

DOC2PPT: Automatic Slide Deck Generation from Documents

AAAI'22



Tsu-Jui Fu



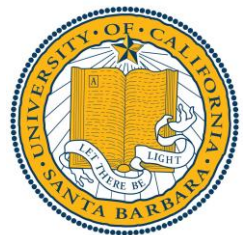
William Wang



Daniel McDuff



Yale Song



DOC2PPT

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

Figure 1: Some samples from the IMAGE-CHAT training set. For each sample we asked humans to engage in a conversation about the given image, where the two speakers, A and B, each have a given provided style.

B, and collect the dialogue using crowdworkers who are asked to both assume those roles, and to be engaging to the other speaker while doing so. It was emphasized in the data collection instructions that the style trait describes a trait of the speaker, not properties of the content of the image they are discussing. Some examples from the training set are given in Figure 1.

4 Models

Data Quality During data collection crowdworkers were manually monitored, checking to ensure they were following the instructions. Poor

Table 2: Module choices on IMAGE-CHAT. We compare different model variations for TRANSRESNET_{GEN}.

Model	Combiner	Text Encoders	Image Encoder	Turn 1			Turn 2			Turn 3			All
				R@1	R@1	R@1	R@1	R@1	R@1	R@1	R@1	R@1	
BR Baseline	na	na	na	na	na	na	na	na	na	na	na	na	2.18
TRANSRESNET _{GEN}	MM-Att	Separate	ResNet152	35.7	44.5	40.3	40.2	40.7	40.2	40.7	40.2	40.7	40.2
TRANSRESNET _{GEN}	MM-Sum	Separate	ResNet152	36.5	46.0	41.3	40.8	41.2	40.8	41.2	40.8	41.2	40.8
TRANSRESNET _{GEN}	MM-Sum	Shared	ResNeXt-IG-3.5B	53.6	49.0	41.3	47.3	47.3	47.3	47.3	47.3	47.3	47.3
TRANSRESNET _{GEN}	MM-Att	Shared	ResNeXt-IG-3.5B	64.4	49.0	41.3	48.9	48.9	48.9	48.9	48.9	48.9	48.9
TRANSRESNET _{GEN}	MM-Att	Separate	ResNeXt-IG-3.5B	51.5	50.5	43.8	49.3	49.3	49.3	49.3	49.3	49.3	49.3
TRANSRESNET _{GEN}	MM-Sum	Separate	ResNeXt-IG-3.5B	54.0	51.9	44.8	48.8	48.8	48.8	48.8	48.8	48.8	48.8

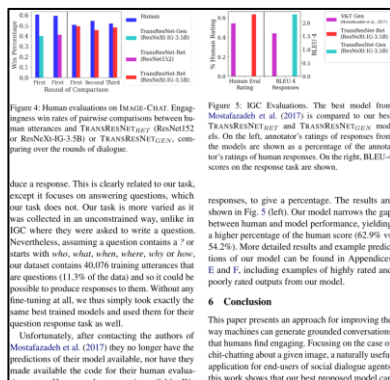
Table 3: Ablation on IMAGE-CHAT. We compare variants of our best TRANSRESNET_{GEN} generative and retrieval models (ResNeXt-IG-3.5B image encoder, and MM-Sum + separate text encoders for retrieval) where we remove modalities: image, dialogue history and style conditioning, reporting R@1/R@10 for retrieval and ROUGE-L for generation for dialogue turns 1, 2 and 3 independently, as well as the average over all turns.

Modality	TRANSRESNET _{GEN} (R@1/R@10)	TRANSRESNET _{GEN} (ROUGE-L)
Image Only	37.6 / 28.1	20.7 / 28.7
Style Only	18.3 / 15.3	17.0 / 16.9
Dialogue History Only	1.0 / 33.7	32.3 / 22.3
Style + Dialogue (no image)	18.3 / 45.4	43.1 / 35.4
Image + Dialogue (no style)	37.6 / 39.4	32.6 / 36.3
Image + Style (no dialogue)	64.0 / 41.1	35.2 / 43.4
Style + Dialogue + Image (full model)	64.0 / 51.9	44.8 / 36.3

condent evaluations at each round of dialogue for each example in the evaluation set, we have a separate set of human evaluators look at the provided conversation turns, and ask them to compare two possible utterances for the next turn of conversation, given the image, dialogue history and relevant style (which is the same for both human author and model, so there is no advantage). We ask the evaluators in a blind test to choose the "more engaging" using

Figure 2: The TRANSRESNET_{GEN} 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 (Marsel and Rodriguez, 2010). The second is a ResNeXt (19 × 484 (Xie et al., 2017) trained on 3.4 billion im-



DOC2PPT

Image-Chat

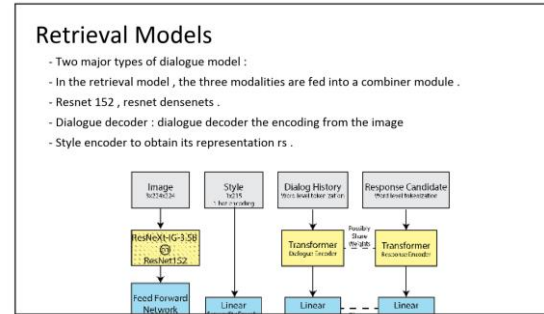
- Speaker b : a & b .
- We apply a set of 215 possible style traits , using an existing set from shuster et al .
- Who will be assigned to a person ?

Figure 3: Dialogue example with images and style traits. The dialogue involves two speakers, A and B, discussing images of food and nature. Style traits like 'I'm so thankful for this delicious food' and 'I love to hike' are used to guide the conversation.

Human Evaluations on IMAGE-CHAT

- Ablation study for both retrieval and generative models
- What is the best of both worlds ?
- Resnet 152 , resnet densenets .
- We ask the evaluators to choose the two possible utterances :

Figure 4: Human evaluation bar chart showing win percentage for Human, TRANSRESNET-Gen, and ResNeXt-IG-3.5B across rounds 1, 2, and 3.



Conclusion

- Can be studied in future work .
- (zhang et al , 2018)
- Humans can relate to social dialogue agents
- Retrieval models outperformed their generative models .
- A new dataset is made of a new dataset .

DOC2PPT

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

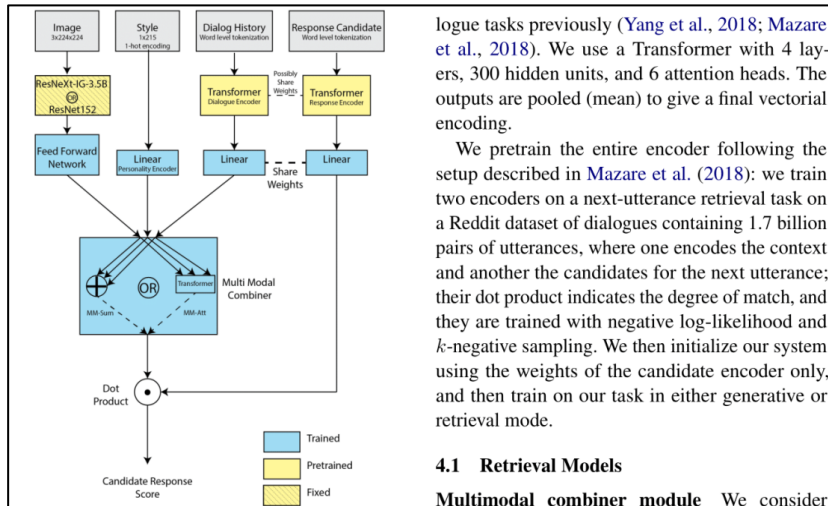


Figure 2: The TRANSRESNETRET multimodal architecture for grounded dialogue. There are several options: different image encoders (ResNet152 or ResNeXt-IG-3.5B), text encoders (shared or separate Transformers for history and response), and different multimodal combiners (sum or attention-based).

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:

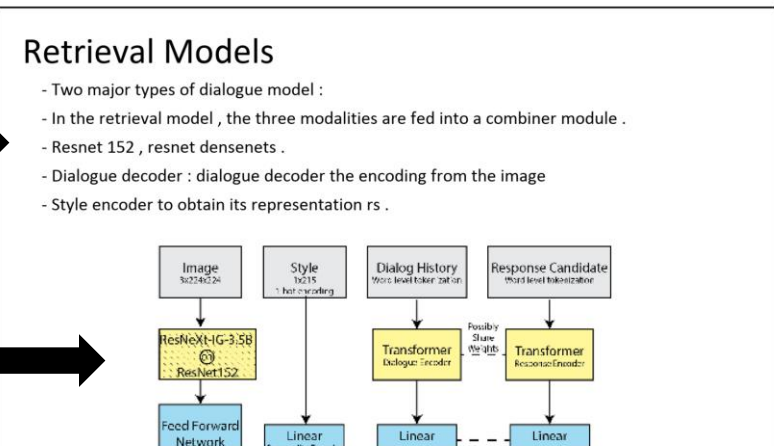
Multimodal sum combiner (MM-sum): Given an input image, style trait and dialogue (I, S, D), together with a candidate response C , the score of the final combination is computed as $s(I, S, D, C) = (r_I + r_S + r_D) \cdot r_C$.

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 vectors r_I, r_S and r_D and feed them to a second

Text Summarization



Figure Retrieval



Retrieval Models

- Two major types of dialogue model :
- In the retrieval model , the three modalities are fed into a combiner module .
- Resnet 152 , resnet densenets .
- Dialogue decoder : dialogue decoder the encoding from the image
- Style encoder to obtain its representation rs .

-
- **Multi-Page**
-

Dataset Building

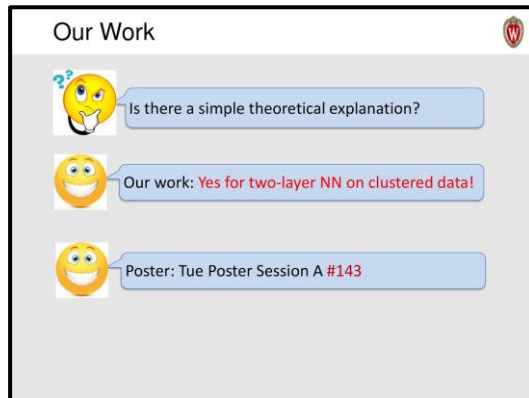
- Crawl **paper-slide pairs** from AI conferences
 - Computer Vision (CVPR, ECCV, ...)
 - Natural Language Processing (ACL, NAACL, ...)
 - Machine Learning (ICLR, ICML, ...)
- **5,873** in total
 - 4,686 / 592 / 595 (train / val / test)
- To prepare the data for training, needs some **preprocessing** in advance

Dataset Building

- Extract **text content** from a slide
 - Azure CV to do **Optical Character Recognition (OCR)**



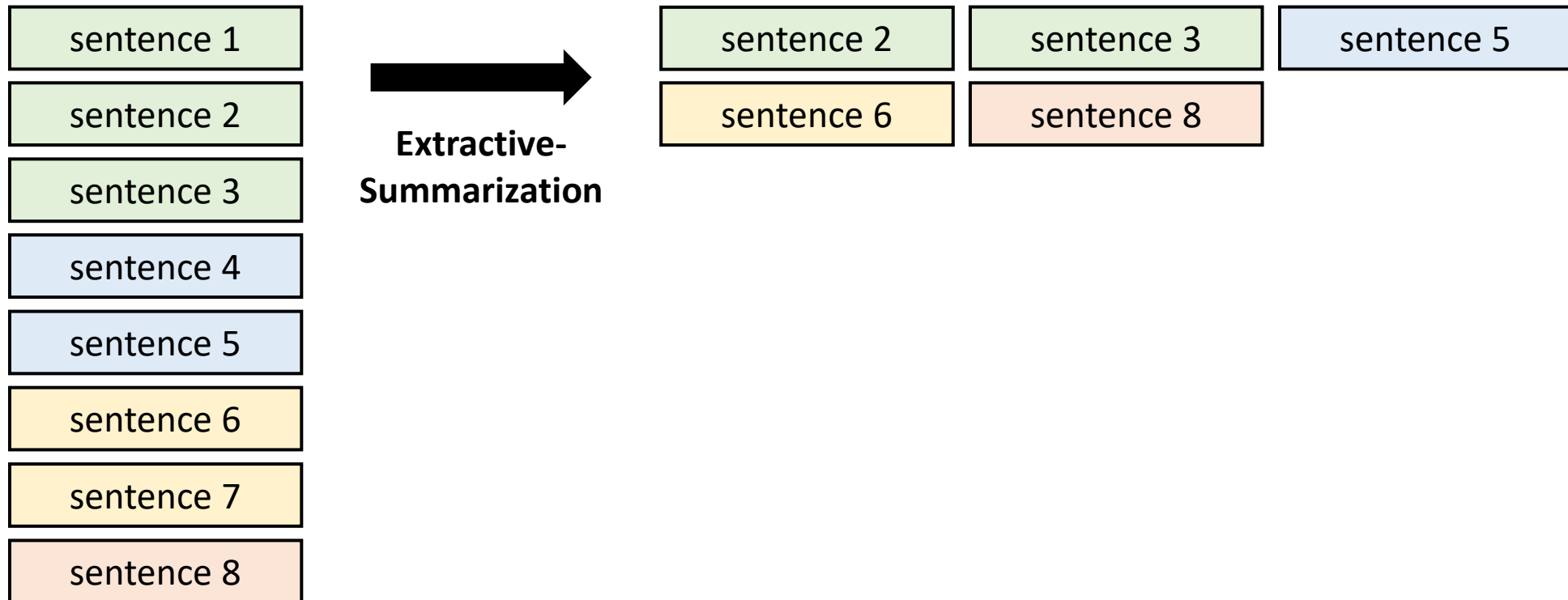
- Learning Over-Parameterized –Neural
- Networks on Structured Data
- Yingyu Liang@UWLMadison
- Joint work with Yuanzhi Li@Princeton -Y Stanford



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

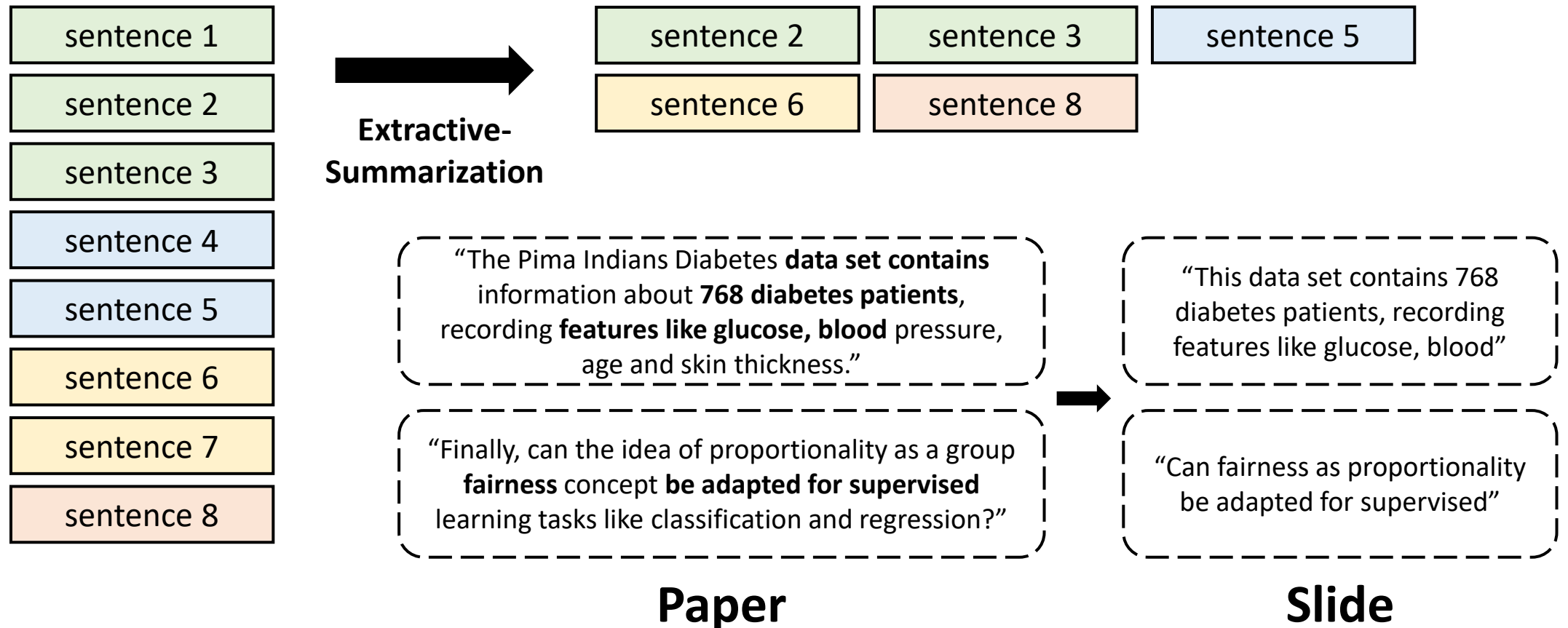
Dataset Building

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



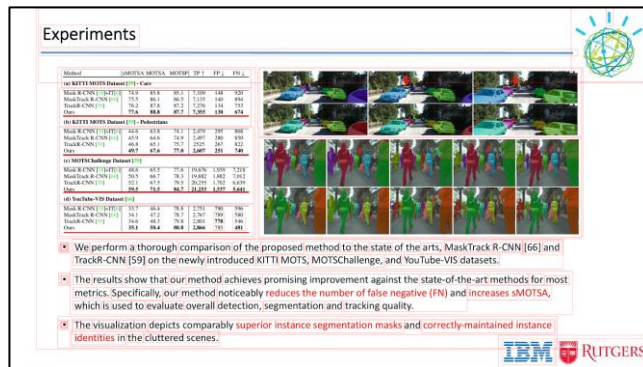
Dataset Building

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

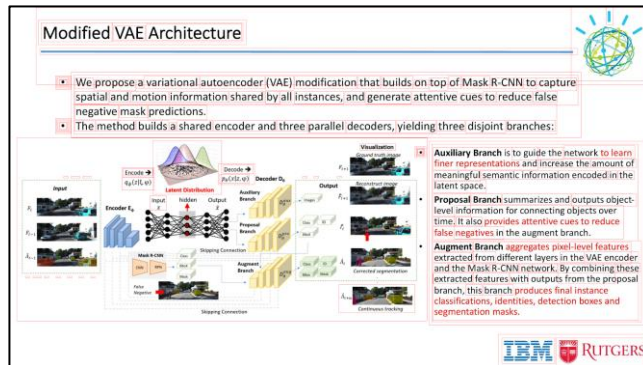
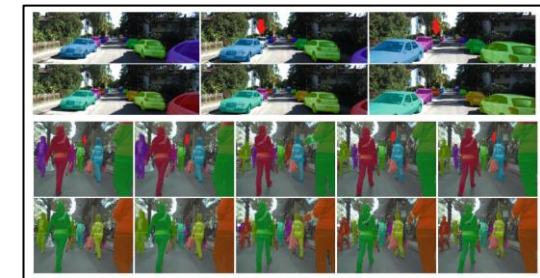
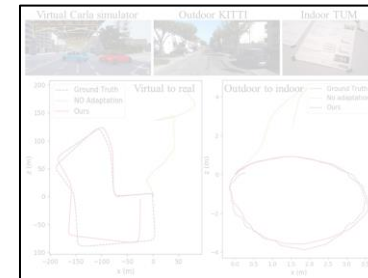


Dataset Building

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



Method	smOTSA	MOTSA	MOTSP	TP ↑	FP ↓	FN ↓
(a) KITTI MOTS Dataset [59] - Cars						
Mask R-CNN [17][10]	74.9	85.8	85.1	7,109	148	928
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	7,385	130	674
(b) KITTI MOTS Dataset [59] - Pedestrians						
Mask R-CNN [17][10]	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	2,525	267	822
Ours	49.7	67.6	77.0	2,607	251	740
(c) MOTChallenge Dataset [59]						
Mask R-CNN [17][10]	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	1,557	5,661
(d) YouTube-VIS Dataset [66]						
Mask R-CNN [17][10]	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



Slide

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
GroundR [35]	19.1	60.2	38.5	35.7	38.9	75.7
Deaf-GroundR	2.2	7.7	7.9	8.0	8.0	27.1
Shuffle-GroundR	13.1	41.8	28.6	27.2	27.6	58.5
Obj-Attr-GroundR	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 [46]	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

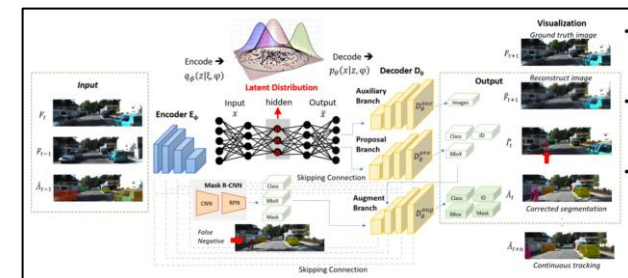
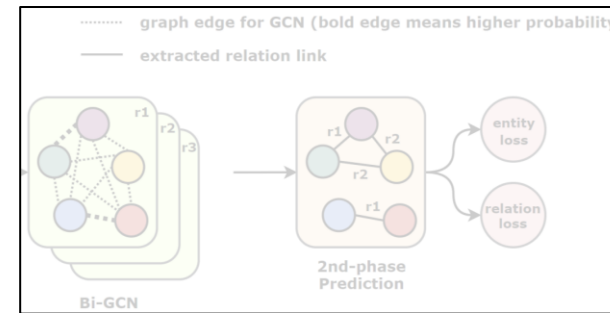
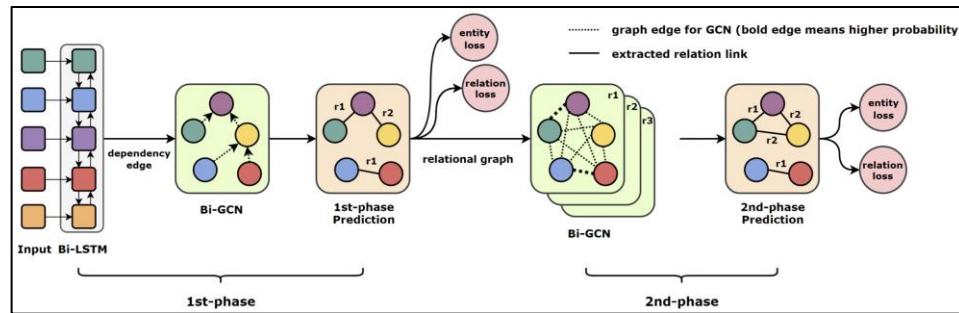


Figure from Paper

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



Different
Expression

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%


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

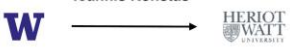
Dataset Building

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

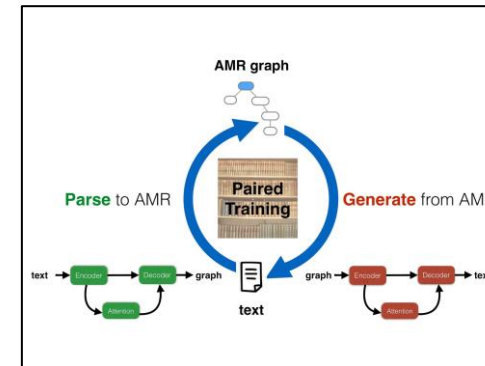
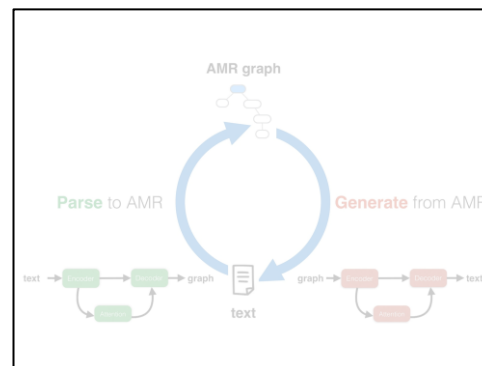
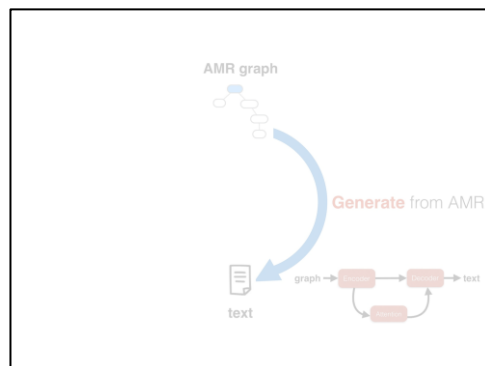


Neural AMR: Sequence-to-Sequence Models for Parsing and Generation

Ioannis Konstas



joint work with Srinivasan Iyer, Mark Yatskar, Yejin Choi, Luke Zettlemoyer



Time Expression Analysis – Eureka!

• Similar syntactic behaviour: (1) POS information cannot distinguish time expressions from common text, but (2) within time expressions, POS tags can help distinguish their constituents.

- (1) For the top 40 POS tags (10 × 4 datasets), 37 have percentage lower than 20%, other 3 are CD.
- (2) Time tokens mainly have NN^s and RB, modifiers have JJ and RB, and numerals have CD.

When seeing (2), we realize that this is exactly how linguists define part-of-speech for language; similar words have similar syntactic behaviour. The definition of part-of-speech for language inspires us to define a type system for the time expression, part of language.

Our Eureka! moment

Time Expression Analysis - Idea

• Summary

- On average, a time expression contains two tokens; one is time token and the other is modifier/numeral. And the time tokens are in small size.

• Idea for recognition

- To recognize a time expression, we first recognize the time token, then recognize the modifier/numeral.

20 days; this week; next year; July 29; ...


Time Expression Analysis - Idea

• Summary

- On average, a time expression contains two tokens; one is time token and the other is modifier/numeral. And the time tokens are in small size.

• Idea for recognition

- To recognize a time expression, we first recognize the time token, then recognize the modifier/numeral.



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)

Two Challenges in VQA

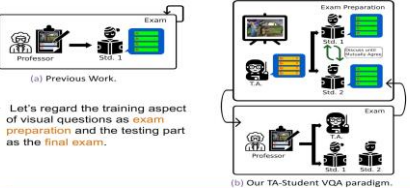
- The "ceiling effect" for simple consecutive training by a single model.
- Questions targeted at a single image are off in format and lack diversity in content.



TA-Student VQA 2

➔ 1. Introduction

Previous Works vs. Ours

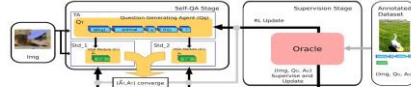


➤ Let's regard the training aspect of visual questions as **exam preparation** and the testing part as the **final exam**.

TA-Student VQA 4

Overview

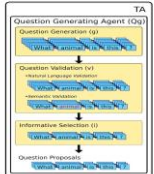
- Self-QA Stage
 - Question Generating Agent (Q_g)
 - Question-Answering Agents (A_{gr})
- Supervision Stage
 - Oracle Check Model (O)



Overview of our approach. The system consists of two stages, Self-QA stage and Supervision Stage, and these two stages will execute iteratively.

TA-Student VQA 5

Question Generating Agent (Q_g)




- **Question Generation Model (g)**: Generate questions based on a given image.
- **Question Validation Model (v)**: Grammar checker and main components checker.
- **Informative Selection Model (i)**: Select the most informative questions among question proposals, based on policy from Oracle Check Model.

TA-Student VQA 6

➔ 3. Approach

Qualitative Results



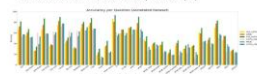
Question Proposals per Iteration. With the update in the Question Generating Agent, the question proposals are with increasing sophistication.

TA-Student VQA 10


Quantitative Results

Model	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
TA-Student VQA	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60

Overall Performance Comparison (%)



Accuracy per Annotated Question in Dataset (%)

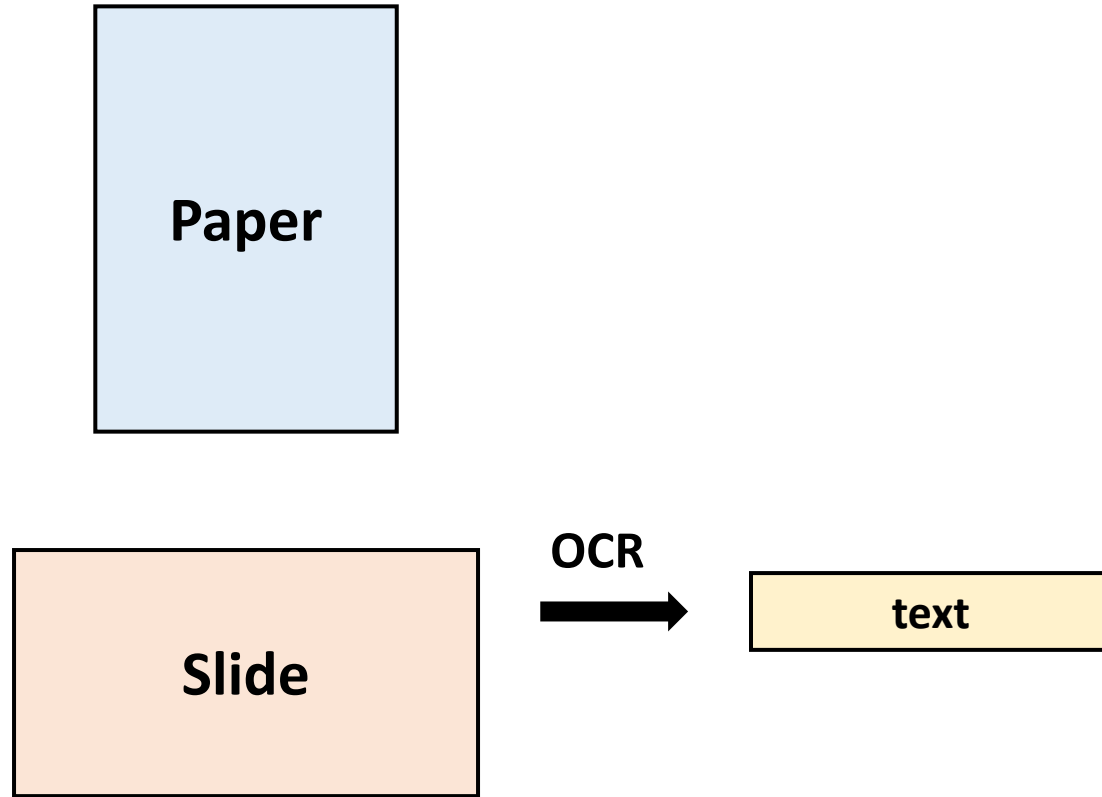


Accuracy per Generated Question in Dataset (%)

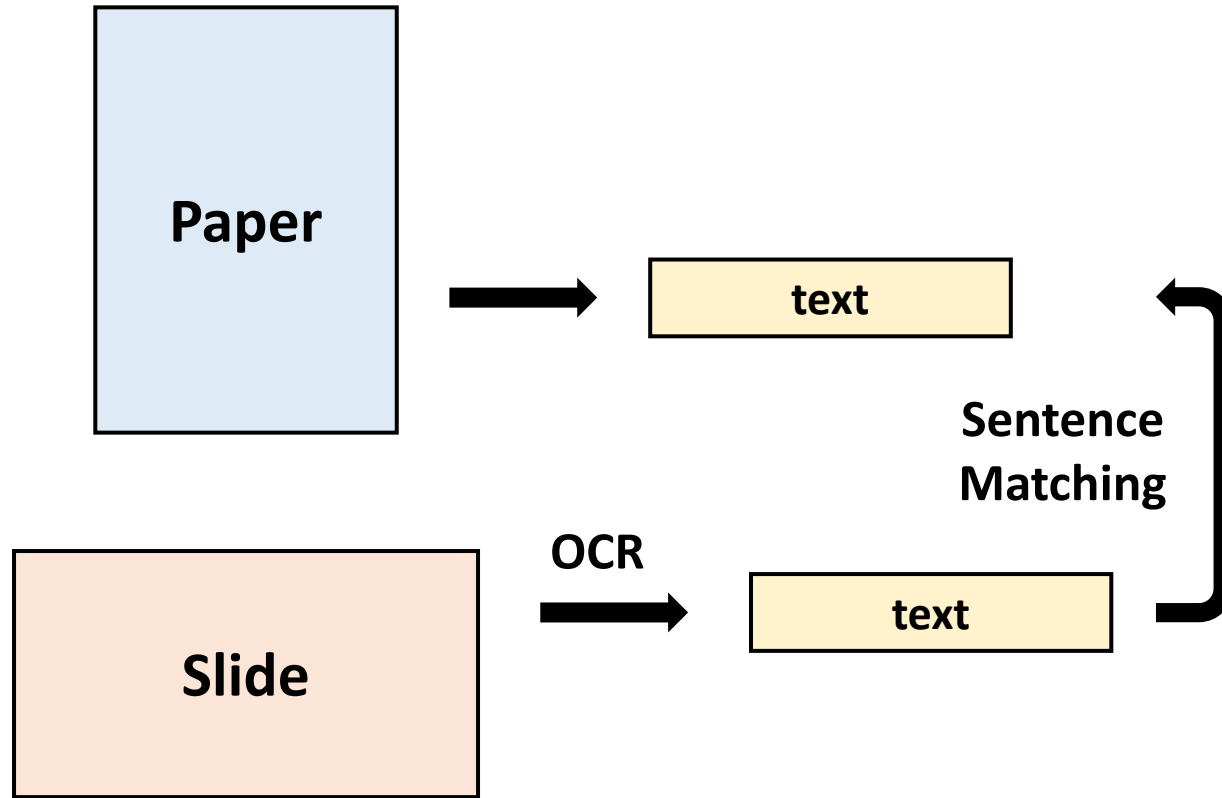
TA-Student VQA 11

➔ 4. Experiments

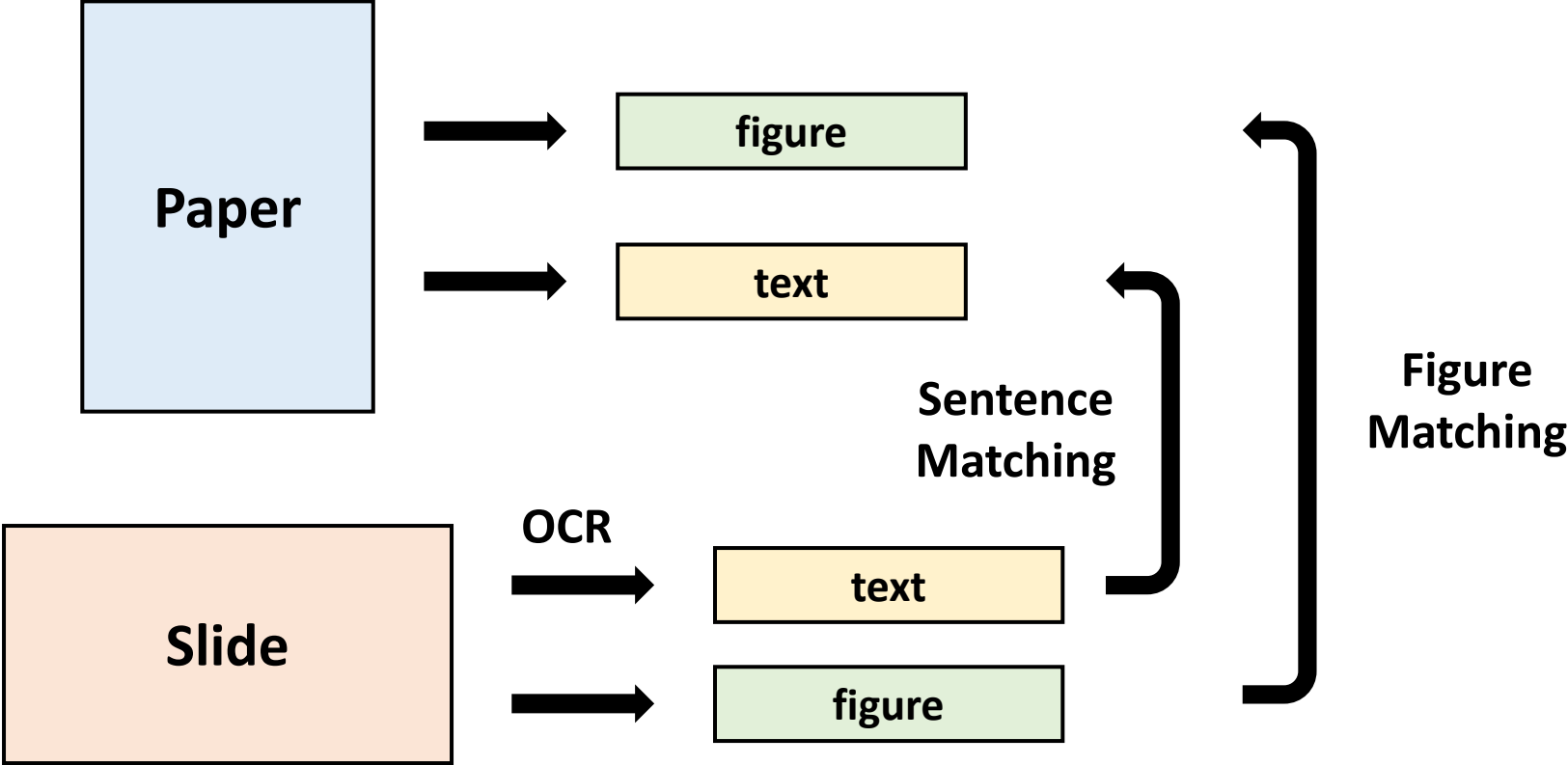
Dataset Building



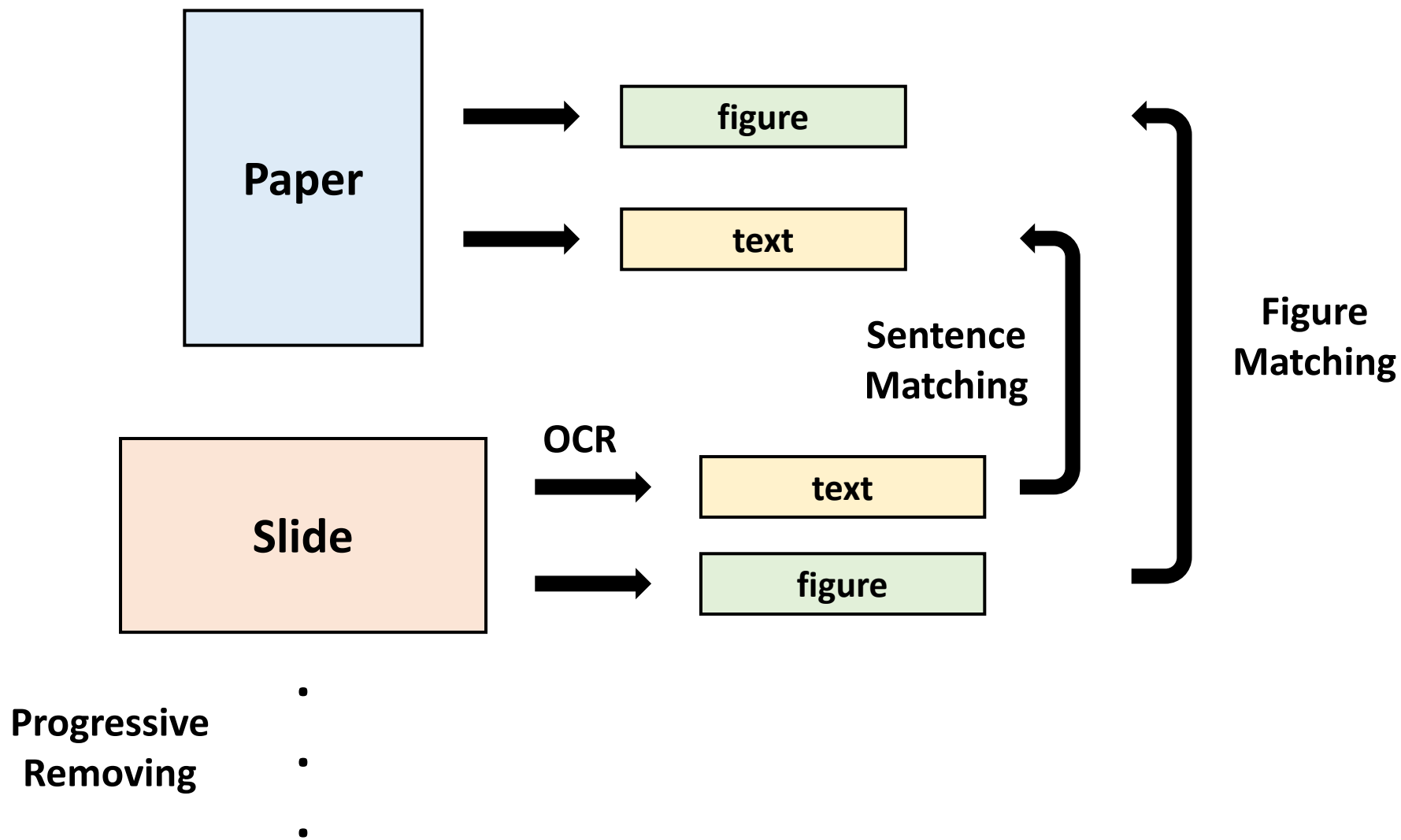
Dataset Building



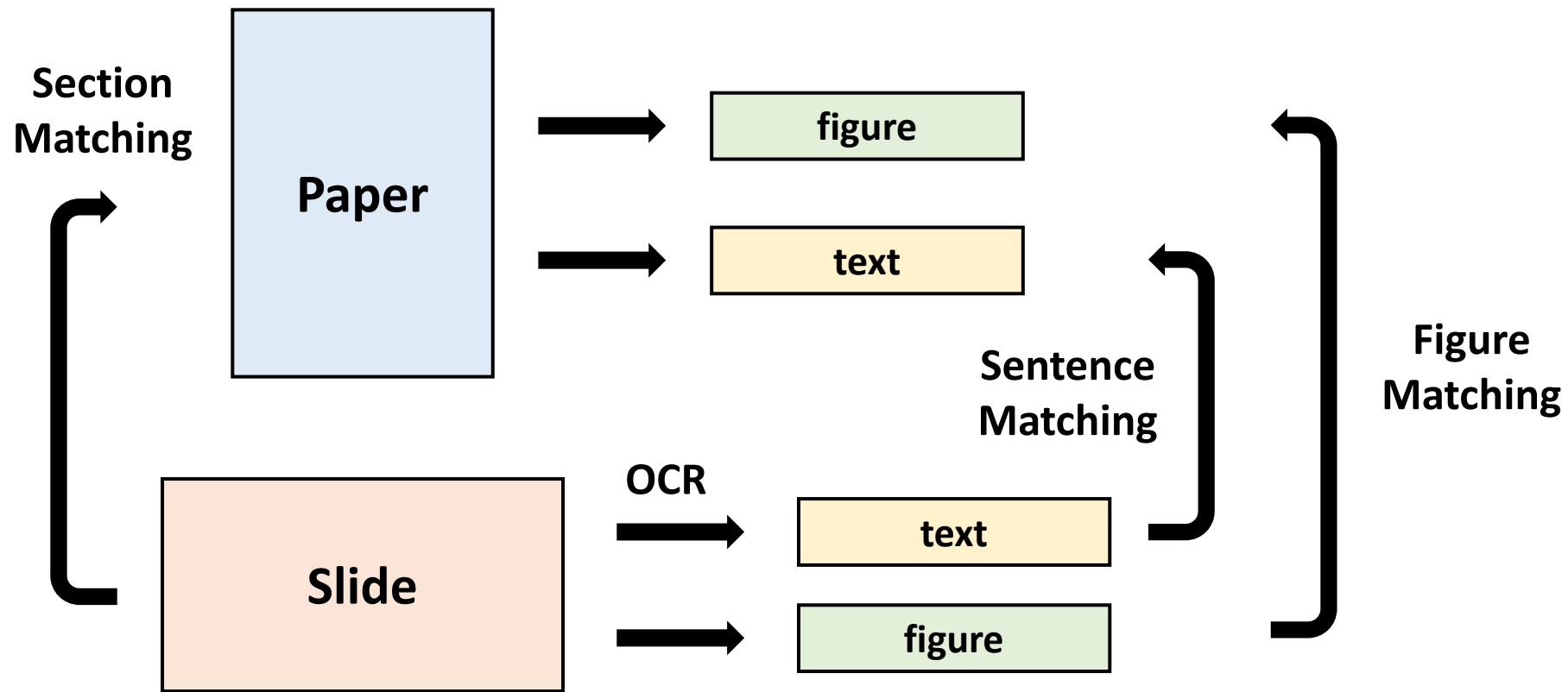
Dataset Building



Dataset Building



Dataset Building



Progressive
Removing

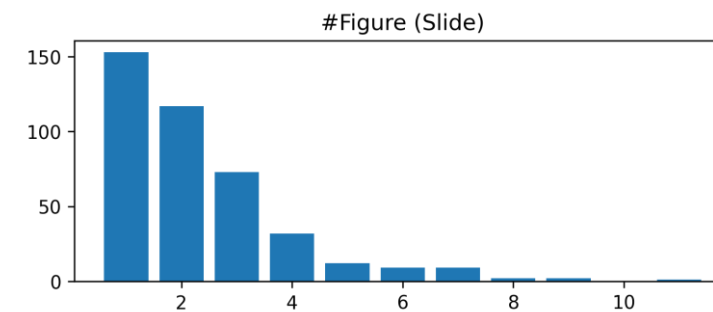
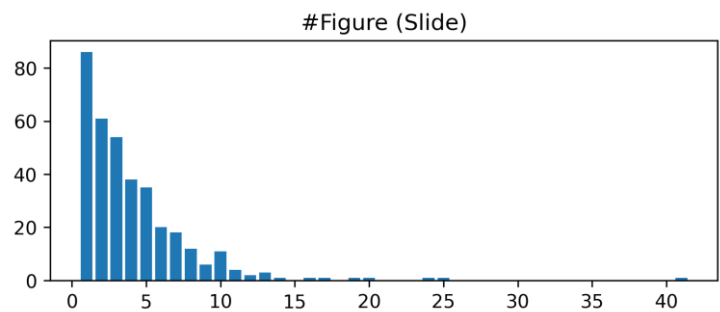
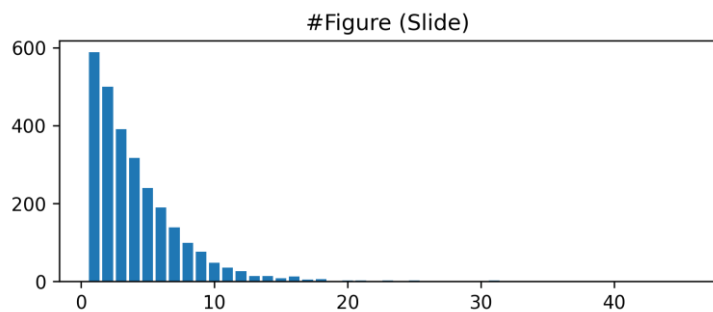
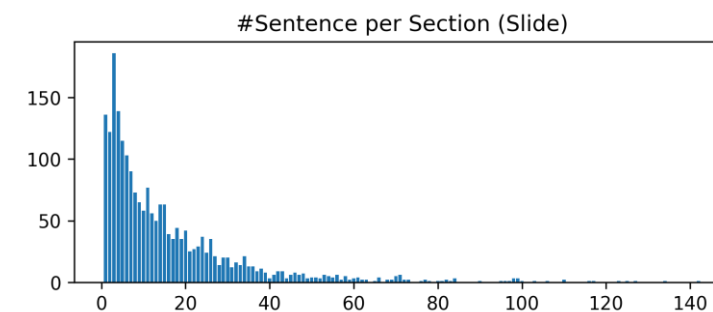
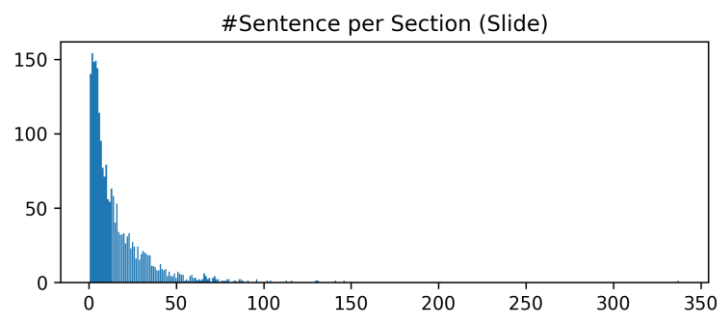
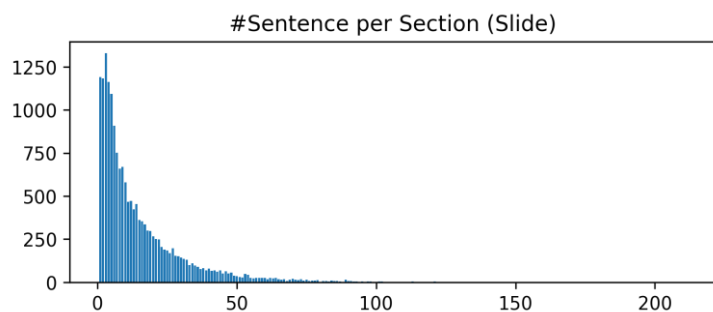
-
-
-

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



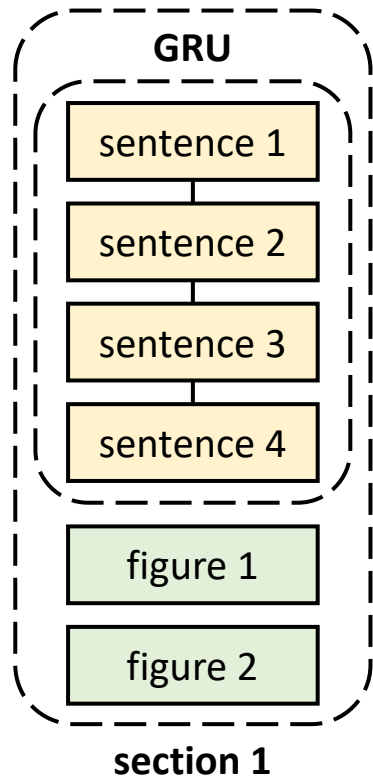
Train

Val

Test (Human)

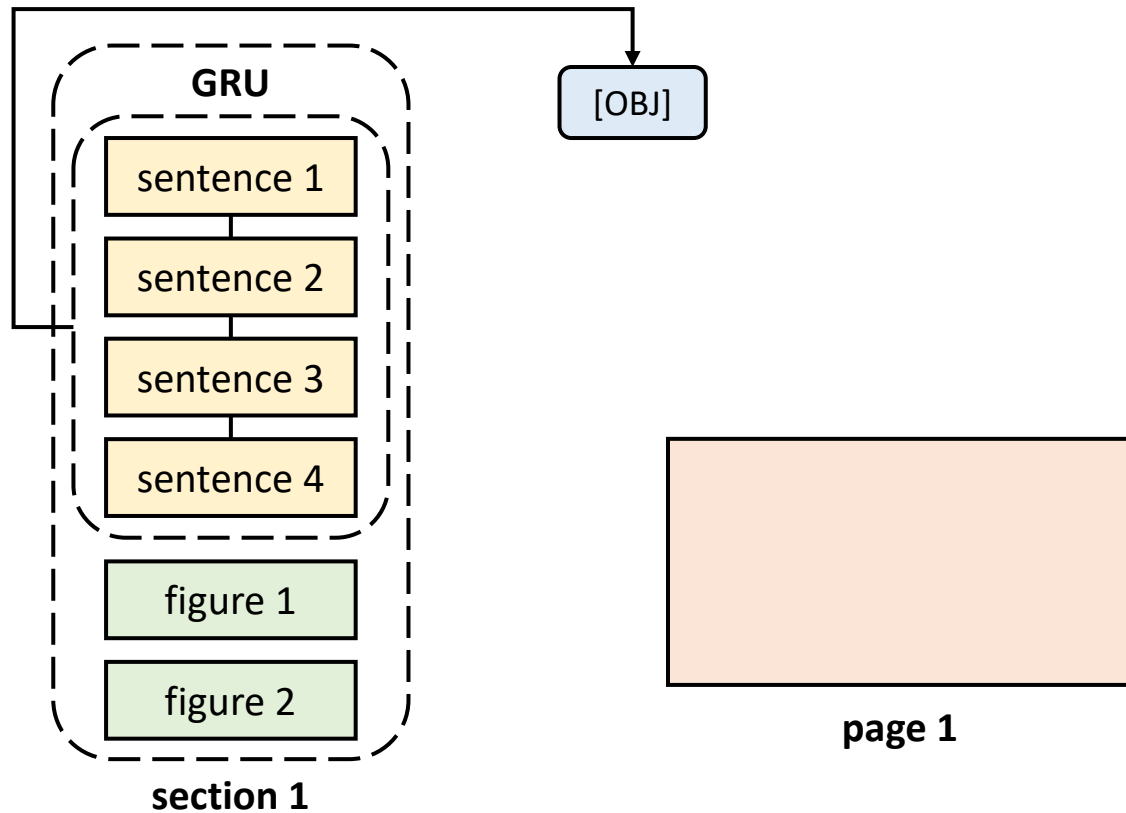
Model (Baseline)

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



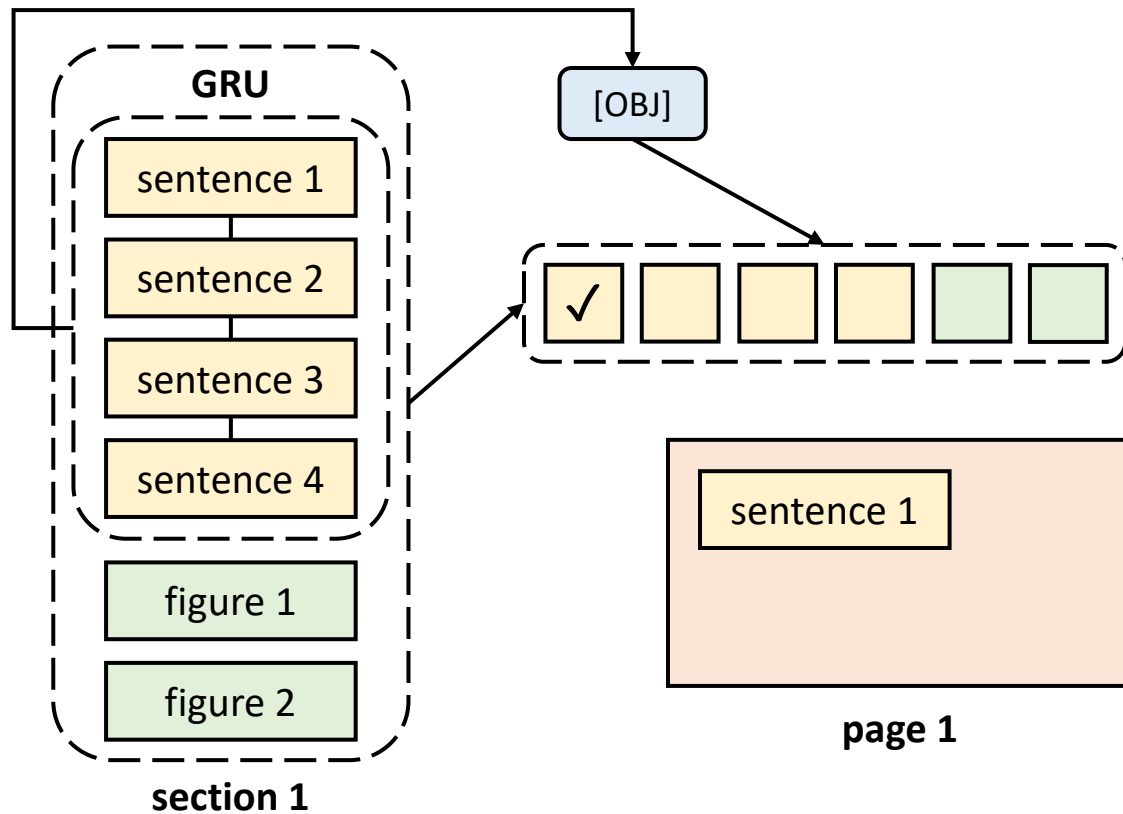
Model (Baseline)

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



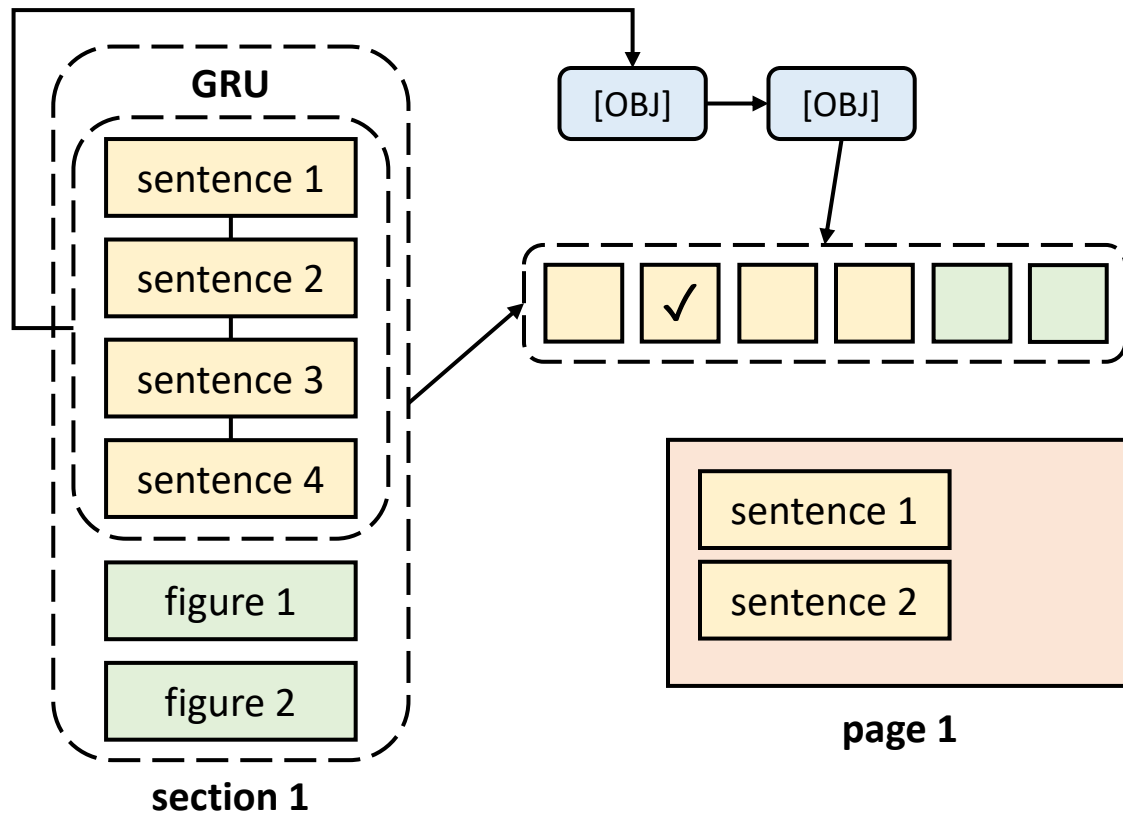
Model (Baseline)

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



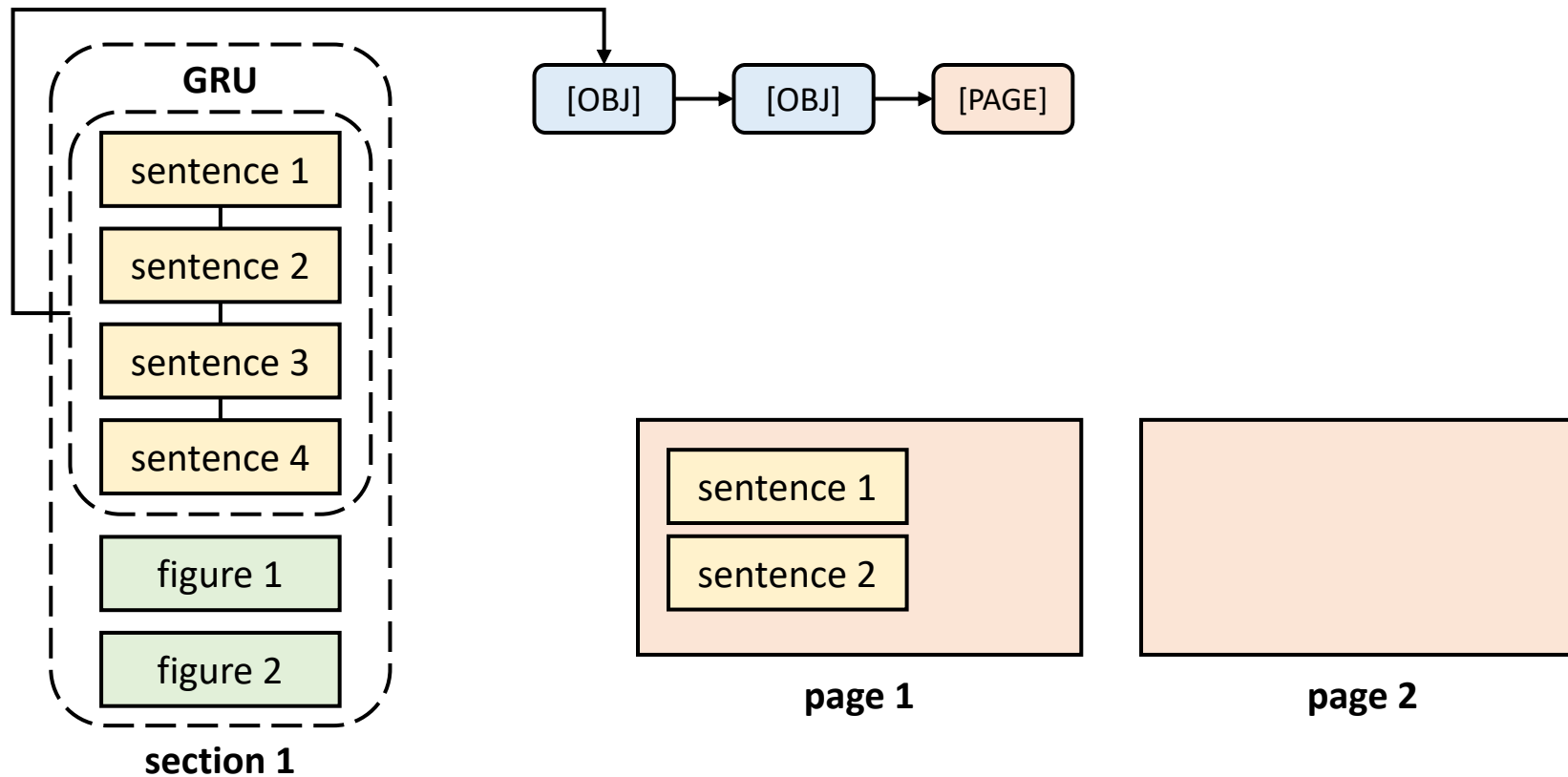
Model (Baseline)

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



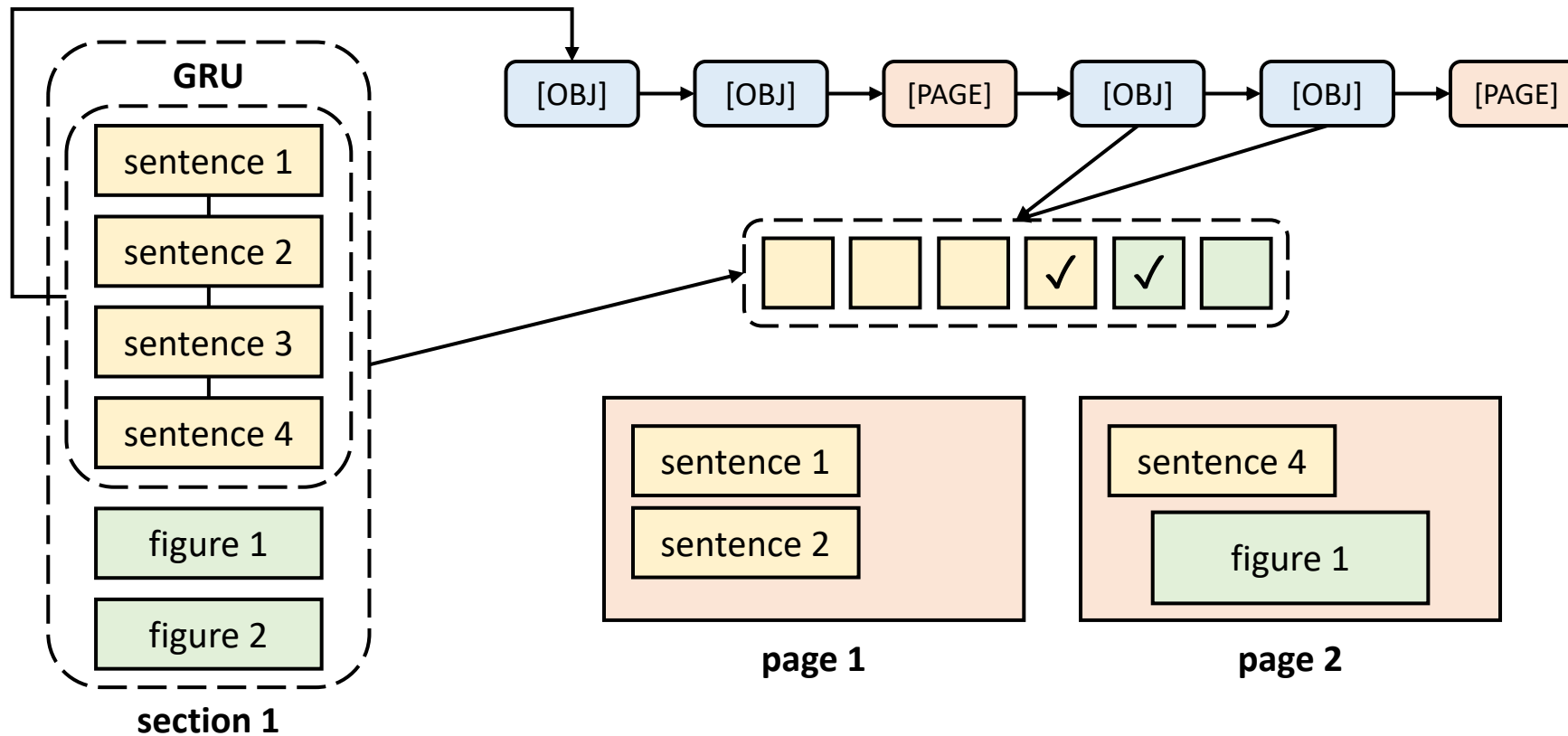
Model (Baseline)

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



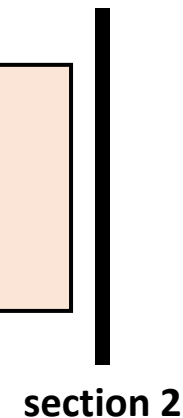
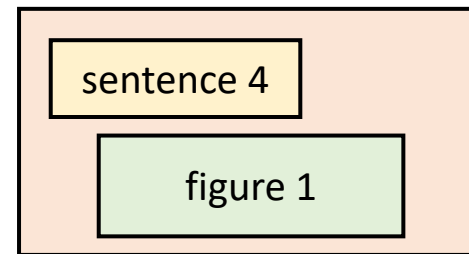
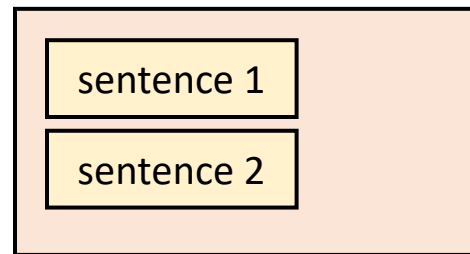
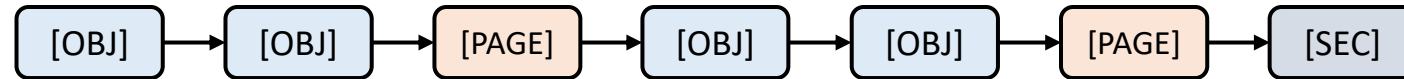
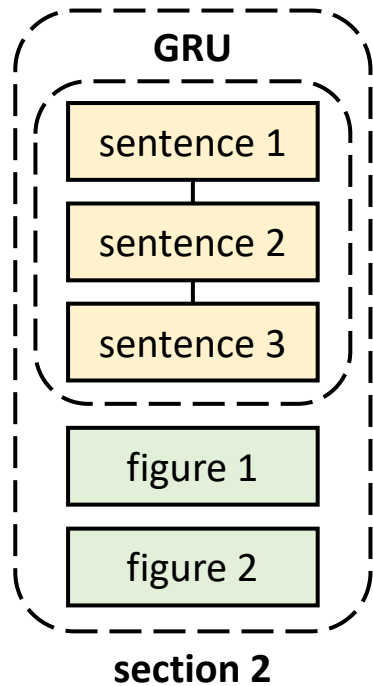
Model (Baseline)

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



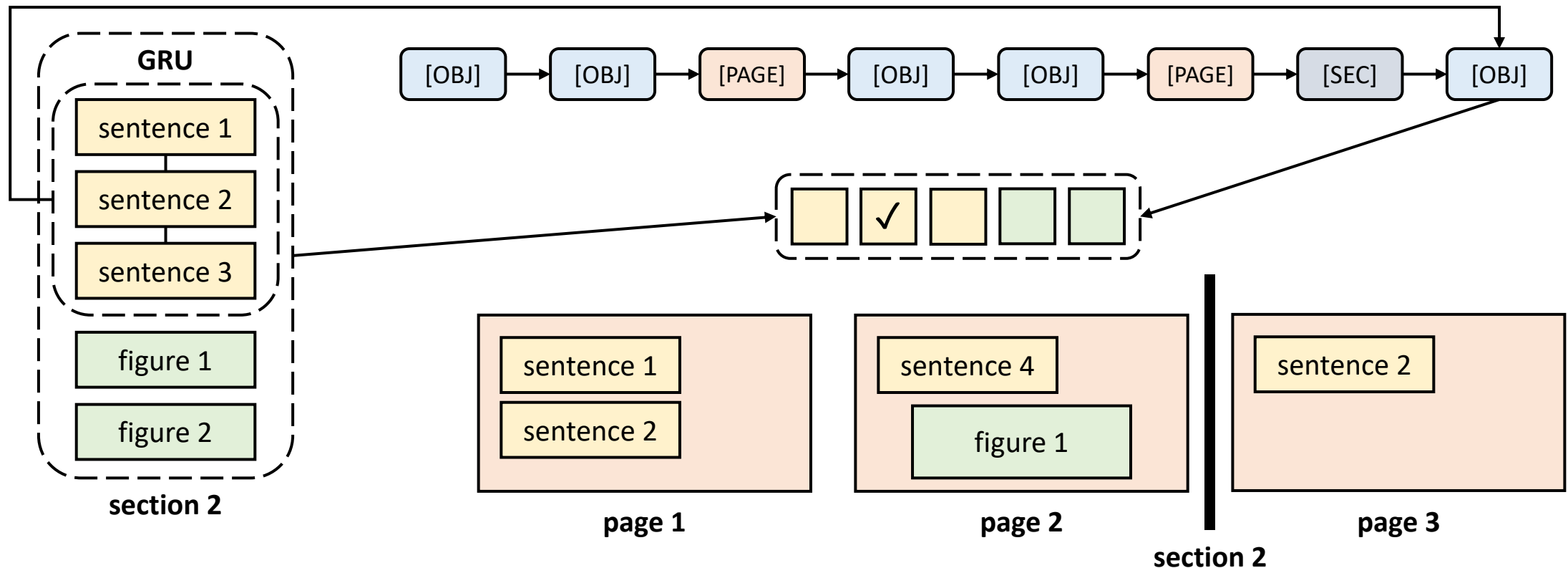
Model (Baseline)

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



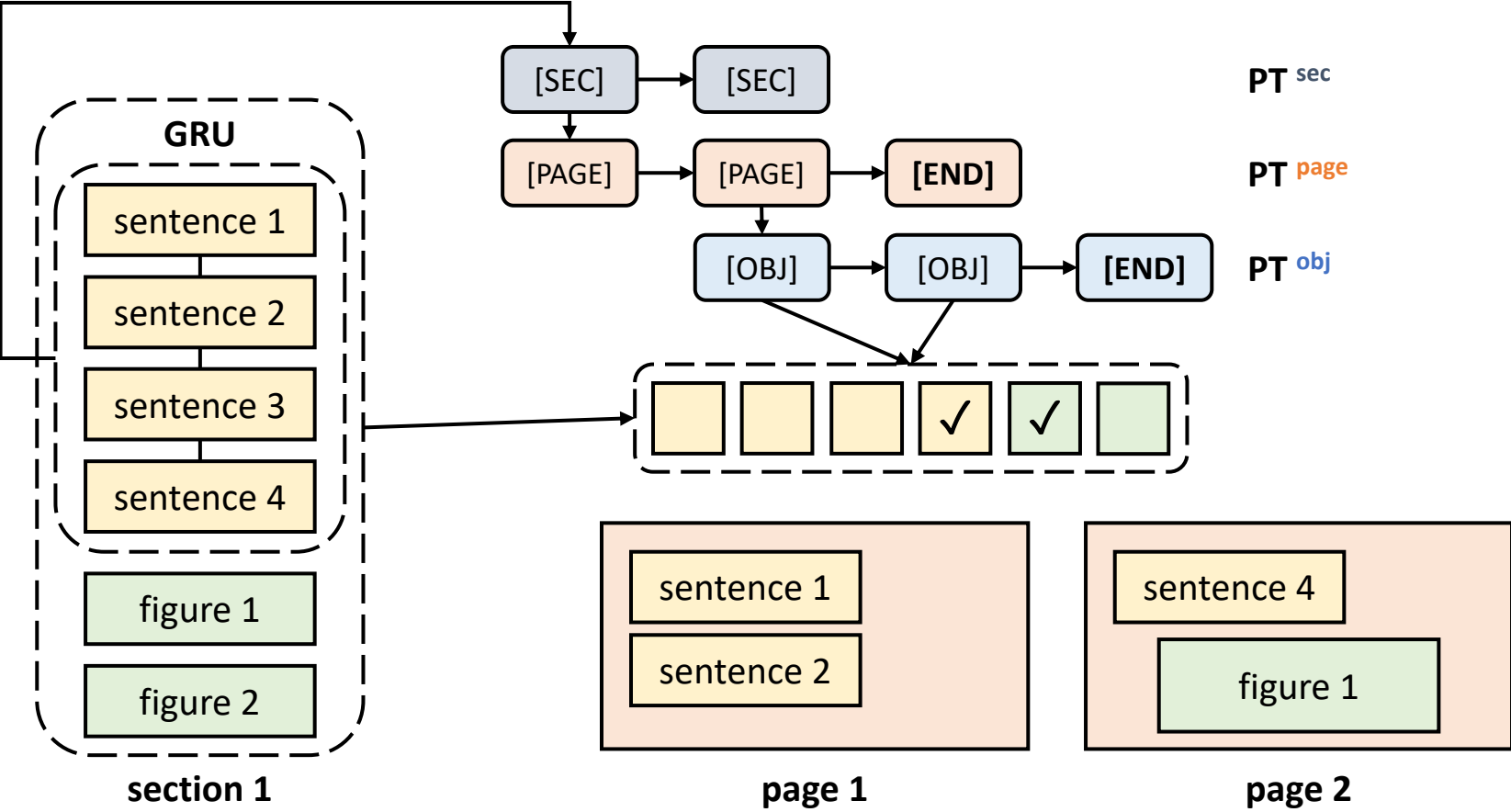
Model (Baseline)

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



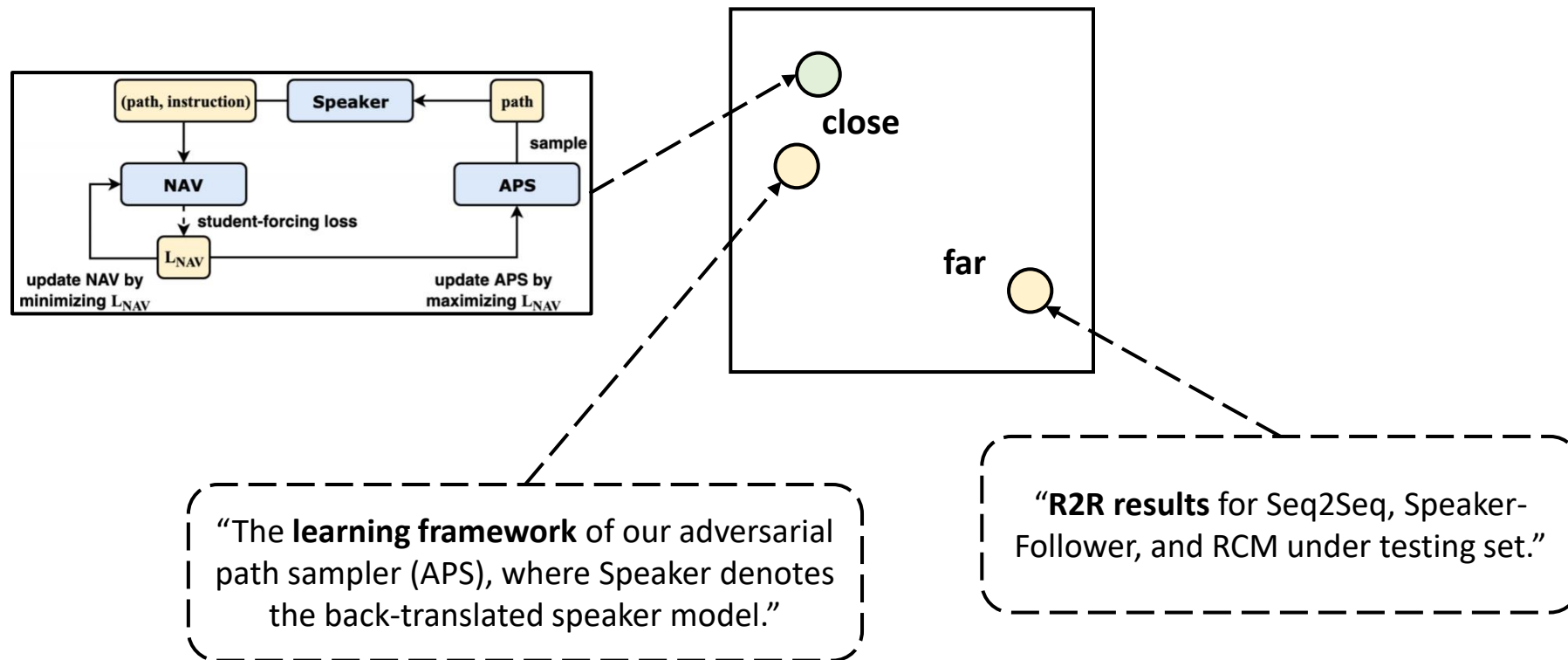
Model (HSE)

- **Hierarchical Slide Extractor (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**



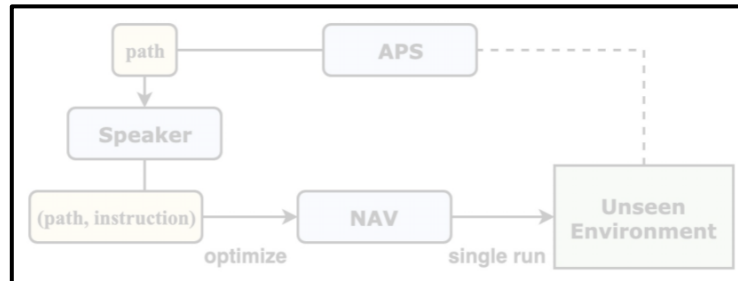
TextFigure Module

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

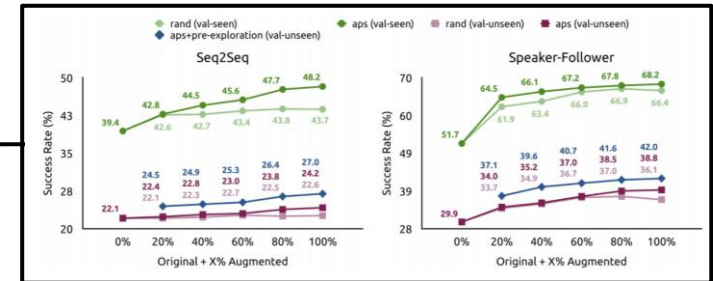
Result

- 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)



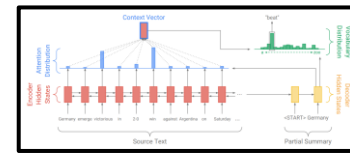
add unused
(related)



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 .”



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



paraphrase

“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

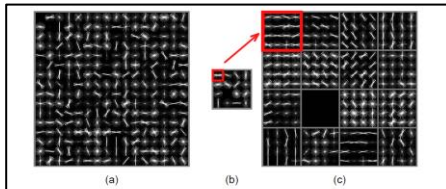


Fig. 3. a: The method proposed by Sadeghi and Forsyth [2] quantizes each cell into one of 256 pre-defined clusters. Nearest neighbour search is a significant bottleneck in their technique. In this paper we use hierarchical clustering instead of flat clustering. b: each cell is first quantized into one 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

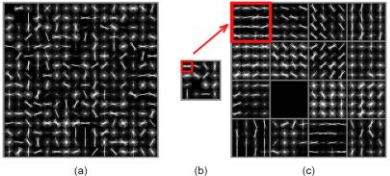


Fig. 3. a: The method proposed by Sadeghi and Forsyth [2] quantizes each cell into one of 256 pre-defined clusters. Nearest neighbour search is a significant bottleneck in their technique. In this paper we use hierarchical clustering instead of flat clustering. b: each cell is first quantized into one 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

Hierarchical Vector Quantization

- We use a hierarchical clustering technique to speed up this process.
- We use vector quantization for the same purpose but with a slightly different approach.
- 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.

HSE w/ TextFigure & Paraphrasing

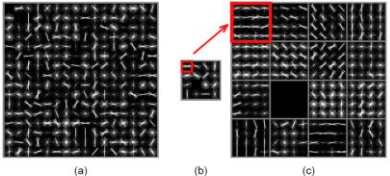


Fig. 3. a: The method proposed by Sadeghi and Forsyth [2] quantizes each cell into one of 256 pre-defined clusters. Nearest neighbour search is a significant bottleneck in their technique. In this paper we use hierarchical clustering instead of flat clustering. b: each cell is first quantized into one 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

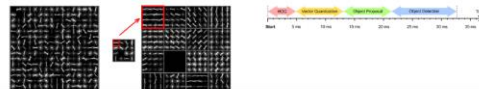
Hierarchical Vector Quantization

- We use a hierarchical clustering technique to speed up this process.
- We use vector quantization for the same purpose but with a slightly different approach.
- 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.

↓
TextFigure
Module

Hierarchical Vector Quantization

- We use a hierarchical clustering technique to speed up this process.
- We use vector quantization for the same purpose but with a slightly different approach.
- 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.



HSE w/ TextFigure & Paraphrasing

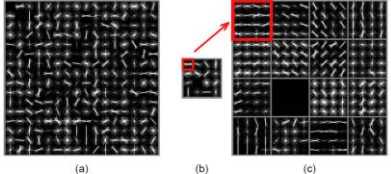


Fig. 3. a: The method proposed by Sadeghi and Forsyth [2] quantizes each cell into one of 256 pre-defined clusters. Nearest neighbour search is a significant bottleneck in their technique. In this paper we use hierarchical clustering instead of flat clustering. b: each cell is first quantized into one 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

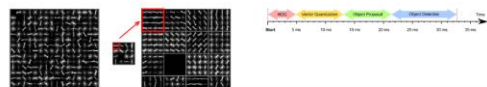
Hierarchical Vector Quantization

- We use a hierarchical clustering technique to speed up this process.
- We use vector quantization for the same purpose but with a slightly different approach.
- 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.

⬇
TextFigure Module

Hierarchical Vector Quantization

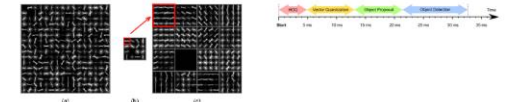
- We use a hierarchical clustering technique to speed up this process.
- We use vector quantization for the same purpose but with a slightly different approach.
- Then according to the nearest cluster in the first step we compare against 16 other clusters to find the nearest cluster (Figure 3, c).
- 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.



➔
Paraphrasing Module

Hierarchical Vector Quantization

- We use a hierarchical clustering technique to speed up this process .
- Vector quantization for the same feature space .
- How to find the nearest cluster in the first step ?
- Results : k-means algorithm using .
- Our method leads to a negligible loss of 0.001 in map



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

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

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

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

- 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

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

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

- 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	X	X	X	27.2	21.8	13.2	21.9	16.5	3.6	
	X	X	X	27.7	22.9	14.6	23.7	18.1	4.3	
HSE	X	✓	X	32.3	26.7	14.6	23.7	18.1	4.7	
	✓	X	X	28.7	24.0	14.8	32.4	20.3	7.9	
	✓	X	✓	28.7	24.0	24.6	40.5	30.6	13.8	
	✓	✓	X	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	X	X	X	27.2	21.8	13.2	21.9	16.5	3.6	
HSE	X	X	X	27.7	22.9	14.6	23.7	18.1	4.3	
	X	✓	X	32.3	26.7	14.6	23.7	18.1	4.7	
	✓	X	X	28.7	24.0	14.8	32.4	20.3	7.9	
	✓	X	✓	28.7	24.0	24.6	40.5	30.6	13.8	
	✓	✓	X	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	X	X	X	27.2	21.8	13.2	21.9	16.5	3.6	
	X	X	X	27.7	22.9	14.6	23.7	18.1	4.3	
HSE	X	✓	X	32.3	26.7	14.6	23.7	18.1	4.7	
	✓	X	X	28.7	24.0	14.8	32.4	20.3	7.9	
	✓	X	✓	28.7	24.0	24.6	40.5	30.6	13.8	
	✓	✓	X	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	X	X	X	27.2	21.8	13.2	21.9	16.5	3.6	
HSE	X	X	X	27.7	22.9	14.6	23.7	18.1	4.3	
	X	✓	X	32.3	26.7	14.6	23.7	18.1	4.7	
	✓	X	X	28.7	24.0	14.8	32.4	20.3	7.9	
	✓	X	✓	28.7	24.0	24.6	40.5	30.6	13.8	
	✓	✓	X	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	X	X	X	27.2	21.8	13.2	21.9	16.5	3.6	
	X	X	X	27.7	22.9	14.6	23.7	18.1	4.3	
HSE	X	✓	X	32.3	26.7	14.6	23.7	18.1	4.7	
	✓	X	X	28.7	24.0	14.8	32.4	20.3	7.9	
	✓	X	✓	28.7	24.0	24.6	40.5	30.6	13.8	
	✓	✓	X	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	X	X	X	27.2	21.8	13.2	21.9	16.5	3.6	
HSE	X	X	X	27.7	22.9	14.6	23.7	18.1	4.3	
	X	✓	X	32.3	26.7	14.6	23.7	18.1	4.7	
	✓	X	X	28.7	24.0	14.8	32.4	20.3	7.9	
	✓	X	✓	28.7	24.0	24.6	40.5	30.6	13.8	
	✓	✓	X	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**

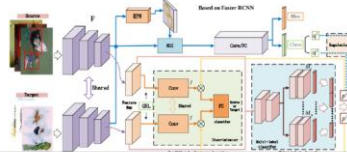
Qualitative Examples

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)

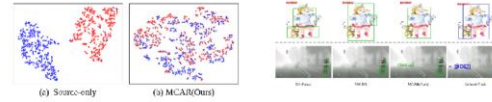
Category Prediction based Regularization

- 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 $z\bar{A}, z\bar{A}$.
- 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 $z\bar{A}, z\bar{A}$.
- Adacoseg : adaptive feature visualization with mutual regularization $\rightarrow p . zho et al .$
- 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



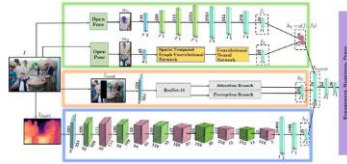
Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	CIDE	SPICE
Baseline	0.18	0.12	0.08	0.05	0.25	0.15	0.10
Adacoseg	0.21	0.14	0.10	0.07	0.28	0.18	0.12
Proposed	0.25	0.17	0.12	0.08	0.32	0.22	0.15

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) .
- $\rightarrow 2$ classification



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 ?

Data type	Dataset	Dataset Size	Agents	Location	Setting	Emotion Labels	Context
Image	EMOTIC [1]	18,116 images	34,320	Web	36 Categories	Yes	No
Image	AFSA [2]	40,000 images	40,000	Web	8 Categories	No	No
Image	CAHR [3]	70,000 images	70,000	TV Shows	7 Categories	No	No
Video	AFSA [2]	1,800 clips	1,800	Web	7 Categories	No	No
Video	CAHR [3]	13,201 clips	13,201	TV Shows	4 Categories	No	No
Video	EMOTIC [1]	12 hrs	12 hrs	TV Shows	4 Categories	Yes	No
Video	GroupWalk	45 clips (10 mins each)	344	Real Settings	4 Categories	Yes	Yes

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.

Labels	Kohri et al. [7]	Zhang et al. [21]	Lee et al. [15]	EmotCon GCN-Based	EmotCon Depth-Based
Affection	27.85	46.30	13.9	36.78	45.53
Anger	69.49	10.87	11.5	14.92	15.46
Amusement	14.06	11.23	16.4	18.45	21.05
Anticipation	38.04	62.64	33.05	38.15	74.17
Annoyance	67.48	5.93	16.2	16.48	17.81
Confidence	76.35	72.40	32.34	39.23	56.65
Disapproval	14.97	11.38	16.04	21.21	19.82
Disconnection	21.32	20.91	22.80	23.17	43.12
Disappointment	16.69	16.54	17.19	16.47	18.73
Doubt/Confusion	29.63	18.68	28.98	13.15	38.12
Embarrassment	18.18	1.94	13.68	11.25	14.97
Engagement	87.53	88.56	46.58	90.45	91.12
Enthusiasm	17.93	13.33	16.26	22.23	24.62
Excitement	77.16	71.30	35.26	82.21	83.26
Fatigue	69.70	13.26	13.04	19.18	16.23

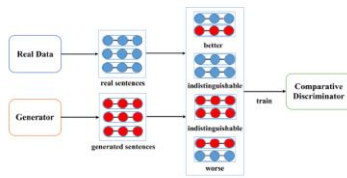
SELF-ADVERSARIAL LEARNING

- 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 ?
- $E (pz (z , m)$

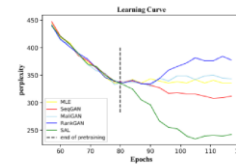


COMPARATIVE DISCRIMINATOR

- 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 .



Model	SIDE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	CIDE	SPICE
Baseline	0.18	0.12	0.08	0.05	0.25	0.15	0.10
Adacoseg	0.21	0.14	0.10	0.07	0.28	0.18	0.12
Proposed	0.25	0.17	0.12	0.08	0.32	0.22	0.15

Qualitative Examples

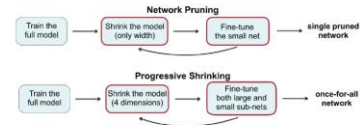
- TextFigure Module (w/o vs w/)

INTRODUCTION

- Decouple the model training stage and search stage
- Specialized sub-nets & sub-nets + sub-nets
- We extensively evaluated the effectiveness of ofa on imagenet
- How to deploy different hardware efficiency constraints ?
- We propose a progressive shrinking algorithm for once-for-all.

INTRODUCTION

- Decouple the model training stage and search stage
- Specialized sub-nets & sub-nets + sub-nets
- We extensively evaluated the effectiveness of ofa on imagenet
- How to deploy different hardware efficiency constraints ?
- We propose a progressive shrinking algorithm for once-for-all.

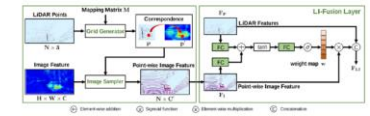


Method

- Epnnet : enhancing point features with image semantics
- Image feature in a point-wise manner .
- Li-fusion rpn effectively combines the lidar point feature
- We combine the point features si with the aid of our li-fusion module .

Method

- Epnnet : enhancing point features with image semantics
- Image feature in a point-wise manner .
- Li-fusion rpn effectively combines the lidar point feature
- We combine the point features si with the aid of our li-fusion module .



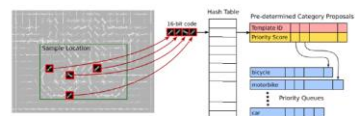
Priority Lists

- Each cell is linked to a list of templates
- Each root has a limited budget of locations
- How to balance proposals among all locations ?
- We propose a score adjustment process .

Method	Ours	Ours	Ours	FTVQ [2]	DPM V5 [3]
Frequency	100Hz	30Hz	15Hz	2Hz	0.07Hz
airplane	0.1630	0.2695	0.3029	0.3320	0.3318
bicycle	0.3563	0.5735	0.5946	0.5933	0.5878
bird	0.0021	0.0069	0.0059	0.1027	0.1019
boat	0.0303	0.0303	0.1141	0.1568	0.1801
bottle	0.0909	0.1938	0.2125	0.2694	0.2535
bus	0.2989	0.4139	0.4720	0.5129	0.5056
car	0.2505	0.4240	0.4996	0.5373	0.5271
cat	0.1368	0.1725	0.1931	0.2251	0.1904
chair	0.0909	0.0909	0.1053	0.2010	0.2046
cow	0.0069	0.1062	0.1194	0.2322	0.2414
diningtable	0.1743	0.2500	0.2510	0.2685	0.2750
dog	0.0507	0.1159	0.1159	0.1260	0.1238
horse	0.2724	0.4735	0.5539	0.5651	0.5709

Priority Lists

- Each cell is linked to a list of templates
- Each root has a limited budget of locations
- How to balance proposals among all locations ?
- We propose a score adjustment process .



Evaluation on Various Target Tasks

- Meta-networks trained by our meta-training scheme (meta-networks)
- The first source model is trained on tinymagenet .
- We use 34 - layer resnet as a source and target model , respectively .
- Ours needs only 50 samples per class

Evaluation on Various Target Tasks

- Meta-networks trained by our meta-training scheme (meta-networks)
- The first source model is trained on tinymagenet .
- We use 34 - layer resnet as a source and target model , respectively .
- Ours needs only 50 samples per class

Source task	TinyImageNet				ImageNet			
	CIFAR-100	STL-10	CUB200	MIT57	Stanford40	Stanford40	Stanford40	Stanford Dogs
Scratch	67.69 _{±0.02}	65.18 _{±0.01}	42.15 _{±0.01}	48.91 _{±0.01}	36.93 _{±0.01}	58.93 _{±0.01}	58.08 _{±0.01}	68.33 _{±0.01}
LoF	69.23 _{±0.01}	68.64 _{±0.01}	45.52 _{±0.01}	53.73 _{±0.01}	39.73 _{±0.01}	60.33 _{±0.01}	60.33 _{±0.01}	69.20 _{±0.01}
AT (one-to-one)	67.56 _{±0.01}	74.19 _{±0.01}	57.54 _{±0.01}	59.84 _{±0.01}	59.20 _{±0.01}	60.30 _{±0.01}	60.30 _{±0.01}	72.67 _{±0.01}
LoF+AT (one-to-one)	68.75 _{±0.01}	75.06 _{±0.01}	58.90 _{±0.01}	61.42 _{±0.01}	60.30 _{±0.01}	60.30 _{±0.01}	60.30 _{±0.01}	76.67 _{±0.01}
FM (single)	69.80 _{±0.01}	75.00 _{±0.01}	47.80 _{±0.01}	53.13 _{±0.01}	42.03 _{±0.01}	66.03 _{±0.01}	66.03 _{±0.01}	76.03 _{±0.01}
FM (one-to-one)	69.97 _{±0.01}	76.38 _{±0.01}	48.93 _{±0.01}	54.88 _{±0.01}	44.50 _{±0.01}	67.25 _{±0.01}	67.25 _{±0.01}	76.25 _{±0.01}
L2F w/ (single)	70.27 _{±0.01}	74.35 _{±0.01}	51.95 _{±0.01}	60.41 _{±0.01}	46.25 _{±0.01}	69.16 _{±0.01}	69.16 _{±0.01}	76.16 _{±0.01}
L2F w/ (one-to-one)	70.02 _{±0.01}	74.42 _{±0.01}	56.61 _{±0.01}	60.78 _{±0.01}	48.19 _{±0.01}	69.04 _{±0.01}	69.04 _{±0.01}	76.04 _{±0.01}
L2F w/ (all-to-all)	70.96_{±0.01}	78.31_{±0.01}	65.85_{±0.01}	64.85_{±0.01}	63.08_{±0.01}	76.08_{±0.01}	76.08_{±0.01}	76.08_{±0.01}

Qualitative Examples

- Paraphrasing Module (w/o vs w/)

Introduction

- Since the phrase representations are produced and attended at each encoder layer, the encoding of each layer is also enhanced with phrase-level attention computation;

Introduction

- The phrase representations are produced by phrase-level attention .

Related work

- Action proposals is an essential part of many methods for action detection, explored by a number of recent
- More related to our work, previous methods [9, 18, 24, 34, 35] explore the temporal order, either by predicting the exact order of consecutive frames [18, 35] or verifying their partial order [9, 24, 34].
- In the video domain, motion has been used as a cue for learning video representations in [1, 26, 33, 7].
- The notion of actionness was first introduced in [5] as a confidence measure of intentional bodily movement of biological agents.

Related work

- Action detection is a key tool for action detection
- Predicting the exact order of consecutive frames :
- Motion has been used as a cue .
- Actionness was first introduced in biological agents

Base Architecture

- The feed-forward layers capture the domainspecific and -independent information by using private output layers for each domain and one shared output layer.
- Word embeddings are derived from a combination GloVe (Pennington et al., 2014) and FastText (Bojanowski et al., 2017) pre-trained word embeddings, as used in (Ma and Hovy, 2016).
- The global objective function is the combination of the NER loss function and domain loss:
- The domain classification objective is to minimize the crossentropy loss $L_{domain}(x, y_d)$ for an input x with domain label y_d .
- We propose a new architecture based on the BiLSTM₂BiCRF model tailored to the three proposed experimental setups.

Base Architecture

- Use private output layers for each domain
- (ma et al . , 2017)
- The global objective function is the combination of the ner loss
- The domain classification objective is to minimize the crossentropy loss .
- We propose a new architecture based on the bilstm model .

Conclusions

- In this paper, we proposed two parameterized benchmark games in which EFCE exhibits interesting behaviors.
- We also provided an alternative saddle-point formulation of EFCE and demonstrated its merit with a simple subgradient method which outperforms standard LP based methods.
- We analyzed those behaviors both qualitatively and quantitatively, and isolated two ways through which a mediator is able to compel the agents to follow the recommendations.
- We hope that our analysis will bring attention to some of the computational and practical uses of EFCE, and that our benchmark games will be useful for evaluating future algorithms for computing EFCE in large games.

Conclusions

- We proposed two parameterized games in which efce exhibits interesting behaviors .
- We propose a saddle-point formulation of efce
- Two ways to compel the agents to follow the recommendations
- We show that our analysis will be useful for computing efce in large games

Qualitative Examples

- Applying Design Ideas

Introduction

- Weak supervision in text classification has the burden of human experts
- How to train a deep neural network?
- We have performed experiments on real-word datasets
- Identify words that are discriminative and highly label-indicative

Introduction

Weak supervision in text classification has the burden of human experts

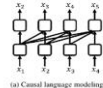
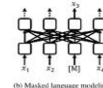
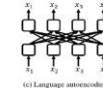
How to train a deep neural network?

We have performed experiments on real-word datasets

Identify words that are discriminative and highly label-indicative

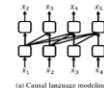
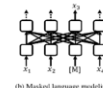
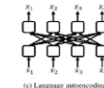
Language Model Baselines

- Feeding the token in the input sequence
- We can obtain the contextualized embeddings


Language Model Baselines

- Feeding the token in the input sequence
- We can obtain the contextualized embeddings






Face photo to drawing generator G

- No pairings need to exist between two domains
- Require the inverse generator to reconstruct a face photo
- A strict loss function for cycle-consistency loss
- $P(p, s)$



Face photo to drawing generator G



- No pairings need to exist between two domains
- Require the inverse generator to reconstruct a face photo
- A strict loss function for cycle-consistency loss
- $P(p, s)$

Detection Results on CityPersons

- Ablation study of PBM
- Breaking the curse of many agents with events

Method	PPFE	R2NMS	R	HO
Baseline	-	-	13.8	59.0
PBM	concat	-	12.5	57.3
PBM	concat	✓	12.1	57.0
PBM	mask	-	12.3	54.9
PBM	mask	✓	11.1	53.3

Detection Results on CityPersons

- Ablation study of PBM
- Breaking the curse of many agents with events

Method	PPFE	R2NMS	R	HO
Baseline	-	-	13.8	59.0
PBM	concat	-	12.5	57.3
PBM	concat	✓	12.1	57.0
PBM	mask	-	12.3	54.9
PBM	mask	✓	11.1	53.3

Multi-task Learning with Self-supervision

- Depending on the type of training samples, the statistical characteristics of the augmented training samples
- Remove unnecessary invariant property of the classifier
- Aggregate the corresponding conditional probabilities to improve the classification accuracy

Dataset	Baseline	DA	MT	SLA+S1
CIFAR10	92.39	90.44	90.79	92.50
CIFAR100	68.27	65.73	66.10	68.68
tmp-ImageNet	63.11	60.21	58.04	63.99

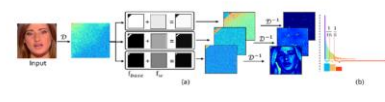
Multi-task Learning with Self-supervision

- Depending on the type of training samples, the statistical characteristics of the augmented training samples
- Remove unnecessary invariant property of the classifier
- Aggregate the corresponding conditional probabilities to improve the classification accuracy

Dataset	Baseline	DA	MT	SLA+S1
CIFAR10	92.39	90.44	90.79	92.50
CIFAR100	68.27	65.73	66.10	68.68
tmp-ImageNet	63.11	60.21	58.04	63.99

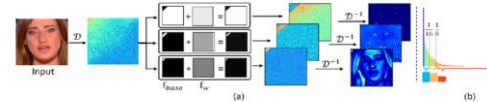
FAD Frequency-Aware Decomposition

- The frequency of frequency-aware components can be inversely transformed
- Number of filters (n=512)
- The frequency filtering is a special case of the input image



FAD Frequency-Aware Decomposition

- The frequency of frequency-aware components can be inversely transformed
- Number of filters (n=512)
- The frequency filtering is a special case of the input image



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!

