Rasbian OS에서 STT API를 활용한 형태소 표현에 대한 연구

박진우* · 임재순 · 이성진 · 문상호

부산외국어대학교

Morphology Representation using STT API in Rasbian OS

Park-jin Woo* · Je-Sun Im · Sung-jin Lee · Sang-ho Moon

Busan University of Foreign Studies

E-mail : pjwpjw715@gmail.com

요 약

국어의 경우 교착어이기 때문에 영어와 같이 어절 토큰화를 통하여 태깅할 경우 발전 가능성이 영어보다 낮은 편이다. KoNLPy를 통해 형태소 단위로 분리하여 코퍼스를 토큰화한 형태를 그래프 데이터베이스로 표현이 되지만 해당 모듈을 그래프 데이터베이스에서 코퍼스로 변환 시 음성파일의 완전 분리 및 실용성에 대한 검증이 필요하다.

본 논문에서는 Raspberry Pi에서 STT API를 활용한 형태소 표현을 나타내고 있다. 코퍼스로 변환된 음성 파일을 KoNLPy로 형태소 분석 후 태깅한다. 분석된 결과는 그래프 데이터베이스로 표현되며 형태소별로 나누어진 토큰으로 구분할 수 있음이 확인되었고, 실용성과 분리 정도를 판단하여 특정 목적성을 지닌 데이터 마이닝 추출이 가능한 것으로 판단된다.

ABSTRACT

In the case of Korean, the possibility of development is lower than that of English if tagging is done through the word tokenization like English. Although the form of tokenizing the corpus by separating it into morpheme units via KoNLPy is represented as a graph database, full separation of voice files and verification of practicality is required when converting the module from graph database to corpus.

In this paper, morphology representation using STT API is shown in Raspberry Pi. The voice file converted to Corpus is analyzed to KoNLPy and tagged. The analyzed results are represented by graph databases and can be divided into tokens divided by morpheme, and it is judged that data mining extraction with specific purpose is possible by determining practicality and degree of separation.

키워드

KoNLPy, Raspberry pi, Graph Database Morpheme Analysis

Ⅰ. 서 론

국어의 경우 교착어이기 때문에 영어에서 사용하는 NLTK와 같이 단어 단위로 토큰화 하는 것이 힘들다. 따라서 대부분의 국어의 토큰화를 돕기 위한 KoNLPy(Python package for Korean Natural Language Processing) 형태소 분석기를 사용하고 어절 단위가 아닌 형태소 단위를 제공하고 있다.

해당 모듈에는 jhannanum, kkma, komoran, twitter, mecab 총 5가지 모듈로 나누어져있고 모듈별 형태소 분석 결과는 다 다른 형태로 분리된다. 그리고 분리된 형태소 분석 결과물들로 기분석사전을 활용하여 과분석을 줄여 속도와 정확도를 높일 수 있다[1].

그러나 구두(口頭)를 통해 전달되는 말들은 STT(Speech To Text)작업 혹은 말 자체의 줄임말, 은어, 외래어로 인해 정답률이 낮아지거나 분석이 올바르게 이루어지지 않는 경우가 있다. 따라서 해당 연구에서는 Raspberry Pi에서 STT작업을 통해 받은 텍스트를 KoNLPy의 5가지 모듈들로 분석하여 해당 결과를 그래프데이터 베이스로 표현한다.

표현된 결과는 그래프 데이터베이스의 형태로

* speaker

손쉽게 확인하여 특정 문장의 오류를 쉽게 확인할 수 있도록 한다. 확인된 결과는 기분석 사전을 통해 이용하여 정확한 결과를 도출해내기 위한 사전 작업이 가능하도록 하게 하는 것을 목표로 한다.

II. KoNLPy

NLP(Natural Language Processing, 자연어처리)는 텍스트에서 의미있는 정보를 분석, 추출하고 컴퓨터에 이해시키는 일련의 기술이다. KoNLPy는 파이썬 패키지로 이용되며 대표적으로 이용되는 5가지 클래스가 존재한다. Hannanum, Kkma, Komoran, Mecab, Twitter로 이루어져있고 5가지 모두 실행시간과 결과에 차이가 있다. 또한 문장의 길이에 따라 실행 시간이 기하급수적으로 늘어나며 용도에 따라 다르게 사용될 수 있다[2].

5가지 클래스는 무분석 방법으로 원형 복구 전 표층형의 전체 혹은 부분 어절을 사전에 미리 등록한다[3]. 형태소 분석시에는 표층형으로 검색하여 분석된 내용을 조합한다. 해당 방식은 등록되어 있는 5클래스 별로 다르게 등록이 되어 있다. 등록이 되어 있지 않은 경우는 어절 그대로 등록 혹은 기분석사전을 참조한다. 문제가 되는 부분은 직접 등록하거나 기분석사전을 사용하여 오류를 수정해 나가야 한다.

또한 말로 전달되는 경우에는 화자의 의도와 다르게 띄어쓰기, 외래어, 은어와 같은 상황에 따라 정규표현식의 영향을 받으며 STT가 분석을 제대로 하지 못할 경우에는 의미 자체가 달라질 수도 있다.

따라서 목적성을 지닌 소통 목적을 지닌 AI스피커 방식에서 형태소를 의도에 맞게 분석하기 위해서는 사전 단어 등록이 필수적이다. 해당 작업을 위해서 본 연구에서는 그래프 데이터베이스를 활용하여 클래스의 특성을 쉽게 확인할 수 있고 수정하고자 하는 데이터를 쉽게 확인할 수 있게 태깅한 결과를 표현한다.

III. Raspberry PI 환경에서 구현한 그래프 데이터베이스

표 1. 실험에 사용된 환경

실험 환경	
OS	Raspbian GNU/Linux 9 (stretch)
CPU	ARM Cortex-A53 MP4
RAM	1GB LPDDR2
Language	Python 3.7.0
Package	openJDK-8.0

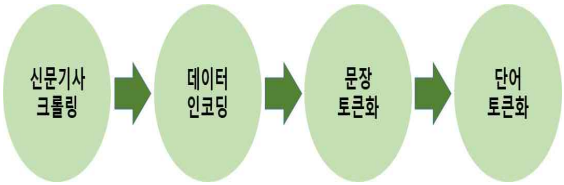

그림 1. 실험 과정

본 논문에서는 위의 표1의 상황에서 진행되고, 5가지 클래스별 실행 결과를 neo4j라는 그래프 데이터베이스로 구현한다.

5가지 클래스별 언어에 사용될 데이터는 인터넷 신문기사 크롤링을 통해 얻은 자료들을 5가지 클래스별로 형태소 분석을 실시한다. 실시하여 얻은 데이터들은 neo4j에 올라가기 위한 데이터 자료로 변환하여 텍스트 형식으로 저장한다.

neo4j는 그래프 데이터 모델로 사용자가 속성과 레이블이 있는 관계로 연결된 그래프로 노드간 관계를 탐색하면서 사용자에게 필요한 데이터를 조회할 수 있다[4]. 해당 그래프 데이터베이스 독립형 모드보다 임베디드 환경에서 실행될 때 쿼리의 성능이 더 뛰어나다. 해당 모델은 구조를 다양하게 변경할 수 있기 때문에 KoNLPy에서 잘못되어진 모델을 확인할 경우에도 쉽게 수정이 가능하다[5].

Raspberry pi에서는 해당 neo4j를 위한 open-jdk-8을 설치하고 nodejs를 통해 서버를 구축한다. 구축이 완료된 neo4j는 형태소 분석이 완료된 데이터들을 받아 클래스별로 언어가 올라갈 것이고 본 연구에서는 Kkma를 활용하여 데이터를 올린다.

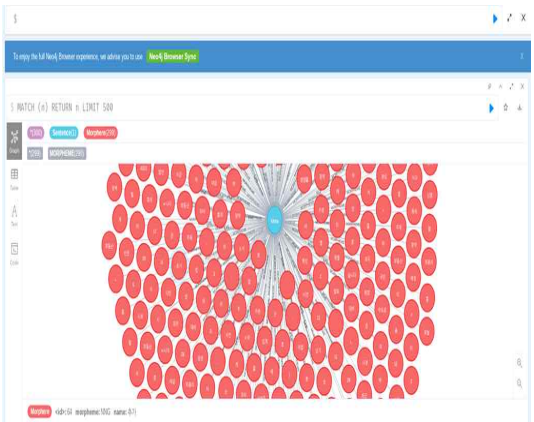

그림 2. Neo4j 실행결과 사진

올라간 데이터들은 위의 사진과 같이 표현된다. 정확한 분석이 되어 형태소별 태깅이 된 것을 확인할 수 있었고 원 형태의 그래프 데이터베이스가 구성되어 진 것을 확인할 수 있다.

Ⅳ. 결 론

본 논문에서는 KoNLPy의 클래스 중에 Kkma로 분석된 데이터가 neo4j로 표현된 것을 확인할 수 있다. 해당 노드간 분석을 통해 어떻게 언어가 분석이 되었는지 그래프로 확인할 수 있다.

그래프 데이터 모델링이 된 데이터는 주로 신문기사 용어로 이루어졌지만 목적성에 따라 일상, 법률 등 여러 부분을 토큰화하여 직접 KoNLPy에 필요한 내용을 등록하거나 기분석사전에 추가할 수 있을 것으로 사료된다.

그래프 데이터베이스를 통한 방식이 수정에 효과적일지에 대한 연구는 추후 세밀한 분석이 필요한 것으로 보이나 Kkma로 분석된 결과는 쉽게 알아 볼 수 있었다. 추후 지속된 연구와 다양한 평가기준을 적용하여 들어온 음성이나 채팅으로 들어온 데이터들을 사전에 통계적 분석을 통해 사전 등록한 후에 목적성에 부합하는지에 대한 결과를 확인하는데 효율적일 것으로 분석된다.

References

[1] Sujeong Kwak, Bogyum Kim, Jae Sung Lee, “Construction of an Efficent Pre-analyzed Dictionary for Korean Morphological Analysis”, in KIPS, Vol. 2, No. 12, pp. 881-888, October. 2013.

[2] KoNLPY. NLP란 무엇인가요? [Internet]. Available : https://konlpy.org/ko/latest/start/

[3] Joon-Choul Shin, Cheol-Young Ock, “A Korean Morphological using a Pre-analyzed Partial Word-phrase Dictionary”, in KIISE, Vol. 5, No. 35, pp. 415-424, May, 2012.

[4] Je-Sun Im, Moon-Kyung Lee, Pum-Mo Ryu, Chulsu Lim, “Implementation of Frame-based Interface for Graph Database.” in KIISE, pp 1589-1591, June, 2021.

[5] Bae Suk Min, Kim Jin Hyung, Yoo Jae Min, Yang Seong Ryul, Jung Jai Jin, “Structural Analysis and Performance Test of Graph Databases using Relational Data”, in Journal of Korea Multimedia Societ, Vol. 22, No. 9, pp. 1036-1045, September, 2019.