ManiFL : 얕은 학습 기반의 더 나은 자연어처리 도구

신준철○, 김완수, 이주상, 옥철영
울산대학교, 한국어처리연구실
ducksjc@gmail.com, kimwansu@outlook.com, dosa510@naver.com, okcy@ulsan.ac.kr

ManiFL : A Better Natural-Language-Processing Tool Based On Shallow-Learning

Joon-Choul Shin○, Wan-Su Kim, Ju-Sang Lee, Cheol-Young Ock
University of Ulsan, Korean Language Processing Lab

요 약

근래의 자연어처리 분야에서는 잘 만들어진 도구(Library)를 이용하여 생산성 높은 개발과 연구가 활발하게 이뤄지고 있다. 이 중에 대다수는 깊은 학습(Deep-Learning, 딥러닝) 기반인데, 이런 모델들은 학습 속도가 느리고, 비용이 비싸고, 사용(Run-Time) 속도도 느리다. 이뿐만 아니라 라벨(Label)의 가짓수가 굉장히 많거나, 라벨의 구성이 단어마다 달라질 수 있는 의미분별(동형이의어, 다의어 번호 태깅) 분야에서 딥러닝은 굉장히 비효율적인 문제가 있다. 이런 문제들은 오히려 기존의 얕은 학습(Shallow-Learning) 기반 모델에서는 없던 것들이지만, 최근의 연구경향에서 딥러닝 비중이 급격히 증가하면서, 멀티스레딩 같은 고급 기능들을 지원하는 얕은 학습 기반 언어모델이 새로이 개발되지 않고 있었다. 본 논문에서는 학습과 태깅 모두에서 멀티스레딩을 지원하고, 딥러닝에서 연구된 드롭아웃 기법이 구현된 자연어처리 도구인 혼합 자질 가변 표지기 ManiFL(**Mani**fold **F**eature **L**abelling : ManiFL)을 소개한다. 본 논문은 실험을 통해서 ManiFL로 다의어태깅이 가능함을 보여주고, 딥러닝과 CRFsuite에서 높은 성능을 보여주는 개체명 인식에서도 비교할만한 성능이 나옴을 보였다.

주제어: 얕은 학습, 자연어처리, CRFsuite, 딥러닝, 다의어, 개체명, 혼합 자질 가변 표지기(ManiFL)

1. 서론

자연어는 기본적으로 순차라벨(Sequential Label)이라는 특성을 가지고 있고, 순차라벨을 다루는 다양한 확률통계기법들이 있으며, 이는 크게 얕은 학습(Shallow Learning)과 깊은 학습(Deep-Learning, 딥러닝)으로 나뉜다. 이런 통계기법들을 구현한 도구들이 존재하는데, 개발자는 이 도구를 이용해 개발의 생산성을 높일 수 있다. 예를 들어 얕은 학습 기반에서는 CRFsuite[1, 2]가 대표적이고, 딥러닝에서는 언어모델인 BERT[3]와, 딥러닝 도구인 텐서플로우[4, 5], 파이토치[6] 등이 대표적이다. 이런 도구들을 이용하면 개발자가 각종 통계기법들을 일일이 구현하지 않아도 되고, 통계기법들의 개별적인 특징과 사용법을 익히는 것만으로도 다양한 자연어처리 모듈을 개발할 수 있다.

근래에는 딥러닝의 연구비중이 급격하게 커지고 있는데, 이는 대부분의 영역에서 딥러닝의 정확률이 얕은 학습의 그것에 비하여 높고, 미학습된 패턴에서도 안정적이고 강건하게 작동한다는 장점이 있기 때문이다. 본래 딥러닝은 노이즈 데이터에서도 안정적이고, 자질간의 복합적인 XOR 같은 관계를 파악할 수 있으며, 여전히 잠재성이 많은 연구 분야임에는 틀림이 없다. 그러나 각종 비용 문제가 있고 라벨의 가지수가 너무 많으면 처리하기에 비효율적이라는 문제 등이 있어서 여전히 얕은 학습 기반의 모듈이 사용되는 곳이 많다.

얕은 학습 기반의 개발은 여전히 수요가 있음에도, 최근 들어서 딥러닝의 비중이 커짐으로 인해 얕은 학습 기반의 도구들은 관리가 되지 않고 있거나 새로이 연구 개발되지 않는 추세다. 최근 CPU의 발전 추이를 보면 속도보다는 코어의 수를 늘려가고 있기 때문에 멀티스레딩(또는 멀티프로세싱)의 중요성이 점점 증가하고 있는데 얕은 학습 기반의 자연어처리 도구들 중에서 멀티스레딩을 효율적으로 활용하는 도구를 찾기란 어렵다. 예를 들어 CRFsuite는 학습 단계에서 멀티스레드를 지원하지 않으며, 라벨링 단계에서도 스레드마다 객체를 개별적으로 생성해야하기 때문에 용량이 큰 학습사전을 메인메모리에 중복해서 로드해야하는 문제가 있다.

본 논문에서는 새로 개발된 얕은 학습 기반의 자연어처리 도구인 혼합 자질 가변 표지기 ManiFL을 소개하고, 이를 이용하면 딥러닝으로는 해결이 곤란한 다의어 번호 태깅을 효율적으로 수행할 수 있음을 보인다. 그리고 딥러닝과 CRFsuite에서 높은 정확률을 보이고 있는 분야인 개체명인식에서도 비교할만한 성능이 나옴을 실험을 통해 보인다.

2. 관련 연구

한국어 자연어처리에서 가장 잘 알려진 연구영역 중 하나는 형태소분석과 동형이의어 태깅이며, UTagger(2014)는 얕은 학습 기반의 형태소분석기 및 동형이의어태거로 그 정확률은 약 96.5%이다[7]. 이삼형(2018)은 UTagger를 대체하기 위해 딥러닝 기반의 형태소분석기를 개발하여 비슷한 정확률를 보였으나 분석 속도가 느려 아직까지는 속도면에서는 기존 방식의 형태소

분석기들을 대체하기 어려운 것으로 보인다[8]. 이는 딥러닝으로 한국어 형태소분석기를 설계하는 것은 가능하지만, 어절마다 토큰(형태소)의 개수가 달라지는 점과, 동형이의어 번호 라벨링에서 단어마다 의미번호가 1번부터 새롭게 매겨지는 점은 딥러닝으로 처리하기에 비효율적인 부분이다. 따라서 딥러닝 형태소분석기는 정확률 면에서 얕은 학습의 그것에 비하여 의미 있는 수준의 차이를 보이지 않고 있으며, 속도가 느려 아직은 실용적으로 사용하기에는 한계가 있다.

딥러닝 동형이의어 번호 라벨링 문제처럼, 다의어 번호 라벨링 또한 딥러닝으로 처리하기에 비효율적인 부분이 있어 얕은 학습 기반으로 설계하는 것이 유리한 측면이 있는데, 신준철(2020)은 모두의 말뭉치를 학습 및 실험 자원으로 활용하고 추가로 UWordMap(한국어 어휘 의미망)을 사용하여 얕은 학습 기반 다의어태거를 개발하여 문어 말뭉치에서 명사 정확률 87.63%의 결과를 보였다[9]. 이 연구에서는 각 자질별 가중치를 결정하는 방법이 자동화되어있지 않아 자질을 계속해서 추가하는데 어려움이 있고, 한 가지 자질종류에서는 하나의 가중치만 사용해야한다는 단점이 있다. 이런 문제는 잘 만들어진 자연어처리 도구를 사용하여 해결이 가능하다.

개체명인식은 현재 BERT를 이용한 딥러닝 방법의 연구가 활발하게 진행되고 있으며, 그 정확률은 말뭉치와 개체명 태그셋에 따라 차이가 있지만 한국어에서는 87%~92%의 정확률을 보이고 있다[10, 11]. 개체명인식기는 얕은 학습으로도 설계가 가능하며, 박서연(2021)이 5개의 태그(PS, LC, OG, DT, TI)와 가장 최근에 발표된 말뭉치인 “모두의 말뭉치”를 이용하여 F1 점수 90.54%을 보이는 개체명인식기를 개발하였고, 이것이 딥러닝과 비교할만한 정확률과 월등히 빠른 속도를 가짐을 보였다[12].

3. ManiFL

ManiFL은 2021년에 설계된 얕은 학습 기반의 자연어처리 도구이며, 기존의 얕은 학습 기반 도구들이 지원하지 않던 멀티스레딩 같은 기능을 지원하고, 딥러닝이 연구되면서 알려진 드롭아웃 기법을 얕은 학습에서도 지원하기 위해 설계되었다.

3.1 자질과 라벨

자질과 라벨은 텍스트로 구성되며, 자질은 라벨(정답)을 맞추기 위한 단서이자 근거로 작용한다. 본 논문에서는 예로 ‘기일’의 다의어 번호 라벨링 문제를 설정하고, 예문1을 이용하여 설명한다.

- 예문1) *다음 선고기일에는 꼭 참석하겠다.*
- 기일 05_00_01 : 정해진 날짜
- 기일 05_00_02 : 법원이나 소송 당사자 또는 그 소송에 관계되는 사람이 모여 소송 행위를 하는 특정한 날이나 기간.

예문1에서 나타난 ‘기일’에는 동형이의어 번호가 1~6까지 사전에 등재되어 있으며, 동형이의어 번호 5 안에 2가지 다의어가 등재되어 있다. 05_00_01과 05_00_02가 그 다의어들의 라벨이며, ‘기일’이라는 글자와 앞뒤로 나타나는 단어들이 정확한 라벨링을 위한 자질이 될 수 있는데, 예문1에서 핵심 적인 자질은 앞 단어인 ‘선고’라고 볼 수 있다.

자질에는 ‘자질종류’라는 개념이 있는데, ‘선고’는 “앞의 단어”라고 그 종류를 정의할 수 있다. ManiFL은 각 자질종류마다 고유의 이름을 정의해야하며, 이런 행위는 CRFsuite 등 다른 도구를 활용할 때에도 흔하게 나타나지만 필수는 아니었다. ManiFL은 자질종류의 이름을 반드시 지정하도록 강제한다는 특징이 있는데, 자질 디버깅에서 자질이름이 사용되기 때문이며, 사용자가 원한다면 가중치가 자질종류 마다 1개씩만 존재하도록 설정할 수 있기 때문이다. 예를 들어 “앞의 단어”가 ‘선고’일 때의 가중치와 ‘공판’일 때 다르게 적용되도록 설정할 수도 있고, 같게 설정할 수도 있다. 일반적으로 전자가 더 정확률이 우수한 편이지만 학습사전의 용량이 커지고, 속도도 조금 느려질 수 있는 단점이 있다.

3.2 드롭아웃

드롭아웃이란 딥러닝에서 과적합을 방지하기 위하여 학습 중에 특정 확률로 일부 뉴런을 비활성화 하여 뉴런의 출력값이 일시적으로 0이 되게 하는 것이다. ManiFL은 빈도가 낮은 자질은 확률적으로 그 값이 0이 되게 하는 드롭아웃 기능을 가지고 있다.

빈도의 제한과 확률은 사용자가 지정할 수 있으며, 이렇게 지정한 값 이하의 빈도를 가진 자질은, 지정한 확률로 드롭아웃이 시도된다. 드롭아웃이 시도되면, ManiFL은 0에서 빈도제한값 사이의 수를 무작위로 생성하고 이 값이 자질의 빈도보다 높다면 자질값을 일시적으로 0으로 계산한다. 예를 들어서 빈도제한은 20, 드롭아웃 확률을 90%로 지정하면, 빈도가 10인 자질이 실질적으로 드롭아웃될 확률은 약 45%가 된다. 왜냐하면 드롭아웃 시도확률 0.9와, 빈도10이 0~20사이의 무작위값보다 클 확률인 0.5를 곱하면 0.45가 되기 때문이다. 따라서 빈도가 낮을수록 드롭아웃 되기 쉽고, 빈도 20 초과의 자질은 드롭아웃 되지 않는다.

3.3 수학 모델

자질-라벨의 연결 정보에서 각 자질의 가중치를 최적화하기 위해 경사하강법을 사용하였다. 경사하강법은 가중치 최적화에 가장 잘 알려진 방법들 중 하나로, 손실함수의 값이 감소되도록 가중치를 갱신하는 것으로, (수식 1, 2)로 표현된다. J는 손실함수라고 하며, ManiFL은 교차엔트로피오차를 사용한다. w는 가중치이자 매개변수이며 a는 학습률이다. 즉, 경사하강법이란 손실함수 J가 가장 작은 값이 나오도록 기울기를 이용하여 매개변수를 갱신하는 방법이다.

$$J = \sum_{k} t_k \log y_k \qquad \text{(수식 1)}$$

$$w = w - a\nabla_w J(w) \qquad \text{(수식 2)}$$

경사하강법은 딥러닝에서도 이용되는데, 일반적으로 가중치들을 여러 그룹으로 나누는 미니 배치 기법이 같이 구현된다. ManiFL도 미니 배치 기법을 사용하고 있으며, 각 미니 배치를 1개의 스레드가 처리하도록 하여 멀티스레딩 기능을 지원하고 있다.

3.4 순차정보 처리

자연어처리에서 말뭉치나 문장은 단어가 순서를 가지고 나열된 것으로 보고, 각 단어나 어절마다 라벨이 있다면 이를 순차라벨 문제로 정의하고 각종 통계모델을 적용해볼 수 있다. 각 라벨을 모르는 상태에서 라벨을 은닉정보로 취급하면 HMM(Hidden Markov Model)을 적용할 수 있는데, 이것은 인접한 라벨끼리는 서로 확률관계가 있다는 가정 하에서 가능한 모든 라벨열 중에 최적해를 찾는 모델이다. 이 과정을 효율적으로 처리하기 위해 모든 라벨열을 비교하지는 않고, 동적프로그래밍 기법인 Viterbi를 사용할 수 있다.

실제로 순차라벨 문제에서 각 라벨은 최초에 미정된 상태이기 때문에 이상적으로는 모든 라벨열에서 최적해를 찾는 것이 정확률 측면에서 합리적인 방식이지만, 신준철(2014)은 한국어 형태소분석 및 동형이의어태깅 문제에서 각 라벨을 미정한 상태로 최적해를 찾는 방법을 쓰는 것과, 순차적으로 라벨을 결정하면서 넘어가는 방법을 비교해보니 정확률 차이가 미미하고 (96.49% vs 96.42%), 속도에서 큰 차이가 나기 때문에 (21.1초 vs 10.0초) 순차적으로 라벨을 결정하는 후자의 방법을 사용하는 것 또한 좋은 방법이라고 제안하였다[7]. ManiFL은 이런 연구를 바탕으로 설계되었으며, 사용자는 이전 라벨을 자질로 사용하는 방법을 사용할 수 있고, 이전 라벨을 미정 상태로 모든 라벨열에서 최적해를 찾아주는 Viterbi는 사용하지 않고 있다.

4. 다의어태깅

4.1 라벨

다의어태깅은 다의어 번호를 라벨링하는 문제로, 3.1절의 예문1을 예로 들 수 있다. 여기서 '기일'은 6가지 동형이의어가 있고 따라서 동형이의어 번호는 1~6이 존재한다. 이 중에서 5번 동형이의어에 2가지 다의어가 존재한다(이 번호 체계는 표준국어대사전을 기반으로 하는 울산대학교의 것[9]을 따른다.). 이처럼 의미번호는 단어마다 차이가 존재하기에 라벨의 가짓수는 수십만에 달할 수 있는데. 실험에 사용된 "모두의 말뭉치 다의어 2020"에는 명사에만 다의어 번호가 주석되어 있기 때문에 실험에서도 명사만 라벨링하였다.

4.2 자질

자질은 인접한 형태소나 어절, 사전의 뜻풀이나 한국어 어휘 의미망 등의 정보에서 얻을 수 있는데, 본 논문에서는 ManiFL의 사용 가능성을 간단히 증명하기 위해 인접 형태소만을 자질로 사용하였다. 자질은 총 5가지 종류만 사용되었으며 자세한 것은 표 1에 표시하고 있다. 각 형태소는 본체글자(예: 기일, 에, 는)와 의미번호(예:01, 05) 그리고 세종품사코드(예: NNG, VV)로 구성되어 있다.

표 1 다의어태깅 자질들

자질 이름	설명	예문1 '기일'의 자질
현재 형태소	현재 형태소만 사용	기일05NNG
좌1	바로 앞의 형태소 1개와 현재 형태소	선고04NNG 기일05NNG
좌2	좌측 2번째 형태소에서 현재 형태소까지 합친 것	다음01NNG 선고04NNG 기일05NNG
우1	현재 형태소와 뒷 형태소	기일05NNG 에JKB
우2	현재 형태소에서 우측 2번째 형태소까지 합친 것	기일05NNG 에JKB 는JX

4.3 실험 결과

실험에 사용된 모두의 말뭉치는 문어와 구어로 나뉘며 각각의 규모와 정확률은 표 2에 표시하였다. 정확률 측정을 위하여 학습은 문어의 80%와 구어의 80%를 동시에 사용하였고, 정확률을 실험할 때에는 미학습한 나머지 20% 분량을 사용하였으며, 문어와 구어를 분리하여 각각의 정확률이 측정하였다. 2020년의 연구[9]와 비교하면 정확률이 상당히 향상된 것을 볼 수 있으며, 문어의 정확률이 구어의 그것보다 여전히 높은 것을 볼 수 있다. 다만 2020년의 연구에서는 모두의 말뭉치의 다의어 번호 체계를 울산대학교 형식으로 맵핑하는 과정에서 일부 문제가 있었기 때문에 이로 인한 정확률 하락이 있었으며, 따라서 완전히 같은 조건에서 비교했다고 보기는 어렵다.

표 2 다의어 말뭉치 정보

구분	총 어절수	총 형태소수	정확률 (%)	2020년 연구[9] 정확률(%)
문어	2,000,213	4,506,499	92.59	87.63
구어	1,006,447	1,916,740	90.84	84.39

상술한 드롭아웃의 효과를 확인하기 위해서 드롭아웃 기능만 제거한 실험도 진행되었다. 이 실험에서는 문어 정확률만 측정하였고 그 결과 92.34%가 나왔으며, 표 2의 문어 정확률인 92.59%보다 낮음이 확인되었다.

이 실험에서 ManiFL은 라벨링 중에는 1초당 약 18만개의 형태소를 처리하였는데 1개의 스레드만 사용한 것이다. 문어 전체가 4.5백만 형태소이며, 이중 20%인 90만 형태소를 처리하는데 약 5초 소모된다는 의미다. 학습은 최대 에폭(epochs)을 20으로 제한하고 진행되었고, 실험에 사용한 컴퓨터의 CPU는 AMD R9 3900X 12코어이며, 24개의 스레드를 사용하여 약 106초가 소모되었다. 스레드를 1개만 사용하였을 때에는 320초가 소모되었는데, 3배가량만 차이가 난 이유는 일부 구간에서는 멀티스레드가 사용되지 못하기 때문이다. 학습사전은 약 104 Mega

표 3 개체명인식 자질

번호	자질 종류 설명	예문2 '국가대표'의 자질
1	앞 라벨, 현재 형태소 모든 글자, 품사	OG-B 국가대표NNG
2	앞 라벨, 현재 형태소 처음과 마지막 글자, 품사	OG-B 국표NNG
3	앞 라벨, 현재 형태소 마지막 두 글자, 품사	OG-B 대표NNG
4	앞 라벨, 현재+다음 형태소 처음과 마지막 글자, 품사	OG-B 국표NNG 팀NNG
5	앞 라벨, 이전+현재 형태소 처음과 마지막 글자, 품사	OG-B 컬링NNG 국표NNG
6	앞 라벨, 현재+다음 형태소 마지막 두 글자, 품사	OG-B 대표NNG 팀NNG
7	앞 라벨, 이전+현재 형태소 마지막 두 글자, 품사	OG-B 컬링NNG 대표NNG
8	앞 라벨, 현재+다음 2개 형태소 처음과 마지막 글자, 품사	OG-B 국표NNG 팀NNG 이JKS
9	앞 라벨, 이전 2개+현재 형태소 처음과 마지막 글자, 품사	OG-B 대국NNP 컬링NNG 국표NNG
10	앞 라벨, 현재 형태소 첫 글자, 다음 2개 형태소 품사	OG-B 표NNG NNG JKS
11	앞 라벨, 이전, 현재 형태소 첫 글자, 다음 형태소 품사	OG-B NNG 표NNG NNG
12	앞 라벨, 이전, 현재 형태소 첫 글자, 다음 2개 형태소 품사	OG-B NNG 표NNG NNG JKS

Bytes이며 ManiFL은 라벨링 중에 이 용량만큼 메인메모리를 사용한다고 볼 수 있다.

5. 개체명인식

5.1 라벨

실험에서 개체명태그셋은 박서연(2021)의 연구[12]와 동일하게 PS(사람), LC(장소), OG(단체), DT(날짜), TI(시간)을 사용하였다. 각 형태소마다 개체명의 시작인지 아닌지를 구분하기 위해 B또는 I가 라벨의 끝에 추가되며, 따라서 실제로는 PS-B, PS-I, LC-B, LC-I 와 같은 형태이고, 개체명이 아닌 것은 O 라벨이 붙는다. 말뭉치는 "모두의 말뭉치 개체명 2020"을 사용하였다.

5.2 자질

본 논문에서는 자질을 설명하기 위해 예문2에서 '국가대표'의 개체명 라벨을 결정하는 문제를 가정하며, 모든 자질은 표 3에 나타내었다. 각 자질들은 현재 형태소(국가대표)와 앞의 형태소(컬링) 그리고 다음 형태소(팀) 등으로 구성되며, 형태소의 전체 글자(국가대표)를 다 사용하기 보다는 첫 글자(국) 또는 끝 글자(표)를 활용하는 경우가 많다. 그리고 모든 자질은 앞의 라벨(컬링, OG-B)을 포함한다.

- 예문2) *대한민국/NNP 컬링/NNG 국가대표/NNG+팀__01/NNG+이/JKS 은메달/NNG+을/JKO 따__01/VV+았/EP+다/EF+./SF*

5.3 실험 결과

실험에 사용된 모두의 말뭉치 개체명은 학습용이 약 360만 형태소 분량이고, 정확률 측정용 미학습 분량은 약 89만 형태소다. 성능은 개체명인식에서 주로 사용되는 F1 점수 측정 방식이 사용되었고, 이것은 정확률과 재현율을 종합한 것으로 실험 결과는 표 4에 표시하였다.

학습은 최대 에폭 20으로 설정하여 진행되었고, 이 실험에 사용한 컴퓨터의 CPU는 AMD R9 5900X 12코어이며

표 4 개체명인식 실험 결과

개체명	설명	정확률	재현율	F1 점수
DT	날짜	96.90	96.60	96.75
LC	장소	81.40	87.75	84.46
OG	단체	86.35	82.88	84.58
PS	사람	91.99	93.22	92.60
TI	시간	95.14	94.12	94.63
micro avg.		90.53	90.68	90.60

총 24개의 스레드가 사용되었으며, 학습의 전 과정에서 소요된 시간은 689초(11분 29초)였다. 라벨링에는 1개의 스레드만 사용하였고, 89만 형태소를 처리하는데 약 16초가 소요되었으며 이는 말뭉치 파일을 열고 자질 텍스트를 생성하고 라벨링 결과를 파일로 출력하는 등의 모든 과정을 포함한다. 학습사전은 약 722 Mega Bytes 크기다.

실험환경은 박서연(2021)이 모두의 말뭉치와 CRFsuite를 사용하여 F1점수 90.54%를 얻은 연구[12]와 거의 동일하지만, 그 연구에서는 형태소의 글자와 품사 태그 외에도 어휘 의미망과 의존 관계 등 다양한 자질들이 추가되어 있었다. 본 논문의 실험에서는 이런 추가 자질들이 없음에도 오히려 조금 높은 F1점수 90.60%가 나온 것에는 "말뭉치 오류 일부 개선", "기본 자질 구성 차이", "오차 범위" 등 다양한 원인들이 있겠지만, "ManiFL과 CRFsuite의 차이"도 있다. ManiFL은 CRFsuite와 달리 viterbi 기능이 없기 때문에 정확률 면에서 불리할 수고 있고, 드롭아웃으로 유리한 면도 있겠는데, 이번 실험 결과에서 ManiFL은 CRFsuite 보다 더 적은 자질을 사용하고도 비교할만한 정확률을 보이고 있어 훌륭한 도구가 될 수 있을 것으로 분석된다.

6. 결론

본 논문은 새로이 개발된 얕은 학습 기반의 자연어처리 도구 ManiFL을 소개하고, 이를 이용하면 딥러닝으로는 라벨링이 매우 어려운 다의어 번호 라벨링이 가능하다는 것과, 최근 딥러닝에서 높은 정확률을 보이고 있는 개체명인식에서도 비교할만한 정확률이 나옴을 실험을

통해 보이고 있다. ManiFL은 속도 면에서 얕은 학습다운 빠른 속도를 가지고 있고, 기존의 잘 알려진 얕은 학습 도구인 CRFsuite 등에서는 지원하지 않던 멀티스레딩을 지원하고 있어서 CPU코어가 늘어날수록 대용량을 처리하는데 유리한 점이 있다. 또한 딥러닝에서 사용되던 드롭아웃이 구현되어 있어서 기존의 다른 얕은 학습보다 정확률이 소폭 높을 것으로 기대된다. 다만 얕은 학습의 태생적 한계로 인해 일반적으로 딥러닝에 비하여 정확률이 낮을 수 있으나, 딥러닝이 가진 비용적 문제와, 복잡한 라벨에서 나타나는 비효율성을 해결할 수 있기 때문에 형태소분석이나 다의어태깅 등 일부 영역에서는 유용할 것이고, 무거운 시스템 운용이 불가능하거나, 얕은 학습 기반으로도 만족할만한 성능이 나오는 분야에서도 사용되어질 수 있다.

ManiFL은 아직 파이선이나 자바 등의 개발언어를 지원하지 않고 오직 C++만 지원하고 있기에 이런 점들이 차후에 개발되어 지원된다면 사용성이 크게 오를 것으로 기대된다. 딥러닝은 자질간의 XOR 관계 까지도 알아낼 수 있는 등 “자질 관계 분석” 능력이 굉장히 뛰어나며, 이런 특성 때문에 연구원들은 자질조합 보다는 신경망 네트워크와 레이어의 구조 등 “자질조합 이외의 것”들을 연구하는데 집중할 수 있다. 이와 다르게 얕은 학습은 자질구성과 자질조합 연구에도 많은 노력이 필요한데, 이러한 과정을 ManiFL과 같은 도구가 일부 도움을 주기 위해서 “주어진 기본적인 자질들로 다양한 자질조합을 자동으로 생성하는 기능”이 연구되어질 필요가 있다. 그리고 ManiFL은 의존관계, 의미역 등 다양한 자연어처리 문제영역에 응용해볼 수 있는데, 각각의 정확률을 CRFsuite 또는 딥러닝의 그것과 비교해볼 필요가 있다.

감사의 글

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(No.2013-2-00131, (엑소브레인-총괄/1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술개발)과 2021년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2020R1I1A1A01073665)의 연구결과임.

참고문헌

[1] Naoaki Okazaki, https://www.chokkan.org/software/crfsuite/

[2] John Lafferty and Andrew McCallum and Fernando Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data”, Proceedings of the International Conference on Machine Learning (ICML-2001), 2001.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of deep bidirectional transformers for language understanding", In Proceedings of NAACL, 2019.

[4] Google, “MNIST For ML Beginners,” (https://www.tensorflow.org/).

[5] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... and Kudlur, M. . "TensorFlow: A System for Large-Scale Machine Learning", OSDI, Vol. 16, pp. 265-283, 2016.

[6] https://pytorch.kr

[7] Joon-Choul Shin, C. Y. Ock, "Korean Homograph Tagging model based on Sub-Word Conditional Probability", Journal of KIPS : Software and Data Enginering, Vol. 3, No. 10, pp. 407-420, Oct. 2014. (in Korean)

[8] 이삼형, “2018년 국어 기초 어휘 선정 및 어휘 등급화 연구”, 국립국어원 과제 결과 보고서, 2018.

[9] 신준철, 이주상, 옥철영, “모두의 말뭉치를 이용한 한국어 다의어 분별”, 한글 및 한국어 정보처리 학술대회, pp. 205~210, 2020.

[10] SK텔레콤, “KoBERT와 CRF로 만든 한국어 개체명인식기”, https://github.com/eagle705/pytorch-bert-crf-ner

[11] 박광현, 나승훈, 신종훈, 김영길, "BERT를 이용한 한국어 자연어처리: 개체명 인식, 감성분석, 의존 파싱, 의미역 결정", 한국정보과학회 학술발표논문집, pp. 584~586.

[12] 박서연, 옥철영 (2021). 한국어 어휘 의미망을 활용한 CRF 모델 기반 개체명 인식. 정보과학회논문지, Vol. 48, No. 5,, pp. 556-567.