SHINEWARE

KOMORAN 2.0

한국어 형태소 분석기

Junsoo Shin

2014-11-24

목차

1. KOMORAN 2.0 소개 3
1.1 개발 배경 3
1.2 기본 알고리즘 3
1.2.1 형태소 분석(Parsing) 3
1.2.2 품사 태깅(POS Tagging) 5
1.3 특징 5
1.3.1 Pure Java 5
1.3.2 외부 라이브러리 독립성 5
1.3.3 경량화 6
1.3.4 쉬운 사용법 6
1.3.5 사전 관리 6
1.3.6 새로운 분석 결과 6
1.4 성능 7
2. 사용방법 9
2.1 사용환경 9
2.2 다운로드 9
2.3 실행 9
2.3.1 Quick start 9
2.3.2 주요 메소드 10
2.3.3 기분석 사전 11
2.3.4 사용자 사전 12
3. 라이선스 13
4. 맺음말 14

1. KOMORAN 2.0 소개

1.1 개발 배경

형태소 분석기란 주어진 텍스트로부터 의미를 갖는 최소 단위인 형태소를 추출하는 것을 말한다. 자연어 처리(Natural Language Processing)의 가장 기본적인 단계이면서 동시에 가장 중요한 단계이다. 외국에서는 이미 높은 성능의 다양한 품사 부착기(Part-of-speech Tagger)들이 연구되어 상용화되었다. 우리나라도 형태소 분석기와 관련된 연구가 활발히 진행되었으나 아직 상용화 단계에서 성공한 형태소 분석기들은 많지 않다. 공개된 대부분의 형태소 분석기들은 97% 이상의 우수한 성능을 보인다고 한다. 그러나 이러한 형태소 분석기들을 실제 필드에서 사용해 보면 97%라는 성능에 대해서 의문이 들 수 밖에 없었다. 그래서 성능이 높은 형태소 분석기보다는 실제 필드에서 쓸 수 있는 유용한 형태소 분석기를 만들어보자는 취지에서 KOMORAN을 개발하게 되었다.

1.2 기본 알고리즘

KOMORAN 2.0은 입력 문장을 구성하고 있는 모든 형태소를 찾는 형태소 분석(Parsing) 부분과 분석된 형태소를 조합하여 최적의 품사열을 찾는 품사 태깅(POS tagging) 부분으로 구성되어 있다.

1.2.1 형태소 분석(Parsing)

입력 문장 내에 포함된 형태소를 찾기 위해서 그림 1과 같이 자소 단위로 구성된 TRIE 기반의 형태소 사전을 이용한다. TRIE를 이용하여 사전을 구성하게 되면 사전 탐색 시 재탐색 횟수를 줄이고 저장 공간을 효율적으로 사용할 수 있다는 장점이 있다.

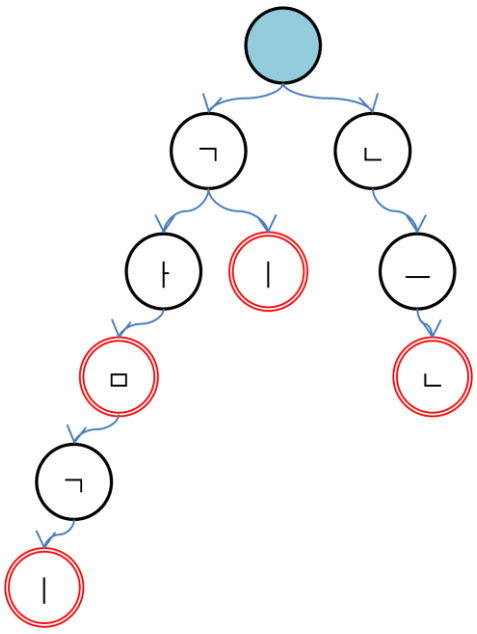

그림 1. 자소 단위 TRIE 구조

그림 1은 "감", "감기", "기", "는"의 형태소를 포함하고 있으며 붉은색 노드는 명사, 동사, 형용사 등과 같은 형태소에 따른 품사 정보를 포함하고 있다. 이렇게 구성된 TRIE 사전을 이용하여 입력된 문장으로부터 형태소를 찾게 된다. 입력 문장으로 "감기는"이 주어지면 먼저 자소 단위 분리를 통해서 "ㄱㅏㅁㄱㅣㄴㅡㄴ"과 같이 변환한다. 변환된 자소로부터 TRIE 사전을 탐색하면 "감", "감기", "기", "는"의 순서로 형태소를 찾을 수 있다. 검색된 형태소들은 Lattice 형태의 자료구조에 저장된다. 이 때 미리 계산되어 저장된 관측 확률(Observation)을 함께 저장한다. 그림 2는 이와 같은 형태소 분석 단계의 순서도를 나타낸다.

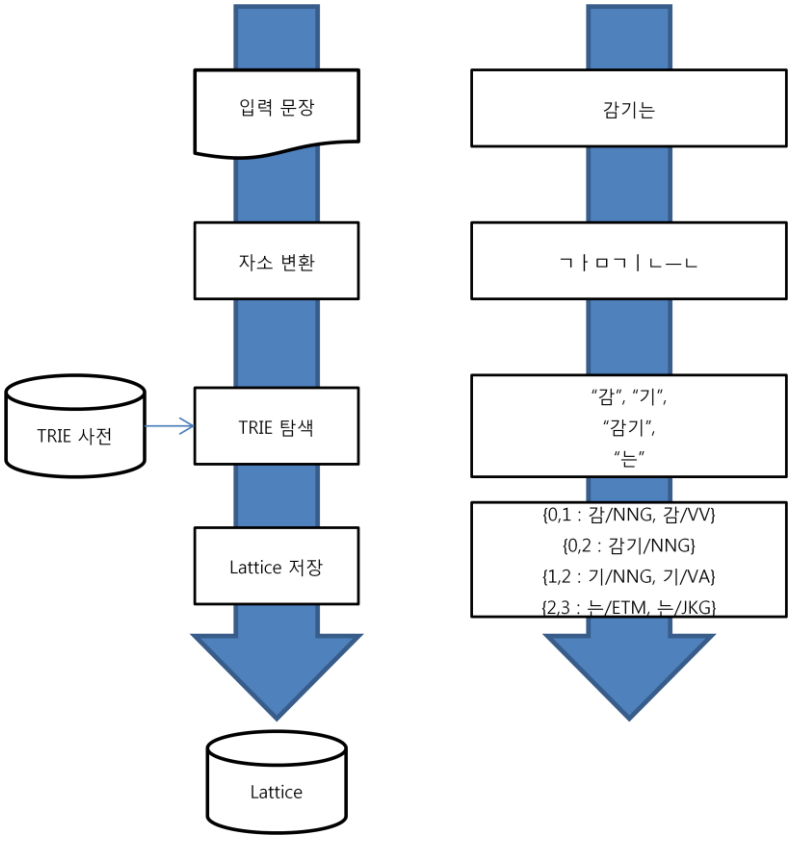

그림 2. 형태소 분석 단계 순서도

형태소 분석 단계의 최종 결과는 그림 3과 같이 각 형태소들로 구성된 Lattice이다. 실제 동작 단계에서는 Lattice를 생성함과 동시에 품사 태깅을 수행하여 연산 속도를 높였다.

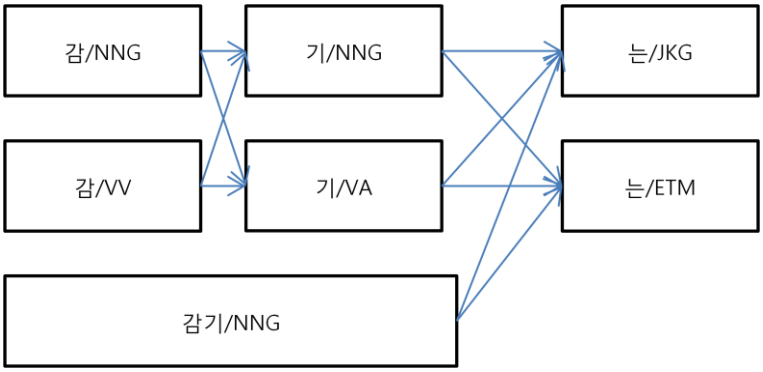

그림 3. 형태소 분석 결과

1.2.2 품사 태깅(POS Tagging)

품사 태깅은 수식 1과 같은 HMM(Hidden Markov Model)을 사용한다.

$$\arg\max_{q_{1,n}} \prod_{t=1}^{n} P(X_t \mid q_t) P(q_t \mid q_{t-1}) \quad (1)$$

Xt는 t번째 단어 x를 나타내며 qt는 t번째 품사 q를 나타낸다. HMM은 연속적인 패턴을 인식하는데 무난한 성능을 보이는 기계 학습 방법이다. 기존의 많은 연구를 통하여 형태소 분석 분야에서 HMM 보다 더 나은 성능을 보이는 CRFs, MEMM, SVM 등과 같은 기계 학습 방법들이 있다. 실제로 CRFs, MEMM, SVM으로 연구된 형태소 분석기들의 성능은 HMM 보다 약 2~5% 정도 좋은 성능을 보였다. 그러나 KOMORAN에서 HMM을 사용한 이유는 앞서 언급한 것과 같이 실제 필드에서 유용한 형태소 분석기를 만들기 위함이다. 추후 사용자 사전, 기분석 사전 등을 별도의 트레이닝이나 소스 코드 수정 없이 기존의 성능을 최대한 유지하면서 반영하기 위해서 비교적 간단한 알고리즘인 HMM을 사용하였다. 실제 형태소 분석기는 분석 성능 자체도 큰 의미가 있지만 시간의 흐름에 단어의 생성과 소멸이 반복적으로 나타나기 때문에 사용자 사전 및 기분석 사전의 관리에 따라 형태소 분석기의 품질이 결정된다.

1.3 특징

1.3.1 Pure Java

KOMORAN은 순수 자바로만 개발되었기 때문에 일반 PC, 스마트폰 뿐만 아니라 자동차 및 냉장고에서도 JVM만 설치된 환경이라면 어디서든지 바로 사용할 수 있다. 실제로 일반 PC 뿐만 아니라 안드로이드, 서버 등에서도 원활하게 동작하는 것을 확인하였다. (누군가 냉장고나 자동차에 탑재시켜 동작하는 것을 확인시켜주시길...(?))

1.3.2 외부 라이브러리 독립성

SHINEWARE에서 자체 제작한 라이브러리들만을 사용하여 구성되었기 때문에 외부 라이브러리와의 의존성 문제에서 자유롭다. 다시 말하면 의존성이 걸린 외부 라이브러리가 업

데이트 등으로 변경되었을 때 관련 메소드의 삭제 및 변경으로 인해서 생기는 부담이 전혀 없다.

1.3.3 경량화

자소 단위 처리, TRIE 사전 사용 등으로 약 32MB에서도 정상동작 하는 것을 확인하였다 (lite 모델 기준).

1.3.4 쉬운 사용법

소스 코드 내부에서 KOMORAN 객체를 생성하는 1줄이면 최소한의 형태소 분석기를 사용할 준비가 끝난다. 또한 사용자 사전 및 기분석 사전과 같은 외부 자원도 각 파일의 위치만 지정해주면 바로 적용이 가능하다.

1.3.5 사전 관리

사용자 사전 및 기분석 사전을 관리하는데 용이한 구조로 설계되었다. 사용자 사전 및 기분석 사전은 라인의 가장 앞에 '#' 문자를 넣어서 주석처리가 가능하다. 또한 일반 텍스트 파일의 형태로 되어있어 가독성이 높으며 바로 삽입/삭제/수정이 가능하다. 사전을 편집하기 위해서 사전의 구조, 규칙, 포맷 등을 파악할 필요가 없다. 또한 편집 후 추가적인 빌드 등과 같은 부가 작업 역시 필요 없다.

1.3.6 새로운 분석 결과

KOMORAN 2.0와 타 형태소 분석기 간의 가장 큰 차이점은 공백을 포함시켜 단일 형태소로 분석이 가능하다는 점이다. 예를 들어 "바람과 함께 사라지다를 봤다."와 같은 입력 문장에 대한 기존 형태소기들의 분석 결과는 대부분 그림 4와 같이 어절 단위를 형태소 분석 대상으로 한다. 그러나 이러한 방식들은 영화명, 노래명, 상호명 등과 같이 띄어쓰기를 포함하여 하나의 고유명사를 이루는 단어가 출현하는 문장을 분석하는 경우에는 적합하지 않다. 고유명사 분석 문제를 해결하기 위해서 제공되어야 하는 사용자 사전 및 기분석 사전 역시 대부분의 기존 형태소 분석기에서는 공백을 허용하지 않는다. 이러한 문제를 해결하기 위해서는 개체명 인식, 전문용어 인식 등과 같은 별도의 후처리 작업이 필요하며 실제로 이를 인식하기 위한 다양한 연구들이 진행되고 있다.

바람과 함께 사라지다를 봤다.	
바람과	바람/NNG+과/JKB
함께	함께/MAG
사라지다	사라지/VV+다/EC+를/JKO
봤다.	보/VV+았/EP+다/EF+./SF

그림 4. 기존 형태소 분석기의 분석 결과

그러나 KOMORAN 2.0은 기존 형태소 분석기와 같이 어절 단위의 분석 결과도 제공하지

만 다수의 어절을 공백을 포함한 하나의 형태소로 분석하는 기능도 제공한다. 예를 들어 사용자 사전에 "바람과 함께 사라지다"를 넣거나 별도의 학습 코퍼스로 학습을 시킨 데이터가 있다면 그림 5와 같은 결과를 출력하게 된다.

```
바람과 함께 사라지다를 봤다.
바람과 함께 사라지다를        바람과 함께 사라지다/NNP+를/JKO
봤다.                         보/VV+았/EP+다/EF+./SF
```

그림 5. KOMORAN 2.0 형태소 분석기의 분석 결과

이는 단순 영화명, 노래명, 상호명 뿐만 아니라 의학 용어, 법률 용어 등과 같이 전문 용어가 포함된 문장을 분석하는데도 중요한 역할을 할 것으로 기대된다.

1.4 성능

KOMORAN 2.0의 성능을 알아보기 위해서 세종 코퍼스 전체 데이터의 90%를 학습데이터로 사용하고 나머지 10%는 실험데이터로 사용하였다. 먼저 분석 정확률(accuracy)에 대한 실험 결과는 아래 표 1과 같다. (2014년 11월 24일 기준 KOMORAN 2.4 버전으로 실험)

표 1. KOMORAN 2.4 분석 정확률

	형태소 별 품사 정확률	어절 정확률
한글 어절	95.89 %	93.60 %
전체 어절(숫자, 기호 등 포함)	95.36 %	92.12 %

한글 어절은 어절 내 숫자, 기호, 영어 등과 같은 문자가 포함되지 않은 순수 한글로만 구성된 어절만을 대상으로 측정한 결과이다. 이 때 분석된 형태소 별 품사 정확률은 95.89% 였으며 어절 전체에 포함된 품사를 정확하게 맞춘 경우는 93.60%였다. 또한 한글로만 구성된 어절 뿐만 아니라 숫자, 기호, 영어 등을 포함한 전체 어절에 대한 형태소 별 품사 정확률 및 어절 정확률은 각각 95.36%, 92.12%였다. 이 수치는 기존의 형태소 분석기와 객관적인 성능 비교를 위한 수치이며, KOMORAN 2.0은 이러한 수치보다 개발 배경에서 언급한 것과 같이 실제 현업에서 사용하기 위한 유연한 기능들에 중점을 두었기 때문에 이에 대한 추가 실험이 필요할 것으로 예상된다.

또한 KOMORAN 2.0의 속도를 알아보기 위하여 표 2와 같이 어절, 문장, 용량에 따른 분석 속도를 측정하였다. 실험에는 2.3GHz Intel Core i5 (Dual core) CPU, 8GB RAM의 노트북이 사용되었다.

표 2. KOMORAN 2.4 분석 속도

단위	초당 단위 분석 수 (단위/sec)
어절 (Word)	30,000
문장 (Sentence)	6,000
용량 (MB)	0.3

실험 결과 1초에 약 3만 어절, 6천 문장, 300kb의 분석 속도를 보였다. 이러한 수치는 타 형태소 분석기에 비해 동등하거나 조금 높은 수준이지만, 메모리 사용량 및 부가적으로 제공되는 기능들을 고려한다면 쓸만한 수치라는 것을 짐작할 수 있다.

2. 사용방법

2.1 사용환경

KOMORAN 2.0은 자바 버전 7(1.7) 이상을 필요로 한다. 자바 7 이상이 설치된 환경이면 플랫폼 독립적으로 사용이 가능하다.

2.2 다운로드

KOMORAN 2.0 및 관련 파일들은 SHINEWARE 블로그 및 공식 사이트에서 다운로드가 가능하다.

- 블로그 : http://shineware.tistory.com
- SHINEWARE 공식 사이트 : http://www.shineware.co.kr

다운로드가 필요한 파일 및 각 파일의 간략한 설명은 표 3과 같다.

표 3. 다운로드 파일명 및 설명

파일명	설명	필수 여부
komoran-2.x-e.jar	KOMORAN 2.0의 핵심 라이브러리	필수
shineware-common-2.x.jar	파일, 문자열 처리 등과 같은 유틸리티 라이브러리	필수
shineware-ds-1.x.jar	Trie, Lattice 등과 같은 자료구조 라이브러리	필수
models-light.zip	형태소 분석 시 필요한 사전, 확률 값 등이 포함된 모델	두 가지 모델 중 한가지만 선택 가능
models-full.zip	models-light.zip의 내용에 한국어 위키피디아의 문서 제목들을 추가로 학습 시킨 모델(용량이 크다고 좋은 성능을 내는 것은 아님)	

2.3 실행

2.3.1 Quick start

간단한 실행을 원한다면 아래와 같은 순서대로 진행하면 분석 결과를 쉽게 얻을 수 있다. 사용자 사전 및 분석 관련 메소드와 관련된 내용은 다음 절에서 설명한다.

a) komoran 실행에 필요한 파일들을 다운로드 받은 후 라이브러리 path 설정을 한다.

b) models-light.zip 또는 models-full.zip의 압축을 해제한다.

c) 그림 6과 같이 소스를 작성한다.

```
Komoran komoran = new Komoran("b)에서 압축 해제한 경로");

List<List<Pair<String,String>>> result = komoran.analyze("이
곳은 입력 문장의 위치입니다.");

for (List<Pair<String, String>> eojeolResult : result) {

        for (Pair<String, String> wordMorph : eojeolResult) {

                System.out.println(wordMorph);

        }

        System.out.println();

}
```

그림 6. KOMORAN 2.0의 간략한 사용 예

2.3.2 주요 메소드

- analyze(String sentence)
  - ■ 입력된 sentence의 분석 결과를 반환
  - ■ 분석 결과의 형태소에 따라 공백 포함 가능
  - ■ 예제

```
입력 문장 :
        바람과 함께 사라지다를 봤다.
분석 결과 :
        바람과 함께 사라지다/NNP+를/JKO
        보/VV+았/EP+다/EF+./SF
```

- analyze(String sentence,int nBest)
  - ■ 2.4 이상에서만 제공
  - ■ 입력된 sentence의 분석 n-best 분석 결과를 반환
  - ■ 분석 결과의 형태소에 따라 공백 포함 가능
  - ■ 예제

```
입력 문장 :
        바람과 함께 사라지다를 봤다.
분석 결과 1 :
        바람과 함께 사라지다/NNP+를/JKO
        보/VV+았/EP+다/EF+./SF
분석 결과 2 :
        바람과 함께 사라지다/NNP+를/JKO
        보/VX+았/EP+다/EF+./SF
```

- 어절 단위의 n-best 분석 결과를 얻기 위해서는 sentence를 공백 단위로 tokenize하여 각 token을 입력

- analyzeWithoutSpace(String sentence)
  - 2.2 이상에서만 제공
  - 입력된 sentence의 분석 결과를 반환
  - 공백 단위 형태소 분석(기존 형태소 분석기들의 분석 대상과 동일)
  - 예제

```
입력 문장 :
        바람과 함께 사라지다를 봤다.
분석 결과 :
        바람/NNG+과/JKB
        함께/MAG
        사라지/VV+다/EC+를/JKO
        보/VV+았/EP+다/EF+./SF
```

- setFWD(String filename)
  - 기분석 사전 적용
  - 기분석 사전은 분석 시 우선순위가 가장 높음

- setUserDic(String filename)
  - 사용자 사전 적용
  - 분석 대상 문장 내 사용자 사전에 해당하는 단어가 있는 경우 그에 따른 품사가 적용

2.3.3 기분석 사전

- 형식
  - [분석할 어절][tab][분석 결과(형태소는 공백으로 구분)]
  - 시작 기호 '#'으로 주석 처리 가능

- 예제

```
#이 라인은 주석입니다.
수입불가          수입/NNG 불가/NNG
자연어처리는      자연어처리/NNP 는/JKG
```

2.3.4 사용자 사전

- 형식
  - [단어][tab][품사]
  - 시작 기호 '#'으로 주석 처리 가능
- 예제

```
#이 라인은 주석입니다.
#단어는 공백 포함 가능
바람과 함께 사라지다      NNP
캐리비안의 해적 NNP
SHINEWARE       NNP
```

3. 라이선스

KOMORAN 2.0은 SHINEWARE에게 모든 저작권이 있으며, 별도의 동의 없이 역컴파일을 할 수 없다. 비상업적인 용도에 한해 자유로운 사용이 가능하다. 상업적인 용도로 사용시 별도의 협의를 통해 사용이 가능하다. 단, 설립 3년 미만이면서 10명 이하의 인력으로 구성된 사업장에서는 자유롭게 사용 가능하다. 그러나 사업장 규모의 확장 및 발전에 따라 설립 3년 이상 또는 인력 10명 초과 시에는 별도의 협의가 필요하다. 이 외의 언급되지 않은 부분은 GPL 2.0 라이선스를 따른다.

4. 맺음말

KOMORAN 2.0과 관련된 문의 사항은 ceo@shineware.co.kr 로 보내주시기 바랍니다.