|  |  |
| --- | --- |
| 교육 제목 | AI 프로젝트 기반 S/W 전문가 양성과정 |
| 교육 일시 | 22.01.13 |
| 교육 장소 | 집 |
| **교육 내용** | |
| 오전 | 1. 언어에 대한 다양한 정의 2. 사람들이 자신의 머리 속에 있는 생각을 다른 사람에게 나타내는 체계 3. 사물, 행동, 생각 그리고 상태를 나타내는 체계 4. 사람들이 자신이 가지고 있는 생각을 다른 사람들에게 전달하는 데 사용하는 방법 5. 사람들 사이에 공유되는 의미들의 체계 6. 문법적으로 맞는 말의 집합(절대적이 아님) 7. 언어 공동체 내에서 이해될 수 있는 말의 집합 8. 🡪 정보 전달 9. 사람의 생각(의도, 정보)를 컴퓨터에게 전달하는 방법 10. Native Interface  * 사람이 이해할 수 있지만, 엄격한 문법과 모호성이 없는 형태의 전달 방식 * 인공언어, ex 프로그래밍 언어  1. 사람이 실제 사용하는 형태에 가까운 전달 방식  * 자연 언어  1. 자연어 처리는 무엇인가? 2. 자연어란? - 자연어 혹은 자연 언어는 사람들이 일상적으로 쓰는 언어를 인공적으로 만들어진 언어인 인공어와 구분하여 부르는 개념 3. NLP   - 사람이 이해하는 자연어를 컴퓨터가 이해할 수 있는 값으로 바꾸는 과정(NLU)  - 더 나아가 컴퓨터가 이해할 수 있는 값을 사람이 이해하도록 바꾸는 과정   1. • Computer Vision • Image Recognition • Object Detection • Image Generation • Super Resolution   • Natural Language Processing • Text Classification • Machine Translation • Summarization • Question Answering • Speech Processing • Speech Recognition (STT) • Speech Synthesis (TTS) • Speaker Identification  • Reinforcement Learning   1. NLP Discrete value를 다룸 – 단어, 문장 분류 문제로 접근 가능 샘플의 확률 값을 구할 수 있음 – P(x = 단어) 2. NLP 연구 요구사항 Domain Knowledge - 언어적 지식이 필요함 Nasty Preprocessing – 전처리가 더러움 3. 전통적인 NLP와 Deep Learning을 이용한 NLP 4. 전통적인 NLP – Symbolic 기반 접근 5. NLP System with Deep Learning 6. NLP 패러다임의 변화 효율적인 embedding을 통한 성능 개선 – 단어, 문장, context embedding 7. NLP는 무엇 때문에 어려울까? 8. Ambiguity(모호성) – 언어는 마치 생명체와 같이 진화하며, 특히 효율성을 극대화하는 방향으로 진화, 최대한 짧은 문장 내에 많은 정보를 담고자 함. 정보량이 낮은 내용은 생략 9. Paraphrase(의역) – 문장의 표현 형식은 다양하고 비슷한 의미의 단어들이 존재하기 때문에 의역의 문제가 존재 10. 연속이 아닌 이산 이산 값을 갖는 자연어는 사람의 입장에서 인지가 쉬울 수 있으나, 기계의 입장에서는 매우 어려운 값임 OHE으로 표현된 값은 유사도나 모호성을 표현할 수 없다. 서로 다른 OH벡터끼리의 유사도나 거리는 모두 동일하다 딥러닝에서는 WORD EMBEDDING을 통해 해결 11. 분절한다 = Tokenization 한다 12. 모호한 띄어쓰기 ex) 농협용인육가공공장, 내동생고기, 안동시체육회 13. 평서문과 의문문의 차이 부재 14. 주어 부재 15. 한자 기반의 언어 16. 단어 중의성으로 인한 문제 발생 17. Neural NLP의 역사 Before Deep Learning 🡪 After Sequence to Sequence 🡪 After Sequence to Sequence with Attention 🡪 Era of Attention 🡪 Pretraining and Fine-tuning 18. Before Deep Learning 전형적인 NLP app 구조 - 여러 단계의 sub-module로 구성되어 복잡한 디자인을 구성 - 매우 무겁고, 복잡하여 구현 및 시스템 구성이 어려운 단점 - 각기 발생한 error가 중첩 및 가중되어 error propagation 19. Before sequence to sequence 20. After Sequence to Sequence with Attention 21. era of Attention 22. NLP Project Workflow : 문제 정의 🡪 데이터 수집 🡪 데이터 전처리 및 분석 🡪 알고리즘 적용(모델 설계) 🡪 평가 🡪 배포 23. 전처리 Workflow : 데이터(코퍼스) 수집 🡪 정제 🡪 레이블링(Optional) 🡪 Tokenization 🡪 Subword Segmentation(Optional) 🡪 Batchify 24. 말뭉치(Corpus)란? 자연어 처리를 위한 방대한 양의 데이터 모음을 코퍼스라고 한다. 언어 분석에 사용되는 실제 언어의 체계적 디지털 모음 둘 이상이면 Corpora(복수형) 25. 서비스 전체 Pipeline 데이터(코퍼스) 수집 – 구입, 외주, 크롤링을 활용한 수집  🡪 정제 – 학습 데이터와 같은 방식의 정제 수행, task에 따른 노이즈 제거, 인코딩 변환  🡪 레이블링(Optional) – 문장마다 또는 단어마다 labeling을 수행  🡪 Tokenization – 학습데이터와 같은 방식의 분절 수행, 형태소 분석기를 활용하여 분절 수행  🡪 Subword Segmentation(Optional) – 학습 데이터로부터 얻은 모델을 활용하여 똑 같은 불절 수행  🡪 Batchify – 학습 데이터로부터 얻은 사전에 따른 word to index mapping 🡪 Prediction – 모델에 넣고 추론 수행, 필요에 따라 search 수행(자연어 생성) 🡪 Detokenization(Optional) – 사람이 읽을 수 있는 형태로 변환 index to word, 분절 복원 26. DATA CRAWLING 무한한 양의 코퍼스 수집 가능 – 원하는 도메인 별로 수집 가능 27. DATA CLEANING – 정제 TASK에 따른 특성 – 풀고자 하는 문제의 특성에 따라 전처리 전략이 다르다 신중한 접근이 필요 ex 이모티콘 정규식을 활용한 정제 – 정규식을 활용하면 복잡한 규칙의 노이즈도 제거/치환 가능, 코딩 없이 단순히 텍스트 에디터로도 가능 Interactive 노이즈 제거 과정 규칙에 의해 노이즈를 제거하기 때문에, 노이즈를 전부 제거하는 것은 어려움 노이즈 확인 🡪 RegEx 구현 🡪 RegEx 적용(끝이 없는 과정임, 노력과 품질 사이의 trade-off, sweet spot을 찾아야 함) 28. REGULAR EXPRESSION “[]” – 2, 3, 4, 5, c, d, e 중의 character [2345cde] “-“ - ~사이 [2-5c-e] “^” – not 개념 “()” – x를 \1에 지정, yz를 \2에 지정 ([a-z])bc([a-z]) → \1\2 예제 실습 |
| 오후 | 1. REGULAR EXPRESSION 이어서 “|” = or “?” = x 또는 y가 나타남. 그리고 \1에 지정 “+” = x가 한 번 이상 나타남 “\*” = x가 나타나지 않을 수도, 반복될 수도 있음 “{}” = n번 반복 2. colab 실습 도구 🡪 설정 🡪 기타 🡪 코기 or 아기고양이 설정 1) 숫자 찾기 2) 숫자가 아닌 것 찾기 3) 이메일 주소를 찾기 4) URL 찾기 5) 문장에서 파일명 찾기 3. 자연어 노이즈 제거 1) 불완전한 문장으로 구성된 대화의 경우 – 한 문장씩 주고 받는 대화와 달리 메신저는 한 문장을 여러 번에 나눠 전송하거나 여러 문장을 한 번에 전송하는 경우가 있음 2) 문장의 길이가 너무 길거나 짧은 경우 – 의미가 없을 수도 있고, 대체로 사용 빈도가 높은 리액션에 해당하는 경우가 많아서 언어 모델을 왜곡시킬 우려가 있기 때문에 제외해 주는게 좋음 4. Labeling Token별로 sequence – token classification 한글 🡪 영어 5. label example  * Sentence 🡪 class TSV 형태의 하나의 파일, 문장과 라벨을 탭으로 분류함 * Sentence 🡪 Sentence 각 row가 대응되는 문장 쌍, 문장 대 문장을 탭으로 분류 두개 이상의 파일로 구성, 같은 순서의 row가 대응되는 문장 쌍  1. 꿀팁: 효율적인 레이블링 도구를 구성하자(ex. 엑셀) 2. Tokenization 두 개이상의 다른 token들의 결합으로 이루어진 단어를 쪼개어, vocabulary 숫자를 줄이고, 희소성(sparseness)를 낮추기 위함 영어는 공백 tokenization 해도 성능이 좀 나오는데, 한글은 그게 안되지롱 3. 형태소 분석 및 품사 태깅 형태소 분석 – 형태소를 비롯하여, 어근, 접두사, 접미사, 품사(POS, part of speech) 등 다양한 언어적 속성의 구조를 파악하는 것 품사 태깅 – 형태소의 뜻과 문맥을 고려하여 그것에 마크업을 하는 일 mecab이 빠름 |