

## 연구 소개

- 적은 학습데이터로 더 좋은 성능을 낼 수 있는 의존구조 분석 모델 연구

The slide bases on two papers :

(1) “Semi-Supervised Learning on Meta Structure: Multi-Task Tagging and Parsing in Low-Resource Scenarios” presented in AAAI 2020.

(2) “SEx BiST: A Multi-Source Trainable Parser with Deep Contextualized Lexical Representations” presented in CoNLL 2018 shared task.

07. Jan. 2020

Presenter: 임경태

한국원자력연구원 지능형컴퓨팅연구실

# Contents

I. 연구분야 소개

II. 연구개요

III. 제안방법

IV. 실험결과

V. 마치면서

V. 참조 (수식)

## 대표 연구 소개

- Bootstrap Methods based on Semi-supervised and Meta Structure  
for Dependency Parsing in Low-resource scenarios
- 적은 데이터로 더 좋은 성능을 낼 수 있는 의존구조 분석 (기계학습) 모델 연구

# 연구 대분류 소개

## ▶ 자연어처리(Natural Language Processing)란?

- 자연어 처리는 컴퓨터가 인간의 말을 이해할 수 있도록 처리하는것을 의미
  - 음악 틀어달라고 시키고 (음성인식)
  - 번역 숙제 대신해달라고 시키고 (기계번역)
  - 궁금한걸 물어보고 (질의응답)
  - 심심하면 대화도하고 (Discourse)

그래서 인공지능의 완성은 기계가 "언어를 완벽히 이해" 하는것이 라고함

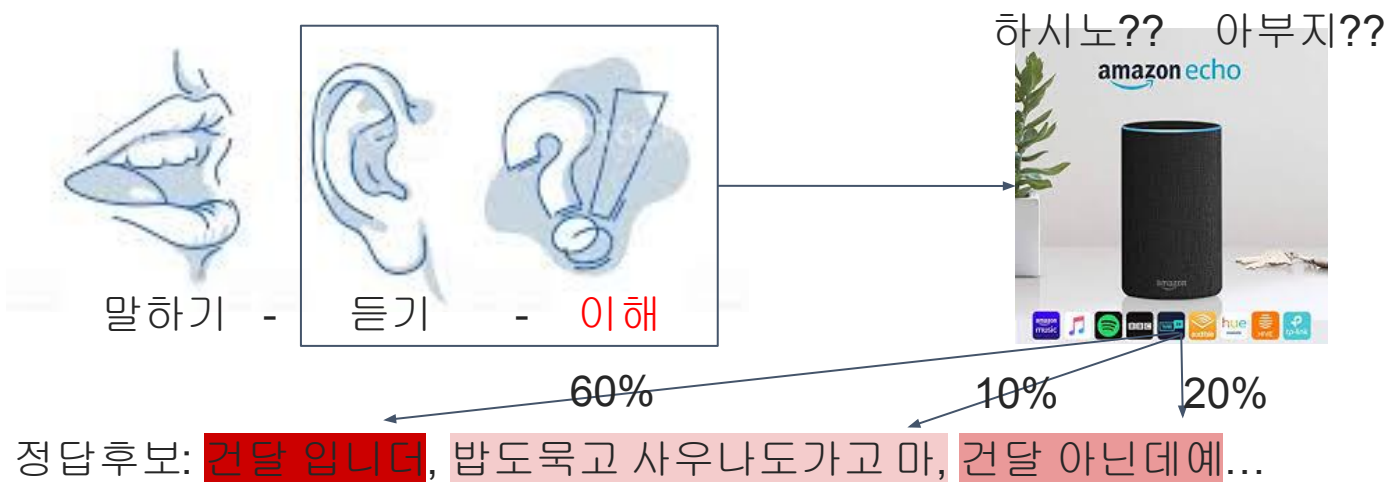
# 연구 대분류 소개

## ▶ 자연어처리의 예제 (질의응답)?

자연어처리: 기계가 "언어를 이해할 수 있도록" 분석/처리 하는 것

예를들어 영화(데이터)만 본 **echo**에게 질문을 하면

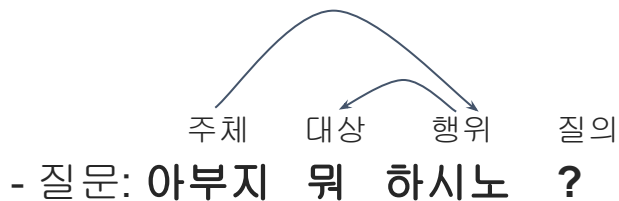
- 질문: 아버지 뭐 하시노?



# 연구 대분류 소개

## ▶ 자연어처리의 어려움

- 기계가 인간의 언어를 이해하기 위해 두 가지 큰 문제점을 해결해야함.
  - **문법적(Syntactic) 모호성**: 확률적 관점에서 생각해 봤을때, 문장이 길어질수록 단어조합이 문법적으로 엄청나게 복잡해질수 있음.  
예제) "철수가방에 들어간다"의 의미는?
  - **의미적(Semantic) 모호성**: 앞서말한 문법적 모호성과 더불어, 같은 말도 주변 상황, 그리고 상식에 따라 의미가 달라질수 있다. (**Contextual Information and Common Sense**)
  - 두 가지 문제를 해결하기 위한  
좋은 방법이 없을까?  
문제를 작게 쪼개보자!



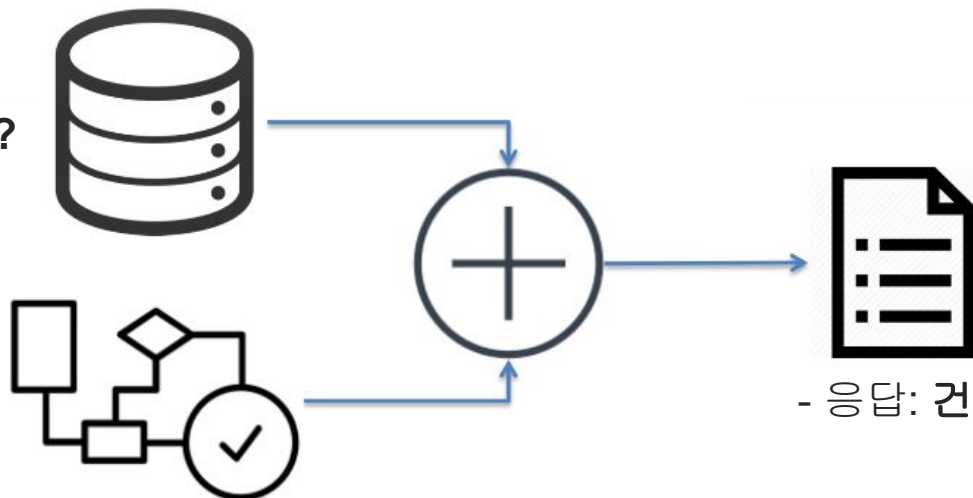
# 연구 대분류 소개

## > 자연어처리의 전통적 방식

- 인간이 먼저 지식과 규칙을 명시해둠.

명시적 프로그래밍

- 입력: 아버지 뭐 하시노?



- 응답: 건달 입니더

지식: 아버지=아버지=주체...

**Rule:** 주체 “has a” 행위...

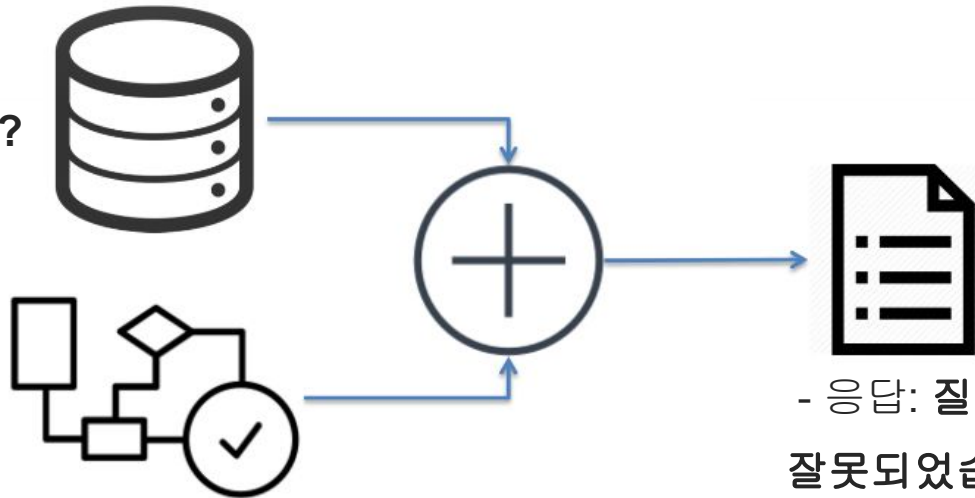
# 연구 대분류 소개

## 자국어처리의 전통적 처리방식

- 인간이 먼저 지식과 규칙을 명시해둠. 기계는 잘못이 없음

명시적 프로그래밍

- 입력: 아빠 뭐 하시노?



- 응답: 질문이  
잘못되었습니다.

지식: 아버지=아버지=주체...

**Rule:** 행위 “needs a” 주체...



# 연구 대분류 소개

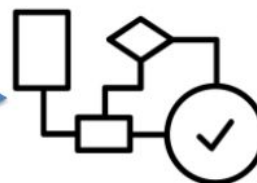
## 자국어처리의 기계학습 방식

- 기계가 지식과 규칙을 스스로 찾음. 틀리면 혼나면서 배움

- 입력: 아빠 뭐 하시노?



- 응답: 건달 입니더



학습된지식: **아빠=건달**=(주체)....,

학습된Rule: 입니더 “needs a” (주체)

하지만, 기계가 “아버지 뭐 하시노?” 를 단 한번도 듣지 못했다면?

“아버지”가 주체가 될 수 없고 따라서 아버지는 건달이 될 수 없다...

# 연구 대분류 소개

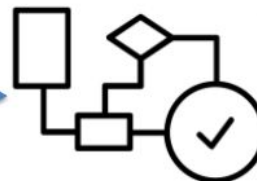
## ▶ 자연어처리의 기계학습 방식

- 기계가 지식과 규칙을 스스로 찾음. 틀리면 혼나면서 배움

- 입력: 아빠 뭐 하시노?



- 응답: 건달 입니더



학습된지식: 아빠=건달=(주체)=(주어)....,

학습된Rule: 입니더 “needs a” (주어)

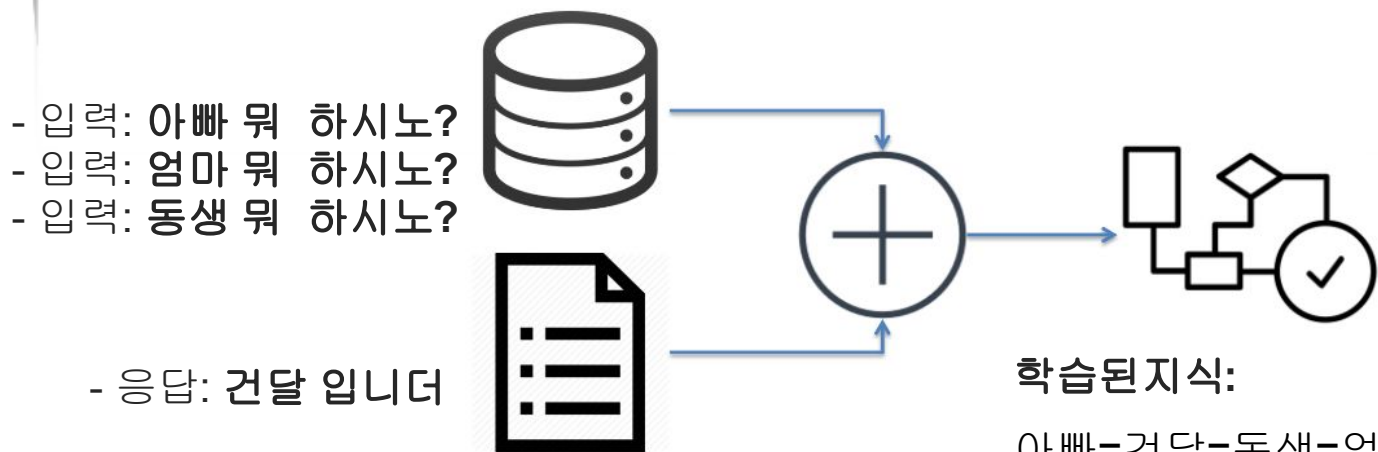
하지만, 기계가 “아버지 뭐 하시노?” 를 단 한번도 보지 못했다면?

주체(추상적의미) = 주어 로 일반할 경우, “아버지”는 하나의 주체로 파악가능

# 연구 대분류 소개

## 자국어처리의 기계학습 방식

- 기계가 **(많은 데이터로부터)** 지식과 규칙을 스스로 찾음. 틀리면 혼나면서 배움



학습된지식:

아빠=건달=동생=엄마=(사람)=(주체)=주어

학습된Rule: 입니더 “needs a” (주어)=(사람)

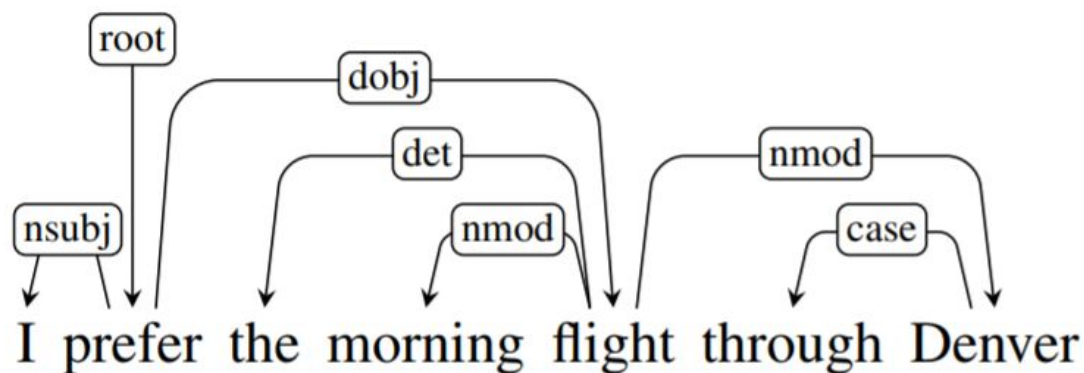
기계가 “아버지 뭐 하시노?” 를 단 한번도 듣지 못했다 하더라도!

입니더 “needs a” (사람) = (주어) 을 통해 유추가 가능해짐.

# 연구 소분류 개요

## ➤ 무엇에 관한 연구인가?

- **Dependency Parser란?**: 문장의 의존구조 분석기. 단어의 관계를 분석하여 기계가 문장의 의미를 파악 할 수 있게 돕는다.
- **왜 중요한가?**: 기계 독해, 정보추출, 챗봇 등 하위 **task**의 핵심적인 전처리 분석기로 쓰임
- **어떻게 만드는가?**: 지도학습 (Supervised learning)을 통한 의존관계 모델링

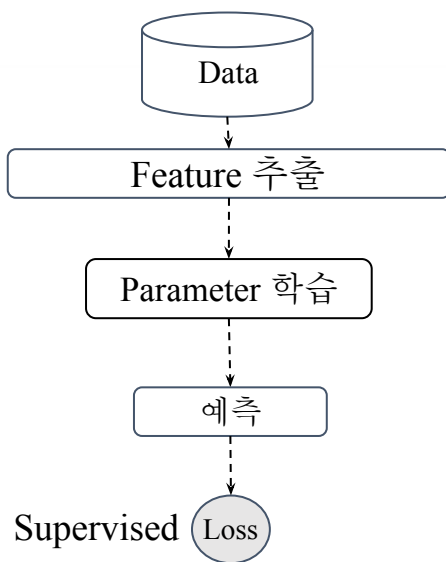


- Dependency Parsing 예제

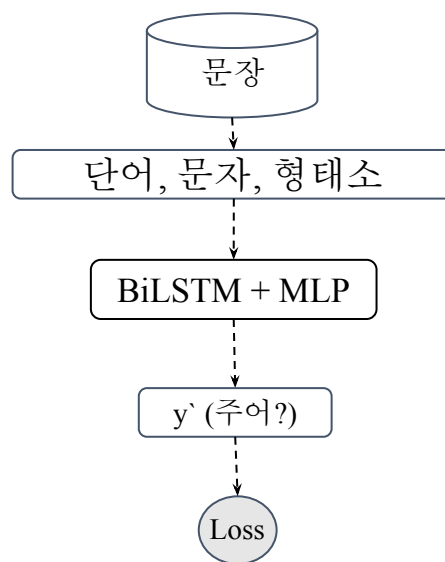
# 개요

## 기존 시계열기반 데이터의 Supervised learning의 문제점

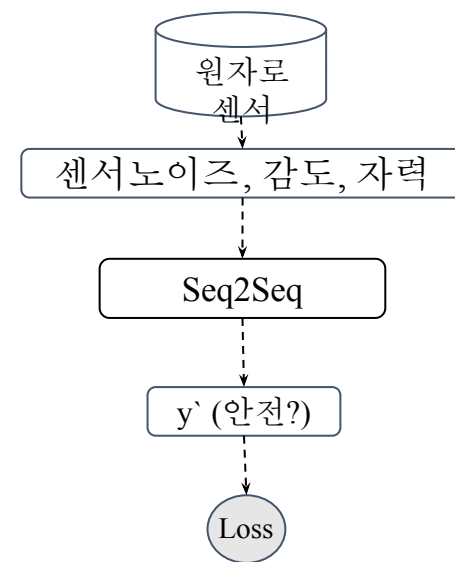
- 시계열 데이터 모델링은 대부분 지도학습 (Supervised learning)에 의존하고 있음
  - 학습 데이터가 필요하며 자연어 처리는 데이터 제작에 많은 시간과 비용이 소요



• Supervised learning



• Parsing



• Engineering

# 문제의 제기

## ➤ 적은 데이터로 더 좋은 성능을 낼 수 있을까?

- 시계열 데이터 모델링은 대부분 지도학습 (Supervised learning)에 의존하고 있음
  - 학습 데이터가 필요하며 자연어 처리는 데이터 제작에 많은 시간과 비용이 소요

```
# sent_id = 2
# text = I have no clue.
1  I      I      PRON   PRP    Case=Nom|Number=Sing|Person=1    2  nsubj  -  -
2  have   have   VERB   VBP    Number=Sing|Person=1|Tense=Pres  0  root   -  -
3  no     no     DET    DT     PronType=Neg                     4  det    -  -
4  clue   clue   NOUN   NN     Number=Sing                      2  obj    -  SpaceAfter=No
5  .      .      PUNCT  .      -                                2  punct  -  -
```

### • Parser의 학습데이터 (labeled) 예제

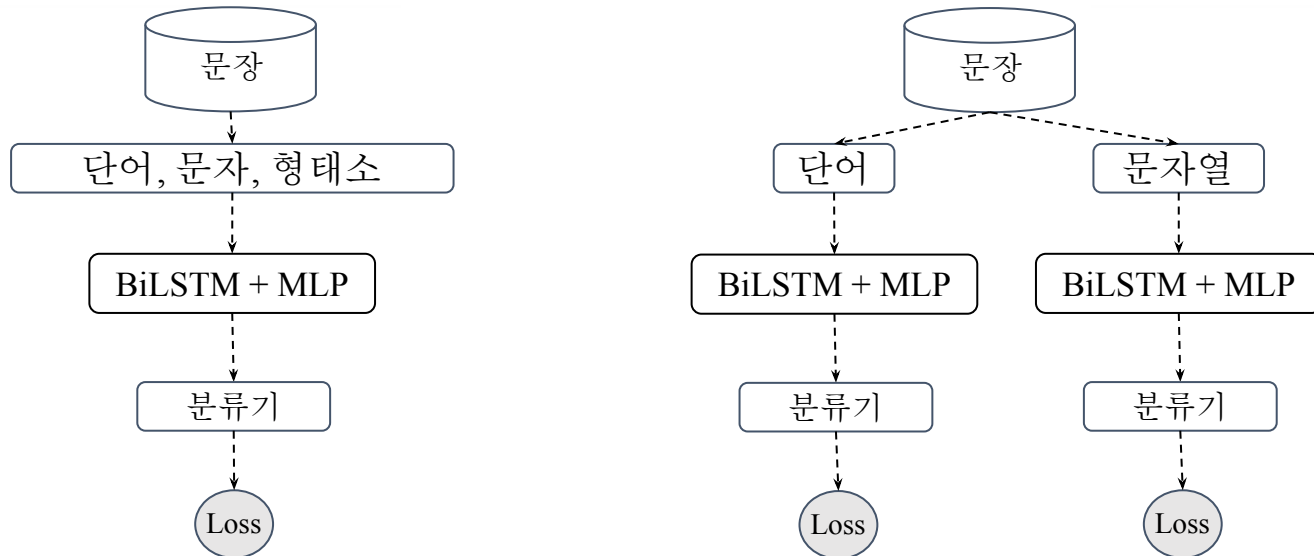
- 본 연구에서는 두 가지 방법을 제시
  - 학습 데이터의 자질(feature)을 최대한으로 활용하자 (Multi-view learning)
  - Unlabeled 데이터를 활용하자 (Co-training)

## 제안 1: Multi-view learning

# 제안 1 (Multi-view)

## ➤ Multi-view Learning으로 학습 데이터의 자질을 최대한 활용해보자

- **Multi-view**: 각 데이터의 자질(feature)을 의미, (예, 센서노이즈, 감도, 자력)
- **Multi-view learning**: 자질들을 조합해 각기 다른 **view**(관점)의 분류기를 학습하는 방법
- 자연어처리의 멀티뷰: 단어(enjoyed), 문자열(en, joy, e,d..), 형태소(동사)





# 제안 1 (Multi-view)

## 언어처리에서 Multi-view learning의 역할은?

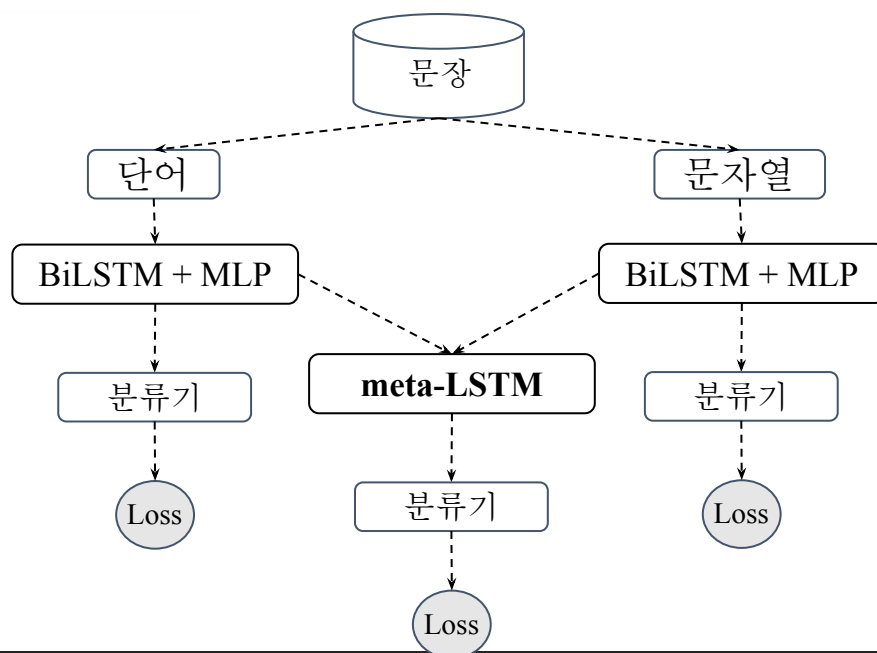
- 각 자질은 서로다른 의미 (semantic), 문법 (syntactic) 정보를 내포하며, 단어자질은 의미역 정보 전달에 효과적이고, 문자열 자질은 문법적 정보 전달에 효과적임

	In Vocabulary				
	<i>while</i>	<i>his</i>	<i>you</i>	<i>richard</i>	<i>trading</i>
LSTM-Word	<i>although</i>	<i>your</i>	<i>conservatives</i>	<i>jonathan</i>	<i>advertised</i>
	<i>letting</i>	<i>her</i>	<i>we</i>	<i>robert</i>	<i>advertising</i>
	<i>though</i>	<i>my</i>	<i>guys</i>	<i>neil</i>	<i>turnover</i>
	<i>minute</i>	<i>their</i>	<i>i</i>	<i>nancy</i>	<i>turnover</i>
LSTM-Char (before highway)	<i>chile</i>	<i>this</i>	<i>your</i>	<i>hard</i>	<i>heading</i>
	<i>whole</i>	<i>hhs</i>	<i>young</i>	<i>rich</i>	<i>training</i>
	<i>meanwhile</i>	<i>is</i>	<i>four</i>	<i>richer</i>	<i>reading</i>
	<i>white</i>	<i>has</i>	<i>youth</i>	<i>richter</i>	<i>leading</i>
LSTM-Char (after highway)	<i>meanwhile</i>	<i>hhs</i>	<i>we</i>	<i>eduard</i>	<i>trade</i>
	<i>whole</i>	<i>this</i>	<i>your</i>	<i>gerard</i>	<i>training</i>
	<i>though</i>	<i>their</i>	<i>doug</i>	<i>edward</i>	<i>traded</i>
	<i>nevertheless</i>	<i>your</i>	<i>i</i>	<i>carl</i>	<i>trader</i>

# 제안 1 (Multi-view)

## 어떤 Multi-view Learning 구조가 좋을까?

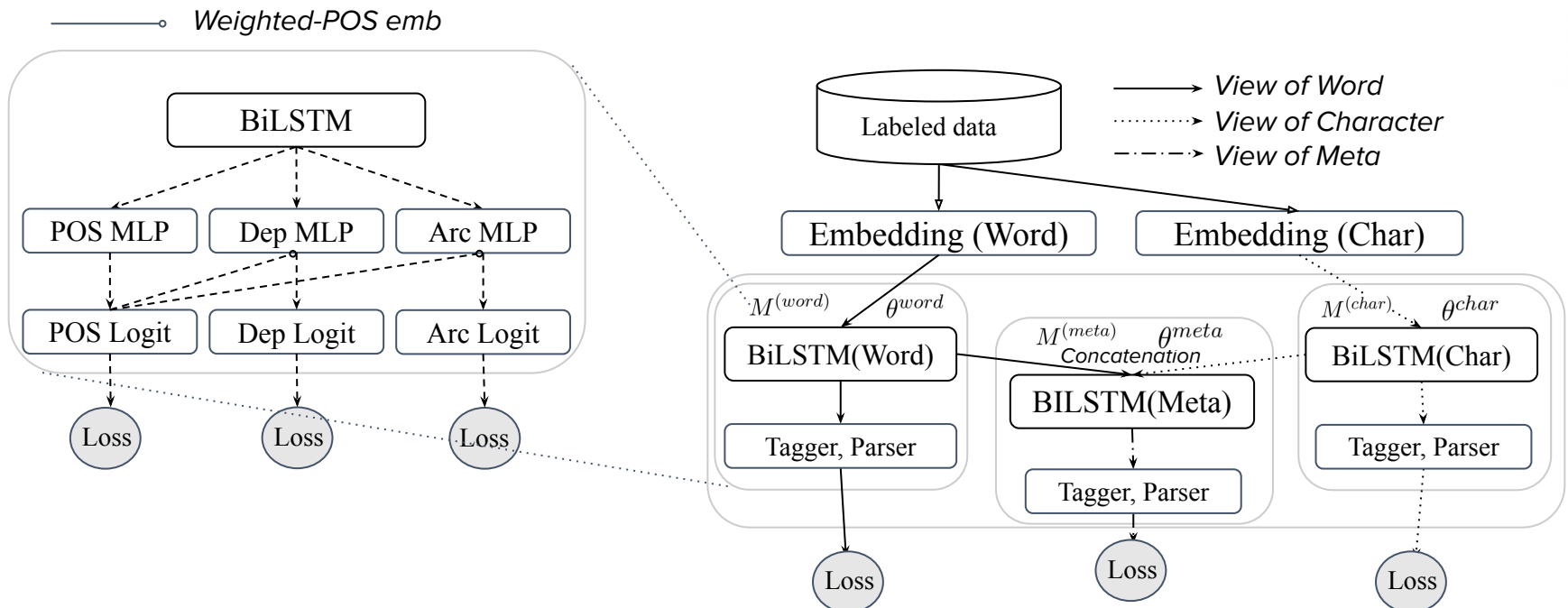
- Multi-view 기반의 자질을 연결하는 meta-LSTM 구조를 도입해 feature 활용을 최대화함
- 각 분류기는 제한된 자질로부터 최대한 높은 성능의 분류기를 학습하려고 노력함 따라서 미세한 변화에도 민감한 문맥 정보 (Context)를 학습



# 제안 1 (Multi-view)

## 최적의 multi-view 모델

- Multi-view 기반의 meta-LSTM 구조를 도입해 feature 활용을 극대화
- Multi-task learning을 활용해 Parser 학습 중 예측된 형태소 분석 정보를 포함



## 제안 2: Co-training

-

## 제안 2 (Co-training)

### ▶ 적은 데이터로 더 좋은 성능을 낼 수 있을까?

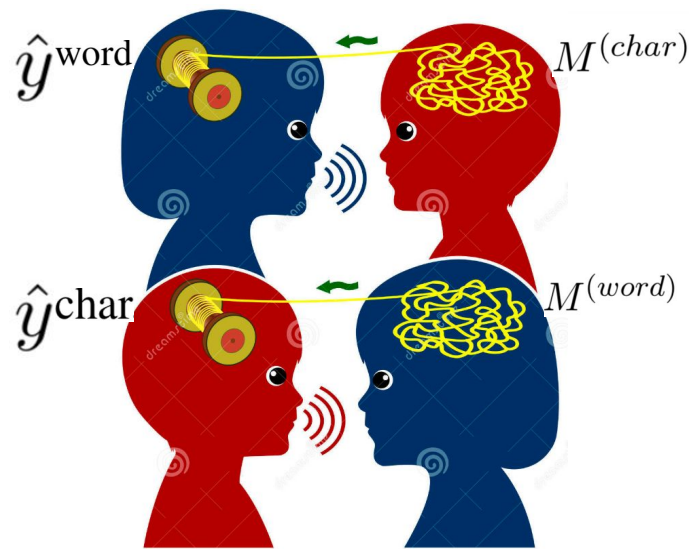
- 시계열 데이터 모델링은 대부분 지도학습 (Supervised learning)에 의존하고 있음
  - 학습 데이터가 필요하며 자연어 처리는 데이터 제작에 많은 시간과 비용이 소요
- 두 가지 가능성 제시
  - 제안1: 학습 데이터의 자질을 최대한으로 활용하자 (Multi-view learning)
  - **제안2: Unlabeled 데이터를 활용하자, Co-training:**
    - Co-training은 준지도 학습 (Semi-supervised learning) 방법 중 하나로 여러 개의 분류기가 서로 협동하여 학습하는 모델로 unlabeled 데이터를 학습데이터로 활용함.
    - 예를들어, 분류기1이 unlabeled 데이터로부터 예측한 정보를 정답으로 설정하고 분류기2를 학습!

## 제안 2 (Co-training)

### ▶ 두 모델중 누가 더 똑똑할까? (경쟁모델)

- Co-training을 적용할 때 누가 누구를 가르칠 것인가를 판단 하는 것이 중요
- Entropy 기반의 경쟁모델을 제안  $vi^* = \operatorname{argmax}_{vi \in V} P(\hat{y}^{vi} | x, \theta^{vi})$
- Entropy 점수는 “문장 A에 대해 해당 모델이 얼마나 confident 한가?”

- When the **word** model is confident than the **char**:  
set  $\hat{y}^{\text{word}}$  as a temporal gold label  $\hat{y}^*$  and update the *chat* model.



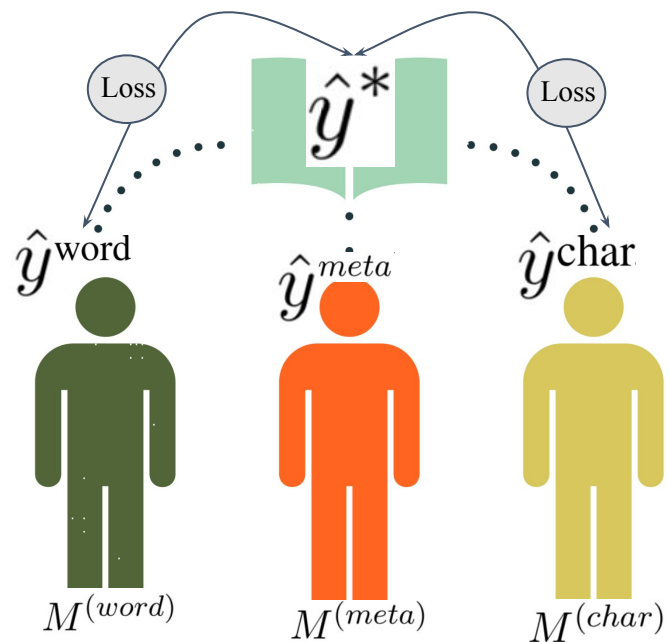
- Entropy model

## 제안 2 (Co-training)

### 어떤 예측이 최적일까? 문제를 셋이 같이 풀어보자 (협동모델)

- 세 모델이 합동하여 최적의 예측 결과를 찾고, 이를 정답으로 설정하고 학습
- Learns from the best possible probability:  $\hat{y}^* = \text{softmax}(\sum_{vi \in V} P(\hat{y}^{vi}|x, \theta^{vi}))$   
Or from the most voted tree.

- Set  $\hat{y}^*$  as a temporal gold label and update each model comparing to each prediction.



- Ensemble or Voting model

# 실험결과 (Official Results)

## ▶ CoNLL 2018 shared task의 57개 언어에 대한 실험결과

### • CoNLL shared task: 자연어처리 All treebanks

경진대회로 87개의 언어 모델에  
대한 종합평가로 순위 책정

- 6개월의 준비 기간 (코딩)
- 2달간의 본선평가 (모델 학습)
- 132개의 참가 팀 중 27개 팀이  
모든 결과 제출
- 언어마다 자질, 데이터 수,  
띄어쓰기 등 변수 존재
- UAS 2등으로 마감

1. HIT-SCIR (Harbin)	software1-P	80.51
2. LATTICE (Paris)	software1-P	78.71
3. ICS PAS (Warszawa)	software1-P	78.22
4. TurkuNLP (Turku)	software1-P	77.97
5. UDPipe Future (Praha)	software1-P	77.90
6. Uppsala (Uppsala)	software1-P	77.53
7. CEA LIST (Paris)	software1-P	77.39
8. Stanford (Stanford)	software2	76.78
9. AntNLP (Shanghai)	software1-P	76.31
10. NLP-Cube (București)	software1-P	76.16
11. ParisNLP (Paris)	software1-P	75.96
12. SLT-Interactions (Bengaluru)	software2-P	75.46
13. IBM NY (Yorktown Heights)	software1-P	74.72
14. UniMelb (Melbourne)	software1-P	74.16
15. LeisureX (Shanghai)	software1-P	74.03
16. KParse (İstanbul)	software1-P	72.75
17. Fudan (Shanghai)	software5-P	72.13
18. BASELINE UDPipe 1.2 (Praha)	software1-P	71.64
19. Phoenix (Shanghai)	software1-P	71.27
20. CUNI x-ling (Praha)	software1-P	70.87
21. BOUN (İstanbul)	software2-P	69.99
22. ONLP lab (Ra'anana)	software3-P	69.56
23. iParse (Pittsburgh)	software1-P	60.04
24. HUJI (Yerushalayim)	software1-P	59.93
25. ArmParser (Yerevan)	software1-P	50.86



# Results (CoNLL shared 2018)

Corpus	Method	UAS(Rank)	LAS(Rank)	Corpus	Method	UAS(Rank)	LAS(Rank)
af_afribooms		87.42 (7)	83.72 (8)	it_isdt	tr	92.41 (6)	89.96 (8)
grc_perseus	tr	79.15 (4)	71.63 (8)	it_postwita	tr	77.52 (6)	72.66 (7)
grc_proiel	tr	79.53 (5)	74.46 (8)	ja_gsd	tr, el	76.4 (6)	74.82 (6)
ar_padt		75.96 (8)	71.13 (10)	ja_modern		29.36 (8)	22.71 (8)
hy_armtdp	tr, mu	<b>53.56 (1)</b>	<b>37.01 (1)</b>	kk_ktb	tr, mu	39.24 (15)	23.97 (9)
eu_bdt		85.72 (7)	81.13 (8)	ko_gsd	tr, el	88.03 (2)	84.31 (2)
br_keb	tr, mu	43.78 (3)	23.65 (5)	ko_kaist	tr, el	<b>88.92 (1)</b>	86.32 (4)
bg_btb		92.1 (9)	88.02 (11)	knr_mg	tr, mu	38.64 (3)	27.94 (4)
bxr_bdt	tr, mu	36.89 (3)	17.16 (4)	la_ittb	tr	87.88 (8)	84.72 (8)
ca_ancora		92.83 (6)	89.56 (9)	la_perseus	tr	75.6 (3)	64.96 (3)
hr_set		90.18 (8)	84.67 (9)	la_proiel	tr	73.97 (6)	67.73 (8)
cs_cac	tr	93.43 (2)	91 (2)	lv_lvtb	tr	82.99 (8)	76.91 (11)
cs_fictree	tr	<b>94.78 (1)</b>	91.62 (3)	pcm_nsc	tr, mu	18.15 (21)	11.63 (18)
cs_pdt	tr	92.73 (2)	90.13 (7)	sme_giella	tr, mu	<b>76.66 (1)</b>	<b>69.87 (1)</b>
cs_pud	tr	89.49 (7)	83.88 (9)	no_bokmaal		91.4 (5)	88.43 (11)
da_ddt		85.36 (8)	80.49 (11)	no_nynorsk	tr	90.78 (8)	87.8 (11)
nl_alpino	tr	90.59 (2)	86.13 (5)	no_nynorskli	tr	76.17 (2)	68.71 (2)
nl_lassyemall	tr	87.83 (2)	84.02 (4)	cu_proiel		77.49 (6)	70.48 (8)
en_ewt	tr, el	<b>86.9 (1)</b>	84.02 (2)	fro_srcmf		91.35 (5)	85.51 (7)
en_gum	tr, el	<b>88.57 (1)</b>	<b>85.05 (1)</b>	fa_seraji		89.1 (7)	84.8 (10)
en_lines	tr, el	<b>86.01 (1)</b>	81.44 (2)	pl_lfg	tr	95.69 (8)	92.86 (11)
en_pud	tr, el	<b>90.83 (1)</b>	<b>87.89 (1)</b>	pl_sz	tr	92.24 (9)	88.95 (10)
et_edt		86.25 (7)	82.33 (7)	pt_bosque		89.77 (5)	86.84 (7)
fo_ofst	tr, mu	48.64 (9)	25.17 (17)	ro_rrt		89.8 (8)	84.33 (10)
fi_fib	tr	89.74 (4)	86.54 (6)	ru_syntagrus	tr	93.1 (4)	91.14 (6)
fi_pud	tr	90.91 (4)	88.12 (6)	ru_taiga	tr	<b>79.77 (1)</b>	74 (2)
fi_idt	tr	88.39 (6)	85.42 (7)	sr_set		90.48 (10)	85.74 (11)
fr_gsd	tr, el	<b>89.5 (1)</b>	86.17 (3)	sk_snk		86.81 (11)	82.4 (11)
fr_sequoia	tr, el	<b>91.81 (1)</b>	<b>89.89 (1)</b>	sl_ssj	tr	87.18 (10)	84.68 (10)
fr_spoken	tr, el	79.47 (2)	73.62 (3)	sl_sst	tr	63.64 (3)	57.07 (3)
gl_ctg	tr	84.05 (7)	80.63 (10)	es_ancora		91.81 (6)	89.25 (7)
gl_treegal	tr	78.71 (2)	73.13 (3)	sv_lines	tr	85.65 (4)	80.88 (6)
de_gsd		82.09 (8)	76.86 (11)	sv_pud	tr	83.44 (3)	79.1 (4)
got_proiel		73 (6)	65.3 (8)	sv_talbanken	tr	89.02 (4)	85.24 (7)
el_gdt		89.29 (8)	86.02 (11)	th_pud	tr, mu	0.33 (21)	0.12 (21)
he_hbt		66.54 (9)	62.29 (9)	tr_imst		69.06 (7)	60.9 (11)
hi_hdtb		94.44 (8)	90.4 (12)	uk_iu		85.36 (10)	81.33 (9)
hu_szeged		80.49 (8)	74.21 (10)	hsb_ufal	tr, mu	54.01 (2)	43.83 (2)
zh_gsd	tr, el	71.48 (5)	68.09 (5)	ur_udtb		87.4 (7)	80.74 (10)
id_gsd		85.03 (3)	77.61 (10)	ug_udt		75.11 (6)	62.25 (9)
ga_idt		79.13 (2)	69.1 (4)	vi_vtb		49.65 (6)	43.31 (8)

Table 1: Official experiment results for each corpus, where *tr* (Treebank), *mu* (Multilingual) and *el* (ELMo) in the column Method denote the feature representation methods used (see Section 2 and 3).

## Results on Extrinsic Parsing Evaluation (EPE 2018)

Task	Precision	Recall	F1(Rank)
Event Extraction	58.93	43.12	<b>49.80 (1)</b>
Negation Resolution	99.08	41.06	58.06 (12)
Opinion Analysis	63.91	56.88	60.19 (9)

Task	LAS	MLAS	BLEX
Intrinsic Evaluation	84.66 (1)	72.93 (3)	77.62 (1)

Table 4: Official evaluation results on three EPE task (see <https://goo.gl/3Fmjke>).

# 끝 마치면서

## ➤ 기계학습과 도메인 지식을 겸비한 시계열 데이터 처리 전문가

- 여러 도메인의 전문가와 함께 일하고 싶습니다.
  - 전산학분야 외의 전자, 기계공학 지식은 전무한 수준이며 해당 분야로 도메인 지식을 넓히는게 저의 개인적인 비전입니다.
  - 큰 욕심없이, NIPS, AAAI, ACL best paper한번 받아보는게 소원입니다.
  - 경진대회 공동 참가 및 새로운 AI관련 문제해결을 위한 Co-work은 언제나 환영합니다. [jujbob@gmail.com](mailto:jujbob@gmail.com)으로 연락주세요!

감사합니다!

Q/A

## Technical details

# Co-meta Training (Meta-LSTM)

## Objective function of META-BASE (Meta-LSTM model)

- We applied a negative log-likelihood objective function for each of three taggers and parsers where  $x$  is a sentence and  $y$  denote a set of labels (POS tag, Dependency head, dependency relation)

$$S_{\text{loss}} = \sum_{(x_j, y_j) \in T} -\log P(y_j | x_j, \theta)$$

which is simply the standard cross entropy loss in this case and  $\log P(y_j | x_j, \theta)$  stands for  $\sum_{vi \in V} \log P(y_j | x_j, \theta^{vi})$  for brevity.

$V = \{\text{word}, \text{char}, \text{meta}\}$

# Co-meta Training (Co-Training)

## Objective function of Co-training

- When we have Unsupervised set  $U$  then the loss only for Co-training is:

$$- \sum_{\substack{vi \in V \setminus \{\text{meta}\} \\ V = \{\text{word}, \text{char}, \text{meta}\}}} \sum_{x \in U} g(\hat{y}^*, \hat{y}^{vi}) \log P(\hat{y}^* | x, \theta^{vi})$$

- Predicted tree for each model
- The best possible tree
- A score function about “How confident the predictions are”

# Co-meta Training (Co-Training)

## Objective function of Joint-Loss

- When we have Training set  $T$  and Unlabeled set  $U$ , our Joint-loss is as:

$$\begin{aligned} J_{\text{loss}} = & \sum_{(x_j, y_j) \in T} -\log P(y_j | x_j, \theta) & \left. \vphantom{\sum_{(x_j, y_j) \in T}} \right\} & \bullet \text{ Supervised} \\ & - \sum_{vi \in V} \sum_{x_k \in U} g(\hat{y}_k^*, \hat{y}_k^{vi}) \log P(\hat{y}_k^* | x_k, \theta^{vi}) & \left. \vphantom{\sum_{vi \in V} \sum_{x_k \in U}} \right\} & \bullet \text{ Unsupervised} \end{aligned}$$

where  $U, T$  might be  $T \subseteq U$  when using  $T$  without labels. We call training with  $J_{\text{loss}}$  on the meta-LSTM structure as Co-meta.

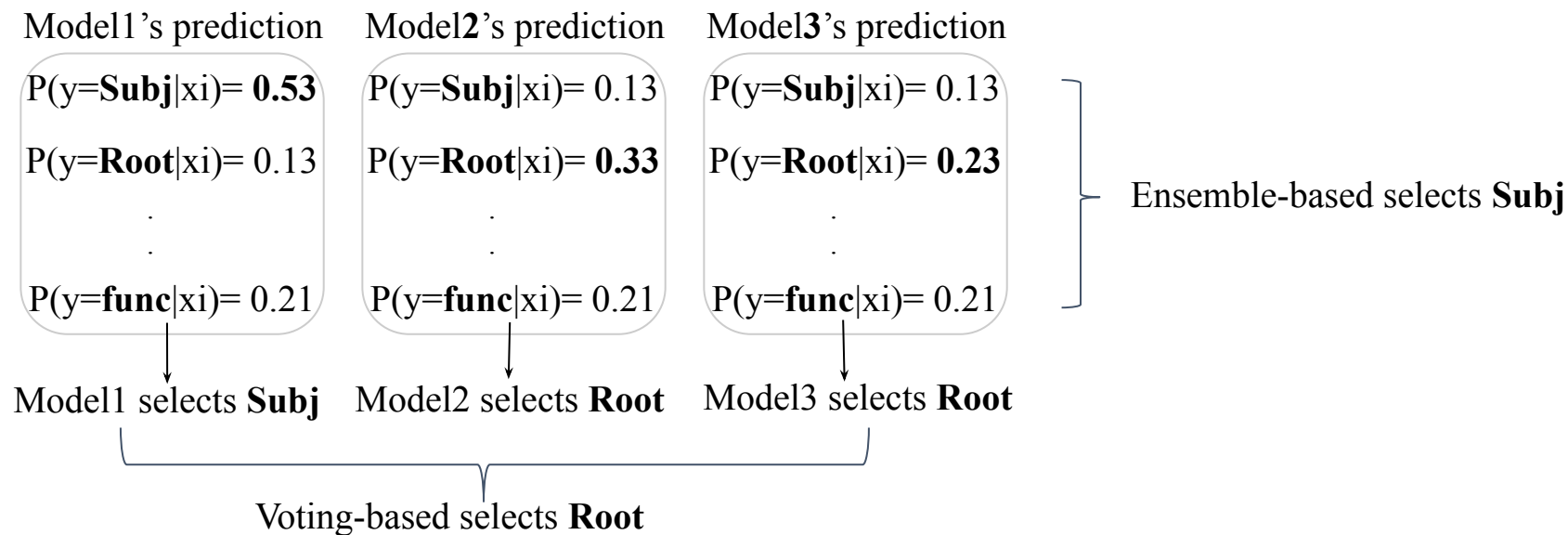


# Co-meta Training (Co-Training)

## Investigation of VOTING and ENSEMBLE approach

- **(Voting-based)** selects the most voted label among the three models for each word.
- **(Ensemble-based)** selects the most possible label by the average sum of the probabilities from three models i.g,  $\text{softmax}(\sum_{vi \in V} P(\hat{y}^{vi}|x, \theta^{vi}))$

[An example of two approaches when we have three models]



# Results

## Results based on the proposed approach

- We used only 50 sentences as the labeled set and the remaining sets as the unlabeled set among the training set.
- Our approach (co-meta) outperforms BASELINE (LATTICE shown in the second best in 2018 CoNLL shared Task), with -0.9~9.3 LAS score.

corpus	unlabeled	VOTING		ENTROPY		ENSEMBLE		META-BASE		BASELINE	
		LAS	POS	LAS	POS	LAS	POS	LAS	POS	LAS	POS
cs_cac (Czech)	23478	47.4	79.4	47.4	79.7	<b>48.7</b>	<b>81.4</b>	45.9	79.0	39.4	74.6
fi_ftb (Finnish)	14981	21.7	43.2	22.0	44.7	21.8	43.5	21.9	<b>44.6</b>	<b>22.6</b>	39.2
en_ewt (English)	12543	45.1	75.7	46.3	<b>76.7</b>	<b>46.5</b>	76.3	45.4	75.2	42.8	71.1
grc_perseus (Ancient Greek)	11460	30.8	70.1	<b>31.7</b>	<b>70.9</b>	31.3	70.7	30.9	70.4	29.5	65.8
he_htb (Hebrew)	5240	47.9	76.9	47.8	77.2	<b>48.4</b>	<b>77.4</b>	47.6	76.7	45.1	75.2
zh_gsd (Chinese)	3997	36.1	70.7	35.1	70.8	<b>36.9</b>	<b>71.1</b>	35.1	70.6	34.8	68.7
el_bdt (Greek)	1162	60.0	84.3	<b>60.6</b>	83.2	60.5	<b>84.2</b>	57.8	82.6	51.7	80.0
kk_ktb (Kazakh)*	12000*	27.6	56.9	27.9	57.0	<b>28.7</b>	57.1	27.8	<b>57.7</b>	26.2	53.0
Average	-	39.6	69.6	39.9	70.0	<b>40.3</b>	<b>70.2</b>	39.0	69.6	36.5	65.9

# Results



## Results on each model and confidence score

- Depending on the type of language, the char model shows better result than the word model (cs, el, grc), and vice versa (zh, en)
- A high CONFIDENCE (Function  $g$ ) denotes both the *word* and *char* model predict the similar structure of a tree.

Method	WORD	CHAR	META	CONFIDENCE
ENTROPY	<b>61.8</b>	66.7	<b>69.1</b>	0.871
ENSEMBLE	61.4	<b>66.9</b>	69.0	<b>0.879</b>
WITHOUT	57.6	65.2	67.4	0.799

Table 2: LAS on el\_bdt corpus for each model, with the average confidence score  $g(\hat{y})$  comparing  $M^{(word)}$  and  $M^{(char)}$  over entire test set.

# Results

## Results based on the domain of unsupervised data

- The result shows that selecting the same domain of unlabeled data is the most important for Greek (Quality-based).
- However, in the case of Chinese, the number of unlabeled set is still important (Quantity-based) since it has a bigger number of vocabulary than others.

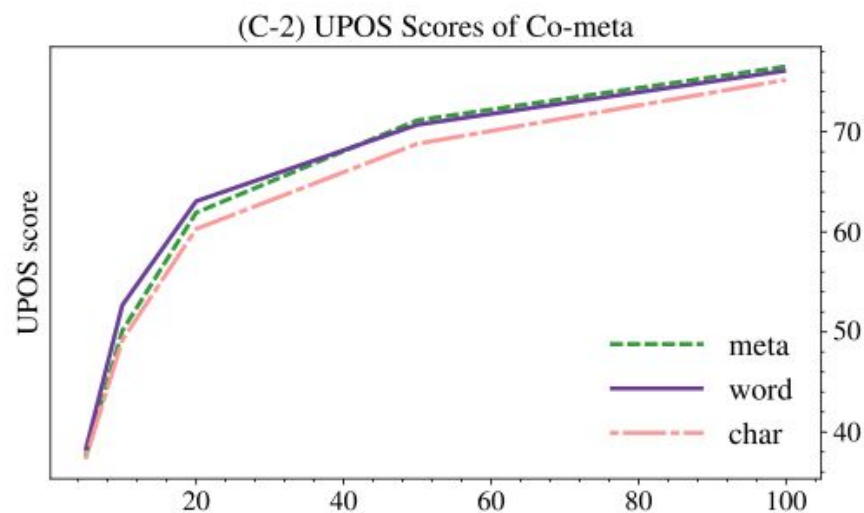
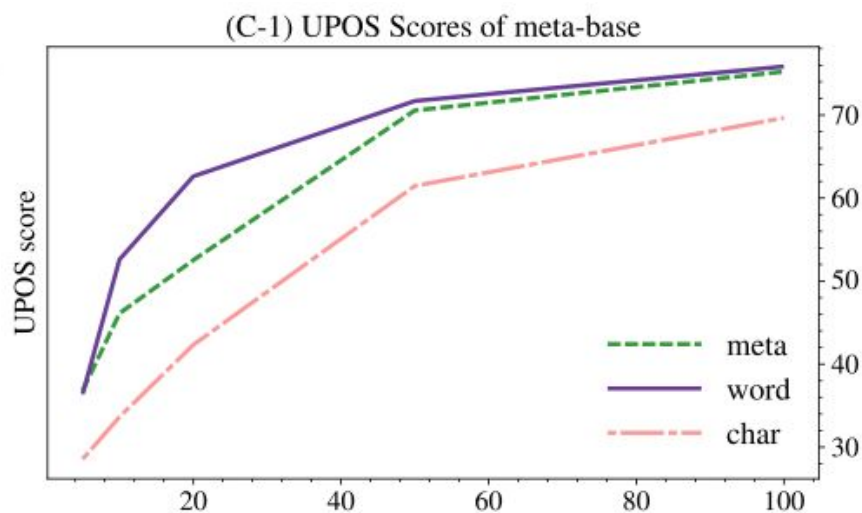
<b>Labeled</b>	<b>Unlabeled</b>	size	LAS	UAS	POS
el_bdt (Greek)	el_bdt	1162	<b>69.0</b>	<b>75.6</b>	88.5
	wikipedia	12000	68.7	75.1	<b>88.7</b>
	crawl	12000	68.3	74.8	88.4
zh_gsd (Chinese)	zh_gsd	3997	45.3	57.9	76.9
	crawl	12000	<b>46.1</b>	<b>59.0</b>	<b>77.8</b>

Table 3: Scores of  $M^{(meta)}$  with the ENSEMBLE method depending on the domain of unlabeled data.

# Results

## ➤ Impact on Co-meta learning

- Char and word model from Co-meta learns each other so the performance gap between two model is smaller than meta-base.



# Results

## Results on High-resource scenarios

Table 4: LAS on the English (en\_ewt) corpus for each model, with the external language models over the entire test set.

Model	LAS	UAS	POS
UDPIPE (2019)	86.97	89.63	96.29
BASELINE (2018)	86.82	89.63	<b>96.31</b>
METABASE	86.95	89.61	96.19
CO-META	<b>87.01</b>	<b>89.68</b>	96.17
BASELINE+ELMO (2018)	88.14	91.07	96.83
METABASE+ELMO	88.28	91.19	96.90
CO-META+ELMO	88.25	91.19	96.84
UDIFY (BERT-MULTI) (2019)	88.50	90.96	96.21
UUPARSER (BERT-MULTI) (2019)	87.80	-	-
BASELINE+BERT-MULTI	89.34	91.70	96.66
CO-META+BERT-MULTI	89.52	91.99	96.80
CO-META+BERT-BASE	<b>89.98</b>	<b>92.25</b>	<b>97.03</b>