(제10회 한글 및 한국어 정보처리 학술대회)

한국어 정보처리의 문제점 및 방법론 고찰

강 승 식

한성대학교 정보전산학부
136-792 서울특별시 성북구 삼선동2가 389
kang@ham.hansung.ac.kr

Overview of Problems and Methodologies for Korean Language Processing

Kang, Seung-Shik
School of Information and Computer Engineering
Hansung University

요약

자연언어 처리 시스템은 시제품 개발이 비교적 용이한 반면에 이를 실용적인 시스템으로 발전시키는데 많은 어려움이 있다. 본 논문에서는 형태소 분석기와 구문분석기, 기계번역 시스템 등을 개발할 때 발생하는 문제점과 방법론을 고찰하고, 실용적인 시스템을 개발하기 위한 효율적인 방법으로 2-step 패러다임과 이를 실현하는 방안으로 기능별 모듈화에 의한 divide and conquer 기법, 단순화 기법, 예외처리 기법 등을 활용하는 방법을 제안한다.

1. 서론

자연언어 처리에 대한 연구는 1940년대 후반부터 영어와 러시아어, 영어와 불어 등 기계번역을 중심으로 시작되었다. 그런데 10여년 동안의 기계번역 연구는 1964년 ALPAC 보고서에서 '실현 가능성' 문제가 지적된 것을 기점으로 한동안 침체기를 맞이하기도 하였다[1].

그러다가 1970년대 말부터 철자 검사기가 활용되기 시작하고 정보검색 시스템에서 대용량 정보자료의 색인 문제를 해결하는 문제가 대두되면서 자연언어 처리는 다시 주목을 받기 시작하였다. 이와같이 소프트웨어 산업뿐만 아니라 모든 연구-개발 분야는 당시의 사회적 요구를 일부나마 충족시키면서 지속적인 발전이 가능하다.

그런데 기계번역을 비롯한 자연언어처리 시스템은 그 속성상 시제품 개발이 비교적 용이한 반면에 실용적인 시스템으로 발전하여 응용 분야의 요구 사항을 충족시키는데 많은 어려움이 있다. 따라서 한국어 정보처리 분야가 지속적으로 발전하려면 이 문제를 어떻게 극복하느냐 하는 문제가 가장 중요한 요소이다.

국내에서는 1980년대 중반 영-한, 일-한 기계번역을 중심으로 연구가 시작되어 일부 상용 시스템이 판매되고 있으나, 사용자들의 요구를 충분히 만족시키지 못한 점이 지적되고 있다[2]. 한국어 정보처리는 1980년대 후반부터 형태소 분석, 구문분석에 대한 연구가 시작되어 철자 검사기와 자동색인 등에 활용되고 있다[3]. 계속해서 실용적인 구문분석기를 개발하는 연구가 진행 중이며, 다양한 응용 분야의 요구 사항을 만족시킬만한 수준으로 발전하기 위한 단계에 와 있다.

형태소 분석이나 구문분석 등 한국어 정보처리 기술은 기계번역 등 일부 응용 분야에서 바로 활용되기도 하지만, 문법검사(grammar check) 라든지 문서 요약(text abstraction), 자연언어 인터페이스 등 다양한 응용 분야에서는 요소 기술로서 활용된다. 따라서 한국어 분석 기술이 요구되는 다양한 응용 분야의 요구 사항을 만족시킬 수 있도록 지원하는 요소 기술이 매우 중요하며, 이는 한국어 정보처리 분야가 지속적으로 발전하는데 많은 영향을 미치게 된다.

특히, 기계번역 기술을 이용한 자동통역, 운영체제 등 광범위하게 활용 가능한 자연언어 인터페이스, 키보드가 필요없는 컴퓨터, 필기체 인식 컴퓨터, 인간 수준의 고품질 음성 합성 및 음성 인식, 음성/화상 정보검색, 지식처리 분야 등 차세대 소프트웨어 기술로서 유망한 많은 소프트웨어 분야에서 자연언어 처리 기술이 필수적으로 요구될 전망이다. 이러한 다양한 분야에서 한국어 정보처리 기술이 활용되려면 한국어 분석 기술이 한 단계 더 높은 수준으로 발

전되어야 하며, 그러기 위해서는 현재 한국어 분석 기술의 현황과 문제점을 고찰하고 이를 극복하기 위한 방안이 마련되어야 한다.

2. 한국어 정보처리 현황

자연언어 처리 분야의 핵심은 자연언어의 분석 및 생성 기술이다. 영어의 경우 수십년 동안 축적된 기초자료와 기반기술을 바탕으로 다양한 응용 소프트웨어와 시스템 소프트웨어에서 활용되고 있다. 그러나 한국어 정보처리는 불과 십여년 동안에 자료 부족 및 기초 연구 미흡 등 많은 어려움 속에서 연구가 진행되어 왔다. 형태-통사론적 특성 연구나 통계 자료 등이 부족하여 타언어에 비해 기술 개발이 쉽지 않았을 뿐만 아니라, 상대적으로 연구 인력 등 연구 환경이 취약하여 영어나 일본어 등 선진국에 비해 분석 기술이 뒤떨어져 있다. 특히, 한국어의 교착어 특성과 비정형(non-configurational language) 언어라는 특성은 형태소 분석과 구문 분석 기술의 발전을 더디게 한 가장 큰 요인이 되고 있다. 현재, 한국어 정보처리의 기초 기술을 영어와 비교하면 표 1과 같다[1)][4,5].

표 1. 한국어 정보처리 기초 기술 비교

	영 어	한국어
기초 연구	◎	○
형태소 분석	◎	◎
구문 분석	□	△
의미 분석	△	△
말뭉치 구축	◎	○

◎ : 응용 및 활용
○ : 연구 및 응용
□ : 연구 및 부분적 응용
△ : 연구 단계

응용 분야에서 활용되는 측면에서는 1992년 문서편집기에서 철자검사 기능으로 사용된 것을 시발점으로 1994년부터는 형태소 분석 기술이 정보검색 시스템의 자동색인 기능으로 활용되고 있다[6]. 기계번역의 경우는 일-한, 한-일 기계번역이 인터넷 정보검색을 비롯한 여러 분야에서 활용되고 있고, 영-한 기계번역은 활용 단계에 와 있다[7,8,9].

표 2에서 각 응용 기술이 개발된 시점에는 많은 차이가 있다. 그러나 기초 기술이 타언어에 많이 뒤떨어진 것에 비하면 응용 기술 분야는 크게 뒤떨어진 편은 아니다. 하지만 전자 사전과 말뭉치 구축, 구문 분석, 시소러스, 용례 사전 등 기초 기술이 취약하여 응용 소프트웨어의 성능을 개선하는데 많은 어려움이 있다.

표 2. 한국어 정보처리 응용 기술 비교

	영 어	한국어
연구 시작	50년대 초	80년대 초
철자 검사	70년대 말	90년대 초
문법 검사	80년대 중	90년대 말
자동 색인	80년대 초	90년대 초
기계 번역	80년대 중	90년대 중

기초 기술이 미흡하고 정보 자료가 부족하기 때문에 체계적이고 분석적인 방법으로 연구를 수행하기가 쉽지 않으며, 경험과 직관에 의한 연구는 신뢰도와 성능을 저하시키는 결과를 낳게 된다. 이러한 여건속에서 현재 한국어 정보처리 기술은 전반적으로 영어나 일본어 등 선진 외국에 비해 5~15년 이상의 격차를 보이고 있는 것으로 추정된다.

3. 한국어 정보처리의 문제점

한국어 정보처리 기술은 지난 10여년간의 연구 결과를 통해 여러 가지 문제점들이 발견되고 있다. 그 중에서도 특히 말뭉치를 활용하여 다양한 언어 현상들을 발견하기 어려운 문제, 지속적이고 체계적인 연구 체계의 미흡, 기초 기술과 응용 기술의 조화 등이 한국어 정보처리 시스템이 한 단계 더 높은 수준으로 발전하기 위해서 극복해야 할 과제로 지적된다.

3.1 언어 정보 구축 및 활용

형태소 분석이나 구문분석 등 한국어 정보처리에서 필요한 정보는 각 기능별 관점에서 과학적-분석적인 방식으로 정립되어야 한다. 어휘 사전의 예를 들면, 용도면에서 기존의 사전과 많은 차이가 있다. 기존의 사전이 주로 단어의 의미를 알기 위한 목적으로 사용되는데 비해, 자연언어 처리에서는 언어를 분석 혹은 생성하기 위한 목적으로 사용되기 때문이다.

기계번역에서는 많은 용어가 수록되는 것이 타당하지만, 형태소 분석시에는 저빈도어가 사전에 수록됨으로 인하여 중의성 발생률이 높아져서 오히려 분석 성능을 저하시키는 경우가 발생하기도 한다[10]. 이와같이 세부 기능별로 어휘의 범위나 수록되는 정보에 따라 요구 사항이 다르기 때문에 자연언어 처리에 필요한 언어 정보는 분석이나 생성, 중의성 해결, 기계번역 등 시스템의 성능 향상에 적합한 형태로 구축되어야 한다.

1) 개략적으로 비교한 것으로 구체적인 비교-평가 항목에 따라 달라질 수 있다.

한국어 분석이나 생성을 위해서는 '한국어의 단어 유형'이나 '단어 구성 전이도', 접두사와 접미사 유형, '문장 구조' 등 기초적인 언어 지식에 대한 연구가 필수적이다. 이에 관한 많은 연구가 있었으나 한국어 분석/생성 기술을 개발하는데 필요한 구체적이고 명확한 언어 정보를 습득하는데는 미흡한 점이 있다. 한국어의 분석 및 생성 기술이 발전하려면 한국어의 형태-통사론적 특성에 대한 구체적인 정보자료의 수집이 가능해야 하며, 이를 위해서는 대량의 말뭉치가 구축되어 이를 활용함으로써 한국어의 언어 현상들을 파악할 수 있어야 한다.

3.2 분석적-체계적인 연구 체계

자연언어 처리 기술을 개발하고 실용화하기까지는 많은 어려움이 따른다. 형태소 분석이나 구문 분석, 의미 분석 등 각 기술의 난이도에 따라 차이가 있지만, 자연언어 처리 시스템의 성능을 향상시키는데는 한계가 있다. 시스템이 어느 정도 안정된 수준에 이르면 개별적인 언어 현상들을 처리하는데 많은 노력을 기울이더라도 전체 시스템의 성능에 미치는 영향이 거의 없는 포화(saturation) 상태에 이르게 된다. 포화 상태의 성능을 100%라 하고 포화 상태에 이르기까지 단계별로 동일한 시간과 비용을 투자했을 때 성능 개선 효과는 아래와 같이 추정된다[2].

1 단계 : 초기 성능 70~90%
2 단계 : 5~20% 성능 개선
3 단계 : 3~10% 성능 개선
4 단계 : 1~5% 성능 개선

2단계 이후의 성능 개선 효과가 1~20%에 머무는 이유는 언어 현상의 다양성과 문서내 출현 빈도 때문이다. 일반적으로 자연언어 처리 시스템을 개발할 때 자주 출현하는 유형들을 우선적으로 처리하게 되는데 흔히 발견되는 유형들이 70~90%를 차지하고 있다.
이러한 고빈도 유형들을 처리하기는 어렵지 않으나 그외 저빈도 유형들은 발견하기도 쉽지 않을뿐더러 이를 구현할 때 고빈도 유형과 충돌이 발생하는 경우가 많다. 따라서 저빈도 유형을 처리할 때 고빈도 유형에서 문제가 발생하는 경우가 있어서 이를 피하면서 성능을 개선하기가 쉽지 않다. 또한 특이한 언어 현상들은 별도의 예외 처리가 요구되기도 한다.
일반적으로 전체 시스템의 성능이 포화상태에 도달하기까지는 2~4 단계의 과정을 거치는데, 형태소 분석의 경우 99% 이상의 성능을 기대할 수 있으며, 구문 분석이나 기계번역은 70~80%일 것으로 추정된다. 성능 개선 효과는 개발 방법론에 따라 차이가 있을 수 있으며, 포화 상태의 시스템 성능 또한 방법론에 따라 가변적이다. 이러한 문제점은 언어 현상들이 실제 문서에서 출현하는 빈도와도 밀접한 관련이 있다.
자연언어 처리에서는 모든 언어 현상을 포괄할 수 있도록 시스템을 설계하기가 매우 어렵다. 따라서 자주 출현하는 언어 현상을 중심으로 시스템을 개발한 후에 새로운 언어 현상들을 추가하는 점진적인 방법론을 취하게 된다. 이때 새로운 언어 현상들이 추가되면 처리 방법이나 자료 구조 등이 달라지는 경우가 발생하기도 한다.
이 때 개별적인 언어 현상을 편법으로 처리하게 되면 새로운 언어 현상은 처리되지만 기존의 보편적인 언어 현상들이 처리되지 못하는 현상이 발생하여 더 이상 시스템을 발전시키기 어려운 '통제 불능' 상태에 빠지기도 한다. 이와 같이 새로운 언어 현상을 추가하기 어려운 통제 불능 상태에 빠지는 이유는 문제의 해결 범위와 해결 방법을 분석적이고 체계적으로 접근하기 어려운 자연언어 처리의 특성 때문이다.
'통제 불능' 상태에 빠진 시스템은 더 이상 발전 가능성이 없으므로 기존의 시스템을 폐기하고 좀더 다양한 언어 현상들을 포괄하도록 처음부터 다시 설계하여야 한다. 그러나 대부분의 경우에 시스템을 재설계하기는 쉽지 않으며 위험 부담이 크다. 따라서 시스템을 처음 설계할 때부터 기능별 모듈화 및 점진적인 발전 가능성을 중점적으로 고려해야 한다.

3.3 기초 기술과 응용 기술

자연언어 처리 분야는 분석-생성 기술, 사전 구축 등 기초 기술과 이를 활용하여 응용 소프트웨어를 구현하는 응용 기술로 구분된다. 초기의 자연언어 처리 연구는 기계번역과 자연언어 이해 시스템이라는 응용 기술로부터 시작되었다. 자연언어 이해 시스템은 그 가능성이 매우 희박하다고 판단되고 있으며, 기계번역 또한 국제화 시대에 언어장벽을 해소할 것이라는 초기의 기대감에는 훨씬 못미치고 있다[3].
다만, 기계번역의 경우 번역 업무를 비롯한 특정 분야(domain-specific)에서 번역 전문가가 수행하던 작업을 부분적으로 대신하는 기능으

2) 기계번역이나 구문 분석 등 알고리즘이 복잡한 기능을 중심으로 추정한 것이며, 통계적 태거와 같이 포화상태에 쉽게 도달될 수 있는 것은 적용되지 않을 수도 있다.

3) 철자 검사와 자동색인 기능은 그 특성상 완벽하지 않더라도 응용 분야의 요구 사항을 어느 정도 만족시켜 주고 있지만, 파급효과가 가장 큰 기계번역의 경우 그 성능이 사용자들의 기대 수준과 차이가 커서 부작용이 발생하기도 하였다. 이는 한-영 기계번역 시스템을 외국에서 주도하게 한 원인의 하나이다.

로 만족하고 있고, 성능이 개선됨에 따라 활용 범위가 확대될 수 있는 가능성이 있다[4].

이와 같이 자연언어 처리 기술은 그 특성상 완벽할 수 없기 때문에 응용 소프트웨어의 요구를 100% 만족시켜 줄 수는 없다. 따라서 그때 그때의 기술 수준만으로 응용 분야에서 요구되는 성능을 어느 정도까지 만족시켜 줄 수 있을 것인지가 자연언어 처리 분야의 지속적인 발전에 미치는 영향이 매우 크다.

이 때 발생할 수 있는 문제점 중 하나는 분석-생성 등 기초 기술이 취약한 상태에서 응용 기술에 치중할 때 발생하고 있다. 형태소 분석이나 구문분석, 사전 구축 등 기초 기술이 취약한 상태에서 응용 소프트웨어의 개발을 추진하면 곧 한계에 도달하기 때문이다.

기초 기술의 기반이 취약한 상태에서 이를 응용 소프트웨어에 적용하면 특정 응용 분야에만 적용되는 현상이 발생하고 발전 가능성이 한계에 이르게 된다.

따라서 자연언어의 분석 및 생성, 사전 구축 등 기초 기술은 모든 응용 분야에 적용될 수 있도록 공통적인 기능을 중심으로 개발되어야 하며, 응용 분야마다 요구 사항이 조금씩 달라질 때 응용 분야마다 적응할 수 있도록 확장성을 충분히 고려해야 한다.

이와같이 자연언어 처리 기술은 정보 검색이나 문자 인식, 음성 인식 등 다양한 응용 분야에서 언어 처리 문제를 해결하는 요소 기술로서 활용되기 때문에 각 응용 분야에서 요구되는 언어 처리 문제를 해결하는 역할을 충실히 수행해 나갈 때에만 지속적으로 발전할 수 있다.

4. 한국어 정보처리 방법론

한국어 정보처리 기술은 실험실 수준의 시제품을 개발하는 단계를 넘어 실용적인 수준의 완제품을 개발하는 단계에 와 있다. 시제품 단계는 기본적인 실험에 통하여 실현 가능성을 검증하는 작업이고, 실용화 단계는 문제 해결 방법론을 정립하여 다양한 언어 현상을 포괄할 수 있도록 구현하는 작업이다.

한국어 정보처리 시스템은 자연언어 처리의 특성상 시제품을 개발하는 일은 어렵지 않은 반면에 실용적인 시스템으로 발전시키기가 쉽지 않다[5]. 시스템이 어느 정도 안정된 상태에서 1%의 성능을 향상시키는 것은 초기 시스템의 성능을 10% 향상시키는 것보다 훨씬 힘들다. 어떤 경우에는 1%의 성능을 향상시키는 노력을 한 결과 전체적으로 1%의 성능 감소 효과가 나타나는 역효과도 종종 발생한다.

이러한 역효과는 적용 범위가 좁은 언어 현상을 처리하는 모듈이 적용 범위가 넓은 언어 현상과 충돌할 때 발생하는데, 어떤 언어 현상들이 서로 충돌할 지에 대해서 미리부터 파악하기는 쉽지가 않아서 시스템을 수정한 후에 다시 backtracking하는 상황이 발생하기도 한다.

따라서 1%의 성능 향상을 위해 기울인 노력이 1%에는 미치지 못할지언정 0.1%라도 성능을 향상시킬 수 있도록 신중하게 처리하는 것이 매우 중요하다. 그러기 위해서 때로는 미시적인 관점이 아니라 거시적인 관점에서 방법론을 포함한 전체 시스템을 재검토하는 것이 더 효율적일 수 있다. 초기 단계에서부터 이러한 문제가 발생하지 않도록 시스템을 설계하는 것이 바람직하지만 불가피하게 통제 불능(dead end) 상태에 빠졌을 때 방법론의 변화나 방향 전환으로 인한 추가 부담을 최소화할 수 있는 방법을 모색하여야 한다.

4.1 2-step 패러다임

자연언어 처리에서 시제품을 개발한 후에 일정한 수준의 신뢰도를 보장할 수 있는 실용적인 시스템(operational system)으로 발전시킬 때는 어느 정도의 노력으로 어느 정도의 성능 개선 효과가 나타날지를 예측하기조차 어렵다. 이러한 특성을 극복하기 위한 2-step 패러다임은 실용적인 시스템을 개발할 때 주어진 문제를 하나의 틀 속에서 접근하는 대신에 두 개의 독립된 단계로 구분하여 접근하는 방법이다.

이러한 패러다임은 시제품을 개발한 후에 그 경험을 바탕으로 실용적인 시스템을 개발하는 것과 같이 일반적으로 사용되고 있기도 하다. 여기서는 이를 자연언어 처리에 적용하여

- 핵심 기능과 확장 기능(또는 부가 기능)
- 처리 범위가 넓은 기능과 범위가 좁은 것
- 해결 방법이 명확한 것과 그렇지 않은 것
- 정상적인 처리가 가능한 것과 아닌 것

등으로 구분하여 1 단계로 처리해야 할 문제와 1 단계 처리가 끝난 후에 확장-보완하여 2 단계에서 처리할 문제로 분할하는 것이다. 1 단계는 기본 단계(basic step)로 어떤 문제에 대한 기본적인 기능(단순하고 처리 방법이 비교적 명확한 기능)이고, 2 단계는 확장 단계(extended step)로서 부가적인 기능 혹은 처리 방법이 명확하지 않고 복잡한 기능이다[6].

---

4) 특히, 기계번역은 실생활에 미치는 영향이 매우 크기 때문에 실현 가능성이 불투명하더라도 이를 실현하기 위해 꾸준히 노력할 만한 가치가 있다.

5) 형태소 분석기는 약 2~4 man/month, 기계번역 시스템은 약 2~4 man/year의 작업으로 시제품 개발이 가능하다.

6) 유사한 방법론으로 강승식(1996)은 형태소 분석시에 단어 유형과 품사 유형을 단순형과 확장형으로 구분하고, 형태소 분석 알고리즘을 어형 인식 단계와 어형 확장 단계로 분리하는 두 단계 확장

이러한 예로는 형태소 분석시에 복합어 분해 문제가 있다. 체언접미사나 보조용언이 결합된 복합어는 형태소 분석 결과에 미치는 영향이 크기 때문에 형태소 분석 과정에서 처리되어야 한다. 그러나 복합명사와 결합형 조사/어미의 분해는 형태소 분석 결과에 미치는 영향이 거의 없고, 오히려 형태소 분석 알고리즘의 복잡도를 증가시키는 역할을 한다. 따라서 이 기능은 조사/어미가 분리된 후에 처리하는 것이 바람직하다.

또 다른 예로는, '은/는'이나 '아/어'와 같이 변형이 일어난 어미를 원형 복원할 것인가, 아니면 문서에 나타난 형태로 보존할 것인가 하는 경우이다. 이밖에도 분석 결과에서 단어 유형이나 조사/어미의 빈도수에 따라 우선 순위를 정하는 것이나 일반적으로 의존명사를 붙여쓴 것의 처리, 준말 처리 등이 확장 단계에서 처리될 수 있다.

이와 같이 자연언어 처리의 실용적 방법론을 모색하기 위한 2-step 패러다임의 실현 방안으로는 기능별 모듈화에 의한 divide and conquer 기법, 단순화(KISS: Keep It Simple and Scalable) 기법[7], 예외 처리(exception handling) 기법 등이 있다.

4.2 기능별 모듈화

한국어 정보처리 시스템은 크게 분석과 생성, 그리고 사전 탐색이나 한글 코드 문제 등 부수적인 기능들로 구분된다. 분석 문제는 다시 형태소 분석, 구문 분석, 중의성 해결 등으로 구분되고, 각 모듈은 다시 세부 기능으로 구성된다. 그런데 형태소 분석이나 구문 분석 등 핵심적인 문제는 하나의 독립된 문제로 간주되는 것이 일반적이다.

형태소 분석의 예를 들면, 입력 단어로부터 형태소들을 분리하고 형태소간의 결합 제약을 검사하는 포괄적인 접근 방법이 가능하다. 그러나 포괄적인 형태소 분석 방법을 취했을 때 형태소 유형에 따른 개별적인 특성들을 반영하기가 어렵다[8]. 즉, 개별적인 언어 현상들을 포괄적으로 처리함으로써 처리 범위가 커져서 통제 불능 상태에 빠지기가 쉬우며, 성능 향상이 불가능한 포화 상태에 빠질 가능성이 높아진다.

그런데 형태소 분리 작업은 조사와 어미 분리, 단위 조사/어미 인식, 선어말 어미 처리, 보조용언 분리, 불규칙 용언의 원형 복원, 복합명사 분해, 숫자와 영문자 처리, 준말 처리 등 세부 기능에 대한 처리 방법이 서로 독립적이므로 세부 기능별로 분할하여 모듈화하는 것이 지속적인 성능 향상을 위해 많은 도움이 된다.

이와 같이 처리 범위가 넓은 모듈은 통제 불능 상태에 빠질 가능성이 많으므로 독립적인 모듈로 세분화할 수 있는 것은 모두 세부 기능으로 독립시키는 분할점령(divide-and-conquer) 기법을 활용하는 것이 효율적이다. 이 때 한 모듈이 다른 모듈에 영향을 미치는 것들은 그 내용은 명확히 명시해 둠으로써 모듈별 upgrade시에 이를 관련 모듈에서 반영될 수 있도록 한다.

기능별 모듈화의 장점은 시제품이나 중간 제품을 폐기하고 재설계해야 할 필요성이 발생할 때 특히 유용하므로 자연언어 처리의 특성에 적합한 방법이다. 독립적인 모듈들의 기능을 명확히 정의함으로써 모듈별 재사용이 쉬우므로 재설계로 인한 부담이 적어지기 때문이다. 또한 모듈별로 보다 효과적인 처리 방법론이 고안되었을 때 이를 전체 시스템에 반영하기가 쉽다.

4.3 단순화 기법

일반적으로 자연언어 처리 시스템을 개발할 때 대표적인 언어 현상들을 처리하는데서부터 출발한다. 즉, 처리하고자 하는 대표적인 문장들을 수집하고 이를 중심으로 전체적인 처리 범위를 설정하여 시스템을 설계한다. 이 때 일반화 오류(generalization error)가 발생하기 쉬우며, 개발 과정에서 처리 범위 혹은 방법론을 수정해야 하는 일이 자주 발생한다. 방법론의 소폭 수정은 불가피하지만 한 방향으로 계속해서 수정되면 전반적인 방향이 의도했던 것과 점점 차이가 커질 수도 있다.

방법론을 수정해야 할 시점에 이르면 개발 과정에서 발견된 문제점들을 고려해서 처음부터 다시 시작해야 하지만, 이 정도 수준의 결과를 얻기까지의 세부적인 작업들을 추적하는 문제와 함께 유사한 노력을 반복해야 하는 문제로 인해 쉽지가 않다[9]. 더군다나 알고리즘 혹은 시스템의 구조가 복잡할수록 재시작 문제는 더욱 어려워진다. 그 이유는 세분화되거나 구조가 복잡할수록 적용 범위가 좁아지고 미묘한 차이에 의해 의도했던 바와 다른 결과가 나타나기 때문이다.

언어 현상은 문법 규칙으로 기술되는 보편적인 현상보다 문법 규칙으로 기술되기 어려운 개별적인 현상들이 훨씬 많다. 따라서 단순화시킨 상태에서 시작하더라도 성능 개선 과정에서 구조가 점점 복잡해지게 된다. 더군다나 언어 유형이 세분화된 상태에서 출발했을 때는 복잡도가 더욱 심화될 수밖에 없다.

모델을 제안한 바 있다.

7) 컴퓨터 하드웨어를 설계하는 패러다임으로 Kai Hwang 교수가 주장하였다.

8) 특정 형태소 유형을 처리하기 위해 기능을 수정할 때 이 기능이 다른 유형의 형태소를 분리하는데 미치는 영향을 고려하기가 어렵다.

9) 그렇더라도 재시작하는 것이 장기적으로 볼 때 바람직하며, 최소한 전반적인 시스템의 구조를 재정립해야 한다.

이와 같이 복잡도가 심해지는 문제를 해결하려면 문제를 단순화시킨 모델로부터 시작하면서 확장 가능성을 충분히 고려해야 하고, 복잡도가 증가하는 것을 억제하면서 이를 확장해 나가는 방식을 취하는 것이 바람직하다. 복잡도를 줄이는 방법 중의 하나는 전체 시스템의 구조과 독립적인 부가적인 문제를 다음 단계의 작업으로 남겨두는 일이다.
예를 들어, 형태소 분석시에 복합명사 분해라든지 조사/어미의 결합형으로부터 단위 조사/어미를 분리하는 기능은 체언부가 인식된 후에 처리해도 되므로 꼭 형태소 분석 과정에서 처리할 필요는 없다.

4.4 통계적 기법과 예외처리

분석적 기법은 언어 현상을 처리하는 규칙에 의해 분석 또는 생성 문제를 접근하기 때문에 모든 규칙을 발견하기가 힘들고 규칙간에 충돌 현상이 발생하여 모호성이 증가하기도 한다. 이러한 문제점을 해결하기 위해 통계적 기법을 사용하기도 하지만 통계적 기법은 개별적인 언어 현상을 처리하기가 어렵다. 따라서 분석적 기법과 통계적 기법, 그리고 예외처리 기법을 적절히 조합함으로써 성능 개선이 용이한 시스템을 구축할 수 있다.

4.5 기타

시스템을 설계하거나 성능을 개선하는 과정에서 '이런 경우는 발생하지 않겠지'라고 단언해서는 안된다. 예를 들어, 50단어가 넘는 문장은 없을 것이라든지, 한 단어의 길이가 20음절을 넘지는 않을 것이라는 가정, 혹은 조사와 결합할 수 있는 품사는 체언밖에 없을 것이라는 제약 등은 심각한 시스템 오류를 발생할 가능성을 내포하고 있다. 이러한 문제가 발생하지 않게 하려면 거의 가능성이 없다고 판단된다고 할지라도 반드시 오류 처리(error handling) 기능으로 처리해 주는 것이 좋다.
자연언어의 개별적 현상들은 일부 특수한 경우에만 적용되는 경우가 많기 때문에 정상적인 방법이 아니라 쉽게 편법으로 처리하려는 유혹이 있다. 그러나 계속되는 성능 개선 과정에서 오류가 발견되고 기존의 작업이 헛수고가 되기도 한다. 따라서 개별적인 언어 현상을 처리할 때 그 유형만을 위한 방법론은 부적합하며 유사한 유형들이 발생할 가능성을 고려하여 확장 가능성을 남겨 두는 것이 현명하다.

5. 결론

자연언어 처리 시스템을 개발하는데 가장 큰 어려움은 시제품으로부터 실용적인 시스템으로 발전할 때 국부적인 성능 개선 효과가 전체 시스템의 성능으로 거의 반영되지 않는 통제 불능 상태에 빠지는 것이다. 이러한 문제점을 극복하면서 지속적으로 성능이 개선될 수 있도록 발전하기 위해서 고려되어야 할 여러 가지 요소들을 고찰하였다.
자연언어 처리 시스템의 생명은 성능 개선을 위한 tuning과의 싸움이라고 일컬어진다. 성능 개선이 지속적으로 이루어지기 위해서는 tuning이 용이하여 tuning에 의한 노력이 성능 향상에 최대한 반영될 수 있는 방법론을 취해야 한다. 이러한 방법론으로서 일반적인 소프트웨어 개발 방법론에서 널리 사용되고 있지만, 시스템 개발 과정에서 간과하기 쉬운 기능별 모듈화 기법, 단순화 기법, 2-step 패러다임, 예외처리 기법 등을 적용하는 방안을 제안하였다.

참고문헌

[1] Hutchins W. J., *Machine Translation: Past, Present, Future*, Ellis Horwood Limited, pp.164-167, 1986

[2] Kwon, H. C., Current Status and Trend of NLP technologies in Korea, *Korea-France Joint Workshop on Language Industries*, pp.7-11, 1997

[3] 권혁철, 한글 및 한국어 정보처리의 현황, 정보과학회지, 12권, 8호, pp.3-16, 한국정보과학회, 1994

[4] 강승식, 한국어 정보처리의 현황 및 발전 방향, 말뭉치와 국어정보, 제9회 한국어 사전편찬실 연찬회, 연세대학교, pp.33-41, 1997

[5] 강승식, 미래의 컴퓨터와 한글공학, 한국여성정보인협회 협회지, 10호, pp.43-55, 1996

[6] 박세영, 멀티미디어 정보검색에서의 한국어 정보처리, 정보과학회지, 12권, 8호, pp.60-66, 한국정보과학회, 1994

[7] 김태석, 일한 기계번역 시스템의 연구 및 개발, 정보과학회지, 15권, 10호, pp.9-15, 한국정보과학회, 1997

[8] 강용희, 일본의 한일 기계번역 시스템에 있어서의 오역과 그 언어 환경, 제9회 한글 및 한국어 정보처리 학술대회 논문집, pp.303-310, 1997

[9] 심광섭, 김영택, 기계번역 시스템, 정보과학회지, 12권, 8호, pp.17-23, 한국정보과학회, 1994

[10] 강승식, 이하규, 한국어 형태소 분석기 HAM의 형태소 분석 및 철자 검사 기능, 제8회 한글 및 한국어 정보처리 학술대회 논문집, pp.246-252, 1996

[11] 강승식, 장병탁, 음절 특성을 이용한 범용 한국어 형태소 분석기 및 맞춤법 검사기, 정보과학회논문지(B), pp.530-539, 1996