|  |  |
| --- | --- |
| 교육 제목 |  |
| 교육 일시 |  |
| 교육 장소 |  |
| **교육 내용** | |
|  | 음성인식 (음성파형)  **자연어(Natural Language**(input도 언어 output도 언어)  -자연어 혹은 자연 언어는 사람들이 일상적으로 쓰는 언어를 인공적으로  만들어진 언어인 인공어와 구분하여 부르는 개념  **Natural Language Processing**  **-사람이 이해하는 자연어를 컴퓨터가 이해할 수 있는 값으로 바꾸는 과정(NLU)**  **-더 나아가 컴퓨터가 이해할 수 있는 값을 사람이 이해하도록 바꾸는 과정(NLG)**  **종류:**  Traditional NLP: 단어를 symbolic 데이터로 취급(분절)  여러 sub-module을 통해 전체 구성  NLP with Deep Learning: 단어를 continuous value로 변환  End-to-End 시스템 추구 , 임베디드    \*모든 딥러닝 은 Backpropagation (라이브러리가 대신 해 줌)을 해야 한다  **NLP장점**  -효율적인 Embedding(문맥을 이해)을 통한 성능 개선  ////단어 사이의 유사도를 반영한 (레이어) 벡터화///  -단어, 문장, context embedding  -End to End구성으로 인한 효율/성능 개선 입력 넣어서 학습 -> 바로 out put  -가볍고, 빠르다.  \*결국, 기계번역의 경우, 다른 분야에 비해 가장 먼저 성공적인 상용(seq2seq) |
|  | **NLP가 어려운 이유**  - Ambiguity (모호성): 동음이의어, 중의 어  -언어는 효율성을 극대화하는 방향으로 진화  =최대한 짧은 문장 내에 많은 정보를 담고자 한다.  -정보량이 낮은 내용(context)은 생략  -여기에서 모호함(ambiguity)이 발생  =>생략된 context를 인간은 효율적으로 채울 수 있지만, 기계는 이러한 task에 매우 취약하다\*한국어 =tokenization(분절화 필수)  - 교착어: 접사 추가에 따른 의미 파생, 유연한 단어 순서 규칙  - 단어 중의성으로 인한 문제 발생 사례  - 모호한 띄어쓰기 - 평서문과 의문문의 차이 부재  - 주어 부재 - 한자 기반의 언어  **역사**  -Before Deep Learning: 전형적인 NLP application 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  -After Sequence to Sequence  -After Sequence to Sequence with Attention  -Era of Attention  -Pretraining and Fine-tuning  전 처리Workflow    **Corpus(말뭉치)**  - 자연어처리를 위한 방대한 양의 데이터 모음을 코퍼스라고 한다.  - 언어 분석에 사용되는 실제 언어의 체계적 디지털 모음!  - 둘 이상의 코퍼스가 있으면 Corpora (복수표현)  - 포함된 언어 숫자에 따라  - Monolingual Corpus (단일 언어 코퍼스)  - Bi-lingual Corpus (이중 언어 코퍼스)  - Multilingual Corpus (다국어 코퍼스)  - Parallel Corpus : 대응되는 문장 쌍이 labeling되어 있는 형태  정규식을 활용한 정제 import re  •정규식(Regular expression)을 활용하면 복잡한 규칙의 노이즈도 제거/치환 가능  • 코딩 없이 단순히 텍스트 에디터(Sublime Text, VSCode등)도 가능  Interactive 노이즈 제거 과정(굉장히 많은 노력 필요)  • 규칙에 의해 노이즈를 제거하기 때문에, 노이즈 전부를 제거하는 것은 어려움  • 따라서, 반복적인 규칙 생성 및 적용 과정이 필요  • 끝이 없는 과정  • 노력과 품질사이의 trade-off  • Sweet spot을 찾아야함  Regular Expression  **[ ]**  • 2, 3, 4, 5, c, d, e 중의 character = [2345cde] = (2|3|4|5|c|d|e)  **[-]**  • 2, 3, 4, 5와 c, d, e 중의 character = [2-5c-e]  **[^]**  • 2, 3, 4, 5와 c, d, e를 제외한 모든 character = [^2-5c-e]  **( )**  • x를 \1에 지정, yz를 \2에 지정= (x)(yz)  **RegEx**  • 양 끝에 알파벳(소문자)으로 둘러싸인 ‘bc’를 제거하기 // • abcd • 0bc1 //  • 적용 예제 => ([a-z])bc([a-z]) → \1 = abcd →ad and 0bc1 →0bc1  | = x 또는 y가 나타남. 그리고 \1에 지정 • (x|y)  **? =** x가 0번 또는 1번 나타남 • x?  **\*** • x가 나타나지 않을 수도, 반복될 수도 있음 ,강력한 표현. 유의해서 사용, • x\*  **X /{n}, {n,}, {n,m}**  N 번 반복 ,N번 이상, N-M까지 반복  • = Any character , 매우 강력한 표현. 유의해서 사용해야 함  **^ $ =**  문장의 시작과 끝을 표시 • ^x$    **자연어 노이즈 제거**  -불완전한 문장으로 구성된 대화의 경우  -문장의 길이가 너무 길거나 짧은 경우  -채팅 데이터에서 문장 시간 간격이 너무 긴 경우  -바람직하지 않은 문장의 사용  **노이즈 유형**  (1) 문장부호: Hi, my name is john  문장 부호 양쪽에 공백을 추가하는 방법을 취하자  (2) 대소문자: First, open the first chapter  모든 단어를 소문자로 바꾸는 방법  (3) 특수문자: He is a ten-year-old boy  사용할 알파벳과 기호들을 정의해 이를 제외하고 모두 제거하자 |