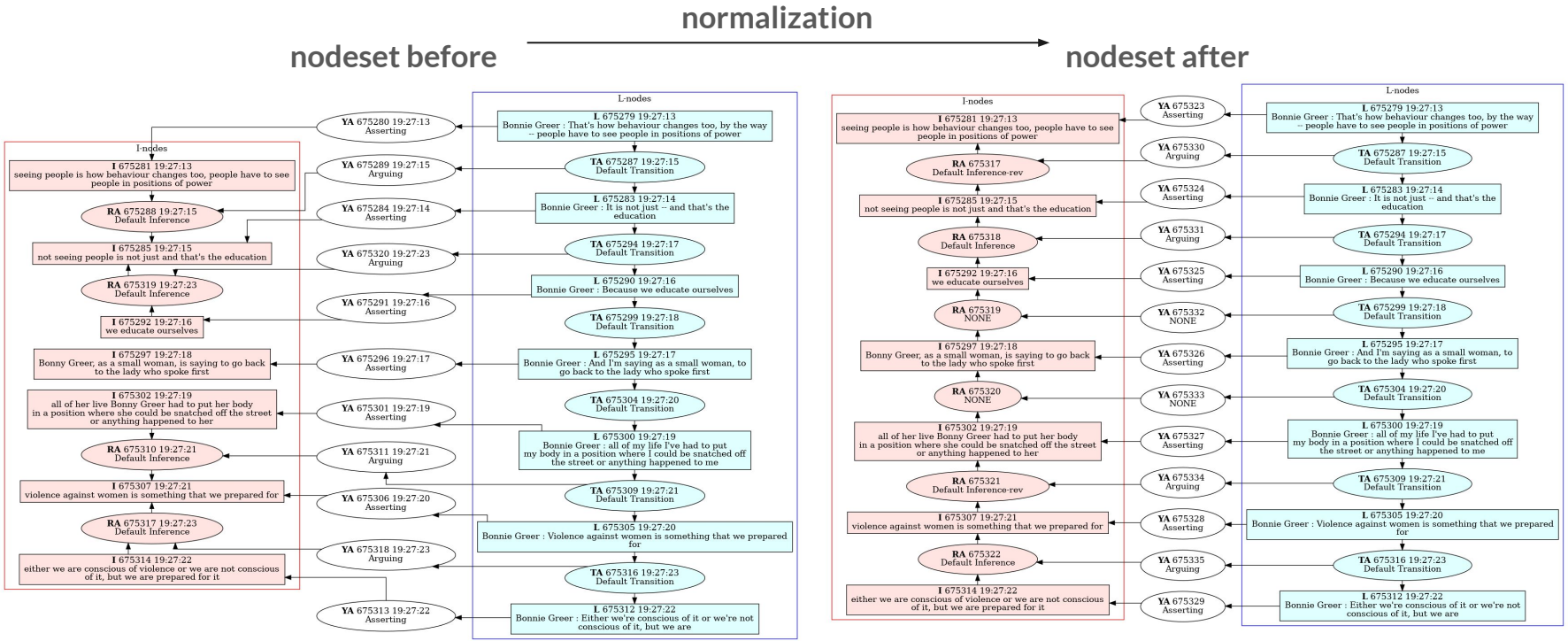
1. **Prepare** the ingredients
2. **Preheat** the oven
3. **Mix** well and fill in the form
4. **Bake** the pie in the oven
5. **Take out** and decorate
6. **Serve** and get some feedback

Step 1: Prepare the ingredients

Our **ingredients** are **nodesets** and they need to be pre-processed and normalized before training:



Step 1: Prepare the ingredients

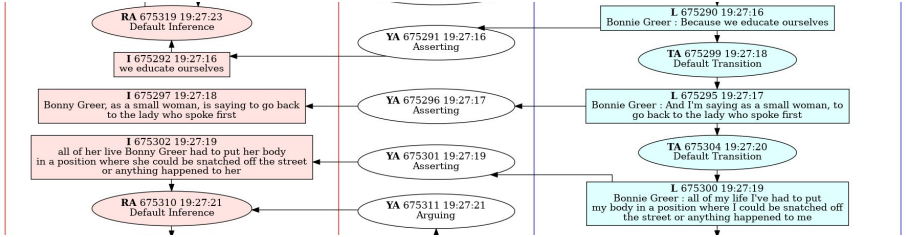


Step 1: Prepare the ingredients

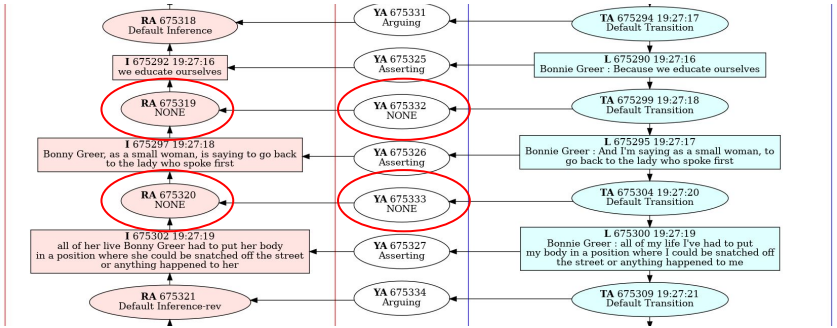
Adding dummy nodes for argumentative relations:

Let's connect all I-nodes following the dialogue flow (L-nodes on the right side with TA-transitions). Those that were not in the original annotations should be assigned label NONE.

nodeset before



nodeset after

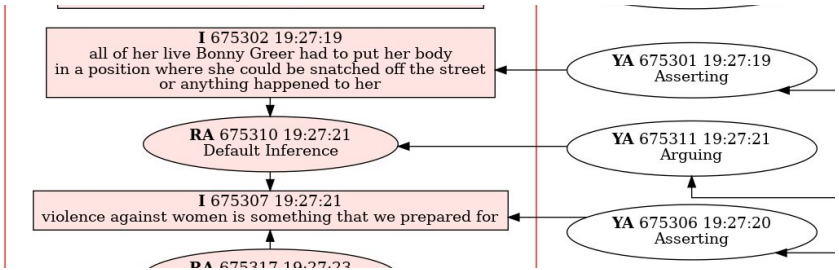


Step 1: Prepare the ingredients

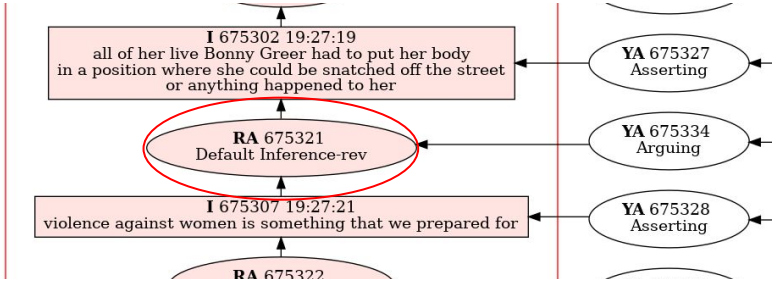
Normalize I-node direction s.t. all arrows connecting I-nodes point upwards.

If we change the original direction, we need to remember this, so we add the *-rev* (e.g. *Inference-rev*) suffix to indicate such cases.

nodeset before



nodeset after



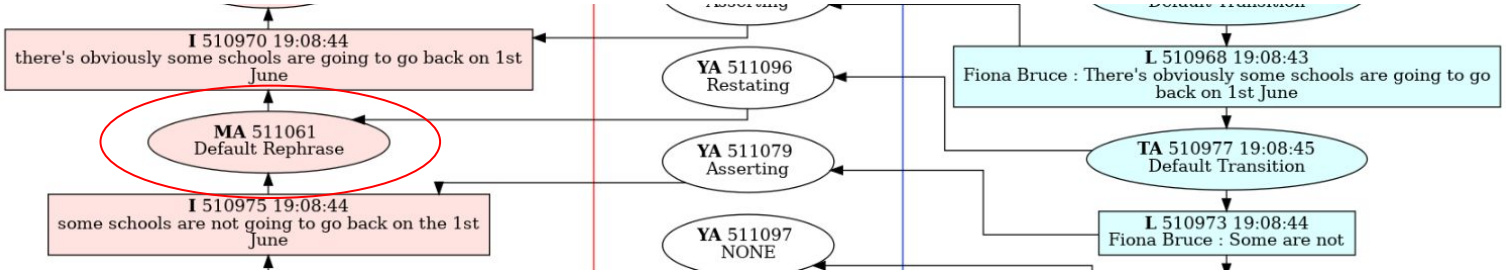
Step 2: Preheat the oven

Before we start baking our new model, we need to “preheat the oven” and **specify the hyperparameters**. In PIE this can be easily done via separate configuration files for your experiment, model, dataset etc.

```
CONFIG
├── datamodule
│   └── _target_: src.datamodules.PieDataModule
│       batch_size: 32
│       num_workers: 8
│       pin_memory: false
│       show_progress_for_encode: true
│
├── taskmodule
│   └── _target_: pie_modules.taskmodules.RETextClassificationWithIndicesTaskMod
│       tokenizer_name_or_path: facebook/bart-large
│       max_window: 512
│       relation_annotation: nary_relations
│       argument_role_to_marker:
│         s_nodes:source: S:S
│         s_nodes:target: S:T
│         ya_i2l_nodes:source: YA-I2L:S
│         ya_i2l_nodes:target: YA-I2L:T
│         ya_s2ta_nodes:source: YA-S2TA:S
│         ya_s2ta_nodes:target: YA-S2TA:T
│       collect_statistics: true
│
├── model
│   └── _target_: pie_modules.models.SequenceClassificationModelWithPooler
│       model_name_or_path: facebook/bart-large
│       learning_rate: 1.0e-05
```

Step 3: Mix well and fill in the form

Like the ingredients of a real pie, **relations** from our nodesets need to be **correctly mixed and prepared** as inputs for the model. So, how do we encode e.g. an argumentative relation between two I-nodes?



Step 3: Mix well and fill in the form

The argumentative S-node relation between two I-nodes can be encoded as **n-ary relation**:

arguments:

role: s_nodes:source:

LabeledSpan(start=533, end=559, text=”*Fiona Bruce : Some are not*”),

role: s_nodes:target:

LabeledSpan(start=455, end=532, text="*Fiona Bruce : There's obviously some schools … on 1st June*"),

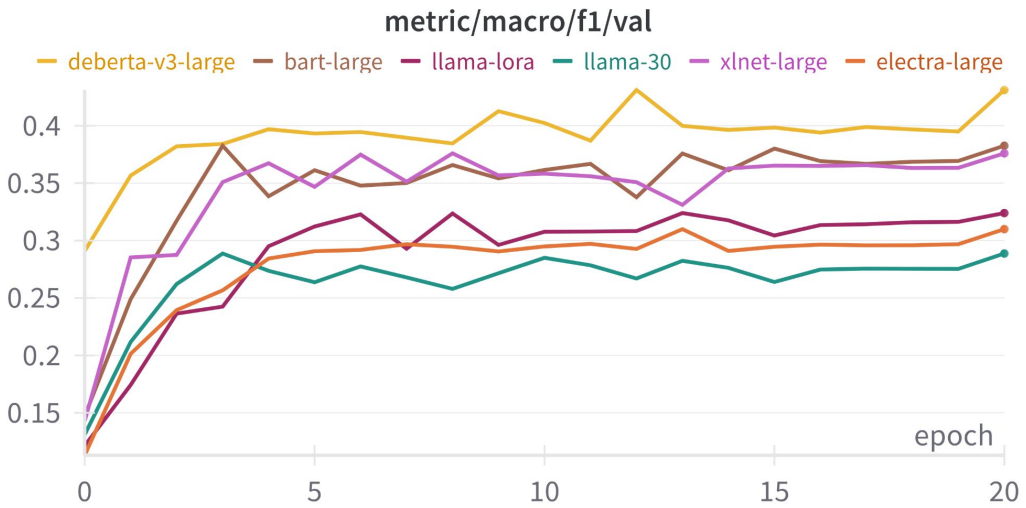
label = s_nodes:Default Rephrase,

Step 4: Bake the pie in the oven

Now we can start the actual **training** and test various models, e.g. DeBERTa, RoBERTa, ELECTRA, BART, XLNet, Llama etc.

Good news: we can (mostly) reuse the configuration files, taskmodule and our data pre-processing code.

Monitoring our baking process:



Step 5: Take out and decorate

After we trained the model we can make predictions, but we also need to specify **how the output should be "decorated"**.

For the DialAM shared task the data are expected to be in a JSON format like this:


```
{
  "nodes": [
    {
      "nodeID": "719340",
      "text": "Lucy Frazer : In my constituency I went to visit a community land trust",
      "type": "L",
      "timestamp": "2021-06-10 20:24:51"
    },
    {
      "nodeID": "719341",
      "text": "Asserting",
      "type": "YA",
      "timestamp": "2021-06-10 20:24:51",
      "scheme": "Asserting",
      "schemeID": "74"
    },
    ...

  ],
  "edges": [
    {
      "edgeID": "924881",
      "fromID": "719341",
      "toID": "719346",
      "formEdgeID": null
    },
    {
      "edgeID": "924882",
      "fromID": "719340",
      "toID": "719341",
      "formEdgeID": null
    },
    ...
  ]
}
```

Step 6: Serve and get some feedback

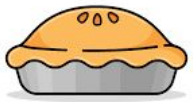
Now let's proceed with evaluation! The official script outputs ILO (illocutionary) and ARI (argumentative) scores at the **focused and general levels**: precision, recall and F1.

Global scores are averaged ARI and ILO results.

Focused: looking only at the related arguments/locutions (according to the gold standard)

General: evaluating the complete argument maps (including unrelated nodes)

Based on the feedback (evaluation script output) we can adjust our recipe and make another pie :-)



Global evaluation results:

	Model	Rank	Precision	Recall	F1-score
Focused	DFKI-MLST	1st	**56.50**	**45.53**	**48.25**
	RoBERTa-BL	2nd	55.1	45.49	47.45
	Pungene	3rd	50.68	43.41	45.23
	Pokemon	4th	50.20	41.15	43.64
	KnowComp	5th	35.95	23.56	26.70
	misaka	5th	35.95	23.56	26.70
	Turiya	7th	31.38	15.15	18.53
	Majority-BL	8th	0	0	0
General	DFKI-MLST	1st	**71.52**	**66.28**	**67.05**
	Pungene	2nd	65.60	63.55	63.70
	KnowComp	3rd	57.39	55.03	55.82
	misaka	3rd	57.39	55.03	55.82
	Pokemon	5th	44.20	55.57	45.00
	Turiya	6th	41.09	44.29	42.03
	RoBERTa-BL	7th	33.85	48.38	36.10
	Majority-BL	8th	31.75	33.09	32.40

DialAM-2024 Takeaways

- framing the dialogue argument mining task as **n-ary relation classification** over dialogue turns
- achieving the **best scores in the global setting**: 48.25 F1 @Focused and 67.05 F1 @General
- experimenting with **11 base models**: from RoBERTa to Llama and Mistral
- exploring **extra configurations** with data augmentation, node text combination, weighted loss etc.

Arne Binder, Tatiana Anikina, Leonhard Hennig, and Simon Ostermann

dfki ai
Deutsches Forschungszentrum
für Künstliche Intelligenz
German Research Center for Artificial Intelligence

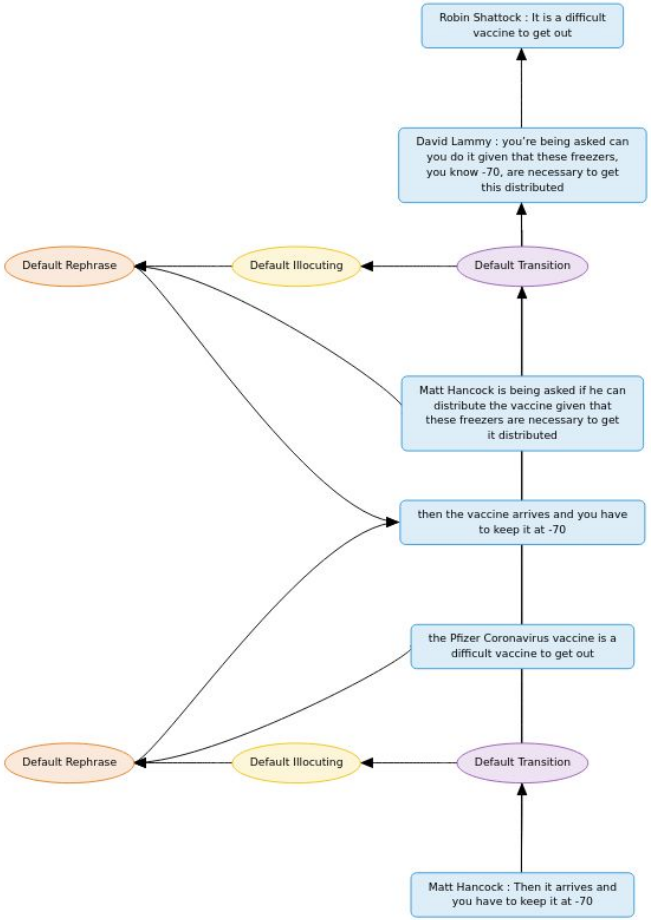
code & paper & slides

source code: github.com/ArneBinder/dialam-2024-shared-task

This work has been supported by the German Ministry of Education and Research (BMBF) as part of the project TRAILS.

Backup

Special Maps w/o L-to-I-node Alignment



Some data were **splitted into chunks** and analysed independently resulting in individual maps corresponding to each section.

The connections between the chunks are sparse and many L and I-nodes have **missing connections**.

Our pre-processing relies on the **L-to-I node alignment** but special maps do not allow us to use alignments in most cases.

Solution: black-list 23 nodesets (1.56% of the original data).

So, how can you win the Shared Task?

Our Recipe:

1. **Start** as early as you can (one week probably won't do)
2. **Find** people who are competent and/or interested in the topic
3. **Distribute** the tasks accordingly
4. **Follow data2model approach** and spend time on (double-and-triple)-checking your data
5. **Ask** questions whenever you have them
6. **Methodically try** different configurations (and document them in a log ;-)
7. **Have fun** and don’t give up!