인공지능기초응용 II

기말 과제

인공지능응용

K2025029 금동환

목차

1. 아래 한글코드체계에 대해 설명하시오

A. KS 완성형

국가 표준으로 2 바이트 고정폭의 문자 집합이다. 한글 사용빈도가 낮은 글자는 져있으므로 `뷁` 처럼 사전에 없는 글자는 입력,저장,검색이 모두 불가능하다.(11,172 중 2,350 은 빠져있음. 8,822 누락)

또한 부호화 규칙을 명시하지 않아, 동일 KS 완성형이라도 EUC, ISO-2022, 조합형 등 인코딩이 플랫폼마다 달라지는 구조적인 한계가 있다.

B. cp949, MS949

MicroSoft 에 의해서 개발되어 Windows 95 부터 채택된 인코딩 방식이다. 누락된 8,822 자를 확장 추가하여 현대 한글 전음절(11,172)을 모두 표현이 가능하다.

반면 확장된 영역은 국제표준(IANA, Unicode BMP) 와는 매칭되지 않아 비 Windows 계열 환경에서는 글자가 깨지거나 인코딩이 되지 않는 경우가 발생한다.

C. EUC-KR

KS 완성형 글자들을 UNIX 계열에서 쓰이는 규칙으로 1 바이트(ASCII) 와 2 바이트(한글,한자)로 부호화한 인코딩 방식이다. 덕분에 KS 환성형보다 호환성이 높지만 글자의 범위는 KS 완성형과 완전 동일 하므로 누락된 글자에 대한 문제가 해결되지 않는다.

1,2 바이트가 혼재되는 가변 구조이므로 문자열 처리 시 오프셋 계산 오류 가능성이 높다.

2. 유니코드와 UTF 인코딩에 대해 설명하시오.

A. 유니코드의 필요성, 코드부여 방식 등

유니코드 이전에는 각국의 문자집합에 부여된 바이트 값이 국가마다 달라서 발생하는 '문자깨짐' 문제와 한 응용프로그램에서 여러 문자 체계를 동시에 처리하지 할 수 없는 '혼용 불가' 문제가 있었다.

이에 따라 '모든 문자를 하나의 통일된 번호로 정의해야 한다' 는 요구가 생겼고, 유니코드 컨소시엄이 출범했다.

유니코드는 0x0000~0x10FFFF 를 17 개 영역으로 분할하여 문자를 배정한다.

B. UTF-8 인코딩 기법

코드 포인트를 1~4 가변길이로 변환하는 방식이다.

1~7 bit 영역은 '0xxxxxxx'로 포현하고

8~11bit 영역은 상위 5, 하위 6 bit 으로 쪼개어 '110xxxxx 10xxxxxx'로 표현한다.

12~16bit 영역(한글이 포함된 영역)은 4,6,6 bit 으로 쪼개어 '1110xxxx 10xxxxxx 10xxxxxx'로 표현한다.

17~21bit 영역은 3,6,6,6 bit 으로 쪼개어 '1110xxx 10xxxxxx 10xxxxxx 10xxxxxx'로 표현한다.

C. UTF-16 인코딩 기법

모든 문자를 2 바이트로 변환하는 방식이다.

BMP 영역(U+0000~U+FFFF)은 변환없이 2 바이트로 표현한다.

그 밖의 코드포인트는 서로게이트 영역에서 표현한다. 코드에서 0x10000 을 빼서 20bit 의 값을 각 상위, 하위로 나누어 상위는 '110110yyyyyyyyyy'로 하위는 '110111xxxxxxxxxx'로 포현한다.

3. 자연어처리의 기본 개념과 관련하여 아래 주제의 특징과 차이점을 설명하시오.

A. 형태소 분석과 품사 태깅

형태소 분석은 한 문장을 의미 최소 단위인 '형태소'로 분해하고, 굴절,접사,어간을 식별해 어형 정보를 복원하는 과정이다. 한국어처럼 교착성이 강한 언어에서는 '먹었습니다 → 먹/V + 었/EP + 습니다/EF'처럼 한 어절을 다수의 형태소로 쪼개야 문법,의미 처리가 가능하다. 품사 태깅은 이렇게 분할된 각 단위에 명사, 동사, 조사 등 문법 범주 레이블을 붙여 주는 단계이다.

형태소 분석이 구조를 형태소 단위로 분해, 품사 태깅은 각 분해된 단위에 기능적 라벨을 부여로 정리할 수 있다.

B. Word segmentation

공백이 없는 언어(중국어, 일본어 등)에서 어절 경계를 찾아 공백을 삽입하는 작업이다.

형태소 분석이 내부 형태 변화를 추적한다면, 워드 세그멘테이션은 어절 단위 경계를 먼저 확정해 이후 태깅,파싱의 전처리를 수행한다.

C. Subword 토크나이저

문자와 단어 사이 중간 단위(서브워드)를 학습한 뒤, 미등록 신어도 작은 조각으로 분해해 표현하는 방법이다. 이는 어휘 크기를 수만 개로 고정하면서 OOV 문제를 근본적으로 완화하고, 다국어 모델을 하나의 공유 어휘로 학습할 수 있게 한다.

4. 워드임베딩과 문서벡터 구성과 관련하여 물음에 답하시오.

A. TF-IDF 방식으로 문서 벡터를 구성하는 방법을 설명하시오.

말뭉치 전체에서 단어 사전을 만든 뒤, 개별 문서에서 각 단어가 등장한 횟수(TF)를 구하고, 동시에 그 단어가 전체 문서 가운데 몇 편에 나타났는지의 역수를 취한 IDF 값을 계산한다. 이어서 두 값을 곱한 TF–IDF(t,d)=tf(t,d)×idf(t)를 단어 수만큼 반복하면, 한 문서는 사전 크기와 정확히 같은 차원을 갖는 실수 벡터가 된다. 이 벡터는 0 이 대부분인 희소 구조이지만, 단어가 자주 쓰이면서도 특정 문서에만 특징적으로 등장할수록 큰 가중치가 부여되므로 문서 간 코사인 유사도나 분류 모델 입력값으로 손쉽게 활용할 수 있다

B. 워드임베딩이란 무엇인가?

각 단어를 의미 정보를 보존하는 수십~수백 차원의 밀집 벡터로 변환하는 절차를 말한다. 원-핫 벡터처럼 수십만 차원에 대부분 0 으로 채워진 표현은 메모리와 연산 비용이 크고 단어 간 의미 관계도 담지 못하는데, 임베딩은 이 문제를 해결한다. 동일하거나 유사한 의미의 단어들이 벡터 공간에서 가깝게 위치하여 “king-man+woman=queen” 같은 연산적 의미 추론이 가능해지기 때문에, 기계 번역,감성 분석 등 대다수 자연어 딥러닝 모델이 임베딩을 입력층으로 사용한다.

C. word2vec 과 fastText, BERT 의 차이점은

주변 단어를 예측하는 Skip-gram 또는 CBOW 를 통해 하나의 단어에 하나의 고정 벡터를 학습한다는 점에서 정적 임베딩의 대표 모델이다. 이에 비해 fastText 는 동일한 확률 모델 위에 단어를 문자 n-gram 단위로 쪼개 합산함으로써 사전에 없는 희귀어라도 내부 부분 문자열로 벡터를 합성할 수 있도록 확장했다. 한편 BERT 는 Transformer 인코더를 이용해 문장 안의 토큰 일부를 가린 뒤 좌우 문맥을 동시에 활용해 이를 복원하도록 학습한다. 그 결과 같은 단어라도 문맥이 바뀌면 서로 다른 벡터를 출력하는 동적 임베딩을 생성하며, 다의어 구별 능력과 하위 태스크 성능 등 에서 높은 성능을 보여준다.

5. 언어 모델과 관련하여 물음에 답하시오.

A. 음성인식 후처리에서 언어모델이 등장하게 된 배경을 설명하시오.

동음어,유사 발음이 많은 한국어,영어 환경에서는 발음은 비슷하되 의미가 전혀 다른("값"/"밥", "있다"/ "있나") 후보가 빈번히 동률로 출력된다. 또한 잡음,발음 변이로 일부 음소가 왜곡되면 어쿠스틱 확률만으로는 올바른 단어 순서를 복원하기 어렵다. 이러한 발음-동일·음향 불확실성을 해결하기 위해, 시스템은 인식 결과를 생성한 뒤 얼마나 자연스러운가를 평가하는 통계적 언어모델을 결합하게 되었다. 즉, 언어모델은 음향 계층이 제시한 후보들 중 문맥적으로 가장 개연성이 높은 어절 열을 재선택해 단어 오류율을 크게 낮추는 역할을 한다

B. N-gram 언어모델에 의해 문장생성 확률을 계산하는 방법은

문장 $W = w_1 \cdots w_T$ 의 정확한 확률은 체인 룰에 따라
$P(W)=\prod_{t=1}^{T} P(w_t \mid w_{1:t-1})$ 로 정의된다.
그러나 전체 앞부분을 모두 고려하면 추정 불가능하므로 n-gram 모델은 k 차(보통 k = 2, 3) 마르코프 가정을 두어
$P(w_t \mid w_{1:t-1}) \approx P(w_t \mid w_{t-(n-1)} \ldots w_{t-1})$
로 단순화한다.

예를 들어 trigram(n = 3)이라면 문장 확률은
$P(W)=\prod_{t=1}^{T} P(w_t \mid w_{t-2}, w_{t-1})$
이며, 각 조건부 확률은 학습 코퍼스에서 등장 빈도로 근사한다:
$P(w_i \mid w_{i-2}, w_{i-1}) = \text{count}(w_{i-2}, w_{i-1}, w_i) / \text{count}(w_{i-2}, w_{i-1})$.

C. Smoothing 기법의 필요성 및 방법을 설명하시오.

실제 말뭉치는 유한하므로 학습 코퍼스에 등장하지 않은 n-gram 은 빈도가 0 이 되고, 위 공식을 그대로 쓰면 미관측 조합의 확률이 0, 즉 로그 확률 $-\infty$로 계산되어 문장 전체 확률을 붕괴시킨다. 이러한 **희소성 문제**를 해결하려면 관측 빈도가 0 인 항목에도 소량의 확률 질량을 재분배해야 한다. 이를 '스무딩'이라고 한다.

스무딩은 불가피한 데이터 희소성을 완화해, n-gram 언어모델이 실제 음성 후처리·자동 완성 등에서 안정적으로 작동하도록 만드는 핵심 절차라 할 수 있다.

내용 중 수식과 자세히 알지 못하는 부분에 대하여 ChatGPT 답변을 참고하여 작성했습니다.