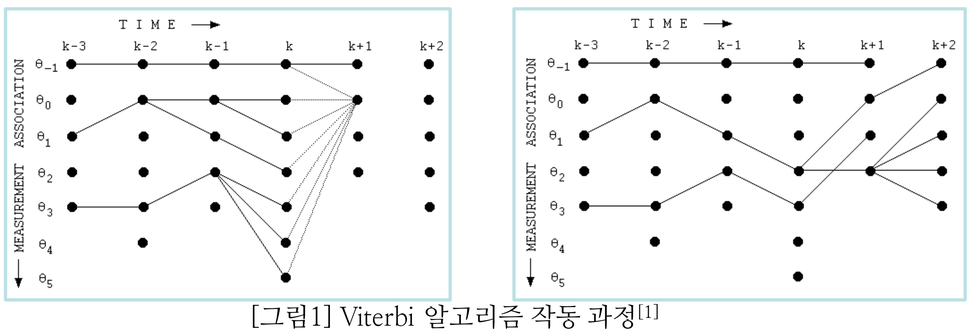
지도학습 기반의 은닉 마르코프 모델(Hidden Markov Model)을 활용한 품사(Part-of-Speech, POS) 태깅

2018 Project 2 포스터 모범사례

Background

- 품사 태깅(POS Tagging)
  - 자연어처리 과정에서, 주어진 문장을 이해하기 위해서는 그 문장이 정확한 의미를 가질 수 있도록 모호성을 없애는 과정이 필요하다. 이는 단계적(음절별, 단어별, 구문별 등)으로 처리해 나가게 된다.
  - 품사(Part-of-Speech, POS) 태깅은 문장의 모호성을 제거하는 과정 중 하나이다. 가령 '새'라는 단어는 '새로운'과 '동물'로 해석이 가능한데 품사 태깅을 통해 '새'라는 단어의 모호성을 해소할 수 있다.

- 지도학습 방식의 은닉 마르코프 모형(Hidden Markov Model, HMM)
  - Viterbi 알고리즘: HMM에서 최적 경로를 찾는 알고리즘



[그림1] Viterbi 알고리즘 작동 과정[1]

만약 첫번째 단어가 θ_2 → 최적경로 (2 ,2, 2, 1, 0, 1)

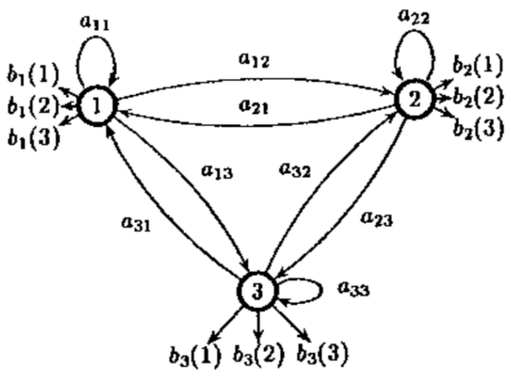
  - 품사 태깅을 위한 HMM 모델링[2]
    - State(X)의 개수는 품사의 개수와 같다. 다시 말해, 1:1 매칭이 된다.
    - Evidence(e)의 개수는 문장의 단어 수와 같다. 세상에 존재하는 모든 단어를 가능한 한 포함해야 하므로 그 값이 크다.
    - 상태전이확률, a_{ij} :

$$a_{ij} = \Pr(q_t = S_j \mid q_{t-1} = S_i),\ 1 \le i,\ j \le N$$

    - 상태확률 – 어떤 상태S_j에서 어떤 단어 e_k가 나올 확률, $b_j(k)$:

$$b_j(k) = \Pr(q_t = w_k \mid q_t = S_j),\ 1 \le j \le N,\ 1 \le k \le M$$

    - 품사와 단어가 3종류인 모델의 예시는 다음의 [그림2]와 같다.



상태 1 : 명사 — 단어 1 : 학교
상태 2 : 동사 — 단어 2 : 날다
상태 3 : 형용사 — 단어 3 : 아름답다

[그림 2] 품사 태깅 HMM 모델의 예

Goals

- 순서가 주요한 영향을 미치는 라벨링된 데이터의 HMM 모델링 과정을 이해한다.
  - 사전확률과 상태전이확률의 도출 과정을 이해한다.
  - 최적 경로를 찾는 방법, 즉 어떤 문장의 품사 태깅 과정을 이해한다.
- 주어진 데이터셋과 품사 태그를 활용하여 품사 태그가 주어지지 않은 문장의 품사를 도출해내는 지도학습 방식의 HMM을 모델링한다.
  - 테스트셋의 결과를 확인하고, 전체 정확도를 확인한다.
  - 데이터셋의 특성에 따른 정확도 차이를 분석한다.

Data Description

- 활용 데이터셋
  - NLTK (Natural Language Toolkit) 패키지[3]: 자연어 처리 및 분석용 Python 라이브러리. 말뭉치와 품사 태깅모듈 등 제공.
    - NLTK에서 제공하는 말뭉치 중 'Penn Treebank Sample' 과 'CESS-CAT Treebank' 을 이용
    - Penn Treebank Sample : (174,0034 bits) 펜실베니아 대학에서 Treebank 형태로 제공하는 1989년 월스트리트 저널 말뭉치
    - CESS-CAT Treebank : (539,6688 bits) 구문론적, 의미론적으로 파싱(Parsing) 된 카탈루니아어 말뭉치. 약 500,000개의 단어 제공.
  - 각각의 데이터셋의 0.9를 Training data로 하여 HMM의 전이확률 및 상태확률 분포를 학습하는 데에 사용하고 나머지 0.1을 Test data로 하여 성능 및 결과 도출에 사용

Model & Algorithm Description

- NLTK 라이브러리 내의 nltk.tag.hmm 모듈[4][5]
  - 기본적인 HMM을 지도학습과 비지도학습 두 가지 방식으로 구현하여 제공
    1. MLE(Maximum Likelihood Estimation) 기반의 지도학습 알고리즘
    2. Expectation-Maximization을 기반으로 한 Baum-Welch 알고리즘

    1., 2. 모두 Viterbi 알고리즘을 바탕으로 최적경로를 제공
  - HiddenMarkovModelTrainer(sequence) 클래스
    - 품사를 상태변수로, 단어를 관측변수로 가지는 HMM 생성
    - 전이확률과 상태확률 분포는 sequence 내에서 해당 사건이 나오는 경우의 수를 모두 더하여 전체 사건 수로 나눈 확률분포를 이용
- 알고리즘 설명

  1. treebank, cess_cat 말뭉치(+품사정보)를 다운 받아 각각 dataT, dataC에 저장
  2. dataT, dataC의 상위 9할을 trainT, trainC에 저장하고, 품사정보를 없앤 나머지 1할을 testT, testC에 저장
  3. trainT, trainC 각각에 대한 HiddenMarkovModelTrainer 클래스를 생성한 뒤, 이를 학습하여 testT, testC의 품사를 태깅한다.
  4. HMM을 이용하여 태깅한 품사와 실제 품사를 비교하여 틀린 갯수를 저장한다.

Results & Analysis

- trainT를 학습하여 testT 문장을 태깅한 결과의 일부는 다음과 같다.
  - 29번째 문장

```
[('The', 'DT'), ('plan', 'NN'), ('relies', 'VBZ'), ('heavily', 'RB'), ('on', 'IN'), ('$', '$'), ('240', 'CD'), ('million', 'CD'), ('*U*', '-NONE-'), ('in', 'IN'), ('credit', 'NN'), ('and', 'CC'), ('loan', 'NN'), ('guarantees', 'NNS'), ('in', 'IN'), ('fiscal', 'JJ'), ('1990', 'CD'), ('in', 'IN'), ('hopes', 'NNS'), ('of', 'IN'), ('*', '-NONE-'), ('stimulating', 'VBG'), ('future', 'JJ'), ('trade', 'NN'), ('and', 'CC'), ('investment', 'NN'), ('.', '.')]
```

  - 46번째 문장

```
[('The', 'DT'), ('airline', 'NN'), ('is', 'VBZ'), ('attempting', 'VBG'), ('*-1', '-NONE-'), ('to', 'TO'), ('show', 'NN'), ('that', 'IN'), ('Israel', 'NNP'), ('and', 'CC'), ('West', 'NNP'), ('Germany', 'NNP'), ('warned', 'VBN'), ('the', 'DT'), ('U.S.', 'NNP'), ('about', 'IN'), ('the', 'DT'), ('impending', 'VBG'), ('attack', 'NN'), ('.', '.')]
```

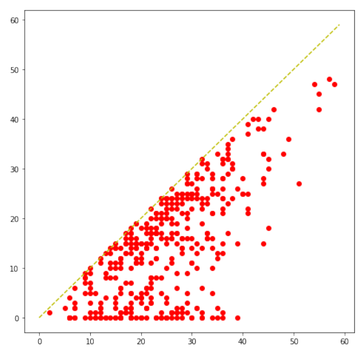
- trainC를 학습하여 testC 문장을 태깅한 결과의 일부는 다음과 같다.
  - 1번째 문장

```
[('La', 'da0fs0'), ('dissolució', 'ncfs000'), ('de', 'sps00'), ('la', 'da0fs0'), ('reunió', 'ncfs000'), ('no', 'rn'), ('va', 'vaip3s0'), ('impedir', 'vmn0000'), ('que', 'cs'), ('alguns', 'pi0mp000'), ('dels', 'spcmp'), ('seus', 'dp3mp0'), ('membres', 'ncmp000'), ('mantinguessin', 'vmsi3p0'), ('el', 'da0ms0'), ('contacte', 'ncms000'), ('durant', 'sps00'), ('la', 'da0fs0'), ('matinada', 'ncfs000'), ('.', 'Fp')]
```

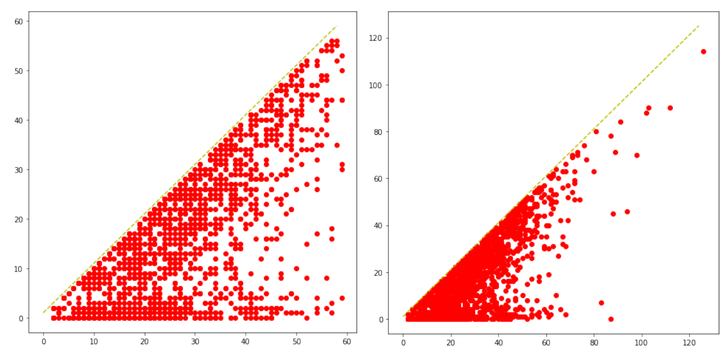
  - 92번째 문장

```
[('El', 'da0ms0'), ('boom', 'ncms000'), ("d'", 'sps00'), ('aquesta', 'dd0fs0'), ('fórmula', 'ncfs000'), ('comercial', 'aq0cs0'), ('és', 'vsip3s0'), ('conseqüència', 'ncfs000'), ('dels', 'spcmp'), ('processos', 'ncmp000'), ('de', 'sps00'), ('concentració', 'ncfs000'), ('en', 'sps00'), ('la', 'da0fs0'), ('distribució', 'ncfs000'), ('i', 'cc'), ('de', 'sps00'), ('les', 'da0fp0'), ('estrictes', 'aq0cp0'), ('limitacions', 'ncfp000'), ('imposades', 'aq0fpp'), ('per', 'sps00'), ('la', 'da0fs0'), ('Generalitat', 'np0000o'), ('per', 'sps00'), ('a', 'sps00'), ('la', 'da0fs0'), ('instal·lació', 'ncfs000'), ('de', 'sps00'), ('grans', 'aq0cp0'), ('superfícies', 'ncfp000'), ('.', 'Fp')]
```

- (x,y) = (문장의 길이, 틀린 품사의 갯수) 를 좌표평면에 도식한 결과는 아래의 [그림3], [그림4]와 같다.



[그림3] testT결과.



[그림4] testC 결과 (왼: 문장길이60이하, 오: 전체)

  - 노란 점선은 문장내 모든 품사가 틀렸을 때를 나타내는 기준선
  - 말뭉치의 크기가 큰 trainC HMM 결과가 더 좋다
  - 확인하려는 문장의 길이가 길수록 오차율의 기대값이 증가한다

References

[1] <비터비 알고리즘>. ratsgo's blog, Nov 2017. (https://ratsgo.github.io/data%20structure&algorithm/2017/11/14/viterbi/, 2018.06.05)

[2] <은닉 마르코프 모델을 이용한 두단계 한국어 품사 태깅>. 이상주(Sang-zoo Lee) 외 2인, 한국정보과학회 언어공학연구회 학술발표 논문집, 제22권 제1호, P136-146.

[3] NLTK Corpora (http://www.nltk.org/nltk_data/, 2018.06.05)

[4] <nltk.tag package>, NLTK 3.3 documentation. (https://www.nltk.org/api/nltk.tag.html?highlight=hmm#nltk.tag.api.TaggerI , 2018.06.05)

[5] <Source code for nltk.tag.hmm>, NLTK 3.3 documentation. (https://www.nltk.org/_modules/nltk/tag/hmm.html, 2018.06.05)