Multilingual/Multimodal Pre-training
多语言/多模态预训练

Nan DUAN (段楠)
Principal Researcher
Microsoft Research Asia
2020-10-16

Microsoft

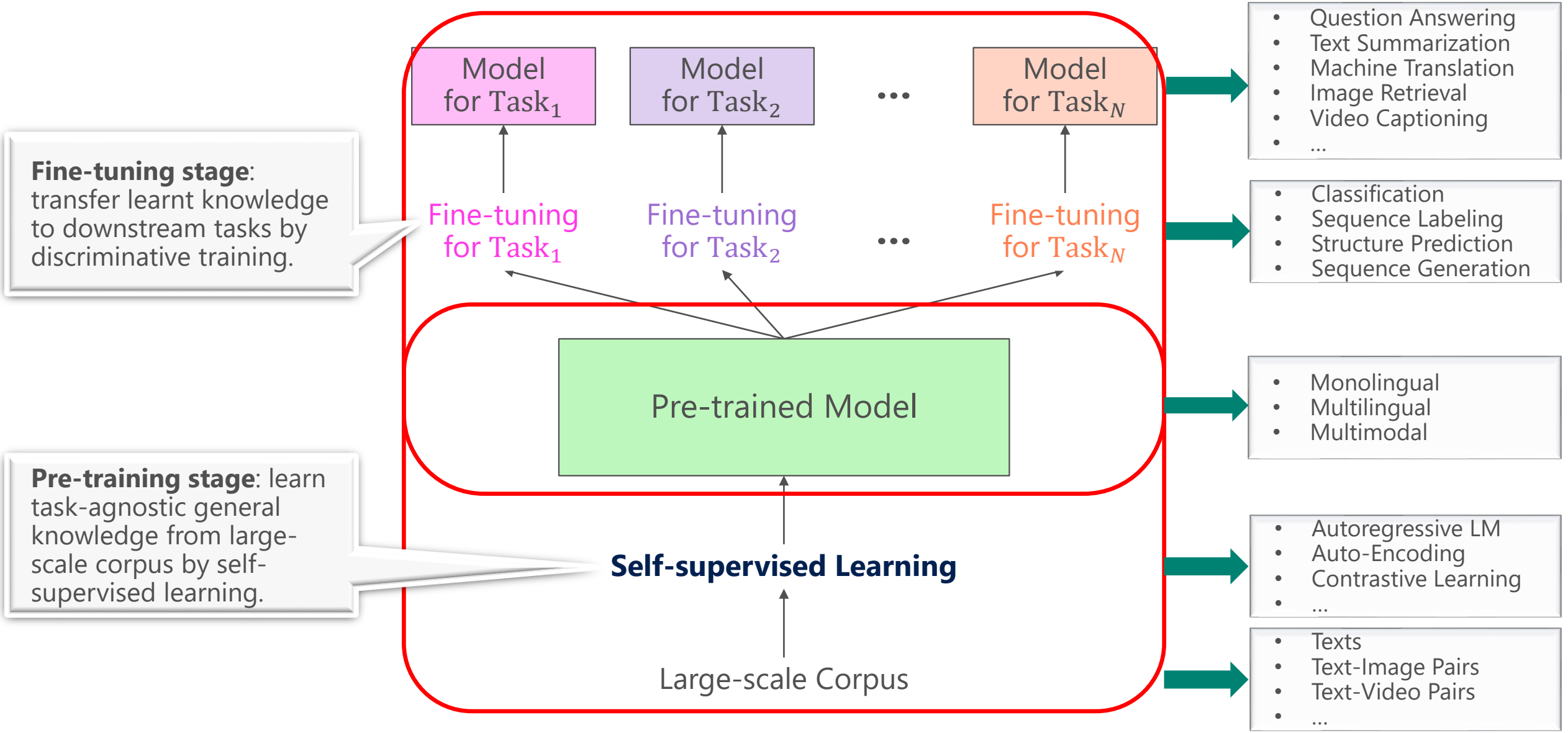
Outline

- **Background**
- **Multilingual/Multimodal Pre-trained Models**
  - Unicoder (for multilingual language tasks)
  - Unicoder-VL (for image-language tasks)
  - Unicoder-VL (for video-language tasks)
- **From Natural Language to Programming Language**
- **Summary & Future Work**

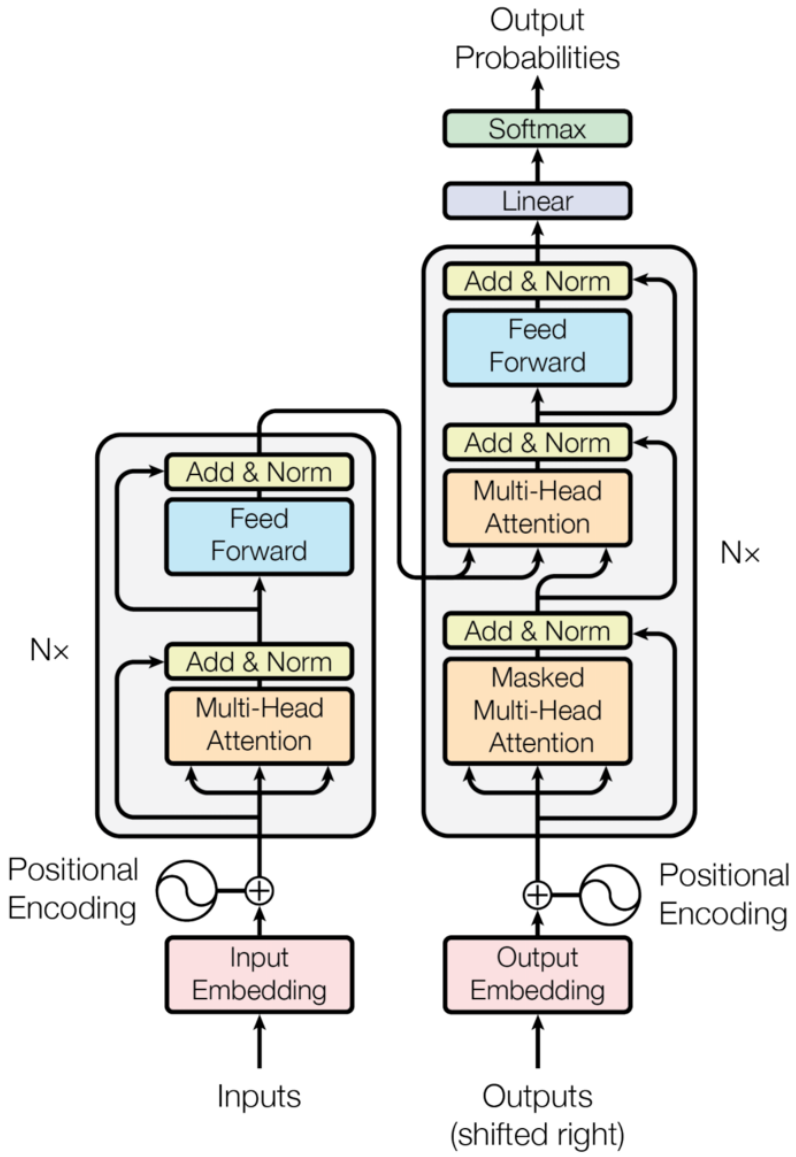
Outline

- **Background**
- **Multilingual/Multimodal Pre-trained Models**
  - Unicoder (for multilingual language tasks)
  - Unicoder-VL (for image-language tasks)
  - Unicoder-VL (for video-language tasks)
- **From Natural Language to Programming Language**
- **Summary & Future Work**

Pre-training + Fine-tuning is a new NLP paradigm.



Key Technologies (1): Transformer as Backbone



Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. **Attention Is All You Need**. NeurIPS, 2017.

Key Technologies (2): Pre-training by Self-supervised Learning

processing

LM is a typical task in natural language ___

natural

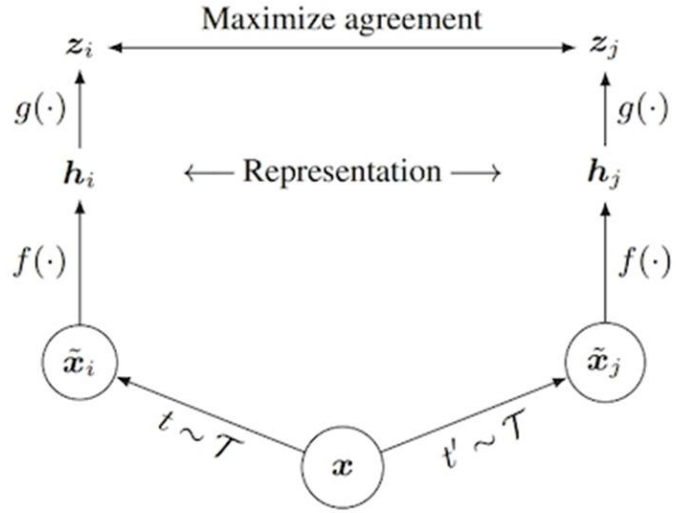
LM is a typical task in ___ language processing

(a) word-level

LM is a typical task in natural language processing

LM is a ~~typical~~ task in processing ___ language

(b) sentence-level



Autoregressive (AR) LM

Auto-encoding (AE)

Contrastive Learning

Self-supervised learning is a form of unsupervised learning where the data itself provides the supervision.

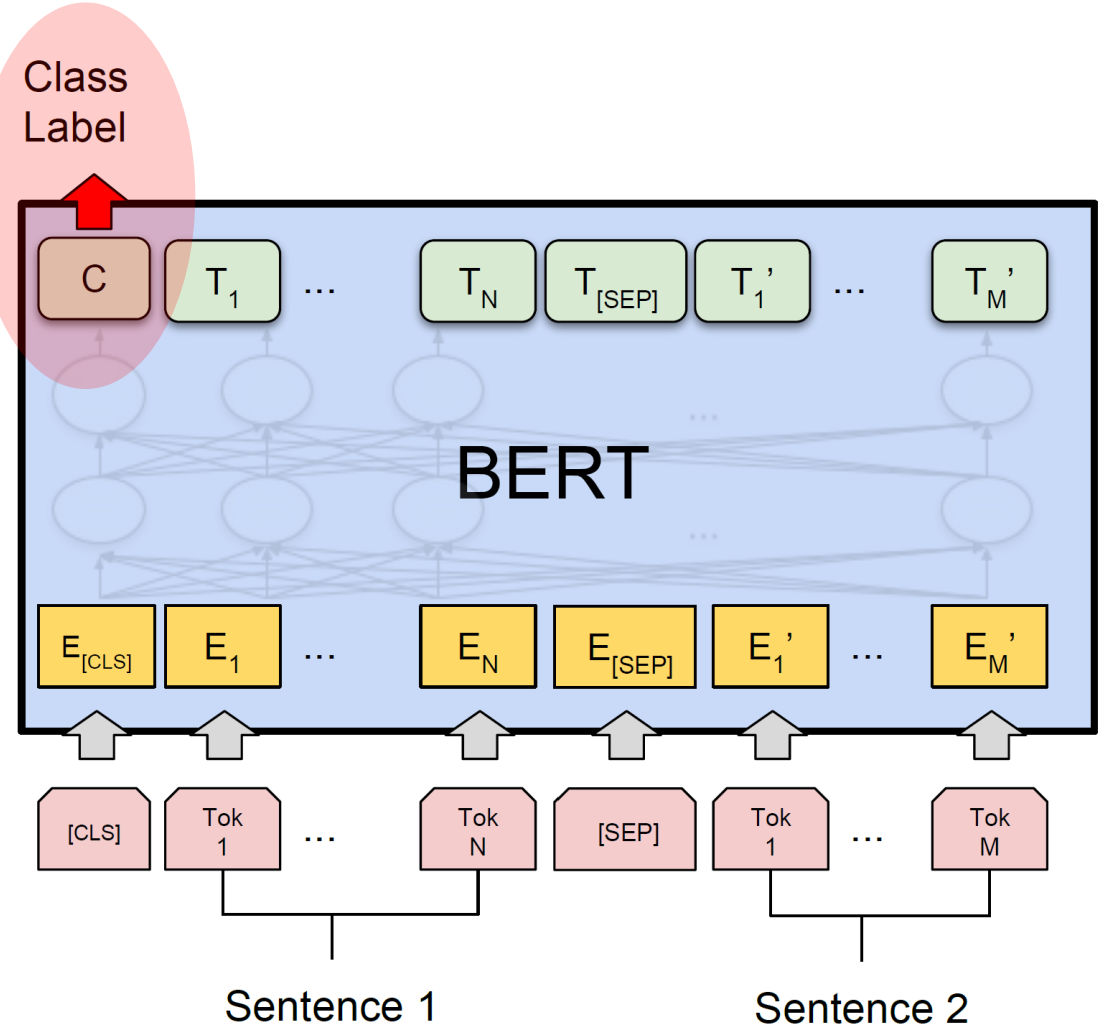
Key Technologies (3): Fine-tuning by Discriminative Training

(take **BERT-based Sentence Pair Matching** as an example)

Given the final hidden vector $C \in \mathbb{R}^H$ of the first input token ([CLS]), fine-tune BERT by a standard classification loss with C and W:

$$\log(\text{softmax}(C\boldsymbol{W}^{\mathrm{T}}))$$

where $\boldsymbol{W} \in \mathbb{R}^{K \times H}$ is a classification layer, K is the number of labels.



Outline

- Background
- **Multilingual/Multimodal Pre-trained Models**
  - Unicoder (for multilingual language tasks)
  - Unicoder-VL (for image-language tasks)
  - Unicoder-VL (for video-language tasks)
- From Natural Language to Programming Language
- Summary & Future Work

Unicoder

an example example 例子 True True

Unicoder

(12 layers, 256K vocabulary size, 100 languages)

this is [MASK] [MASK]	this is an [MASK] 这 是 一个 [MASK]	this is an example 这 是 一个 例子	
Masked LM (Word/Phrase-level)	**Translation LM** (Word/Phrase-level)	**Contrastive Learning** (Word/Phrase/Sentence-level)	**Knowledge Regularization** (Entity/Relation)
• this is an example	• this is an example • 这 是 一个 例子	• this is an example • 这 是 一个 例子 • 0-0 1-1 2-2 3-3	• Bill Gates founded MS

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Ming Zhou. **Unicoder: A Universal Language Encoder by Pre-training with Multiple Cross-lingual Tasks**. EMNLP, 2019.

XGLUE

Home Intro **Leaderboard** Contact

XGLUE Dataset and Leaderboard

Tasks

1. NER
2. POS Tagging (POS)
3. News Classification (NC)
4. MLQA
5. XNLI
6. PAWS-X
7. Query-Ad Matching (QADSM)
8. Web Page Ranking (WPR)
9. QA Matching (QAM)
10. Question Generation (QG)
11. News Title Generation (NTG)

New Tasks!

Relevant Links

XGLUE Submission Guideline/Github | XGLUE Paper | Unicoder Paper(Baseline)

Leaderboard (05/25/2020-Present) ranked by XGLUE Score (average score on 11 tasks)

Rank	Model	Submission Date	NER	POS	NC	MLQA	XNLI	PAWS-X	QADSM	WPR	QAM	QG	NTG	XGLUE Score
1	**Unicoder Baseline** (XGLUE Team)	2020-05-25	79.7	79.6	83.5	66.0	75.3	90.1	68.4	73.9	68.9	10.6	10.7	64.2

https://microsoft.github.io/XGLUE/

Unicoder scaled Bing intelligent question answering to 100 languages and 200 regions in the world.

Bing blogs

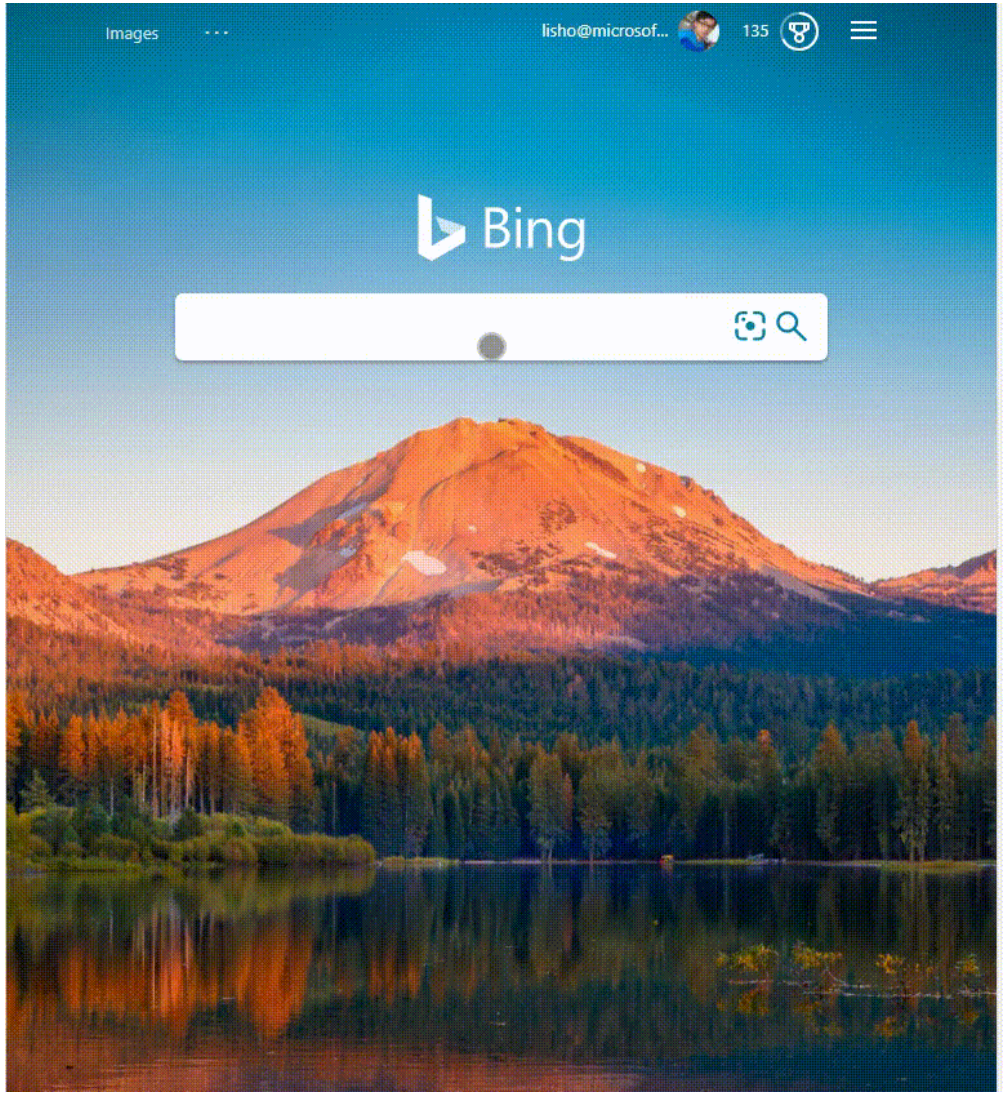
OCTOBER 1 2020

Bing Releases Intelligent Question-Answering Feature to 100+ Languages

Intelligent question-answering is one of the most useful and delightful features of search. As a user, you ask a question (e.g., "what are the benefits of eating apricots") and can get the answer directly (e.g., info about health and nutrition benefits of apricots) at the top of the page without further need to search for relevant content by yourself. The feature aims to direct users to the most concise and precise answers from web documents, thus saving users time and efforts.

English-language question answering from web has been enabled on Bing for several years, and another dozen of languages, like French and German, have been added within the last year. But our work isn't done - there are thousands of languages in the world! Not all of them have rich enough web content to derive good answers, but for those that do, uses of those spoken languages deserve the same useful, delightful, time-saving experience.

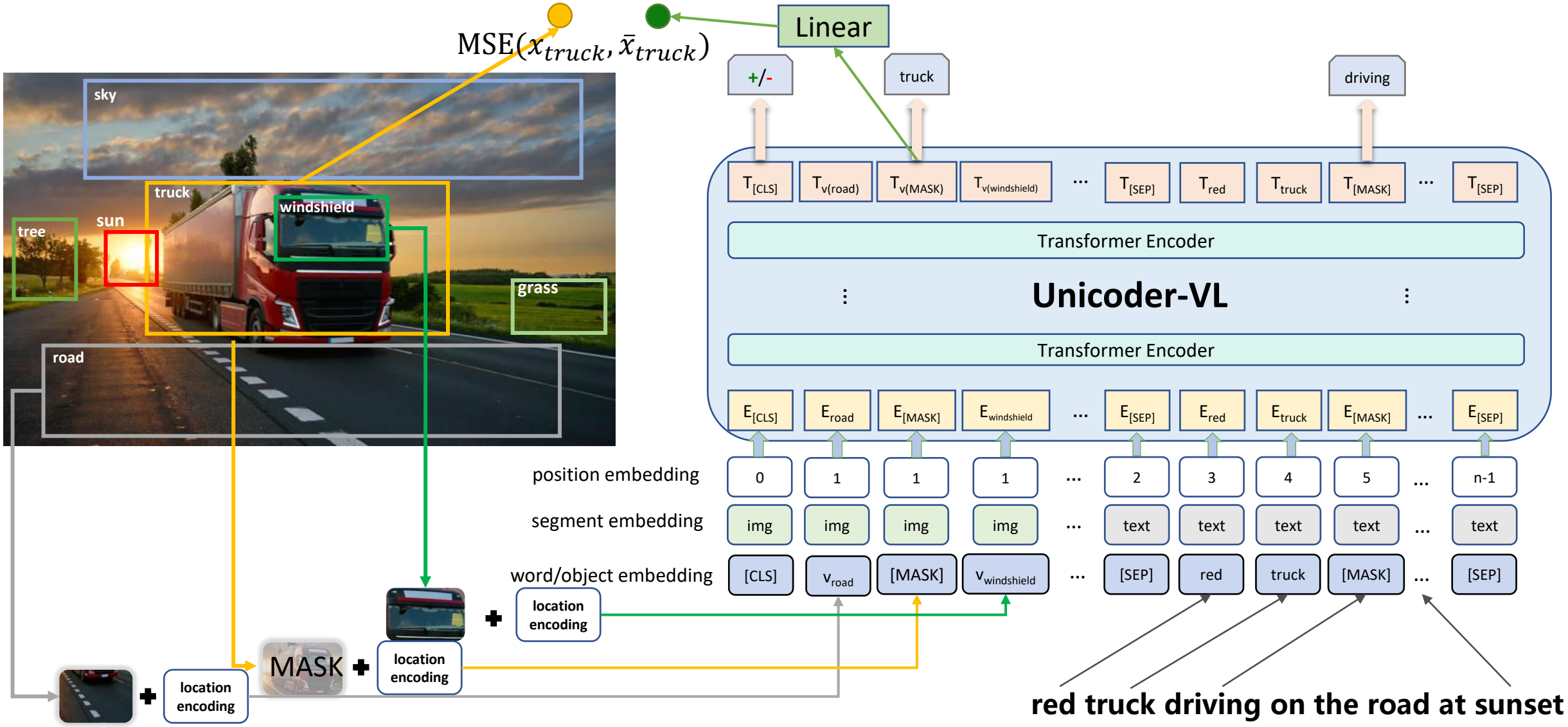
Recently, Bing expanded its intelligent question-answering feature to more than 100 languages, making AI and Bing itself more inclusive and accessible. What is amazing is this is achieved by using a language agnostic approach. In other words, the AI model generating the intelligent question-answering in Urdu is the same one generating the intelligent question-answering in Romanian. Here are some examples of this experience in various languages (if you speak a language other than English, feel free to give it a try, but be reminded to set your browser to the relevant language):



Outline

- Background
- **Multilingual/Multimodal Pre-trained Models**
  - Unicoder (for multilingual language tasks)
  - Unicoder-VL (for image-language tasks)
  - Unicoder-VL (for video-language tasks)
- From Natural Language to Programming Language
- Summary & Future Work

Unicoder-VL for Image-Language Tasks



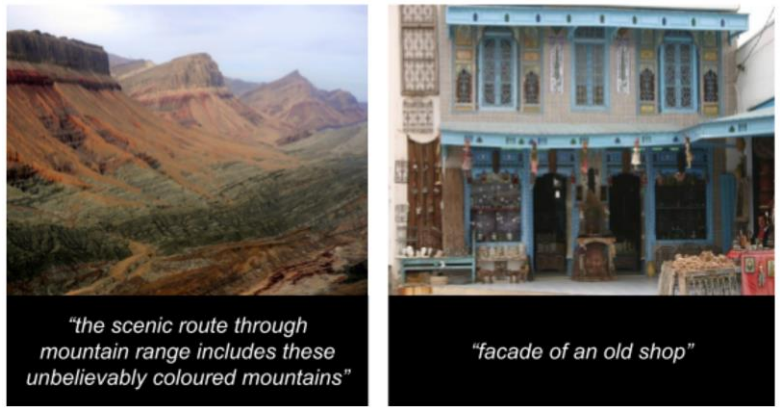
Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, Ming Zhou. **Unicoder-VL: A Universal Encoder for Vision and Language by Cross-modal Pre-training**. AAAI 2020.

Evaluation Results: Image-Text Retrieval

Model	Text-to-Image Retrieval (Flickr30k)			Image-to-Text Retrieval (Flickr30k)		
	R@1	R@5	R@10	R@1	R@5	R@10
ViLBERT (Lu et al., 2019)	58.2	84.9	91.5	-	-	-
UNITER (Chen et al., 2019)	71.5	91.2	95.2	84.7	97.1	**99.0**
Unicoder-VL (Li et al., 2020)	**73.1**	**92.3**	**95.9**	**88.0**	**97.3**	98.6

Model	Text-to-Image Retrieval (MSCOCO)			Image-to-Text Retrieval (MSCOCO)		
	R@1	R@5	R@10	R@1	R@5	R@10
UNITER (Chen et al., 2019)	48.4	76.7	85.9	63.3	87.0	93.1
Unicoder-VL (Li et al., 2020)	**50.5**	**78.7**	**87.1**	**66.4**	**89.8**	**94.4**

- Pre-training dataset
  - 3,318,333 image-caption pairs from Google's Conceptual Captions



Evaluation Results: Visual QA & Reasoning (GQA)

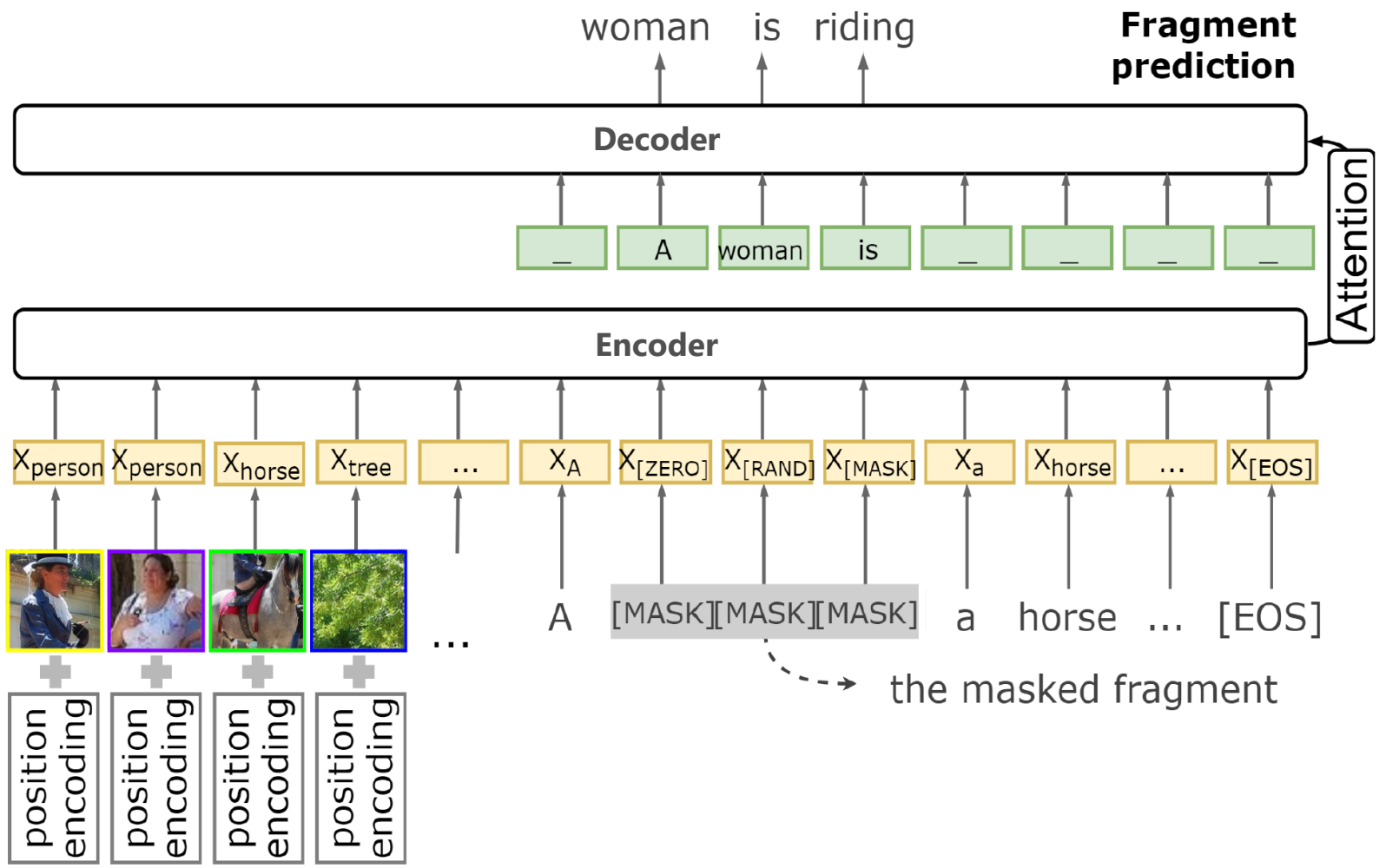
What color is the food on the red object left of the small girl that is holding a hamburger, yellow or green?

Rank	Participant team	Binary	Open	Consistency	Plausibility	Validity	Distribution	Accuracy	Last submission at
1	Human Performance (human)	91.20	87.40	98.40	97.20	98.90	0.00	89.30	2 years ago
2	DREAM+Unicoder-VL (MSRA)	84.46	68.60	91.47	83.75	96.42	3.68	76.04	1 year ago
3	TRRNet (Ensemble)	82.12	66.89	89.00	83.58	96.76	1.29	74.03	8 months ago
4	MIL-nbgao	80.80	67.64	91.76	83.90	96.73	1.70	73.81	24 days ago
5	Kakao Brain	79.68	67.73	77.02	83.70	96.36	2.46	73.33	1 year ago
6	AIOZ (Coarse-to-Fine Reasoning, Sing)	81.16	64.19	90.96	84.81	96.77	2.39	72.14	10 months ago
7	270	77.50	63.82	86.94	83.77	96.65	1.49	70.23	1 year ago
8	NSM ensemble (updated)	80.45	56.16	93.83	84.16	96.53	2.78	67.55	1 year ago
9	TRRNet (Single)	77.91	50.22	89.84	85.15	96.47	5.25	63.20	7 months ago
10	NSM single (updated)	78.94	49.25	93.25	84.28	96.41	3.71	63.17	1 year ago

1/95

https://evalai.cloudcv.org/web/challenges/challenge-page/225/leaderboard/733

Extend Unicoder-VL to Image Captioning

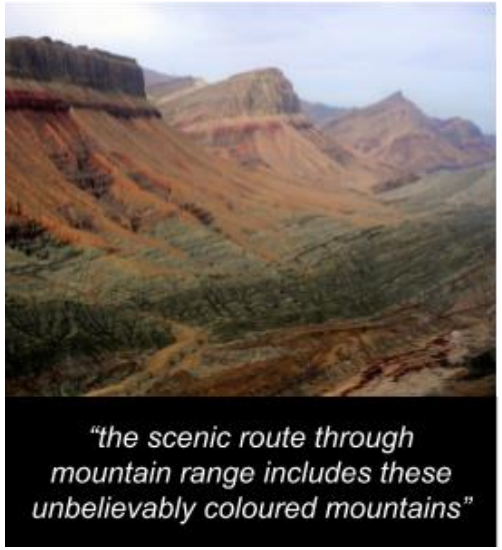


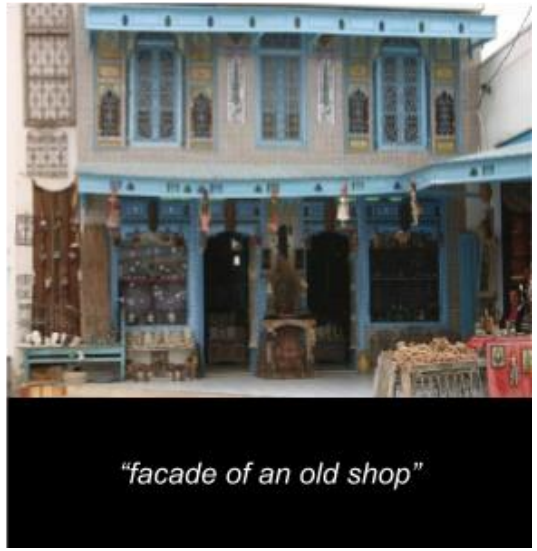
Qiaolin Xia, Haoyang Huang, Nan Duan, Dongdong Zhang, Lei Ji, Zhifang Sui, Edward Cui, Taroon Bharti, Xin Liu, Ming Zhou. **XGPT: Cross-modal Generative Pre-Training for Image Captioning**. arXiv 2020.

Evaluation Results: Image Captioning

- Pre-trained with Conceptual Captions dataset
  - ~3.3M images annotated with captions
- Evaluated on MSCOCO dataset

Methods	Image Caption			
	BLEU@4	METEOR	CIDEr	SPICe
BUTD (Anderson et al. 2018)	36.2	27.0	113.5	20.3
NBT (Lu et al. 2018)	34.7	27.1	107.2	20.1
VLP (Zhou et al. 2018)	36.5	28.4	116.9	20.8
Unicoder-VL (Huang et al., 2020)	**37.2**	**28.6**	**120.1**	**21.8**





Extend Unicoder-VL to Multilingual Scenarios (a.k.a. M^3P)

Pre-training Overview

Multilingual Image-Text Retrieval

Multilingual Image Captioning

Multimodal Machine Translation

Multitask Multilingual Multimodal Pre-trained Model
($\mathbf{M^3P}$ for 100 languages)

Multilingual Pre-trained Model
(Unicoder for 100 languages)

- masked language model
- translation language model
- text denoising auto-encoding

- Multilingual corpus (text only)
- Bilingual corpus (text only)

Vision-Language Pre-trained Model
(Unicoder-VL for English)

- masked token loss
- contrastive loss
- image captioning

- Image-Text (English) pairs

(Research) Evaluation Datasets

En - Two cars are racing on a track while the audience watches from behind a fence
De - Zwei Rennautos fahren auf der Restricken in die Kurve (Tr: Two race cars drive on the race track in the curve)
Fr – Deux voitures roulent sur un circuit. (Tr: Two race cars drive on the race track in the curve)
Cs– Dvě auta jedou po závodní dráze (Tr: Two cars ride the race track)

Multi30K dataset (en/de/fr/cs):

- 31,783 images in total
- 5 captions per image in **English (en)** and **German (de)**
- 1 caption per image in **French (fr)** and **Czech (cs)**

En - A young man playing frisbee in a grassy park

Cn - 两个男人在公园的草地上跳起来接飞盘
(Tr: Two men jump on the grass in the park and pick up the Frisbee)

Ja - 芝生の上で女性がフリスビーで遊んでいます
(Tr: A woman is playing frisbee on the grass)

MSCOCO dataset (en/ja/zh):

- 123,287 images in total
- 5 captions per image in **English (en)** and **Japanese (ja)**
- 1~2 captions per image in **Chinese (zh)**

Haoyang Huang, Lin Su, Di Qi, Nan Duan and Others. **M3P: Learning Universal Representations via Multitask Multilingual Multimodal Pre-training**. arXiv 2020.

Evaluation Results

Task	Multilingual Image-Text Retrieval (Multi30K + MSCOCO)						Multilingual Image Captioning (Multi30K + MSCOCO)						Multimodal MT (Multi30K)	
	en	de	fr	cs	ja	zh	en	de	fr	cs	ja	zh	en→fr	en→de
SoTA	**92.7**	72.1	65.9	64.8	76.0	74.8	**37.4**	3.8	5.0	2.8	38.5	36.7	53.8	31.6
M^3P_B	88.0	**82.0**	**73.5**	**70.2**	**86.8**	**81.8**	34.7	**16.6**	**8.7**	**5.4**	**40.2**	**39.7**	**55.5**	**35.7**
Δ	4.7 ↓	9.9 ↑	7.6 ↑	5.4 ↑	10.8 ↑	7.0 ↑	3.7 ↓	12.8 ↑	3.7 ↑	2.6 ↑	1.7 ↑	3.0 ↑	1.7 ↑	4.1 ↑

Blue numbers indicates the best result for a task. For retrieval tasks, we use *mean Recall* as the metric, which is an average score of R@1, R@5 and R@10 on i2t and t2i tasks. For captioning and translation tasks, we use *BLEU-4* as the metric.

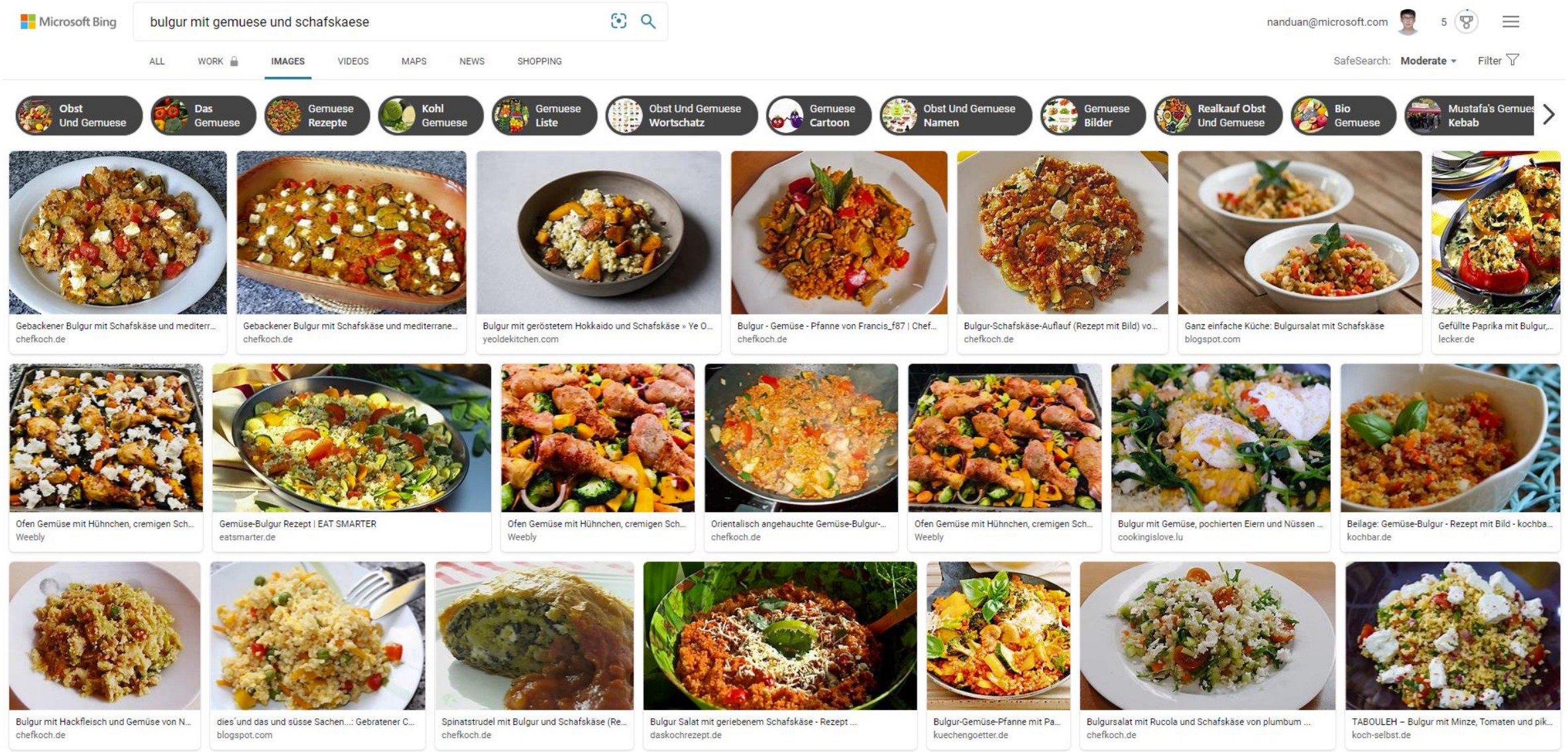
image caption output (**zh**): 一辆载着人和纸糊的房子的卡车行驶在街道上
(translation: a truck carrying people and paper houses travels down the street)

image caption input (**en**): A Boston Terrier is running on lush green grass in front of a white fence.

caption translation output (**fr**): Le Boston Terrier court sur l'herbe verte luxurie devant une clôture blanche.
(translation: The Boston Terrier runs on lush green grass in front of a white fence.)

caption translation output (**de**): Ein Hund läuft auf grünem Rasen vor einem weißen Zaun.
(translation: A dog runs on green grass in front of a white fence.)

Multilingual Unicoder-VL scaled Bing image search to 8 top-tier languages and 17 markets.

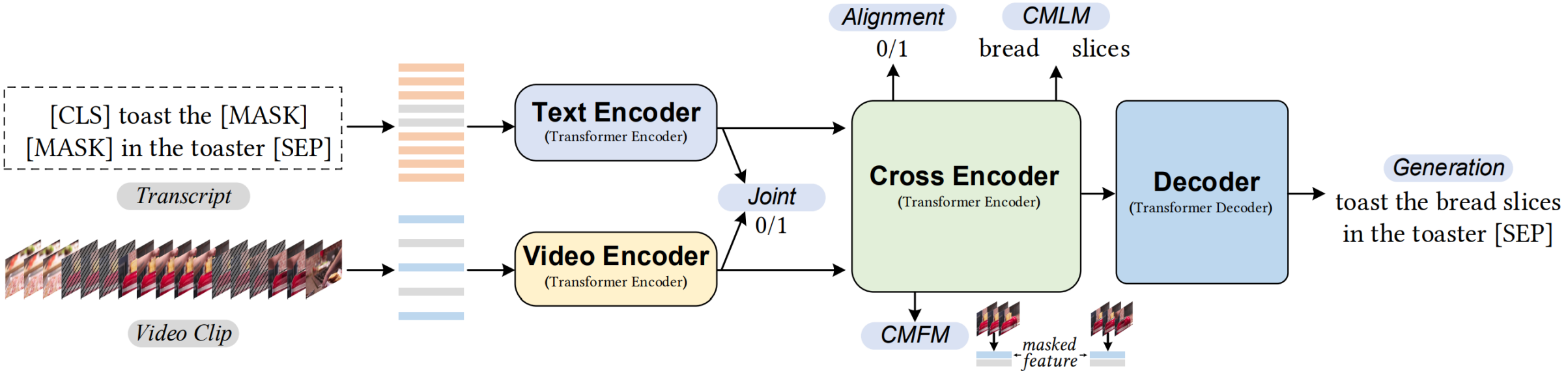


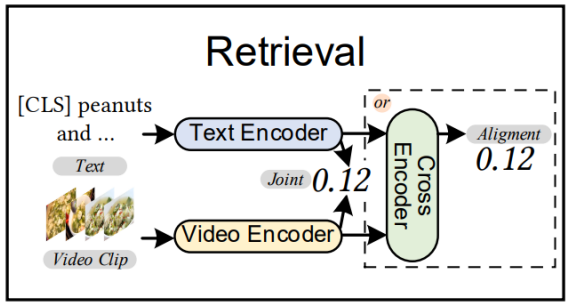
Outline

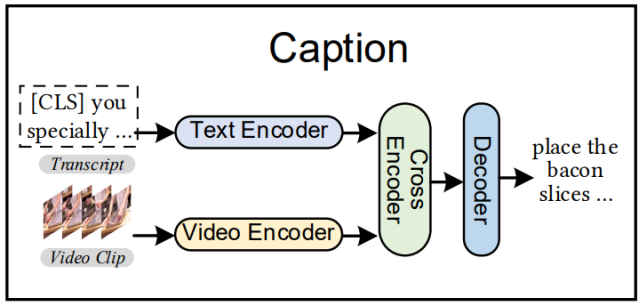
- Background
- **Multilingual/Multimodal Pre-trained Models**
  - Unicoder (for multilingual language tasks)
  - Unicoder-VL (for image-language tasks)
  - Unicoder-VL (for video-language tasks)
- From Natural Language to Programming Language
- Summary & Future Work

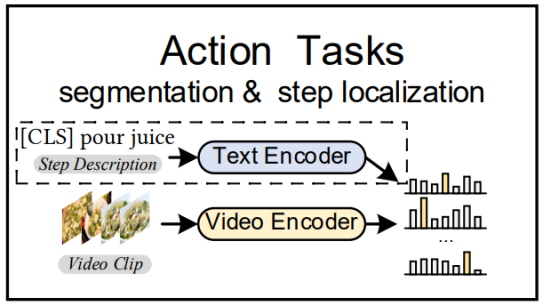
Unicoder-VL for Video-Language Tasks

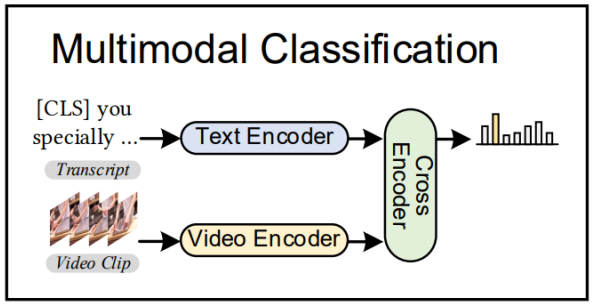
1. Video-Text Joint Embedding **2.** Video-Text Alignment **3.** Masked Frame Model **4.** Masked Language Model **5.** Caption Generation











Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, Ming Zhou. **UniVL: A Unified Video and Language Pre-Training Model for Multimodal Understanding and Generation**. arXiv 2020.

Pre-training Tasks

- Video-Text Joint

$$\mathcal{L}_{Joint}(\theta) = -E_{(\mathbf{t},\mathbf{v})\sim\mathbf{B}} \log \text{MIL-NCE}\,(\mathbf{t},\mathbf{v})$$

$$\text{MIL-NCE}\,(\mathbf{t},\mathbf{v}) = \frac{\sum_{(\hat{\mathbf{v}},\hat{\mathbf{t}})\in\mathcal{P}_{\mathbf{v},\mathbf{t}}} \exp(\hat{\mathbf{v}}\hat{\mathbf{t}}^\top)}{\mathcal{Z}}$$

$$\mathcal{Z} = \sum_{(\hat{\mathbf{v}},\hat{\mathbf{t}})\in\mathcal{P}_{\mathbf{v},\mathbf{t}}} \exp(\hat{\mathbf{v}}\hat{\mathbf{t}}^\top) + \sum_{(\tilde{\mathbf{v}},\tilde{\mathbf{t}})\in\mathcal{N}_{\mathbf{v},\mathbf{t}}} \exp(\tilde{\mathbf{v}}\tilde{\mathbf{t}}^\top)$$

- Conditioned Masked Language Model (CMLM)

$$\mathcal{L}_{CMLM}(\theta) = -E_{t_m\sim\mathbf{t}} \log P_\theta\,(t_m \mid t_{\neg m}, \mathbf{v})$$

- Conditioned Masked Frame Model (CMFM)

$$\mathcal{L}_{CMFM}(\theta) = -E_{v_m\sim\mathbf{v}} \log \text{NCE}\,(v_m \mid v_{\neg m}, \mathbf{t})$$

$$\text{NCE}\,(v_m \mid v_{\neg m}, \mathbf{t}) = \frac{\exp(\mathbf{f}_{v_m}\mathbf{m}_{v_m}^\top)}{\mathcal{Z}}$$

$$\mathcal{Z} = \exp(\mathbf{f}_{v_m}\mathbf{m}_{v_m}^\top) + \sum_{v_j\in\mathcal{N}(v_m)} \exp(\mathbf{f}_{v_m}\mathbf{m}_{v_j}^\top)$$

- Video-Text Alignment

$$\mathcal{L}_{Align}(\theta) = -E_{(\mathbf{t},\mathbf{v})\sim\mathbf{B}} \log \frac{\exp\big(s(\mathbf{t},\mathbf{v})\big)}{\mathcal{Z}}$$

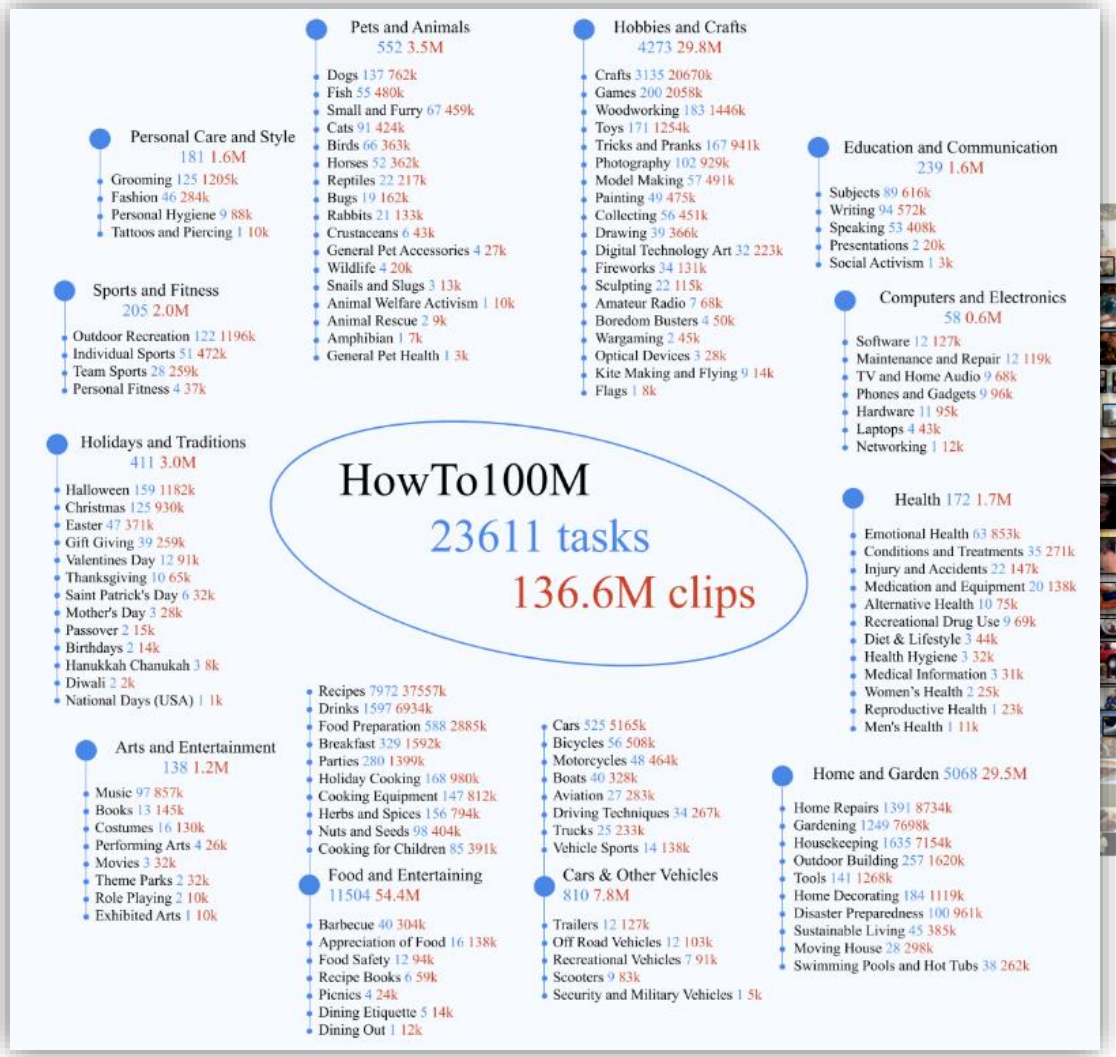
$$\mathcal{Z} = \exp\big(s(\mathbf{t},\mathbf{v})\big) + \sum_{\mathbf{u}\in\mathcal{N}(\mathbf{v})} \exp\big(s(\mathbf{t},\mathbf{u})\big)$$

- Video Caption Generation

$$\mathcal{L}_{Decoder}(\theta) = -E_{\hat{t}_i\sim\hat{\mathbf{t}}} \log P_\theta\,\big(\hat{t}_i \mid \hat{t}_{<i}, \mathbf{t}, \mathbf{v}\big)$$

Pre-training Corpus

HowTo100M (Miech et al., 2019): 136M video clips with captions from 1.2M Youtube videos.



Evaluation Results: Text-based Video Retrieval

- **MSR-VTT** (Xe et al., 2016): 200K clip-text pairs from 10K videos in 20 categories
- **YouCook2** (Zhou et al., 2018): 14k clip-text pairs from 2k videos.

Input: **Query:** cook a pizza
Video:

Output: Yes

Methods	R@1	R@5	R@10	Median R
Random	0.03	0.15	0.3	1675
HGLMM (Klein et al., 2015)	4.6	14.3	21.6	75
HowTo100M (Miech et al., 2019)	8.2	24.5	35.3	24
MIL-NCE (Miech et al., 2020)	15.1	38.0	51.2	10
ActBERT (Zhu and Yang, 2020)	9.6	26.7	38.0	19
VideoAsMT (Korbar et al., 2020)	11.6	-	43.9	-
UniVL (FT-Joint)	22.2	52.2	66.2	5
UniVL (FT-Align)	**28.9**	**57.6**	**70.0**	**4**

Table 1: Results of text-based video retrieval on Youcook2 dataset.

Methods	R@1	R@5	R@10	Median R
Random	0.1	0.5	1.0	500
C+LSTM+SA (Torabi et al., 2016)	4.2	12.9	19.9	55
VSE (Kiros et al., 2014)	3.8	12.7	17.1	66
SNUVL (Yu et al., 2016)	3.5	15.9	23.8	44
Kaufman et al. (2017)	4.7	16.6	24.1	41
CT-SAN (Yu et al., 2017)	4.4	16.6	22.3	35
JSFusion (Yu et al., 2018)	10.2	31.2	43.2	13
HowTo100M (Miech et al., 2019)	14.9	40.2	52.8	9
MIL-NCE (Miech et al., 2020)	9.9	24.0	32.4	29.5
ActBERT (Zhu and Yang, 2020)	8.6	23.4	33.1	36
VideoAsMT (Korbar et al., 2020)	14.7	-	52.8	-
UniVL (FT-Joint)	20.6	49.1	62.9	6
UniVL (FT-Align)	**21.2**	**49.6**	**63.1**	**6**

Table 2: Results of text-based video retrieval on MSR-VTT dataset.

Evaluation Results: Video Captioning

- **YouCook2** (Zhou et al., 2018): 14k clip-text pairs from 2k videos.

Methods	Input	B-3	B-4	M	R-L	CIDEr
Bi-LSTM (Zhou et al., 2018a)	V	-	0.87	8.15	-	-
EMT (Zhou et al., 2018b)	V	-	4.38	11.55	27.44	0.38
VideoBERT (Sun et al., 2019b)	V	6.80	4.04	11.01	27.50	0.49
CBT (Sun et al., 2019a)	V	-	5.12	12.97	30.44	0.64
ActBERT (Zhu and Yang, 2020)	V	8.66	5.41	13.30	30.56	0.65
VideoAsMT (Korbar et al., 2020)	V	-	5.3	13.4	-	-
AT (Hessel et al., 2019)	T	-	8.55	16.93	35.54	1.06
DPC (Shi et al., 2019)	V + T	7.60	2.76	18.08	-	-
AT+Video (Hessel et al., 2019)	V + T	-	9.01	17.77	36.65	1.12
UniVL	V	16.46	11.17	17.57	40.09	1.27
UniVL	T	20.32	14.70	19.39	41.10	1.51
UniVL	V + T	**23.87**	**17.35**	**22.35**	**46.52**	**1.81**

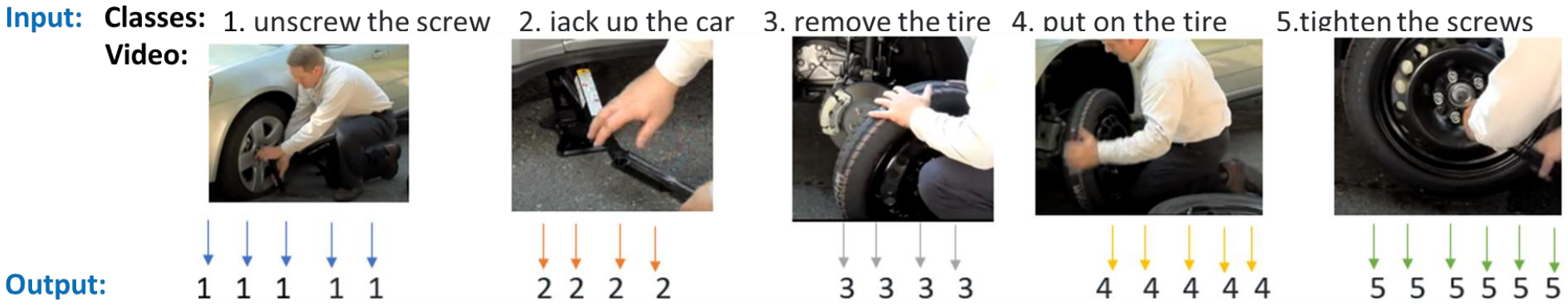
Table 3: The multimodal video captioning results on Youcook2 dataset. 'V' means video and 'T' means Transcript.

Input:

Output: sprinkle some cheese on top of pizza and bake them in the oven

Evaluation Results: Frame-wise Action Classification

- **COIN** (Tang et al., 2019): 11,827 videos related to 180 different tasks in 12 domains.



Methods	Frame Accuracy (%)
NN-Viterbi (Richard et al., 2018)	21.17
VGG (Simonyan and Zisserman, 2014)	25.79
TCFPN-ISBA (Ding and Xu, 2018)	34.30
CBT (Sun et al., 2019a)	53.90
MIL-NCE (Miech et al., 2020)	61.00
ActBERT (Zhu and Yang, 2020)	56.95
UniVL	**70.02**

Table 4: Action segmentation results on COIN.

Evaluation Results (4): Video Sentiment Analysis

- **CMU-MOSI** (Zadeh et al., 2018): 2,199 videos for multimodal sentiment analysis.

Input:

Output:

- Highly Positive
- Positive
- Weakly Positive
- Neutral
- Weakly Negative
- Negative
- Highly Negative

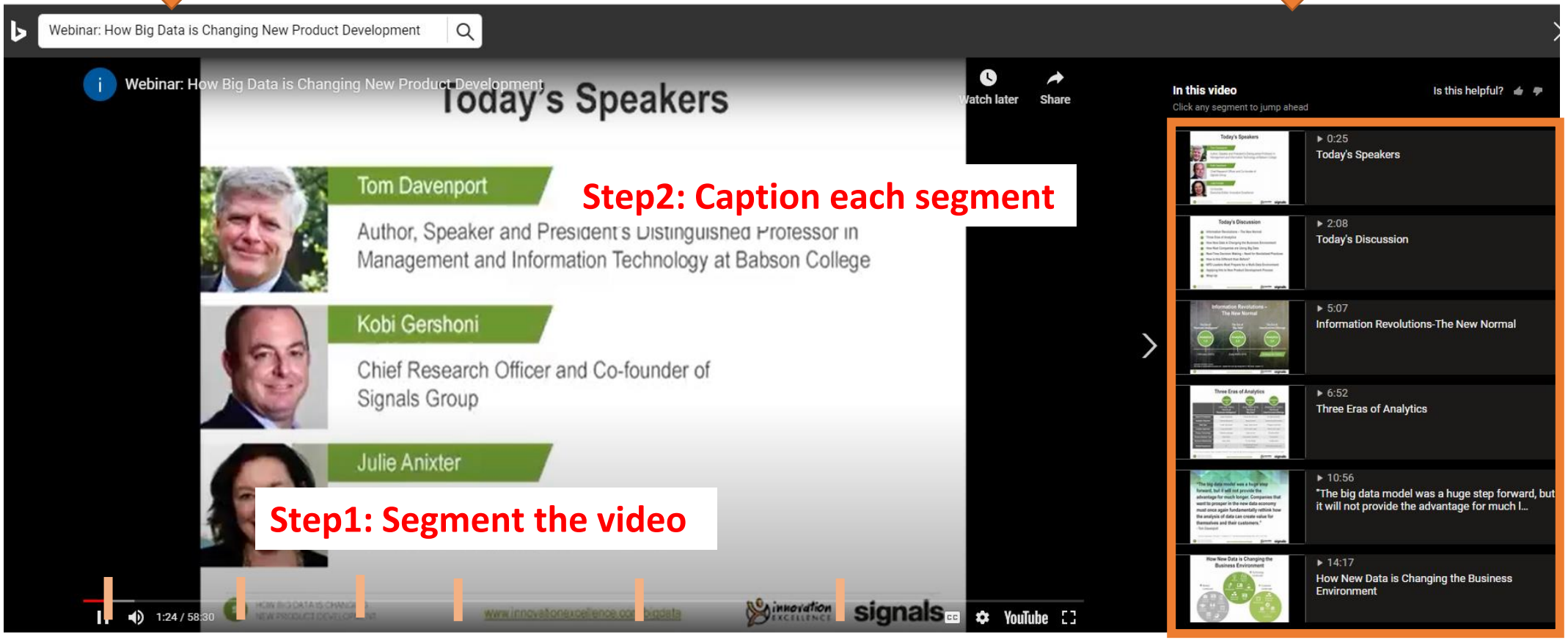
Methods	BA	F1	MAE	Corr
MV-LSTM (Rajagopalan et al., 2016)	73.9/-	74.0/-	1.019	0.601
TFN (Zadeh et al., 2017)	73.9/	73.4/-	1.040	0.633
MARN (Zadeh et al., 2018b)	77.1/	77.0/-	0.968	0.625
MFN (Zadeh et al., 2018a)	77.4/	77.3/-	0.965	0.632
RMFN (Liang et al., 2018)	78.4/	78.0/-	0.922	0.681
RAVEN (Wang et al., 2019)	78.0/	-/-	0.915	0.691
MulT (Tsai et al., 2019)	/83.0	-/82.8	0.870	0.698
FMT (Zadeh et al., 2019)	81.5/83.5	81.4/83.5	0.837	0.744
UniVL	**83.2/84.6**	**83.3/84.6**	**0.781**	**0.767**

Table 6: Multimodal sentiment analysis results on CMU-MOSI dataset. BA means binary accuracy, MAE is Mean-absolute Error, and Corr is Pearson Correlation Coefficient. For BA and F1, we report two numbers following Zadeh et al. (2019): the number on the left side of / is calculated based on the approach from Zadeh et al. (2018b), and the right side is by Tsai et al. (2019).

Unicoder-VL enables video chaptering.

Input a video

Output video chapters



Webinar: How Big Data is Changing New Product Development

Outline

- **Background**
- **Multilingual/Multimodal Pre-trained Models**
  - Unicoder (for multilingual language tasks)
  - Unicoder-VL (for image-language tasks)
  - Unicoder-VL (for video-language tasks)
- **From Natural Language to Programming Language**
- **Summary & Future Work**

From Natural Language to Programming Language

CodeBERT:
A Pre-Trained Model for Programming and Natural Languages

Zhangyin Feng[1]*, Daya Guo[2]*, Duyu Tang[3], Nan Duan[3], Xiaocheng Feng[1]
Ming Gong[4], Linjun Shou[4], Bing Qin[1], Ting Liu[1], Daxin Jiang[4], Ming Zhou[3]
[1] Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China
[2] The School of Data and Computer Science, Sun Yat-sen University, China
[3] Microsoft Research Asia, Beijing, China
[4] Microsoft Search Technology Center Asia, Beijing, China
{zyfeng,xcfeng,qinb,tliu}@ir.hit.edu.cn
guody5@mail2.sysu.edu.cn
{dutang,nanduan,migon,lisho,djiang,mingzhou}@microsoft.com

Abstract

We present CodeBERT, a *bimodal* pre-trained model for programming language (PL) and natural language (NL). CodeBERT learns general-purpose representations that support downstream NL-PL applications such as natural language code search, code documentation generation, etc. We develop CodeBERT with Transformer-based neural architecture, and train it with a hybrid objective function that incorporates the pre-training task of replaced token detection, which is to detect plausible alternatives sampled from generators. This enables us to utilize both "*bimodal*" data of NL-PL pairs and "*unimodal*" data, where the former provides input tokens for model training while the latter helps to learn better generators. We evaluate CodeBERT on two NL-PL applications by fine-tuning model parameters. Results show that CodeBERT achieves state-of-the-art performance on both natural language code search and code documentation generation. Furthermore, to investigate what type of knowledge is learned in CodeBERT, we construct a dataset for NL-PL probing, and evaluate in a zero-shot setting where parameters of pre-trained models are fixed. Results show that CodeBERT performs better than previous pre-trained models on NL-PL probing.[1]

and RoBERTa (Liu et al., 2019) have dramatically improved the state-of-the-art on a variety of natural language processing (NLP) tasks. These pre-trained models learn effective contextual representations from massive unlabeled text optimized by self-supervised objectives, such as masked language modeling, which predicts the original masked word from an artificially masked input sequence. The success of pre-trained models in NLP also drives a surge of multi-modal pre-trained models, such as ViLBERT (Lu et al., 2019) for language-image and VideoBERT (Sun et al., 2019) for language-video, which are learned from *bimodal* data such as language-image pairs with *bimodal* self-supervised objectives.

In this work, we present CodeBERT, a *bimodal* pre-trained model for natural language (NL) and programming language (PL) like Python, Java, JavaScript, etc. CodeBERT captures the semantic connection between natural language and programming language, and produces general-purpose representations that can broadly support NL-PL understanding tasks (e.g. natural language code search) and generation tasks (e.g. code documentation generation). It is developed with the multi-layer Transformer (Vaswani et al., 2017), which is adopted in a majority of large pre-trained models. In order to make use of both *bimodal* instances

GRAPHCODEBERT: PRE-TRAINING CODE REPRESENTATIONS WITH DATA FLOW

Daya Guo[1]*, Shuo Ren[2]*, Shuai Lu[3]*, Zhangyin Feng[4]*, Duyu Tang[5], Shujie Liu[5], Long Zhou[5],
Nan Duan[5], Alexey Svyatkovskiy[6], Shengyu Fu[6], Michele Tufano[6], Shao Kun Deng[6],
Colin Clement[6], Dawn Drain[6], Neel Sundaresan[6], Jian Yin[1], Daxin Jiang[7], and Ming Zhou[5]
[1]Sun Yat-sen University, [2]Beihang University, [3]Peking University,
[4]Harbin Institute of Technology, [5]Microsoft Research Asia, [6]Microsoft Devdiv, [7]Microsoft STCA

ABSTRACT

Pre-trained models for programming language have achieved dramatic empirical improvements on a variety of code-related tasks such as code search, code completion, code summarization, etc. However, existing pre-trained models regard a code snippet as a sequence of tokens, while ignoring the inherent structure of code, which provides crucial code semantics and would enhance the code understanding process. We present GraphCodeBERT, a pre-trained model for programming language that considers the inherent structure of code. Instead of taking syntactic-level structure of code like abstract syntax tree (AST), we use data flow in the pre-training stage, which is a semantic-level structure of code that encodes the relation of "where-the-value-comes-from" between variables. Such a semantic-level structure is neat and does not bring an unnecessarily deep hierarchy of AST, the property of which makes the model more efficient. We develop GraphCodeBERT based on Transformer. In addition to using the task of masked language modeling, we introduce two structure-aware pre-training tasks. One is to predict code structure edges, and the other is to align representations between source code and code structure. We implement the model in an efficient way with a graph-guided masked attention function to incorporate the code structure. We evaluate our model on four tasks, including code search, clone detection, code translation, and code refinement. Results show that code structure and newly introduced pre-training tasks can improve GraphCodeBERT and achieves state-of-the-art performance on the four downstream tasks. We further show that the model prefers structure-level attentions over token-level attentions in the task of code search.

1 INTRODUCTION

Pre-trained models such as ELMo (Peters et al., 2018), GPT (Radford et al., 2018) and BERT (Devlin et al., 2018) have led to strong improvement on numerous natural language processing (NLP) tasks. These pre-trained models are first pre-trained on a large unsupervised text corpus, and then fine-tuned on downstream tasks. The success of pre-trained models in NLP also promotes the development of pre-trained models for programming language. Existing works (Kanade et al., 2019; Karampatsis & Sutton, 2020; Feng et al., 2020; Svyatkovskiy et al., 2020; Buratti et al., 2020) regard a source code as a sequence of tokens and pre-train models on source code to support code-related tasks such as code search, code completion, code summarization, etc. However, previous works only utilize source

CodeBERT

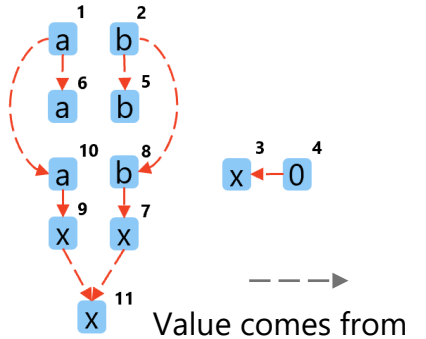
Source code

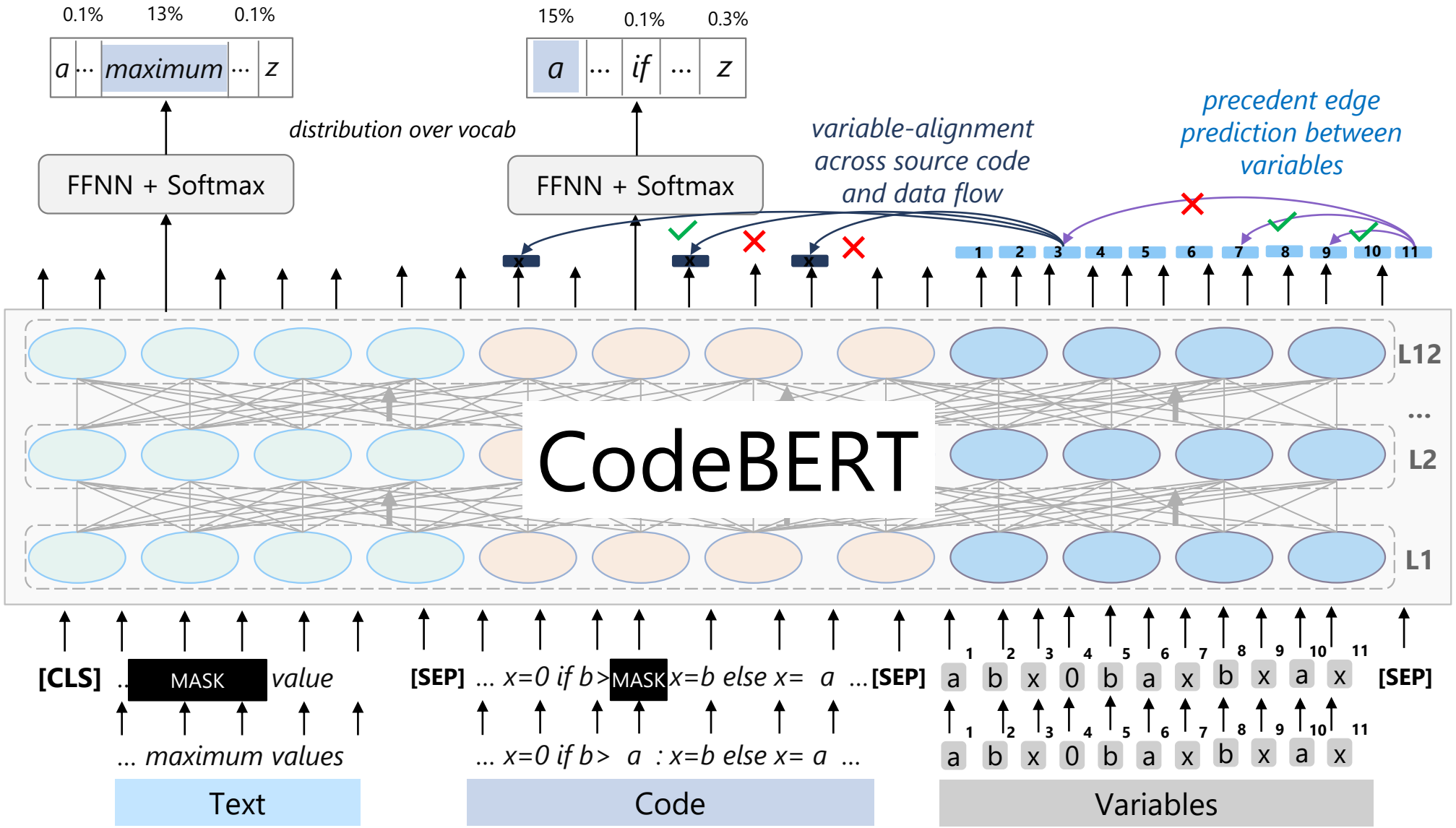
```
def max(a, b):
    x=0
    if b>a:
        x=b
    else:
        x=a
    return x
```

Comment

Return maximum value

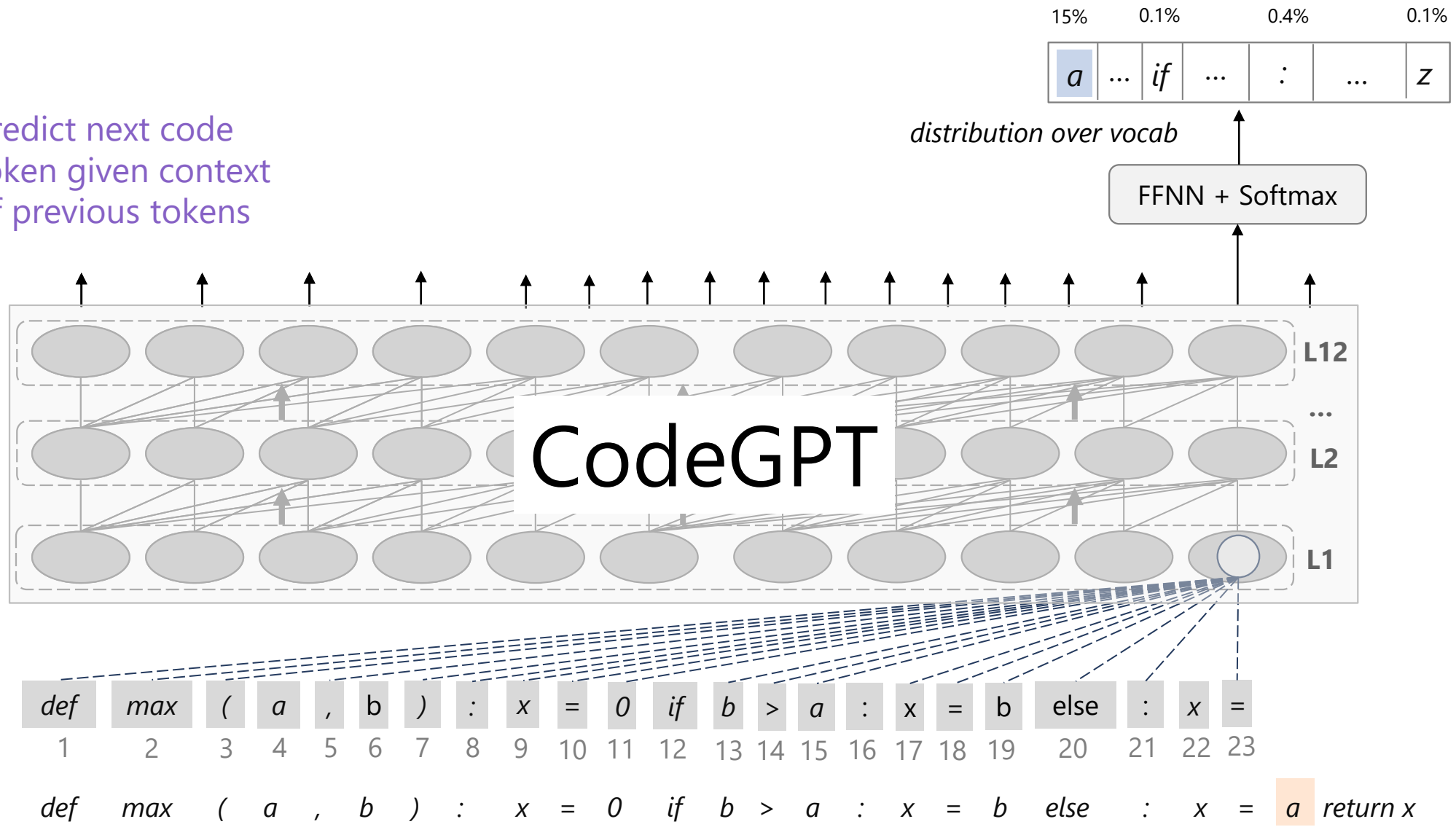
Structure





CodeGPT

Predict next code token given context of previous tokens



Trained for Python, C#, Java, C++

CodeXGLUE

Category	Task	Dataset Name	Language	Train/Dev/Test Size	Baselines
Code-Code	Clone Detection	BigCloneBench	Java	900K/416K/416K	CodeBERT
		POJ-104	C/C++	32K/8K/12K	
	Defect Detection	Defects4J	C	21k/2.7k/2.7k	
	Cloze Testing	CT-all	Python, Java, PHP, Javascript, Ruby, Go	-/-/176k	
		CT-max/min	Python, Java, PHP, Javascript, Ruby, Go	-/-/2.6k	
	Code Completion	PY150	Python	100k/5k/50k	CodeGPT
		GitHub Java Corpus	Java	13k/7k/8k	
	Code Refinement	Bugs2Fix	Java	98K/12K/12K	Encoder-Decoder
	Code Translation	CodeTrans	Java-C#	10K/0.5K/1K	
Text-Code	NL Code Search	CodeSearchNet, AdvTest	Python	251K/9.6K/19K	CodeBERT
		StacQC, WebQueryTest	Python	2.9k/0.9k/1.9k	
	Text-to-Code Generation	CONCODE	Java	100K/2K/2K	CodeGPT
Code-Text	Code Summarization	CodeSearchNet	Python, Java, PHP, Javascript, Ruby, Go	908K/45K/53K	Encoder-Decoder
Text-Text	Document Translation	Microsoft Docs	English-Latvian/Danish/Norwegian/Chinese	156K/4K/4K	

CodeXGLUE: A benchmark dataset and open challenge for code intelligence

Published September 29, 2020

Research Area

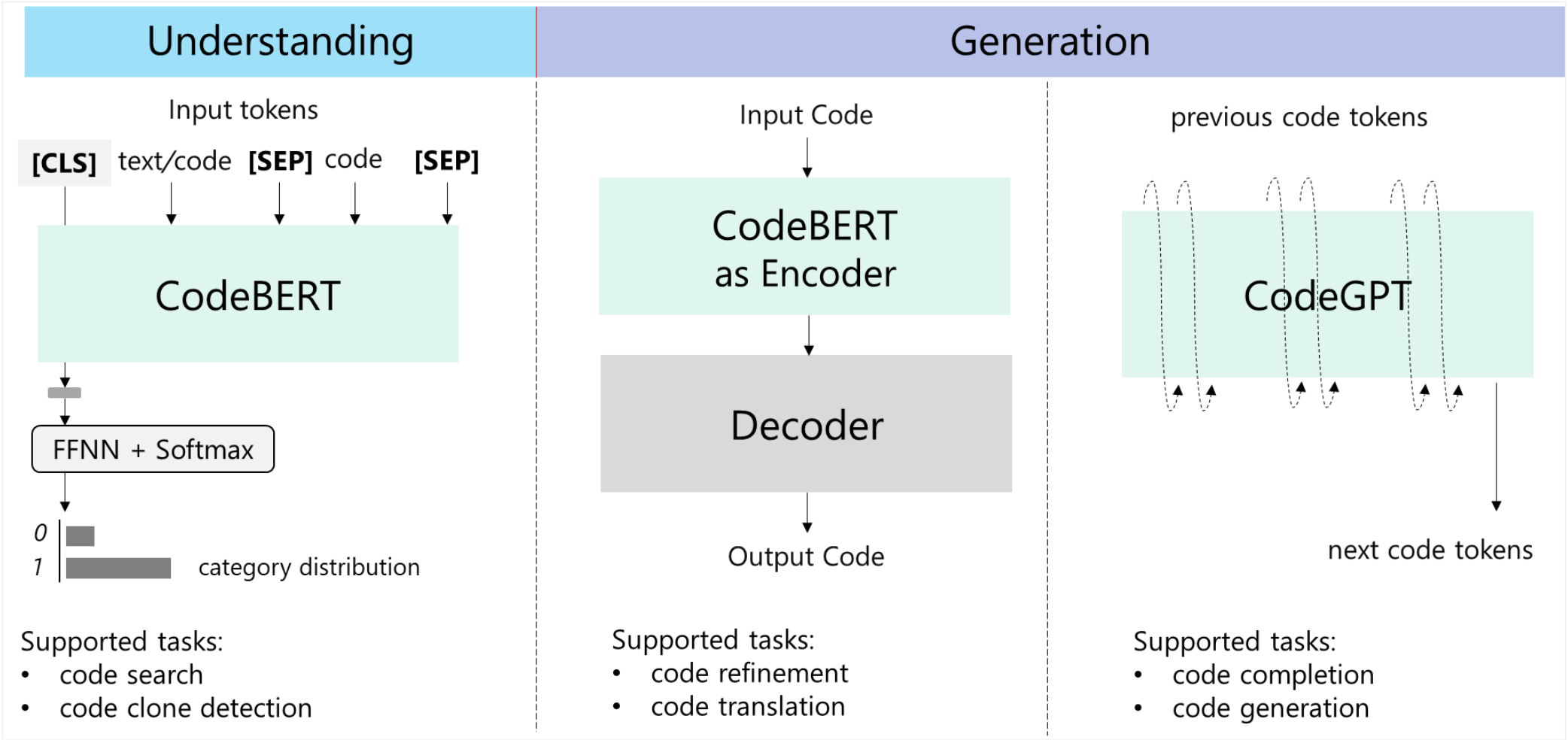
Artificial intelligence

Human language technologies

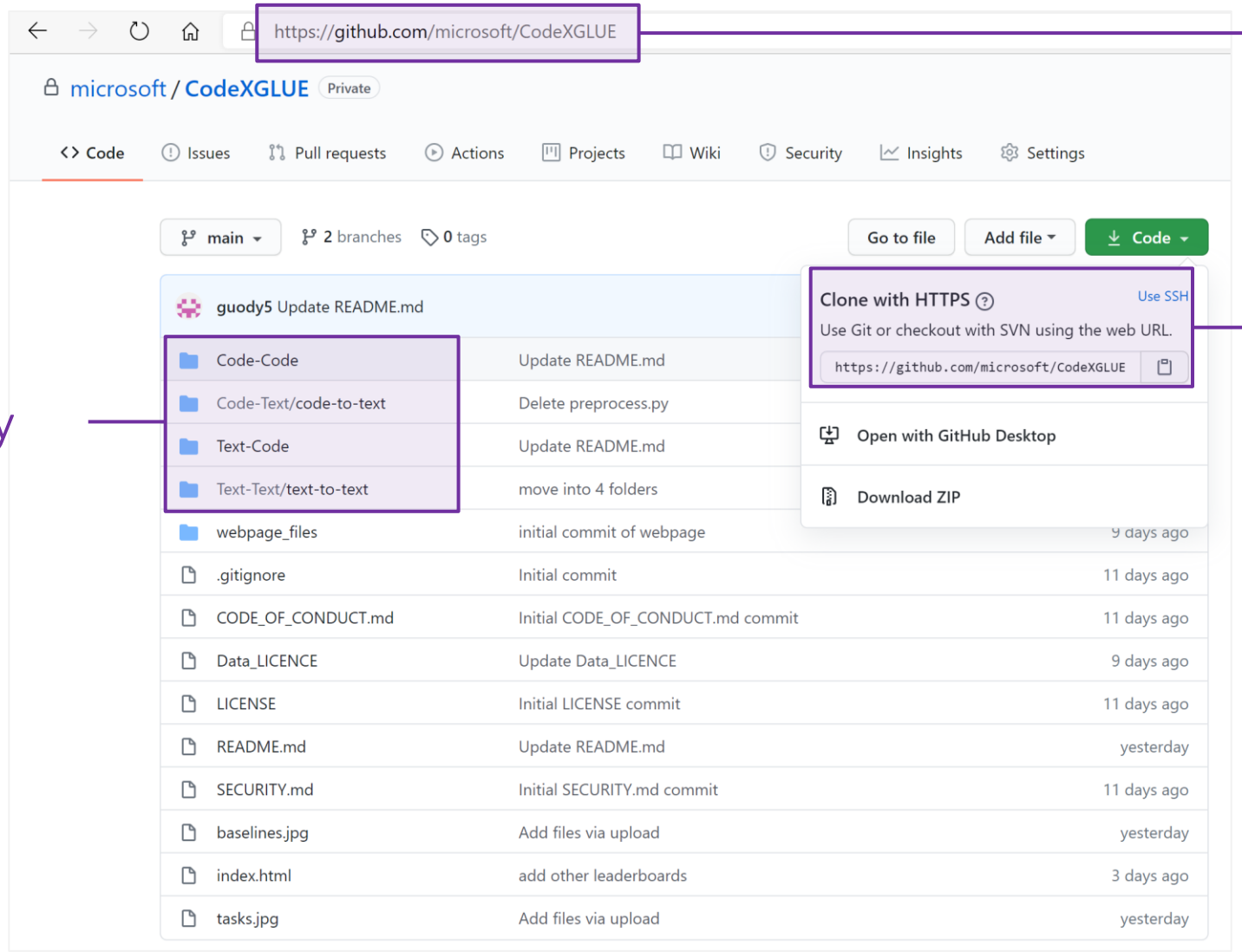
According to Evans Data Corporation, there are 23.9 million professional developers in 2019, and the population is expected to reach 28.7 million in 2024. With the growing population of developers, code intelligence, which aims to leverage AI to help software developers improve the productivity of the development process, is growing increasingly important in both communities of software engineering and artificial intelligence.

When developers want to find code written by others with the same intent, code search systems can help automatically retrieve semantically relevant code given natural language queries. When developers are confused about what to write next, code completion systems can help by automatically completing the following tokens given the context of the edits being made. When developers want to implement Java code with the same function of some existing body of Python code, code-to-code translation systems can help translate from one programming language (Python) to another (Java).

Provide Strong Baseline Models to Facilitate Participants



GitHub Repo (https://github.com/microsoft/CodeXGLUE)



GitHub website

Git clone command

Folder for each category

Outline

- **Background**
- **Multilingual/Multimodal Pre-trained Models**
  - Unicoder (for multilingual language tasks)
  - Unicoder-VL (for image-language tasks)
  - Unicoder-VL (for video-language tasks)
- **From Natural Language to Programming Language**
- **Summary & Future Work**

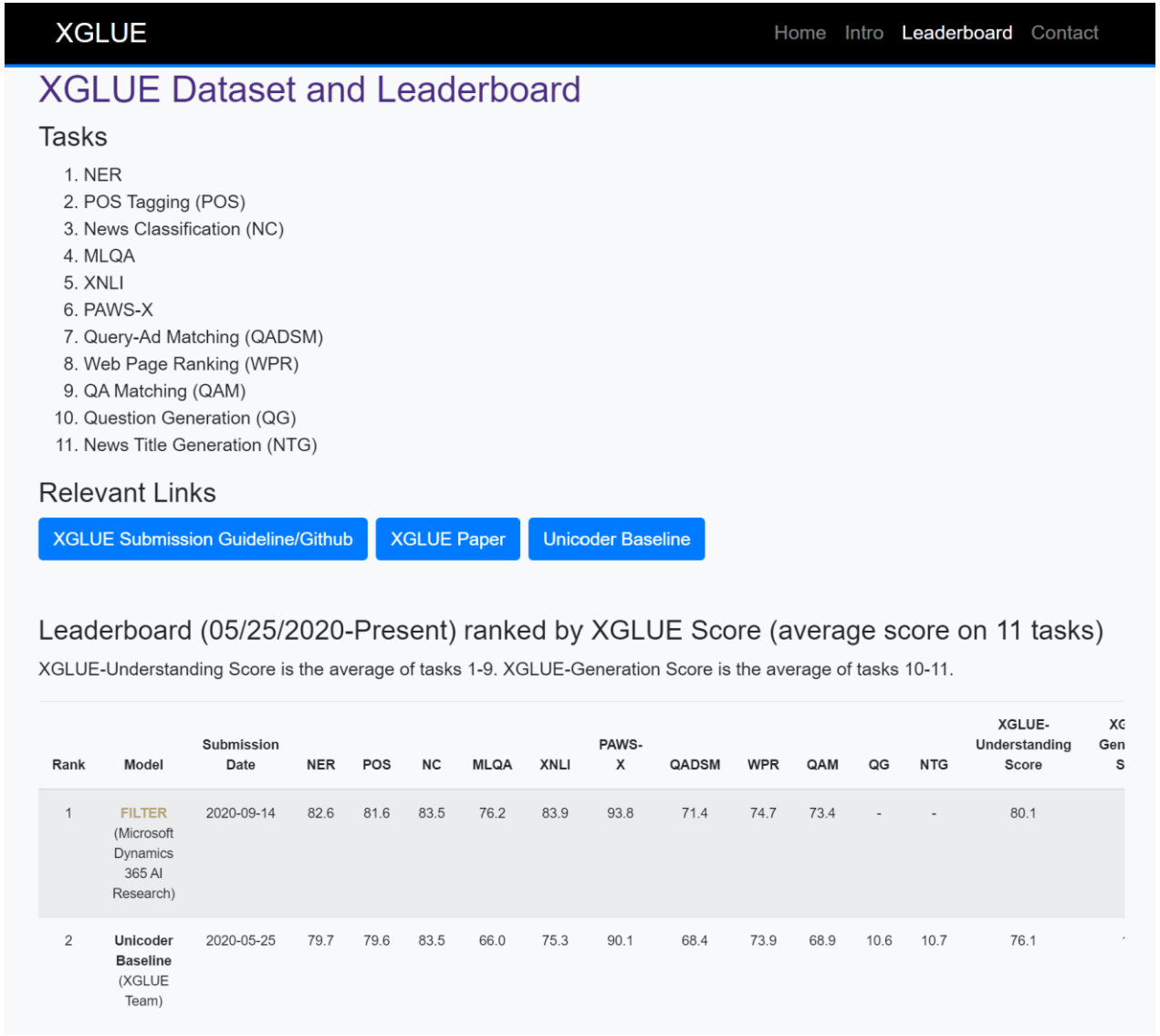
Summary

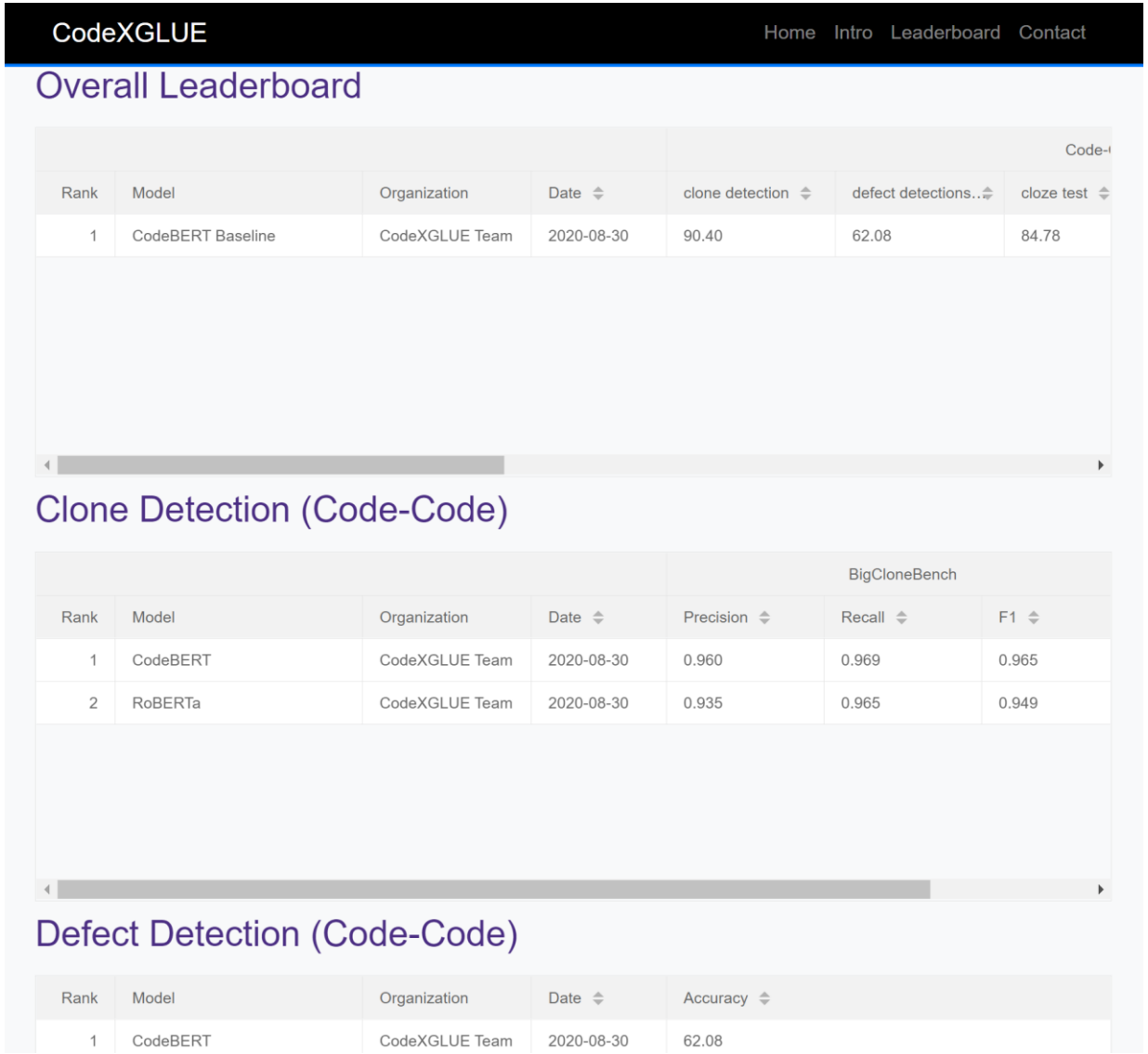
- **Advantages**
  - Alleviation of low-resource issues
  - One model for different tasks
  - State-of-the-art performance
- **Issues**
  - Very high computational costs in training
  - Very large model sizes and slow inference speed
  - Lack of common sense, world knowledge and interpretable reasoning capabilities

Future Work

- Pre-trained models with **new tasks**, **modalities** and **architectures**.
- Pre-trained models with **smaller** model sizes and **faster** training.
- Pre-trained models with **structured knowledge** and **common sense**.
- Pre-trained models with **reasoning** mechanisms.
- Pre-trained models with **interpretability** mechanisms.
- …

Thank you and welcome to use our datasets!





XGLUE: https://microsoft.github.io/XGLUE/

CodeXGLUE: https://microsoft.github.io/CodeXGLUE/