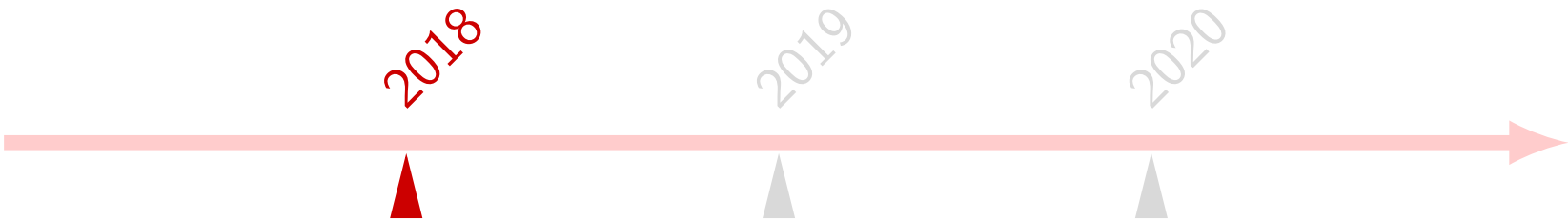
Recent Advances on Multilingual NER

Yong Jiang

Alibaba DAMO NLP

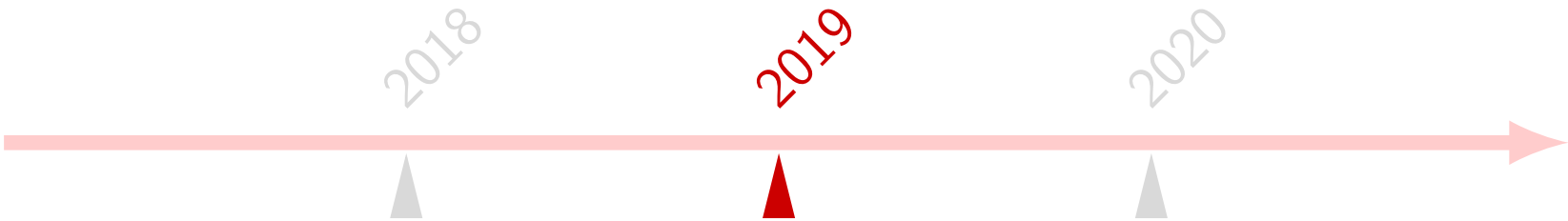
joint work with Xinyu Wang, Zechuan Hu, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang & Kewei Tu

Research Roadmap



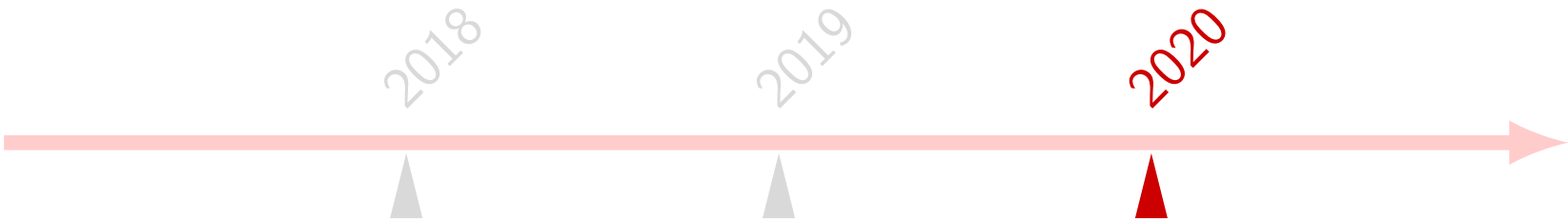
- **Before Joining Alibaba DAMO Academy:**
- 2019: Ph.D from ShanghaiTech University
- 2018: Grammar Induction (Unsupervised Dependency Parsing)
- 2019: Multilingual Grammar Induction

Research Roadmap



- **After Joining Alibaba DAMO Academy:**
- 2019: Knowledge Distillation Approaches.

Research Roadmap



- **After Joining Alibaba DAMO Academy:**
- 2020: Monolingual NER
- 2020: Cross-lingual NER

Presentation for Today

- Why multilingual NER ?
- How to build state-of-the-art NER models under different settings ?

1. Why Multilingual NER ?

– A basic module for many applications.

Why NER ?

- A classical problem in information extraction.
- Entity recognition is a basic tool for building knowledge graph.
- In search, structured understanding for query & document.

Challenges of building SOTA Multilingual NER Models:

- Natural language sentences are flexible. Examples in CoNLL 2003:
  - West Bromwich 3 0 2 1 2 3 2
  - Man City.
- Ambiguity and lack of knowledge.
  - Hong Kong: LOC
  - Hong Kong Newsroom: ORG
  - Hong Kong Open: MISC
- Low resource.
  - multi-lingual: 7000+ languages
  - multi-domain: social media, news, biomedical, e-commerce.

These challenges also exist in many kinds of NER tasks!

On building SOTA mono-lingual models:

- More Embeddings, Better Sequence Labelers?, Findings of EMNLP 2020
- An Investigation of Potential Function Designs for Neural CRF. Findings of EMNLP 2020
- Automated Concatenation of Embeddings for Structured Prediction, ACL 2021
- Improving Named Entity Recognition by External Context Retrieving and Cooperative Learning, ACL 2021

On building cross-lingual models:

- Risk Minimization for Zero-shot Sequence Labeling. ACL 2021
- Multi-View Cross-Lingual Structured Prediction with Minimum Supervision. ACL 2021

On building unified models:

- Structure-Level Knowledge Distillation for Multilingual Sequence Labeling. ACL 2020
- Structural Knowledge Distillation: Tractably Distilling Information for Structured Predictor. ACL 2021

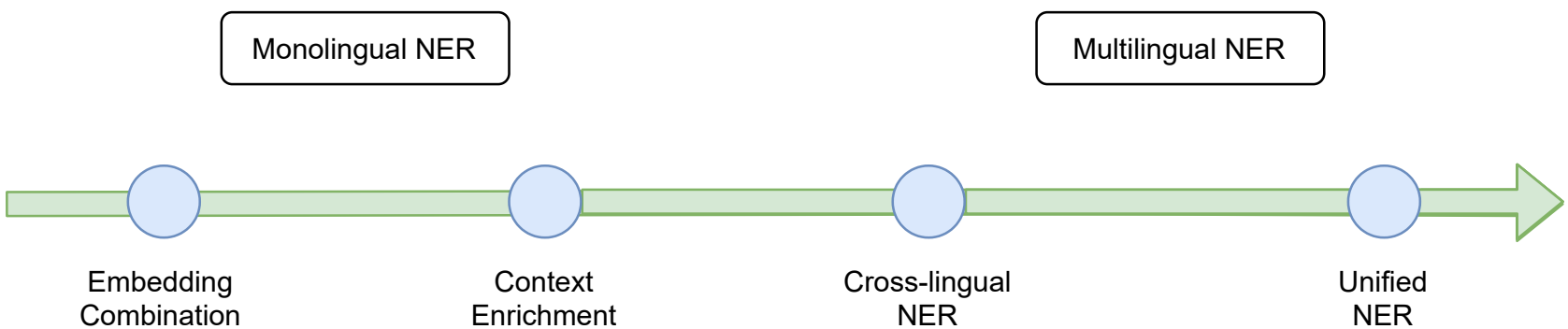
On speeding up models:

- AIN: Fast and Accurate Sequence Labeling with Approximate Inference Network. EMNLP 2020

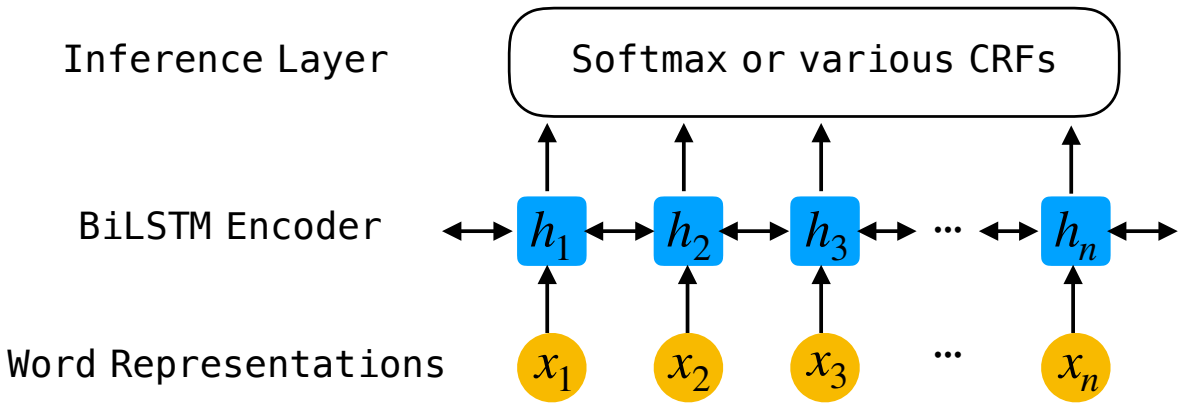
2. How to build a SOTA NER model ?

Research Roadmap for Today

Monolingual NER	Embedding Combination (Findings of EMNLP 2020)
	Automatic Combination of Embedding (ACL 2021)
Context Enrichment	External Context Retrieval (ACL 2021)
Low-resource NER	Risk Minimization for Zero-shot NER (ACL 2021)
	Multi-view Learning (ACL 2021)
Unified Multi-NER	Knowledge Distillation (ACL 2020, ACL 2021)



Background: NER as Sequence Labeling



Choices:

- Pick specific embeddings, BERT, FLAIR, Elmo, word2vec.
- Directly finetune transformer-based architechtures.

Background: Knowledge Distillation Part 1

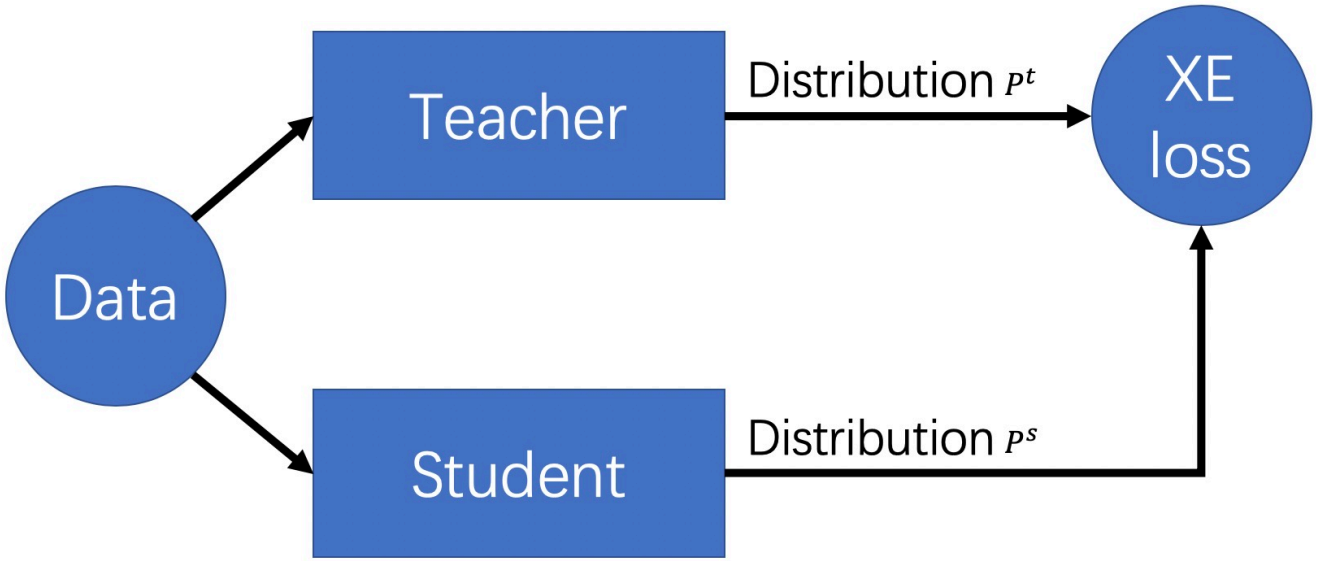


Figure 1: Knowledge distillation

Loss function:

$$\mathcal{L}_{\mathrm{KL}}(\mathbf{x}) = \mathbb{KL}(p^t(\mathbf{y}|\mathbf{x})||p^s(\mathbf{y}|\mathbf{x}))$$

Properties:

- Teaching in a soft manner
- Do not rely on gold labels

Background: Knowledge Distillation Part 2

$$\mathcal{L}_{\mathsf{KL}}(\mathbf{x}) = \mathbb{KL}(p^t(\mathbf{y}|\mathbf{x})||p^s(\mathbf{y}|\mathbf{x}))$$

Generalize to

- Different model family/structure: $\rightarrow$ how $p(\mathbf{y}|\cdot)$ is modeled. (Complexity issues) [Two unified multilingual NER projects]

$$\Rightarrow \mathcal{L}_{\mathsf{KL}} = \mathbb{KL}(p^t_{CRF}(\mathbf{y}|\mathbf{x})||p^s_{Softmax}(\mathbf{y}|\mathbf{x}))$$

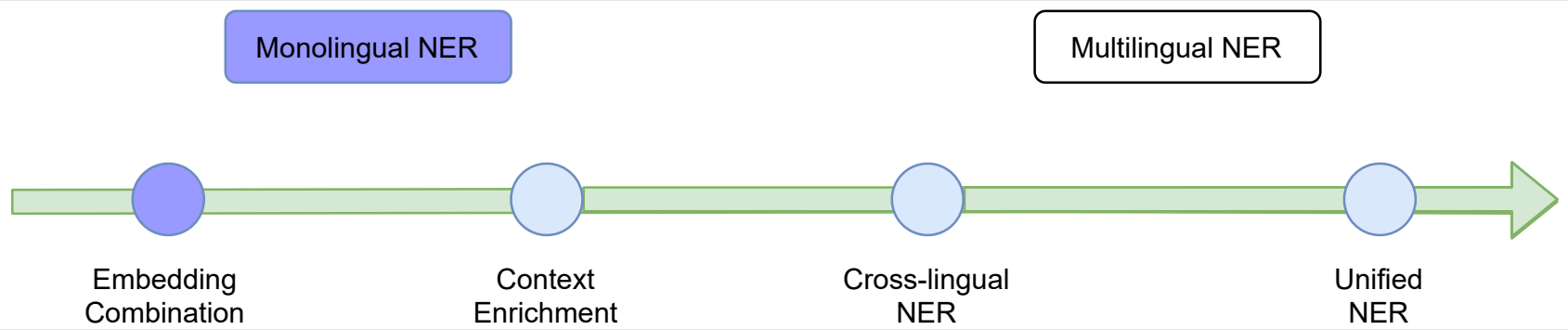
- Same model but different inputs: [Monolingual NER project]

$$\Rightarrow \mathcal{L}_{\mathsf{KL}} = \mathbb{KL}(p(\mathbf{y}|\mathbf{a},\mathbf{b},\mathbf{c})||p(\mathbf{y}|\mathbf{d},\mathbf{e}))$$

- Different models with different inputs: [X-lingual NER project]

$$\Rightarrow \mathcal{L}_{\mathsf{KL}} = \mathbb{KL}(p^t(\mathbf{y}|\mathbf{a},\mathbf{b},\mathbf{c})||p^s(\mathbf{y}|\mathbf{d},\mathbf{e}))$$

2.1 On building a SOTA Monolingual NER model



More Embeddings, Better Sequence Labelers?
Findings of EMNLP 2020

Motivation: Insights from Preliminary Experiments

	NER	Chunking
BERT	**83.8**	91.3
FLAIR	82.1	**92.3**

Table 1: Single embeddings results on 8 NER and 2 chunking dataset

Motivation: Insights from Preliminary Experiments

	NER	Chunking
BERT	**83.8**	91.3
FLAIR	82.1	**92.3**

Table 1: Single embeddings results on 8 NER and 2 chunking dataset

Questions ?

- For sequence labeling, will multiple embeddings be better than one ?
- Will this conclusion hold for different situations? like, low-resource.
- Is word embedding still helpful ?

Plenties of Embedding are Available

We divide the embedding variants according to:

- **Static word embedding**: glove/fasttext/MUSE
- **Static char embedding**: char-CNN/char-BiLSTM
- **Contextual word embed**: Elmo
- **Contextural char embeddings**: FLAIR/m-FLAIR
- **Contextural subword embeddings**: BERT/mBERT/RoBERTa/XLMR

In our experiments:

- **Static word embedding**: fasttext
- **Static char embedding**: char-BiLSTM
- **Contextural char embeddings**: FLAIR
- **Contextural subword embeddings**: mBERT

More Emb, Better Seq-Labelers?

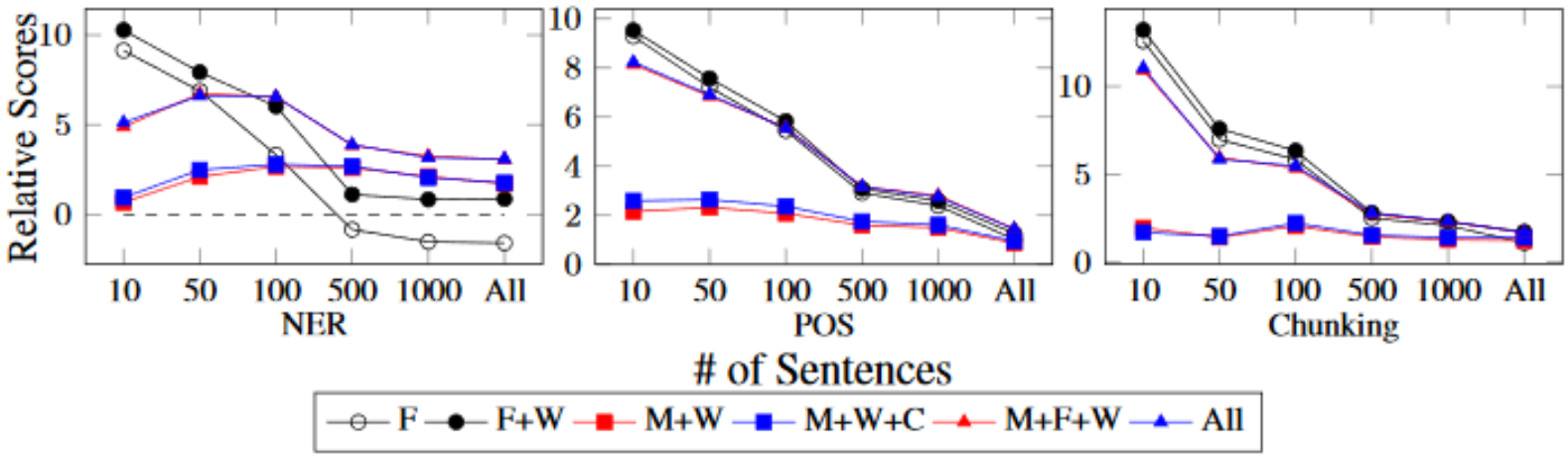


Figure 1: Relative score improvements against models with M-BERT embeddings for three tasks.

An extensive study of concatenating different embeddings.
Findings of:

- **More embedding variants:** generally better.
- **Do we still need static word embeds:** yes! word embeddings are always helpful.
- **Extreme low resources ?** not always better.

More Emb, Better Seq-Labelers?

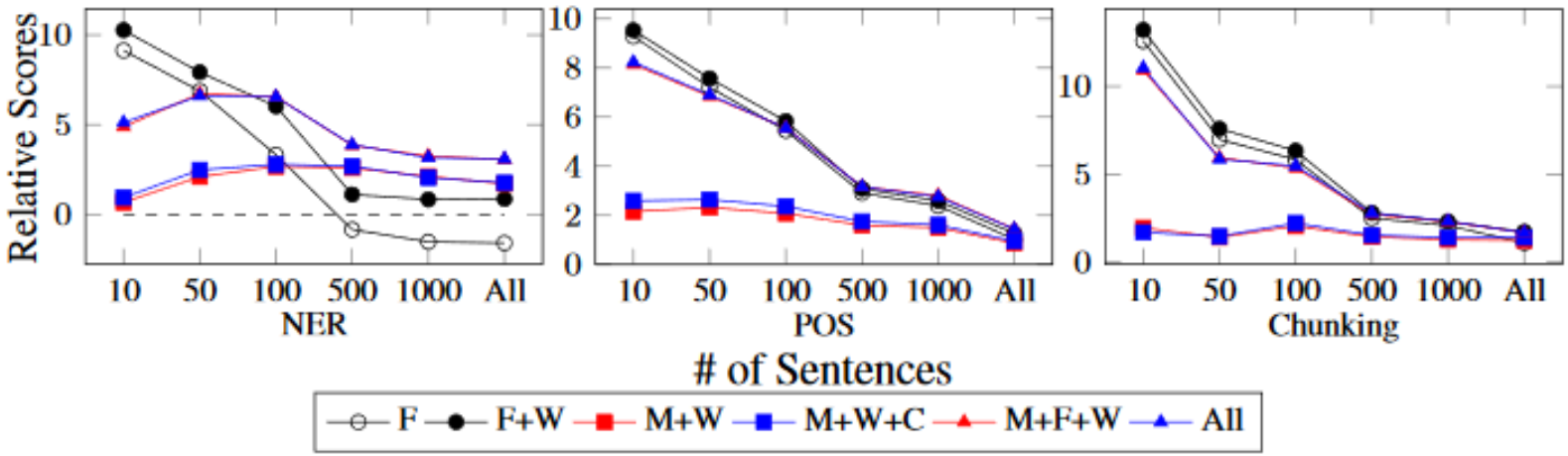


Figure 1: Relative score improvements against models with M-BERT embeddings for three tasks.

An extensive study of concatenating different embeddings.
Findings of:

- **More embedding variants:** generally better.
- **Do we still need static word embeds:** yes! word embeddings are always helpful.
- **Extreme low resources ?** not always better.

In real applications, **hill climb** to search for the best configuration.

2.2 On Automating the Previous Process?

Automated Concatenation of Embeddings for Structured Prediction.
ACL 2021

In Real Applications

How to build a realistic model in practice?

- Thousands of embedding choices to pick for a given task.
- Different tasks may depend on different embeddings.
- How to select task-specific embeddings ?

Reformulate the Problem

Think about how we select the embeddings for a given task:

- Step #1: Pick some embeddings, check the performance.
- Step #2: Compare the performance with previous records.
- Step #3: Get a feeling on which embedding is useful and which is not, and update the "selection" model.

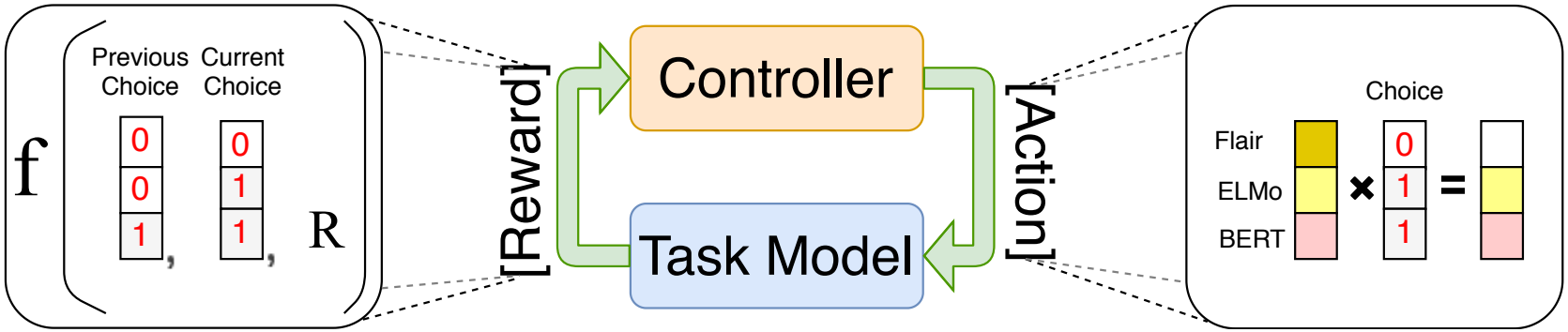
Reformulate the Problem

Think about how we select the embeddings for a given task:

- Step #1: Pick some embeddings, check the performance.
- Step #2: Compare the performance with previous records.
- Step #3: Get a feeling on which embedding is useful and which is not, and update the "selection" model.

Automated Concatenation of Embeddings (ACE).

Approach



Two Modules:

- Controller: which embeddings to pick?

$$\mathbf{P}^{\text{ctrl}}(a;\theta) = \prod_{l=1}^{L} P_l^{\text{ctrl}}(a_l;\theta_l)$$

- Task Model: structured predictor.

$$\mathbf{P}^{\text{seq}}(\mathbf{y}|\mathbf{x}) = \text{BiLSTM-CRF}(V(\mathbf{x}),\mathbf{y})$$

$$\mathbf{P}^{\text{graph}}(\mathbf{y}|\mathbf{x}) = \text{BiLSTM-Biaffine}(V(\mathbf{x}),\mathbf{y})$$

Objective Function:

$$J(\theta) = E_{\mathbf{P}^{\text{ctrl}}(a;\theta)}[R(\mathit{Model}, \mathit{DevData})]$$

Setting: Extensive Experiments

Tasks:

- NER: CoNLL NER
- POS: Twitter POS
- Chunking: DE/EN Chunk
- Aspect Extraction: SemEval 14/15/16
- Dependency Parsing: PTB
- Semantic Dependency Parsing: SemEval 2015

Embeddings: (11 embeddings)

- ELMo
- BERT, XLMR
- Glove, fastText
- char rnn
- mulilingual BERT
- FLAIR, mulilingual FLAIR (forward & backward)

Experiments: Comparisons with Random Search

Table 1: Comparison with concatenating all embeddings and random search baselines on 6 tasks.

	NER				POS			AE							
	de	en	es	nl	Ritter	ARK	TB-v2	14Lap	14Res	15Res	16Res	es	nl	ru	tr
ALL	83.1	92.4	**88.9**	89.8	90.6	92.1	94.6	82.7	88.5	74.2	73.2	74.6	75.0	67.1	67.5
RANDOM	84.0	92.6	88.8	91.9	91.3	92.6	94.6	83.6	88.1	73.5	74.7	75.0	73.6	68.0	70.0
ACE	**84.2**	**93.0**	**88.9**	**92.1**	**91.7**	**92.8**	**94.8**	**83.9**	**88.6**	**74.9**	**75.6**	**75.7**	**75.3**	**70.6**	**71.1**

	CHUNK	DP		SDP						AVG
	CoNLL 2000	UAS	LAS	DM-ID	DM-OOD	PAS-ID	PAS-OOD	PSD-ID	PSD-OOD	
ALL	96.7	96.7	95.1	94.3	90.8	**94.6**	92.9	82.4	81.7	85.3
RANDOM	96.7	96.8	95.2	94.4	90.8	**94.6**	93.0	82.3	81.8	85.7
ACE	**96.8**	**96.9**	**95.3**	**94.5**	**90.9**	94.5	**93.1**	**82.5**	**82.1**	**86.2**

- ACE consistently outperforms Random & All in all datasets & tasks.

[Automated Concatenation of Embeddings for Structured Prediction, ACL 2021]

Marrying finetuning LMs. ACE beats SOTAs

	NER						POS		
	de	de$_{06}$	en	es	nl		Ritter	ARK	TB-v2
Baevski et al. (2019)	-	-	93.5	-	-	Owoputi et al. (2013)	90.4	93.2	94.6
Straková et al. (2019)	85.1	-	93.4	88.8	92.7	Gui et al. (2017)	90.9	-	92.8
Yu et al. (2020)	86.4	90.3	93.5	90.3	93.7	Gui et al. (2018)	91.2	92.4	-
Yamada et al. (2020)	-	-	94.3	-	-	Nguyen et al. (2020)	90.1	94.1	95.2
XLM-R+Fine-tune$^{\diamond}$	87.7	91.4	94.1	89.3	95.3	XLM-R+Fine-tune	92.3	93.7	95.4
ACE+Fine-tune	**88.3**	**91.7**	**94.6**	**95.9**	**95.7**	ACE+Fine-tune	**93.4**	**94.4**	**95.8**

	CHUNK		AE							
	CoNLL 2000		14Lap	14Res	15Res	16Res	es	nl	ru	tr
Akbik et al. (2018)	96.7	Xu et al. (2018)†	84.2	84.6	72.0	75.4	-	-	-	-
Clark et al. (2018)	97.0	Xu et al. (2019)	84.3	-	-	78.0	-	-	-	-
Liu et al. (2019b)	**97.3**	Wang et al. (2020)	-	-	-	72.8	74.3	72.9	71.8	59.3
Chen et al. (2020)	95.5	Wei et al. (2020)	82.7	87.1	72.7	77.7	-	-	-	-
XLM-R+Fine-tune	97.0	XLM-R+Fine-tune	85.9	90.5	76.4	78.9	77.0	77.6	77.7	74.1
ACE+Fine-tune	**97.3**	ACE+Fine-tune	**87.4**	**92.0**	**80.3**	**81.3**	**79.9**	**80.5**	**79.4**	**81.9**

	DP			SDP					
	PTB			DM		PAS		PSD	
	UAS	LAS		ID	OOD	ID	OOD	ID	OOD
Zhou and Zhao (2019)†	97.2	95.7	He and Choi (2020)‡	94.6	90.8	96.1	94.4	86.8	79.5
Mrini et al. (2020)†	97.4	96.3	D & M (2018)	94.0	89.7	94.1	91.3	81.4	79.6
Zhang et al. (2020)	96.1	94.5	Wang et al. (2019)	93.7	88.9	93.9	90.6	81.0	79.4
Wang and Tu (2020)	96.9	95.3	F & G (2020)	94.4	91.0	95.1	93.4	82.6	82.0
XLNET+Fine-tune	97.0	95.6	XLNet+Fine-tune	94.2	90.6	94.8	93.4	82.7	81.8
ACE+Fine-tune	**97.2**	**95.7**	ACE+Fine-tune	**95.6**	**92.6**	**95.8**	**94.6**	**83.8**	**83.4**

- Finetuning LMs makes ACE stronger, achieves SOTA over 24 datasets.

[Automated Concatenation of Embeddings for Structured Prediction, ACL 2021]

Take-Away Message

Main Ideas:

- Again, embedding combination is very useful.
- Before performing concatenation, perform finetuning!

Take-Away Message

Main Ideas:

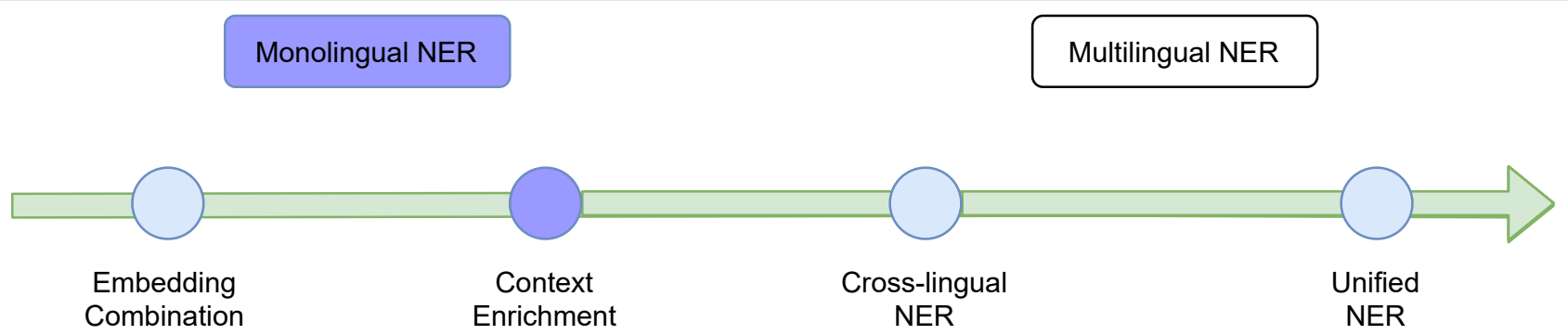
- Again, embedding combination is very useful.
- Before performing concatenation, perform finetuning!

Main Results:

- ACE achieves SOTA performance.
- RL is an useful method to combine related modules.
- Reward design in RL is important.
- How to design? Try and explore!

Our code is released: `https://github.com/Alibaba-NLP/ACE`

2.3 On Enriching Context



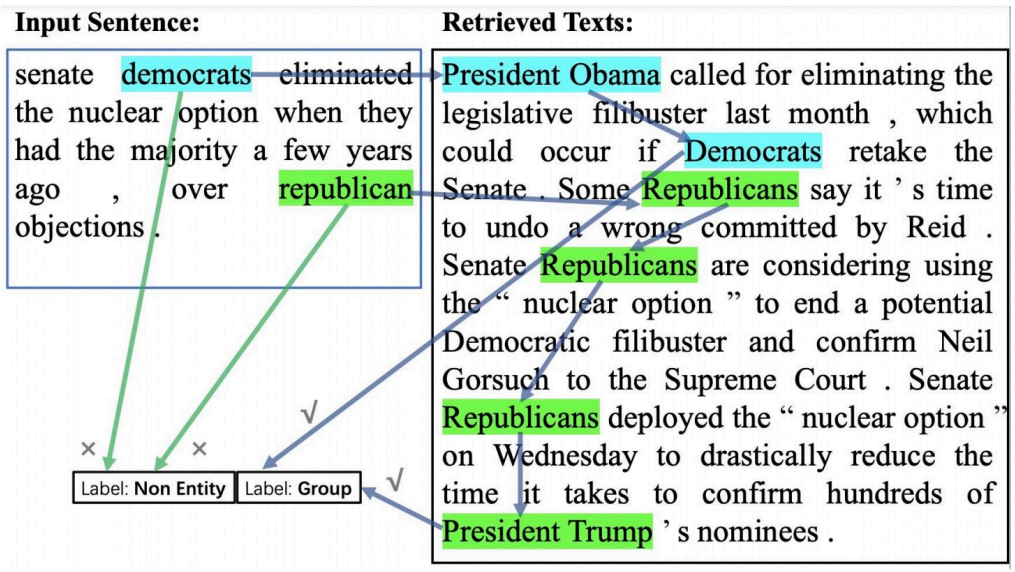
Improving Named Entity Recognition by Retrieving External Contexts and Cooperative Learning. ACL 2021

Motivation

- In ACE, multiple embeddings provide various information to improve NER. (implicit knowledge)
- Another observation from document NER: human-written context is indeed helpful. (explicit knowledge)
- Given a text, we can retrieve text-specific knowledge that helps to perform disambiguation. (explicit knowledge)

Example:

Social Media NER



Model

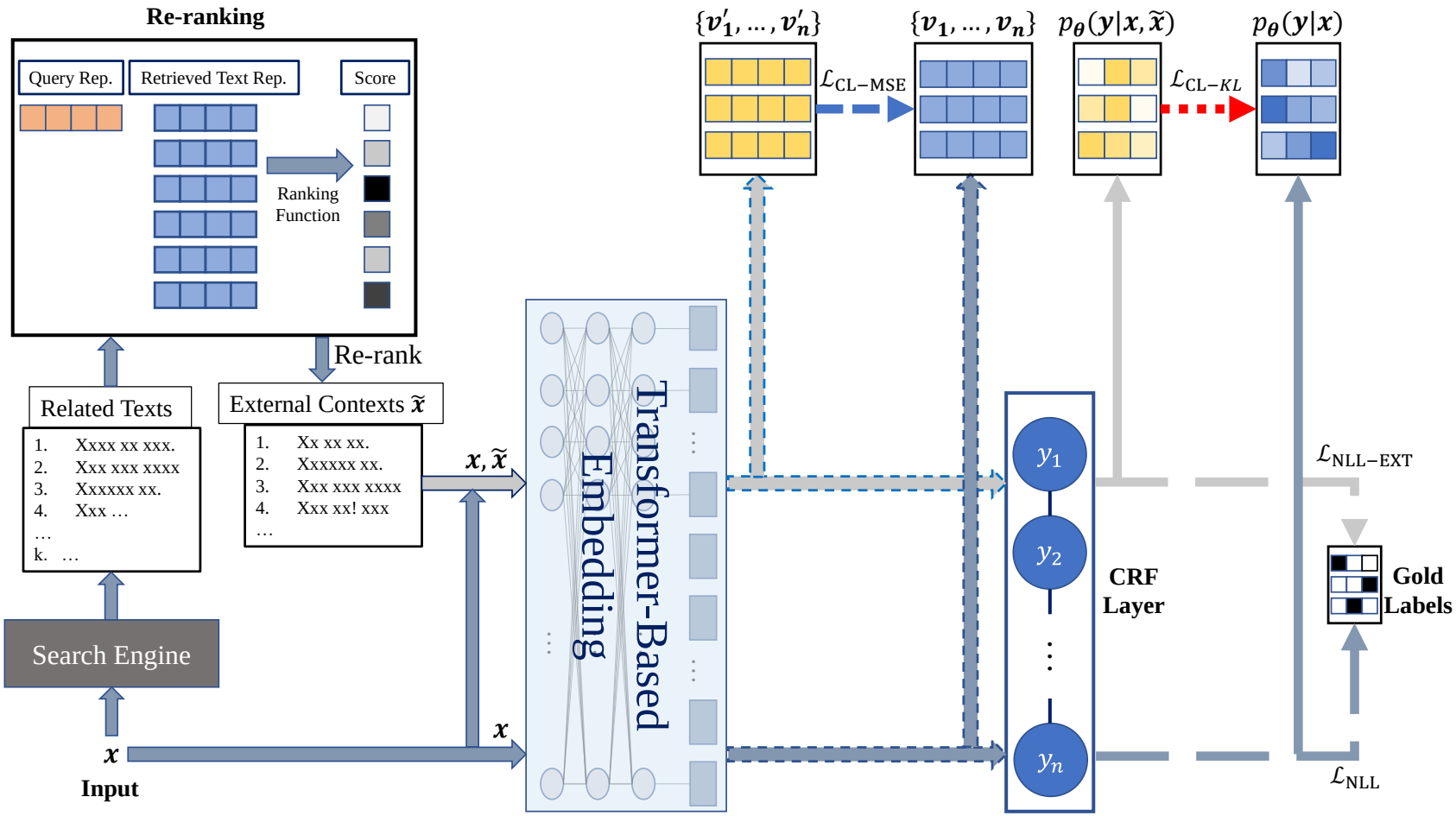


Figure 3: Cooperative Learning/Multi-View Learning Framework

$$\mathcal{L} = -\log p(\mathbf{y}|\mathbf{x}, \tilde{\mathbf{x}}) - \log p(\mathbf{y}|\mathbf{x}) + \mathbb{KL}(p(\mathbf{y}|\mathbf{x})||p(\mathbf{y}|\mathbf{x}, \tilde{\mathbf{x}}))$$

Experiments

	Social Media		News		Biomedical		E-commerce
	WNUT-16	WNUT-17	CoNLL-03	CoNLL++	BC5CDR	NCBI	
Zhou et al. (2019)	55.43	42.83	-	-	-	-	-
Nguyen et al. (2020)	52.10	56.50	-	-	-	-	-
Nie et al. (2020)	55.01	50.36	-	-	-	-	-
Baevski et al. (2019)	-	-	93.50	-	-	-	-
Wang et al. (2019)	-	-	93.43	94.28	-	-	-
Li et al. (2020)	-	-	93.33	-	-	-	-
Nooralahzadeh et al. (2019)	-	-	-	-	89.93	-	-
Bio-Flair (2019)	-	-	-	-	89.42	88.85	-
Bio-BERT (2020)	-	-	-	-	-	87.70	-
Evaluation: W/O CONTEXT							
LUKE (2020)	54.04	55.22	92.42	93.99	89.18	87.62	77.64
W/O CONTEXT	56.04	57.86	93.03	94.20	90.52	88.65	81.47
CL-L_2	57.35†	58.68†	93.08	94.38†	90.70†	89.20†	82.43†
CL-KL	58.14†	59.33†	93.21†	94.55†	90.73†	**89.24**†	82.31†
Evaluation: W/ CONTEXT							
W/ CONTEXT	57.43†	60.20†	93.27†	94.56†	90.76†	89.01†	83.15†
CL-L_2	58.61†	60.26†	93.47†	94.62†	**90.99**†	89.22†	83.87†
CL-KL	**58.98**†	**60.45**†	**93.56**†	**94.81**†	90.93†	88.96†	**83.99**†

Figure 4: Results on NER Dataset

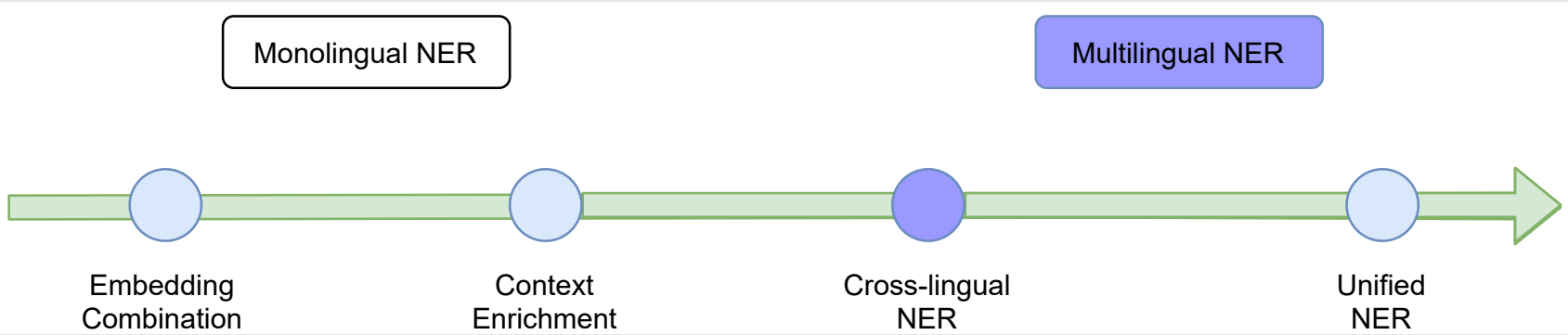
- SOTA in various NER tasks over multiple domains.
- Enriching context is very useful both in training & testing.

Take-Away Message

Main Ideas:

- Utilizing external context helps removing ambiguity. (especially for short text)
- Well-retrieved explicit knowledge might be stronger than implicit knowledge.
- Cooperative learning helps to minimize the gap.

2.4 On Zero-shot Sequence Labeling



Risk Minimization for Zero-shot Sequence Labeling.
ACL 2021

Motivation

- In real-world, we may have 7000+ languages.
- We have lots of domains to process.

Question:

- How can we transfer the knowledge from multiple models of rich resources to low resources?

Motivation

- In real-world, we may have 7000+ languages.
- We have lots of domains to process.

Question:

- How can we transfer the knowledge from multiple models of rich resources to low resources?

Key idea:

- Utilize the prediction results from multiple source models on target languages.
- The prediction is **not fully trustable**!
- **Uncertainty modeling** in cross-lingual learning.
- This uncertainty is designed through a **mapping** that models the relations between the predicted labels from the source models to the true labels.

Existing Approaches

Direct transfer:

$$\mathcal{J}(\boldsymbol{\theta}) = -\log P_{\boldsymbol{\theta}}(\widehat{\mathbf{y}}|\mathbf{x}) = -\sum_{i=1}^{n} \log P_{\boldsymbol{\theta}}(\hat{y}_i|\mathbf{x})$$

Knowledge distillation: [Wu et al. ACL 2020]

$$\mathcal{L}_{\mathsf{KL}}(\mathbf{x}) = \mathbb{KL}(p_T(\mathbf{y}|\mathbf{x})||p_S(\mathbf{y}|\mathbf{x}))$$

Model

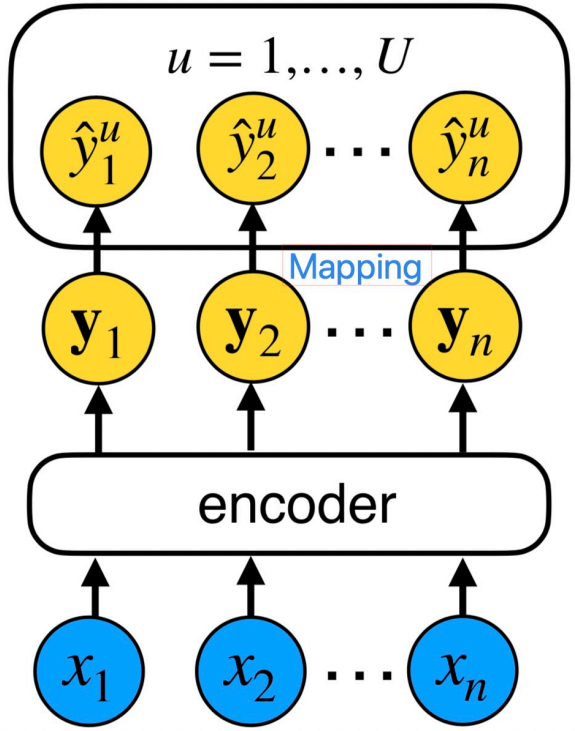
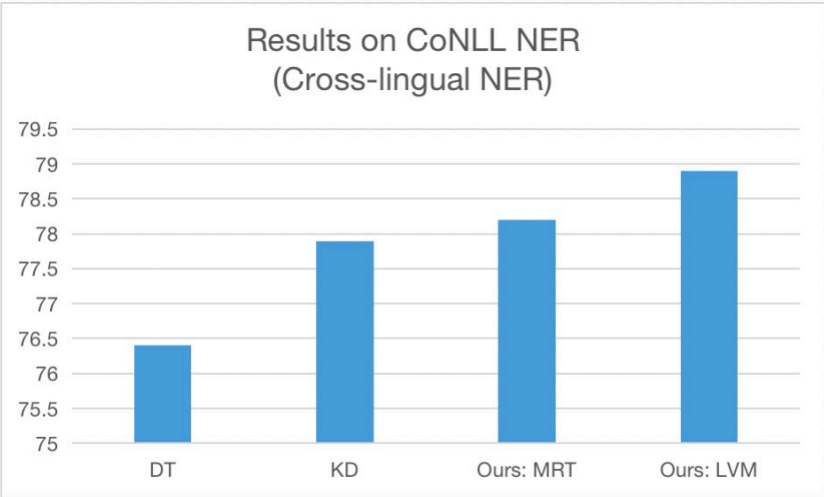


Figure 5: Risk Minimization for Zero-shot NER

$$\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{P_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}[R(\widehat{\mathbf{y}}, \mathbf{y})]$$

- MRT: fixed mapping. LVM: tuned mapping.

Experiments on Cross-lingual & Cross-domain NER



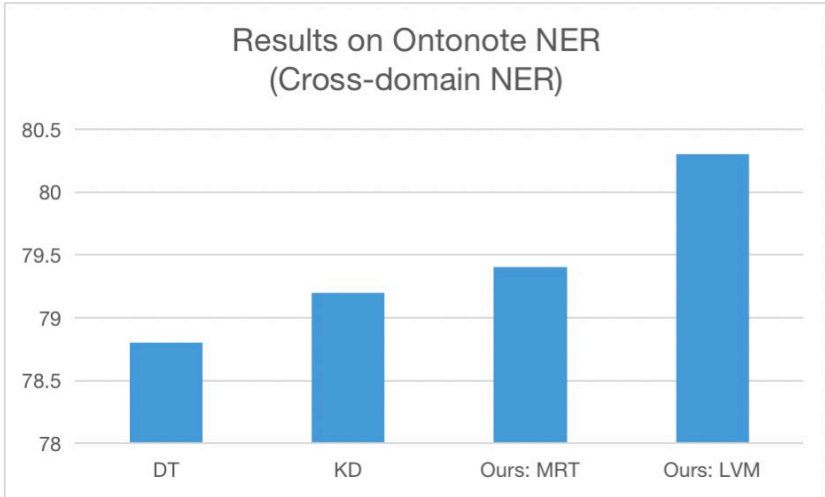


Figure 6: Results on Cross-lingual & Cross-domain Transfer

Take-Away Message

Main Ideas:

- Modeling the uncertainty is useful for zero-shot NER.
- By making the mapping tunable, we obtain better performance.

2.5 On Few-shot Sequence Labeling

Multi-View Cross-Lingual Structured Prediction with Minimum Supervision.
ACL 2021

Motivation

- In the multi-source transfer problem, not all source models are equally important and some may hurt performance on the target language.

Source \ Target	EN	DE	NL	ES
EN	—	72.77	79.47	75.13
DE	75.96	—	78.47	70.74
NL	69.38	72.35	—	74.16
ES	68.55	63.37	69.12	—

Figure 7: Direct bilingual transfer results on the CoNLL02/03 NER task.

Motivating Example

	LOCKERBIE	-	JUICIO	CHAVEZ	PIDE	AYUDA	A	...
En	O	O	B-PER	I-PER	O	B-LOC	O	...
De	B-LOC	O	B-PER	I-PER	O	O	O	...
Nl	O	O	O	B-PER	O	O	O	...
Mean	O	O	B-PER	I-PER	O	O	O	...
Best	O	O	O	B-PER	O	O	O	...
Gold	B-LOC	O	O	B-PER	O	O	O	...

Figure 8: A negative transfer example on Spanish target language.

Model

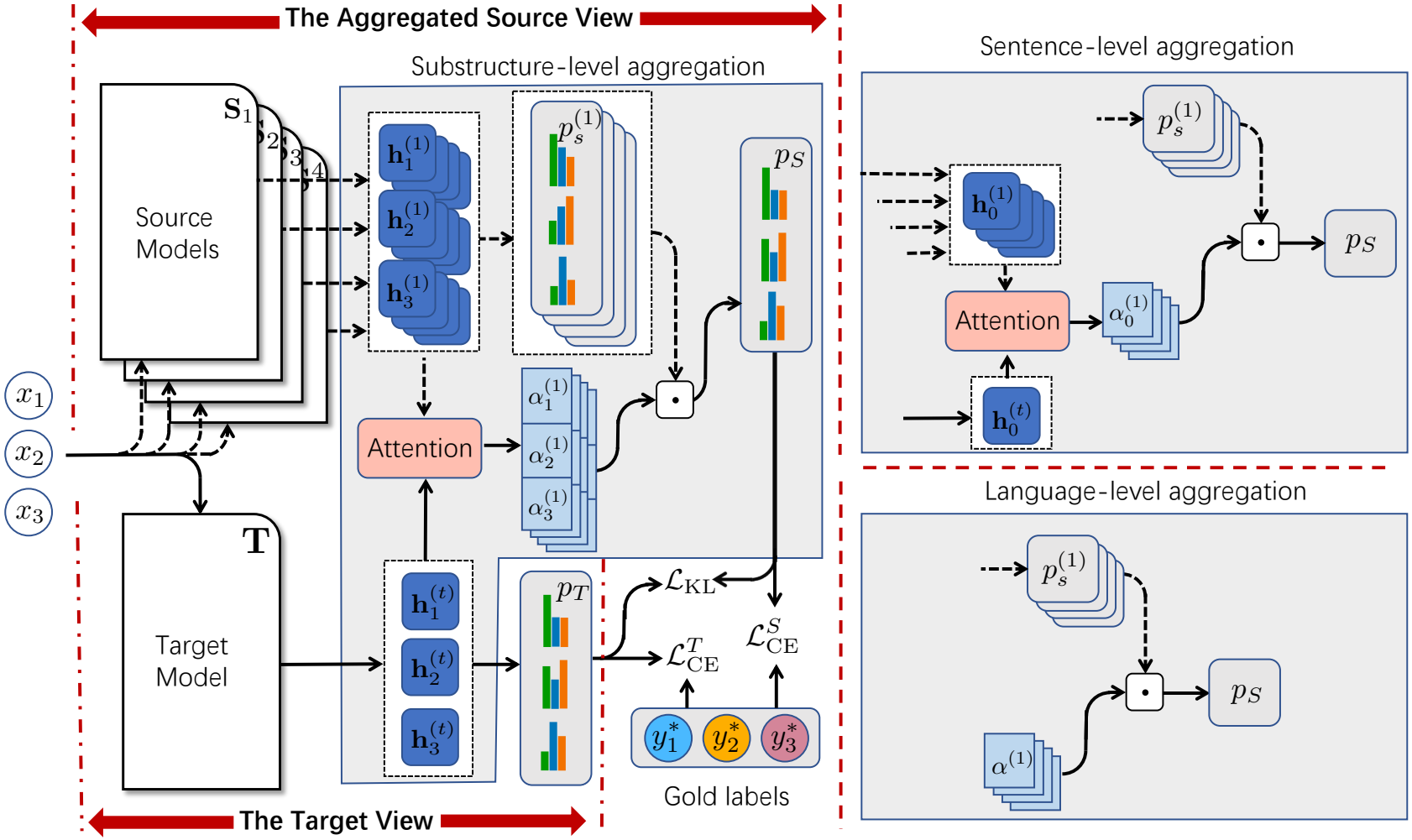


Figure 9: Multi-View Cross-Lingual Structured Prediction

$$\mathcal{L}_{\mathrm{KL}}(\mathbf{x}) = \mathbb{KL}(p_{View1}(\mathbf{y}|\mathbf{x})||p_{View2}(\mathbf{y}|\mathbf{x}))$$

Experiments

With 50 labeled data		CoNLL02/03 NER					POS TAGGING						
		EN	DE	NL	ES	Avg.	EN	CA	ID	HI	FI	RU	Avg.
✗	*DT-gold*	90.13	84.60	89.09	84.30	87.03	95.71	96.80	94.67	94.39	92.18	97.39	95.19
✗	*DT-max(lang)*	80.85	74.27	81.00	78.42	78.64	87.38	94.26	89.33	87.96	82.47	91.71	88.85
✓	DT-Finetuning	72.71	54.49	57.07	70.82	63.77	85.58	92.90	86.73	86.17	72.57	87.65	85.27
✗	DT-vote	81.81	74.52	81.66	78.51	79.13	89.73	94.26	90.29	89.09	82.82	92.51	89.78
✗	DT-max	82.21	74.98	82.19	78.74	79.53	89.71	94.49	90.13	89.13	83.97	92.78	90.04
✗	DT-mean	82.57	75.33	82.19	78.93	79.76	90.04	94.38	90.40	89.26	83.72	92.86	90.11
✓	hard-KD-cat	83.73	75.56	82.30	79.07	80.17	90.22	94.41	90.60	89.52	84.26	92.80	90.30
✓	hard-KD-vote	83.45	75.80	82.48	79.18	80.23	90.06	94.38	90.52	89.53	83.77	92.65	90.15
✓	hard-KD-max	83.14	75.39	82.27	79.40	80.05	90.16	94.56	90.45	89.41	84.89	92.99	90.41
✓	hard-KD-mean	83.42	75.67	82.306	79.29	80.17	90.32	94.46	90.67	89.61	84.48	92.96	90.41
✓	UMM	78.99	75.26	82.48	78.26	78.75	88.14	93.88	89.65	88.42	83.03	93.26	89.40
✓	Self-training[1]	80.76	75.96	82.91	79.63	79.81	89.68	94.46	90.13	89.16	83.72	94.02	90.19
✓	Tri-training[2]	80.63	76.62	83.14	79.10	79.87	89.83	94.40	90.04	89.69	83.94	**94.05**	90.32
✓	soft-KD-avg[3]	83.52	75.84	82.46	79.24	80.26	90.31	94.62	90.75	89.69	84.55	93.22	90.52
✓	soft-KD-sim[4]	83.58	75.99	82.94	79.63	80.54	89.79	94.80	90.79	89.70	84.55	93.54	90.53
✓	Ours-lang	83.48	75.88	83.02	79.79	80.54	90.27	94.73	90.81	89.62	84.78	93.44	90.61
✓	Ours-sent	83.83	76.13	82.92	80.07	80.74	90.31	94.80	90.91	89.71	84.93	93.51	90.70
✓	Ours-sub	**84.78**	76.56	**84.12**	**80.34**	**81.45**	**91.12**	**95.30**	**91.15**	**90.11**	**85.68**	93.57	**91.16**

[1] Yarowsky (1995); McClosky et al. (2006) [2] Ruder and Plank (2018) [3,4] Wu et al. (2020)

Figure 10: Results on Open-source Cross-lingual Transfer (CoNLL NER & POS)

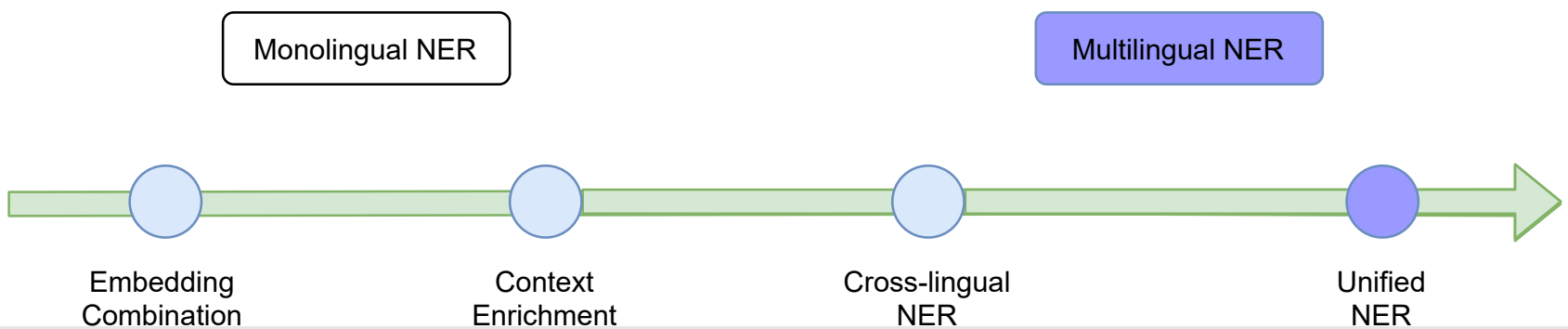
Experiments

With 50 labeled data		EN UAS	EN LAS	CA UAS	CA LAS	ID UAS	ID LAS	HI UAS	HI LAS	FI UAS	FI LAS	RU UAS	RU LAS	Avg. UAS	Avg. LAS
✗	*DT-gold*	93.30	87.67	92.80	88.61	89.10	81.80	88.54	80.41	84.65	74.67	92.20	86.20	90.10	83.23
✗	*DT-max(lang)*	77.71	67.90	84.39	76.17	76.86	68.37	70.49	52.64	76.62	56.98	72.31	64.45	76.40	64.42
✓	DT-Finetuning	49.75	41.87	53.59	48.38	47.19	38.39	50.32	41.58	32.88	22.22	35.87	28.78	44.93	36.87
✗	DT-vote	80.94	71.65	83.82	75.81	77.79	66.76	75.98	63.18	68.23	52.82	79.80	69.62	77.76	66.64
✗	DT-max	81.07	71.56	84.29	75.87	77.46	65.78	76.54	63.42	69.13	52.79	79.42	69.22	77.99	66.44
✗	DT-mean	81.79	72.96	84.52	76.68	78.45	67.56	76.80	64.31	68.83	54.11	80.54	70.77	78.49	67.73
✓	hard-KD-cat	82.16	74.29	84.41	77.13	78.28	68.26	77.26	65.56	69.61	55.80	80.28	70.90	78.67	68.66
✓	hard-KD-vote	82.46	74.09	84.47	77.02	78.05	67.99	77.83	65.79	69.39	55.31	80.78	71.44	78.83	68.61
✓	hard-KD-max	82.35	74.16	85.13	77.73	77.62	67.45	78.19	66.42	69.49	54.68	80.79	71.52	78.93	68.66
✓	hard-KD-mean	82.69	74.61	84.85	77.41	78.11	68.45	78.23	66.45	69.88	56.04	81.15	72.08	79.15	69.17
✓	UMM	82.89	73.44	83.02	73.24	78.28	63.21	75.36	61.38	66.85	49.13	80.40	70.84	77.80	65.21
✓	Self-training[1]	83.89	74.64	83.76	74.10	79.01	63.31	77.56	63.31	67.95	50.39	80.78	72.20	78.82	66.33
✓	Tri-training [2]	83.97	74.64	83.80	75.34	79.17	63.49	77.94	63.89	68.35	51.07	80.51	71.84	78.96	66.71
✓	soft-KD-avg[3]	82.07	74.64	84.80	77.82	78.18	68.73	78.27	67.46	68.90	54.84	80.83	72.12	78.84	69.27
✓	soft-KD-sim[4]	81.49	72.46	85.49	78.39	77.59	67.90	78.28	67.38	68.63	54.58	80.93	72.19	78.74	68.82
✓	Ours-lang	82.07	74.67	84.94	78.03	78.26	68.76	78.62	67.78	68.66	54.49	81.10	72.62	78.94	69.39
✓	Ours-sent	82.33	74.89	85.25	78.10	78.62	69.03	78.74	67.91	69.06	56.13	81.19	72.54	79.20	69.77
✓	Ours-sub	83.95	**76.67**	**86.00**	**79.25**	**79.41**	**70.13**	**79.40**	**68.58**	**72.36**	**60.21**	**82.15**	**73.70**	**80.54**	**71.42**

[1] Yarowsky (1995); McClosky et al. (2006) [2] Ruder and Plank (2018) [3,4] Wu et al. (2020)

Figure 11: Results on Open-source Cross-lingual Transfer (Parsing)

2.6 On Building Unified NER Model?



Structure-Level Knowledge Distillation For Multilingual Sequence Labeling. ACL 2020

Structural Knowledge Distillation: Tractably Distilling Information for Structured Predictor. ACL 2021

Motivation: Why Unified Multilingual Sequence Labeling?

- Training and serving multiple monolingual models online is time consuming.
- A unified multilingual model: smaller, easier, more generalizable.

General Approach: Many-to-one Knowledge Distillation

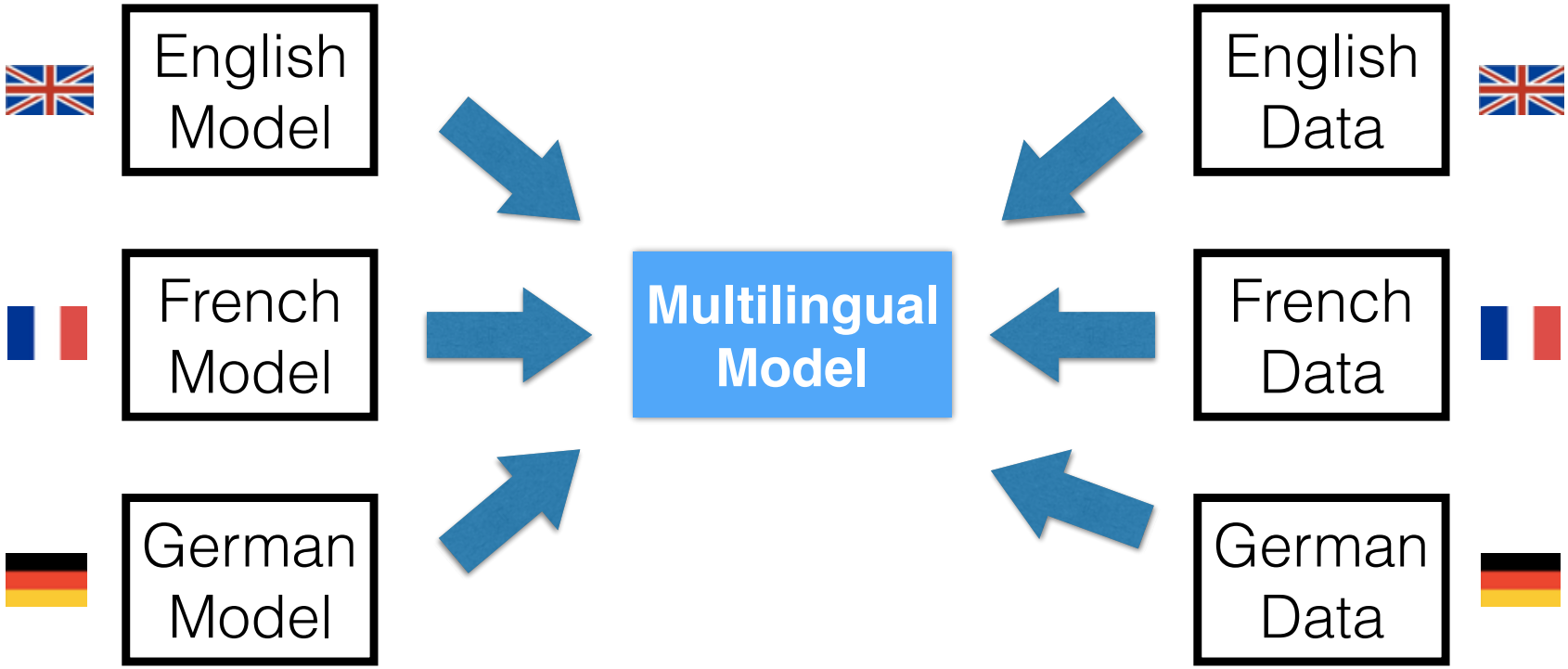


Figure 12: Many-to-one Knowledge Distillation

$$\mathcal{L}_{\text{ALL}} = \underbrace{\lambda \mathcal{L}_{\text{KD}}}_{\text{hard!}} + \underbrace{(1-\lambda)\mathcal{L}_{\text{NLL}}}_{\text{easy}}$$

We consider to **approximate** the green term.

Methods: Full Picture

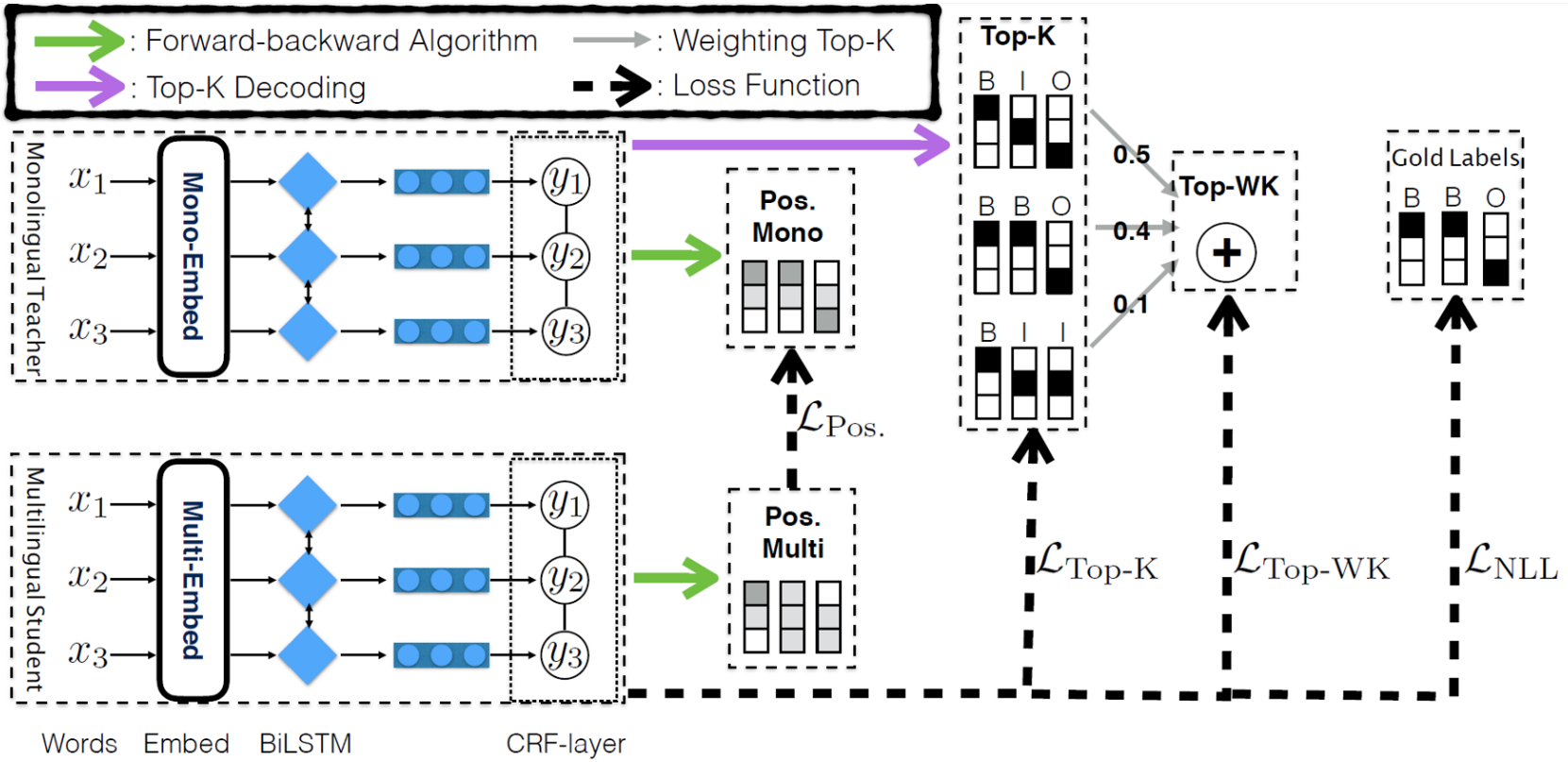


Figure 13: Unified Multilingual NER training

[Structure-Level Knowledge Distillation For Multilingual Sequence Labeling. Wang, **Jiang**, Bach, Wang, Huang, Tu, ACL 2020]

Experiments on High Resource Languages

Task	CoNLL NER	Aspect Extraction	WikiAnn NER	UD POS
TEACHERS	89.38	70.20	88.97	96.31
BASELINE	87.36	66.54	87.48	94.06
EMISSION	87.55	65.79	87.43	94.13
TOP-K	87.62	67.18	87.53	94.12
TOP-WK	87.64	67.22	87.57	94.14
POSTERIOR	87.72	**67.49**	**87.83**	**94.29**
POS.+TOP-WK	**87.77**	67.34	87.71	94.20

- Monolingual teacher models outperform multilingual student models.
- Our approaches outperform the baselines.

Experiments on Zero-shot Transfer

	NER	POS
TEACHERS	41.85	56.01
BASELINE	50.86	84.11
EMISSION	50.19	84.17
POSTERIOR	**51.43**	**84.28**
POSTERIOR+TOP-K	51.14	84.24

Table 6: Averaged results of zero-shot transfer on another 28 languages of the NER task and 24 languages of the POS tagging task.

Conclusions:

- Our approaches improve the performance of multilingual models over 4 tasks on 25 datasets.
- Our distilled model has stronger zero-shot transfer ability on the NER and POS tagging tasks.

Structural KD. [ACL 2021]

$$\mathcal{L}_{\text{KD}} = -\sum_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} P_t(\mathbf{y}|\mathbf{x})\log P_s(\mathbf{y}|\mathbf{x})$$

$$= -\sum_{\mathbf{u}\in\mathbf{U}_s(\mathbf{x})} P_t(\mathbf{u}|\mathbf{x})\text{Score}_s(\mathbf{u},\mathbf{x}) + \log\mathcal{Z}_s(\mathbf{x})$$

	de	en	es	nl	Avg.
Teachers	84.00	92.43	89.19	91.90	89.38
Student (w/o KD)	82.16	90.13	88.06	89.11	87.36
ACL 2020: Top-WK KD	82.15	90.52	88.64	89.24	87.64
ACL 2020: Pos. KD	82.22	90.68	88.57	89.41	87.72
ACL 2020: Pos.+Top-WK	**82.31**	90.53	88.66	89.58	87.77
ACL 2021: Struct. KD	82.28	**90.86**	**88.67**	**90.07**	**87.97**

Table 2: A comparison of KD approaches for multilingual NER.

Take-Away Message

Main Ideas:

- X-lingual embedding (mBERT/XLMR) has strong X-lingual transfer ability.
- Monolingual models are good at language-specific tasks.
- Knowledge distillation bridges the gap between them.

Take-Away Message

Main Ideas:

- X-lingual embedding (mBERT/XLMR) has strong X-lingual transfer ability.
- Monolingual models are good at language-specific tasks.
- Knowledge distillation bridges the gap between them.

Questions:

- Can the approach adapt to other settings? Yes.
  Semi-supervised NER, X-lingual KD.

Summary

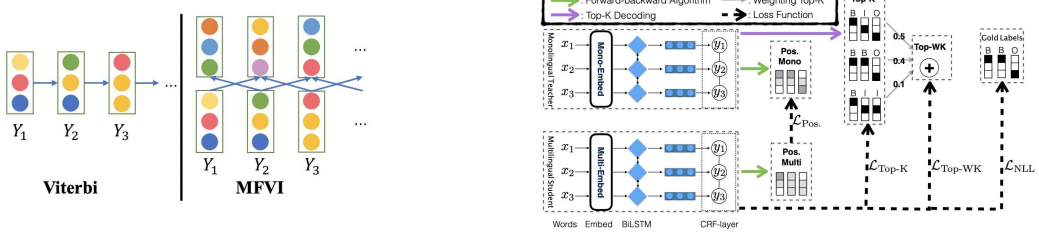
Research Summary for Mono-lingual NER

EMNLP 2020 | ACL 2021 | ACL 2021

BERT | CombEmbed | ACE | ContextEnrich

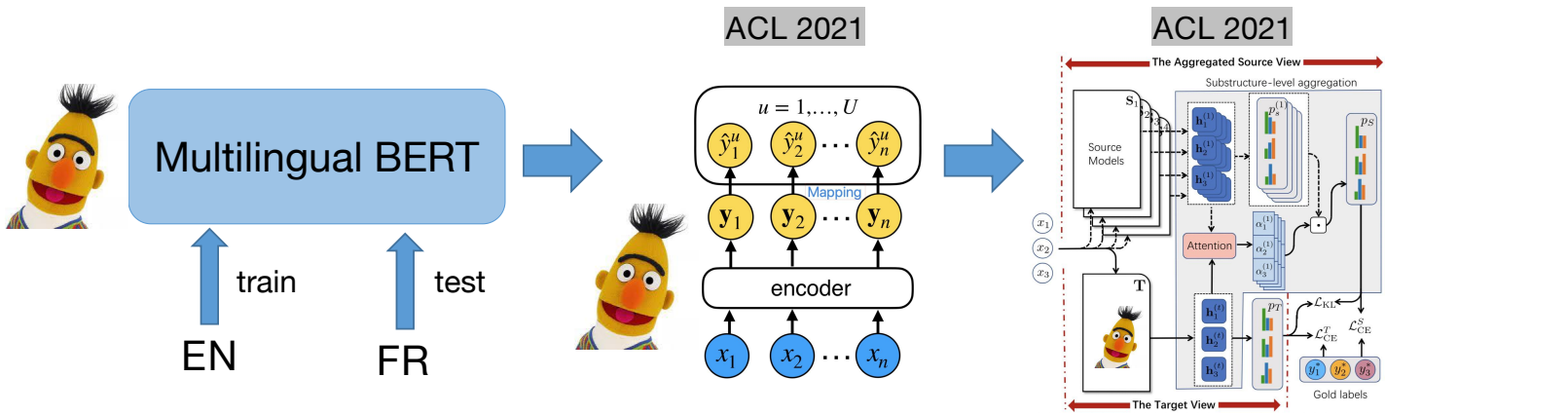
Speed Up Inference

EMNLP 2020 | ACL 2020/2021



AIN | Structure-level KD

Research Summary for Multi-lingual NER



Direct Transfer

Risk Minimization (Uncertainty Modeling)

Multi-View Train (Clever Knowledge Transfer)

Future Work

For mono-lingual NER:

- 'Finetuning' the search system to improve NER models.

For cross-lingual NER:

- Utilizing external resources (translation, dictionary...) to improve NER models

For knowledge distilled NER:

- further bridge the gap between teacher models & student models.

Backup Slides

Analysis on ACE

Comparing with random search:

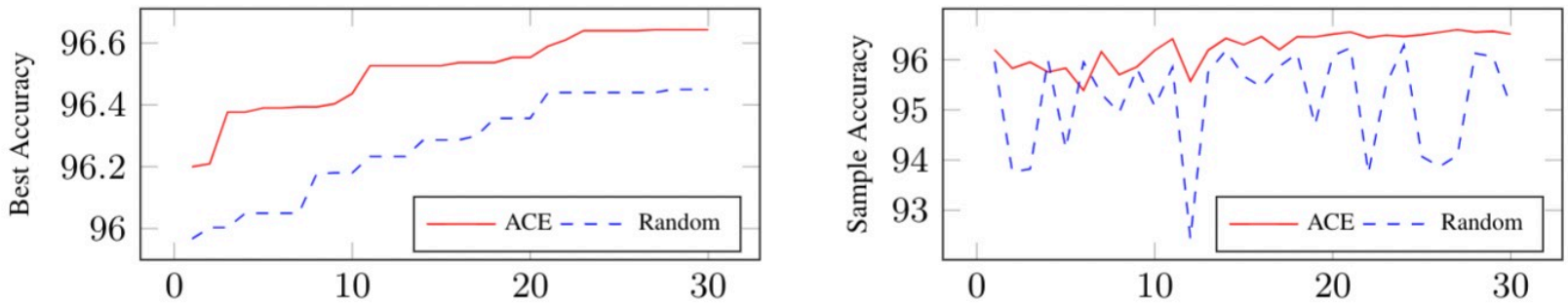


Figure 2: Comparing the efficiency of random search (`Random`) and ACE. The x-axis is the number of time steps. The left y-axis is the averaged best validation accuracy on CoNLL English NER dataset. The right y-axis is the averaged validation accuracy of the current selection.

Ablation study on reward function design:

	Dev	**Test**
ACE	**93.18**	**90.00**
No discount	92.98	89.90
Simple	92.89	89.82

Comparisons with Other Baselines & with Doc Information

Table 6: A comparison among `All`, `Random`, ACE, `All+Weight` and `Ensemble`. CHK: chunking.

	NER	POS	AE	CHK	DP		SDP	
					UAS	LAS	ID	OOD
All	92.4	90.6	73.2	96.7	96.7	95.1	94.3	90.8
Random	92.6	91.3	74.7	96.7	96.8	95.2	94.4	90.8
ACE	**93.0**	**91.7**	**75.6**	**96.8**	**96.9**	**95.3**	**94.5**	**90.9**
All+Weight	92.7	90.4	73.7	96.7	96.7	95.1	94.3	90.7
Ensemble	92.2	90.6	68.1	96.5	96.1	94.3	94.1	90.3
$\text{Ensemble}_{\text{dev}}$	92.2	90.8	70.2	96.7	96.8	95.2	94.3	90.7
$\text{Ensemble}_{\text{test}}$	92.7	91.4	73.9	96.7	96.8	95.2	94.4	90.8

Table 7: Results of models with document context on NER. +sent/+doc: models with sentence-/document-level embeddings.

	de	de_{06}	en	es	nl
All+sent	86.8	90.1	93.3	90.0	94.4
ACE+sent	87.1	90.5	93.6	92.4	94.6
BERT (2019)	-	-	92.8	-	-
Akbik et al. (2019)	-	88.3	93.2	-	90.4
Yu et al. (2020)	86.4	90.3	93.5	90.3	93.7
All+doc	87.6	91.0	93.5	93.3	93.7
ACE+doc	**88.0**	**91.4**	**94.1**	**95.6**	**95.5**

- ACE outperforms `Ensemble` methods, consistently.
- Context (document) information is important for NER. (We may go deeper in this direction.)

Ali wins xxx. Yusong beats yongjiang.

LOC or ORG or PER ??

ORG or PER ??

Comparing with Retraining

Table 12: A comparison among retrained models, All and ACE. We use the one dataset for each task.

	NER	POS	Chunk	AE	DP-UAS	DP-LAS	SDP-ID	SDP-OOD
All	92.4	90.6	96.7	73.2	96.7	95.1	94.3	90.8
Retrain	92.6	90.8	**96.8**	73.6	96.8	95.2	**94.5**	**90.9**
ACE	**93.0**	**91.7**	**96.8**	**75.6**	**96.9**	**95.3**	**94.5**	**90.9**

- Surprisingly, ACE even outperforms retraining sometimes.
- One possible reason is that ACE may generalize well by learning the bias of other embeddings.

Model

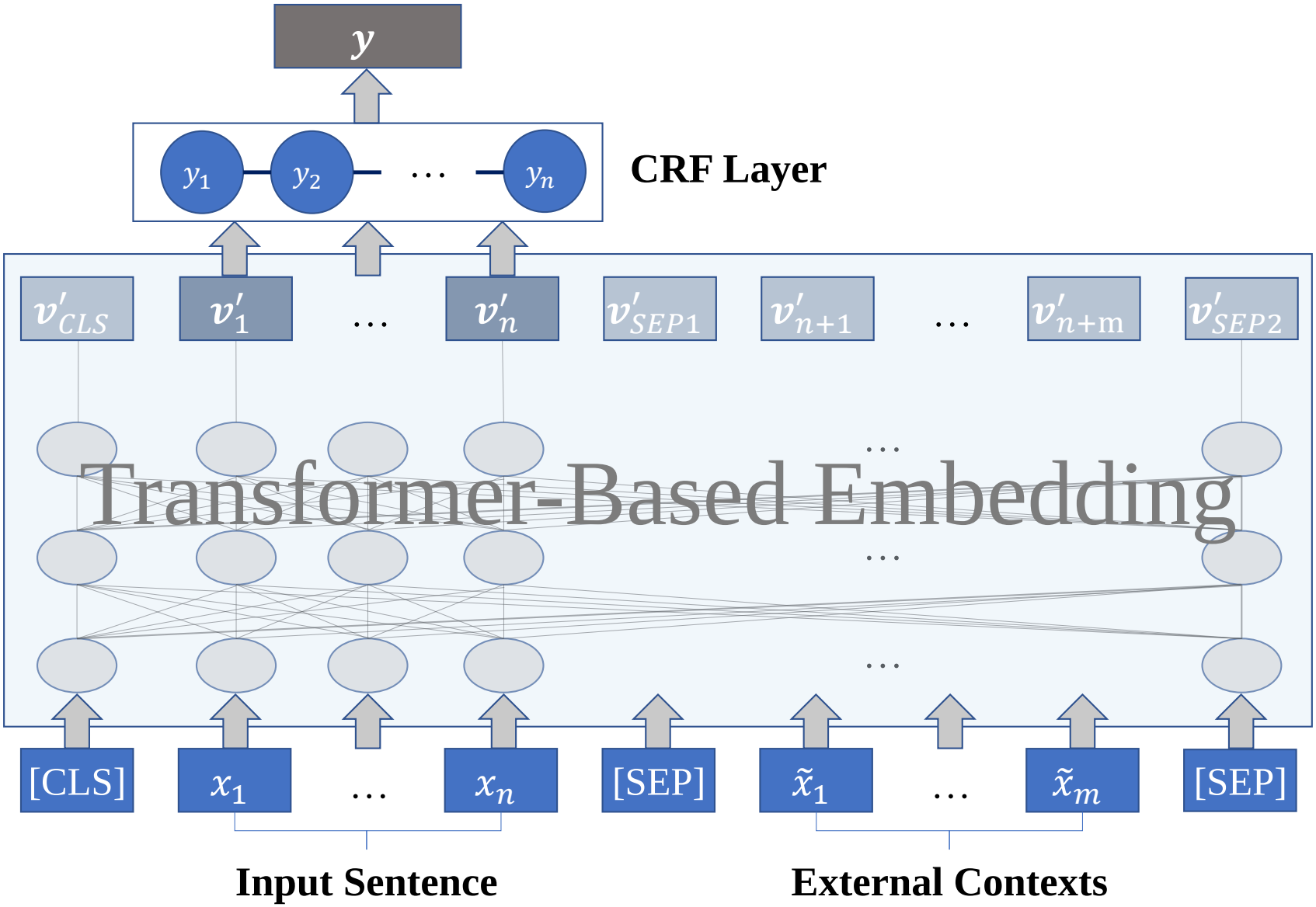


Figure 14: Retrieving External Contexts