한국어의 특징과 한글 코드 문제

국민대학교 인공지능학부
강승식

한국어, 영어, 일본어, 중국어의 차이점

- 문자
  - 영어: 알파벳 26자
  - 한글
    - 자음과 모음
    - 초성/중성/종성 : 19 x 21 x (27+1)
  - 일본어
    - 히라카나
    - 가다카나
    - 한자
  - 중국어
    - 한자

단어와 문장

- 어휘론
  - 교착어(agglutinative language): 조사/어미(기능어)
  - 굴절어(inflectional language): 단어의 변형

- 구문론(문장의 구조)
  - 구조적 언어(configurational language)
    - SVO (주어-술어-목적어)
  - 부분 자유 어순(partially free word-order)
    - SOV (주어-목적어-술어)

각 언어의 차이점

- 조어 방식 (전문용어 등 신조어)
  - 영어
  - 한국어
  - 중국어

- 어휘의 개수

- 주어 생략 문제

  “아침에 일어나서, 세수를 하고, 밥을 먹고, 친구 만나러 갔다”

자연어처리의 관점

- 형태소 분석, 품사 태깅
  - study(명사, 동사), hard(형용사, 부사)

- 구문 분석
  - 구구조 파싱
  - 의존구조 파싱

- 의미 분석
  - WSD(word sense disambiguation): ‘hard disc’

띄어쓰기 차이점

[참고] 띄어쓰기를 하는 이유는 무엇인가?

- 영어
- 한국어
- 중국어/일본어

한글코드: 유니코드와 KS완성형

KS 완성형, cp949, EUC-KR

- 한글/한자 정의 영역

  [A1-FE][A1-FE] : 94 x 94 (2350 + 4888)

  가(0xB0A1) ~ 힝(0xC8FE) : 2,350자

- cp949
  - 11,172 - 2,350 = 8,822자

- EUC-KR
  - 아스키 코드 + KS완성형(한글/한자/2바이트부호)

KS 완성형, cp949, EUC-KR: 문제점

- 소팅 문제: 2,350자와 8,822자 혼용된 경우
  - 똠/폞/웍/슒/힇 등

- 텍스트 파일에서 다국어 표현 문제 -- [A1-FE][A1-FE]
  - 한글 + 일본어
  - 한글 + 중국어

- 운영체제 문제
  - 한글 OS, 일본어 OS, 중국어 OS

유니코드

- 모든 문자를 “정수로 mapping”
  - 21 bit

- BMP 영역: 1~16 bit
  - 65,536 x 1
  - ‘가’(U+AC00) ~ ‘힣’(U+D7A3)

- SMP 영역: 17~21 bit
  - 65,536 x 16

UTF 인코딩

- UTF-8
  - 1~4 바이트 가변길이
- UTF-16 LE/BE
  - 2 또는 4 바이트
  - LE: 00 AC
  - BE: AC 00
- UTF-32 LE/BE
  - 4 바이트
  - LE: A3 D7 00 00
  - BE: 00 00 D7 A3

BOM(Byte Order Mark): U+FEFF
UTF-8 : EF BB BF
UTF-16 (BE): FE FF
UTF-16 (LE): FF FE
UTF-32 (BE): 00 00 FE FF
UTF-32 (LE): FF FE 00 00

UTF-8: 1~4 바이트 가변길이

- 1 바이트: 1~7 비트 영역
  `0xxxxxxx`
- 2 바이트: 8~11 비트 영역
  `110xxxxx 10xxxxxx`
- 3 바이트: 12~16 비트 영역
  `1110xxxx 10xxxxxx 10xxxxxx`
- 4 바이트: 17~21 비트 영역
  `11110xxx 10xxxxxx 10xxxxxx 10xxxxxx`

예) '가'(U+AC00), '힣'(U+D7A3)

'가'(U+AC00): 1010 1100 0000 0000
'힣'(U+D7A3): 1101 0111 1010 0011

UTF-16: 2 or 4 바이트

- 2 바이트: BMP 영역

- 4 바이트: SMP 영역

  U' = yyyyyyyyyyxxxxxxxxxx // U - 0x10000

  ```
  110110yyyyyyyyyy   // 0xD800 + yyyyyyyyyy
  110111xxxxxxxxxx   // 0xDC00 + xxxxxxxxxx
  ```

SMP 영역(100만자)
  1,024 x 1,024

high surrogates
  0xD800–0xDBFF
low surrogates
  0xDC00–0xDFFF

코드변환: 윈도 메모장

- 파일 - ”다른 이름으로 저장” - 인코딩
  - ANSI : cp949
  - UTF-8
  - UTF-8 BOM
  - UTF-16 LE
  - UTF-16 BE