SyntaxNet 구현 및 최소학습 판단 시스템 적용

조재춘

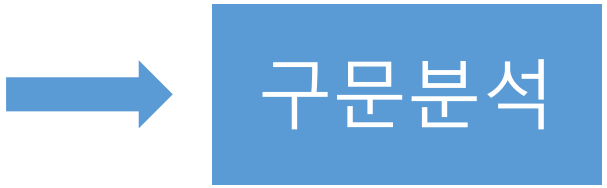
SyntaxNet이 무엇인가?

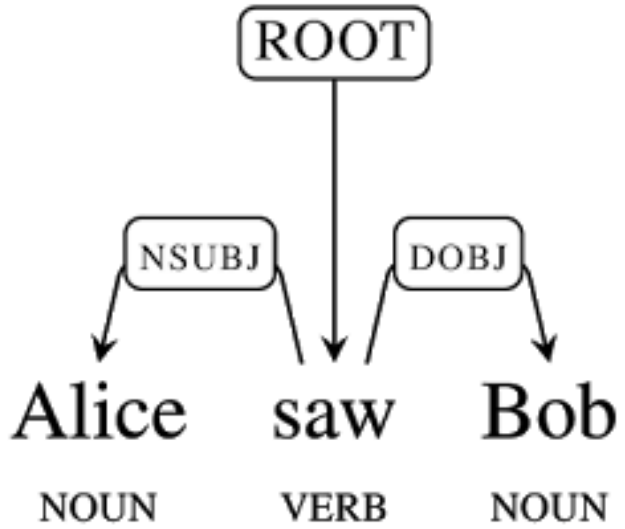
- 인공 신경망 프레임워크
- 오픈소스
- TensorFlow를 통한 자연어 이해
- 새로운 언어 학습 가능
- 구문 분석기 – Parsey McParseface
  - 언어의 구조를 분석
  - 세계에서 가장 정확도가 높은 모델
  - 자동 정보 추출, 언어 번역, 자연어 이해에 활용 가능

SyntaxNet 작동 원리

Alice Saw Bob

문장

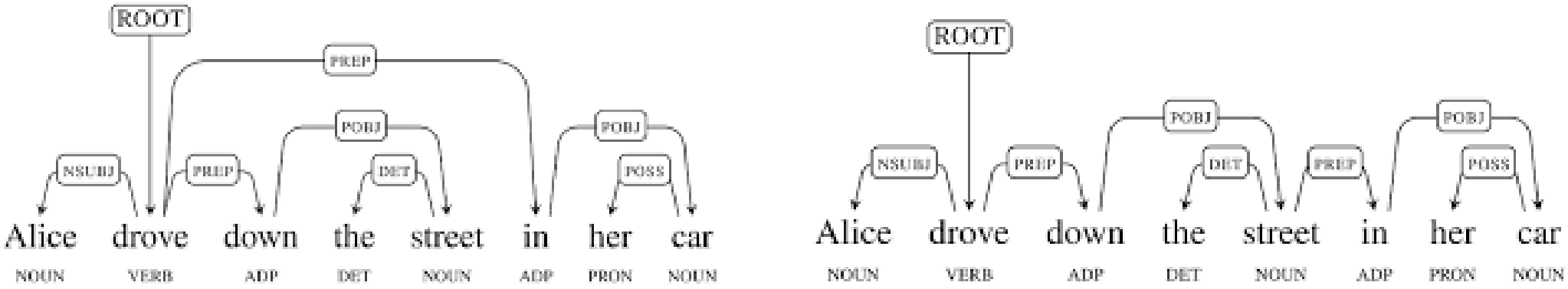




의존성 구문분석 트리

구문분석 원리

- 컴퓨터 구문분석에서 가장 큰 문제 : 모호함 (Ambiguity)
- 사람들은 보통 20~30개 단어 길이 문장을 사용
- 수만가지의 문법적 경우의 수를 가짐

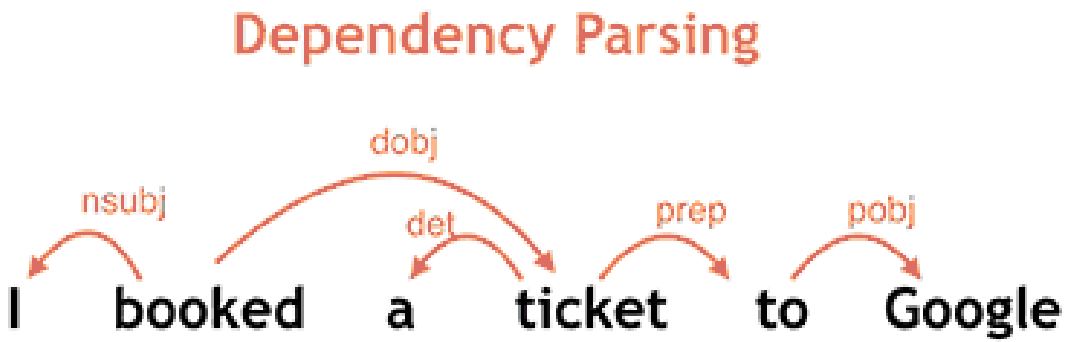


전치사 접속 모호성(Prepositional phrase attachment ambiguity)

구문분석 원리

- 가능한 다양한 중의적 해석을 모두 검토하려면 그 조합의 경우 수가 급격히 증가함
- 상당수는 상식적으로 말이 안되는 해석임
- 그럼에도 문법적으로 가능한 해석이라면 후보군으로 고려해야 함
- SyntaxNet은 모호성 문제를 인공신경망을 적용하여 해결함

SyntaxNet 원리

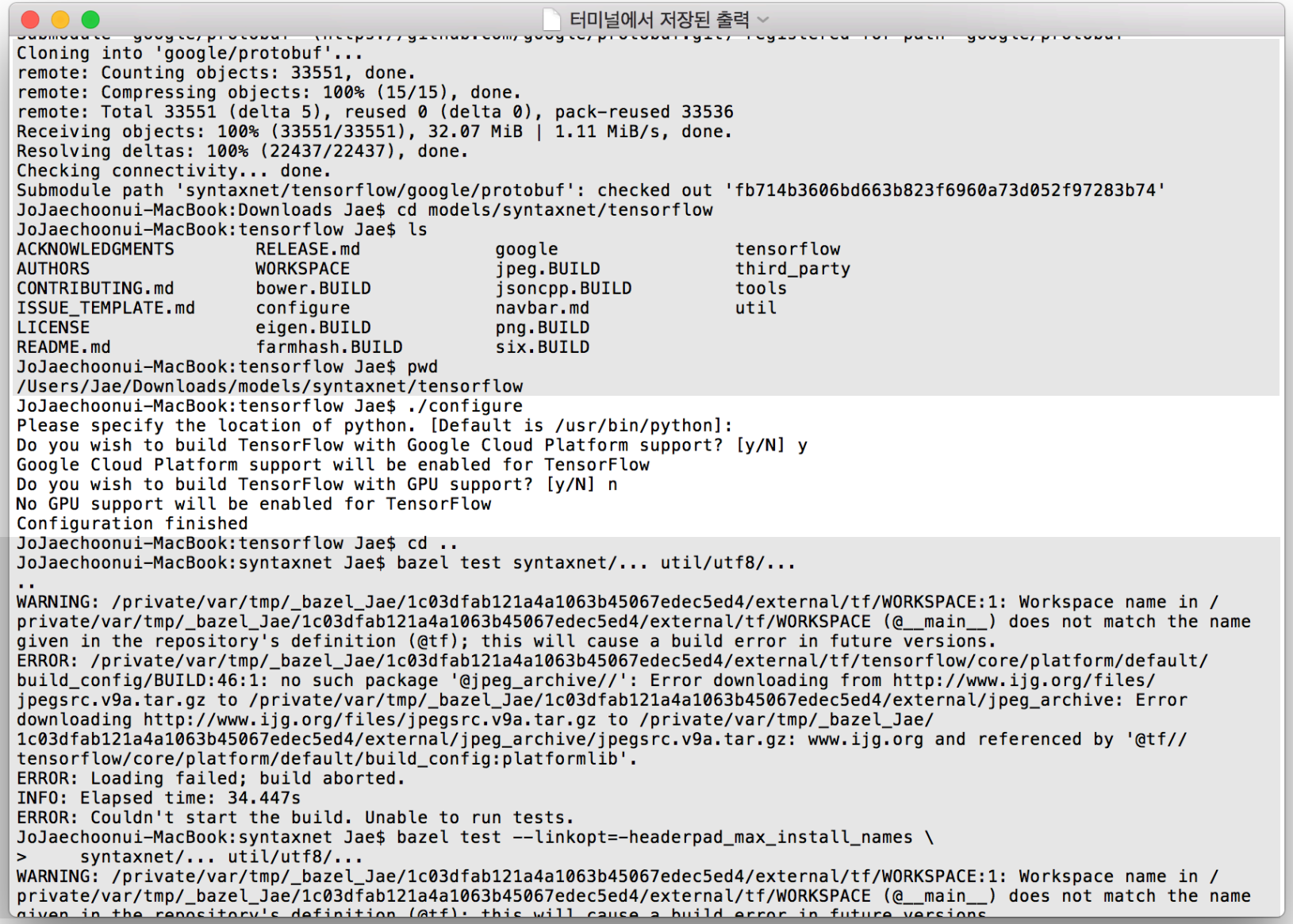


- 입력된 문장을 왼쪽에서부터 오른쪽으로 처리하면서, 각 단어들 사이의 의존성을 고려하여 점진적으로 추가
- 각 단계마다 모호한 가능성들에 대해 명확한 판단을 내려야하는데, 이때 Neural network을 사용하여 조금 더 그럴듯한(plausibility) 방향으로 해석하도록 결정
- 매번 선택의 순간마다 최적우선탐색을 하도록 단순반복하는 것보다는, 다양한 가능성을 유지시키다가 명백한 결격사유가 등장했을 때에야 비로소 후보군에서 탈락시키고 최적의 모범답안을 채택

SyntaxNet 설치

- python 2.7:
  - python 3 support is not available yet
- bazel: 다양한 플랫폼에서 코드를 빌드하고 테스트할 수 있도록 하는 패키지
  - **versions 0.2.0 - 0.2.2b, NOT 0.2.3**
- swig: Simplified Wrapper and Interface Generator
  - apt-get install swig on Ubuntu
  - brew install swig on OSX
- protocol buffers, with a version supported by TensorFlow:
  - check your protobuf version with pip freeze | grep protobuf
  - upgrade to a supported version with pip install -U protobuf==3.0.0b2
- asciitree, to draw parse trees on the console for the demo:
  - pip install asciitree
- numpy, package for scientific computing:
  - pip install numpy

환경설정

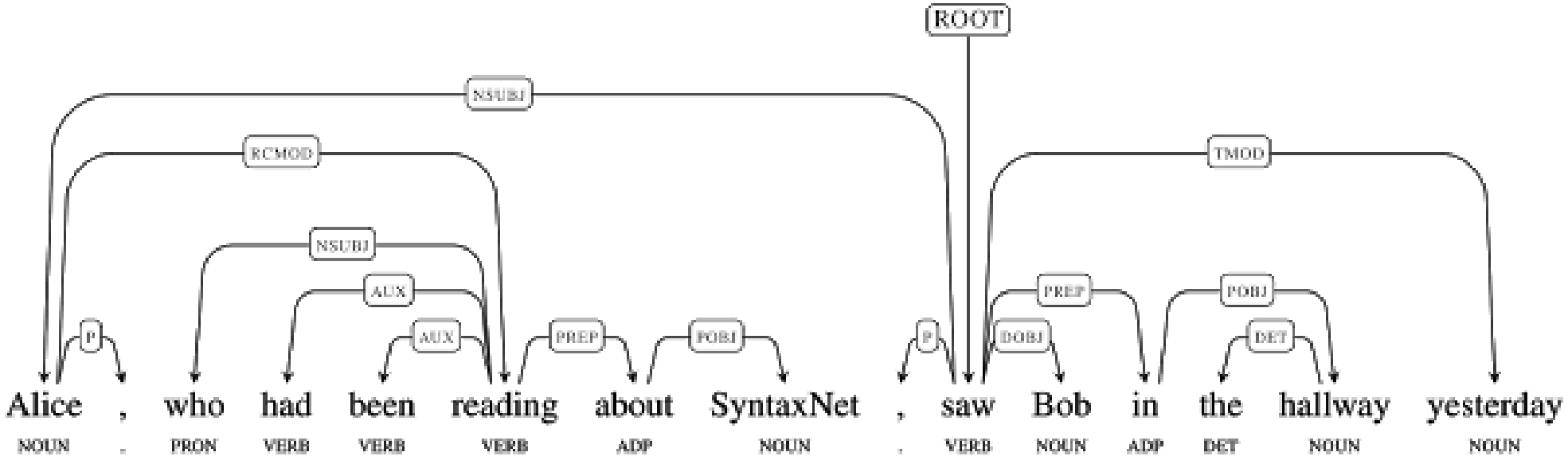


- 파이썬의 기본 경로는 어디인지? (default : /usr/bin/python)
- TensorFlow 빌드할 때 구글 클라우드 플랜폼에서 가져올 것인지? (default : N)
- TensorFlow에서 GPU 옵션을 할 것인지? (default : N)

SyntaxNet 시연

```
echo 'Bob brought the pizza to Alice.' | syntaxnet/demo.sh
Input: Bob brought the pizza to Alice .
Parse:
brought VBD ROOT
   +-- Bob NNP nsubj
   +-- pizza NN dobj
   |    +-- the DT det
   +-- to IN prep
   |    +-- Alice NNP pobj
   +-- . . punct
```

최소학습 판단 시스템 적용



- 최소학습 판단 시스템 적용 -> 자동 학습 판단 시스템
- 핵심 단어 추출
  - 문장에서, 동영상에서 의미 있는 단어를 추출할 수 있음
- 자동 질문 생성으로 의미를 부여한 퀴즈 제시
  - 자동 질문과 답변을 생성
  - Ex1) Alice가 누구를 보았는가?, Bob을 본 사람은 누구인가?, Alice가 읽은 책은 무엇에 관한 것인가?
  - Ex2) 빈칸, Alice, who had been reading about SyntaxNet, saw _____ in the hallway yesterday

SyntaxNet 적용 한계점 및 향후 연구방향

- 활용 예시와 소스가 많지 않음
- KOMORAN 영어 형태소 분석기 적용
- 영어 콘텐츠에 대한 최소학습 판단을 위한 실험 설계

```
public static void main(String[] args) throws Exception {
	EnPosta posta = new EnPosta();
	posta.load("model");
	//사용자 사전 추가
	posta.appendUserDic("dic.user");
	posta.buildFailLink();

	List<String> resultList = posta.analyze("Launch a new institute at the University of Washington to cond
uct independent, rigorous evaluations of health programs worldwide.");
	for (String result : resultList) {
		System.out.println(result);
	}
}
```

```
//단어 가져오기¤¶
String szUrl = "http://video.google.com/timedtext?lang=en&v=" + url;¤¶
InputStream is = null;¤¶
InputStreamReader isr = null;¤¶
is = new URL(szUrl).openStream();¤¶
isr = new InputStreamReader(is, "utf-8");¤¶
StringBuffer sb = new StringBuffer(); int c;¤¶
while ((c = isr.read()) != -1) {sb.append((char) c);}¤¶
isr.close(); ¤¶
is.close();¤¶
String text = sb.toString();¤¶
¤¶
//단어 테그 및 불필요한 단어 삭제¤¶
text = text.replaceAll("<(/)?([a-zA-Z]*)(\\s[a-zA-Z]*=[^>]*)?(\\s)*(/)?>", " ");¤¶
text = text.replaceAll("&amp;", " ");¤¶
text = text.replaceAll("quot;", " ");¤¶
text = text.replaceAll("quot;", " ");¤¶
text = text.replaceAll("<\\?xml version\\=\"1.0\" encoding\\=\"utf-8\" \\?>", " ");¤¶
text = text.replaceAll("[^a-zA-Z0-9\\s]", "");¤¶
```

```
EnPosta posta = new EnPosta();¤¶
String pa = this.getClass().getResource("/").getPath() + "/models/eng";¤¶
¤¶
posta.load(pa);¤¶
//사용자 사전 추가¤¶
//posta.appendUserDic("dic.user");¤¶
posta.buildFailLink();¤¶
¤¶
ArrayList<ContentWord> contentWordList = new ArrayList<ContentWord>();¤¶
String[] values = null;¤¶
int index = 0;¤¶
boolean chk = true;¤¶
List<String> resultList = posta.analyze(text);¤¶
for (String result : resultList) {¤¶
»   values = result.split("/");¤¶
»   //명사만 추출하여 빈도 수 처리¤¶
    if ("NNG".equals(values[1]) || "NNP".equals(values[1]) || "NN".equals(values[1]) ){¤¶
        String word = values[0];¤¶
        ¤¶
        if ("".equals(word)) continue;¤¶
        ¤¶
```

```
  index = 0;¤¶
  chk = true;¤¶
  if ( !contentWordList.isEmpty() ){¤¶
      for (ContentWord contentWord : contentWordList) {¤¶
          if (word.equals(contentWord.getWordKname())){¤¶
          »   contentWord.setFrequency(contentWord.getFrequency() + 1); //빈도 수 처리¤¶
              contentWordList.set(index, contentWord);¤¶
              chk = false;¤¶
          }¤¶
          index++;¤¶
      }¤¶
  }¤¶
  ¤¶
  //단어가 처음 나왔을 경우 결과 리스트에 추가하는 작업¤¶
  if (chk){¤¶
      ContentWord addItem =  new ContentWord();¤¶
      addItem.setContentSeq(contentSeq);¤¶
      addItem.setWordKname(word);¤¶
      addItem.setFrequency(1);¤¶
      addItem.setWordLength(word.length());¤¶
      addItem.setLangCode("eng");¤¶
      contentWordList.add(addItem);¤¶
  }¤¶
```