졸업논문 주제 제안서 **(w/** 예비실험**)**

2016-19620 정대용

Topic

POS-aware CoSDA-ML 의 Low Resource Language 감성분류 성능 분석과 도구 개발

General Problem Definition

Low-Resource Language (이하 LL)는 NLP 태스크를 진행하기 어렵다. LL 의 학습 데이터 부족 문제를 해결하기 위해, 데이터 증강 기법과 언어 전이 학습이 발전하였다.

Specific Problem Definition

(1) PA-CoSDA-ML 은 데이터 증강 측면과 언어 전이 학습 측면에서 새로운 방법을 제안하였으나, 이는 영어와 거리가 가까운 언어나 HL 에 대한 결과였을 뿐, 실제 LL 에서의 효과는 경험적으로 보인 바 없다.
(2) CoSDA-ML 과 PA-CoSDA-ML 은 M-BERT 와 XLM 을 사전학습 모델로 사용했으나, 감성분류에 사용된 tgt 언어는 XLM 에 사전학습된 바 없어 실험 자체를 진행하지 않았다. 이후 뛰어난 성능의 XLM-R 이 등장하였기에, XLM-R 로 모델을 교체 실험해볼 필요가 있다.
(3) PA-CoSDA-ML 을 LL 에 적용하기 위해 English-LL 사전을 구해야 하는데, LL 특성상 해당 표준화된 사전을 구하기 어렵다.

Content

(1) PA-CoSDA-ML 을 XLM-R 기반으로 변경한다.
(2) 사전(EN-LL dictionary) 빌드 도구를 개발한다. (구글 translate api 활용)
(3) (1)과 (2) 진행 후 완성된 PA-CoSDA-ML 에 5 개 이상의 LL 을 적용시켜 CoSDA-ML 과의 성능을 비교 분석한다.

의의

(1) PA-CoSDA-ML 에 SOTA 언어 모델을 적용함으로써 가장 높은 성능을 끌어낼 수 있다.
(2) 사전 빌드 도구의 개발은 CoSDA-ML 의 실험에 핵심적인 툴을 제공하여 후속 연구를 촉진시킬 수 있을 뿐 아니라, LL 의 데이터 확보 자체에도 도움이 된다.
(3) SOTA 를 달성한 모델에 실제 LL 을 적용해봄으로써 희귀 언어의 재구라는 NLP 본연의 목적을 달성할 수 있다.

예비 실험 결과

(1) XLM-R 기반 PA-CoSDA-ML 의 결과가 XLM 및 MBERT 기반 PA-CoSDA-ML 의 결과보다 뛰어난 성능을 내었다. (실험한 언어에서 f1 값이 모두 2%p 이상 상승)
(2) PA-CoSDA-ML 을 영어와 거리가 먼 새로운 언어에 적용했을 때에도 좋은 CoSDA-ML 보다 뛰어난 성능을 내었다. (한국어의 경우 80%이상을 기록)

Background

CoSDA-ML 이란?

CoSDA-ML 은 (Libo Qin et al.)이 제안한 언어 전이 학습에서의 코드 스위칭 데이터 증강 기법이다. 데이터가 풍부한 src 언어의 문장 내 토큰들을, 데이터가 부족한 tgt 언어의 토큰으로 교체하는 방식에서 나아가, 두 개 이상의 tgt 언어들의 토큰들로 교체하는 것이 핵심적인 알고리즘이다. 이를 통해 같은 intent 에 대해 백터공간 내에서 언어 간 거리를 더욱 가깝고 더 크게 overlap 시킬 수 있음을 보였다.

POS-aware CoSDA-ML (PA-CoSDA-ML)

Code switching data augmentation 은, zero-shot 언어 전이 학습에서, src 언어 문장의 토큰을 code-switching 하는 방식으로 tgt 언어의 데이터를 증강하는 기법을 의미한다. tgt 언어와 src 언어의 dictionary 만으로 데이터를 쉽게 증강할 수 있기 때문에 low resource language 의 NLP 태스크에 곧잘 사용된다. POS-aware 은 Code switching 시에 품사 정보를 바탕으로 교체할 토큰을 선택하는 기법이다. ADJ, ADV, V 가 다른 품사에 비해 더 큰 감성 정보를 갖기 때문에 해당 방식을 통해 감성분류 태스크의 성능을 제고할 수 있다.

PA-CoSDA-ML 의| Algorithm

```
CAND = {adj, adv, v}
for TOKEN in src_sentence:
        if TOKEN is in CAND:
                ReplaceToken()
        if TOKEN is not in CAND:
                Do as CoSDA-ML
```