Journal of the Korean Society of Marine Engineering, Vol. 42, No. 7 pp. 580~586, 2018
J. Korean Soc. of Marine Engineering (JKOSME)
https://doi.org/10.5916/jkosme.2018.42.7.580

ISSN 2234-7925 (Print)
ISSN 2234-8352 (Online)
Original Paper

LiAS: 점진적 사전 확장과 기계학습을 활용한 선형구조의 언어정보 부착 시스템

노경목[1] · 김창현[2] · 최민석[3] · 윤 호[4] · 김재훈†

(Received July 17, 2018 ; Revised August 8, 2018 ; Accepted August 8, 2018)

LiAS: A linguistic information annotation system for linear structures of language based on incremental expansion of dictionary and machine learning

Kyung-Mok Noh[1] · Chang-Hyun Kim[2] · Minseok Choi[3] · Ho Yoon[4] · Jae-Hoon Kim†

요약: 최근 많은 관심을 가지고 있는 기계학습이나 인공지능 시스템들은 많은 학습 자료가 필요하다. 이런 학습 자료는 일반적으로 수동으로 구축되며 많은 시간과 노력이 필요할 뿐 아니라 수동으로 구축된 학습 자료의 일관성을 유지하는 것이 매우 어려운 일이다. 이런 문제를 완화하려고 본 논문에서는 대량의 학습 자료를 구축하기 위한 언어 정보 부착 시스템을 개발한다. 이 시스템은 자동 부착(automatic tagging), 오류 수정(error correction), 확률 검수(sample examination), 사전 확장(dictionary expansion) 및 점진 학습(incremental learning) 등의 과정을 반복하면서 학습 자료(언어 정보 부착 말뭉치)를 확장한다. 자동 부착은 사전과 기계학습 시스템을 이용해서 원문에 개체명, 구묶음, 전문용어 등과 같은 언어 정보를 부착한다. 오류 수정은 자동 부착된 언어 정보의 오류를 수정한다. 확률 검수는 오류 수정된 언어 정보로부터 확률적으로 일부의 문장을 추출하여 검수한다. 사전 확장 및 점진 학습은 검수된 문서를 학습 자료에 포함하여 사전을 확장하고 기계학습 시스템의 성능을 확장한다. 이 시스템은 서버-클라이언트 모델로 구현되어 여러 사람이 협업을 통해서 더욱 효과적으로 언어 정보를 부착할 수 있었으며 많은 시간과 노력을 절약할 수 있었다. 결과적으로 대량의 학습 자료를 효과적으로 구축하는 데 크게 기여하였다.
주제어: 부착 도구, 정보 추출, 해양 전문용어, 개체명, 구묶음

Abstract: Systems based on machine learning and/or artificial intelligence, particularly deep learning, rely on large-scale training data for supervised learning. Manually building the data can not only be time-consuming and expensive but also cause data inconsistency. To alleviate the burden, we introduce an annotation tool for semi-automatically building the training data (tagged corpora). The tool increases the training data by iterating the following four steps as follows: (1) automatic tagging: assign tags to text spans (named entity, chunk, term, among others) based on a predefined dictionary and a machine learning-based recognition system; (2) error correction: manually correct errors in the tagged spans through a GUI. (3) sample examination: sample and examine the corrected spans through the GUI. (4) dictionary expansion and performance enhancement: expand the size of the dictionary through the examined spans and enhance the performance of the machine learning system. We can collaboratively build training data through this tool with the client–server model, thereby saving the time and effort required to build it, and moreover, retaining the data consistency more easily. As a consequence, we have observed that this tool contributes to effectively build training data.
Keywords: Annotation tool, Information extraction, Marine term, Named entity, Chunking

1. 서 론

최근 심층학습(deep learning)은 대량의 학습 자료를 이용해서 영상인식, 음성인식, 자연언어처리 등의 분야에서 매우 좋은 성능을 보이고 있다**[1]**. 특히 지도학습을 이용한 대부분의 자연언어처리 시스템들은 대량의 학습 자료, 즉 언어 정보 부착 말뭉치에 의존하고 있다. 그러나 이와 같은 학습 자료를 구축하는 데는 많은 시간과 비용과 노력이 필요하다**[2][3]**. 또한 여러 작업자들이 자료를 구축할 때 작업자들의 실수뿐 아니라 작업자 간의 의견 불일치로 구축된 학습 자료에는 많은 오류가 포함되어 있다**[4]**. 또한 학습

† Corresponding Author (ORCID: http://orcid.org/0000-0001-8655-2591): Professor, Department of Computer Engineering, Korea Maritime and Ocean University, 727, Taejong-ro, Yeongdo-gu, Busan 49112, Korea, E-mail: jhoonk@kmou.ac.kr, Tel: 051-410-4574
1 M.S. candidate, Department of Computer Engineering, Korea Maritime and Ocean University, E-mail: kmq7542@gmail.com, Tel: 051-410-4896
2 Senior Researcher, Electronics and Telecommunications Research Institute, E-mail: chkim@etri.re.kr, Tel: 042-860-6485
3 M.S. candidate, Department of Computer Engineering, Korea Maritime and Ocean University, E-mail: ehdgus5136@naver.com, Tel: 051-410-4896
4 M.S. candidate, Department of Computer Engineering, Korea Maritime and Ocean University, E-mail: 4168615@naver.com, Tel: 051-410-4896

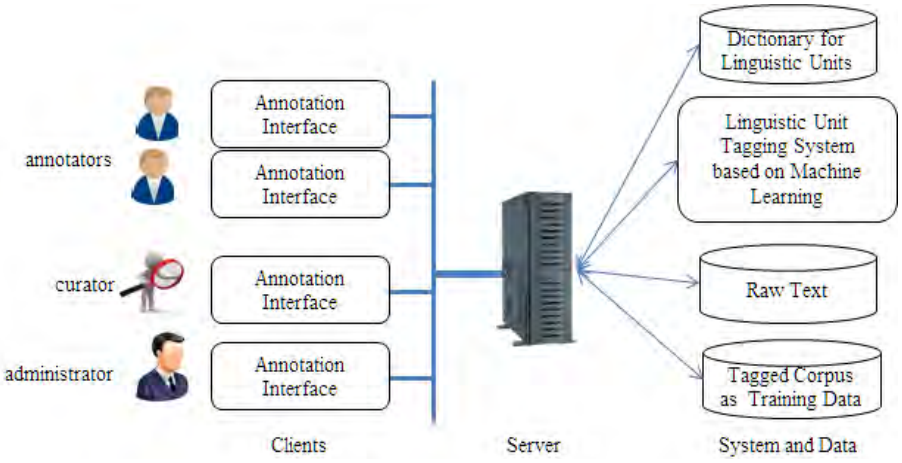

Figure 1: Overall system configuration of LiAS

자료 구축을 위한 지침서에서 다양한 모든 언어 현상을 다룰 수 없기 때문에 또 다른 불일치를 나타낸다. 이런 오류를 최소화하기 위해서 많은 연구자들은 말뭉치 구축 도구를 개발하였다**[5]-[12]**. 많은 도구들은 GUI를 통해서 효율적인 말뭉치를 구축하는 것에 초점을 맞추고 있다. 예를 들면 어떤 단어에 개체명 정보가 부착되면 이후에 나오는 같은 단어에는 같은 개체명 정보를 부착한다. 그러나 이것이 다른 파일을 작업할 경우에는 적용되지 않으며 다른 작업자에게도 영향을 주지 못한다. 이런 문제는 작업자 간의 일치도가 크게 떨어뜨리게 된다.

본 논문에서는 작업자 간 사전을 공유함으로써 이런 문제를 개선하려고 한다. 그러나 초보 작업자의 경우에는 작업의 신뢰도 떨어지므로 바로 사전에 등록하지 않고 검수자(curator)의 검수를 통과하면 그 결과를 사전에 등록한다. 따라서 본 논문에서는 작업의 신뢰도를 높이기 위해 네 단계의 사용자(참관자(guest), 작업자(annotator), 검수자(curator), 관리자(administrator))가 작업에 참여한다(2.1절 참조). 본 논문에서 제안된 시스템은 자동 부착, 오류 수정, 확률 검수, 사전 확장 및 점진 학습 등의 과정을 반복하면서 학습 자료를 확장한다(2.5절 참조). 자동 부착은 사전과 기계학습 시스템을 이용해서 원문에 개체명, 구묶음, 전문용어 등과 같은 언어 정보를 부착한다. 오류 수정은 자동 부착된 언어 정보의 오류를 수정한다. 따라서 본 논문에서 제안된 언어 정보 부착 시스템은 반자동 말뭉치 구축 도구이다. 확률 검수는 오류 수정된 언어 정보로부터 확률적으로 일부의 문장을 추출하여 검수한다. 사전확장 및 점진 학습은 검수된 문서를 학습 자료에 포함하여 사전을 확장하고 기계학습 시스템의 성능을 확장한다.

본 논문에서 제안된 언어 정보 부착 시스템은 서버-클라이언트 모델로 구현되어 여러 사람이 협업을 통해서 더욱 효과적으로 언어 정보를 부착할 수 있었으며 많은 시간과 노력을 절약할 수 있었다. 결과적으로 대량의 학습 자료를 효과적으로 구축하는 데 크게 기여하였다.

본 논문은 2장에서 제안된 언어 정보 부착 시스템(LiAS)에 대해서 자세히 설명한다. 3장에서 LiAS의 구현 및 다른 도구들과 비교하고 분석한다. 4장에서 결론 및 향후 연구를 기술한다.

2. LiAS: 언어 정보 부착 시스템

본 절에서는 선형구조를 가지는 언어 정보 부착 시스템(Linguistic information Annotation System, LiAS)에 대해서 자세히 기술한다. 언어 정보는 언어 단위에 따라 구조적인 단위(structure unit)과 연속적인 단위(consecutive unit)로 나눌 수 있다. 구조적인 단위는 의존관계(dependency relation), 구문트리(syntactic tree), 상호참조(co-reference) 등과 같이 2차원 공간에서 언어 정보가 표현되는 것을 의미한다. 반면에 연속적인 단위는 음절(syllable), 형태소(morpheme), 단어(word), 개체명(named entity) 등 연속적인 공간(text span)에서 언어 정보를 표현하는 것을 의미하며 본 논문에서 연속적인 단위를 언어 단위라고 한다. 예를 들면 개체명(예: 한국해양대학교), 구묶음(예: ~ㄹ 수 있다), 복합명사(예: 공중전화), 전문용어(예: 가스연료추진선박) 등을 본 논문에서는 언어 단위라고 한다. **Figure 1**은 LiAS의 전체 시스템 구성도이며, 크게 사용자 인터페이스(annotation interface)와 서버(Server)와 인식 시스템(tagging system)으로 구성된다. 각 구성요소는 이하의 절에서 자세히 설명한다.

2.1 사용자 인터페이스

사용자는 앞에서 언급했듯이 참관자와 작업자와 검수자와 관리자로 구분된다. 참관자는 실제 언어 정보 부착에 참여할 수 없으며, 언어 정보 부착을 확인하거나 숙련된 작업자의 작업 결과를 확인할 수 있다. 따라서 언어 정보에 대한 충분한 지식을 없는 미숙련 작업자를 교육하려는 용도로 사용된다. 작업자는 언어에 대한 약간의 지식[1]을 가지

[1] 일반적으로 언어 정보 부착 작업자는 응용 분야에 대한 충분한 언어적인 지식을 보유해야 한다. 그러나 본 시스템서는 참관자로서 일정 기간 동안 교육 후에 작업을 진행하며, 많은 정보를 서로 공유하고 풍부한 도움말을 통해서 스스로 학습이 가능하므로 약간의 지식만으로도 충분했다.

Figure 2: Screenshot of using GUI to annotate linguistic units

는 사람으로 먼저 응용분야에 따른 언어 단위(예 개체명)를 찾고, 해당 언어 단위에 정보(tag)를 부착한다. 검수자는 작업자가 부착한 정보가 올바른지를 확인하는데 모든 문장이나 문서를 확인할 수 없으므로 문장이나 문서의 약 1%임의 추출하여 오류가 3% 미만일 경우 검수를 완료한다. 관리자는 작업 양이나 서버와 사전 등을 관리한다. 전체적인 인터페이스의 구성은 **Figure 2**와 같다. **Figure 2**에서 왼쪽 영역(①)은 작업자들의 파일 관리자이며 이 영영의 위 부분은 원문(raw text)이고 아래 부분은 언어 정보 부착 말뭉치(tagged corpus)이다. 가운데 영역(②)는 작업 공간으로 원하는 언어 단위(linguistic unit)를 마우스로 선택하고 단축키(shortcut key)로 언어 정보(tag)를 부착한다. 이하의 파일에서 부착된 언어 단위를 찾아서 같은 언어 정보를 부착한다. 이 경우 같은 언어 단위가 서로 다른 정보를 가질 수 있으므로 수정이 가능해야 한다(2.2절 참조). 부착된 언어 정보를 수정하려면 마우스 오른쪽 단추를 이용하여 언어 정보를 삭제하거나 추가하는 방법으로 수정할 수 있다. 수정의 범위는 선택된 언어 단위 하나만 수정하거나 파일 전체의 같은 언어 단위를 수정할 수도 있다. 오른쪽 영역(③)은 검수자의 영역이며 검수 요청된 파일을 임의로(사용자별 약 1%) 선택하여 정밀도(precision)이 0.97 이하일 경우에 검수를 보류하고 작업자에게로 되돌려 보내서 전체를 다시 작업하도록 한다. 검수자가 검수할 때도 ② 영역을 사용하며 오류가 발견되면 간단한 주석을 달아 둔다. 이 정보는 검수가 거절되었을 때, 작업자가 참조하여 수정할 수 있을 것이다. 사용자 인터페이스를 통한 구체적인 작업 과정은 2.4절에서 기술할 것이다.

2.2 사전

사전은 규칙 기반으로 언어 정보를 이용하는 가장 기본적인 방법 중 하나이다. 언어 단위는 연속적이므로 **Table 1**과 같은 형식으로 필요한 정보를 저장하며 길이가 1인 언어 단위는 저장하지 않는다.

Table 1에서 첫 번째 열은 사전 항목(Entry)이며, 예로 "서울시(Seoul City)"에서 "서울시"가 사전 항목이고 "Seoul City"는 번역어이다. 두 번째 열은 언어 단위의 정보(Tags)이며, 개체명(named entity)의 경우는 인명(PERson), 기관명(ORGanization), 지명(LOCation) 등이 있고, 전문용어의 경우에는 전문용어(YES)이고 그 밖의 모든 언어 단위의 정보는 NO라는 태그를 사용하고, 구묶음의 경우 보조용어구(VUX,

Table 1: An example of a dictionary for linguistic units

Entry	Tags	Frequency	Remarks
서울시 (Seoul City)	ORG	10	named entity
서울시 (Seoul City)	LOC	22	named entity
빌 클린턴 (Bill Clinton)	PER	5	named entity
간석지 (tidal flat)	YES	3	marine term
간척제방 (polder dike)	YES	1	marine term
ㄹ 수 있 (can)	VUX	20	chunk
사과 한 개 (a apple)	NX	2	chunk

Chunk of Auxiliary Verb), 명사구(NX, Chunk of Noun) 등이 사용된다. 세 번째 열은 지금까지 부착된 말뭉치(학습 자료)에서 출현한 빈도수(Frequency)를 나타낸다. 네 번째 열은 비고란(Remarks)으로 실제 사전에서 포함되어 있지 않다.

Table 1에서 사전 항목에서 모호성이 있을 경우에는("서울시") 여러 항목으로 저장되고 작업자에게 자동으로 추천할 때 빈도수가 높은 것을 우선적으로 추천한다. 이 빈도수는 작업이 지속되면 계속적으로 증가된다.

2.3 기계학습 기반 언어 정보 인식 시스템

인식 시스템은 기계학습 기반으로 설계되었으면 기본 언어 단위는 음절을 사용하며 여러 개의 음절이 하나의 정보를 가질 경우에는 IOB(Inside-Outside-Beginning) 형식**[13]**으로 정보를 부착한다. **Table 2**는 IOB 형식의 언어 정보를 보이고 있다.

Table 2: An example of IOB format for Korean named entities

Syllables	2	일		서	울	시	에	서		회	의	가		있	다	.
IOB tags	B	I		B	I	I	O	O		O	O	O		O	O	O
Remarks	DAT			LOC												

Table 2에서 첫 번째 행은 음절(Syllables)로 표현된 문장이며, 두 번째 행은 IOB 정보(IOB tags)를 표현하고 있다. IOB 정보에서 B는 언어 단위의 시작을 나타내고, I는 시작 이외의 언어 정보 범위 내에 있음을 나타내며, O는 그 외의 단어들을 표현한다. 따라서 LiAS는 원문(raw text)를 입력받아서 작업 인터페이스를 통하여 **Table 2**와 같은 형식의 말뭉치(학습 자료)를 생성하는 것이 주목적이다. 언어 정보 인식 시스템은 **[14]**를 이용한다. **[14]**에서는 조건부 무작위장(conditional random field, CRF)**[15]**을 이용하여 영어 개체명 인식에 적용된 모델이다. 본 논문에서는 이를 수정하여 입력 단위를 음절로 하고 한국어 언어 단위 인식 시스템을 구현하여 개체명, 구묶음, 전문용어 등을 인식하는 시스템을 구현하였으며, 언어 단위(개체명, 구묶음, 전문용어 등)에 따라 학습 말뭉치(학습 자료)가 다르므로 구현된 인식 시스템에 적합한 형식으로 변환하는 전처리 과정이 필요하다. **Table 3**은 언어 단위가 개체명일 경우 인식 시스템의 자질(feature)를 보이고 있다.

Table 3: An example of a feature set for Korean named entities

Kind	**Feature set**	**Examples**
Syllable	s_{-2}, s_{-1}, s_0, s_1, s_2	서, 울, 시, 에, 서
Bigram	$s_{-2,-1}$, $s_{-1,0}$, $s_{0,1}$, $s_{1,2}$	서울, 울시, 시에, 에서
Trigram	$s_{-2,0}$, $s_{-1,1}$, $s_{0,2}$	서울시, 울시에, 시에서
Character	$type(s_0)$	Hangul

Table 3에서 s_0는 현재 음절을 의미하고 s_i는 현재 음절을 기준으로하는 상대위치에 해당하는 음절을 의미한다. $s_{i:j}$는 i번째 음절에서 j번째 음절을 의미한다. $ctype(s)$는 음절 s의 유니코드 형태(type)를 의미하며, 한글(Hangul), 한자(Hanja), 기호(Symbol) 등으로 구분된다.

2.4 기계학습과 사전의 결합

본 논문에서는 사전 정보와 기계학습 기반 인식 시스템의 결과를 결합하여 최종 추천하는 방법을 사용한다. 그러나 결합할 때 두 종류의 오류가 발생할 수 있다. 하나는 경계 오류이고, 다른 하나는 부착 정보 오류이다. 경계 오류의 해결 방법의 대전제(prerequisite)는 최장일치법(longest preference)을 사용하며, **Table 4**와 같은 유형이 있을 수 있다.

Table 4: Types of overlapping combination between two recommendations of the dictionary and the recognition system

Type of overlapping		**Preference**	**Remarks**
Complete matching		no	
Disagreement		each	
Partial		longest	ambiguity
		longest	
Inclusive		dictionary	
		recognition	

% dictionary:

recognition:

Table 4에서 첫 번째(Complete matching)와 두 번째 행(Disagreement)은 경계 모호성이 존재하지 않아서 문제가 발생하지 않는다. 세 번째 행(Partial)의 경우는 부분적으로 겹치는 경우로 경계를 명확하게 정할 수 없으므로 두 중에 길이가 긴 언어 단위를 선호한다. 네 번째 행(Inclusive)은 둘 중 어느 하나가 다른 언어 단위에 완전히 포함되므로 이 경우도 긴 언어 단위를 선호한다. 언어 정보 부착 오류의 해결 방법의 대전제는 사전 우선 정책(dictionary preference)이다. 대부분은 경계 오류에서 이미 해결되었으나 **Table 4**의 첫 번째 행(Complete matching)의 경우는 언어 정보 부착 오류가 발생할 수 있으며 이 경우 사전 우선 정책을 사용해서 해결한다.

2.5 작업 시나리오

학습 자료의 구축 과정은 다음과 같은 과정을 통해서 완성된다.

① **파일 선택**: 원시 말뭉치로부터 작업할 파일을 선택한다.
② **자동 부착**:
  ②-1: 선택된 파일은 사전을 이용하여 언어 정보를 부착한다.

②-2: 동시에 기계학습 기반 언어 정보 인식 시스템을 통해서 언어 정보를 부착한다.
②-3: ②-1과 ②-2의 결과를 결합한다(2.4절 참조).
③ **오류 수정**:
③-1: 사용자 인터페이스를 통해서 오류를 수정한다.
③-2: ①에서 ③-1까지의 과정을 여러 번 반복한다. 즉 여러 개의 파일에 대해서 오류를 수정한다.
④ **검수 요청:** 오류가 수정된 파일에 대해 검수자에게 검수를 요청한다.
⑤ **검수 시행**:
⑤-1: 요청된 파일에서 약 1% 정도의 파일을 선택하고 사용자 인터페이스를 통해 오류가 있는지를 확인한다.
⑤-2: 정보가 부착된 언어 단위들 중에서 오류가 3% 미만이면 검수를 완료한다.
⑤-3: 오류가 3% 이상이면 다시 검토하도록 검수 요청을 되돌려 보낸다.
⑥ **사전 및 인식 시스템 개선**:
⑥-1: 검수가 완료된 파일들에 포함된 언어 정보를 **Table 1**과 같은 형식으로 사전에 추가한다.
⑥-2: 검수가 완료된 파일을 학습 말뭉치에 추가하여 확장된 말뭉치를 이용해서 학습하여 인식 시스템의 성능을 개선한다.
⑦ **작업 완료**: 계획된 모든 파일에 대해서 언어 정보가 부착되면 작업을 완료한다.

위의 과정은 여러 명의 작업자가 동시에 수행되며, 검수가 완료된 정보를 서로 공유할 수 있어서 효과적으로 학습 자료를 구축할 수 있을 뿐 아니라 신뢰성이 높고 일관적인 학습 자료를 구축할 수 있을 것이다.

3. LiAS의 구현 및 토의

3.1 LiAS의 구현

LiAS는 Java와 Python3으로 구현되었다. 사용자 인터페이스는 Java로 개발되었으며, 언어 분석 도구는 Python3로 구현되었다. 사전은 트라이(Trie, pytrie 모듈을 수정함)를 이용했으며 검색은 최장일치를 적용한다. 기계학습 기반 언어 정보 인식 시스템은 2.3절에서 언급한 바와 같이 CRF(NLTK[2] CRFTagger 모듈 사용)를 이용해서 구현되었으며, 연속적인 언어 단위만 인식하므로 개체명, 구묶음, 전문용어, 복합명사, 연어 등의 언어 단위를 같은 시스템으로 인식할 수 있다. 현재는 개체명과 구묶음은 학습이 완료되어 적용되었으며, 전문용어와 복합명사와 연어 등은 차후에 학습 자료가 준비되는 대로 적용할 계획이다. 현재 클라이언트인 사용자 인터페이스(**Figure 2** 참조)는 Windows에서 실행하고 있으며 서버인 언어 분석 시스템은 Linux에서 실행되고 있다.

[2] http://www.nltk.org/

3.2 시스템 비교 분석

본 절에서는 다양한 언어 정보 부착 시스템들은 간략히 소개하면서 이들 시스템은 비교하고 분석한다(**Table 5)[8][16][17]**. **Table 5**에서 소개된 시스템은 대부분 공개된 시스템(PPEditor 제외)이며 유니코드를 지원하므로 한국어를 기본적으로 다룰 수 있을 것이나 한국어 언어 정보 인식 시스템이 Plug-in 형식으로 추가할 수 없어서 한국어 자동 추천 기능은 잘 동작하지 않는다.

Table 5의 첫 번째 열(Systems)은 시스템 이름과 출처를 나타내고, 두 번째 열(OS)은 실행할 수 있는 운영체제를 나타낸다. 대부분의 시스템은 Java와 Python으로 개발되어 OS에 관계없이 실행할 수 있으나, Anafora과 BRAT는 Mac과 Linux에만 실행되며, C#으로 개발된 PPEditor을 Windows에만 실행된다. LiAS는 Java와 Python3로 구현되어 OS에 관계없이 설치 및 실행이 가능하다.

세 번째 언어 열(Development language)은 주된 개발 언어를 나타내며, 대부분은 Java나 Python으로 개발되었다. 일반적으로 Java나 Python을 OS에 관계없이 한국어를 지원할 수 있다. 그러나 Python2의 경우 한국어가 정상적으로 지원하기 위해서 코드 변환에 관련된 여러 가지 기능을 지원하는 보조 모듈이 필요하다. 따라서 자연스럽게 한국어는 지원할 수 없으므로 많은 부분에서 프로그램의 수정이 필요하다. LiAS는 Java와 Python3로 구현되었다.

네 번째 행(Self-consistency)은 다른 어떤 도구(tool)에 의존하지 않으면 동그라미(O)이고, 다른 도구에 의존하면 가위표(X)이다. PPEditor는 사용자 인터페이스에 초점을 맞추고 있으며, 언어 분석 도구는 외부의 시스템들을 사용하였다(Kim and Park, 2006). LiAS는 다른 도구에 관계없이 시스템 자체로 실행할 수 있다.

다섯 번째 열(Linguistic information)은 부착할 수 있는 언어 정보의 범위를 나타낸다. 여기서 span은 전문용어와 같이 문자열의 일정 부분을 언어 단위로 지정할 수 있음을 의미하고 relation은 대용어 참조(co-reference)와 같이 어떤 두 언어 단위들 사이를 관계를 지정할 수 있음을 의미하며, struct는 의존구문분석과 같은 언어 구조를 지정할 수 있음을 의미한다. LiAS는 연속적인 언어 단위에 초점을 맞추어서 설계되고 구현되어 많은 계산 자원을 필요하지 않으며 하나의 알고리즘으로 다양한 언어 단위(개체명, 구묶음, 전문용어, 복합명사 등을 사용할 수 있다.

여섯 번째 열(Recommendation)은 자동 추천 기능이 포함되어 있는지를 의미한다. 사용자가 원하거나 초기에 파일을 불러올 때 언어 단위의 정보를 제공하는 기능이다. LiAS는 초기에 파일을 불러올 때 사전과 인식 시스템으로부터 초기 언어 정보(태그)를 제공할 수 있다. 물론 선택사항(option)으로 사전만 이용하거나 인식 시스템만도 이용할 수 있다.

일곱 번째 열(Dictionary)는 외부 사전을 이용할 수 있는지를 의미한다. 개체명과 같은 경우에는 인터넷 등으로부터

Table 5: Comparison with annotation tools for linguistic units

Systems	OS	Development language	Self-consistency	Web-based	Linguistic information	Recommendation	Dictionary	Korean Support
WordFreak **[12]**	M/L/W	Java	O	X	span	O	X	O
GATE **[7]**	M/L/W	Java	O	O	span/relation	O	O	O
Atomic **[9]**	M/L/W	Java	O	X	span/relation/struct	X	X	O
WebAnno **[10]**	M/L/W	Java	X	O	span/relation	O	X	O
Anafora **[5]**	M/L	Python2	X	O	span/relation	X	X	△
BRAT **[18]**	M/L	Python2	X	O	span/relation	O	X	△
YEDDA **[8]**	M/L/W	Python2	O	X	span	O	X	△
PPEditor **[11]**	W	C#	X	X	span/relation/struct	O	X	O
LiAS	M/L/W	Java/Python	O	△	span	O	O	O

% OS: M − Mac OS; L − Linux; W - Windows

사전 정보를 추출하여 이 정보를 부착도구에 제공함으로 작업의 효율을 크게 향상시킬 수 있다. 한국어는 경우 길이가 3 이상의 개체명은 정규표현으로 대부분 정확하게 인식할 수 있다. 따라서 지명이나 기관명 경우에 따라서는 인명의 경우도 외부의 언어 자원으로부터 사전을 제공할 수 있다. LiAS는 외부로부터 사전 정보를 제공할 뿐 아니라 작업을 진행하면서도 사전을 지속적으로 확장할 수 있다.

LiAS는 한국어에 특화되어 있지만 영어, 일본어, 중국어 등 모든 언어를 지원한다. 현재 실험을 한국어를 중심으로 검증되었다. 향후에 일본어와 중국어 개체명 인식을 위한 학습 자료를 구축할 계획이다.

4. 결론 및 향후 계획

본 논문에서는 대량의 학습 자료를 구축하기 위한 부착 시스템을 개발했다. 이 시스템은 자동 부착, 오류 수정, 확률 검수, 사전 확장 및 점진 학습 등의 과정을 반복하면서 학습 자료를 확장한다. 자동 부착은 사전과 기계학습 시스템을 이용해서 원문에 개체명, 구묶음, 전문용어 등과 같은 언어 정보를 부착한다. 오류 수정은 자동 부착된 언어 정보의 오류를 수정한다. 확률 검수는 오류 수정된 언어 정보로부터 확률적으로 일부의 문장을 추출하여 검수한다. 사전확장 및 점진 학습은 검수된 문서를 학습 자료에 포함하여 사전을 확장하고 기계학습 시스템의 성능을 확장한다. 이 시스템은 서버-클라이언트 모델로 구현되어 여러 사람이 협업을 통해서 더욱 효과적으로 언어 정보를 부착할 수 있었으며 많은 시간과 노력을 절약할 수 있었다. 결과적으로 대량의 학습 자료를 효과적으로 구축하는 데 크게 기여하였다.

향후에는 언어 구조나 관계에 해당하는 정보를 부착할 수 있도록 시스템을 확장할 계획이다. 또한 웹 기반 사용자 인터페이스를 개발하여 브라우저가 구동되는 어떤 컴퓨터에서도 쉽게 작업할 수 있도록 할 계획이다. 또한 형태소 분석과 같이 음절이 분리되거나 변환되는 경우도 가능할 수 있도록 기능을 확장할 계획이다. 또한 일본어와 중국어의 개체명 학습 말뭉치 구축에도 적용할 계획이다.

후 기

이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(R7119-16-1001, 지식증강형 실시간 동시통역 원천기술 개발)과 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017M3C4A7068187)

References

[1] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, “A survey on deep learning for big data,” Information Fusion, vol. 42, pp. 146-157, 2018.

[2] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, “Building a large annotated corpus of English: the penn treebank,” Computational Linguistics, vol. 19, no. 2, pp. 313-330, 1993.

[3] A. Saif, N. Omar, U. Z. Zainodin, and M. J. A. Aziz, “Building sense tagged corpus using Wikipedia for supervised word sense disambiguation,” Procedia Computer Science, vol. 123, pp. 403-412, 2018.

[4] B. Plank, D. Hovy, and A. Søgaard, “Linguistically debatable or just plain wrong?,” Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 507-511, 2014.

[5] W. T. Chen and W. Styler, “Anafora: A Web-based general purpose annotation tool,” Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pp. 14-19, 2013.

[6] M. G. Choi, H. W. Seo, H. S. Kwon, J. H. Kim, “Detecting and correcting errors in Korean POS-tagged corpora,” Journal of the Korean Society of Marine Engineering, vol. 37, no. 1, pp. 227-235, 2013 (in Korean).

[7] H. Cunningham, V. Tablan, A. Roberts, and K.

Bontcheva, “Getting more out of biomedical documents with GATE's full lifecycle open source text analytics,” PLoS Computional Biology, vol. 9, no. 2, pp. 1-16, 2013.

[8] J. Yang, Y. Zhang, L. Li, and X. Li, “YEDDA: A lightweight collaborative text span annotation tool,” CoRR, vol. 1711.03759, 2018.

[9] S. Druskat, L. Bierkandt, V. Gast, C. Rzymski, and F. Zipser, “Atomic: An open-source software platform for multi-level corpus annotation,” Proceedings of the 12th Konferenz zur Verarbeitung natürlicher Sprache pp. 228-234, 2014.

[10] R. Eckart de Castilho, É. Mújdricza-Maydt, S. M. Yimam, S. Hartmann, I. Gurevych, A. Frank, and C. Biemann, “A Web-based tool for the integrated annotation of semantic and syntactic structures,” Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities, pp. 76-84, 2016.

[11] J. H. Kim and E. J. Park, “PPEditor: Semi-automatic annotation tool for Korean dependency structure,” The Transaction of the Korean Information Processing Society, vol. 13-B, no. 1, pp. 63-70, 2006 (in Korean).

[12] T. Morton and J. LaCivita, “WordFreak: an open tool for linguistic annotation,” Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pp. 17-18, 2003.

[13] L. A. Ramshaw and M. P. Marcus, “Text chunking using transformation-based learning,” S. Armstrong, K. Church, P. Isabelle, S. Manzi, E. Tzoukermann, D. Yarowsky (eds), Natural Language Processing Using Very Large Corpora, Text, Speech and Language Technology, pp. 157-176, Springer, 1999.

[14] J. H. Kim, H. C. Kim, and Y. S. Choi, “Feature generation of dictionary for named-entity recognition based on machine learning,” Journal of Information Management, vol 41. no. 2, pp. 31-46, 2010 (in Korean).

[15] J. Lafferty, A. McCallum, and F. CN Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” Proceedings of the 18th International Conference on Machine Learning, pp. 282-289, 2001.

[16] G. W. Ko and S. P. Choi, “Research on self-enhancing text annotation systems for information extraction based on machine learning,” Proceedings of the Korean Computer Congress, pp. 616-618, 2018 (in Korean).

[17] K. M. Noh, C. H. Kim, M. A. Cheon, H. M. Park, H. Yoon, J. K. Kim, and J. H. Kim, “A semi-automatic annotation tool based on named entity dictionary,” Proceedings of the 29th Annual Conference on Human and Cognitive Language Technology, pp. 309-313, 2017 (in Korean).

[18] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii, “BRAT: A Web-based tool for NLP-assisted text annotation,” Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 102-107, 2012.