LEXml을 이용한 기계가독형 전자사전의 표식

정휘웅, 부산대학교 인지과학협동과정, hwjeong@pusan.ac.kr
윤애선, 부산대학교 불어불문학과, asyoon@pusan.ac.kr

Describing a MachineReadable Electronic Dictionary using LEXml

Hwi woong Jeong, Department of Cognitive Science, Pusan National University, hwjeong@pusan.ac.kr
Aesun Yoon, Department of French, Pusan National University, asyoon@pusan.ac.kr

요 약

XML(eXtensible Markup Language)이 1996년 등장한 이후, 기존의 정보를 XML 기반 으로 나타내기 위한 연구가 활발하게 이루어지고 있다. 언어자원(language Resource)과 관련된 분야는 80년대부터 그 연구가 있었으나, XML의 등장 이후, 보다 다양하고 특화된 영역의 정보를 구조화 하기 위한 연구결과가 최근 소개되기 시작하였다. 본 연구에서는 이러한 분야중 전자사전을 표식(markup)하는 XML기반 표준언어인 LEXml(Presentation/Representation of Entries in Dictionaries: LEXml)에 대하여 살펴보고, 기존에 XML로 구축된 전자사전을 LEXml로 변환하여, 그 구조의 확장성과 유효성을 검증할 것이다. 기반자료로써 2000년도에 구축된 MultiDICO의 불어 정보를 이용하였다. 이를 위해 MultiDICO의 XML문서 구조와 LEXml 구조 사이의 각 요소(element)별 대응표를 만들었으며, 이를 바탕으로 XSL(Extensible Style sheet Language)를 작성하였다. 본 연구결과 LEXml이 기존에 구축된 전자사전들을 표식하는데 어려움이 없을 뿐만 아니라, 기계가독성을 높일 수 있는 구조적 유연성이 매우 높은 것을 확인할 수 있었다.

1. 서론

XML의 가장 큰 장점은 인간과 기계사이의 언어 표현에 대한 틈을 줄여준 것일 것이다. 물론 80년대 발표된 SGML(Standard Generalized Markup Language)과 같은 형태의 표식언어(Markup Language) 역시 이러한 차이를 줄이기 위한 노력이었으나, 보다 엄격하고 단순한 형태를 띠는 XML은 그 차이를 줄이기 위한 연구를 더욱 촉진했다. XML의 엄격하고도 단순한 형태의 기준은 수많은 지식을 표상하기 위한 연구가 계속 된 것에서도 찾아볼 수 있으며, 차세대 인터넷의 기반 지식으로써 XML이 그 자리를 차지하고 있음을 알 수 있다. XML 기술을 토대로 하는 차세대 인터넷은 웹 환경을 데스크탑 환경에 보다 가까운 형태로 제공해주고, 생산성(productivity)을 극대화 할 수 있는 기반 기술군의 조합을 의미하며, 2004년 소개되었다. 이의 기술 기반으로써 AJAX(asynchronus Java and XML), xHTML(eXtensible HTML), RSS(Really Simple Syndication/Rich Site Summary), Web log, Web service 등이 있다.[1]

그러나 XML을 중심으로 표준화되고 표식 되어야 할 분야는 여전히 산적해 있으며, 본 연구에서 살펴볼 LEXml도 그 연장선에 있다. 실제 언어자원과 연관된 언어정보의 표준화 및 표식방안 연구는 그 역사가 오래되었다. 언어자원이라는 개념이 실제 등장한 것은 전산화된 언어정보가 급속도로 증가하기 시작한 1990년대 후반 들어서며, 이를 보다 명확히 정의하자면 ""정규화 되어 있으며, 가공성과 재사용성이 있는 형태로 제시되는 인간의 언어와 연관된 지식자원"으로 볼 수 있다.[2] 말뭉치를 구축하고, 여기서 자연언어처리(Natural Language Processing)를 위한 정보를 구조화하는 연구는 이미 60년대부터 컨소시엄 형태의 대규모 프로젝트로 진행되어 왔으며, SGML의 등장과 TEI(Text Encoding Initiative) 등 표준화된 표식언어와 규약은 월드와이드웹(World Wide Web)의 도래 이전부터 언어자원의 표준화된 저장과 활용을 가능케 하였다.

비록 당시의 연구는 기계가독성과 확장성을 고려하였으나, 그 정의가 자의적이어서 호환성이 부족하였을 뿐만 아니라, 전자사전과 같이 특수한 구조의 정보를 기술하기에는 부족함이 많았다. 정보의 재활용이라는 목적은 같았으나, 기계에 좀 더 의존적인지, 인간에게 좀 더 의존적인지에 따라 태그세트의 정의나 문서의 구조 복잡도, 표현 가능성 등이 각기 달랐으며, 시간이 지날수록 그 구조의 틈은 커졌다. 실례로 기계가독성이 극대화된 전자사전 표식언어인OLIF(Open Lexicon Interchange Format)는 기계간 자원의 교환은 매우 원활하게 지원하나, 예문과 같은 인간에게 더욱

의미 있는 정보의 표현은 지원하지 못하고 있다. 반대로 인간에게 친숙한 형태의 전자사전은 단순한 정보 입력 수준을 벗어나지 못하고 있다. OLIF는 교환성과 기계가독성을 높이기 위해 각 속성(attribute)에 들어가는 값을 미리 정의하고, 범위를 벗어나지 않도록 하였다. 이에 따라 시스템간 교환성은 좋으나, 구조의 확장성은 매우 낮아졌다.

이러한 문제점을 해결하기 위해 ISO TC37/SC2 위원회는 2004년 LEXml을 제안하고 있다. LEXml은 현재 공표 바로 전단계인 DIS 단계에 와 있으며, 전문위원회의 통과가 확실시된다. 보다 자세한사항은 [3]을 참조하라.LEXml은 현재의 기존에 구축된 기계가독형 전자사전이 지나치게 기계가독성을 강조하고, 예문이나 굴절형과 같은 언어 문법과 연관된 언어정보를 표현하는데 부족한 단점과, 단순한 검색을 목적으로 구축된 전자사전의 낮은 재사용성을 모두 수용할 수 있는 합리적인 대안이다.[3] 본 고에서는 이러한 LEXml의 특성을 살펴보고, 기존의 XML기반 전자사전을 LEXml로 변환해 봄으로써, 그 효용성과 정보 표현력을 검증해 보고자 한다.

2장에서는 현재까지 이루어진 기계가독형 전자사전을 표식하기 위한 선행 연구에 대해 살펴본다. 3장에서는 본 고에서 표식하고자 하는 LEXml에 대하여 소개하며, 4장에서는 본 연구 이전에 구축된 전자사전의 XML 문서 구조를 분석하여 LEXml의 정보구조에 대응하고, 5장에서는 XSL 문서를 이용하여 사전 정보를 변환하며, 이의 구조를 검증한다. 6장에서는 결론과 앞으로 연구 방향에 대하여 살펴보겠다.

2. 기계가독형 전자사전의 표식

어떤 하나의 사전에 기계가독성이 있다고 하는 것은 (1) 모든 기록되는 정보가 기계가 판독할 수 있는 코드체계로 구성되어 있으며, (2) 특정한 이진(binary) 코드나, 표식언어에 따라서 특정한 프로그램에 의해 표현될 수 있는 형태로 기술(script)되어 있다는 것으로 간주할 수 있다. 이러한 관점에서 본다면, 오늘날 하드웨어에 완전히 이식되어 시판되는 전자사전에 기록된 정보 역시 기계가독형 전자사전으로 보아야 할 것이다.

그러나 오늘날 기계가독형 전자사전이라 함은 자연언어처리를 위한 기반자료 및 그 성능을 향상하기 위한 추론 기반 정보로써, 그 범위가 더욱 확대되었다고 볼 수 있다. 오늘날 기계가독형 전자사전의 기본적인 분류는 90년대 중반 연구에서 살필 수 있는데, 크게 단일어/다국어(mono/multilingua dictionary) 사전, 개념/용어사전(concept dictionary/ Thesaurus), 빈도사전(frequency dictionary)으로 볼 수 있다.[4] 단일어/다국어 사전은 인쇄 사전의 정보를 단순히 기계에 이식한 수준이며, 오늘날까지 국내에서 구축된 대부분의 전자사전도 이에 해당한다. 본 고에서 이용하는 LEXml은 본 인쇄 사전의 부족한 기계가독성과 재사용성을 극대화하기 위한 표준안에 해당한다.

개념/용어사전(concept dictionary, thesaurus)은 오늘날 온톨로지(ontology) 연구로 각광받고 있는 각종 의미망(semantic network)에 해당한다. 90년대 중반에는 개념망에 대한 연구가 부족하여, 개념망과 용어사전을 분리하여 독자적인 사전으로 연구되었으나, 최근 연구는 개념과 용어사전을 하나의 의미망(혹은 온톨로지)으로 보는 견해가 점차 대두되고 있다. 대표적으로 Miller와 Fellbaum에 의해 구축된 워드넷(WordNet)은 약 8만개의 명사를 비롯하여 동사, 형용사, 부사 의미를 포함한 약 11만 의미를 지원하고 있다.[5] 워드넷에서 주목할 부분은 동물과 식물에 대한 의미가 각각 7,458개, 7994개로 일반적인 개념의 어휘 의미에 비해 매우 풍부한 의미를 지원하고 있다.(각각 약 10%) 이에 반해 "noun.top"이라는 최상위 개념을 설명하는 의미군은 불과 45개로서 워드넷은 전문용어의 망(network)과 상위 개념을 기술하는 개념의 망(network)이 혼재하는 형태로 볼 수 있다. 빈도사전은 말뭉치 내부에서 특정 어휘가 발생하는 빈도수를 기록한 사전이다. 주로 자연언어처리를 위한 기반 정보로도 사용되나, 1987년 구축된 Collins cobuild 사전은 1억 어절 코퍼스의 어휘 발생빈도를 이용하여 구축되기도 하였다.

이런 기계가독형 전자사전은 90년대까지 XML에 의한 정보의 표준화 관련 연구가 진행되기 이전에는 SGML에 의해 이루어졌다. SGML은 표식언어의 시초이기는 하나, 그 사용이 까다로웠기 때문에 실제로 많이 사용되지는 않았으며, 주로 학술정보나 특정 정보나 말뭉치를 구축하는데 많이 활용되었다. 가장 대표적인 SGML기반 언어자원 프로젝트는 1980년대 후반 진행된 TEI가 있으며, 오늘날 XML과 호환될 수 있는 TEI P4도 소개되어 있다.[6] 그러나 코퍼스나 사전을 구축할 때 기계가 별도로 판독해야 하는 지식을 나타내기 위해서는 각기 다른 형태의 태그세트나 표식문자를 이용하고 있었으며, 개념망이 더욱 확장된 형태의 정보를 표식하기 위해서는 완전히 다른 형태의 기술 방법을 채택하였다. 가령 워드넷은 별도로 설계된 프로그래밍 언어와 유사한 형태의 스크립트를 작성하고, 이를 컴파일(compile)하는 형태를 준용하였으며, Nirenberg의 Mikrokosmos Ontology의 경우 LISP 언어의 형태를 준용한 TMR(Text Meaning Representation)이라는 독자적인 기술 방법을 준용하였다.

이러한 움직임은 [예1]을 보아도 알 수 있는데, 언뜻 표식된 형태가 SGML로 구성되어 기계가독성이 높을 듯하나, 줄 3에서 "a leg;a (walking) limb;arms;tentacles"정보를 세미콜론(;)으로 구분하

여 이를 판독하기 위해서는 SGML 판독기 이외의 별도 프로그램이 요구된다. 또한, 줄 7에서 "다리가 굵은[가는]"의 형태는, "[가는]"이라는 정보를 구분하기 위해 시작 요소([)와 마치는 요소(])를 다시 탐색해야 하므로 시스템 구조가 복잡해질 뿐만 아니라, 문서의 구조 역시 정교하지 못하다. [6]

```
1 160 다리1
2 <POS> NN
3 <SENSE 1> a leg;a (walking)
  limb;arms;tentacles
4 <COLL 1> 다리 운동 leg exercises
5 <COLL 2> 다리가 굵은 heavy-legged;thick-legged
6 <COLL 3> 다리가 가는 slender-legged
7 <COLL 4> 다리가 굵은[가는] 여자 a heavy-legged
  [slender-legged] woman;a
  woman with plump [slender] legs
8 </SENSE>
```

[예1: 표식언어 형태로 나타나 있으나, 호환성이 낮은 구조. 세미콜론(;)과 같은 문자를 판독하기 위해 별도의 판독기가 필요하다.]

이처럼 기존의 전자사전은 기본적인 정보 구조를 표현하는 데에는 나름의 규칙성을 보이고 있으나, 기계가독성을 극대화하기 위해 프로그래밍에 요구되는 언어정보들은 각기 편의에 의해 기록함으로써, 다음과 같은 문제점을 일으켰다. 첫째, 별도로 정의된 표식문자(;, [,])는 특정한 의미가 있으나, 이를 판독할 수 있는 프로그램은 설계한 프로그래머 자신만이 알고 있으며, 이 정보가 호환성을 가지기 위해서는 매번 다른 프로그램도 해당 표식문자에 대한 판독 기능을 추가해야 하며, 이는 호환성이 낮음을 반증한다. 둘째, 판독 정보가 겹치는 경우(가령, [문자가 줄 7의 중간에 나타나고]문자가 8의 끝에 나타나는 경우), 시스템은 포함된 지식이 사용자의 의도에 의한 것인지, 오류인지를 판별할 수 없다. 따라서 자료의 신뢰도가 떨어지며, 해당 정보구조가 맞는지 확인하기 위한 별도의 프로그램을 구성해야 한다.

따라서 [예1]과 같이 표준형 표식정보에 덧붙여 개별적으로 표식언어를 확장하는 경우 사전정보는 하나의 사전에 대해서는 높은 확장성과 저작(authoring)의 편의성을 확보할 수 있으나, 기계가독성과 재사용성은 현저히 떨어진다. 그러나 자의적 표식문자의 설정 및 호환성이 떨어지는 문제점은 XML의 등장과 함께 상당부분 해소되었다. 요소 내부에 들어가는 정보는 모두 텍스트문자의 범주로 정의하여, 내부의 문자가 다른 의미가 있는 것을 원천적으로 막았으며, 문서의 구조를 검증하기 위한 XML schema가 소개되면서, 정합성에 대한 문제도 해결되었다. XML schema는 각 요소에 저장되는 텍스트 정보를 정의하고, 각 요소의 자식에 올 수 있는 요소를 XML로 정의하는 언어다.

그러나 여전히 각 요소의 의미에 대한 표준화 문제는 여전히 문제로 남게 되었으며, 2000년대 들어서 이에 대한 정규화 노력이 함께 진행되게 되었다. 전자사전에서도 기계가독형 전자사전에 보다 중점을 둔 TMF(Terminology Markup Framework)와 이를 기반으로 구성된 TBX(Term Based Exchange)가 소개되었다. 그러나 인쇄사전을 정규화 하는 문제는 큰 어려움으로 남았으며, 이에 대한 연구의 결과로 소개된 것이 ISO TC37/SC2 위원회에서 제시한 LEXml이다.

3. LEXml

LEXml은 크게 구성요소(component)를 나타내는 요소(entity), 구조(structure)를 나타내는 요소로 나뉜다. 구조와 연관된 요소는 그 수가 많지 않으나, 구성요소를 나타내는 요소는 사전의 정보를 기술하는 것으로써, 다시 세분화되어 나뉜다. 구조적인 요소는 크게 "Dictionary"와 "DictionaryEntry"로 구분되어 나열되며, "Dictionary"요소가 문서의 뿌리(root)를 이룬다. XML 문서는 오로지 하나의 뿌리 요소를 가질 수 있다. 일반적으로 인쇄사전은 특정한 표제어 순서에 의한 나열로 간주하고 있고, 그 나열은 순차적이다. 따라서 이러한 단순한 구조가 사전을 기술하기에는 더욱 합리적인 구성일 것이다. [예2]는 LEXml에 의한 가장 기본적인 사전 정보의 표식이다. 줄 1에서 Dictionary, 그리고 기본 언어와 대상 언어를 선언하며, 이후 줄 2~10에 나타난 것과 같은 형식의 사전 표제어(entry)가 반복적으로 나타날 수 있다.[7]

```
1  <Dictionary version = 'LEXmlV08' profile =
   'LEXmlV08subset'  xml:lang = 'en'
   targetLanguage = 'fr' >
2    <DictionaryEntry>
3      <Headword>hello</Headword>
4      <SenseGroup>
5        <Definition>an expression of greeting
   used on meeting
6          a person</Definition>
7        <Translation>bonjour</Translation>
8        <Translation>salut</Translation>
9      </SenseGroup>
10   </DictionaryEntry>
11 </Dictionary>
```

[예2: LEXml의 정보 표식."Dictionary" 요소를 중심으로

"DictionaryEntry"가 여러 번 나열된다. "Headword"요소를 통하여 표제어를 구분하여 표식할 수도 있다.]

LEXml의 구조적인 특징은 하나의 언어정보를 나타내기 위해 저장소(container)의 개념을 도입하고 있다는 것이다. 하나는 "~Ctn"이라는 작은 형태의 개념이며, 또 하나는 "~Block"이라는 좀 더 큰 형태의 개념이다. "~Ctn"은 하나의 정보에 대해 부가적인 정보가 필요할 경우에 사용되며, "~Block"은 동일한 성격을 가진 여러 정보를 하나로 묶을 경우에 사용된다. 이는 XML의 정보 역시 객체지향(Object Oriented)의 개념을 도입하고, 상위 요소(element)의 특성은 하위 요소로 상속(inheritance)되는 개념에서도 살펴볼 수 있는데, "~Block"은 객체지향의 개념에서 보는 모음(Collection)의 개념으로 볼 수 있다. 객체지향의 주요 개념으로는 상속(inheritance), 캡슐화(encapsulation), 추상화(abstraction), 다형성(polymorphism) 등이 있으며, LEXml은 상속과 캡슐화의 개념을 뿐만 아니라, 추상화의 개념을 지원한다.

가령 [예3]에서 살펴보면 다음과 같다. "Derivation(파생형)"정보는 줄 2와 같이 "<Derivation>" 단독으로 사용될 수도 있으나, 줄 3~6이나 줄 7~11의 "<DerivationCtn>" 형태로도 기술 될 수 있다. 만약 "<Derivation>" 요소가 하나만 나타나는 경우에는 "<DerivationBlock>"을 생략할 수도 있으며, "<Derivation>"요소 아래에 추가적인 정보가 있을 경우 "<DerivationCtn>"요소로 대체하여 사용할 수도 있다. 줄 5, 9, 10은 실제 "<Derivation>"의 정보와 연관된 메타정보(meta-information)을 싣고 있으며, 이러한 구조는 LEXml이 다양한 형태의 사전정보를 효율적으로 표현하면서도 다른 정보와 섞이지 않고 효율적으로 구성될 수 있음을 보여주고 있다.[8]

```
1  <DerivationBlock>
2    <Derivation>cleaved</Derivation>
3    <DerivationCtn>
4      <Derivation>cleft</Derivation>
5      <Pronunciation>kl</Pronunciation>
6    </DerivationCtn>
7    <DerivationCtn>
8      <Derivation>clove</Derivation>
9      <Pronunciation>kl</Pronunciation>
10     <SenseQualifier>literary</SenseQualifier>
11   </DerivationCtn>
12 </DerivationBlock>
```

[예3: LEXml을 이용한 Block과 Ctn. "Derivation"이 부가정보를 포함해야 하는 경우 "~Ctn"에 요소를 저장한다.]

그러나 LEXml의 이러한 정보 구조는 확장성이 있기는 하나, 도리어 확장성이 정보의 구조를 엉성하게 만들 수 있는 위험성도 있다. 가장 대표적인 것이 재귀적인(recursive) 정보의 정의다. [예3]의 "<Derivation>"이라는 요소는 LEXml의 "SecondaryTopic"에 해당하는 정보다. "SecondaryTopic"의 정보는 약어, 파생, 예문, 동의, 굴절 등의 정보를 담을 수 있으며, "~Block"요소와 "~Ctn"요소를 아우르고 있다. 구조적으로 살펴보면 "<DerivationCtn>"요소가 "<Derivation>"을 포함하기 위해서는 "SecondaryTopic"이 "SecondaryTopic"을 포함할 수 있어야만 [예3]과 같은 형태의 표식이 가능해진다. 왜냐하면, XML의 문서 정의에 있어 요소에 대한 정의는 오로지 한 번만 제시되어야 하기 때문이다. XML의 DTD에서 하나의 요소(element)에 대해서는 한 번만 정의할 수 있으며, 포함될 수 있는 요소를 나타내는 데에는 반복적으로 사용될 수 있다.

검증의 관점에 있어 하나의 요소가 재귀적으로 발생할 수 있다는 것은 검증시 시스템이 무한루프를 가동할 가능성이 매우 크다는 점을 시사하나, 역으로 생각하여, 만약 본 정의를 바탕으로 초기 XML 문서를 생성할 경우, 이러한 재귀적 정의로 인해 시스템이 무한루프에 빠질 가능성은 매우 높다. 실제로 XML 문서구조에서 재귀적으로 같은 요소를 반복적으로 정의하는 경우는 매우 드물다는 점과, 개방형 구조를 함으로써, 앞으로 매우 다양한 형태의 정보를 표식 할 수 있다는 장점이 있다. 따라서 본 연구에서는 LEXml의 언어적 확장성을 측정하기 위하여 기보유중인 XML기반 전자사전의 구조를 분석하고, 이를 LEXml로 표식하였다.

4. LEXml 변환을 위한 MultiDICO 분석

MultiDICO는 1994년부터 구축된 다국어지원 전자사전이다. 초기에는 2장에서 소개한 것과 같은 형태의 단순한 형태의 표식언어 구조로 되어 있었으나, XML의 발표와 함께 정보의 구조를 XML로 변환하였다. 또한 다국어 지원을 위하여 불-한 약 46000표제어를 비롯하여 독일어, 서반아어, 러시아어, 그리스어 등 5개 국어를 지원하는 사전으로 2000년 확장되었다.[9]

당시 XML문서의 구조는 그 표식구조가 매우 단순하여 다의어의 구분과, 문법구문 및 예문 정보 구분 수준의 정보를 위한 태그세트를 지원하였다. [예4]에서 살펴보면 하나의 표제어는 "<ENTITY>"라는 요소 아래 저장되며, 관계형 데이터베이스에 저장되므로, LEXml의 "<Dictionary>"와 같은 전체를 아우르는 요소는 없다. 또한 줄 4와 5에서 보듯 "<MEAN>"이나 "<SEN>"과 같은 유사한의미의 요소가 중복되어 표시되는 등 구조가 효율적이지 않다. 각 태그세트의 의미는 [표1]을 참조하라.

```
1    <ENTITY>
2      <WORD>à</WORD>
3      <SEP CLAS="prép" PRON="a" seq="1">
4        <MEAN seq="1">
```

```
5          <SEN lang="KOR">-에게. -을.</SEN>
6          <GRAM>
7            <SEN lang="KOR">au=à+le. aux=à+les 간접 타동사의목적보어 또는 동사에서 온 명사형 목적보어 앞에</SEN>
8          </GRAM>
9          <EXAM>
10           <SEN lang="FRN">nuire àsa santé</SEN>
11           <SEN lang="KOR">건강을 해치다.</SEN>
12         </EXAM>
13         <EXAM>
14           <SEN lang="FRN">parler à +QN</SEN>
15           <SEN lang="KOR">-에게 말하다.</SEN>
16         </EXAM>
       (중략)
17       </SEP>
18   </ENTITY>
```

[예4: MultiDICO의 XML 문서(일부)]

[예4]의 문서구조 각 요소를 발췌하여 LEXml의 구성요소로 대응하면 [표1]과 같다. 특히 "ENTITY"요소의 경우 하나의 어휘이기는 하나, LEXml의 경우 반드시 뿌리 요소를 정의 해주어야만 하므로 "<Dictionary><DictionaryEntry>" 형태의 태그 정의가 이루어져야 하며, 예문의경우에도 번역문이 들어가기 때문에, 하나의 예문을 "ExampleCtn"으로 구성해야 한다. 이러한 변환표를 바탕으로 5장에서는 XSL 문서를 작성하고 MultiDICO의 자료를LEXml로 변환한다.

[표 1 LEXml을 변환하기 위한 MultiDICO의 요소 대치]

변환전	유형	설명	LEXml 요소
ENTITY	E*	하나의 어휘	Dictionary/DictionaryEntry
WORD	E	표제어	HeadWordCtn + HeadWord
SEP	E	다의구분	HomographGroup
MEAN	E	의미	SenseGroup
EXAM	E	예문	ExampleCtn + Example + TranslationCtn + Translation
PHR	E	숙어	CompositionalPhraseCtn + CompositionalPhrase + TranslationCtn + Translation
SEN	E	일반문장	FreeTopic(혹은 부모요소에 따라서 상이하게 해석)
pron	A*	발음	Pronunciation
clas	A	품사	PartOfSpeech
seq	A	일련번호	senseNumber, identifier
an	A	반의어	Antonym
sy	A	동의어	Synonym
es	A	영어동의	Translation + xml:lang ="en"
ab	A	약어	Domain
lang	A	언어	Xml:lang

*E: 요소(element), A: 속성(attribute)

5. 자료의 변환

본 연구에서는 4장에서 대비된 문서를 바탕으로 MultiDICO의 문서를LEXml로 변환하였다. 각 순서를 살펴보면 다음과 같다.

① 모든 표제어가 XML의 "wellformedness"를 준수하는지 확인한다. 준수되지 않으면 해당 오류를 찾아 수정한다. XML에 있어 "wellformedness"라 함은 열리는 요소와 닫히는 요소가 겹치지 않고 이름이 일치하며, 문서의 뿌리는 오로지 하나고, 속성(attribute)의 값이 인용부호로 맞게 기재되어 있음을 뜻한다.

② 기존의 문서를 검증하기 위한 XML schema 문서를 작성한다. 기존의 XML 문서를 XML schema에 적용하여 해당 문서가 지정된 구조에 적합한지를 다시 검증한다.

③ LEXml로 변환된 뒤의 문서를 검증하기 위해 LEXml에서 지원되는 DTD문서를 XML schema로 변환한다.

④ [표1]을 바탕으로 MultiDICO를 변환할 XSL(XML style language) 문서를 작성한다.

⑤ 웹 브라우저에 XSL 문서를 적용하여 MultiDICO의 문서가 변환된 결과를 살핀다..

이 다섯단계의 절차를 거쳐 생성된 LEXml의

문서는 [예5]와 같다. MultiDICO 문서의 변환을 위한 XSL 문서는 http://corpus.fr.pusan.ac.kr/lrms/xsl/lexml.xsl을 참조하라. 또한 LEXml의 구조 검증을 위한 XML schema 문서는 http://corpus.fr.pusan.ac.kr/lrms/xsl/lexml_schema.xsd를 참조. [예4]의 줄 1은 [예5]의 줄 1~3 사이의 태그세트로 변환되어 있으며, [예4]의 줄 5에 설명된 정보는 [예5]의 줄 10에 나타난 "<FreeTopic>"요소로 변환 된 것을 확인할 수 있다. 또한 [예4]의 줄 9~12에 나타난 예문 역시 [표1]에 의거 [예5]의 줄 15~20으로 맞게 변환되어 있음을 확인할 수 있다.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <Dictionary version="LEXmlV08"
   identifier="FKODC">
3    <DictionaryEntry identifier="000001"
   sourceLanguage="fr" targetLanguage="ko">
4      <HeadWordCtn >
5        <HeadWord>à</HeadWord>
6        <PartOfSpeech>prép</PartOfSpeech>
7        <Pronunciation>a</Pronunciation>
8      </HeadWordCtn>
9      <SenseGroup senseNumber="1">
10       <FreeTopic>-에게. -을.</FreeTopic>
11       <GrammaticalNoteCtn>
12         <FreeTopic>
13           au=à+le. aux=à+les 간접 타동사의 목적보
   어 또는 동사에서 온 명사형 목적보어 앞에
14         </FreeTopic>
15         <ExampleCtn>
16           <Example xml:lang="fr">nuire à sa
   santé</Example>
17           <TranslationCtn xml:lang="ko">
18           <Translation>건강을 해치
   다.</Translation>
19 </TranslationCtn>
20         </ExampleCtn>
21       </GrammaticalNoteCtn>
22       <collocationCtn>
23         <collocation xml:lang="fr">parler à
   +QN</collocation>
24       <TranslationCtn xml:lang="ko">
25       <Translation>-에게 말하다.</Translation>
26         </TranslationCtn>
27 </DictionaryEntry>
28 </Dictionary>
```

[예5: LEXml로 변환된 MultiDICO의 문서]

6. 결론 및 논의

본 연구에서는 기존의 전자사전 정보를 LEXml로 표식하고, 이의 구조를 검증하였다. 2장에서는 기존의 사전에 대한 표식구조를 살펴보았으며, LEXml과 같은 기계가독형 사전구조의 필요성에 대해 논의하였으며, 3장에서는 LEXml의 구조적 특성을 살펴보았다. LEXml의 구조는 태그의 정의가 길고 서술적인데 반해, 매우 표현이 직관적이고 일관되기 때문에 언어전문가들이 기계가독형 전자사전을 손쉽게 구축할 수 있는 이점이 있다. 그러나 LEXml은 언어구조에 있어 좀 더 보완해야 할 점이 있다.

첫째, 각 요소의 재귀적 정의는 시스템의 오류를 발생시킬 잠재적인 가능성이 있으므로, 요소들 간의 포함관계를 좀 더 명확하고도 정교하게 구분해야 한다. 둘째, LEXml을 불-한 사전에만 적용하였을 뿐, 다른 형태의 사전과는 연동하지 못했으므로, 아직 하나의 사전을 변환한 결과만으로 구조의 효율성을 확단하기 어렵다. 따라서 추가적인 사전구조에 대한 변환 실험이 있어야 하겠다. 마지막으로 LEXml을 이용하여 구축된 정보를 바탕으로 실제 언어처리에 필요한 정보를 추출하고, 이를 바탕으로 자연언어처리 시스템에 적용한 사례가 없으므로, 이에 대한 연구 역시 뒤따라야 할 것이다.

Acknowledgements

본 연구는 정보통신부 정보통신 학술연구(자유연구 05-학술-019) "효율적 기계번역을 위한 언어자원 국제표준의 통합화에 관한 연구"의 지원에 의해 이루어 졌음.

참고문헌

[1] O'Reilly, Tim(2005) *"What is web 2.0 – Design patterns and business models for the next generation of software"*,

http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[2] Liberman, Mark, Cieri, Christopher (1998) "*The Creation, Distribution and Use of Linguistic Data", Proceedings ofthe First International Conference on Language Resources and Evaluation*

[3] 윤애선, 정휘웅(2006), "전자사전 국제기술표준 LEXml의 정합성 및 확장성", 한국프랑스학논집 제 54집, pp.55~96

[4] Miyoshi, Hideo, Sujiyama, Kenji, Kobayashi, Masahiro, Ogino, Takano(1996), *"An Overview of the EDR electronic Dictionary and the Current Status of Its Utilization"*, COLING-96, pp.1090~1093.

[5] Fellbaum, Christine(eds.)(1999), "WordNet-An electronic lexical database", MIT press

[6] 강범모(2003), "언어, 컴퓨터, 코퍼스 언어학-컴퓨터를 이용한 국어 분석의 기초와 이론", 고려대학교 출판부

[7] Marquis, Johanne(eds.)(2004), Presentation/Representation of Entries in Dictionaries ISO/TC 37/Sc2/N300 Document DIS 1951

[8] 정휘웅, 윤애선(2005), "LEXml기반의 표준화된 전자사전 설계", 『한국사전학회 제7회 학술대회 발표논문집』, 한국사전학회, pp. 123-151

[9] 윤애선 외(1999, 2000), "사전개발자를 위한 인터넷기반 Workbench 구현 및 멀티미디어 다국어 전자사전 개발", 정보통신부, 연차보고서, 최종 연구개발결과보고서