2 한글 맞춤법 검사기의 원리와 개발 현황

이 기 황*

1 머리말

LaTeX은 중독성이 강한 소프트웨어이다. 한 번 LaTeX에 맛을 들이면 가능한 모든 문서를 LaTeX으로 작성하려고 시도하게 되고, 워드프로세서는 멀리하게 되는 경향이 있다. 마음에 드는 쓸만한 편집기를 이용해서 LaTeX 문서를 작성하는 데에 익숙해지면 워드프로세서의 필요성을 느낄 때가 별로 없는 것이 사실이다. 그런데 많은 한글 LaTeX 사용자들이 부러워하는 워드프로세서의 기능 가운데 하나를 들자면 바로 '맞춤법 검사기'이다. 특히 규모가 큰 문서를 편집하는 경우에 '맞춤법 검사기'는 매우 유용한 도구이다. 그러나 현재 '한글 맞춤법 검사기'는 모두 워드프로세서 등의 소프트웨어에 내장되어 있기 때문에 LaTeX 사용자가 이용하기는 쉽지 않다.

이 글에서는 간략하게 한글 맞춤법 검사기의 원리와 개발 현황을 알아 보고, 아직 한계가 많으나 윈도 환경에서 LaTeX 사용자들이 이용할 수 있는 한글 맞춤법 검사기를 소개한다. 또한 자유 소프트웨어 한글 맞춤법 검사기 개발 가능성에 대한 글쓴이의 짧은 의견도 제시하고자 한다.

*k.lee at ed.ac.uk

2 한글 맞춤법

'맞춤법'이란 '어떤 문자로써 한 언어를 표기하는 규칙'으로 정서법, 정자법, 혹은 철자법이라고도 한다(표준국어대사전). 따라서 '한글 맞춤법'은 '한글로 한국어를 표기하는 규칙'이다.[1] 현재 사용되고 있는 한글 맞춤법은 문교부 고시 제 88-1 호로 1988 년 1 월 19 일에 공포되었으며, 이후 오늘에 이르기까지 몇 차례의 개정을 거친 것이다.[2]

'한글 맞춤법'에서는 먼저 표준어를 소리대로 적되, 어법에 맞도록 하고', '문장의 각 단어는 띄어 씀을 원칙으로 한다'는 총칙을 제시하고, 현대 한국어를 표기할 때에 필요한 한글 자모를 규정한다. 이어서 된소리 표기법(국수 ○ 국쑤 ×) 등의 소리에 관한 규정, 체언과 조사(집에 ○ 지베 ×), 어간과 어미(많고 ○ 만코 ×)를 구별하여 적는 형태에 관한 규정, 그리고 띄어쓰기와 그 밖의 것에 관한 규정이 실려 있다. 부록으로는 문장 부호 사용법이 붙어 있다.

3 맞춤법 검사기

'맞춤법 검사기'는 컴퓨터에 저장된 문서의 단어들이 맞춤법 규정에 따라 옳게 표기가 되었는가를 자동으로 검사하는 소프트웨어이다. 요즘 널리 사용되는 맞춤법 검사기들은 표기의 맞고 틀림을 판단하는 것뿐만 아니라 잘못된 표기에 대해서는 교정 후보를 제시해 주는 '맞춤법 교정기'의 기능을 포함하고 있다. 맞춤법 검사의 대상은 한 개 또는 여러 개의 문서이지만, 실제로는 하나의 낱말, 우리말의 경우에는 띄어쓰기의 단위인 '어절'이 맞춤법 검사기의 입력인 것이 보통이다. 따라서 발음이나 표기가 비슷한 낱말({그러므로, 그럼으로}, {로서, 로써})에 의해 발생하는 '유사어 오류' 등 대상 어절이 쓰인 문맥 혹은 문장 단위의 정보가 필요한 오류의 검사와 교정은 불가능하다. 이와 같은 오류는 '문법 검사기' 또는 '문체 검사기'의 도움을 받아야 교정이 가능하다.

4 맞춤법 검사기의 원리

맞춤법 검사기의 핵심 기능은 검사 대상 문서에 포함된 각각의 어절들 가운데에서 표기가 잘못된 어절들을 거르는 기능이다. 이 기능의 구현을 위해 가장 생각하기 쉬운 방법은 맞춤법에 맞도록 표기된 어절들의 목록(사전)을 저장해 두고 검사 대상 문서에 포함된 어절들 가운데 이 목록에서 발견되지 않는 어절은 맞춤법에 어긋나게 표기된 것으로 간주하여 걸러내는 방법이다. 세부적으로는 약간의 차이가 있으나 영어, 불어 등 낱말 자체가 형태를 바꾸는 유형의 언어

[1] 한국어를 로만 알파벳으로 표기하는 규칙은 '로마자 표기법'이며, 외국어와 외래어를 한글로 표기하는 규칙은 '외래어 표기법'이다.

[2] 한글 맞춤법 등이 포함된 '어문 규정'은 국립국어연구원의 웹사이트 http://www.korean.go.kr에서 볼 수 있다.

를 위한 맞춤법 검사기는 이와 같은 방법에 기초하고 있다.[3]

그러나 한국어와 같이 문법 요소가 연속적으로 추가될 수 있는 언어에서는 이와 같은 방법의 적용이 현실적으로 불가능하다. 장석배 (1999)에 따르면 신문, 소설 등 다양한 장르의 글이 포함된 4,180만 어절 규모의 한국어 텍스트[4]에 나타난 어절 유형의 갯수는 3,283,079 개에 이른다. 말뭉치의 규모가 증가하면 어절 유형의 갯수는 더 많아질 것이다. 물론 이 어절들이 모두 맞춤법 규정에 맞게 표기된 어절은 아니며, 모든 어절들이 골고루 비슷한 빈도로 사용되는 것도 아니다. 어찌되었건 맞춤법에 맞게 표기된 한국어의 모든 어절을 모으는 일은 간단한 일이 아니며, 모두 모은다고 해도 이를 효율적으로 이용하기는 매우 힘들다.

여기까지 오면 한글 맞춤법 검사기의 구현은 영영 불가능해 보인다. 그런데 다행인 것은 매우 복잡하고 많은 예외가 있기는 하지만, 언어 현상에는 규칙성이 있다는 사실이다. 그러한 차원에서 우리말 어절들을 잘 살펴 보면, 어절은 어절보다 작은 단위들이 일정한 규칙에 따라 결합하여 이루어짐을 알 수 있다. 예를 들어, 어절 '먹었습니다'는 '먹' + '었' + '습니다'로, '학교에서조차도'는 '학교' + '에서' + '조차' + '도'로 구성되어 있다. 이 예에서와 같이 어절의 구성 요소인 언어 단위를 '형태소'라고 부르며, 어절을 형태소들로 분절하는 절차를 '형태소 분석'이라고 한다.[5] 결국 한글 맞춤법 검사기를 구현하기 위해서는 먼저 '형태소 분석기'를 구현해야 한다는 결론에 이르게 된다. 형태소 분석기를 구현하는 방법은 여러 가지나 있으나, 공통적인 요소는 체언, 용언, 어미, 조사 등으로 구성된 품질이 좋은 사전과 형태소들간의 접속 가능 관계가 표현된 정밀한 접속정보표이다.[6] 형태소 분석기로 하나의 어절이 입력되면 먼저 이 어절을 형태소들로 분절하고 그 분절 결과가 적합한 것인지 접속정보표를 이용하여 확인한다. 형태소 분절이 앞의 예처럼 비교적 단순한 경우도 있지만 불규칙 활용, 축약 현상 등으로 인해 제법 복잡한 경우도 많다. 만일 입력 어절을 성공적으로 형태소 분석할 수 없을 경우 이 어절은 맞춤법이 틀린 어절로 간주하게 된다.

맞춤법에 맞지 않게 표기된 것으로 판단되어 걸러진 어절은 띄어쓰기 오류 검사 및 교정, 맞춤법 교정 단계를 거치게 된다. 띄어쓰기 오류 검사 및 교정에도 여러 가지 방법이 있는데, 대체로 조사/어미 등 문법 형태소의 음절 정보나 대규모의 말뭉치에서 얻어진 음절간의 결합 정보 등을 이용한다. 맞춤법 교정 후보 제시에는 발음, 자판 배치 등을 참조하거나 통계 정보를 이용한다.[7]

[3] Unix/Linux에서 널리 쓰이는 ispell/aspell도 이와 같은 원리로 작동한다.

[4] 이를 '말뭉치(corpus)'라 부른다.

[5] 엄밀히 말하면 '형태소 분석(morphological analysis)'이라는 용어는 '형태론적 분석', 혹은 분석 대상을 강조하여 '어절 분석'이라고 하는 것이 옳을 것이다.

[6] '품질 좋은 사전'이 어떤 사전인지 정의하기는 쉽지 않다. 맞춤법 검사를 위한 형태소 분석의 경우 표제어가 수가 너무 많으면 과분석을 유발한 가능성이 있으므로 반드시 좋다고만 할 수는 없다.

[7] 형태소 분석기 및 맞춤법 검사기에 대해서 더 자세히 알고 싶은 독자는 강승식 (2002)을 참조하기 바란다.

5 한글 맞춤법 검사기의 개발 현황

우리 나라에서의 형태소 분석기 및 맞춤법 검사기에 관한 연구 및 개발은 1980년대 중반부터 시작되었으며, 1992년 상용 워드프로세서에 한글 맞춤법 검사기가 최초로 포함되었다. 자동 형태소 분석 연구 초기에는 분석 방법론 및 알고리즘 자체에 관한 논의가 주를 이루었으며 최근에는 고속 고효율의 형태소 분석과 형태론적 중의성 해결[8]에 관한 연구가 집중적으로 이루어지고 있다. 띄어쓰기 오류 검사와 교정에 관해서는 비교적 최근에 연구가 이루어졌으며 맞춤법 오류 교정에 관한 연구도 일부 진행이 되었다.[9]

형태소 분석기/맞춤법 검사기에 관한 연구와 개발은 주로 대학을 비롯한 연구 기관 및 관련 기업에서 이루어지고 있으며, 공개적으로 개발된 것은 알려져 있지 않다. 사용상 몇 가지 제약과 불편이 따르기는 하나 공개된 형태소 분석기 및 맞춤법 검사기로는 다음과 같은 것들이 있다.[10]

- MoA
  - 기능: 형태소 분석
  - 위치: `ftp://ftp.kreonet.ne.kr/pub/hangul/cair-archive/nlp/MoA/`
  - 배포 형태: 소스 및 간단한 설명서
  - 라이센스: GPL
- KTS
  - 기능: 형태소 분석 및 중의성 해결
  - 위치: `ftp://ftp.kreonet.ne.kr/pub/hangul/cair-archive/nlp/kortaggers/kts/`
  - 배포 형태: 소스 및 간단한 설명서
  - 라이센스: 밝혀져 있지 않음.
- POSTAG
  - 기능: 형태소 분석 및 중의성 해결
  - 위치: `http://nlp.postech.ac.kr/DownLoad/k_api.html`
  - 배포 형태: 리눅스 바이너리(별도의 연구용 소스 라이센스 체결 가능)
  - 라이센스: 비상업적 연구용.
- HAM
  - 기능: 형태소 분석 및 맞춤법 검사

[8] '고구려 → 고구려(명사) | 고(동사) + 구려(어미)'에서와 같이 하나의 어절이 두 개 이상의 형태소 분석 결과를 가지는 것을 형태론적 중의성이라고 한다.

[9] 이들 연구 결과는 주로 관련 학회 학술지 및 학술회의를 통하여 발표되었다. 보다 일반적인 경로를 통하여 발표된 자료로는 금장철 (1996)과 한국어정보처리연구소 (1996)가 있다.

[10] 여기서 보인 것들 외에도 형태소 분석기가 포함된 용례 추출기 등의 도구가 세종 계획 웹사이트(`http://www.sejong.or.kr`)을 통해 배포되고 있으며, 웹사이트를 통한 형태소 분석기 데모도 상당수 존재한다.

  - 위치: http://nlp.kookmin.ac.kr/HAM/kor/download.html
  - 배포 형태: 리눅스/윈도 바이너리 및 라이브러리(형태소 분석기), 윈도 바이너리 및 라이브러리(맞춤법 검사기)
  - 라이센스: 비영리 연구 및 실험용(배포 제한 및 실행 기간 제한)
- MACH
  - 기능: 형태소 분석
  - 위치: http://cs.sungshin.ac.kr/~shim/ (현재 다운로드 안 됨.)
  - 배포 형태: 리눅스/솔라리스/윈도 바이너리
  - 라이센스: 연구 또는 실험 목적(배포 제한)
- 지능형 형태소 분석기
  - 기능: 형태소 분석 및 중의성 해결
  - 위치: http://www.sejong.or.kr 자료실 - 국어 정보 처리 프로그램
  - 배포 형태: 윈도 바이너리
  - 라이센스: 비상업적 연구용

위에서 보인 바와 같이 비록 사용상 제한이 따르기는 하나, 맞춤법 검사기로서 공개된 것은 HAM에 포함된 맞춤법 검사기가 유일하다. 형태소 분석기들은 대부분 비상업적 연구 또는 실험 목적을 위해 바이너리 형태로만 공개되고 있으며, MoA만이 라이센스를 GPL로 명시하고 있다.

6 HAM-SPELL 및 SPELLX 이용하기

마이크로소프트 윈도에서 동작하는 공개 맞춤법 검사기인 HAM-SPELL[11]을 LaTeX문서 작업에 이용할 수 있다.

먼저 맞춤법 검사기를 위에서 보인 위치에서 내려 받아서 적당한 디렉토리에 압축을 푼다. 이 디렉토리에는 맞춤법 검사기의 실행 파일인 spell.exe 및 라이브러리 파일이 있으며, 사전 및 설정 파일, 그리고 이 프로그램의 소스가 포함된 두 대의 하위 디렉토리가 있다.

명령행 창에서 맞춤법 검사기를 실행하면 [그림 2.1]과 같이 간단한 사용법 안내가 표시된다. 현재 프로그램 옵션들은 제대로 동작하지 않으나 하위 디렉토리 hdic에 들어 있는 ham2000.ini 파일을 통해 설정 가능하다. 사용법은 매우 간단하여 입력 텍스트 파일을 spell.exe의 인자로 지정하면 맞춤법 검사 결과가 화면에 표시된다. 출력 파일을 지정하면 결과가 파일로 저장된다.

다음은 이 글의 초고 파일에 대한 맞춤법 검사 결과의 일부이다.

[11] 이 맞춤법 검사기는 한국어 분석 모듈 HAM의 일부로 제공되므로, HAM-SPELL이라고 부르기로 하겠다.

```
C:\bin\ham>spell
usage: spell [-options] [input.txt] [output.txt]

    no options & I/O files --> default options applied
        -c: get all spell-corrected candidates
        -i: maximum candidates are specified: 'i' is one of 0-9
        -v: split verb + '아/어' + xverb + Eomi
        -n: compound noun --> noun + noun + ... + noun
        -a: all ascii-included words are regarded as correct
        -x: don't blank-insertion check bet'n 2 words
        -w: echo-back input word itself

    Options may be combined like -c2, -wcv, or -nxvc3
    WITH OPTION & NO I/O FILES SPECIFIED --> INTERACTIVE TESTING
-------------------------------------------------------------------
(c) 1993-2001 Kookmin Univ. Kang Seung-Shik, Tel.(+82-2)910-4800
    Email: sskang@kookmin.ac.kr, http://nlp.kookmin.ac.kr/
-------------------------------------------------------------------
```

그림 2.1: HAM-SPELL의 사용법 표시 화면

```
...
이기황
    No candidates found!
...
\documentclass[twocolumn]{article}
    No candidates found!
...
워드프로세서는
    No candidates found!
워드프로세서의
    No candidates found!
워드프로세서의
    No candidates found!
워드프로세서
    No candidates found!
...
국쑤
    [1] 국수 (13)
    [2] 국부 (45)
    [3] 국무 (56)
지베
    [1] 집에 (4)
    [2] 지배 (6)
    [3] 지게 (34)
```

```
...
만코
    [1] 만조 (23)
    [2] 만보 (27)
    [3] 만도 (29)
...
맞춤법
    [1] 맞춤법 (18)
...
```

위의 결과를 살펴 보면, 글쓴이의 이름인 '이기황', '워드프로세서'는 사전에 등록되지 않아서 맞춤법 검사에서 걸러졌는데 교정 후보는 제시되지 못하고 있다. 이 글을 작성하는 과정에서 의도적으로 맞춤법에 어긋나게 표기된 '국쑤', '지베', '만코'의 세 어절에 대해서는 모두 교정 후보가 제시되었는데, '만코'에 대한 교정 후보 제시는 부정확하다. 글쓴이의 오타로 발생한 오류인 '맞춤범'에 대해서는 올바른 교정 후보가 제시되었다. 또한 LaTeX 명령어도 오류 어절로 판단된 것을 볼 수 있다.[12]

HAM-SPELL의 맞춤법 검사 기능은 완전하지는 않으나 상당히 유용하다. 미등록어로 인해 발생하는 맞춤법 오류는 `hdic` 디렉토리에 있는 `ham-usr.dic`을 편집하면 해결할 수 있다. 예를 들어 '이기황'을 사전에 등록하려면 편집기로 `ham-usr.dic`을 열고 가나다순 정렬을 유지하면서 '이것 P'와 '이때 N' 사이에 '이기황 N'을 추가하면 된다. 'N'은 추가하는 낱말의 품사를 표시하는 것으로 여기서는 '명사'를 의미한다.

HAM-SPELL의 맞춤법 검사 기능은 완전하지는 않으나 상당히 유용하다. 그러나 아직 HAM-SPELL은 ispell/aspell에서와 같은 대화형 맞춤법 검사를 지원하지 않으며, 설치된 디렉토리에서만 맞춤법 검사를 실행할 수 있도록 되어 있다. 글쓴이는 맞춤법 검사기와 함께 제공되는 소스를 일부 수정하여, 설정 파일인 `ham2000.ini`의 위치를 환경 변수로 지정하고 사전의 위치를 이 파일에서 설정한 다음, 실행 파일의 경로만 PATH 변수에 추가되면 어느 위치에서나 맞춤법 검사를 실행할 수 있는 SPELLX를 만들었다. 이에 관해서는 KTUG Faq SpellX 페이지[13]를 참조하기 바란다.

7 자유 소프트웨어 한글 맞춤법 검사기의 개발

LaTeX 사용자들에게 있어서 HAM-SPELL은 가뭄의 단비와 같은 존재이다. 그러나 HAM-SPELL은 일부 소스와 라이브러가 제공되기는 하나 배포 및 실행 기간에 제한이 있다. 또한 맞춤법 검사 기능의 소스가 공개되지 않아 소프트웨어의 확장 및 수정이 불가능하다. 따라서 현재 절실히 요청되는 것은 자유로운 수정과 확장, 배포가 가능한 자유 소프트웨어 한글 맞춤

[12] detex나 untex 등을 이용하여 LaTeX 명령어를 제거한 다음 맞춤법 검사를 실행할 수도 있다.

[13] `http://faq.ktug.or.kr/mywiki/SpellX`

법 검사기의 개발이다.

이 일은 쉬운 일은 아니나 전혀 불가능한 일도 아니다. 앞서 밝혔듯이 맞춤법 검사기의 핵심인 형태소 분석기의 관한 연구는 이미 상당한 수준에 이르렀으며, 참조할 만한 연구 결과가 풍부하다. 노력이 집중되어야 할 부분은 형태소 분석기를 위한 형태소 사전과 상대적으로 연구가 미미한 교정 후보 제시 기법의 개발이다. 또한 맞춤법 검사기를 편리하게 사용하기 위한 사용자 인터페이스의 개발도 중요한 과제이다.

글쓴이가 제안하는 자유 소프트웨어 한글 맞춤법 검사기의 개발 전략은 대체로 다음과 같다.

1. 철저한 모듈화
   구성이 복잡한 소프트웨어, 여러 명의 개발자가 참여하는 프로젝트의 경우에는 당연히 요구되는 사항이다. 서로 연관되어 있지만 분리 가능한 여러 기능 단위로 구성된 맞춤법 검사기에 있어서 모듈화는 특히 중요하다. 예를 들어, 사용자 인터페이스는 비교적 쉽게 모듈화가 가능하며, HAM-SPELL에서 제공되는 라이브러리를 이용하여 사용이 편리한 맞춤법 검사기의 제작 또한 가능할 것으로 보인다.

2. 두 단계 패러다임
   강승식 (2002)에서 강조하고 있는 것으로, 형태소 분석 등의 복잡한 언어 현상을 다루는 자연 언어 처리 분야에서 큰 의미가 있다. 요약하면 적용 범위가 넓고 해결 방법이 명확한 비교적 핵심적인 기능을 먼저 구현한 후에 확장 혹은 부가 기능을 구현하는 방법론이다. 이에 실현을 위해서는 앞서 언급한 모듈화와 함께 단순화, 예외 처리 등에 기법이 뒤따라야 한다. 또한 소프트웨어 원형의 개발은 Python과 같이 쓰기 쉽고 확장 기능이 우수한 언어를 이용하면서 차차 C/C++ 등의 언어로 이식하는 것이 유리할 것이다.

3. 기존 자원/연구 결과의 효율적 이용
   저작권 및 특허에 문제가 되지 않는 한도 내에서 기존 자원과 연구 결과를 철저히 이용해야 한다. 특히 형태소 사전의 경우 기존 사전 자료, 빈도 조사 결과 등을 참조하여 제작하는 것이 절대적으로 중요하다. 형태소 분석 및 맞춤법 검사 기법 및 알고리즘도 학계에 발표된 연구 결과를 최대한 참조해야 할 것이다. 기존에 개발된 시스템은 새로 개발되는 시스템의 성능 평가에도 매우 유용할 것이다.

4. 사용자 피드팩과 유지 보수
   자유 소프트웨어 개발에서 가장 중요한 요소이다. 특히 맞춤법 검사기의 기능 향상을 위해서는 실제 사용자들의 피드백과 이의 효율적인 반영이 필수적이다.

8 맺는말

이 글에서는 LaTeX 사용자들에게 절실한 한글 맞춤법 검사의 원리와 개발 현황에 대해서 간략히 알아 보고, 현재 이용 가능한 HAM-SPELL의 사용법을 보였다. 또한 공개 소프트웨어 한글 맞춤법 검사기의 개발 가능성 및 전략에 관한 글쓴이의 의견을 제시하였다.

이 글이 많은 LaTeX 사용자들에게 조금이나마 보탬이 되기를 바라며, 한글 맞춤법 검사기에 관한 다양한 논의가 KTUG를 통하여 이루어지기를 기대한다.

9 참고 문헌

강승식. 2002. 한국어 형태소 분석과 정보 검색. 홍릉과학출판사.

금장철. 1996. “형태소 분석기”. 마이크로소프트웨어 .

장석배. 1999. “말뭉치 규모와 어절 유형 증가간의 상관성에 관한 연구”. 서상규 편, 언어 정보의 탐구, 1. 연세대학교 언어정보개발연구원.

한국어정보처리연구소. 1996. 우리말 으뜸꼴 자동 인식. 도서출판 골드.