Fine-Grained Named Entity Recognition

2021. 05. 21

발표자: 조 경 선

발표자 소개

❖ 이름: 조경선

- 고려대학교 산업경영공학과 석사과정(2020.09~)
- 데이터마이닝 및 품질애널리틱스 연구실(김성범 교수님)

❖ 관심연구분야

- Machine Learning & Deep Learning
- Natural Language Processing

❖ 연락처

- abc0323@korea.ac.kr

목차

Introduction

❖ Named Entity Recognition(NER) 이란?

- 이름을 가진 개체(named entity)를 인식하겠다는 것
- 사람, 기관, 장소, 의학 코드, 시간 표현, 양, 금전적 가치, 퍼센트 등 미리 정의된 분류로 텍스트의 개체명을 분류
- 다양한 Application에서 굉장히 중요한 Pre-processing 과정
  - ✓ Semantic Search, Question Answering, Machine Translation 등

Introduction

Michal Jeffrey Jordan was born in Brooklyn, New York.

⇩

Named Entity Recognition

⇩

Michal Jeffrey Jordan - Person

Brooklyn - Location

New York – Location

Introduction

❖ Named Entity Recognition(NER) 이 중요한 이유

- 주요 정보를 추출하여 **텍스트의 내용을 이해**하거나 **데이터베이스에 저장할 중요한 정보**를 수집

- 대규모 데이터 세트를 처리해야하는 경우, 비정형 데이터를 정렬하고 중요한 정보를 감지하는 데 도움

Introduction

❖ Named Entity Recognition(NER) 의 Use case

- **[CASE1] 고객 지원 상담 및 피드백**

- 고객 상담을 처리하는 경우 NER 기술을 사용하여 고객 요청을 더 빠르게 처리
  - ✓ 예) 제품 이름이나 일련 번호와 같은 관련 데이터 추출 → 해당 문제를 처리하는 데 가장 적합한 상담원이나 팀 배정
- 모든 고객 피드백을 구성하고 반복되는 문제 추출
  - ✓ 예) 부정적인 고객 피드백에서 가장 자주 언급되는 점을 감지 → 해당 문제를 처리하는 데 집중

Introduction

❖ Named Entity Recognition(NER) 의 Use case

- **[CASE2] 추천 시스템**


22 dead in 'callous terrorist attack'

Theresa May says the security services believe they know the attacker's identity - but are not yet able to confirm it.

3 minutes ago Manchester

LIVE

Manchester suicide blast: Latest updates

- Police: This was a terrorist attack
- 'I was thrown 30ft by the blast'
- Manchester: Subdued but strong
- Footage shows arena chaos

Witnesses' stories from Manchester

Manchester attacker 'evil loser' - Trump

Manchester Arena attack: What we know so far

Eyewitnesses tell of blast horror

In pictures: Manchester blast aftermath

- 뉴스 게시자는 유사한 기사를 사용자에게 추천
  - ✓ 예) 특정 기사의 엔티티 추출 → 엔티티가 일치하는 다른 기사 그룹화
- 컨텐츠 검색 기록을 기반으로 사용자에게 제안
  - ✓ 예) Netflix에서 코미디를 많이 시청하면 코미디 엔티티로 분류 된 더 많은 컨텐츠 추천

Introduction

❖ Named Entity Recognition(NER) 평가지표

- $\boldsymbol{Precision}$(정밀도) $= \frac{TP}{TP+FP}$

  ✓ 특정 Entity 라고 예측한 경우 중에서 실제 특정 Entity 로 판명되어 예측이 일치한 비율

- $\boldsymbol{Recall}$(재현률) $= \frac{TP}{TP+FN}$

  ✓ 전체 특정 Entity 중에서 실제 특정 Entity 라고 정답을 맞춘 비율

- $\boldsymbol{F-score} = \frac{Precision \times Recall}{Precision+Recall}$

  ✓ 정밀도와 재현률로부터 조화 평균(harmonic mean)을 구한 것

		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
Actual Class	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

- True Positive (TP): NER에 의해 인식되고 실제와 일치하는 Entity.
- False Positive (FP) : NER에 의해 인식되었지만 실제와 일치하지 않는 Entity.
- False Negative (FN): 참인 Entity이지만 NER에서 인식하지 못하는 Entity.

Deep Learning Techniques for NER

❖ The taxonomy of DL-based NER

1. Distributed Representations for Input

✓ 단어 및 문자 임베딩

✓ POS tag 및 색인와 같은 추가 정보 통합

2. Context Encoder Architectures

✓ CNN, RNN 또는 기타 네트워크를 사용

3. Tag Decoder Architectures

✓ 입력 시퀀스의 토큰에 대한 태그를 예측

✓ 예) B-(begin), I-(inside), E-(end), S-(singleton), O-(outside)

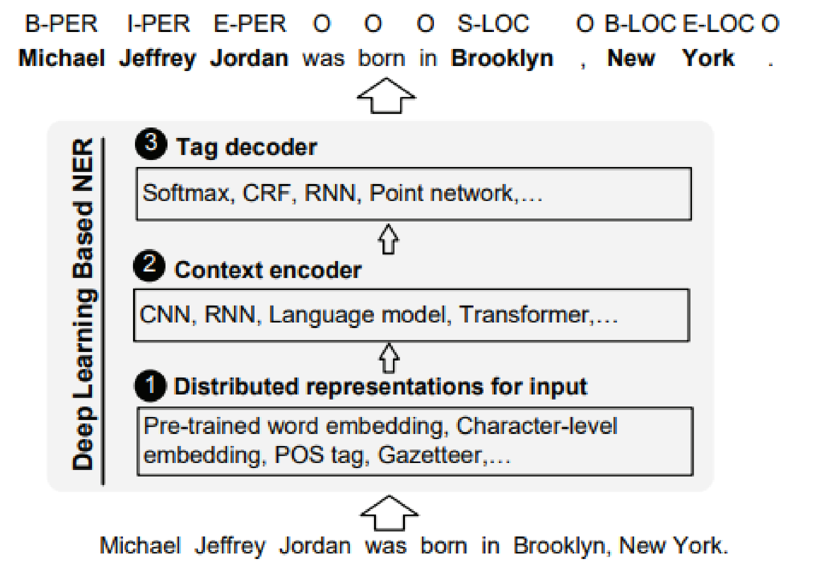


Fig. 2. The taxonomy of DL-based NER. From input sequence to predicted tags, a DL-based NER model consists of distributed representations for input, context encoder, and tag decoder.

Deep Learning Techniques for NER

❖ BIO Tag

- NER 과 같은 Information Extraction 작업에 자주 이용되는 Tag Set
- 하나의 개체명이 여러개의 형태소로 이루어져 있을 경우 유용함

- BIO 의미
  - ✓ B: Begin의 약자로 개체명이 시작되는 부분
  - ✓ I: Inside의 약자로 개체명의 내부 부분
  - ✓ O: Outside의 약자로 개체명이 아닌 부분
  - ✓ (참고)E(L): END(LAST) 의 약자로 개체명의 끝 부분
  - ✓ (참고) S: 단독 개체명

Michal Jeffrey Jordan was born in Brooklyn, New York.

⇩

Named Entity Recognition

⇩

Entity	BIO Tag
Michal	B-Person
Jeffrey	I-Person
Jordan	I-Person
was	O
born	O
in	O
Brooklyn	B-Location
New	B-Location
York	I-Location

Deep Learning Techniques for NER

❖ Distributed Representations for Input

1. Word-level Representation

✓ pre-trained 된 word 임베딩: Google Word2Vec, Stanford GloVe, Facebook fastText 등

✓ 사전 학습 된 word 임베딩을 고정하거나 NER 모델 학습 중에 fine-tuning

2. Character-level Representation

✓ prefix 및 suffix와 같은 명시적인 하위 단어 수준 정보를 이용하는 데 유용

✓ out-of-vocabulary 처리 → 보이지 않는 단어에 대한 표현을 추론하고 형태소 수준의 규칙성 정보를 공유

✓ CNN, RNN 등

3. Hybrid Representation

✓ 추가 정보 (예 : 색인, 어휘 유사성, 시각적 특징 등) 통합

✓ NER 성능 향상

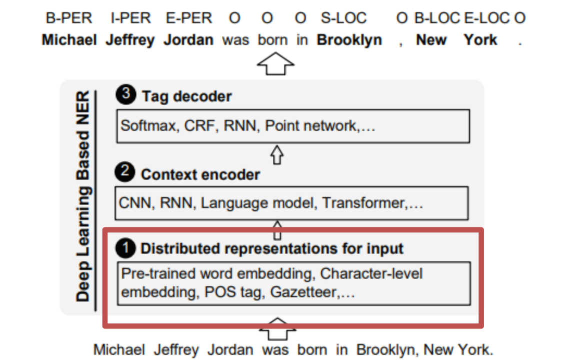
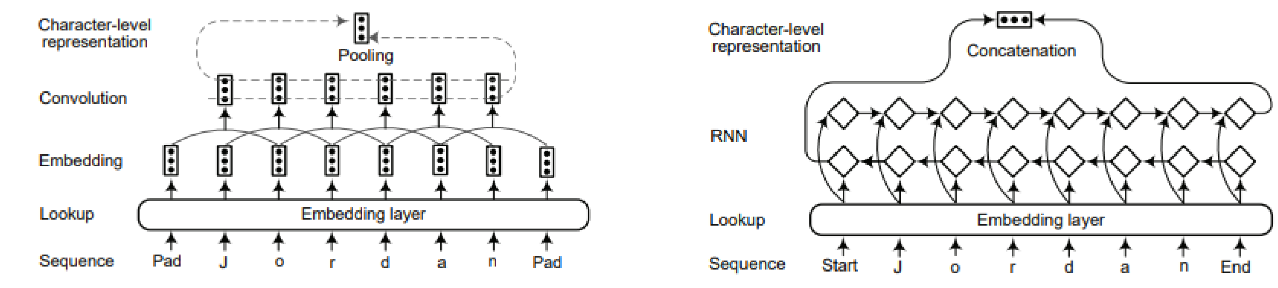


Fig. 2. The taxonomy of DL-based NER. From input sequence to predicted tags, a DL-based NER model consists of distributed representations for input, context encoder, and tag decoder.



(a) CNN-based character-level representation. (b) RNN-based character-level representation.

Fig. 3. CNN-based and RNN-based models for extracting character-level representation for a word.

Deep Learning Techniques for NER

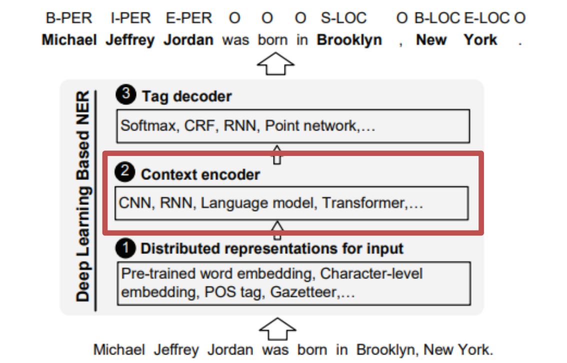


Fig. 2. The taxonomy of DL-based NER. From input sequence to predicted tags, a DL-based NER model consists of distributed representations for input, context encoder, and tag decoder.

❖ Context Encoder Architectures

1. **Convolutional Neural Networks**

2. **Recurrent Neural Networks**

   ✓ GRU(gated recurrent unit), LSTM(long-short term memory), bidirectional RNN과 같은 변형으로 성과 ↑

3. **Recursive Neural Networks**

4. **Deep Transformer**

   ✓ 일반적으로 인코더 및 디코더에 사용되는 convolutional or recurrent networks 를 제

   ✓ self-attention, pointwise, fully connected layers을 활용하여
   인코더 및 디코더를 위한 기본 블록을 구축

   ✓ 품질이 우수하면서도 훈련하는 데 훨씬 적은 시간이 소요됨

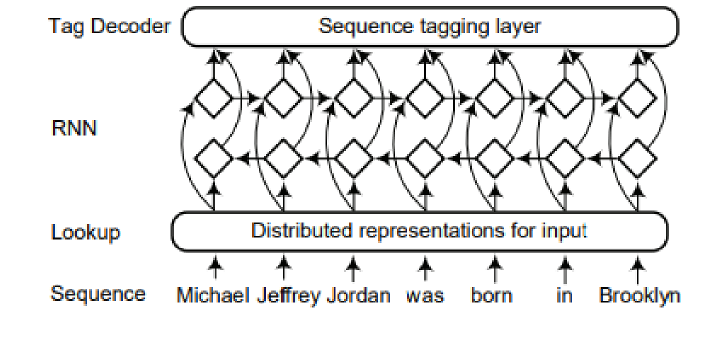


Fig. 7. The architecture of RNN-based context encoder.

Deep Learning Techniques for NER

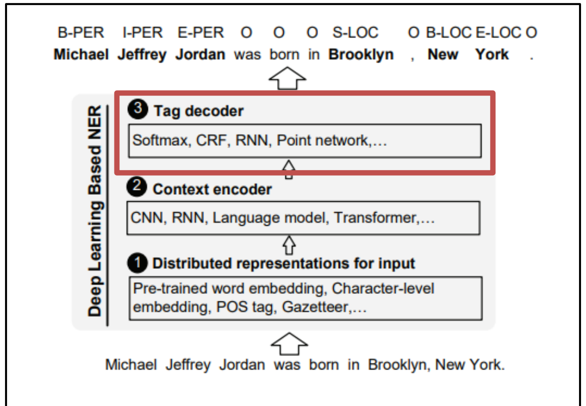


Fig. 2. The taxonomy of DL-based NER. From input sequence to predicted tags, a DL-based NER model consists of distributed representations for input, context encoder, and tag decoder.

❖ Tag Decoder Architectures

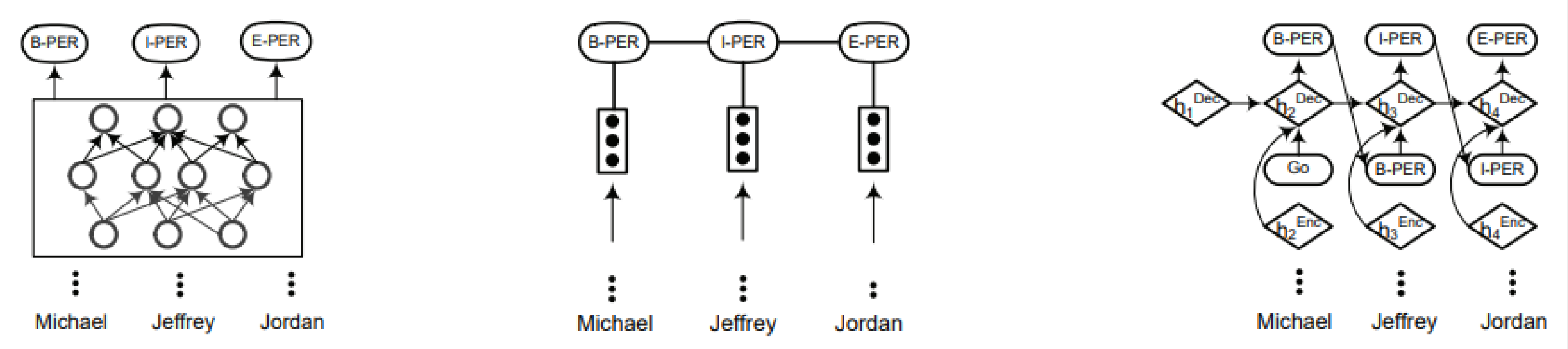
1. **Multi-Layer Perceptron + Softmax**

   ✓multi-class classification problem

2. **Conditional Random Fields**

   ✓ 가장 일반적인 방법

3. **Recurrent Neural Networks**

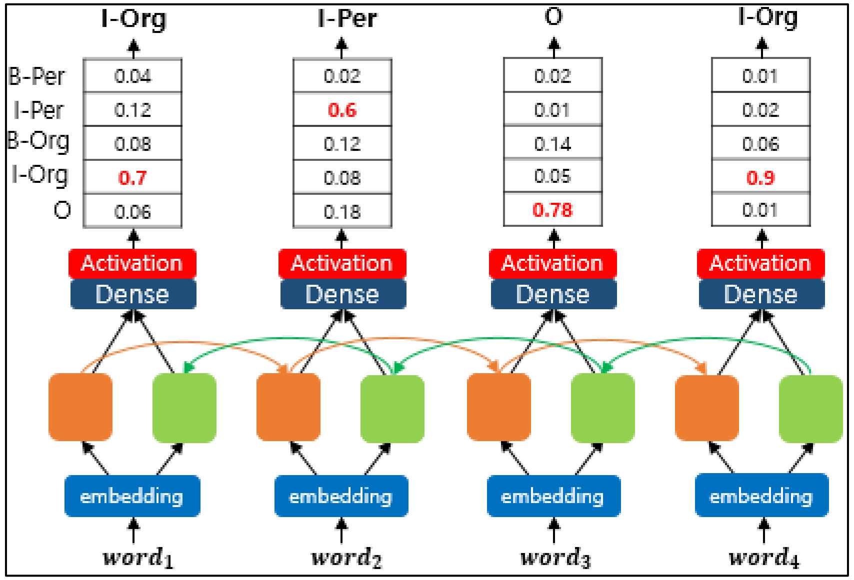


(a) MLP+Softmax (b) CRF (c) RNN

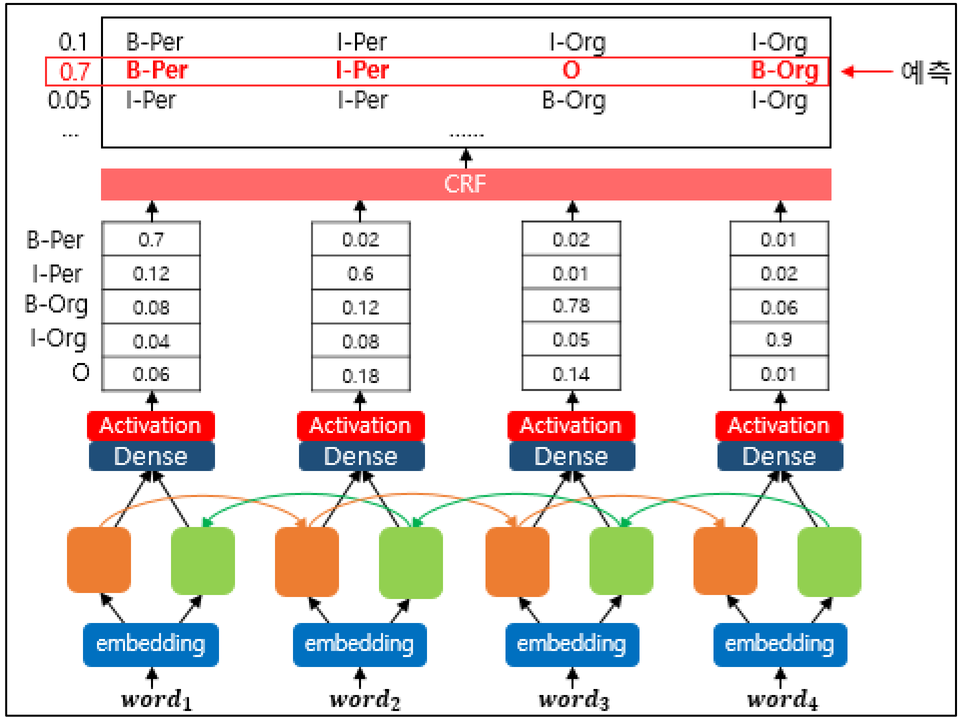
Deep Learning Techniques for NER

❖ Tag Decoder Architectures

- Conditional Random Field (CRF)
  - ✓ BIO Tag에서 일관성 유지
  - ✓ Output Label에 대한 양방향 문맥 반영



[그림1]



[그림2]

출처: https://wikidocs.net/34156

Deep Learning Techniques for NER

TABLE 3
Summary of recent works on neural NER. LSTM: long short-term memory, CNN: convolutional neural network, GRU: gated recurrent unit, LM: language model, ID-CNN: iterated dilated convolutional neural network, BRNN: bidirectional recursive neural network, MLP: multi-layer perceptron, CRF: conditional random field, Semi-CRF: Semi-markov conditional random field, FOFE: fixed-size ordinally forgetting encoding.

Work	Input representation			Context encoder	Tag decoder	Performance (F-score)
	Character	Word	Hybrid			
[93]	-	Trained on PubMed	POS	CNN	CRF	GENIA: 71.01%
[88]	-	Trained on Gigaword	-	GRU	GRU	ACE 2005: 80.00%
[94]	-	Random	-	LSTM	Pointer Network	ATIS: 96.86%
[89]	-	Trained on NYT	-	LSTM	LSTM	NYT: 49.50%
[90]	-	SENNA	Word shape	ID-CNN	CRF	CoNLL03: 90.65%; OntoNotes5.0: 86.84%
[95]	-	Google word2vec	-	LSTM	LSTM	CoNLL04: 75.0%
[99]	LSTM	-	-	LSTM	CRF	CoNLL03: 84.52%
[96]	CNN	GloVe	-	LSTM	CRF	CoNLL03: 91.21%
[104]	LSTM	Google word2vec	-	LSTM	CRF	CoNLL03: 84.09%
[19]	LSTM	SENNA	-	LSTM	CRF	CoNLL03: 90.94%
[105]	GRU	SENNA	-	GRU	CRF	CoNLL03: 90.94%
[97]	CNN	GloVe	POS	BRNN	Softmax	OntoNotes5.0: 87.21%
[106]	LSTM-LM	-	-	LSTM	CRF	CoNLL03: 93.09%; OntoNotes5.0: 89.71%
[102]	CNN-LSTM-LM	-	-	LSTM	CRF	CoNLL03: 92.22%
[17]	-	Random	POS	CNN	CRF	CoNLL03: 89.86%
[18]	-	SENNA	Spelling, n-gram, gazetteer	LSTM	CRF	CoNLL03: 90.10%
[20]	CNN	SENNA	capitalization, lexicons	LSTM	CRF	CoNLL03: 91.62%; OntoNotes5.0: 86.34%
[115]	-	-	FOFE	MLP	CRF	CoNLL03: 91.17%
[100]	LSTM	GloVe	-	LSTM	CRF	CoNLL03: 91.07%
[112]	LSTM	GloVe	Syntactic	LSTM	CRF	W-NUT17: 40.42%
[101]	CNN	SENNA	-	LSTM	Reranker	CoNLL03: 91.62%
[113]	CNN	Twitter Word2vec	POS	LSTM	CRF	W-NUT17: 41.86%
[114]	LSTM	GloVe	POS, topics	LSTM	CRF	W-NUT17: 41.81%
[117]	LSTM	GloVe	Images	LSTM	CRF	SnapCaptions: 52.4%
[108]	LSTM	SSKIP	Lexical	LSTM	CRF	CoNLL03: 91.73%; OntoNotes5.0: 87.95%
[118]	-	WordPiece	Segment, position	Transformer	Softmax	CoNLL03: 92.8%
[120]	LSTM	SENNA	-	LSTM	Softmax	CoNLL03: 91.48%
[123]	LSTM	Google Word2vec	-	LSTM	CRF	CoNLL03: 86.26%
[21]	GRU	SENNA	LM	GRU	CRF	CoNLL03: 91.93%
[125]	LSTM	GloVe	-	LSTM	CRF	CoNLL03: 91.71%
[141]	-	SENNA	POS, gazetteers	CNN	Semi-CRF	CoNLL03: 90.87%
[142]	LSTM	GloVe	-	LSTM	Semi-CRF	CoNLL03: 91.38%
[87]	CNN	Trained on Gigaword	-	LSTM	LSTM	CoNLL03: 90.69%; OntoNotes5.0: 86.15%
[109]	-	GloVe	ELMo, dependency	LSTM	CRF	CoNLL03: 92.4%; OntoNotes5.0: 89.88%
[107]	CNN	GloVe	ELMo, gazetteers	LSTM	Semi-CRF	CoNLL03: 92.75%; OntoNotes5.0: 89.94%
[132]	LSTM	GloVe	ELMo, POS	LSTM	Softmax	CoNLL03: 92.28%
[136]	-	-	BERT	-	Softmax	CoNLL03: 93.04%; OntoNotes5.0: 91.11%
[137]	-	-	BERT	-	Softmax +Dice Loss	CoNLL03: 93.33%; **OntoNotes5.0: 92.07%**
[133]	LSTM	GloVe	BERT, document-level embeddings	LSTM	CRF	CoNLL03: 93.37%; OntoNotes5.0: 90.3%
[134]	CNN	GloVe	BERT, global embeddings	GRU	GRU	CoNLL03: 93.47%
[131]	CNN	-	Cloze-style LM embeddings	LSTM	CRF	**CoNLL03: 93.5%**
[135]	-	GloVe	Plooled contextual embeddings	RNN	CRF	CoNLL03: 93.47%

Challenges for NER

❖ Challenges

- **Data Annotation**
  - ✓ 많은 데이터가 필요
  - ✓ 시간과 비용의 문제
  - ✓ 리소스가 부족한 언어와 특정 도메인의 어려움
  - ✓ 언어 모호성 ex) Bank Account(은행 계좌) vs. River Bank(강둑)

- **Noisy in Informal Text**
  - ✓ 사용자 생성 텍스트 혹은 비형식적 텍스트(댓글, SNS 등) 에 대해서는 낮은 정확도
  - ✓ 도메인별 차이

Fine-grained NER

❖ Fine-grained NER in Domain-specific Area

- Fuzzy-LSTM-CRF
- AutoNER

❖ Informal Text with Auxiliary Resource

- Gazetteer-enhanced sub-tagger

Fine-grained NER in Domain-specific Area

❖ Fuzzy-LSTM-CRF

- Shang, Jingbo, et al. "Learning named entity tagger using domain-specific dictionary." arXiv preprint arXiv:1809.03599 (2018).
- 목적
  - ✓ Multi-labels 또는 unknown-type의 token 처리

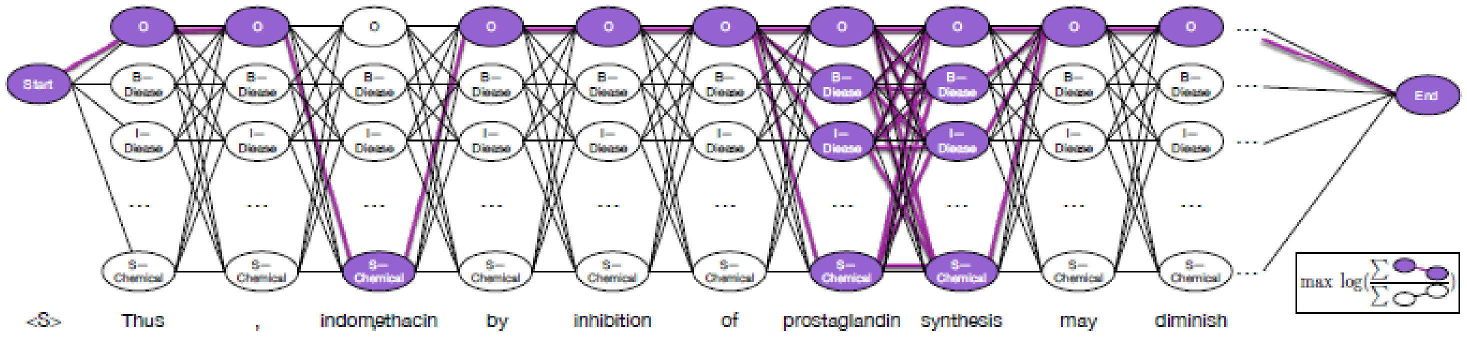


Figure 1: The illustration of the Fuzzy CRF layer with modified `IOBES` tagging scheme. The named entity types are {`Chemical`, `Disease`}. "indomethacin" is a matched `Chemical` entity and "prostaglandin synthesis" is an unknown-typed high-quality phrase. Paths from `Start` to `End` marked as purple form all possible label sequences given the distant supervision.

Fine-grained NER in Domain-specific Area

❖ Fuzzy-LSTM-CRF

- BIOES Tag 수정
  - ✓ Token Type
    1. 일치하는 Entity Type 이 있는 경우
    2. Unknown Type의 Token → (Type의 개수 * 4(BIES) + 1) 개의 label이 가능
    3. Non-entity

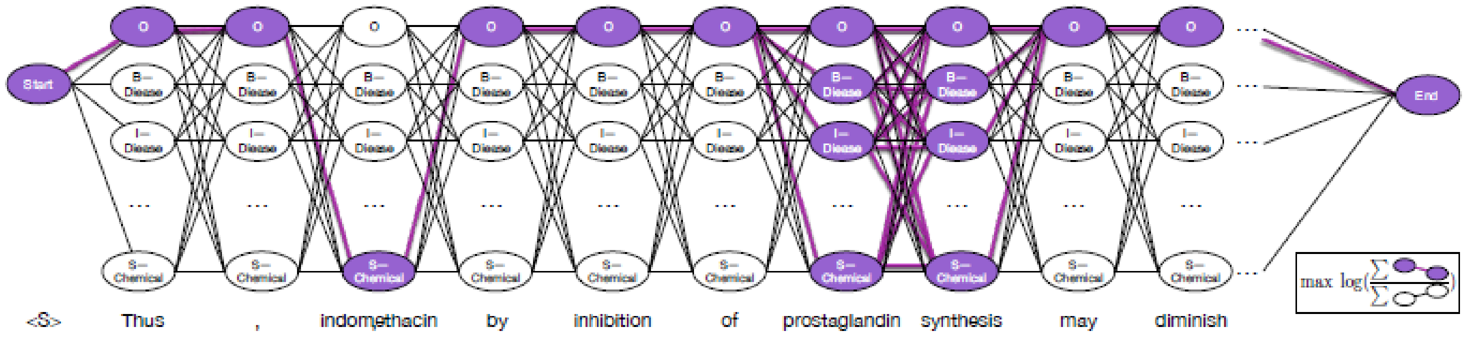


Figure 1: The illustration of the Fuzzy CRF layer with modified IOBES tagging scheme. The named entity types are {Chemical, Disease}. "indomethacin" is a matched Chemical entity and "prostaglandin synthesis" is an unknown-typed high-quality phrase. Paths from Start to End marked as purple form all possible label sequences given the distant supervision.

Fine-grained NER in Domain-specific Area

❖ Fuzzy-LSTM-CRF

- BIOES Tag 수정
  - ✓ Example (Entity Type : {Chemical, Disease})
    - Unknown Type의 가능한 labeled: {O, B-Disease, I-Disease, E-Disease, S-Disease, B-Chemical, I-Chemical, E-Chemical, S-Chemical}
    - Non-entity : {O}

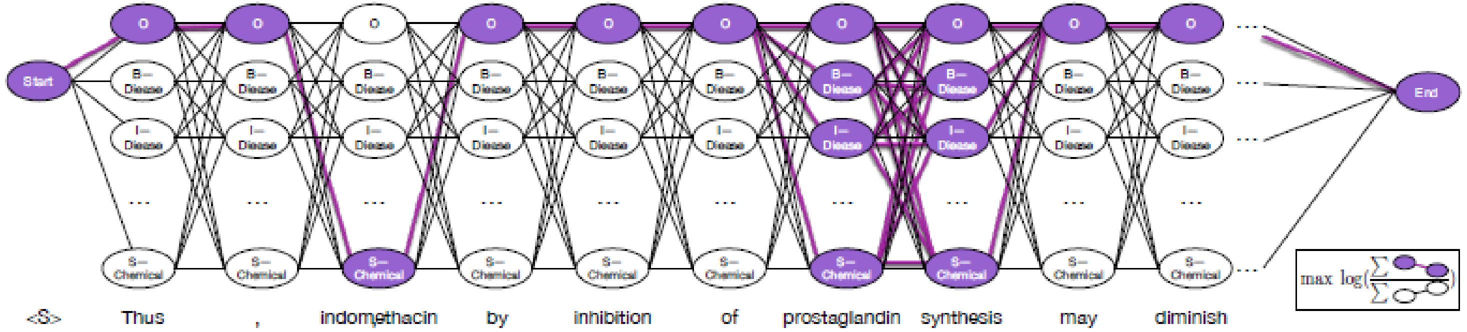


Figure 1: The illustration of the Fuzzy CRF layer with modified `IOBES` tagging scheme. The named entity types are {`Chemical`, `Disease`}. "indomethacin" is a matched `Chemical` entity and "prostaglandin synthesis" is an unknown-typed high-quality phrase. Paths from `Start` to `End` marked as purple form all possible label sequences given the distant supervision.

Fine-grained NER in Domain-specific Area

❖ Fuzzy-LSTM-CRF

- Calculate
  - ✓ Word sequence: $(X_1, X_2, \cdots, X_n)$
  - ✓ The score of the predicted sequence $(y_1, y_2, \cdots, y_n)$

$$s(X, y) = \sum_{i=0}^{n} \Phi_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}$$

  - ✓ Maximizes the total probability

$$p(y|X) = \frac{\sum_{\tilde{y} \in Y_{possible}} e^{s(X, \tilde{y})}}{\sum_{\tilde{y} \in Y_X} e^{s(X, \tilde{y})}}$$

Fine-grained NER in Domain-specific Area

❖ AutoNER

- Shang, Jingbo, et al. "Learning named entity tagger using domain-specific dictionary." arXiv preprint arXiv:1809.03599 (2018).
- 목적
  - ✓ 인접한 token이 간의 관계 정의

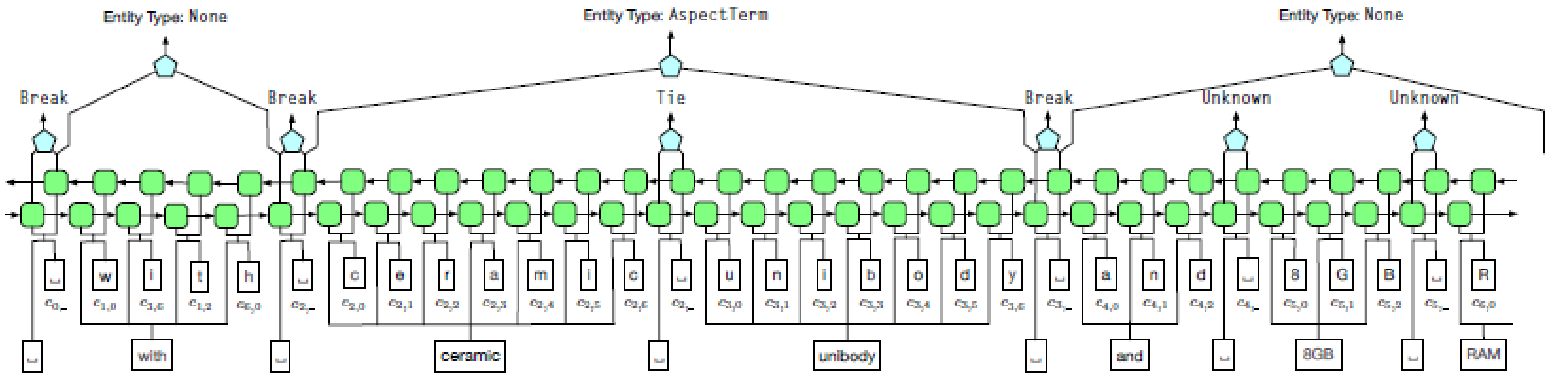


Figure 2: The illustration of AutoNER with `Tie or Break` tagging scheme. The named entity type is {`AspectTerm`}. "ceramic unibody" is a matched `AspectTerm` entity and "8GB RAM" is an unknown-typed high-quality phrase. `Unknown` labels will be skipped during the model training.

Fine-grained NER in Domain-specific Area

❖ AutoNER

- Tie or Break
  - ✓ Adjacent Token Type
    1. Tie: 두 Token이 동일한 Entity
    2. Unknown: Token 중 적어도 하나가 unknown-typed 속하는 경우
    3. Break

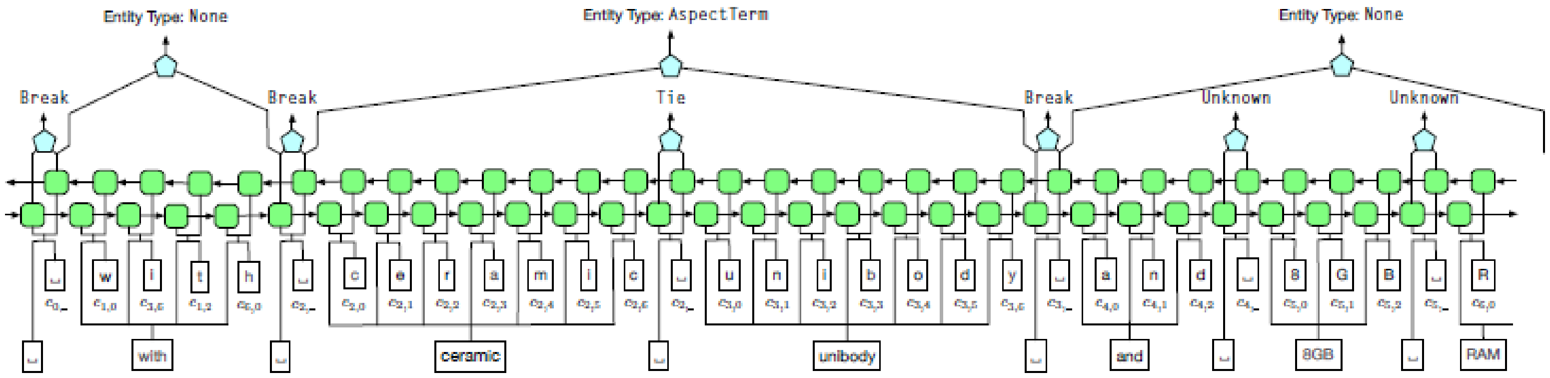


Figure 2: The illustration of AutoNER with `Tie or Break` tagging scheme. The named entity type is {`AspectTerm`}. "ceramic unibody" is a matched `AspectTerm` entity and "8GB RAM" is an unknown-typed high-quality phrase. `Unknown` labels will be skipped during the model training.

Fine-grained NER in Domain-specific Area

❖ AutoNER

- Tie or Break
  - ✓ Example
    1. Tie: ceramic unibody
    2. Unknown: 8GB RAM
    3. None

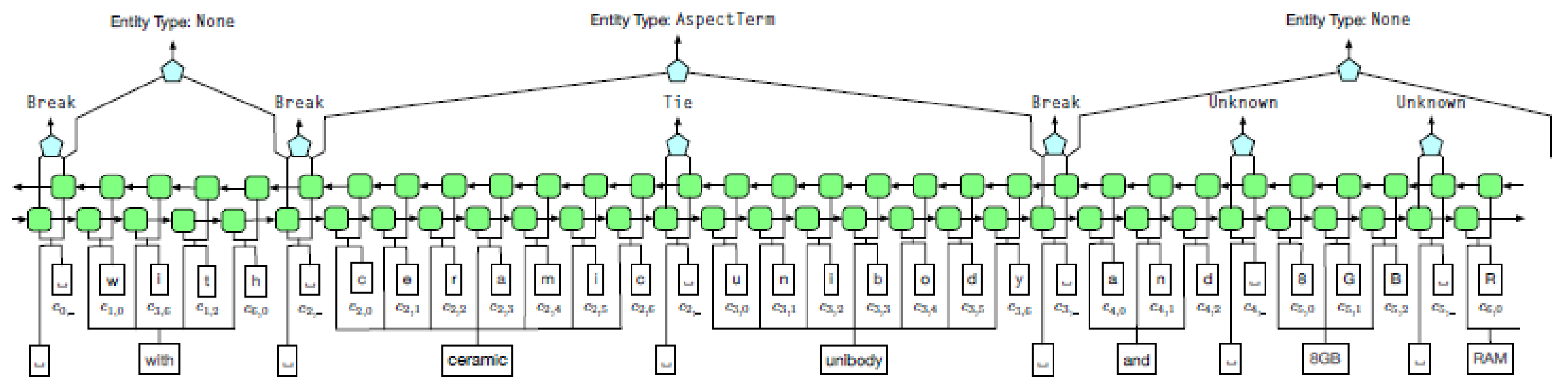


Figure 2: The illustration of AutoNER with `Tie or Break` tagging scheme. The named entity type is {`AspectTerm`}. "ceramic unibody" is a matched `AspectTerm` entity and "8GB RAM" is an unknown-typed high-quality phrase. `Unknown` labels will be skipped during the model training.

Fine-grained NER in Domain-specific Area

Table 2: [Biomedical Domain] NER Performance Comparison. The supervised benchmarks on the BC5CDR and NCBI-Disease datasets are LM-LSTM-CRF and LSTM-CRF respectively (Wang et al., 2018). SwellShark has no annotated data, but for entity span extraction, it requires pre-trained POS taggers and extra human efforts of designing POS tag-based regular expressions and/or hand-tuning for special cases.

Method	Human Effort other than Dictionary	BC5CDR			NCBI-Disease		
		Pre	Rec	F1	Pre	Rec	F1
Supervised Benchmark	Gold Annotations	88.84	85.16	**86.96**	86.11	85.49	**85.80**
SwellShark	Regex Design + Special Case Tuning	86.11	82.39	84.21	81.6	80.1	**80.8**
	Regex Design	84.98	83.49	**84.23**	64.7	69.7	67.1
Dictionary Match	None	93.93	58.35	71.98	90.59	56.15	69.32
Fuzzy-LSTM-CRF		88.27	76.75	82.11	79.85	67.71	73.28
AutoNER		88.96	81.00	**84.8**	79.42	71.98	**75.52**

Informal Text with Auxiliary Resource

❖ Gazetteer-enhanced sub-tagger

- Gazetter: 장소 이름과 정보를 제공하는 책. (지명사전)

- Liu, Tianyu, Jin-Ge Yao, and Chin-Yew Lin. "Towards improving neural named entity recognition with gazetteers." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019.

- 특징
  - ✓ Gazetteer를 별도의 모듈로 추가
  - ✓ Hybrid semi-Markov CRF
    - Token level label을 사용하여 span level score를 구할 수 있음

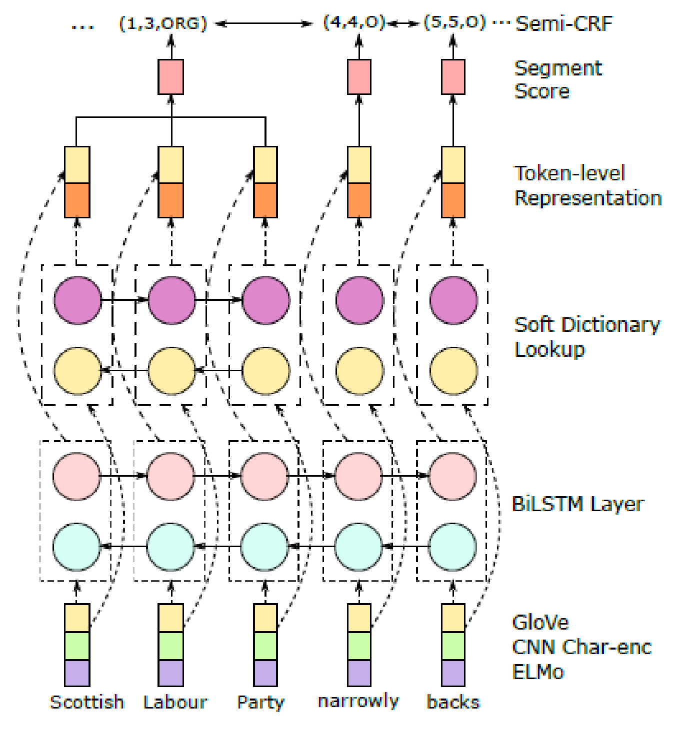


Figure 1: Overall architecture

Informal Text with Auxiliary Resource

❖ **Hybrid semi-Markov CRFs**

Model	Test Set F1-score(±std)	
	CoNLL	OntoNotes
Ma and Hovy (2016)	91.21	-
Lample et al. (2016)	90.94	-
Liu et al. (2018)	91.24±0.12	-
Devlin et al. (2018)	92.8	-
Chiu and Nichols (2016) [5]	91.62±0.33	86.28±0.26
Ghaddar and Langlais '18	91.73±0.10	87.95±0.13
Peters et al. (2018)	92.22±0.10	89.04±0.27
Clark et al. (2018)	92.6 ±0.1	88.8±0.1
Akbik et al. (2018)	93.09±0.12	89.71
HSCRF	92.54±0.11	89.38±0.11
HSCRF + concat	92.52±0.09	89.73±0.19
HSCRF + gazemb	92.63±0.08	89.77±0.20
HSCRF + softdict	92.75±0.18	89.94±0.16

Table 1: Results on CoNLL 2003 and OntoNotes 5.0

Conclusion

Introduction of NER

Deep Learning Techniques for NER

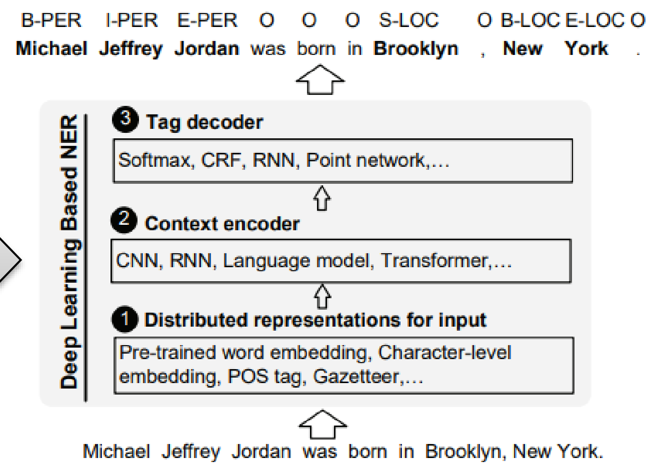


Fig. 2. The taxonomy of DL-based NER. From input sequence to predicted tags, a DL-based NER model consists of distributed representations for input, context encoder, and tag decoder.

Fine-grained NER

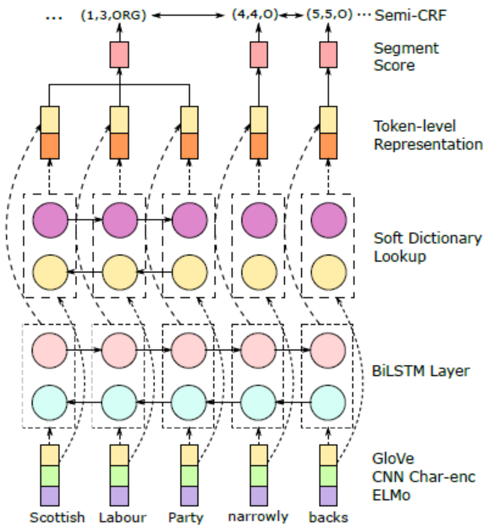


Figure 1: Overall architecture

- 기존의 많은 연구은 coarse-grained NER에 초점
- 다양한 실제 단어 적용을 지원하기 위해 영역 별 fine-grained NER 필요

감사합니다