DOC2PPT: Automatic Slide Deck Generation from Documents

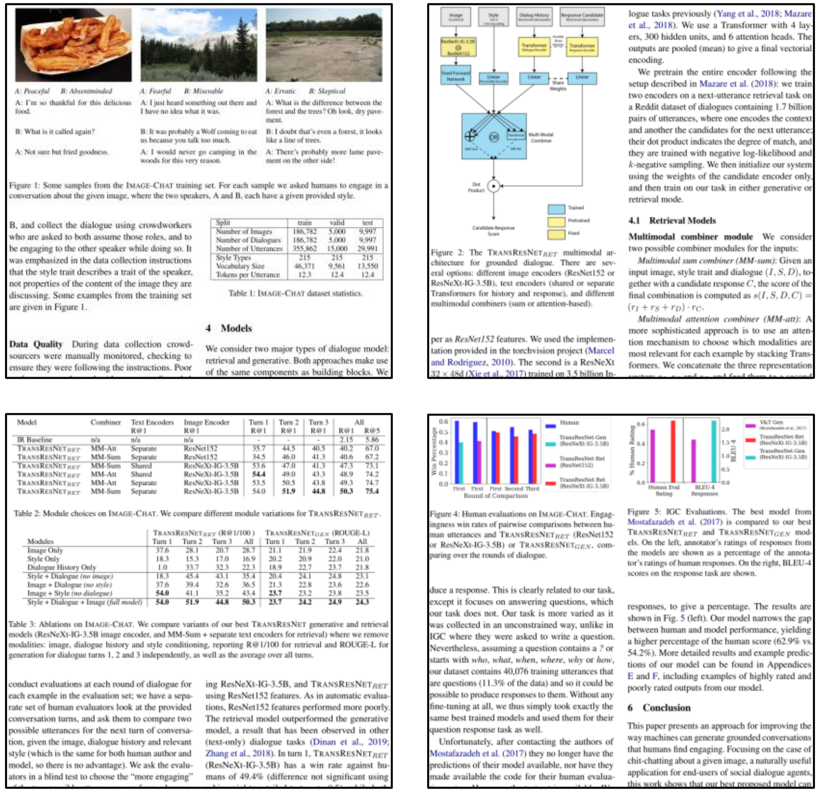
AAAI’22



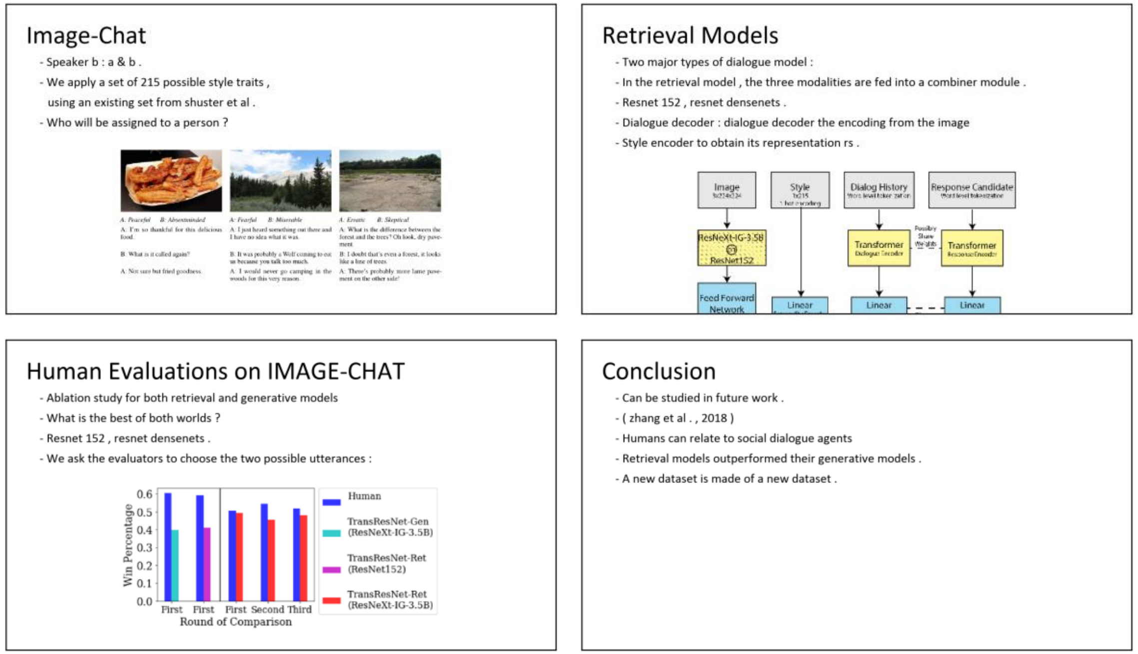

Tsu-Jui Fu William Wang Daniel McDuff Yale Song

DOC2PPT

- Generate a **slide** from an academic **paper**



DOC2PPT



DOC2PPT

- Multi-modal summarizer
  - **Text Summarization + Figure Retrieval + Multi-Page**

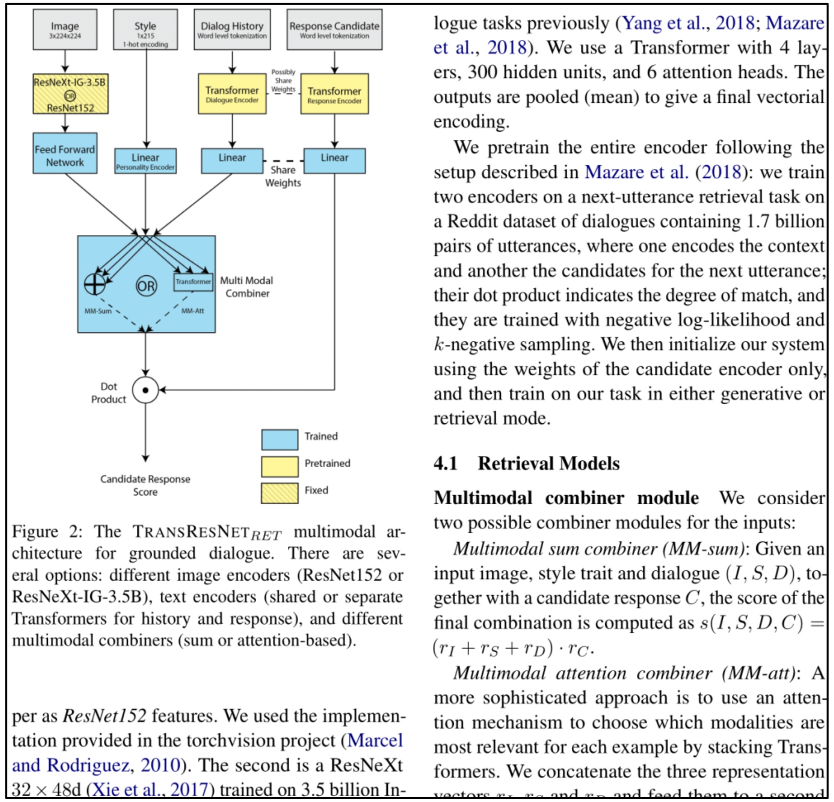


Figure 2: The TRANSRESNET$_{RET}$ multimodal architecture for grounded dialogue. There are several options: different image encoders (ResNet152 or ResNeXt-IG-3.5B), text encoders (shared or separate Transformers for history and response), and different multimodal combiners (sum or attention-based).

per as *ResNet152* features. We used the implementation provided in the torchvision project (Marcel and Rodriguez, 2010). The second is a ResNeXt 32×48d (Xie et al., 2017) trained on 3.5 billion In-

logue tasks previously (Yang et al., 2018; Mazare et al., 2018). We use a Transformer with 4 layers, 300 hidden units, and 6 attention heads. The outputs are pooled (mean) to give a final vectorial encoding.

We pretrain the entire encoder following the setup described in Mazare et al. (2018): we train two encoders on a next-utterance retrieval task on a Reddit dataset of dialogues containing 1.7 billion pairs of utterances, where one encodes the context and another the candidates for the next utterance; their dot product indicates the degree of match, and they are trained with negative log-likelihood and k-negative sampling. We then initialize our system using the weights of the candidate encoder only, and then train on our task in either generative or retrieval mode.

4.1 Retrieval Models

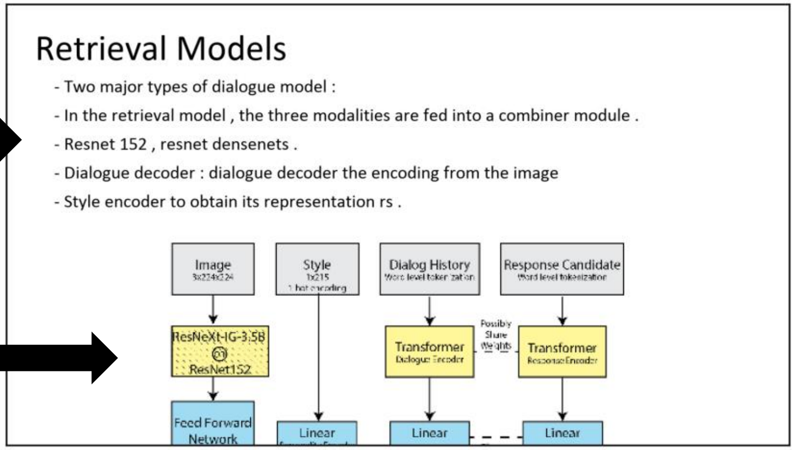
Multimodal combiner module We consider two possible combiner modules for the inputs:

Multimodal sum combiner (MM-sum): Given an input image, style trait and dialogue (I, S, D), together with a candidate response C, the score of the final combination is computed as $s(I, S, D, C) = (r_I + r_S + r_D) \cdot r_C$.

Multimodal attention combiner (MM-att): A more sophisticated approach is to use an attention mechanism to choose which modalities are most relevant for each example by stacking Transformers. We concatenate the three representation

Text Summarization

Figure Retrieval



-
- **Multi-Page**
-

Dataset Building

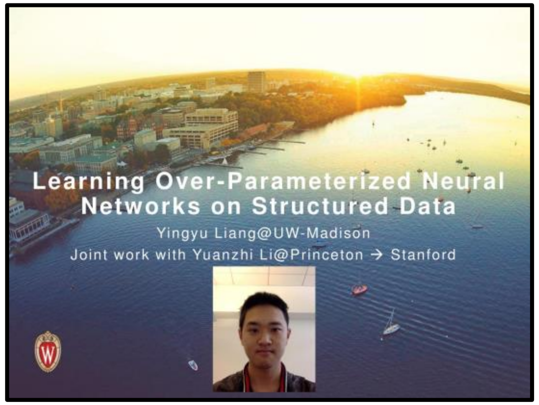
- Crawl **paper-slide pairs** from AI conferences
  - Computer Vision (CVPR, ECCV, …)
  - Natural Language Processing (ACL, NAACL, …)
  - Machine Learning (ICLR, ICML, …)

- **5,873** in total
  - 4,686 / 592 / 595 (train / val / test)

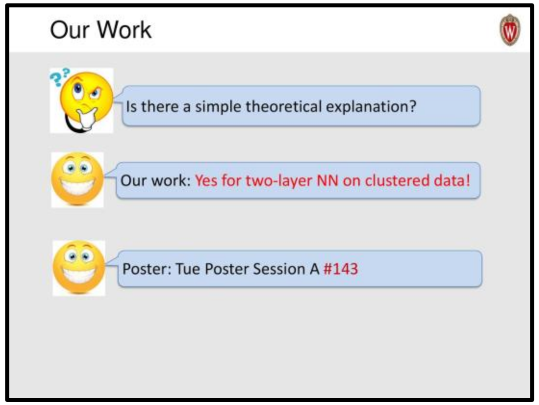
- To prepare the data for training, needs some **preprocessing** in advance

Dataset Building

- Extract **text content** from a slide
  - Azure CV to do **O**ptical **C**haracter **R**ecognition (OCR)



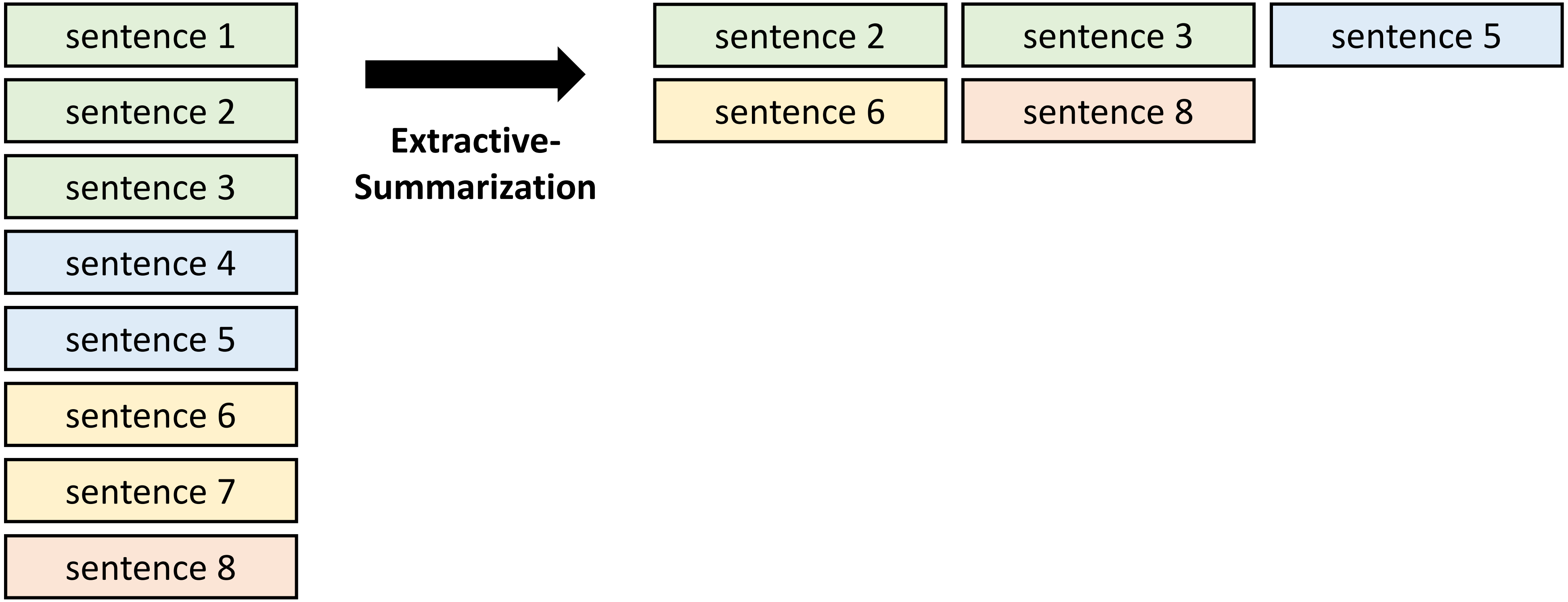
- Learning Over-Parameteiized –Neural
- Networks on Structured Data
- Yingyu Liang@UWLMadison
- Joint work with Yuanzhi Li@Princeton -Y Stanford



- Our Work
- Is there a simple theoretical explanation?
- Our work: Yes for two-layer NN on clustered data!
- Poster: Tue Poster Session A #143

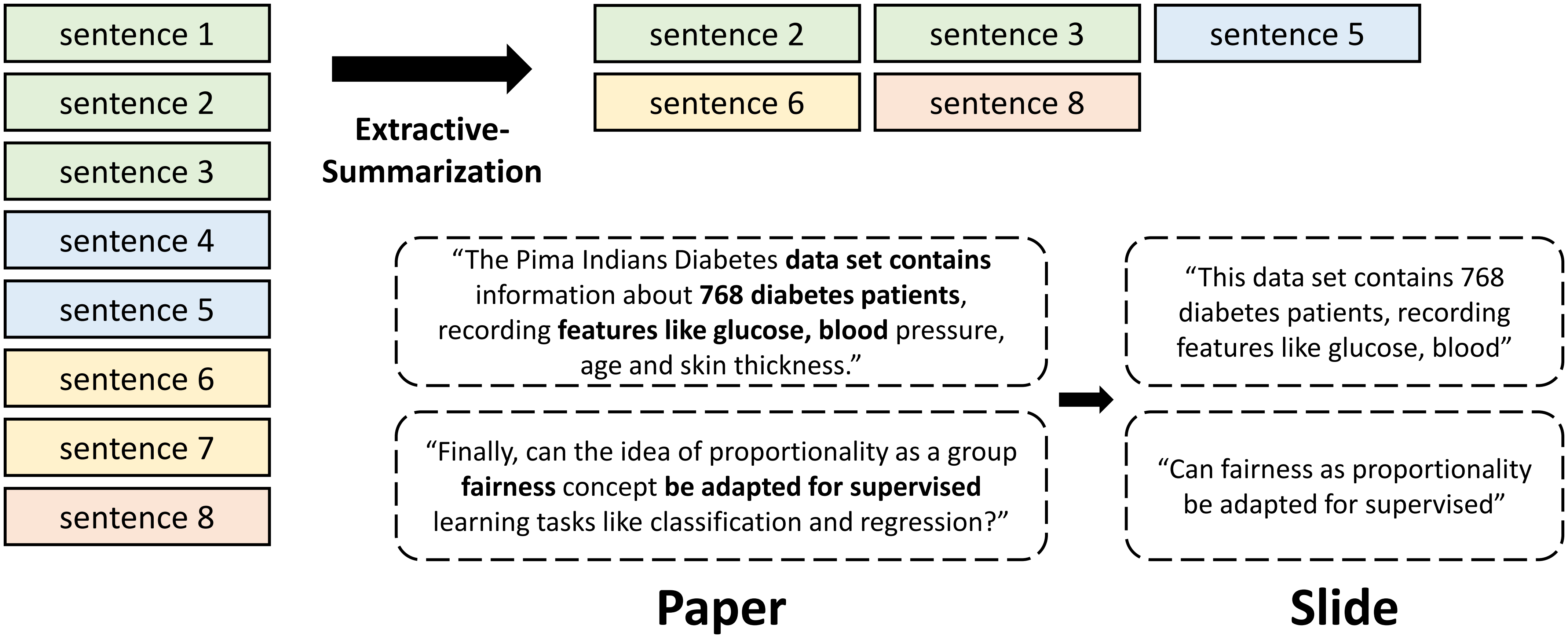
Dataset Building

- **Match sentences** from slide to paper
  - **Extractive**-based summarization



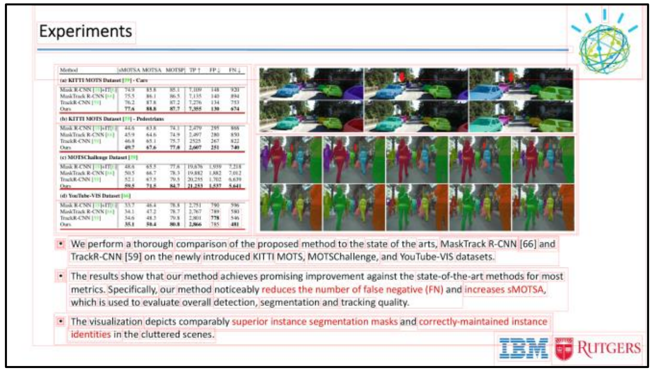
Dataset Building

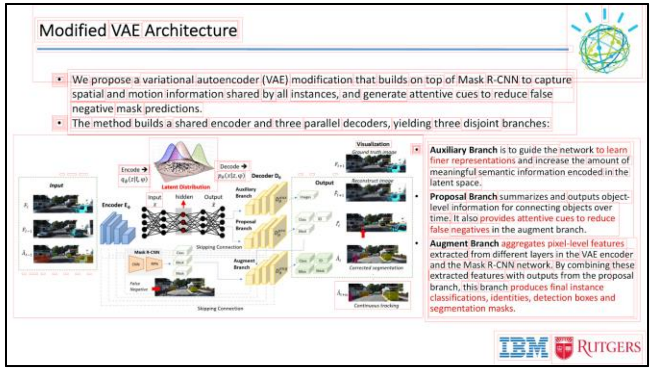
- **Match sentences** from slide to paper
  - **Extractive**-based summarization



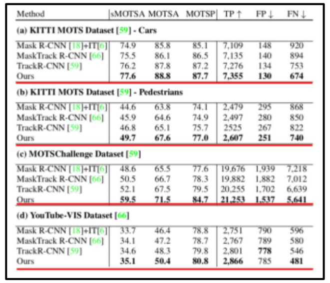
Dataset Building

- **Match figures** from slide to paper
  - **CNN feature** to do similarity matching

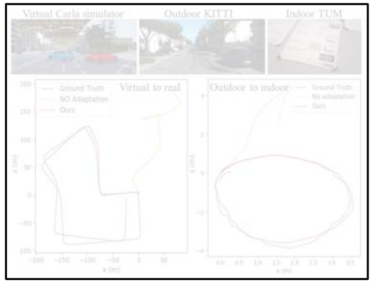


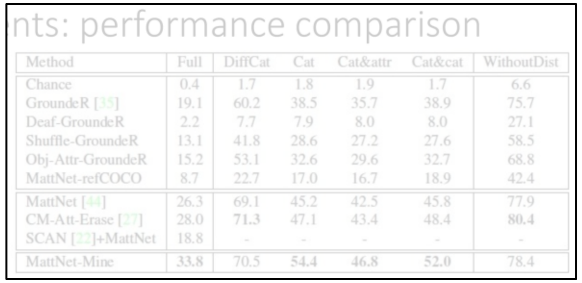


Slide



Method	sMOTSA	MOTSA	MOTSP	TP ↑	FP ↓	FN ↓
(a) KITTI MOTS Dataset [59] - Cars						
Mask R-CNN [18]+IT[6]	74.9	85.8	85.1	7,109	148	920
MaskTrack R-CNN [66]	75.5	86.1	86.5	7,135	140	894
TrackR-CNN [59]	76.2	87.8	87.2	7,276	134	753
Ours	**77.6**	**88.8**	**87.7**	**7,355**	**130**	**674**
(b) KITTI MOTS Dataset [59] - Pedestrians						
Mask R-CNN [18]+IT[6]	44.6	63.8	74.1	2,479	295	868
MaskTrack R-CNN [66]	45.9	64.6	74.9	2,497	280	850
TrackR-CNN [59]	46.8	65.1	75.7	2525	267	822
Ours	**49.7**	**67.6**	**77.0**	**2,607**	**251**	**740**
(c) MOTSChallenge Dataset [59]						
Mask R-CNN [18]+IT[6]	48.6	65.5	77.6	19,676	1,939	7,218
MaskTrack R-CNN [66]	50.5	66.7	78.3	19,882	1,882	7,012
TrackR-CNN [59]	52.1	67.5	79.5	20,255	1,702	6,639
Ours	**59.5**	**71.5**	**84.7**	**21,253**	**1,537**	**5,641**
(d) YouTube-VIS Dataset [66]						
Mask R-CNN [18]+IT[6]	33.7	46.4	78.8	2,751	790	596
MaskTrack R-CNN [66]	34.1	47.2	78.7	2,767	789	580
TrackR-CNN [59]	34.6	48.3	79.8	2,801	**778**	546
Ours	**35.1**	**50.4**	**80.8**	**2,866**	785	**481**





nts: performance comparison

Method	Full	DiffCat	Cat	Cat&attr	Cat&cat	WithoutDist
Chance	0.4	1.7	1.8	1.9	1.7	6.6
GroundeR [35]	19.1	60.2	38.5	35.7	38.9	75.7
Deaf-GroundeR	2.2	7.7	7.9	8.0	8.0	27.1
Shuffle-GroundeR	13.1	41.8	28.6	27.2	27.6	58.5
Obj-Attr-GroundeR	15.2	53.1	32.6	29.6	32.7	68.8
MattNet-refCOCO	8.7	22.7	17.0	16.7	18.9	42.4
MattNet [44]	26.3	69.1	45.2	42.5	45.8	77.9
CM-Att-Erase [27]	28.0	71.3	47.1	43.4	48.4	80.4
SCAN [22]+MattNet	18.8	-	-	-	-	-
MattNet-Mine	33.8	70.5	54.4	46.8	52.0	78.4

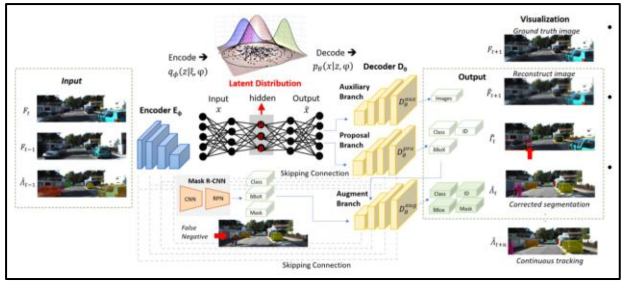
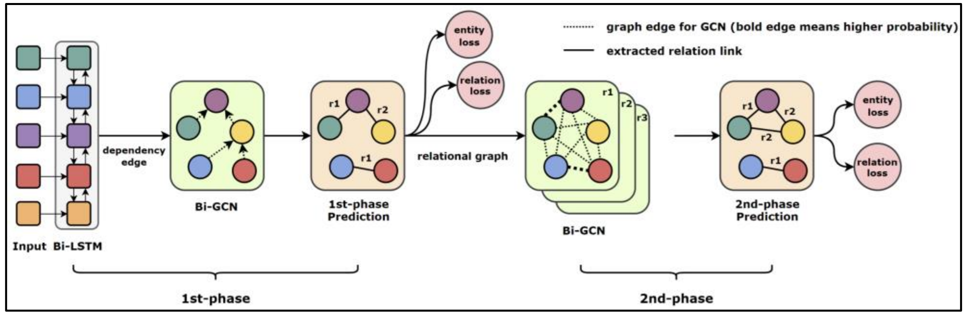


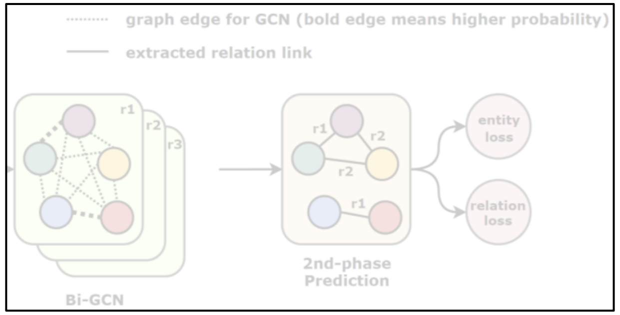
Figure from Paper

Dataset Building

- **Match figures** from slide to paper
- **Not** always perfect (currently 50.5% F1)
  - Leave as future work for **better label** to learn from



Partial Matching



Method	NYT			WebNLG		
	Precision	Recall	F1	Precision	Recall	F1
NovelTagging	62.4%	31.7%	42.0%	**52.5%**	19.3%	28.3%
OneDecoder	59.4%	53.1%	56.0%	32.2%	28.9%	30.5%
MultiDecoder	61.0%	56.6%	58.7%	37.7%	36.4%	37.1%
GraphRel_{1p}	62.9%	57.3%	60.0%	42.3%	39.2%	40.7%
GraphRel_{2p}	**63.9%**	**60.0%**	**61.9%**	44.7%	**41.1%**	**42.9%**

Different Expression

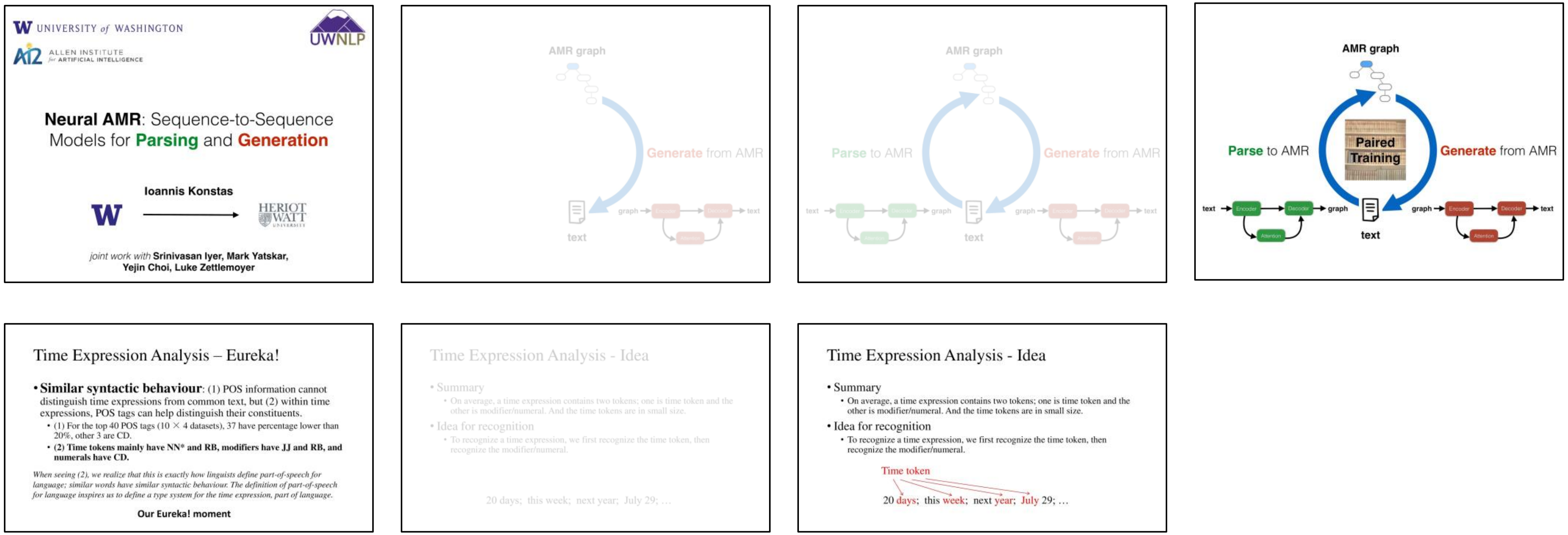
Method	P	R	F1	NER
NovelTag	62.4%	31.7%	42.0%	-
CopyRE	61.0%	56.6%	58.7%	-
GraphRel_{1p}	62.9%	57.3%	60.0%	88.8%
GraphRel_{2p}	**63.9%**	**60.0%**	**61.9%**	**89.2%**

Dataset Building

- **Match figures** from slide to paper
- **Not** always perfect (currently 50.5% F1)
- Apply **human labeling** for testing set
    - **Golden testing set** for fair evaluation

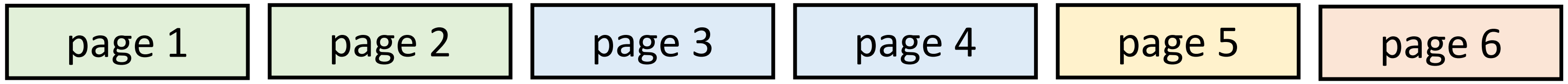
Dataset Building

- Remove the **progressive** page
  - **OCR cover rate** > 80% (Acc ~90%)
  - Keep the **last** one



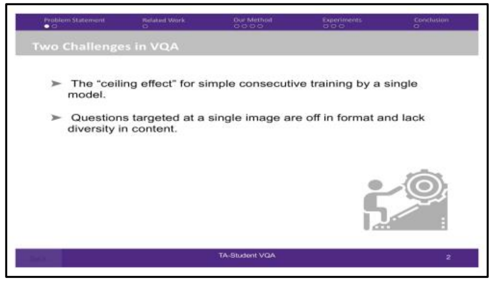
Dataset Building

- Generate pages **for each section** and combine them all
  - BERT to match **text** (page) with **paragraph** (section)
    - Consider **continuity**

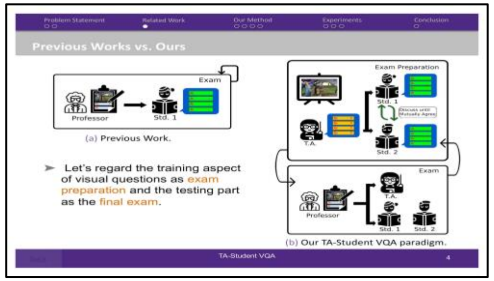


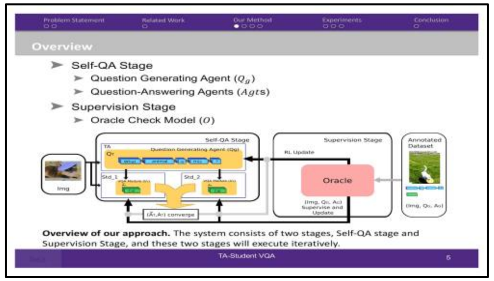
Dataset Building

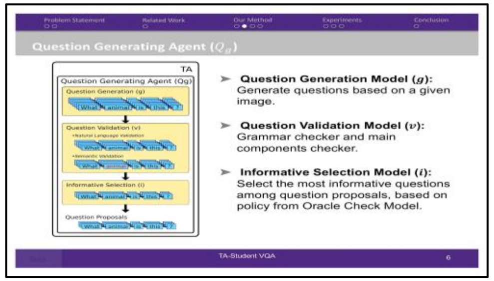
- Generate pages **for each section** and combine them all
  - BERT to match **text** (page) with **paragraph** (section)



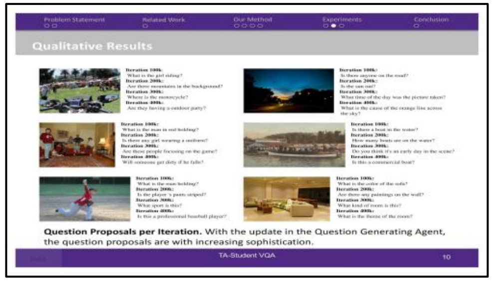
1. Introduction

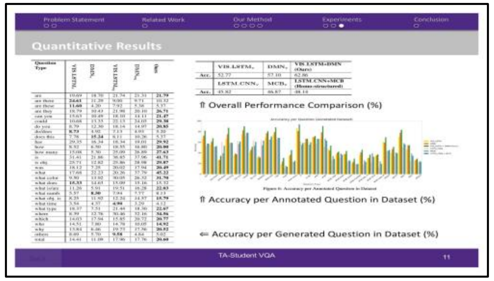






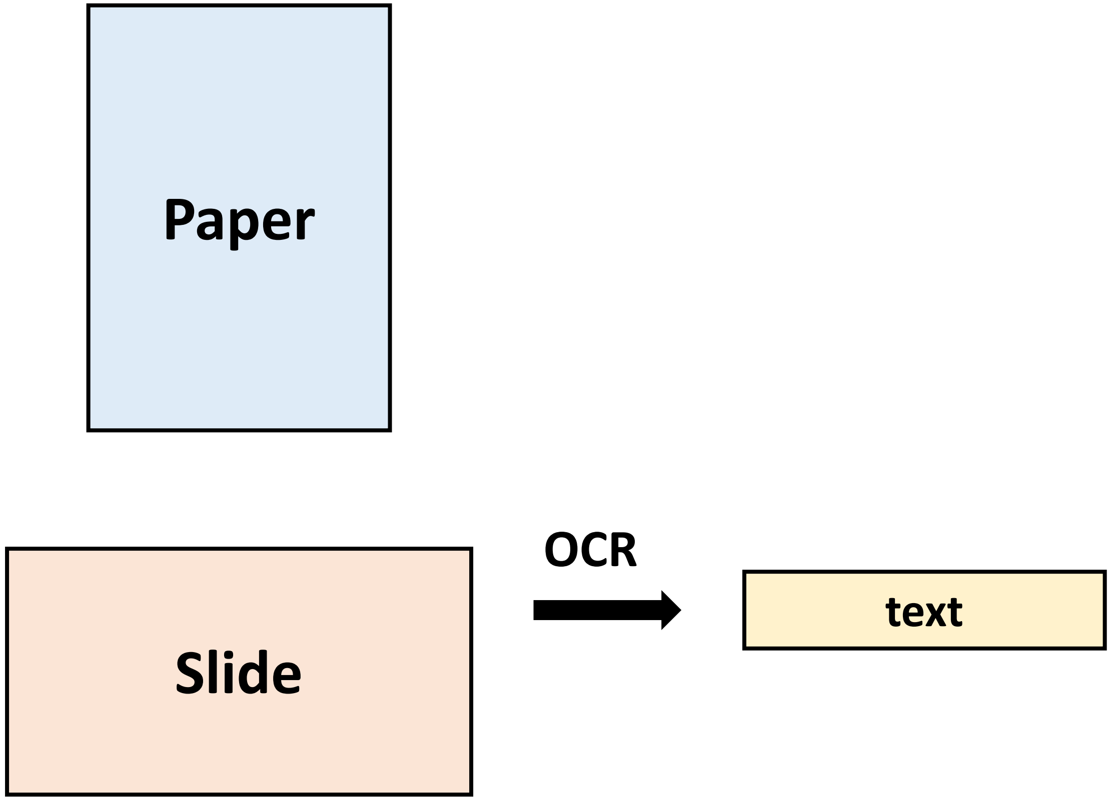
3. Approach



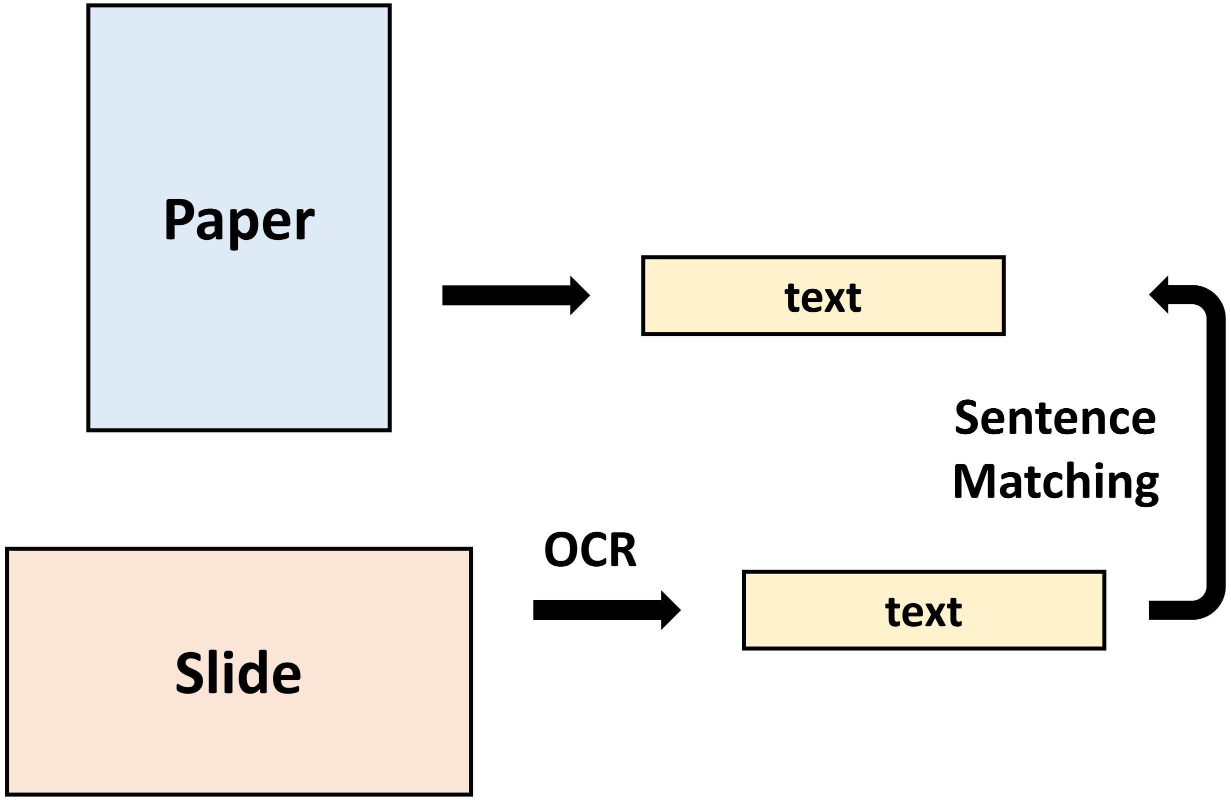


4. Experiments

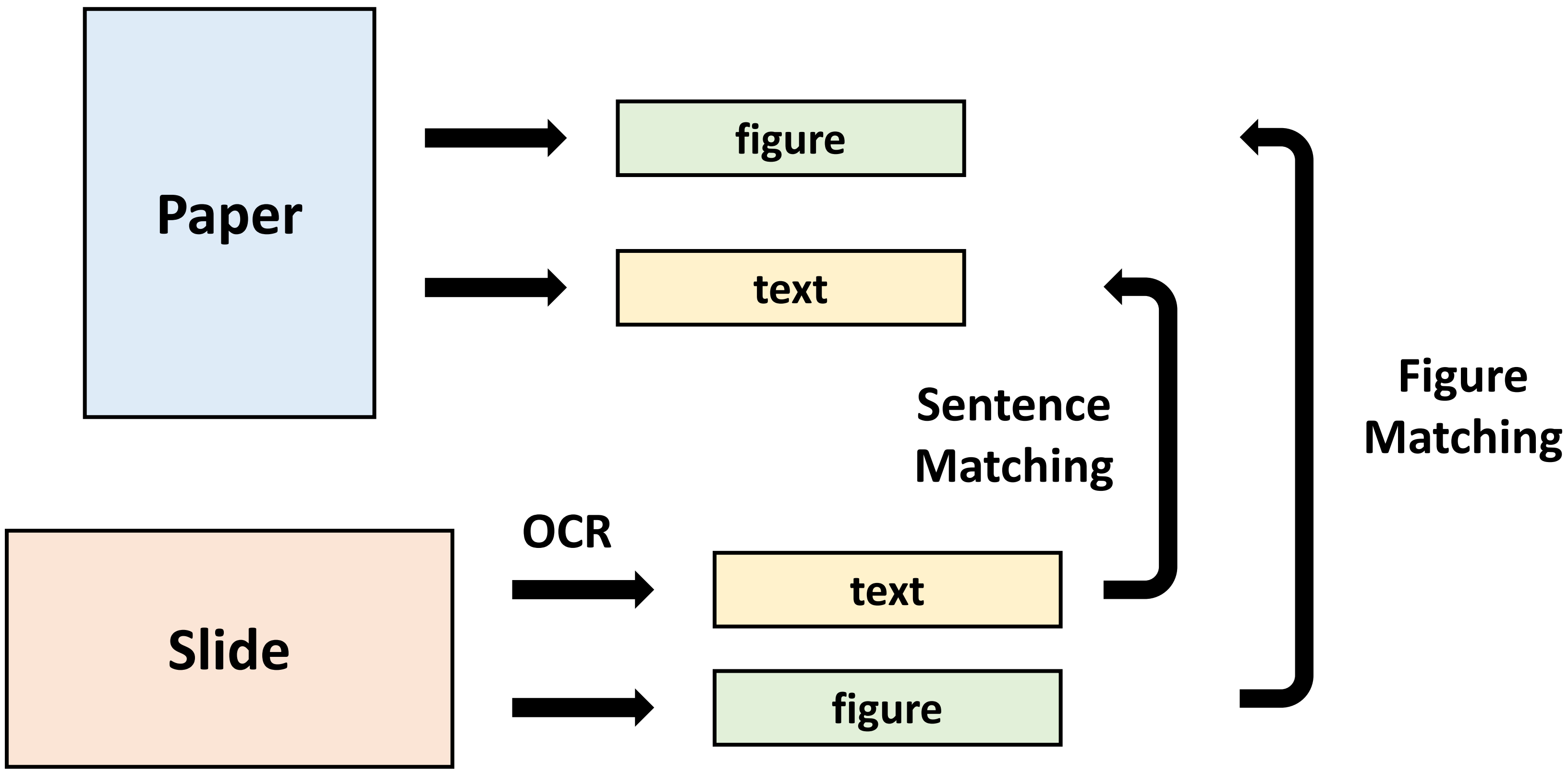
Dataset Building



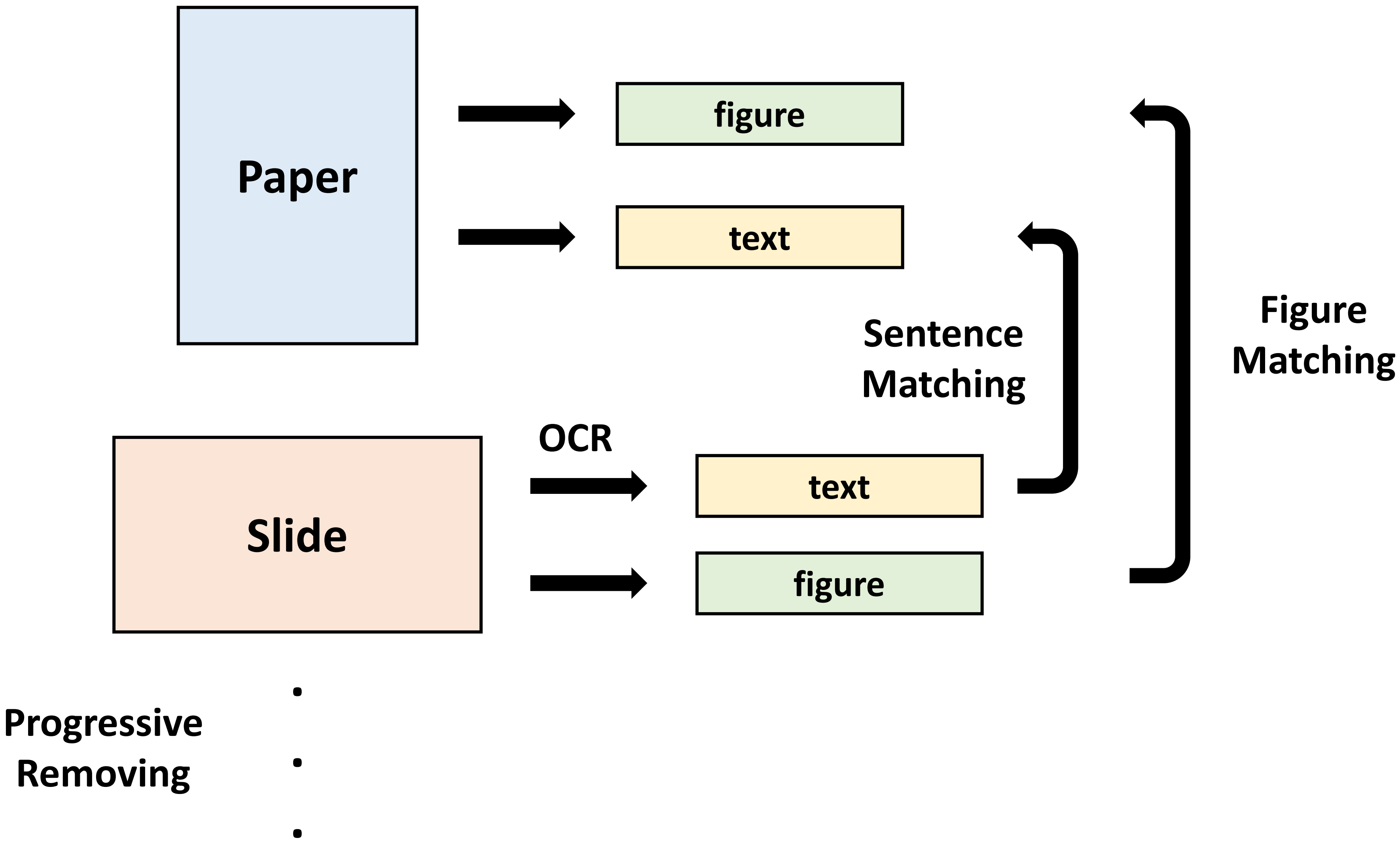
Dataset Building



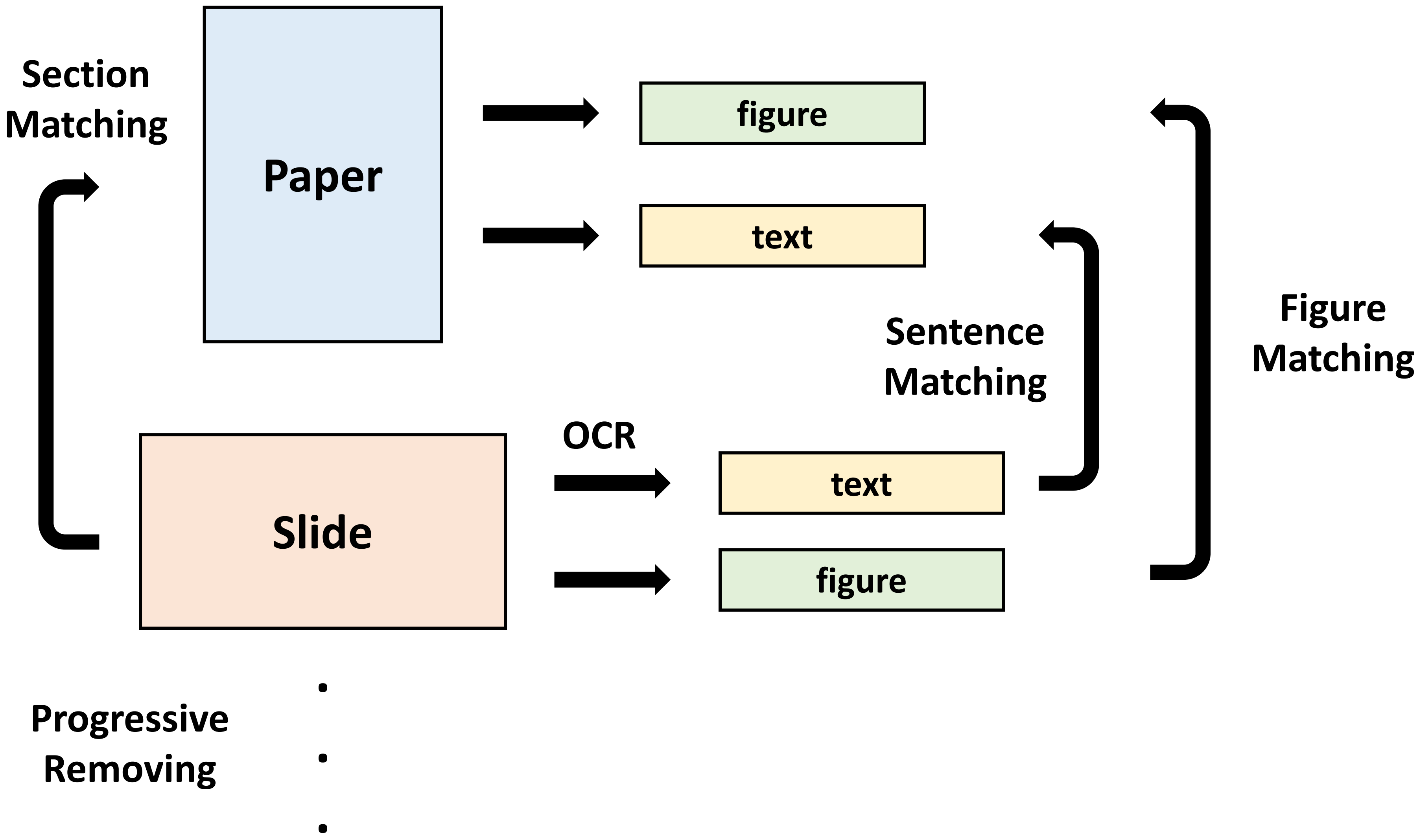
Dataset Building



Dataset Building



Dataset Building

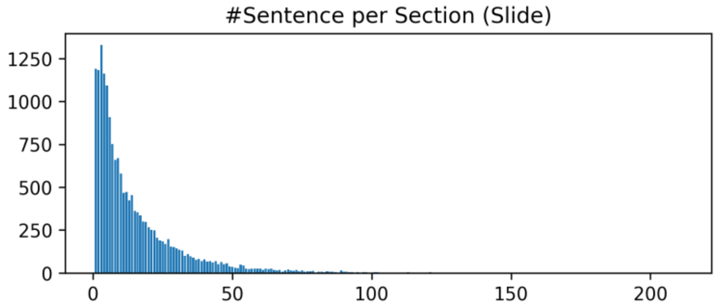


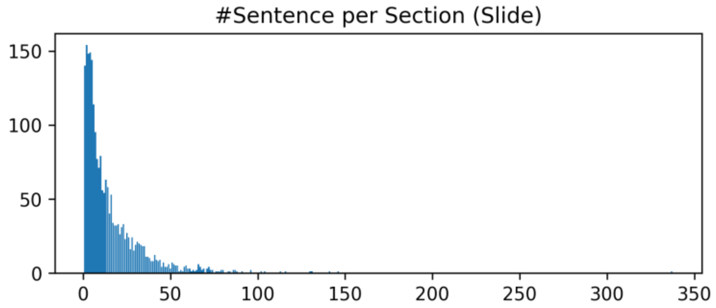
Dataset Building

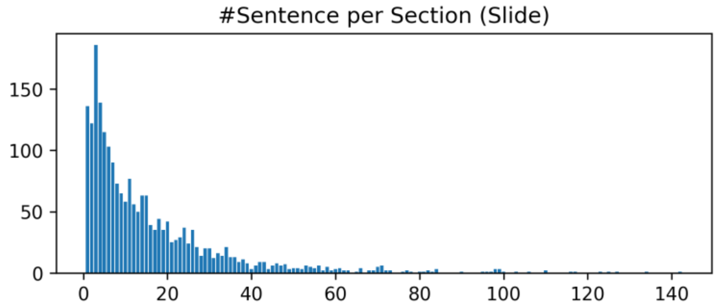
	Paper				Slide		
	num	#section	#sentence (per section)	#figure	#page	#sentence (per section)	#figure
Train	4,686	6.9	42.9	8.3	16.9	8.1	2.4
Val	592	6.9	42.6	8.3	16.8	8.1	2.5
Test	595	6.9	42.4	8.4	16.5	8.1	2.6
Test (Human)	-						2.3

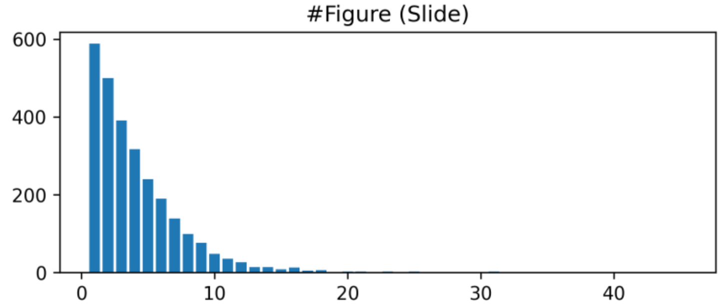
Dataset Building

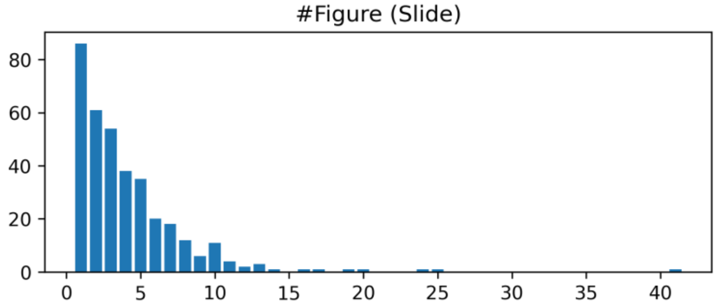
- Distribution of **#sentence** and **#figure** in slide
  - Similar between train, val, and test

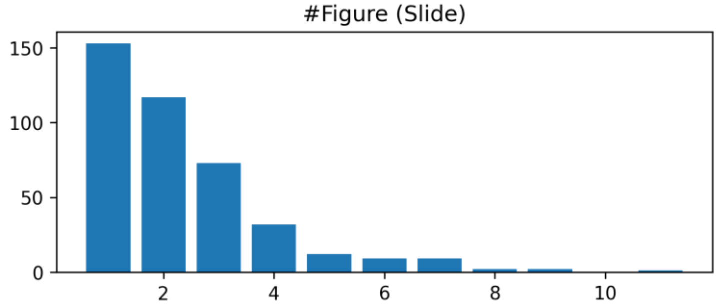








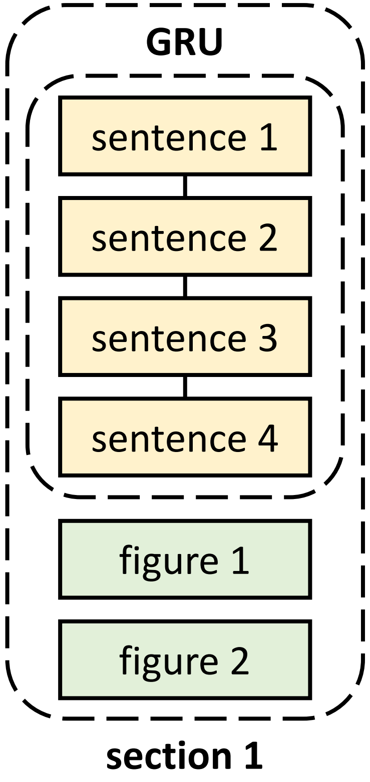




Train **Val** **Test (Human)**

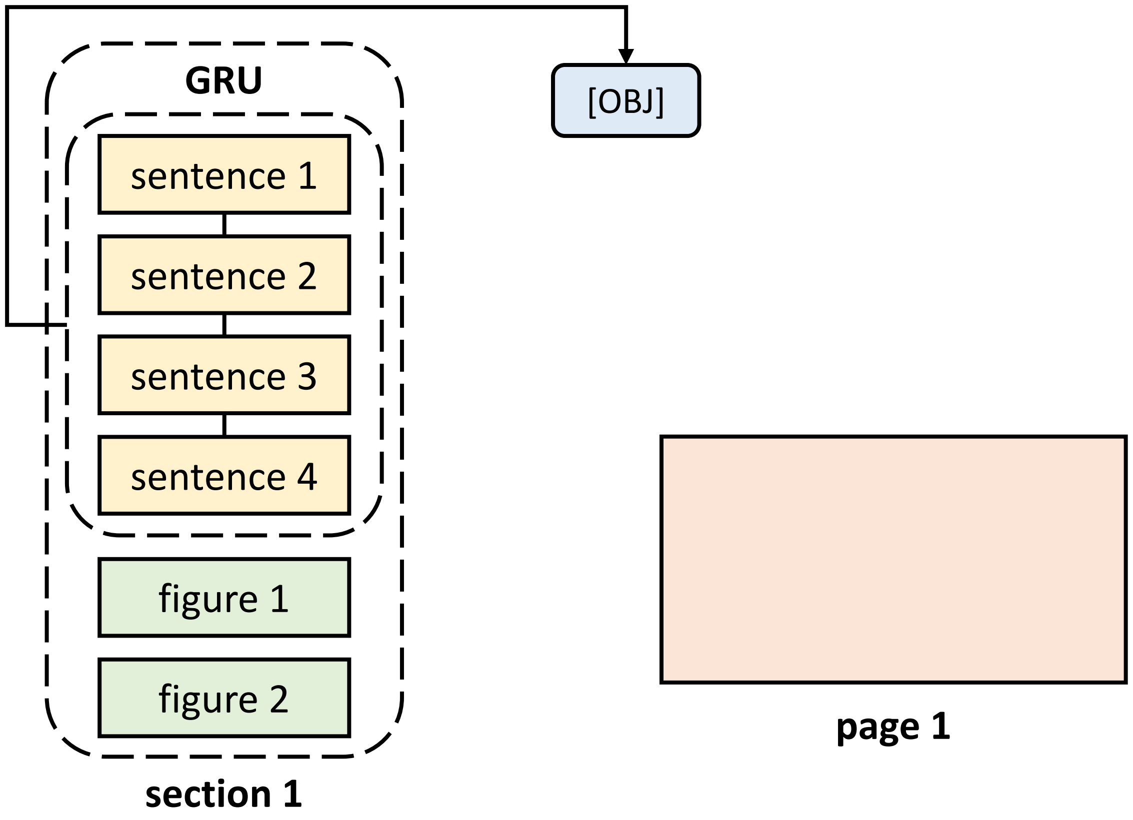
Model (Baseline)

- Recurrent extractor to build the slide step-by-step
  - [OBJ], [PAGE], [SECTION] token
  - **Section-based** generation and **classification** for extraction



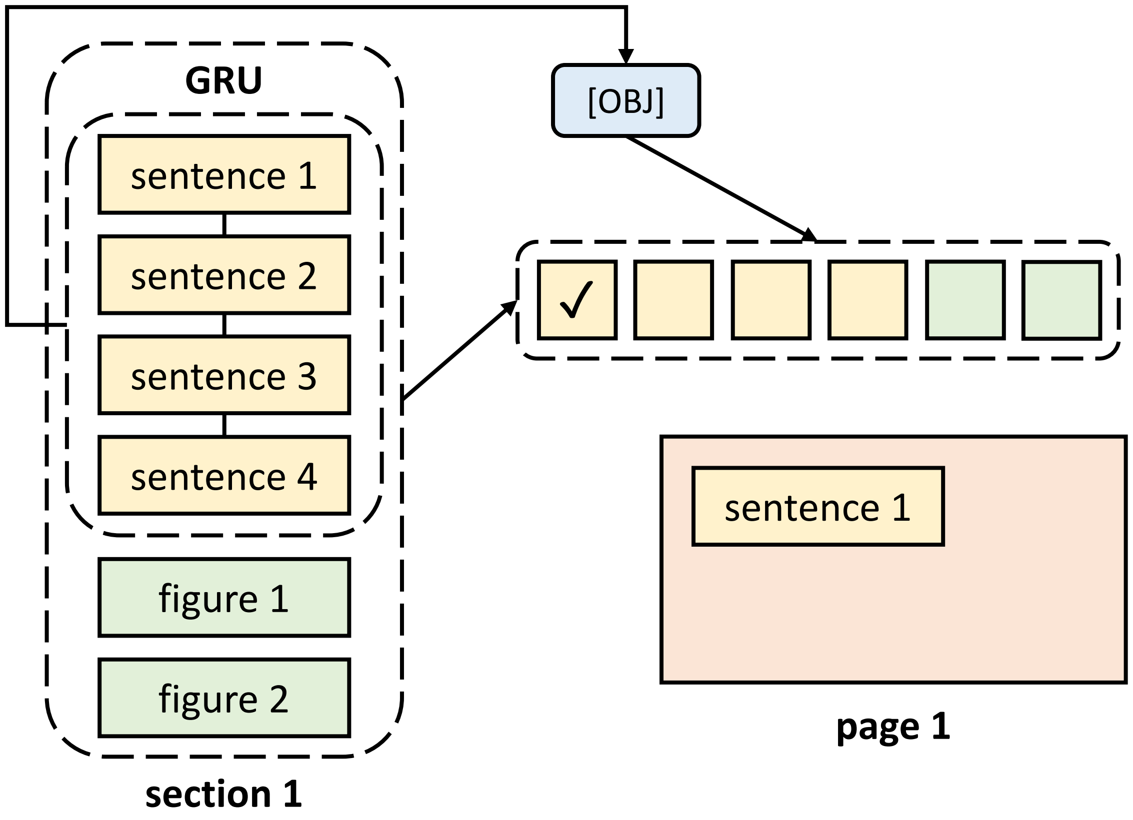
Model (Baseline)

- Recurrent extractor to build the slide step-by-step
  - [OBJ], [PAGE], [SECTION] token
  - **Section-based** generation and **classification** for extraction



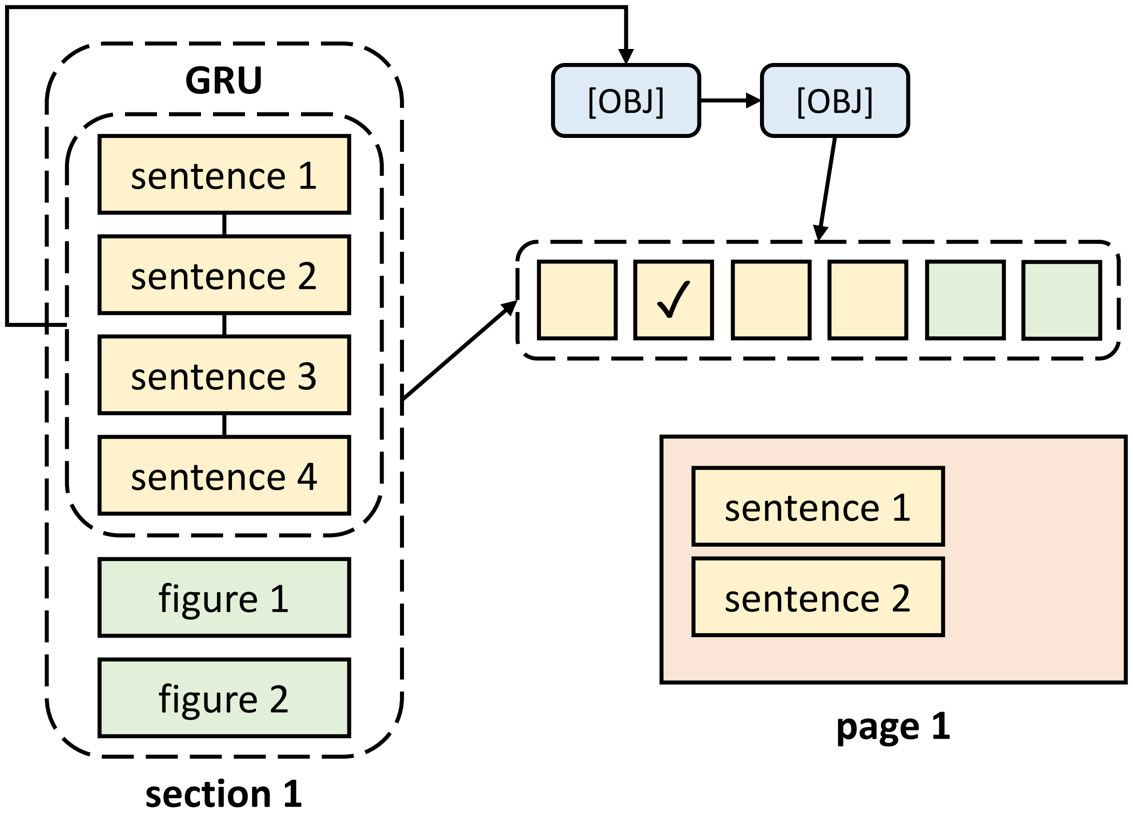
Model (Baseline)

- Recurrent extractor to build the slide step-by-step
  - [OBJ], [PAGE], [SECTION] token
  - **Section-based** generation and **classification** for extraction



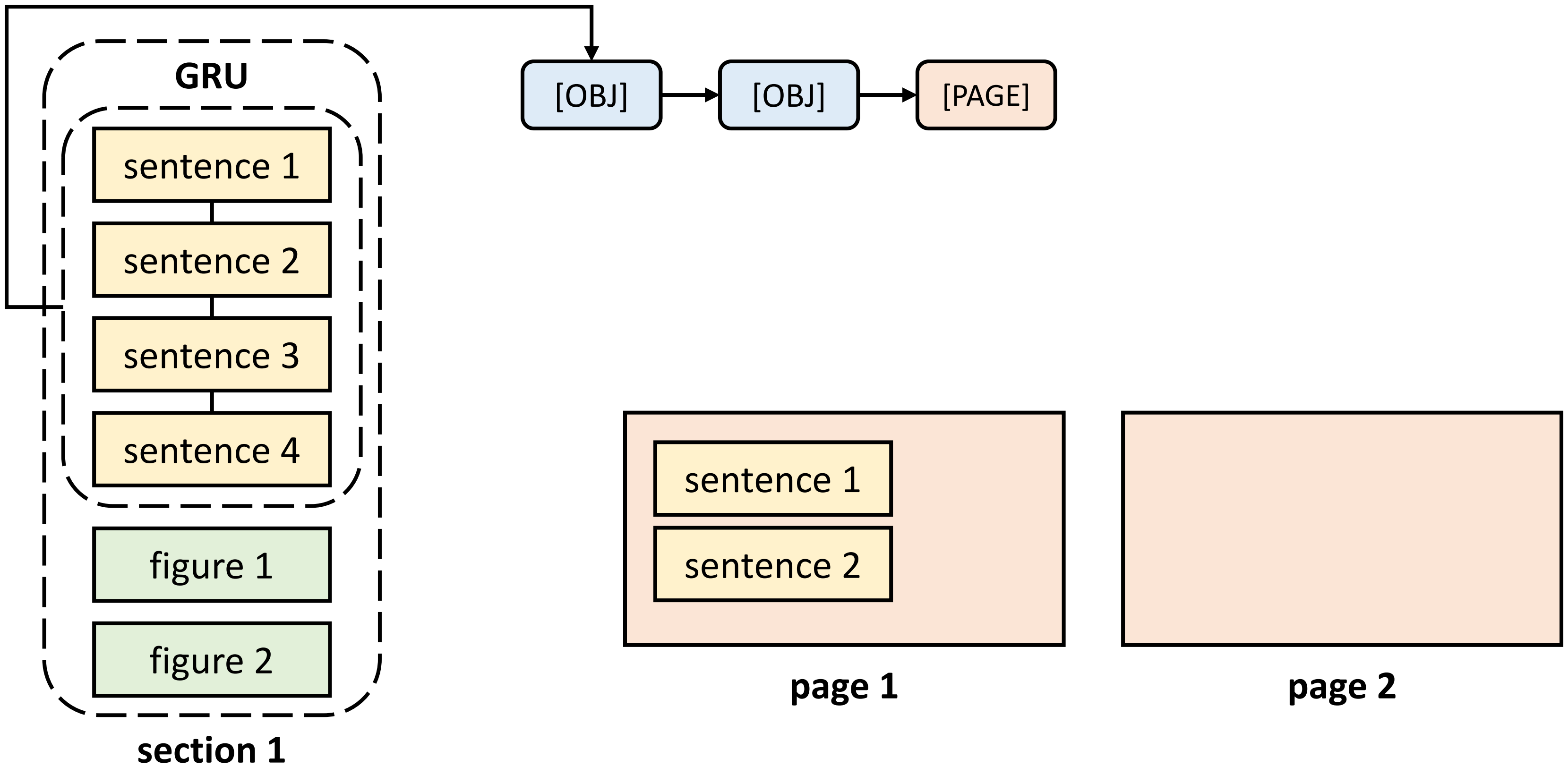
Model (Baseline)

- Recurrent extractor to build the slide step-by-step
  - [OBJ], [PAGE], [SECTION] token
  - **Section-based** generation and **classification** for extraction



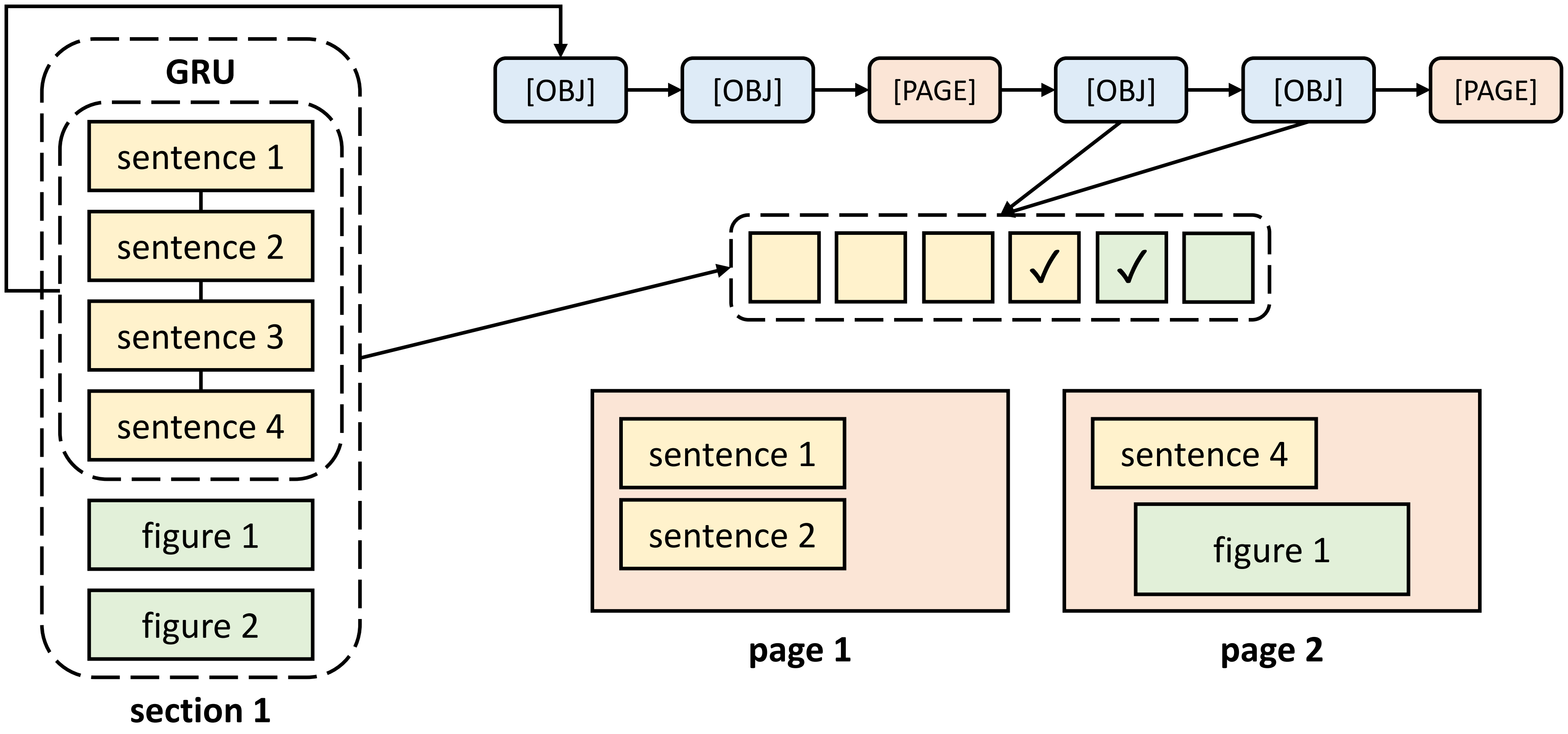
Model (Baseline)

- Recurrent extractor to build the slide step-by-step
  - [OBJ], [PAGE], [SECTION] token
  - **Section-based** generation and **classification** for extraction



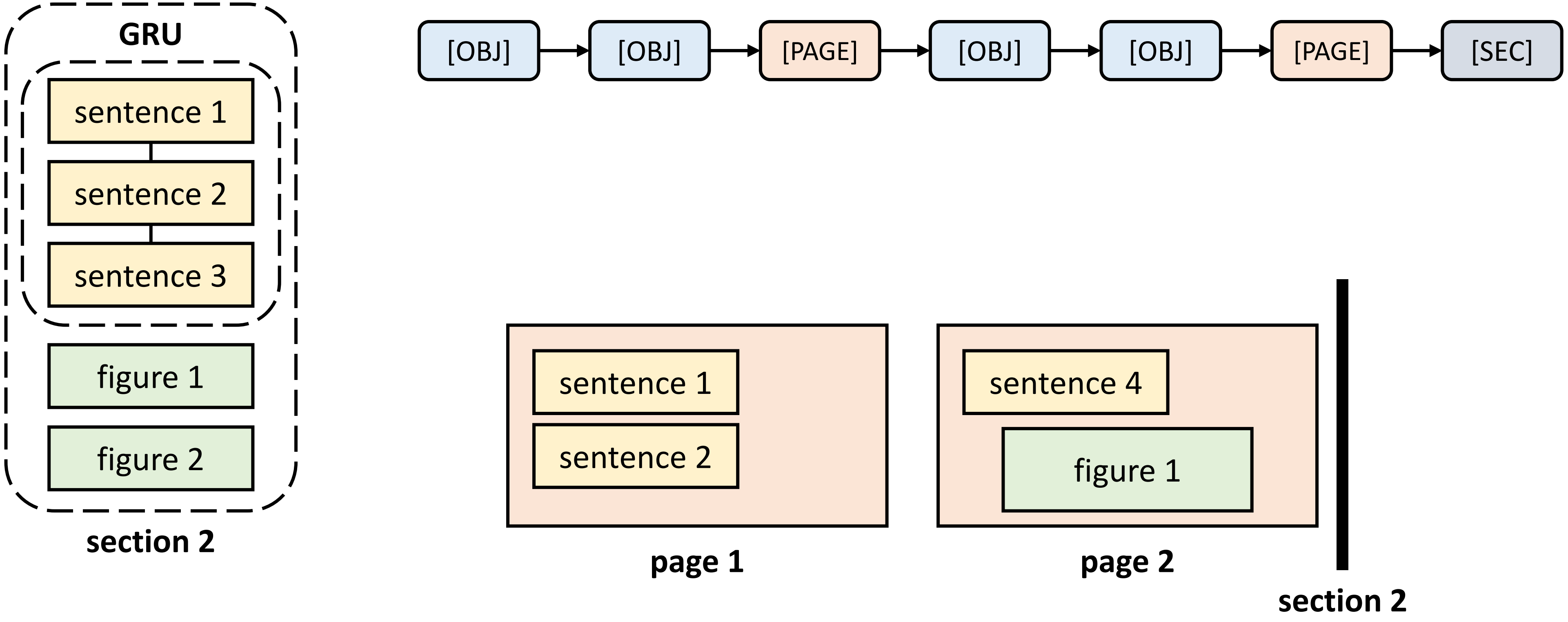
Model (Baseline)

- Recurrent extractor to build the slide step-by-step
  - [OBJ], [PAGE], [SECTION] token
  - **Section-based** generation and **classification** for extraction



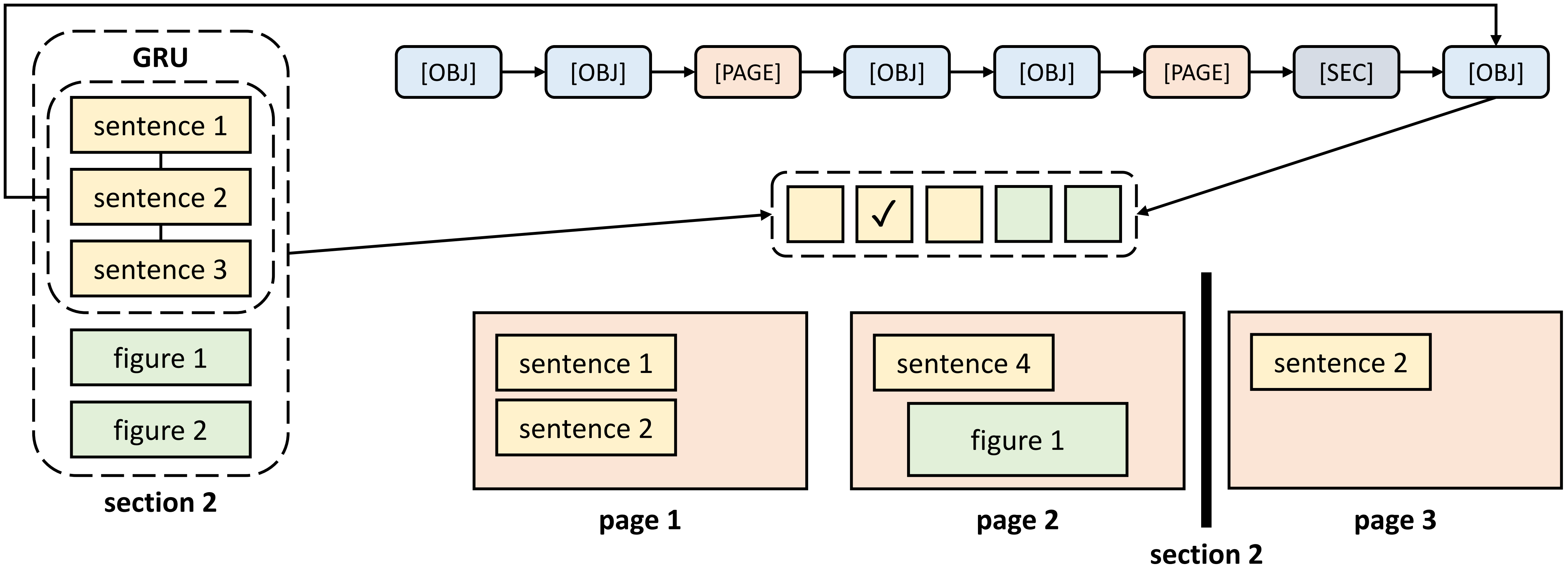
Model (Baseline)

- Recurrent extractor to build the slide step-by-step
  - [OBJ], [PAGE], [SECTION] token
  - **Section-based** generation and **classification** for extraction



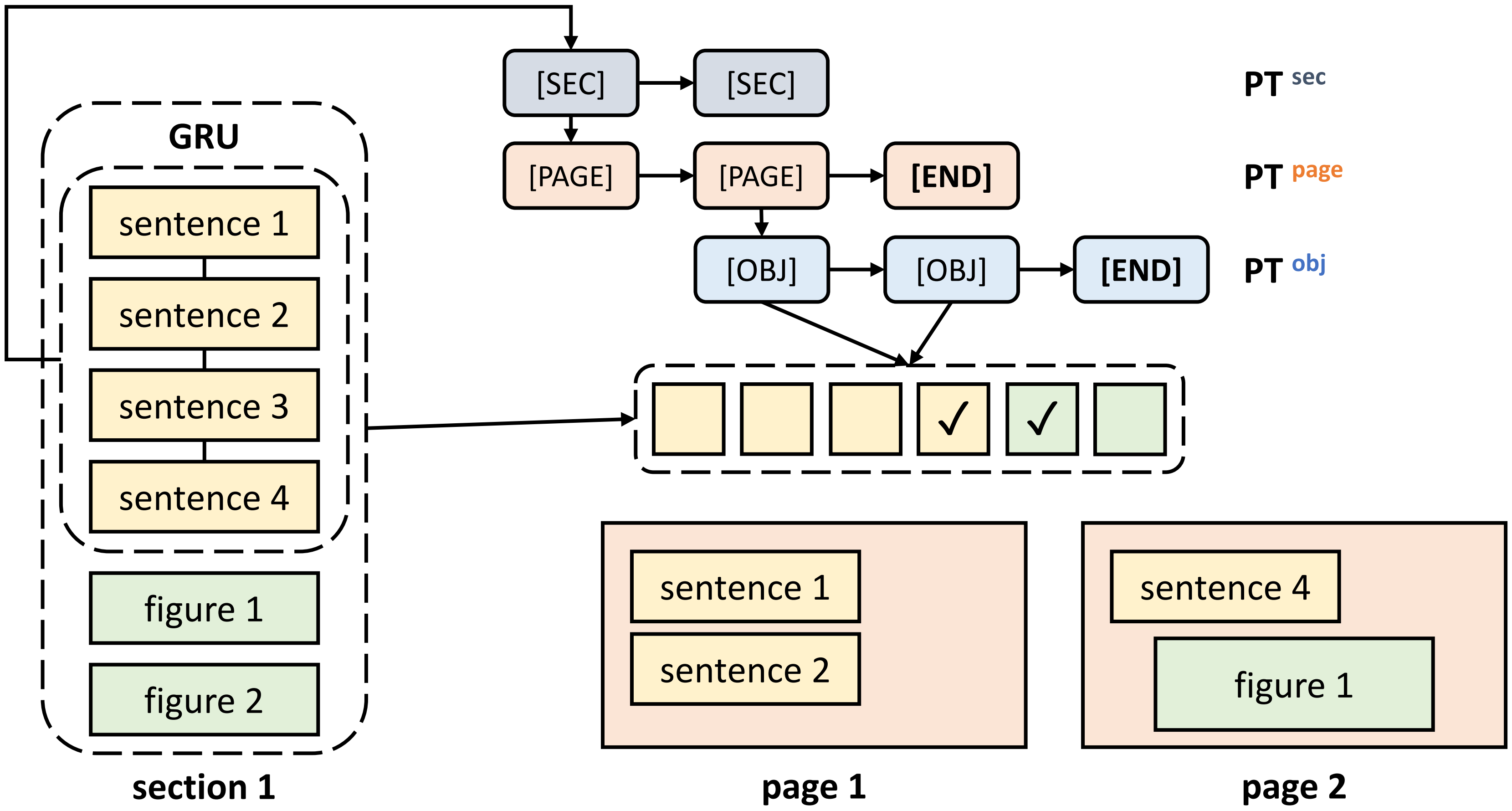
Model (Baseline)

- Recurrent extractor to build the slide step-by-step
  - [OBJ], [PAGE], [SECTION] token
  - **Section-based** generation and **classification** for extraction



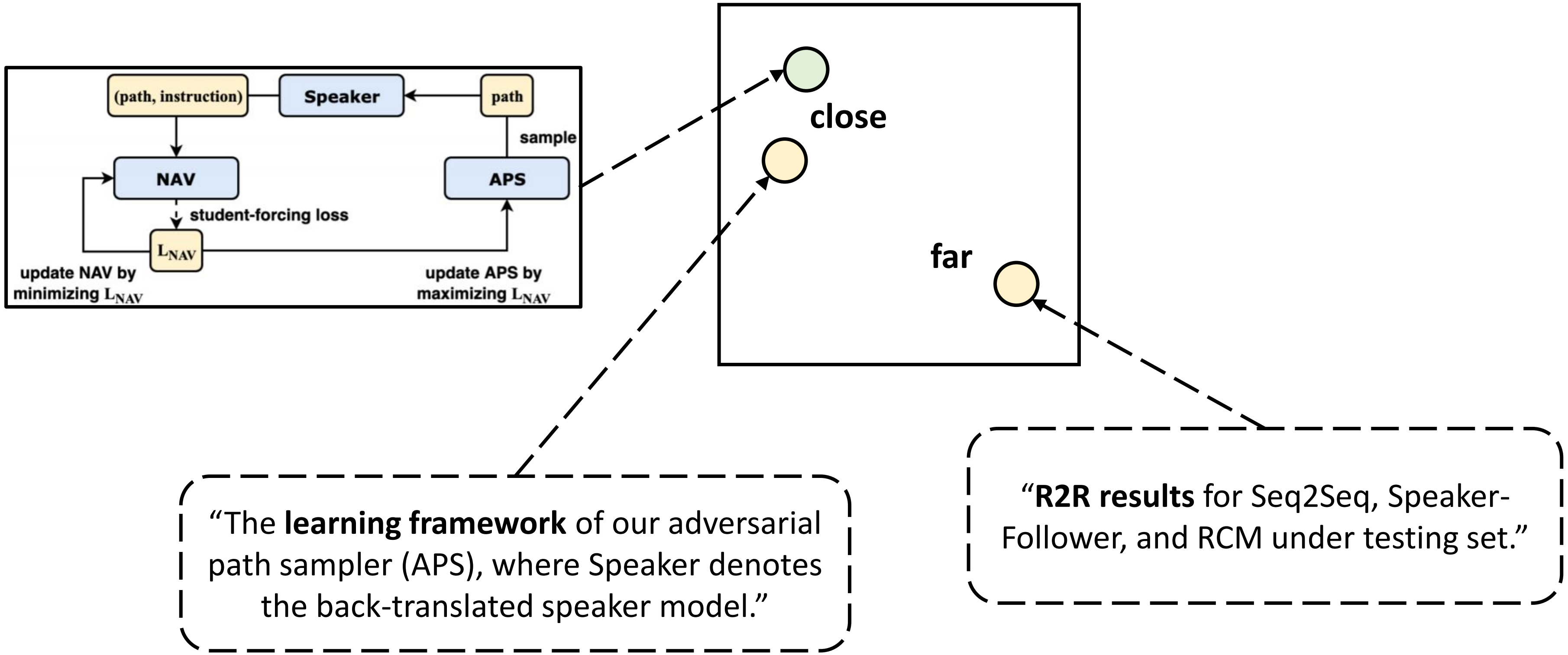
Model (HSE)

- **H**ierarchical **S**lide **E**xtractor (HSE)
  - Different RNNs for **section-**, **page-**, and **object- level**



TextFigure Module

- Constrain the **coherence** between figure-text
  - Co-train with HSE
  - Related figure-text should be **close on embedding space**

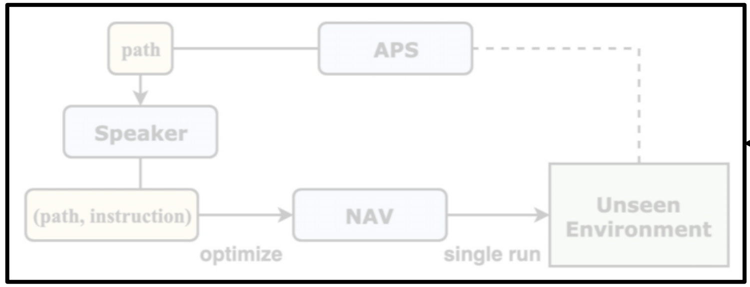


TextFigure Module

- Right figures put with right texts
  - **Filter out unrelated** and **add unused related** figures

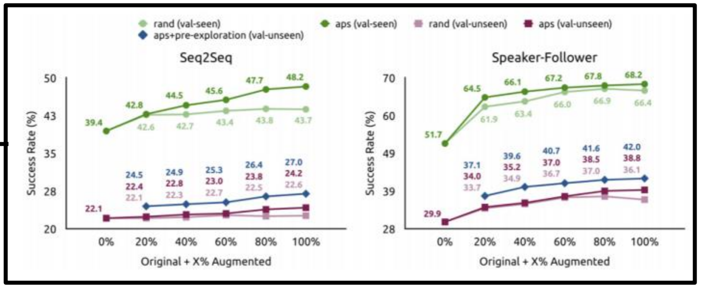
Result

- Randomly sampled stop improving when using more than 60%
- APS sampled helps both seen and unseen
- Pre-Exploration further helps unseen environments



filter out (unrelated)

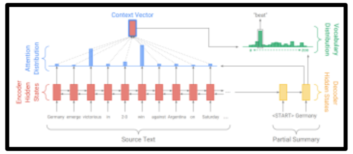
add unused (related)



Paraphrasing Module

- Rewrite extracted sentences as **slide-style**
  - Seq2seq model (w/ copy attention)

"to understand the spread of individual judgements on a sentence , we compute the standard deviation of **ratings for each sentence** and then **take the mean** over all sentences ."



paraphrase

"we collect multiple ratings for a sentence and take the mean ."

"we perform **empirical evaluation** and analysis of a variety of **classification methods** for the above task ."

"empirical evaluation of classification methods"

HSE w/ TextFigure & Paraphrasing

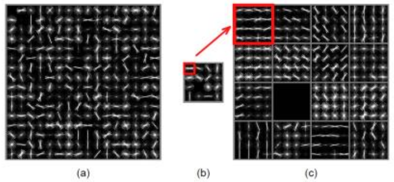


Fig. 3. a: The method proposed by Sadeghi and Forsyth [2] quantizes each cell into one of 256 pre-defined clusters. Nearest neighbour search is a significant bottleneck in their technique. In this paper we use hierarchical clustering instead of flat clustering. **b**: each cell is first quantized into one of the 16 clusters. **c**: Depending on the first level, the cell is clustered into one of 16 clusters in the respective group in c. Note that hierarchical clustering reduces the number of comparisons from 256 per cell to two stages of 16 comparisons per cell.

3 Hierarchical Vector Quantization

Several optimization techniques have been employed to speed up Deformable Parts Model object detectors. The fastest was proposed by Sadeghi and Forsyth [2]. This is nearly two orders of magnitude faster than the original implementation of [21]. The key to their success is a vector quantization technique that decreases the computation demand by a large factor. They vector quantize HOG features and compute template scores by indexing certain look-up tables and adding their scores.

We use vector quantization for the same purpose but with a slightly different approach. The main computation bottleneck in [2] is vector quantization. They need 70ms per image to quantize HOG features for one image. The high computational demand is due to the fact that each HOG cell needs to be compared against every one of 256 cluster centers. (Figure 3, a). We use a hierarchical clustering technique to speed up this process. We first cluster each cell into 16 clusters (Figure 3, b). Then according to the nearest cluster in the first step we compare against 16 other clusters to find the nearest cluster (Figure 3, c). We pre-compute clusters using k-means algorithm.

Our experiments show that the proposed hierarchical clustering technique leads to a negligible loss of 0.001 in mAP. In contrast, the speed-up gain is about 8-fold.

HSE w/ TextFigure & Paraphrasing

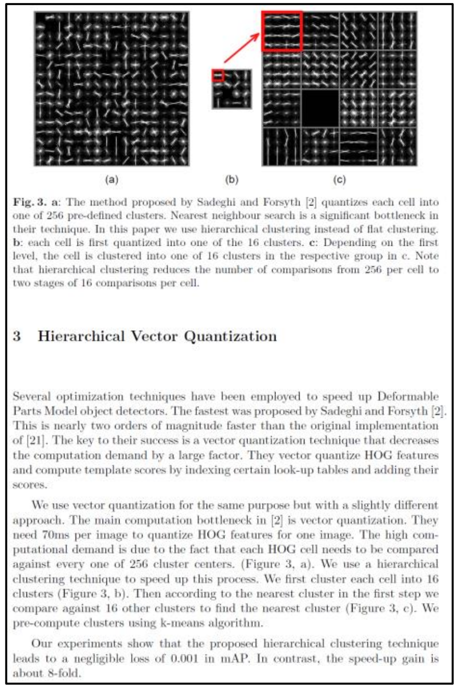


Fig. 3. a: The method proposed by Sadeghi and Forsyth [2] quantizes each cell into one of 256 pre-defined clusters. Nearest neighbour search is a significant bottleneck in their technique. In this paper we use hierarchical clustering instead of flat clustering. **b**: each cell is first quantized into one of the 16 clusters. **c**: Depending on the first level, the cell is clustered into one of 16 clusters in the respective group in c. Note that hierarchical clustering reduces the number of comparisons from 256 per cell to two stages of 16 comparisons per cell.

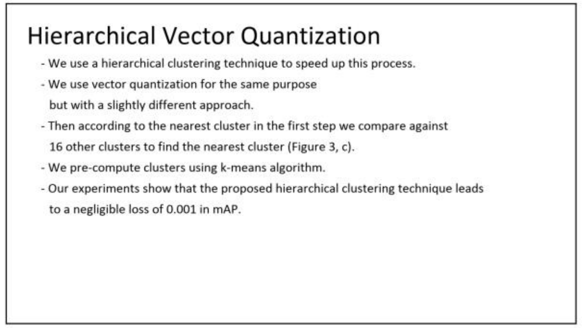
3 Hierarchical Vector Quantization

Several optimization techniques have been employed to speed up Deformable Parts Model object detectors. The fastest was proposed by Sadeghi and Forsyth [2]. This is nearly two orders of magnitude faster than the original implementation of [21]. The key to their success is a vector quantization technique that decreases the computation demand by a large factor. They vector quantize HOG features and compute template scores by indexing certain look-up tables and adding their scores.

We use vector quantization for the same purpose but with a slightly different approach. The main computation bottleneck in [2] is vector quantization. They need 70ms per image to quantize HOG features for one image. The high computational demand is due to the fact that each HOG cell needs to be compared against every one of 256 cluster centers. (Figure 3, a). We use a hierarchical clustering technique to speed up this process. We first cluster each cell into 16 clusters (Figure 3, b). Then according to the nearest cluster in the first step we compare against 16 other clusters to find the nearest cluster (Figure 3, c). We pre-compute clusters using k-means algorithm.

Our experiments show that the proposed hierarchical clustering technique leads to a negligible loss of 0.001 in mAP. In contrast, the speed-up gain is about 8-fold.

HSE



HSE w/ TextFigure & Paraphrasing

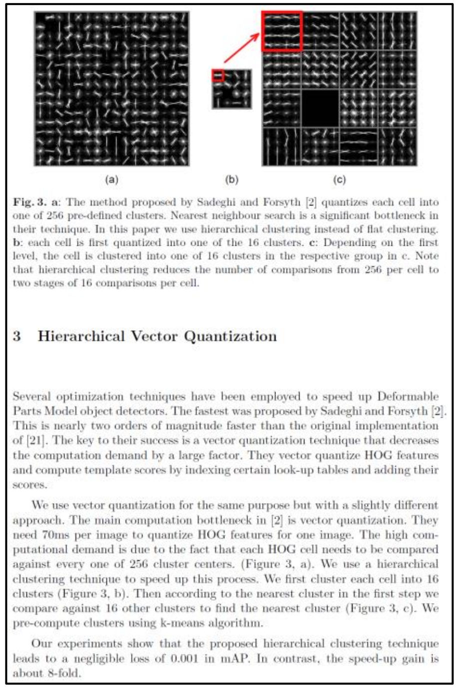

(a) (b) (c)

Fig. 3. **a**: The method proposed by Sadeghi and Forsyth [2] quantizes each cell into one of 256 pre-defined clusters. Nearest neighbour search is a significant bottleneck in their technique. In this paper we use hierarchical clustering instead of flat clustering. **b**: each cell is first quantized into one of the 16 clusters. **c**: Depending on the first level, the cell is clustered into one of 16 clusters in the respective group in c. Note that hierarchical clustering reduces the number of comparisons from 256 per cell to two stages of 16 comparisons per cell.

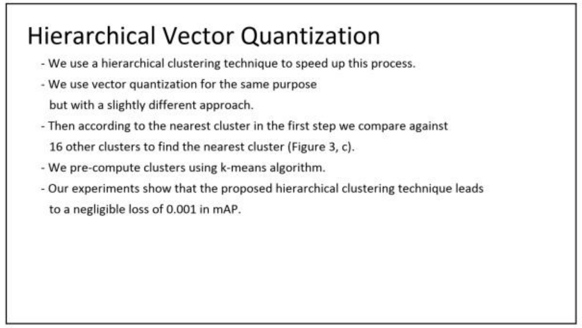
3 Hierarchical Vector Quantization

Several optimization techniques have been employed to speed up Deformable Parts Model object detectors. The fastest was proposed by Sadeghi and Forsyth [2]. This is nearly two orders of magnitude faster than the original implementation of [21]. The key to their success is a vector quantization technique that decreases the computation demand by a large factor. They vector quantize HOG features and compute template scores by indexing certain look-up tables and adding their scores.

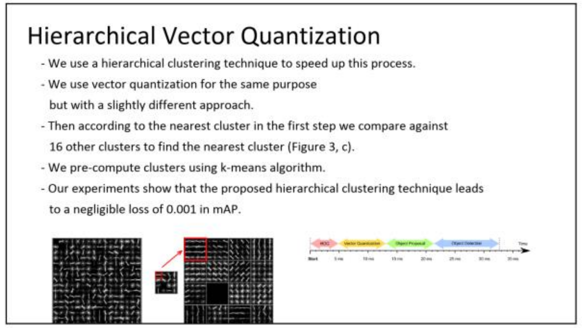
We use vector quantization for the same purpose but with a slightly different approach. The main computation bottleneck in [2] is vector quantization. They need 70ms per image to quantize HOG features for one image. The high computational demand is due to the fact that each HOG cell needs to be compared against every one of 256 cluster centers. (Figure 3, a). We use a hierarchical clustering technique to speed up this process. We first cluster each cell into 16 clusters (Figure 3, b). Then according to the nearest cluster in the first step we compare against 16 other clusters to find the nearest cluster (Figure 3, c). We pre-compute clusters using k-means algorithm.

Our experiments show that the proposed hierarchical clustering technique leads to a negligible loss of 0.001 in mAP. In contrast, the speed-up gain is about 8-fold.

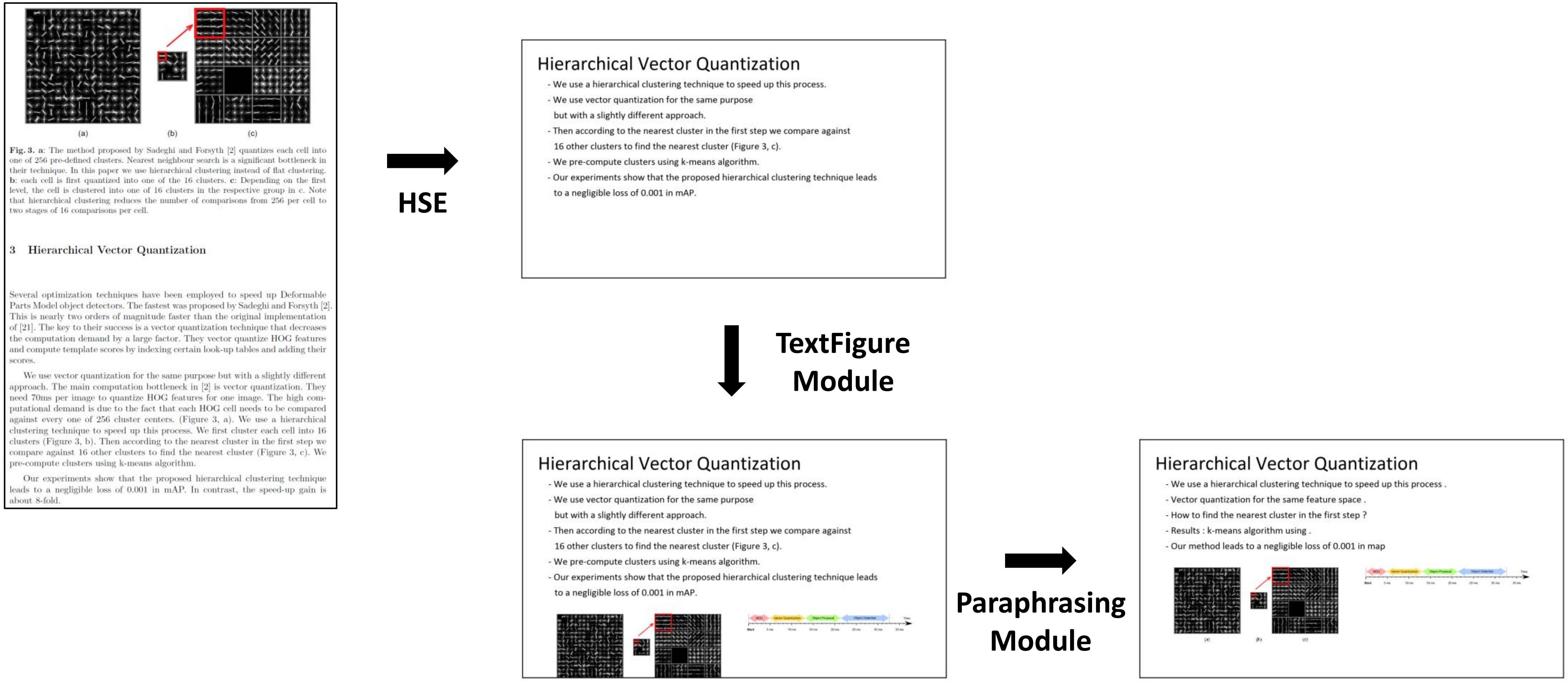
HSE



TextFigure Module



HSE w/ TextFigure & Paraphrasing



Experiments

- Evaluation metrics

Text

the cat is sleeping **on bed**

the brown **cat is** sitting **on bed**

Rouge-L: 83.3 / 71.4 / **76.9**

Experiments

- Evaluation metrics

Text

the cat is sleeping **on bed**

the brown **cat is** sitting **on bed**

Rouge-L: 83.3 / 71.4 / **76.9**

$$\mathbf{Rouge} \times \; \boldsymbol{e}^{-\frac{|\mathbf{P}-\mathbf{Q}|}{\mathbf{Q}}}$$

- consider **Page** Difference
- **P:** $\#\text{Page}_{pd}$
- **Q:** $\#\text{Page}_{gd}$

Experiments

- Evaluation metrics

Text

the cat is sleeping **on bed**

the brown **cat is** sitting **on bed**

Rouge-L: 83.3 / 71.4 / **76.9**

$$\textbf{Rouge} \times e^{-\frac{|\mathbf{P}-\mathbf{Q}|}{\mathbf{Q}}}$$

- consider **Page** difference
- **P:** $\#\text{Page}_{pd}$
- **Q:** $\#\text{Page}_{gd}$

Figure

A / D / C / **F** / **E**

A / **F** / B / **E**

LC-P/R/F: 60.0 / 75.0 / **66.7**

Experiments

- Evaluation metrics

Text

the cat is sleeping **on bed**

the brown **cat is** sitting **on bed**

Rouge-L: 83.3 / 71.4 / **76.9**

$$\textbf{Rouge} \times e^{-\frac{|P-Q|}{Q}}$$

- consider **Page** Difference
- **P:** $\#Page_{pd}$
- **Q:** $\#Page_{gd}$

Figure

A / D / C / **F** / **E**

A / **F** / B / **E**

LC-P/R/F: 60.0 / 75.0 / **66.7**

TextFigure

the cat is sleeping **on bed** **A**

the brown **cat is** sitting **on bed** **A**

Rouge-L

the **fast fox** **jumped** over **B**

fast brown **fox** **jumped** up **B**

Experiments

1st / 2nd

Model	Co-Train	w/ Module		Text		Figure			TextFigure
	TextFigure	Paraphrase	TextFigure	Rouge-L	w/ Page	LC-P	LC-R	LC-F1	Rouge-L
Baseline	✗	✗	✗	27.2	21.8	13.2	21.9	16.5	3.6
HSE	✗	✗	✗	27.7	22.9	14.6	23.7	18.1	4.3
	✗	✓	✗	**32.3**	**26.7**	14.6	23.7	18.1	4.7
	✓	✗	✗	28.7	24.0	**14.8**	**32.4**	**20.3**	7.9
	✓	✗	✓	28.7	24.0	**24.6**	**40.5**	**30.6**	**13.8**
	✓	✓	✗	**33.6**	**28.2**	**14.8**	**32.4**	**20.3**	8.2
	✓	✓	✓	**33.6**	**28.2**	**24.6**	**40.5**	**30.6**	**15.5**

Experiments

Model	Co-Train	w/ Module		Text		Figure			TextFigure
	TextFigure	Paraphrase	TextFigure	Rouge-L	w/ Page	LC-P	LC-R	LC-F1	Rouge-L
Baseline	✗	✗	✗	27.2	21.8	13.2	21.9	16.5	3.6
HSE	✗	✗	✗	**27.7**	**22.9**	**14.6**	**23.7**	**18.1**	**4.3**
	✗	✓	✗	32.3	26.7	14.6	23.7	18.1	4.7
	✓	✗	✗	28.7	24.0	14.8	32.4	20.3	7.9
	✓	✗	✓	28.7	24.0	24.6	40.5	30.6	13.8
	✓	✓	✗	33.6	28.2	14.8	32.4	20.3	8.2
	✓	✓	✓	33.6	28.2	24.6	40.5	30.6	15.5

- **Hierarchical architecture** extracts slide
  - Helps both **text quality** and **figure retrieval**

Experiments

Model	Co-Train	w/ Module		Text		Figure			TextFigure
	TextFigure	Paraphrase	TextFigure	Rouge-L	w/ Page	LC-P	LC-R	LC-F1	Rouge-L
Baseline	✗	✗	✗	27.2	21.8	13.2	21.9	16.5	3.6
HSE	✗	✗	✗	27.7	22.9	14.6	23.7	18.1	4.3
	✗	✓	✗	**32.3**	**26.7**	14.6	23.7	18.1	**4.7**
	✓	✗	✗	28.7	24.0	14.8	32.4	20.3	7.9
	✓	✗	✓	28.7	24.0	24.6	40.5	30.6	13.8
	✓	✓	✗	**33.6**	**28.2**	14.8	32.4	20.3	**8.2**
	✓	✓	✓	33.6	28.2	24.6	40.5	30.6	15.5

- **Paraphrasing module** rewrites sentences into slide-style
  - Better **text** as a slide

Experiments

Model	Co-Train	w/ Module		Text		Figure			TextFigure
	TextFigure	Paraphrase	TextFigure	Rouge-L	w/ Page	LC-P	LC-R	LC-F1	Rouge-L
Baseline	✗	✗	✗	27.2	21.8	13.2	21.9	16.5	3.6
HSE	✗	✗	✗	27.7	22.9	14.6	23.7	18.1	4.3
	✗	✓	✗	32.3	26.7	14.6	23.7	18.1	4.7
	✓	✗	✗	**28.7**	**24.0**	**14.8**	**32.4**	**20.3**	**7.9**
	✓	✗	✓	28.7	24.0	24.6	40.5	30.6	13.8
	✓	✓	✗	33.6	28.2	14.8	32.4	20.3	8.2
	✓	✓	✓	33.6	28.2	24.6	40.5	30.6	15.5

- Co-train with **TextFigure constrain**
  - Learns the **correlation** between text and figure

Experiments

Model	Co-Train	w/ Module		Text		Figure			TextFigure
	TextFigure	Paraphrase	TextFigure	Rouge-L	w/ Page	LC-P	LC-R	LC-F1	Rouge-L
Baseline	✗	✗	✗	27.2	21.8	13.2	21.9	16.5	3.6
HSE	✗	✗	✗	27.7	22.9	14.6	23.7	18.1	4.3
	✗	✓	✗	32.3	26.7	14.6	23.7	18.1	4.7
	✓	✗	✗	28.7	24.0	14.8	32.4	20.3	7.9
	✓	✗	✓	28.7	24.0	**24.6**	**40.5**	**30.6**	**13.8**
	✓	✓	✗	33.6	28.2	14.8	32.4	20.3	8.2
	✓	✓	✓	33.6	28.2	24.6	40.5	30.6	15.5

- **TextFigure module** removes unrelated or adds related
  - Benefits **figure retrieval** a lot

Experiments

Model	Co-Train	w/ Module		Text		Figure			TextFigure
	TextFigure	Paraphrase	TextFigure	Rouge-L	w/ Page	LC-P	LC-R	LC-F1	Rouge-L
Baseline	✗	✗	✗	27.2	21.8	13.2	21.9	16.5	3.6
HSE	✗	✗	✗	27.7	22.9	14.6	23.7	18.1	4.3
	✗	✓	✗	32.3	26.7	14.6	23.7	18.1	4.7
	✓	✗	✗	28.7	24.0	14.8	32.4	20.3	7.9
	✓	✗	✓	28.7	24.0	24.6	40.5	30.6	13.8
	✓	✓	✗	33.6	28.2	14.8	32.4	20.3	8.2
	✓	✓	✓	**33.6**	**28.2**	**24.6**	**40.5**	**30.6**	**15.5**

- Combines both **Paraphrasing** and **TextFigure** module
  - Fully improves on **all aspects of metrics**

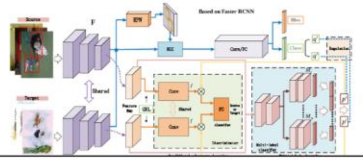
Qualitative Examples

Introduction

- We propose a novel multi-label conditional alignment methodology to bridge domain divergence while preserving the discriminability of the features .
- Mcar : multi-label conditional distribution alignment and detection regularization model
- Minimize the cross-domain feature distribution gaps .
- A whole image can have complex multimodal structures .
- Global (image-level) feature alignment (image-level)

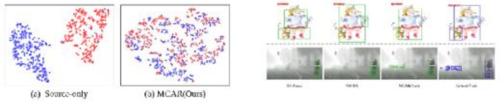
Category Prediction based Regularization

- What is the structure of the graph ?
- We propose a novel category prediction mechanism for object detection .
- Each proposal will be classified as a regressor r .
- Region proposal network (rpn) 23-28 august zÃ¸zÃ¸
- Loss function : = + (x)



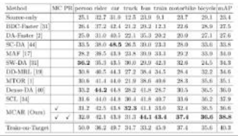
Adaptation from Clear to Foggy Scenes.

- Cross-domain detection from real to virtual image scenarios
- Domain adaption from normal / clear images to foggy image
- Pascal voc , pascal voc to comic



Ablation Study

- Qualitative results : quantitative results 23-28 august zÃ¸zÃ¸
- Adacoseg : adaptive feature visualization with mutual regularization---p . zhao et al .
- We use the foggy cityscapes dataset as the target domain .
- Train on labeled data in the target domain
- Multiple auxiliary loss terms in the proposed learning objective

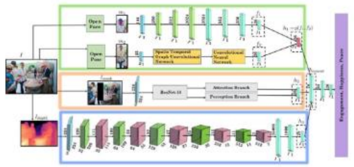


Introduction

- What is a good emotion classification task ?
- We use the context principle for emotion recognition .
- Context 1 : incorporating cues from different modalities
- Multimodal emotion recognition (cvpr 2020)
- Not asking for the meaning of a word in isolation and instead of finding the meaning in isolation .

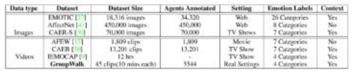
Network Architecture

- How to train your neural network ?
- To train the soft margin loss function :
- We combine the two loss functions , lmultiplicative (from eq . 1) .
- + Î»2lclassification



Datasets

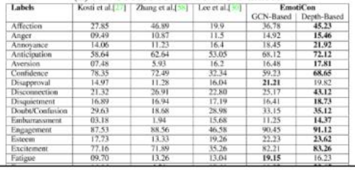
- We present a comparison with other datasets .
- The apparent emotional states of the people
- How do we evaluate the annotation process ?
- How do we evaluate the friendliness ?



Analysis and Discussion

- Emotic dataset . emotic dataset was collected for
- Two-stream network (two-stream) [2]
- Gcn (ours) depth-based (ours)
- Groupwalk dataset was difficult to test on groupwalk .

(a) AP Scores for EMOTIC Dataset.

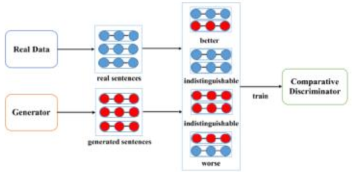


SELF-ADVERSARIAL LEARNING

- For a training set with n real samples , we have
- Sal (ours) (a) sal
- How to suffer from the reward sparsity ?
- Sal (ours) 3 (ours)

TRAINING

- The comparative discriminator can offer more informative learning signals from the comparative discriminator .
- How to enhance the generalization ability of the comparative discriminator ?
- E (pz (z) , m)

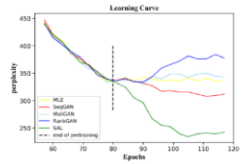


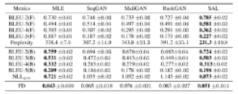
COMPARATIVE DISCRIMINATOR

- The self-improvement mechanism corresponds to the comparative discriminator .
- How to construct the model to supervise the model ?
- (goodfellow et al . , 2014)

RESULTS IN REAL DATA

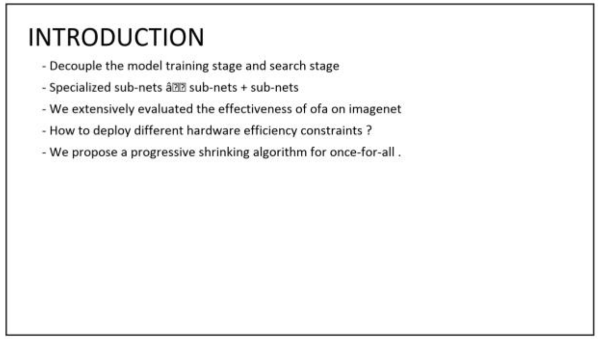
- Table 3 . the results of coco image caption .

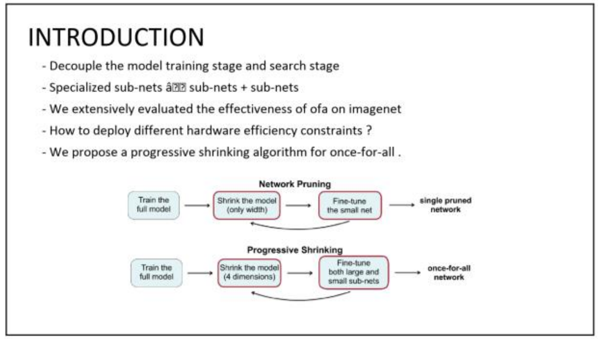


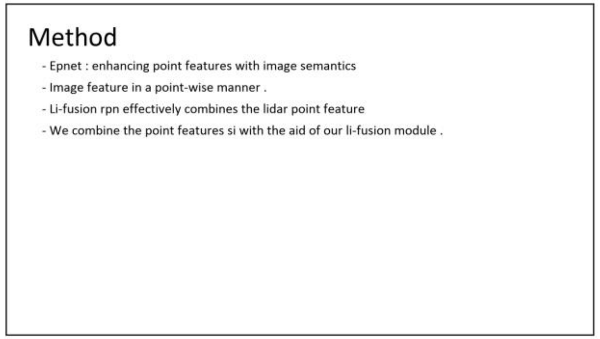


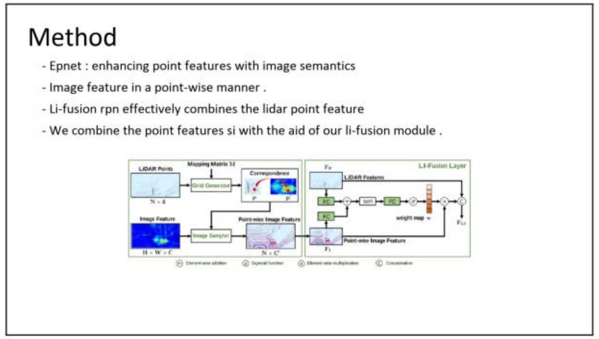
Qualitative Examples

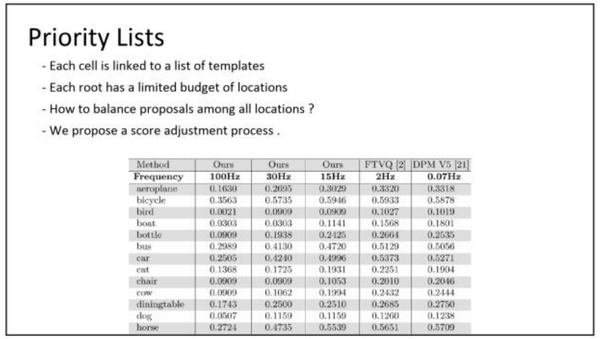
- TextFigure Module (**w/o** vs **w/**)

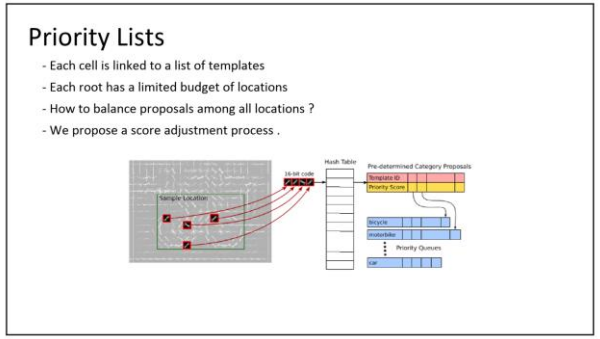


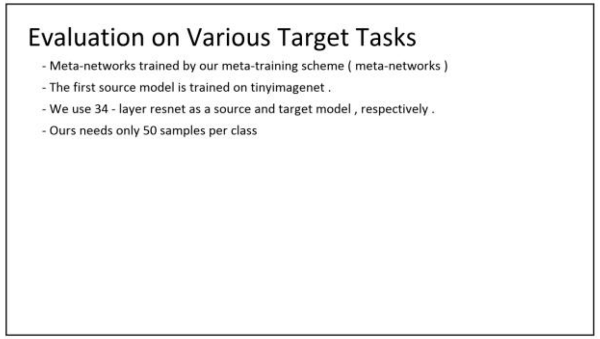


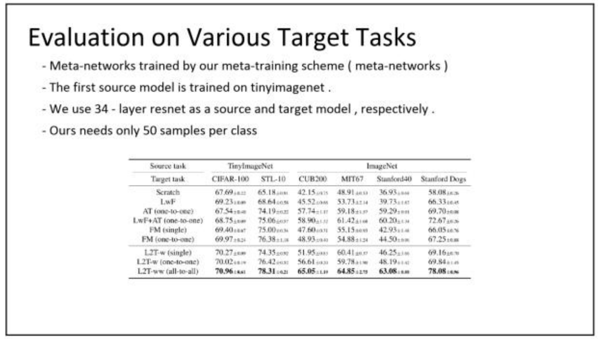






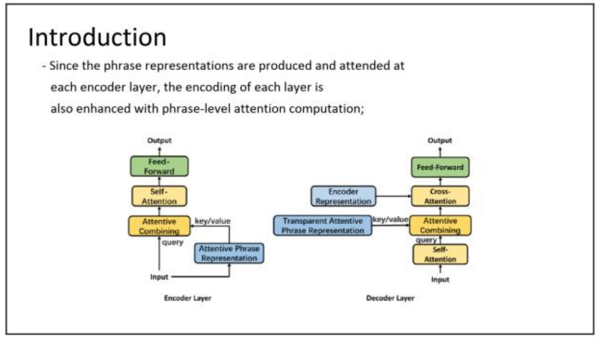


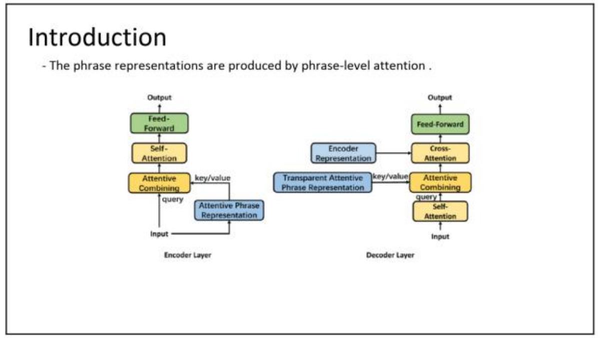


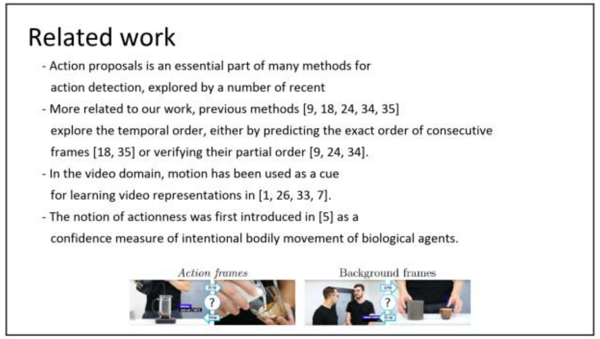


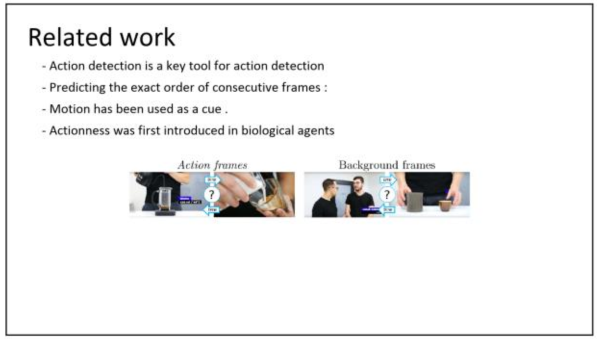
Qualitative Examples

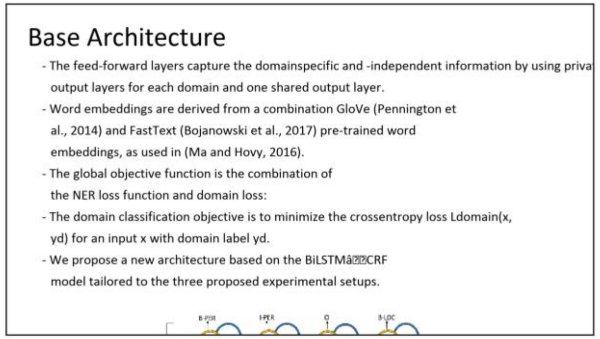
- Paraphrasing Module (**w/o** vs **w/**)

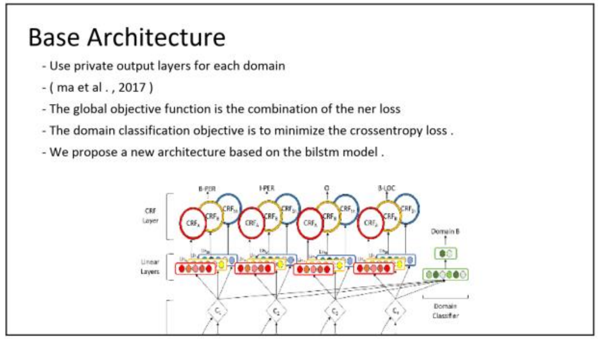


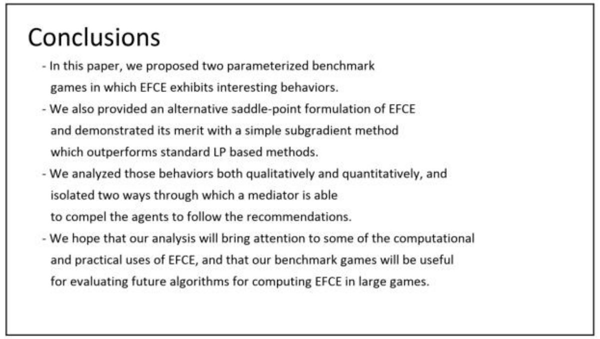


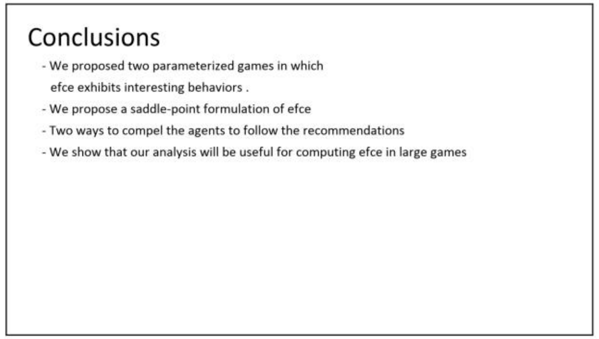






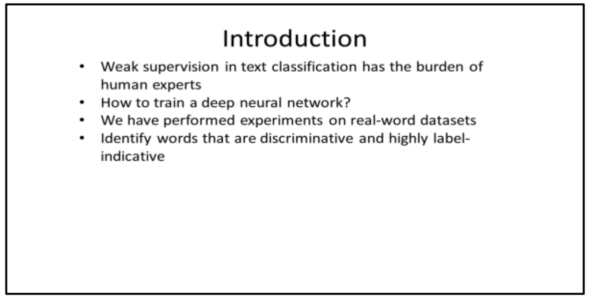


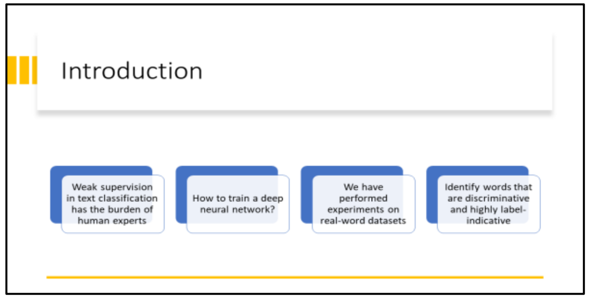


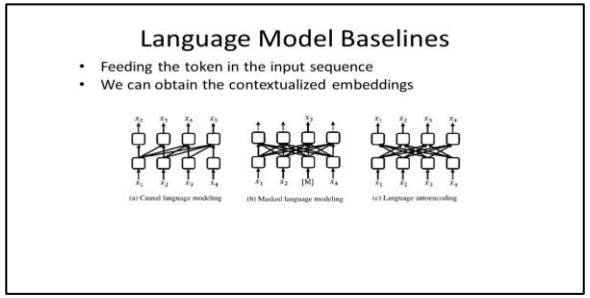


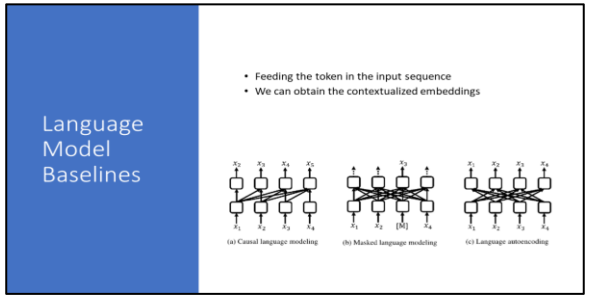
Qualitative Examples

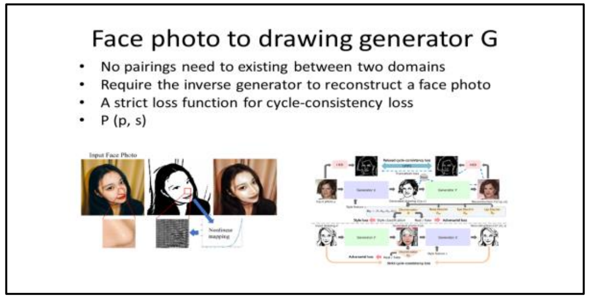
- Applying **Design Ideas**

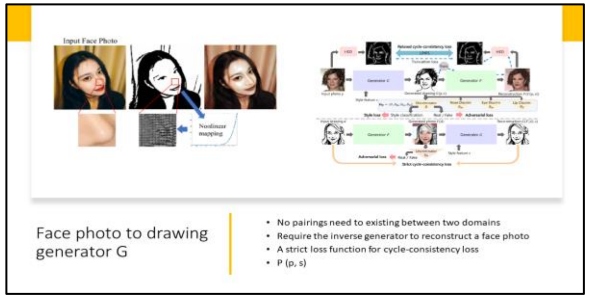


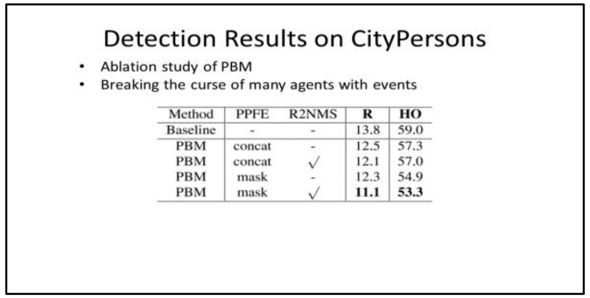


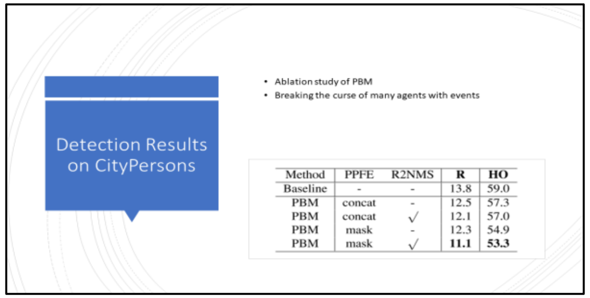


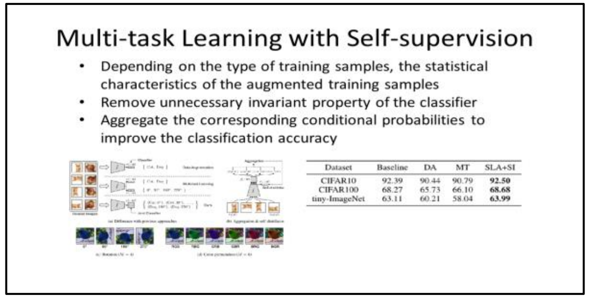


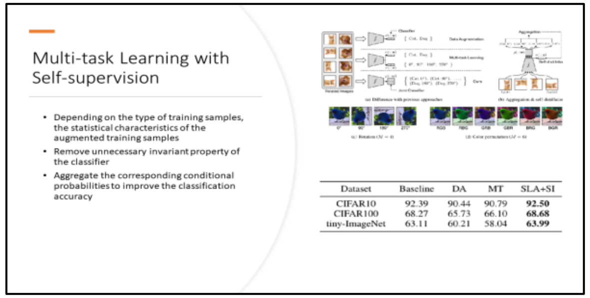


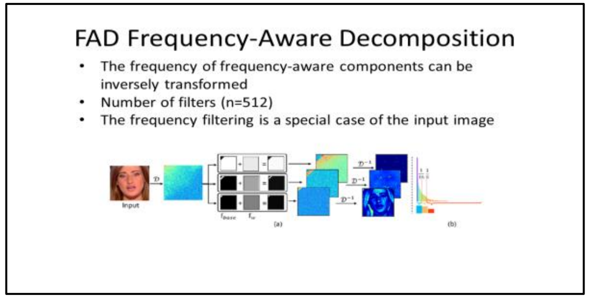


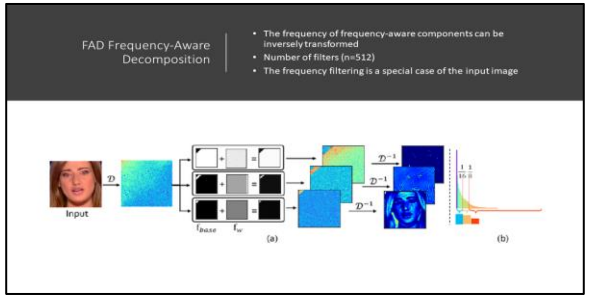












Conclusion

- DOC2PPT serves as a **multi-modal summarizer** to generate slide from academic documents
- We propose **hierarchical architecture**, **text-figure constrain**, and **paraphrasing module** to improve the quality of slide generation
- DOC2PPT **provides useful outline and flow** to make building a slide more efficiency


thanks!
Tombow ABT
Acid Free
N15