# Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation

Keio University, Fujisawa, Kanagawa, Japan

Conference on Computational Natural Language Learning (CoNLL), 2016

# 1. Purpose


- NED(Name entity Disambiguation) 의 문제점은 entity mention 이 가지는 의미의 모호성

Ex) 문서의 "Washington” 은 배우 덴젤워싱턴, 미국의 수도 워싱턴, 대통령 조지 워싱턴 등 많은 entity 로서 언급될 수 있다.

# 2. Proposed method

- Words 와 entity 를 같은 vector space 안에 같이 embedding 을 시키는 method 를 사용
  - skim-gram 을 기반으로한 skim-gram extend Model 을 제안
  - 제시하는 Model 은 세가지 모델로 구성된다.

1) skip-gram model

- target word 와의 neighboring words 예측

1) KB graph model

- Knowledge Base 의 link graph 를 통한 target entity 의 neighboring entities 예측

1) anchor context model

- target entity 를 통해 neighboring words 예측

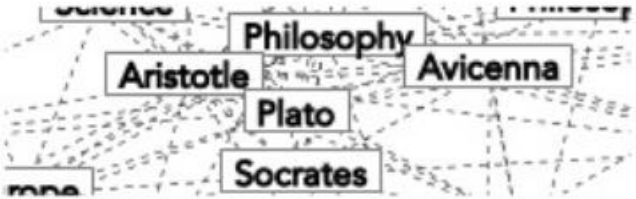
# 2. Proposed method

- skip-gram model
  - skim-gram 은 target word 로 context words 를 예측
  - objective function :

$$\mathcal{L}_w = \sum_{t=1}^{T} \sum_{-c \le j \le c, j \neq 0} \log P(w_{t+j} | w_t)$$

  - c 는 window size

# 2. Proposed method




- KB graph model
  - skim-gram 을 기반으로한 skim-gram extend Model 을 제안
  - Knowledge Base 의 link structure 를 이용하여 entity 간의 관련성 학습 모델
  - objective function :

$$\mathcal{L}_e = \sum_{e_i \in E} \sum_{e_o \in C_{e_i}, e_i \neq e_o} \log P(e_o|e_i)$$

  - E 는 KB 의 모든 entity set, $C_e$ 는 entity e 와 link 된 모든 entity

# 2. Proposed method


- Anchor Context Model
  - KB 로서 wikipedia 를 이용
  - anchor 를 통해 KB 에서 많은 entity 와 그에 해당하는 neighboring words 예측
  - anchor 를 통해 획득된 entity 를 통해 entity 의 context words 를 예측한다.
  - objective function :

$$\mathcal{L}_a = \sum_{(e_i,Q)\in A} \sum_{w_o \in Q} \log P(w_o|e_i)$$

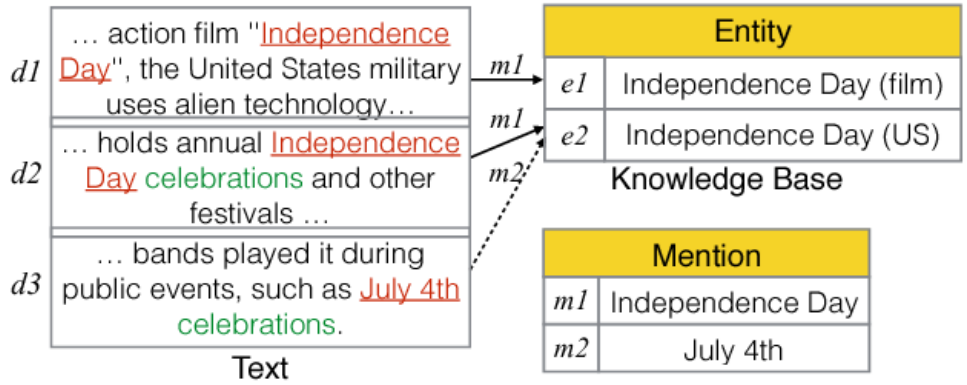
# 2. Proposed method

- Final Training
  - objective function :

$$\mathcal{L} = \mathcal{L}_w + \mathcal{L}_e + \mathcal{L}_a$$

  - 하지만 계산복잡도의 문제로 인해 Negative sampling 사용
  - 모델 training 하기위해 wikipedia 사용
  - Optimization 으로 SGD 사용

# 3. Named Entity Disambiguation Using Embedding




- 2개의 sub-tasks 로 NED 분리
  - candidate generation 과 mention disambiguation
  - candidate generation : 각각의 entity mention 에 대해 해당 entity 의 candidates 생성
  - Mention Disambiguation : candidate generation 에서 생성된 candidate entity 에서 가장 관련 있는 entity 를 선택하는 task

# 3. Named Entity Disambiguation Using Embedding

- Modeling Textual Context and Coherence
  - Modeling Textual Context : candidate entity 와 context 의 similarity 로 entity 선택

    $\vec{v_{c_w}}$ 와 $\vec{v_e}$ 의 similarity
  - context : 해당 문서의 noun vector 들의 average

$$\vec{v_{c_w}} = \frac{1}{|W_{c_m}|} \sum_{w \in W_{c_m}} \vec{v_w}$$

  - 문서 d 에서 모든 명사어를 context words 로 사용

# 3. Named Entity Disambiguation Using Embedding

- Modeling Textual Context and Coherence
  - Modeling Coherence : candidate entity 와 context entity 의 similarity 로 context entities 계산
  - context : 해당 문서의 target entity 를 제외한 entity vector 들의 average

$$\vec{v_{c_e}} = \frac{1}{|E_{c_m}|} \sum_{e^* \in E_{cm}} \vec{v_{e^*}}$$

- 앞선 2개의 similarity scores 를 통해 mention 에 대한 candidate entity 의 rank 를 매김

# 4. Experiment

- Accuracy scores of the proposed method and the sota methods.
  - Using CoNLL Dataset and TAC(Text Analysis Conference) 2010 Dataset 를 통해 모호성 제거 성능 평가

| | CoNLL (Micro) | CoNLL (Macro) | TAC10 (Micro) |
|---|---|---|---|
| Our Method | **93.1** | **92.6** | **85.2** |
| Hoffart et al., 2011 | 82.5 | 81.7 | - |
| He et al., 2013 | 85.6 | 84.0 | 81.0 |
| Chisholm & Hachey, 2015 | 88.7 | - | 80.7 |
| Pershina et al., 2015 | 91.8 | 89.9 | - |