컨볼루션 신경망을 이용한 구인 광고 데이터 정형화 시스템

김현지*, 서인*, 이승민**, 황준승*, 홍기재**, 한욱신***
*포항공과대학교 창의IT융합공학과
**포항공과대학교 컴퓨터공학과
***포항공과대학교 창의IT융합공학과/컴퓨터공학과
e-mail:hjkim@dblab.postech.ac.kr

A System for Relation Extraction from Job Postings using Convolutional Neural Network

Hyeon-Ji Kim*, In Seo*, SeungMin Lee**, JunSeung Hwang*, KiJae Hong**, Wook-Shin Han***
*Dept of Creative IT Engineering, POSTECH
**Dept of Computer Science and Engineering, POSTECH
***Dept of Creative IT Engineering/Dept of Computer Science and Engineering, POSTECH

요 약

데이터 정형화 기술은 자연어 처리 및 인공 지능 분야, 데이터베이스 등 다양한 분야에서 중요한 핵심적인 기술 중 하나이다. 최근 정형화 문제를 푸는 많은 신경망 기반 알고리즘들이 제안되었으나, 기존의 모든 알고리즘이 키워드의 후보가 입력으로 주어진다고 가정하고 있으며, 알고리즘 대부분은 두 개의 속성(attribute)을 가지는 이진 관계(binary relation)만 처리할 수 있다는 한계가 있다. 본 논문에서는 컨볼루션 신경망을 이용한 N항 관계 정형화 방법을 제안하고, 이를 이용한 구인 광고 정형화 시스템을 개발하고 성능을 평가한다.

1. 서론

데이터 정형화는 비정형 데이터와 정형 데이터 스키마가 주어졌을 때 비정형 데이터를 주어진 데이터 스키마를 가지는 정형 데이터로 변환하는 문제로, 자연어 처리 및 인공 지능 분야, 데이터베이스 등 다양한 분야에서 중요한 핵심적인 기술 중 하나로서 많은 응용 프로그램에서 사용되고 있다 [1]. 다양한 형식의 비정형 데이터를 정형화할 경우 인덱스를 이용한 효율적인 색인이 가능하고, 범위 질의 등의 복잡한 질의를 처리할 수 있다. 본 논문에서 개발한 구인 광고 정형화 시스템은 다양한 형식의 구인 광고 데이터를 정형화하여, 구인 광고 데이터에 대한 효과적인 검색 기능을 지원할 수 있다. 예를 들어, 구인 광고의 정형화 데이터를 이용하면 사용자가 원하는 지역에 대해 구인 광고를 검색할 수 있다(그림 1).

최근 딥 러닝 기술의 발전으로 인해 정형화 문제를 푸는 많은 신경망 기반 알고리즘들이 제안되었다[3, 4, 5, 7, 8]. 그러나 이 알고리즘들은 정형 데이터로 저장될 키워드의 후보가 입력으로 주어진다고 가정하고 있으며, 대부분 알고리즘은 두 개의 속성(attribute)을 가지는 이진 관계(binary relation)만 처리할 수 있다는 한계가 있다. 또한, 기존 방법들은 입력 데이터가 완전한 문장으로 주어진다고 가정하고 있다. 그러나 구인 광고 데이터는 웹 데이터로 완전한 문장이 아니다.

본 논문에서는 컨볼루션 신경망 모형을 활용하여 후보 키워드가 입력으로 주어지지 않고, 세 개 이상의 속성을 가지는 데이터 스키마의 정형 데이터로의 정형화를 처리할 수 있는 정형화 방법을 개발하고, 이를 이용하여 구인 광고 정형화 시스템을 개발하였다. 개발한 구인 광고 시스템은 먼저 웹에서 구인 광고를 크롤링하여 비정형 데이터를 수집한다. 수집한 비정형 데이터를 이용하여 신경망 분류기를 통해 최적의 키워드를 스키마의 각 속성에 대응시켜 정형 데이터를 추출한다.

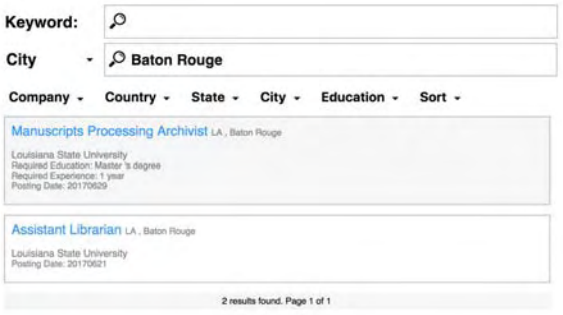

그림 1 구인 광고 정형 데이터를 이용한 검색의 예시.

"본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 ICT명품인재양성사업의 연구결과로 수행되었음" (IITP-2017-R0346-16-1007)

2. 관련 연구

[1]은 영문 텍스트와 데이터 스키마가 주어졌을 때, 영문 텍스트를 관계형 데이터베이스의 튜플(tuple)로 변환하는 규칙 학습 기반 정형화 기술 RAPIER를 제안하였다. RAPIER는 규칙을 학습하기 위해 세 가지 패턴을 정의한다. 첫 번째는 필러(filler) 바로 앞의 텍스트와 일치하는 사전-필러(pre-filler) 패턴이다. 두 번째는 실제 텍스트와 일치하는 필러 패턴이다. 마지막 세 번째는 필러 바로 뒤에 나오는 텍스트와 일치하는 사후-필러(post-filler) 패턴이다. RAPIER는 이 세 가지 패턴을 이용해 데이터 정형화를 위한 규칙을 학습한다. RAPIER는 패턴 규칙 학습을 기반으로 하여 텍스트 데이터에서 빈번하게 등장하는 패턴을 자동으로 인식하고 정형 데이터를 추출한다. 따라서 다른 알고리즘에 비교해 높은 정확도를 나타낸다. 그러나 데이터에 등장한 적이 없는 패턴을 처리할 수 없기 때문에 재현율이 매우 낮다.

최근 지식 베이스 시스템의 발전에 따라 지도 학습에 이용 가능한 데이터의 양이 증가하였고, 이에 따라 신경망을 이용한 정형화 기술이 활발히 연구되고 있다. 신경망 기반 정형화 연구들은 크게 이진 관계 분류(binary relation classification), 슬롯 채우기(slot filling), N항 관계 추출(N-ary relation extraction) 연구로 나눌 수 있다.

이진 관계 분류에 대한 연구는 컨볼루션 신경망 기반의 방법[3, 4, 5]과 순환 신경망/장단기 기억 신경망(recurrent neural networks/Long Short Term Memory networks) 기반의 방법[6, 7]이 있다. [3]은 컨볼루션 신경망 기반 이진 관계 분류 방법으로, 입력으로 주어진 두 개의 명사에 대해 특징점을 추출하고 컨볼루션 신경망 접근법을 사용해 문장 수준의 특징점을 학습하여 두 개의 명사 사이 관계를 예측한다. [4]는 컨볼루션 신경망 접근법에 개별적 최댓값 선정(piecewise max pooling) 기법을 적용하여 기존 특징점 추출 방식의 오류를 개선한 알고리즘을 제안하였다. [5]는 선택적 어텐션(selective attention) 기법을 적용한 컨볼루션 신경망 접근법을 사용하여 학습 모형의 정확도를 향상함과 동시에, 잘못 분류 표시된 학습 데이터의 문제를 해결하였다. [6]은 문장의 종속 관계 기반 파스 트리(dependency-based parse tree)에서 두 키워드 사이의 최단 거리를 구한 뒤, 종속 관계 기반 순환 신경망을 이용하여, 최단 거리상의 단어가 가지는 의미를 분석함으로써 관계 분류 문제 성능을 향상했다. [7]은 장단기 기억 신경망 접근법을 사용하였는데, 데이터셋 내에 두 키워드 중 어느 하나의 키워드만을 포함하는 데이터를 컨텍스트로 정의하고, 이 컨텍스트를 신경망의 입력값으로 하여 기존보다 정밀한 관계 추출이 가능한 학습 모형을 제안하였다. 이러한 기술들은 모두 2개 키워드 사이의 관계를 추출하는 문제에 한정되어 있다. 따라서 스키마의 속성이 2개라고 가정하기 때문에, 본 발명에서 해결하고자 하는 3개 이상의 속성을 가지는 정형 데이터를 다룰 수 없다는 큰 차이점이 있다. 또한, 이 연구들에서는 2개의 키워드가 입력으로 주어지며, 2개의 키워드는 한 문장 내에 동시에 등장한다고 가정하고 있다.

다음으로 슬롯 채우기는 텍스트 뭉치(corpus)와 키워드 그리고 하나의 속성을 줬을 때, 키워드와 해당 속성 관계를 갖는 또 다른 키워드를 텍스트 뭉치 내에서 찾아내는 문제이다. [9]는 장단기 기억 신경망을 이용한 방법으로, 키워드 위치 정보를 활용하여 키워드와 속성 관계를 갖는 다른 키워드 추출 성능을 향상했다. 슬롯 채우기 문제는 N항 관계를 추출할 수 있으나, 본 발명에서 해결하고자 하는 문제와는 달리 키워드가 입력으로 주어진다.

한편, 최근 그래프 기반 장단기 기억 신경망(graph LSTM)을 활용한 N항 관계 추출 연구가 제안되었다[8]. N항 관계 추출 연구는 텍스트, N항 관계의 목록을 줬을 때, 텍스트에서 N개 키워드의 순서쌍이 관계 표현 목록 중 어떤 관계에 대응되는지 찾아내는 문제이다. 그러나, N항 관계 추출 연구 또한 N개의 키워드 순서쌍이 입력 자료로 주어진다고 가정하고 있다.

3. 문제 정의

본 논문에서는 정형화 문제를 다음과 같이 정의한다. 텍스트 T와 N개의 속성을 가지는 데이터 스키마 (col_1, col_2, …, col_N)가 주어졌을 때 T 내의 모든 키워드 c를 (col_1, col_2, …, col_N, NA) 중 하나의 클래스로 분류한다. 이 때, 키워드 c는 T의 단어의 시퀀스로, 시퀀스는 1 이상 n 이하 길이이다. 최종적으로 col_1, col_2, …, col_N 중 하나를 클래스로 가지는 키워드가 정형 데이터에 포함된다.

4. 컨볼루션 신경망 기반 정형화 방법

본 논문에서 제시하는 신경망 기반 데이터 정형화 기술을 개발하면서 중요하게 고려해야 할 세 가지 이슈가 있다. 첫째, 이진 관계 추출을 목표로 하는 기존 연구와는 달리, N항 관계를 추출하고자 할 경우 하나의 문장에 키워드들이 함께 존재하지 않는 경우가 많다. 따라서, 기존의 문장을 이용한 방법을 그대로 적용할 수 없으며, 서로 다른 문장에 속하는 후보 키워드 간의 관계를 고려해야 한다. 둘째, 텍스트의 길이가 문장에 비교해 길기 때문에, 하나의 문장 대신 텍스트를 신경망의 입력으로 그대로 이용할 경우 입력의 길이가 하나의 문장에 비교해 매우 길기 때문에 신경망의 정확도가 대폭 하락할 수 있다. 셋째, 기존 신경망 기반 데이터 정형화 기술과는 달리 키워드가 입력으로 주어지지 않기 때문에 입력으로 주어진 영문 텍스트로부터 데이터 스키마에 대응할 주요 키워드들을 자동으로 찾아내야 한다.

제안하는 방법은 다음의 과정을 통해 위 세 가지 이슈를 해결한다. 먼저, 영문 텍스트 내의 각 후보 키워드와 해당 키워드를 포함한 문장을 신경망의 입력 데이터로 설정하고, 하나의 영문 텍스트 내에 후보 키워드가 여러 문장에 등장할 경우 문장별로 키워드의 점수를 구해 서로 비교하여 가장 큰 점수를 가지는 문장을 해당 후보 키워드의 중심 문장으로 결정한다. 이처럼 신경망의 입력 단

위는 문장으로 하되, 후보 키워드를 포함하는 모든 문장을 고려하여 영문 텍스트와 후보 키워드의 관계를 충분히 고려하도록 함으로써 첫 번째 문제와 두 번째 문제를 해결한다. 컨볼루션 신경망을 이용해 입력된 영문 텍스트에 나타난 모든 후보 키워드를 다 고려한다. 이때 후보 키워드의 점수가 특정 한계점보다 낮으면 어떤 특징점에도 대응하지 않으며, 문법적으로 옳지 않은 후보 키워드는 속성에 대응할 필요가 없는 키워드라는 가정하에, 해당 키워드의 앞과 뒤 단어를 신경망의 특징점으로 사용하여 세 번째 문제를 해결한다.

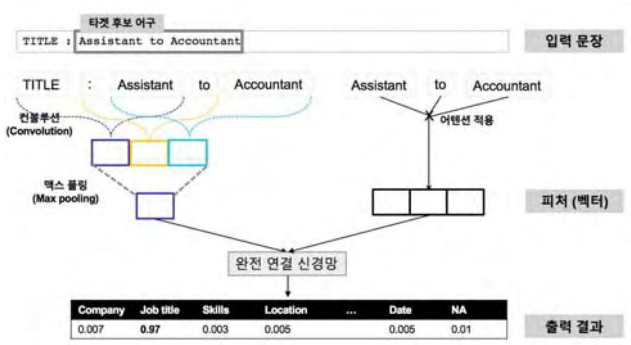

그림 2 컨볼루션 신경망과 어텐션 메카니즘을 이용한 분류기 모델.

정형화의 전체 과정을 요약하면 다음과 같다. 먼저, 입력 텍스트를 자연어 처리를 통해 신경망이 처리 가능한 실수 벡터 형식으로 변형한다. 다음으로, 입력 텍스트로부터 후보 키워드를 추출한 뒤, 각 후보 키워드에 대한 특징점을 추출한다. 후보 키워드의 특징점은 후보 키워드를 포함한 문장을 입력으로 하는 컨볼루션 신경망의 연산 결과와 후보 키워드 및 후보 키워드 직전/직후 키워드의 정보로 구성된다. 다음으로, 추출한 특징점을 입력 데이터로 하고 키워드가 속하는 속성을 출력하는 신경망을 이용하여 각 키워드의 속성을 계산한다. 이때, 키워드가 여러 문장에 등장할 경우, 문장별로 키워드의 점수를 계산해서 가장 큰 점수를 가지는 속성을 그 키워드의 속성으로 결정하도록 한다. 그림 2는 타깃 후보 어구 'Assistant to Accountant'의 예시에 대한 분류기 모델의 처리 과정이다.

알고리즘 1은 본 논문에서 제시하는 구인 광고 정형화 시스템의 신경망 기반 분류기 알고리즘을 기술하고 있다. 알고리즘에서는 먼저 비정형 구인 광고 jp와 데이터 스키마 sc를 입력받는다. InitializeSchema(·)는 결과 정형 데이터를 저장할 d를 입력받은 sc를 데이터 스키마로 가지는 빈 튜플로 초기화한다(1번째 줄). 다음으로, GetCandidateKeywords(·)는 구인 광고 jp로부터 후보 키워드 집합 kc를 생성한다(2번째 줄). 두 함수를 호출하여 d를 초기화하고 kc를 생성한 다음, kc에 속한 각 키워드 k에 대해 k를 포함하는 jp 내의 모든 문장을 찾고(4번째 줄), 각 문장 s에 대해 특징점을 추출한다(7-8번째 줄). 특징점은 문장의 특징점과 키워드 k의 특징점으로 나뉘는데, 문장의 특징점은 컨볼루션 신경망을 이용하여 추출하고(7번째 줄), k의 특징점은 GetKeywordFeatures(·) 함수를 통해 추출한다(8번째 줄). 추출한 특징점을 소프트맥스 회귀(Softmax regression)의 입력값으로 하여, 키워드 k가 속하는 속성과 점수를 계산한다(9번째 줄). 키워드 k를 포함하는 모든 문장 s에 대해서 특징점 추출 및 소프트맥스 회귀 과정을 반복하는데(6-11번째 줄), 이때 문장에 따라 키워드 k의 속성과 점수가 달라질 수 있다. 따라서, $|s_k|$ 개의 결괏값 중에 가장 큰 점수를 가지는 속성에 키워드 k를 대응한다(10-13번째 줄). 이 때, 데이터 스키마의 각 속성에 대한 점수의 한계점이 [0, 1] 사이의 실숫값인 하이퍼파라미터(hyperparameter)로 주어진다. GetThreshold(·)는 각 속성의 한계점을 반환하는 함수로써, 계산한 점수 $score_{max}$가 한계점을 넘지 않는 키워드 k는 속성에 대응하지 않는다(12번째 줄). 위 과정을 kc에 포함된 모든 키워드 k에 대해 반복한다(3-13번째 줄). 최종적으로 각 속성에 대응하는 모든 키워드가 저장된 정형 데이터 d를 출력한다.

알고리즘 1 RelationExtraction(jp, sc)
입력: 구인 광고 jp, 데이터 스키마 sc
출력: 정형 데이터 d

```
1:  d := InitializeSchema(sc);
2:  kc := GetCandidateKeywords(jp);
3:  for each k in kc do
4:      sc := GetSentences(jp, k);
5:      scoremax := 0; r := NA;
6:      for each s in sc do
7:          cvs := GetSentenceFeatures(s);
8:          f(s,k) := GetKeywordFeatures(s, k);
9:          (rs, scores) := Regression(sc, (f(s,k), cvs));
10:         if rs != NA and scoremax <scores then
11:             (r, scoremax) := (rs, scores);
12:     if r != NA and scoremax >GetThreshold(r) then
13:         d := AddKeyword(r, k);
```

5. 구현

개발한 분류기의 학습 속도와 정확도를 개선하기 위해 길이가 비슷한 문장을 묶어서 배치(batch)를 나누는 방식으로 배치 처리를 하였다. 구체적으로, 전체 문장 집합을 문장 길이별로 9개의 집합 (10, 20, 30, 40, 50, 70, 100, 150, 300 tokens)으로 나누고 패딩(padding)을 하였다.

정확도를 향상하기 위해 하이퍼파라미터 튜닝으로 오답 학습 데이터(negative example)의 비율, 배치 크기, 학습률(learning rate) 등을 조절하였다. 그 결과, 본 실험의 기본 설정으로 오답 학습 데이터의 비율은 95%, 배치 크기는 16K, 학습률은 0.001, 후보 키워드 시퀀스의 최대 길이(n)는 7로 설정되었다.

6. 실험

6.1 실험 환경

본 실험에서는 7개의 웹 페이지[1]에서 무작위 추출한 구인 광고 900개, 300개를 각각 학습 데이터, 테스트 데이터

[1] http://polaris.gseis.ucla.edu/labuse/Jobs/
https://groups.yahoo.com/neo/groups/careercenter-am/conversations/messages
http://www.careerbuilder.com/jobs
http://www.careerage.com/search?field=&submit=Search+Job&SORT=Addate
http://www.linkup.com/results.php?q=&l=
https://www.beyond.com/jobs/search?soid=3&k=&l=
http://austin.jobs

로 이용한다. 총 1200개의 구인 광고에 대해 아래 12가지 속성의 데이터 스키마를 이용하였는데, 이는 [1]의 실험에서 이용된 스키마와 같다.

1. Company: 회사 이름, 2. Job Title: 모집하는 직업 이름, 3-5. Country/State/City: 구인하는 장소, 6. Salary: 봉급, 7. Skill: 요구하는 전문적인 기술, 8-9. Desired/Required Experience: 기대하는/요구하는 경력, 10-11. Desired/Required Education: 기대하는/요구하는 학력, 12. Posting Date: 구인 광고 게재 날짜

실험에 사용한 컴퓨터의 환경은 4개의 Intel Xeon E5-2680 v4 2.4GHz (14 cores per CPU), 756-GB DRAM, 8개의 NVIDIA Tesla P40, Ubuntu 14.04이다.

본 실험에서는 직접 순환 신경망(RNN), 장단기 기억 신경망(LSTM), 어텐션 기반 컨볼루션 신경망(ABCNN) 기반 정형화 방법을 구현하여 대조군으로 이용하였고, 규칙 학습 기반의 [1]을 대조군으로 이용하였다. 기존의 신경망 기반 방법들 [3, 4, 5, 7, 8]은 앞서 설명하였듯이 본 논문에서 풀고자 하는 문제와 다른 문제를 다루고 있어 대조군에서 제외하였다.

6.2 실험 결과

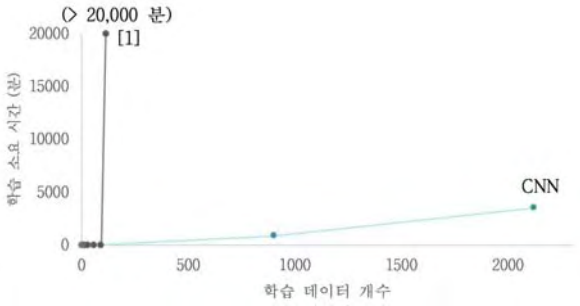

그림 3 [1]과 개발한 방법의 학습 소요 시간 비교.

그림 3은 [1]과 개발한 방법의 학습 소요 시간을 비교한 표이다. [1]은 학습 데이터 900개에 대해 300시간 이상이 소요되어 학습을 중단하였다. 기존 데이터셋에서 무작위로 추출한 학습 데이터 90개, 테스트 데이터 100개에 대해서 [1]은 정밀도 0.78, 재현율 0.32로 나타났다. [1]은 패턴 매칭 규칙을 학습하는 방법으로 정밀도는 높게 나타나지만, 재현율이 매우 낮게 나타난다. 개발한 컨볼루션 신경망 기반 기법은 기존 데이터셋에 대해서 정밀도 0.77, 재현율 0.60으로 [1]과 유사한 수준의 정밀도를 나타내면서 [1]보다 0.28 높은 우수한 재현율을 나타내었다.

그림 4는 다양한 신경망 기반의 정형화 방법과 개발한 방법의 성능을 비교한 결과이다. 순환 신경망 기반 방법은 학습 소요 시간이 가장 짧지만, F-1 점수가 개발한 방법 대비 0.29 낮게 나타났으며, 네 가지 방법 중 가장 뒤떨어졌다. 어텐션 기반 컨볼루션 신경망의 경우 학습 소요 시간은 개발한 방법과 유사하나 F-1 점수는 개발한 방법 대비 0.06 낮게 나타났다. 장단기 기억 신경망의 경우 F-1 점수는 개발한 방법보다 0.02 낮게 나타났으며, 신경망의 특성상 학습 소요 시간이 매우 오래 걸리는 것으로 나타났다. 개발한 방법은 학습 소요 시간과 정밀도/재현율을 모두 고려했을 때 대조군보다 우수한 성능을 나타낸다고 할 수 있다. 구인 광고 데이터는 웹 데이터이기 때문에 각 문장의 구분이 불명확하다. 단일 문장을 정확히 구별하지 못하는 문제로 인해 입력 문장이 매우 길어질 수 있는데, 순환 신경망은 이러한 케이스에 대해 매우 취약하다. 순환 신경망과 마찬가지로 문장 내의 모든 정보를 기억해야 하는 장단기 기억 신경망 기반 방법의 경우 재현율이 낮아지는 것으로 나타났다. 반면, 본 논문에서 개발한 컨볼루션 신경망 기반 방법은 입력 문장 내에서 의미 있는 문구를 특정지어 처리하는 방법으로 문장 길이와 무관하게 처리할 수 있기 때문에 재현율이 상대적으로 높게 나타났다.

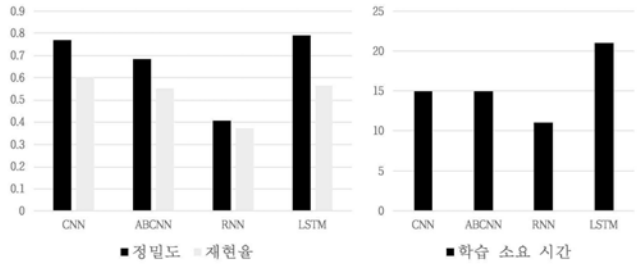

그림 4 다양한 신경망 기반 방법들과 개발한 방법의 성능 비교.

참고문헌

[1] Mary Elaine Califf and Raymond J. Mooney: Relational Learning of Pattern-Match Rules for Information Extraction. AAAI/IAAI 1999: 328-334

[2] Jonas Poelmans, Paul Elzinga, Alexey Neznanov, Guido Dedene, Stijn Viaene, Sergei O. Kuznetsov: Human-Centered Text Mining: A New Software System. ICDM 2012: 258-272

[3] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao: Relation Classification via Convolutional Deep Neural Network. COLING 2014: 2335-2344

[4] Daojian Zeng, Kang Liu, Yubo Chen, Jun Zhao: Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. EMNLP 2015: 1753-1762

[5] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, Maosong Sun: Neural Relation Extraction with Selective Attention over Instances. ACL (1) 2016

[6] Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, Houfeng Wang: A Dependency-Based Neural Network for Relation Classification. ACL (2) 2015: 285-290

[7] Daniil Sorokin, Iryna Gurevych: Context-Aware Representations for Knowledge Base Relation Extraction. EMNLP 2017: 1785-1790

[8] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, Wen-tau Yih: Cross-Sentence N-ary Relation Extraction with Graph LSTMs. TACL 5: 101-115 (2017)

[9] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, Christopher D. Manning: Position-aware Attention and Supervised Data Improve Slot Filling. EMNLP 2017: 35-45