Patent Tokenizer: 형태소와 SentencePiece를 활용한 특허문장 토크나이즈 최적화 연구

박진우○, 민재옥, 심우철, 노한성
한국특허정보원, R&D센터
{znu808, okauto, sim0915, neodream}@kipi.or.kr

Patent Tokenizer: a research on the optimization of tokenize for the Patent sentence using the Morphemes and SentencePiece

Jinwoo Park○, Jae-Ok Min, Woo-Chul Sim, Han-Sung Noh
Korea Institute of Patent Information, R&D Center

요 약

토큰화(Tokenization)는 사람이 작성한 자연어 문장을 기계가 잘 이해할 수 있도록 최소 단위인 토큰으로 분리하는 작업을 말하여, 이러한 토큰화는 자연어처리 전반적인 태스크들의 전처리에 필수적으로 사용되고 있다. 최근 자연어처리 분야에서 높은 성능을 보이며, 다양한 딥러닝 모델에 많이 활용되고 있는 SentencePiece 토큰화는 여러 단어에서 공통적으로 출현하는 부분단어들을 기준으로, BPE 알고리즘을 이용하여 문장을 압축 표현하는 토큰화 방법이다.

본 논문에서는 한국어 기반 특허 문헌의 초록 자연어 데이터를 기반으로 SentencePiece를 비롯한 여러 토큰화 방법에 대하여 소개하며, 해당 방법을 응용한 기계번역 (Neural Machine Translation) 태스크를 수행하고, 토큰화 방법별 비교 평가를 통해 특허 분야 자연어 데이터에 최적화된 토큰화 방법을 제안한다. 그리고 본 논문에서 제안한 방법을 사용하여 특허 초록 한-영 기계번역 태스크에서 성능이 향상됨을 보였다.

주제어: 한국어, 자연어처리, 특허, 토큰화, 형태소, SentencePiece, 기계번역, NMT

1. 서론

오늘날 인공지능 기반 기술들이 발전하면서 검색, 챗봇, 지능형 비서 등 자연어 데이터를 기반으로 한 여러 인공지능 서비스들이 출현함에 따라, 기계학습 관점에서 자연어 데이터의 전처리에 대한 중요성이 증대되고 관련 연구가 활발히 진행되고 있다. 특히 4차 산업혁명과 맞물려 기술발전이 가속화됨에 따라 지식재산권의 중요성이 높아지고 있고, 그 중 많은 비중을 차지하고 있는 특허는 기업 및 국가연구기관의 투자방향 설정 및 과학기술정책수립, 기술 분쟁 방지 등에 핵심적인 역할을 하고 있으며, 매년 출원 건수가 증가하고 있는 추세이다. 또한, 특허문서는 출원된 발명의 내용을 제3자가 명세서만으로 쉽게 알 수 있도록 공개하여 누구나 접근이 가능하며, 특허권으로 보호받고자 하는 기술적 내용과 범위를 명확하게 자연어로 기술한 문서로, 자연어 데이터로서의 활용도가 매우 높다.

이러한 자연어 데이터를 이용한 기계학습 모델을 구현하기 위해서는 기계가 이해하기 용이하도록 정제하고 변환하여 모델의 입력으로 전달하는 전처리가 매우 중요하다. 특히 전처리 과정 중 문장의 단어들을 유기적인 관계를 포함하여 n-차원의 실수 벡터로 표현하는 방법을 단어 임베딩(Word Embedding)[1]이라 하며, 이러한 단어 임베딩을 위하여 문장을 적절한 단위로 분리하는 것을 토큰화(Tokenization)라 한다.

문장을 토큰화하는 방법에 대해서는 영어권 나라를 필두로 지금도 다양한 연구가 꾸준히 진행되고 있지만, 그 기술을 한국어에 대해 그대로 적용하기에는 여러 문제점들이 발생한다. 특히 영어의 경우 고립어의 특성을 가지고 있기 때문에 띄어쓰기 기준의 어절단위로 분리하는 경우 단어의 의미가 그대로 반영되지만, 한국어의 경우 교착어의 특성을 가지고 있어 띄어쓰기 기준으로 문장을 분리하는 경우 토큰에 접사가 포함되어 단어의 의미가 정확히 반영되지 못하는 문제점이 발생한다. 이러한 문제점을 해결하기 위해서는 한국어의 의미를 가지는 최소 단위인 형태소를 기준으로 분리하여 토큰화 하는 것이 적합한 것으로 알려져 있다.[2]

하지만, 형태소만을 고려하여 토크나이저(Tokenizer)를 구성한다면, 한국어 같은 형태소의 활용이 다양한 언어의 경우 학습데이터 내의 단어나 형태소의 수가 많아 토크나이저의 사전의 크기가 커져 모델의 복잡도가 증가하고, 사전에 등록되어 있지 않은 미등록 어휘에 대해서는 토큰화가 잘 되지 않는 문제가 발생한다.[3]

또한, 특허문서의 자연어 데이터는 일반적인 문장의 구성과는 다른 문체적, 문법적 특징이 존재하고, 일반적으로 잘 쓰이지 않는 전문분야에 속하는 기술을 설명하는 전문용어가 많으며, 특허라는 전문 기술을 다루는 문서의 특징을 반영하기 위하여 두 개 이상의 형태소가 결합된 합성어나 어근에 접두사 또는 접미사가 붙어 이루어진 파생어가 많기 때문에[4] 위에서 언급한 문제들이 더욱 크게 발생한다.

이러한 문제점들을 보완하기 위해서 문장을 1차로 토

큰화하여 분리한 후 각 단어의 공통적인 부분들을 기준으로 BPE(Byte Pair Encoding)[5] 알고리즘을 적용하여 더 작은 단위인 부분단어(Subword)로 2차로 분리하는 부분단어 토큰화(Subword Tokenization) 방법이 유효하다고 알려져 있다.[6] 최근 자연어처리 분야에서 각광 받고 있는 언어모델인 BERT(Bidirectional Encoder Representations from Transformers)[7]나 GPT(Generative Pre-Training)[8]에서도 이러한 토큰화 방법을 사용하여, 널리 알려진 자연어처리 공개 태스크인 GLUE Benchmark[9]에서 기록을 갱신하며 높은 성능을 입증하였다.

본 논문에서는 성능이 우수하다고 알려진 공개용 한국어 형태소 분석기 중 하나인 Mecab-ko[10]와 BERT 모델에서도 활용된 Google에서 공개한 부분단어 토크나이저 중 하나인 SentencePiece[11]를 이용하여 특허문서 초록 데이터를 학습하고, Mecab-ko 사용자 사전 및 SentencePiece Vocabulary를 구축하였다. 그리고 특허문서 초록 한-영 번역 데이터 셋을 대상으로 기계번역 태스크를 수행하여 BLEU(Bilingual Evaluation Understudy)[12] 평가를 진행함으로써 다양한 토큰화 처리 방식별 성능 비교로 특허분야 자연어 데이터에 적합한 토큰화 방법을 제안하고자 한다.

2. 관련 연구

2.1 한국어 형태소 분석기 Mecab-ko

영어 문장의 경우 띄어쓰기를 기준으로 한 어절이 의미를 가지는 최소단위가 되지만, 한국어 문장의 경우에는 어절이 어간과 접사로 이루어진 교착어의 형태를 이루고 있다. 그래서 품사 분석을 통해 의미를 가지는 최소단위인 형태소 기준으로 문장을 분리하기 위해서는 한국어 형태소 분석기 기반 토크나이저가 필수적이다.

특히 특허문서의 경우 기술문서의 특성상 전문용어의 출현빈도가 높고, 발명된 기술을 보호하기 위해 기술된 자연어 데이터들이 많은 대용량 코퍼스 중 하나이다.

한국어 위키백과를 이용한 형태소 분석기별 성능 비교[13]와 특허 문서 대상의 형태소 분석기별 성능 비교평가 결과[14]에 따르면, Mecab-ko가 그림 1과 같이 특허문서의 태깅 처리속도가 가장 빠르고, 품사별 유사도 및 유추 테스트 결과를 종합하였을 때 가장 성능이 우수한 것으로 알려져 있다. 또한, Mecab-ko는 특허문서에서 출현하는 신규 전문용어의 사용자사전 추가가 용이하기 때문에, 본 논문에서는 Mecab-ko를 문장을 1차적으로 분리하는 Pre-tokenizer로 사용하였다.

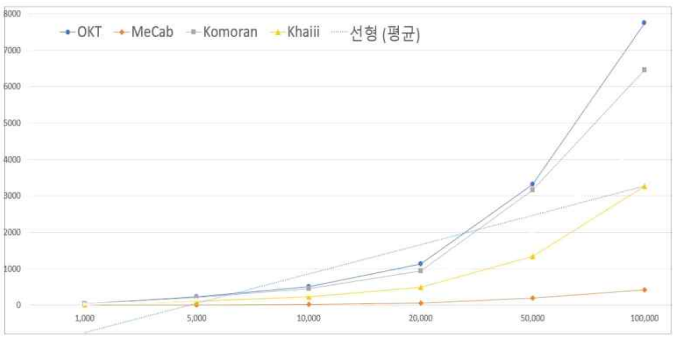

그림 1. 특허 건수에 따른 품사 태깅 시간 [14]

2.2 SentencePiece 토크나이저

형태소 분석기만을 이용하여 토크나이저를 구성하는 경우 학습데이터가 커질수록 토크나이저의 사전 크기가 커져 모델의 복잡도를 증가시키고, 사전에 미등록 어휘가 발생하는 경우 토큰화가 정상적으로 수행되지 않는 문제가 발생한다. 이런 문제들을 해결하기 위한 한국어의 특성을 반영한 벡터 생성에 대한 연구들이 많이 이루어져 왔으며, 최근에는 문장을 형태소와 자소 또는 음절단위의 부분단어 토큰화를 이용하여 벡터를 생성함으로써 기존 어절단위의 토큰화 방법대비 효과적인 한국어 벡터 표현방식이 제안되고 있다.[15]

그리고 위에서 언급된 문제는 일반적인 문장과는 다른 문체적, 문법적 특징이 존재하여 데이터의 복잡도가 높고, 전문기술용어를 비롯한 다양한 어휘가 존재하는 특허 자연어 데이터에서 더욱 두드러진다.

본 논문에서는 이러한 문제들을 보완하기 위하여, 문장을 한국어 형태소 분석기인 Mecab-ko를 통해 1차적으로 분리하고, BPE 기반의 토크나이저인 SentencePiece를 Post-Tokenizer로 사용하여 2차적으로 분리하였다. 이는 결국 다양한 토큰들을 압축함으로써 토크나이저의 사전의 크기를 줄여 모델의 복잡도를 감소시키고, 부분단어들의 결합 및 발생빈도를 중요지표로 하여 사전을 생성함으로써 미등록 어휘에 대한 문제점을 해결하였다.

본 논문에서는 특허문서 초록 데이터를 기반으로 학습된 Mecab-ko와 SentencePiece를 결합하여 만든 토크나이저를 MSP 토크나이저(Mecab Sentencepiece Patent Tokenizer)로 명명하여 사용하였으며, MSP 토크나이저 구조 및 예시는 그림 2와 같다.

그림 2. MSP 토크나이저 구조 및 예시

2.3 기계번역 모델

이종 언어 간의 번역을 수행하는 기계번역 태스크는 연구가 활발히 진행되고 있는 널리 알려진 자연어처리 태스크 중 하나이며, Google에서 공개한 SentencPiece Experiments[16]와 같이 토크나이저의 성능을 평가하는 주요 태스크로 사용되고 있다. 본 논문에서는, 이러한 기계번역 태스크의 번역품질을 평가하기 위해 ngram 방식으로 언어에 구애받지 않고 사용가능 하고, 속도가 빠른 BLEU Score와 모델의 구조가 단순하고, 기계번역 태스크에서 일반적으로 사용하는 LSTM (Long Short-Term Memory models) 기반의 Seq2Seq (Sequence To Sequence) 딥러닝 모델[17]을 Keras 프레임워크로 구현하였으며, 구현된 모델의 구조는 그림3과 같다.

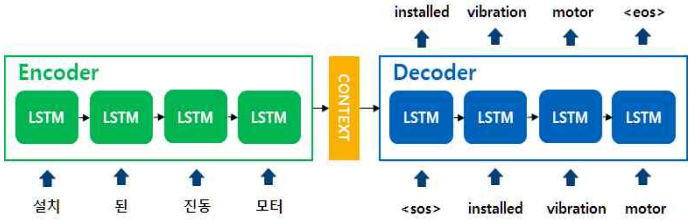

그림 3. Seq2Seq 모델 구조

2.4 한국어 명사, 복합명사 추출기

한국어 특허문장 데이터의 경우 일반적인 한국어 문장과는 달리 “엔도펩티다제”와 같은 복잡한 전문용어 및 “티비”, “테레비”, “텔레비전”과 같은 동일한 의미를 가지지만 발음이 다르게 표현된 어휘가 많아, 일반적인 단어를 학습한 Mecab-ko로는 특허 문장을 제대로 분리하기가 어렵다.

본 논문에서는 특허분야 전문용어 추출을 위해 어절을 명사와 조사의 결합인 Left-Right 그래프 방식의 통계적 분석을 통해 분리하여 추출하는 오픈소스 기반의 Soynlp 명사추출기[18]를 사용하였으며, 학습데이터 내 명사와 복합명사를 추출 후 정제과정을 거쳐 Mecab-ko의 사용자 사전에 추가하여 실험을 진행하였다.

3. 데이터 셋

3.1 특허 초록 한-영 번역 문장쌍

본 논문에서는 2011년도부터 2013년도까지의 전체 특허문서 초록의 한글-영어 번역 문장쌍 데이터 중 원활한 실험을 위해 한글은 100자 미만, 영어는 300자 미만으로 제한하여 총 문장쌍 학습데이터 243,201건을 구축하였다. 이 중 기계번역 추론 및 BLEU 평가를 위한 데이터 Test 2000건을 제외한 241,202 건을 9:1의 비율로 분할하여 Train 217,080건을 사용하였고 모델의 검증을 위해 Dev 24,121건을 사용하였다. 실험에 사용된 데이터 셋의 통계와 예시는 아래의 표1 및 표2와 같다.

표 1. 데이터 셋 통계

Train	Dev	Test	Total
217,080	24,121	2,000	243,201

표 2. 한-영 번역 문장쌍 예시

Korean	English
태양전지모듈용 리본은 버스 리본과 인터커넥션 리본 연결부위의 저항을 감소시킴으로써, 광변환 효율을 향상시킬 수 있다.	A ribbon for a solar cell module is provided to improve phosphor conversion efficiency by reducing the resistance of a connection part of a bus ribbon and an interconnection ribbon.

4. 기계번역 태스크 평가

4.1 실험 공통

공정한 성능 비교를 위해 토큰화를 통해 분리된 토큰의 인덱스를 별도의 임베딩 알고리즘을 사용하지 않고 Seq2Seq모델의 Encoder 입력으로 사용하였고, Decoder 부분에서 사용하는 영어의 경우 모든 실험에서 SentencePiece 토크나이저 및 공통된 사전을 사용하였다. 또한, 기계번역 태스크에서 사용된 딥러닝 모델의 하이퍼 파라미터 및 실험 환경은 모두 동일하게 사용하였으며 해당 값은 아래의 표3 및 표4와 같다.

표 3. 실험 공통 환경

Environments	Version
Operating system	Ubuntu 16.04
GPU	NVIDIA P100 12G 2EA
Python	3.7
Keras	2.3.1
Tensorflow	1.15.0
Mecab-ko	mecab-0.996-ko-0.9.2
SentencePiece	0.1.91

표 4. 실험 공통 파라미터

Parameters	Value
Eng tokenizer	SentencePiece
Eng tokenizer type	BPE
Eng vocab size	8000
Optimizer	rmsprop
Learning rate	0.001
Latent dimension	256
Batch size	128
Epochs	10

4.2, 4.3, 4.4의 실험은 동일한 프로세스를 반복적으로 수행하여 실험하였으며, 4.5는 명사와 복합명사를 추출하여 Mecab-ko의 사용자 사전에 등록하는 프로세스가 추가되어 그림4와 같이 수행되었다.

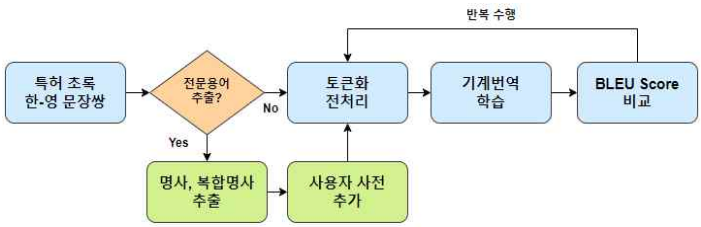

그림 4. 전체 실험 프로세스

4.2 SentencePiece 토큰 타입별 비교

SentencePiece에서 제공하는 토큰의 타입은 Char, Unigram, BPE 가 있으며 각 타입 별로 데이터를 학습하여 사전을 생성하고, 이를 이용하여 기계번역 태스크를 수행한 결과는 표5와 같다. 이를 통하여 세 가지의 토큰 타입 중 BPE 타입으로 생성된 사전을 사용하는 경우 평균토큰개수 29.29, BLEU Score 50.94로 가장 높은 성능을 보였다.

표 5. SentencePiece 토큰 타입별 성능 비교

Type	Vocab size	Average token num	BLEU score
Char	1,050	76.15	47.83
Unigram	8,000	30.66	50.52
BPE	8,000	29.29	50.94

4.3 SentencePiece Vocabulary 크기별 비교

SentencePiece에서 문장을 분리하는 기준이 되는 토큰들의 사전을 Vocabulary라 하며, 학습 코퍼스내의 단어 출현빈도가 Vocabulary의 크기를 제한하는 중요 지표로 사용된다. 그러므로 Vocabulary의 크기가 작을수록 고빈도의 짧은 단어가 포함되어, 문장이 분리된 토큰의 개수가 많이 나올 가능성이 높고, 이와 반대로 크기가 커질수록 저빈도의 긴 단어가 포함되어 토큰의 개수가 상대적으로 적게 나올 수 있다. 이러한 관점에서 특허문서의 평균문장길이 대비 최적의 토큰개수를 도출하여 Vocabulary의 크기를 결정하기 위해서 아래의 표6과 같은 실험을 진행하였다.

Vocabulary 크기를 8000, 16000, 32000 으로 나누어 성능 비교 실험을 해본 결과, Vocabulary 크기가 8000의 경우 평균토큰개수 29.29, BLEU Score 50.94 로 가장 좋은 성능을 보이는 것으로 확인되었다.

표 6. SentencePiece Vocabulary 크기별 성능 비교

Type	Vocab size	Average token num	BLEU score
BPE	8,000	29.29	50.94
	16,000	26.32	49.98
	32,000	24.23	49.90

4.4 MSP 토크나이저 적용 비교

형태소분석기 Mecab-ko를 Pre-tokenizer로, 그리고 형태소분석기 기반 토크나이저의 단점을 보완하기 위해 SentencePiece를 Post-tokenizer로 결합한 MSP 토크나이저를 적용하여 Vocabulary 크기별로 기계번역 태스크를 수행하였다. 또한, 형태소 분석기인 Mecab-ko 만을 사용하여 토큰화하는 경우의 성능 비교를 위하여, 학습 데이터 셋 기반 48,503 건의 사전을 구축하고 기계번역 태스크도 추가적으로 수행하였다.

그 결과 표7과 같이 형태소 단위로 토큰화를 수행한 Mecab-ko, MSP 토크나이저 모델들이 SentencePiece 만을 사용한 모델보다 전반적으로 성능이 우수한 것을 확인할 수 있었다. 그리고 그림5와 같이 SentencePiece와 MSP의 BLEU Score 결과의 분포를 비교하여 보면, 50점 이상의 점수가 나온 것은 SentencePiece의 경우 850개로 테스트 데이터의 42.5%, MSP의 경우 1153개, 57.65%로 전반적으로 번역품질이 더 우수한 것을 확인할 수 있다.

특히 MSP 토크나이저의 경우 Vocabulary 크기 16000, 평균토큰개수 36.46, BLEU Score 51.27로 성능이 가장 우수한 것으로 확인되었다. Mecab-ko만을 사용한 모델도 BLEU Score 51.03으로 나쁘지 않은 성능을 보였으나, MSP 토크나이저에 비해 Vocabulary의 크기가 2배 이상인 것은 결국 학습 데이터 셋의 크기가 커질수록 모든 문장을 커버하기 위해 Vocabulary의 사이즈가 커져 딥러닝 모델의 복잡도를 증대시키는 단점을 가지게 된다.

표 7. MSP 토크나이저 적용 성능 비교

Tokenizer	Type	Vocab size	Average token num	BLEU score
SentencePiece	BPE	8,000	29.29	50.94
Mecab-ko	Morphs	48,503	35.87	51.03
MSP	Morphs, BPE	8,000	37.51	50.98
		16,000	36.46	51.27
		32,000	36.05	51.08

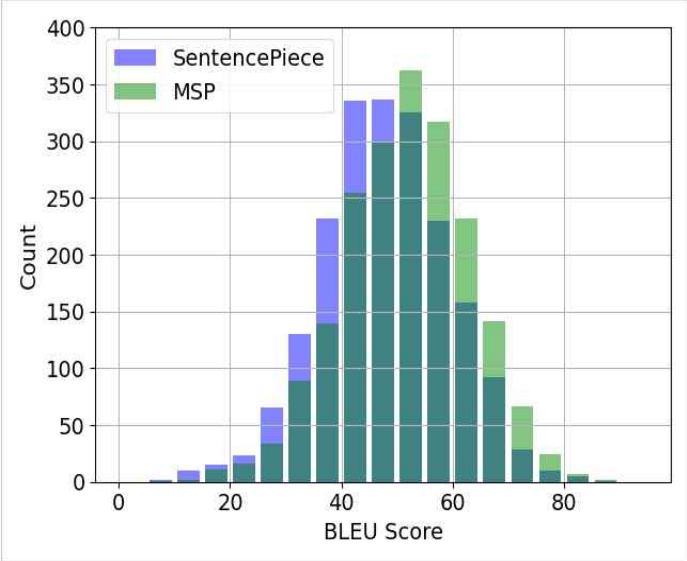

그림 5. SentencePiece, MSP 결과 분포 비교

4.5 Mecab-ko 사용자 사전 추가 실험

본 논문에서 사용된 Mecab-ko의 기본사전에는 특허문서에서 사용되는 “세퍼레이터”와 같은 전문기술용어가 미포함되어, 단어의 원형 그대로 분리되지 않고 “세”, “퍼”, “레이터”로 분리되는 문제가 발생한다. 이를 개선하기 위해 통계기반의 오픈소스 Soynlp 명사 추출기를 사용하여 학습 데이터 셋 내 명사 및 복합명사를 추출하였다. 이후 용어 품질을 위해 글자수 2개 이상 & 숫자, 특수문자를 제외한 순수 한글로만 이루어진 명사 56,727건, 복합명사 47,942건을 Mecab-ko의 사용자 사전에 추가 등록하였다.

그리고 위 사용자사전이 추가된 MSP 토크나이저로 기계번역 태스크를 재 수행하였고, 표8과 같이 이전의 실험과 비교하였다. 그 결과 Vocabulary 크기 8000, 평균토큰개수 37.14, BLEU Score 51.75를 획득하였으며, 기존 사용자사전이 추가되지 않은 MSP 토크나이저 대비 약 0.5 향상된 성능이 이루어졌다. 이는 토크나이저 학습데이터 내 특허 전문기술용어들의 추가로 인하여 토큰화가 잘 수행되어 결국 기계번역기의 성능이 향상되는 결과로 이루어짐을 확인할 수 있었다.

표 8. Mecab-ko 사용자 사전 추가 성능 비교

Tokenizer	Type	Vocab size	Average token num	BLEU score
MSP	Morphs, BPE	16,000	36.46	51.27
MSP with Patent dictionary	Morphs, BPE	8,000	37.14	51.75
		16,000	35.58	51.17
		32,000	34.70	50.57

4.6 결과분석

4.1부터 4.5까지의 실험결과들을 분석해 보았을 때, 특허문서에 존재하는 미등록 전문기술용어들을 추출하여 Mecab-ko 의 사용자 사전에 등록하고, 이를 이용하여 형태소 기반의 Pre-tokenizing 처리 후 SentencePiece BPE를 이용하여 Post-tokenizing 하는 MSP 토크화 방법이 특허 초록 한-영 기계번역 태스크에서 매우 우수한 방법임을 확인할 수 있었으며, 이는 결국 특허 데이터를 이용한 다양한 자연어처리 태스크에서도 유효한 방법임을 확인할 수 있었다.

또한, 위 실험들을 통하여 확인되었듯이 Vocabulary 크기는 토크나이저의 성능에 큰 영향을 주는 요인으로, 이에 따라 문장이 분리되는 토큰의 개수가 달라진다. 본 논문에서 사용된 데이터 셋은 평균적으로 75.15의 문장 길이를 가지고 있으며, MSP with Patent Dictionary 8000으로 평균 37.14개의 토큰으로 분리하였을 때 가장 우수한 성능을 보이고 있다. 만일 데이터 셋이 확장 또는 변경되는 경우에는 위 실험의 결과를 역으로 이용하여, 데이터 셋의 평균문장길이에 따른 최적의 토큰개수를 도출하고, 이를 이용해 Vocabulary를 새로 구축할 수 있다.

5. 결론 및 향후 방향

본 논문에서는 특허분야의 자연어 데이터를 이용한 기계학습에 필수적인 전처리 과정 중 토큰화에 대하여 효과적인 방법을 제안하고, 이를 검증한 실험결과를 공유하였다. 결과적으로 한국어 특허 자연어 데이터 특성에 맞는 형태소와 부분단어 토큰화 방법의 장점을 결합한 하이브리드 형태의 MSP 토크나이저를 제안하였고, 특허 초록 한-영 기계번역 태스크에서 다른 일반적인 방법대비 성능이 우수하다는 사실을 증명하였다. 우리는 본 논문을 통해 연구결과를 공유함으로써 향후 다른 연구자들이 특허 분야의 자연어 전처리를 수행함에 있어서 시행착오를 거치지 않고 효율적으로 문제를 해결하여 관련 분야의 연구 활성화에 기여하고자 한다.

우리는 본 논문의 실험을 통해 최적화한 노하우를 바탕으로 다양한 모습으로 진화하고 있는 특허 데이터를 이용한 언어모델 생성에 대한 연구를 계속하고자 한다. 뿐만 아니라 학습 데이터를 지속적으로 확장 구축하여 공유하고, 특허분야 실세계에 산재하는 문제들에 기반한 다양한 자연어처리 태스크에 대해 연구할 예정이다.

Acknowledgement

본 연구는 2020년도 한국과학기술정보연구원(KISTI) 주요사업 과제로 수행한 것입니다.

참고문헌

[1] Omer Levy and Yoav Goldberg, “Dependency-Based Word Embeddings”, The ACL, pp.302-308, 2014
[2] 조현수, 이상구, “FastText를 적용한 한국어 단어 임베딩”, 한국정보과학회 학술발표논문집, pp.705-707, 2017
[3] 안성만, 정여진, 이재준, 양지헌, “한국어 음소 단위 LSTM 언어모델을 이용한 문장 생성”, 지능정보연구, 23(2), pp.71-88, 2017
[4] 장지현, 진두현, 이숙의, “특허문서의 특징과 언어학적 분석”, 한국언어문학회, pp.85-116, 2018
[5] P. Gage, “A new Algorithm for data compression”, C Users J., vo. 12, no.2, pp.23-38, 1994
[6] Chanjun Park, Chanhee Lee, Yeongwook Yang, Heusiseok Lim, “Ancient Korean Neural Machine Translation”, IEEE Access, vo. 8, pp.116617-116625, 2020
[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding”, arXiv preprint arXiv:1810.04805, 2018
[8] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, “Improving Language Understanding by Generative Pre-Training”, OpenAI, 2018
[9] Wang, Alex, et al. “Glue: A multi-task benchmark and analysis platform for natural language understanding.” arXiv preprint arXiv:1804.07461, 2018
[10] 은전한닢 Mecab-ko, https://bitbucket.org/eunjeon/mecab-ko/src/master/
[11] Taku Kudo, John Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing", arXiv preprint arXiv:1808.06226, 2018
[12] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation", The ACL, pp.311-318, 2002
[13] 강현석, 양장훈, “한국어 Word2vec 모델을 위한 최적의 형태소 분석기 선정”, 한국정보처리학회, pp.376-379, 2018
[14] 이유진, 김세빈, 홍현석, 김장원, "특허 문서를 위한 형태소 분석기 비교 평가", 한국정보기술학회, pp.264-265, 2019
[15] 윤준영, 이재성, “부분단어와 품사 태깅 정보를 활용한 형태소 기반의 한국어 단어 벡터 생성”, 한국정보과학회논문지 47(4), pp.395-403. 2020
[16] SentencePiece Experiment, https://github.com/google/sentencepiece/blob/master/doc/experiments.md
[17] Ilya Sutskever, Oriol Vinyals, Quoc V. Le, "Sequence to Sequence Learning with Neural Networks", arXiv preprint arXiv:1409.3215, 2014
[18] Soynlp Noun Extractor, https://github.com/lovit/soynlp