

MUNI
FI



Constructing Datasets from Dialogue Data

Mgr. Ondřej Sotolář

FI MUNI

Summary

1. Storing & Retrieving
2. Training Examples
3. Analyzing & Splitting Data

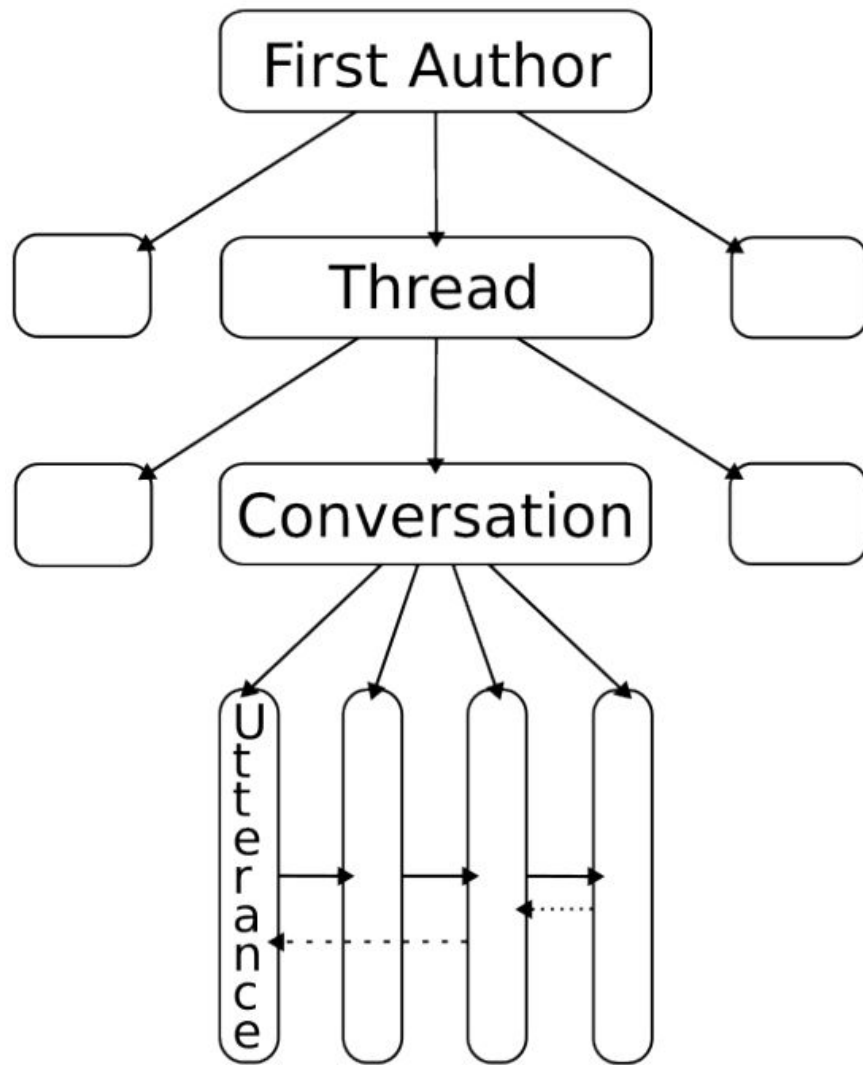
Dialogue

Time	Author	Utterance
01:44:07	John	I'll finish the Math task tomorrow
01:44:13	John	Like, I really have to do it
01:44:28	Tim	The math task looks easy to me
01:44:49	Tim	You have 6 hours to deadline, chill
01:46:34	John	But I'm really tired after the day
01:47:51	Tim	I'm having some tea and I'm super

Dialogue – Utterance Classification

Utterance	Class
I'll finish the Math task tomorrow	none
Like, I really have to do it	none
The math task looks easy to me	Emotional Support
You have 6 hours to deadline, chill	Emotional Support
But I'm really tired after the day	none
I'm having some tea and I'm super	none

Storing and
Retrieving



Storing

column_name	type
<u>uid first author</u>	integer
<u>thread id</u>	integer
<u>conversation id</u>	integer
<u>line num</u>	integer
<i>time</i>	timestamp
<i>uid_author</i>	integer
<i>reaction_to</i>	integer
<i>label</i>	integer
<i>text</i>	varchar

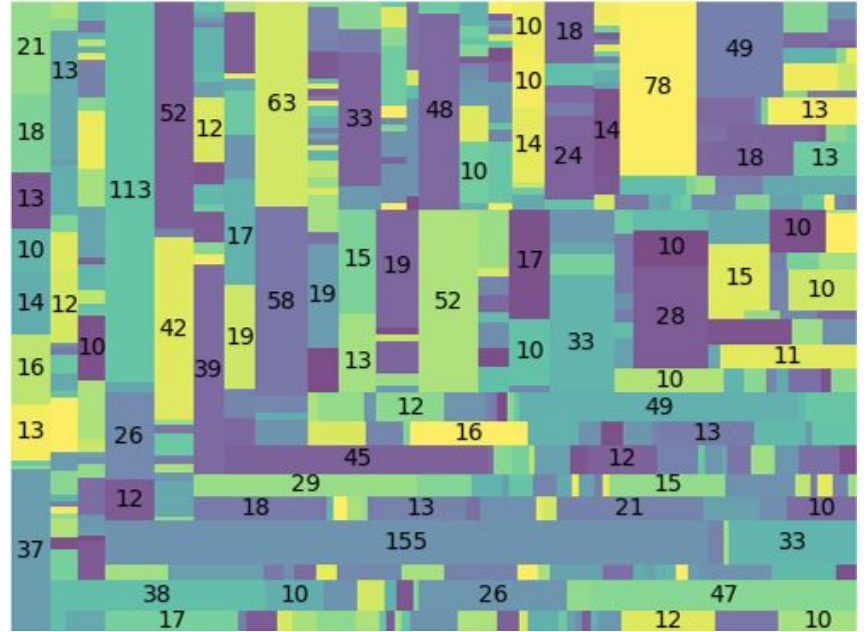
Retrieving

```
SELECT  
  ARRAY_AGG(d.line_text::TEXT ORDER BY line_num asc) AS texts,  
  ARRAY_AGG(d.label::INTEGER ORDER BY line_num asc) AS labels  
FROM dialogues s  
WHERE d.label = 0 OR d.label = 1  
GROUP BY d.uid_first_author, d.thread_id, d.conversation_id;
```

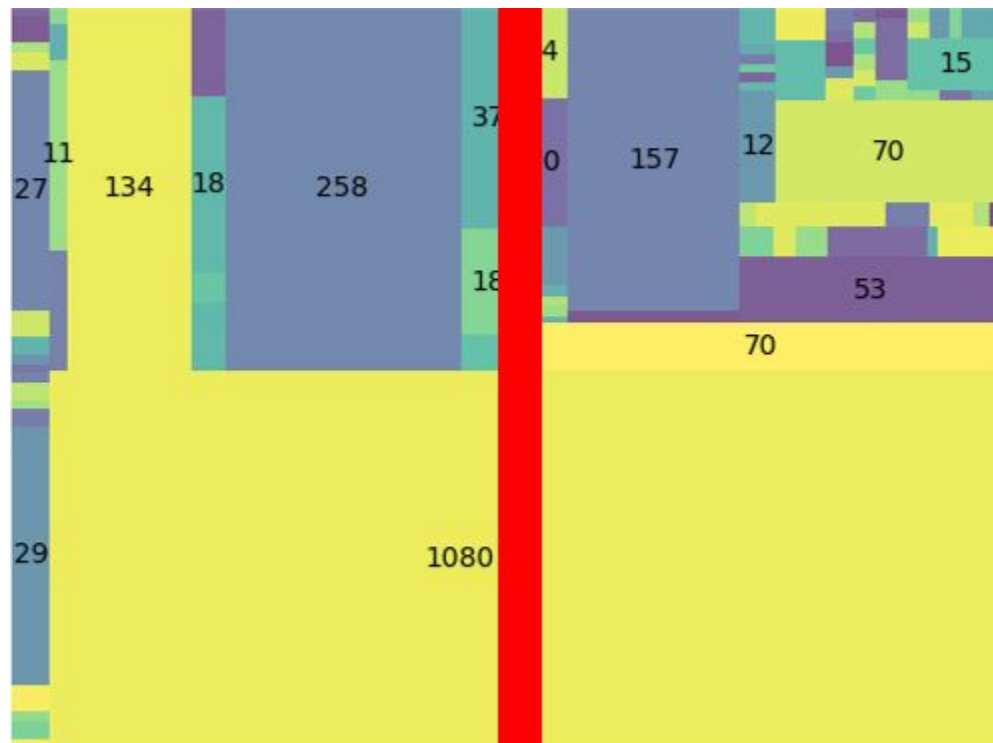
Creating Training Examples with Context: Soft Char Limit

Time	Author	Utterance	Class
01:44:07	John	I'll finish the Math task tomorrow	none
01:44:13	John	Like, I really have to do it	none
01:44:28	Tim	The math task looks easy to me	Emotional Support
01:44:49	Tim	You have 6 hours to deadline, chill	Emotional Support
01:46:34	John	But I'm really tired after the day	none
01:47:51	Tim	I'm having some tea and I'm super	none

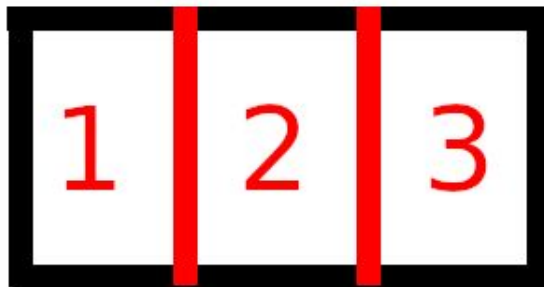
Analyzing Dialogue Contributor Distribution



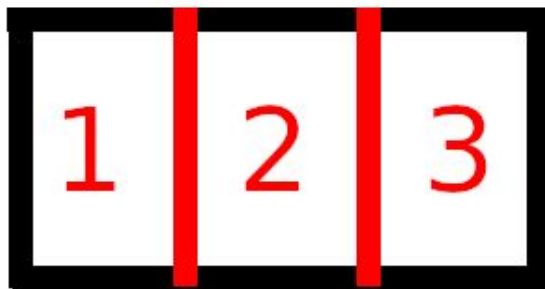
Naive Splits Can Introduce New Biases



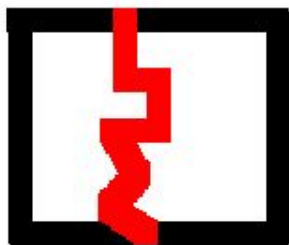
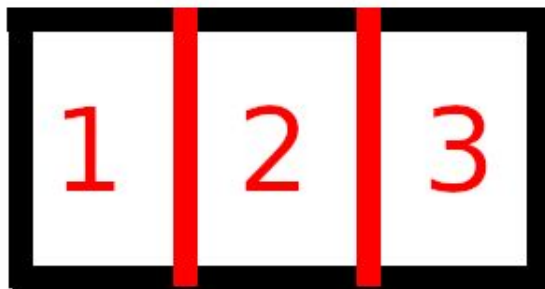
Better Splitting Algorithm: Evaluation



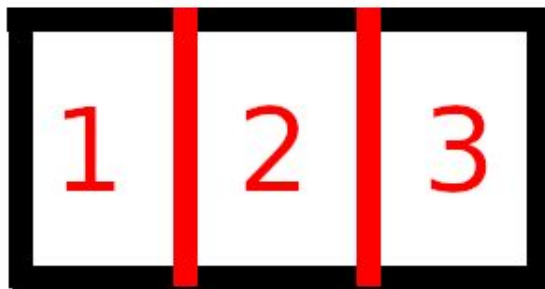
Better Splitting Algorithm: Evaluation



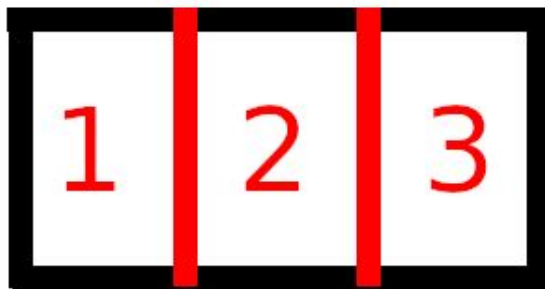
Better Splitting Algorithm: Evaluation



Better Splitting Algorithm: Evaluation



Better Splitting Algorithm: Evaluation



$$D_1 < D_2$$

Algorithm: 1 – computing conflicts

conversation	thread	#0	#1	#a1	#a2	#a3	#a4	#a5
1	1	10	1	5	6	0	0	0
2	1	5	0	3	2	0	0	0
3	1	7	1	4	4	0	0	0
4	2	3	0	2	0	1	0	0
5	2	5	1	3	0	3	0	0
6	3	10	0	5	0	0	5	0
7	3	5	2	3	0	0	4	0
8	4	10	1	0	0	0	0	11

$T \leftarrow \text{group_by_author_tuples}(E);$

$G \leftarrow (g_1, \dots, g_k);$

$R \leftarrow \emptyset;$

while $\exists t \in T : \neg \exists g \in G : t \in g$ **do**

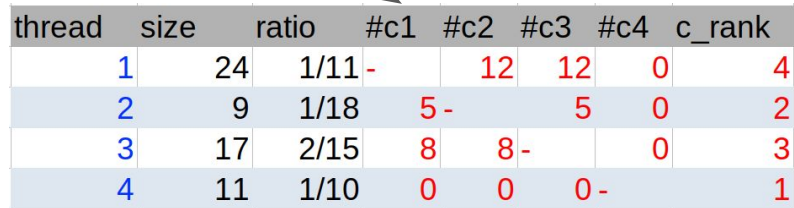
$t \leftarrow t$ with $\min(t.c_rank)$, **if** tied **then** use $\max(t.size)$;

$g \leftarrow \text{best_group}(G, t, \max_{c_rank}, \frac{\text{size}(E)}{k}, \text{label_ratio}(E));$

$g.add(t);$

$R \leftarrow T \setminus \{t : t \in g, g \in G\};$

return $G, R;$



thread	size	ratio	#c1	#c2	#c3	#c4	c_rank
1	24	1/11	-	12	12	0	4
2	9	1/18	5	-	5	0	2
3	17	2/15	8	8	-	0	3
4	11	1/10	0	0	0	-	1

Algorithm: 2 – putting threads into best fitting groups

thread	size	ratio	#c1	#c2	#c3	#c4	c_rank
1	24	1/11 -		12	12	0	4
2	9	1/18	5 -		5	0	2
3	17	2/15	8	8 -		0	3
4	11	1/10	0	0	0 -		1

$k=2$

group	size	ratio	members
1			
2			

$g_m \leftarrow$ group with minimum conflicts with t ;
if $g_m.conflicts(t) > max_{c_rank}$ **then**
 | **return** \emptyset ;
 $g_n \leftarrow$ group with maximum $|g_n.size - size_{desired}|$ with t ;
 $g_o \leftarrow$ group with maximum $|g_n.class_ratio - class_ratio_{desired}|$ with t ;
 $g \leftarrow$ select from $\{g_m, g_n, g_o\}$ with most votes, **if tied then** take g_m ;
return g

Algorithm: 2 – putting threads into best fitting groups

thread	size	ratio	#c1	#c2	#c3	#c4	c_rank
1	24	1/11	-	12	12	0	4
2	9	1/18	5	-	5	0	2
3	17	2/15	8	8	-	0	3
4	11	1/10	0	0	0	-	1

$k=2$

group	size	ratio	members
1	11	1/10	4
2	9	1/8	2

Algorithm: 2 – putting threads into best fitting groups

thread	size	ratio	#c1	#c2	#c3	#c4	c_rank
1	24	1/11	-	12	12	0	4
2	9	1/18	5	-	5	0	2
3	17	2/15	8	8	-	0	3
4	11	1/10	0	0	0	-	1

$k=2$

group	size	ratio	members
1	11	1/10	4
2	33	1/10	2,1

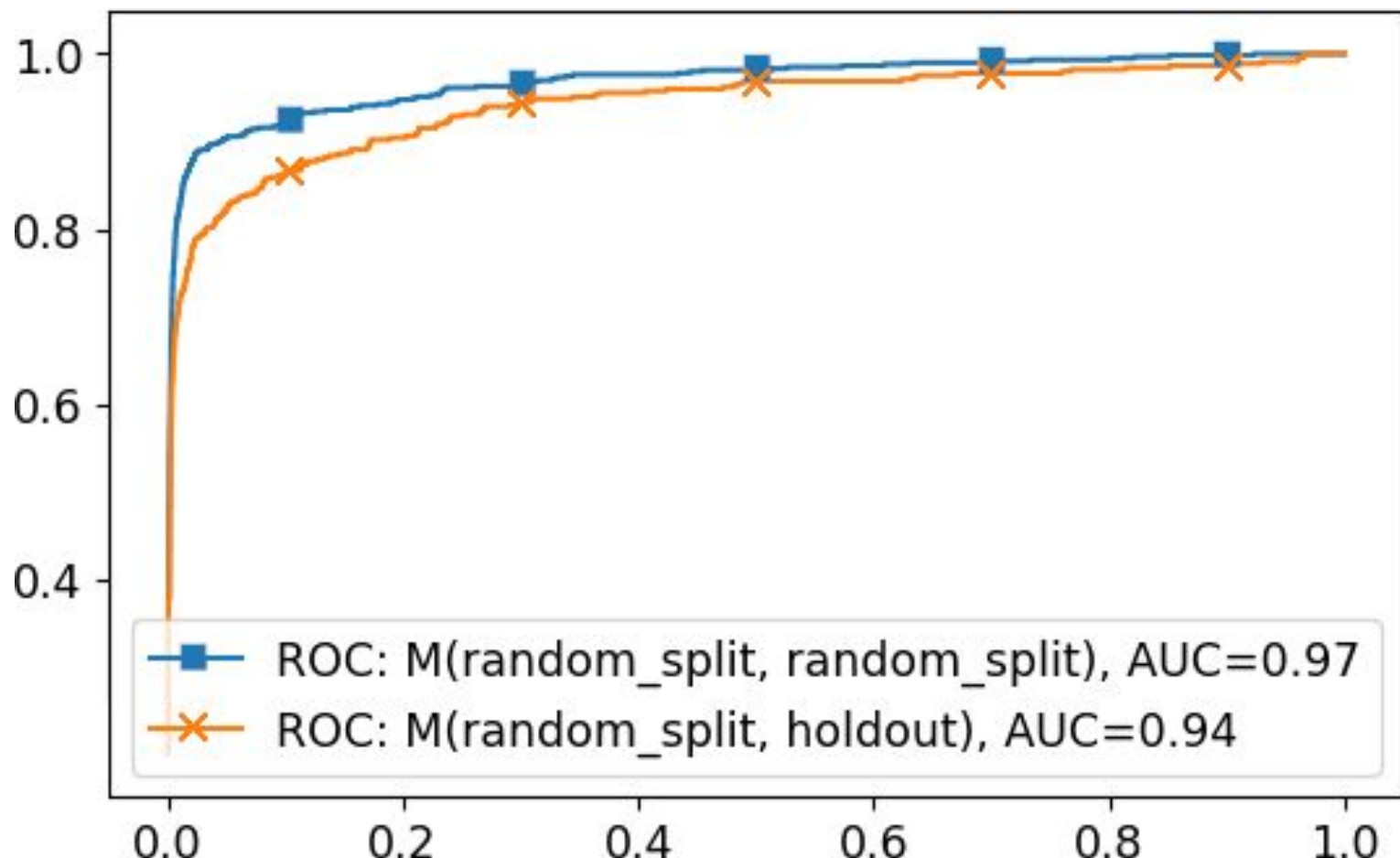
Algorithm: 2 – putting threads into best fitting groups

thread	size	ratio	#c1	#c2	#c3	#c4	c_rank
1	24	1/11	-	12	12	0	4
2	9	1/18	5	-	5	0	2
3	17	2/15	8	8	-	0	3
4	11	1/10	0	0	0	-	1

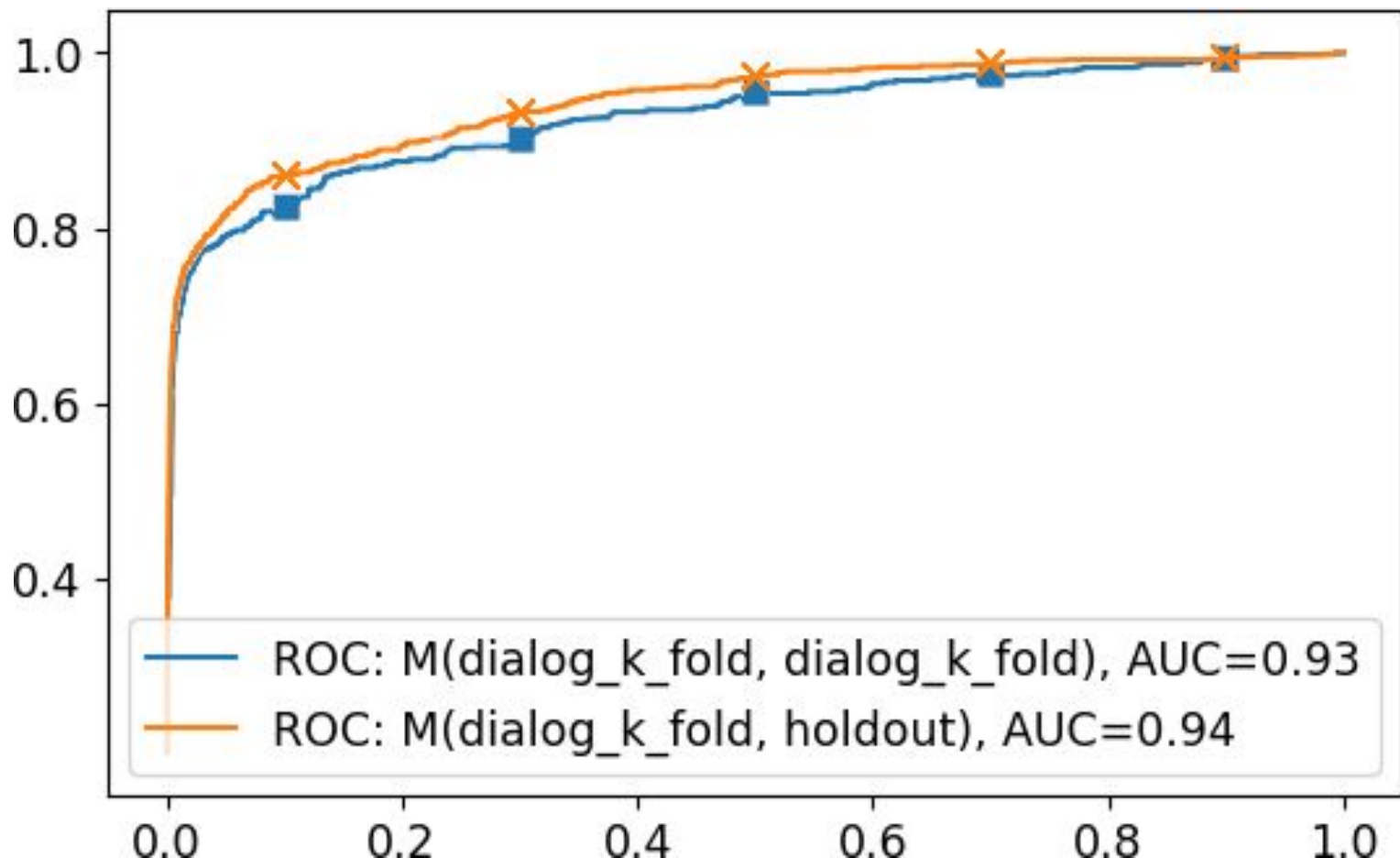
$k=2$

group	size	ratio	members
1	28	3/25	4,3
2	33	1/10	2,1

Results:
Naive



Results:
Proposed



Summary

1. Practical

- a. Storing & retrieving
- b. Construncting examples
- c. Analyzing

2. Test for splitting algorithms

3. Splitting algorithm

Limitations and Future Work

- Evaluation on public dataset
- Measure statistical significance of results
- Provide implementation