자동 번역용 대용량 번역 지식 DB 관리 시스템 설계 및 구현

장현숙°* 임점미* 유원경* 홍기형* 박상규**
*성신여자대학교 컴퓨터정보학부, **한국전자통신연구원
{hsjang, limjm, wyoo, khhong}@cs.sungshin.ac.kr skpark@computer.etri.re.kr

A Knowledge Management Tool for ETRI Korean/English and English/Korean Automatic Translation Systems

Hyun-Suk Jang°*, Jum-Mi Lim*, Won-Kyung Yoo*, Ki-Hyung Hong* and Sang-Kyu Park**
*School of Computer Science and Engineering, Sungshin Women's University,
**ETRI

요 약

본 논문은 영/한, 한/영 자동 번역용 대용량 번역 지식의 효과적인 관리를 위한 시스템 개발이다. 현재 개발 중인 ETRI의 영/한, 한/영 자동 번역 시스템의 개발 환경을 분석하고, gdbm 기반의 번역지식관리의 문제점을 정리하였다. 본 논문에서 제시한 번역 지식 관리 시스템은 클라이언트/서버 구조를 가지는 MS SQL 서버 기반의 시스템이다. 번역 지식은 관계형 DB로 모델링하여 스키마를 설계하고 구현하였다. 관리 시스템에는 기존에 gdbm 파일로 구축해 놓은 지식을 번역지식 DB로 변환하기 위한 이전 도구, 번역지식 DB에 저장된 번역 지식을 검색하기 위한 검색 도구, 번역지식의 삽입, 삭제, 변경을 지원하는 구축도구로 구성된다.

1. 서론

본 논문은 영/한, 한/영 자동번역 시스템의 개발 과정에서 필요한 지식을 관리하기 위한 DB기술 개발 연구이다. 실용적인 자동 번역 시스템의 개발을 위해서는 초대용량의 정확한 번역 지식과 효과적인 관리가 반드시 필요하다. 번역 지식의 범위는 대용량의 사전과 번역 패턴, 그리고 코퍼스 등이 해당되며, 이들로부터 추출된 다양한 정보를 포함한다.

자동 번역 시스템의 구현을 위하여는 다수의 연구 인력이 공동 작업을 수행하여야 하며, 대용량 코퍼스 및 번역 사전, 그리고 이들로부터 추출된 지식의 공유가 필요하다. 지식의 공유는 자동번역 시스템 개발에 참여하는 각 개인이 생성하는 지식을 다른 사람이 쉽게 접근할 수 있도록 하여, 연구의 효율성을 제고할 수 있다. 이를 위해서는 트랜잭션 관리 기술이 접목되어야 하며, 잘 정의된 통일된 지식 표현 및 저장 방법의 설계가 필요하다. 이러한 번역지식은 매우 방대하므로 다수의 구축자가 존재하며, 번역지식 시스템의 각 구성요소를 사용하는 사용자 역시 다수이다. 그러나 현재 자동 번역 시스템 개발 환경은, GNU에서 제공하는 gdbm[1]을 이용하여 지식 관리를 하고 있다. gdbm은 매우 단순한 API를 가진 해쉬(hash)를 이용하여 지식 관리를 하고 있다.

본 연구는 1999년 한국전자통신연구원 위탁과제로 수행되었음.

이러한 파일을 이용한 번역 지식의 관리는 여러가지 문제가 있다. 본 논문의 목표는 이러한 관리상의 문제를 해결하기 위하여 클라이언트/서버로 운영되는 DBMS기반 지식 관리 시스템의 개발이다. 본 논문에서는 기존 번역지식 관리 상황을 분석하고 효과적인 번역 지식 관리 시스템의 요구 사항을 정리하였다. 이러한 요구 사항을 바탕으로 하여 자동 번역 시스템과 연계한 번역지식 관리 시스템의 구조를 제안하였다.

본 논문의 구성은 다음과 같다. 2절에서는 기존 자동 번역 시스템의 단점을 분석하고 해결 방안을 대용량 번역 지식 관리 시스템의 기능 요구사항으로 정리한다. 3절에서는 영/한, 한/영번역을 위한 지식 데이타베이스 스키마의 설계에 대하여 설명한다. 4절에서는 gdbm API를 지원하는 클라이언트 라이브러리의 설계와 구현에 대하여 기술하고, 5절에서는 구축및 이전도구의 설계와 구현에 대해 설명한다. 마지막 6절에서는 향후과제에 대하여 논하고 결론을 맺는다.

2. 번역지식 관리 시스템의 요구사항 및 구성

기존 번역 지식 관리 시스템은 번역지식 구축도구로 gdbm기반 관리도구를 사용하고 있다. gdbm은 단순한 파일 시스템이다. gdbm 라이브러리는 정적 라이브러리로 클라이언트/서버 구조를 지원하지 못한다.

이러한 이유로 표1과 같은 문제점을 갖게 된다.

데이타 종속성	개별적인 파일 구조와 그에 종속된 응용프로그램으로 인하여, 파일 구조의 변경은 번역 프로그램의 재개발을 야기 시킨다
지식공유 불가	개별적으로 가지고 있는 지식이 개별적인 파일과 응용 프로그램으로 관리됨으로써 지식의 공유가 매우 어렵다
다중성 지원 못 함	특히, 운영체제가 제공하는 단순 파일 시스템은 다수 사용자를 지원하기 위한 트랜잭션 처리 기능이 없다
지식의 중복성	동일한 지식이나 유사 지식이 다른 지식처리 프로그램에서 사용되는 경우, 지식의 중복이 발생할 수 있다
접근 권한 제어 기능 없음	불특정 다수가 사용하고 있는 범용 운영체제에서 접근 권한을 관리하는 것이 매우 어렵다
복구 기능 미흡	미디어 오류나 하드웨어 오류 등으로 지식 파일의 내용이 파괴될 경우, 그 복구가 거의 불가능 하다.
단일 검색 방법	지식의 검색 방법이 개별 응용 프로그램에 의하여 고정되어 있어, 검색 방법의 변경이 요구되거나 다른 검색 방법이 추가될때 번역 프로그램의 재개발이 필요하다
불필요한 지식 정보의 결합과 해석	gdbm의 데이타표현 구조는 키(key)와 내용(contents)의 두가지 필드로 구성되는 매우 단순한 것이다

표 1. gdbm을 이용한 지식관리의 문제점

표1과 같은 문제점을 통해 자동 번역 시스템의 기능적인 요구사항을 도출해 낼 수 있고, 번역지식의 효과적인 관리와 기존 시스템의 문제점을 해결할 수 있는 번역지식 관리 시스템을 설계할 수 있다.

본 논문에서는 그림1과 같은 번역지식 관리 시스템을 구성한다. 새로운 시스템의 구조는 MS SQL 서버[2] 기반의 클라이언트/서버 구조이다. 영/한 또는 한/영 번역지식으로 접근은 관리시스템을 통해 이루어지는데, 관리시스템은 세가지 구성요소를 가진다. 먼저, 검색도구는 번역지식의 정보 구조에 따른 다양한 지식 검색 기능을 제공하고 구축한 지식 DB를 확인할 수 있도록한다. 구축도구는 기본적으로 지식의 삽입, 삭제, 변경을 담당하며, 개별 지식의 입력과 일반 파일 형태로 편집 저장된 다수의 지식을 한번에 수입(import)하는 기능을 포함한다. 마지막으로 이전도구는 gdbm으로 구축된 번역 지식을 MS SQL 서버가 관리하는 영/한 또는 한/영 DB 로 이전(migration)하기 위한 도구이다. 또한, MS SQL DB를 gdbm파일로 변환하는 역이전 기능도 포함한다.

번역 지식 DB설계는 기존 자동 번역 시스템이 사용하고 있는 지식의 정보구조를 분석하여 관계형 데이타베이스[4,5]로 설계하였고, 그 결과는 MS SQL 서버의 T-SQL(Transact-SQL) 스크립트이다.

3. 데이타베이스 스키마 설계

자동 번역 시스템을 위한 데이타베이스는 크게 영문 대 한국어 번역을 위한 영/한 데이타베이스와 한국어 대 영문 번역을 위한 한/영 데이터베이스로 구성하여 설계하였다.

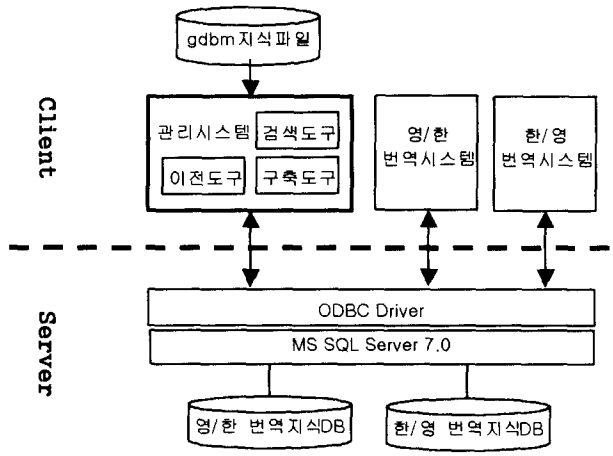

그림1. 번역 지식 관리 시스템 구성

3.1 영/한 데이타베이스 스키마 설계

본 연구에서 설계한 영/한 데이타베이스 구성은 표2와 같다. 테이블 분류중 해석대역은 번역시스템이 정한 12개 영문품사에 따라 테이블을 구성하였고, 연속숙어, 비연속숙어도 같은 구성을 갖는다. 번역메모리는 한번 이상 번역되었던 문장의 정보를 저장하고 있어 곧바로 대역문을 조회할 수 있는 테이블이다. 확률사전은 단어의 품사별 확률값을 저장하고 있다. 원문틀과 대역문틀 제약조건, 대역문틀 테이블은 공통된 속성에 따라 상호 참조할 수 있도록 구성하였다.

테이블 분류	내용
해석대역	기본적인 어휘 정보
연속 숙어	영어 연속 숙어 정보
비연속 숙어	영어 비연속 숙어 정보
번역메모리	한번 이상 번역된 문장 정보
확률사전	영어 단어의 품사별 확률값
원문틀	번역모듈 처리후 생성된 원문틀 정보
대역문틀 제약 조건	번역시스템의 제약조건에 따라 생성되는 원문틀에 대한 대역문틀 ID정보
대역문틀	대역문틀 제약 조건에 의해 검색된 대역문틀의 내용정보
슬롯 대역 규칙	원문과 대역문의 품사열 정보
말뭉치	영어 원문대 한국어 대역문 쌍

표 2. 영/한 데이타베이스 전체구성

슬롯 대역 규칙은 원문의 품사열, 대역문의 품사열 정보를 저장한다. 말뭉치는 영어 원문과 태깅된 원문, 그리고 한국어 대역문을 저장한다.

3.2 한/영 데이타베이스 스키마 설계

본 연구에서 설계한 한/영 데이타베이스 구성은 표 3과 같다. 테이블 분류중 형태소통합사전은 형태소 뿐만 아니라, 복합명사, 연속형 고정표현과 같은 한국어 어휘의 형태소 정보를 저장한다. 원시문틀은 문장내에 나타나는 주요 어휘/품사 지표들을 내용으로 하여 문장 전체를 대상 범위로 하는 일종의 패턴 또는 프레임을 저장한다. 상호정보는 단일어휘 또는 어휘쌍이 구절에 나타난 빈도 정보를 저장한다. 부분대역패턴은 동사어휘 앞에 올수 있는 조사와 해당 동사의 조합된 여러 형태를 저장한다. 구문연결패턴은 동사 어휘를 첫머리로 하는 패턴정보를 저장한다. 번역메모리는 이미 번역된 정보를 저장하여 번역 관리를 위한 자료로 활용된다.

테이블 분류	내용
형태소통합사전	한국어 어휘의 정보
원시문틀	한국어 문장의 패턴
상호정보	어휘 각각에 대한 빈도
부분대역패턴	조사와 동사어휘의 조합된 여러형태
구문연결패턴	동사어휘를 머리로 하는 패턴정보
번역메모리	한번 이상 번역된 문장 정보

표 3. 한영 데이타베이스 전체 구성

4. gdbm API 지원 라이브러리 개발

기존 gdbm을 사용하여 작성된 번역 시스템 구성 모듈들은 클라이언트/서버 구조가 아니다. 하나의 번역 시스템 모듈은 gdbm API를 이용하여 동일한 시스템 안에 존재하는 gdbm 파일에 접근할 수 있지만, gdbm은 정적 라이브러리 형태로 제공되므로 gdbm 라이브러리를 함께 링크하여야 한다. 본 연구에서는 이러한 번역모듈의 변경을 최소화하면서 MS SQL 서버

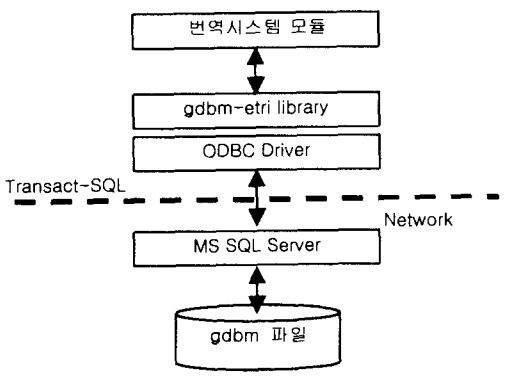

그림2. SQL Server와 gdbm-etri를 이용한 구조

에 구축한 데이타베이스에 접근할 수 있는 라이브러리를 설계, 구현하였다. gdbm-etri 라이브러리는 그림2와 같이ODBC[3] API를 이용하여 gdbm의 API를 구현하였다.

5. 구축및 이전도구의 설계와 구현

본 절에서는 데이타베이스의 데이타를 손쉽게 관리할 수 있는 구축도구와 이전도구에 대해 서술한다. 구축도구및 이전도구는 그림3과 같다. 구축도구는 데이타베이스 데이타를 조회, 삽입, 삭제하는 기능을 지원하며, 영/한과 한/영 데이타베이스를 하나의 구축도구에서 관리하도록 한다. 이전도구는 gdbm 파일로 기존에 구축해 놓은 번역지식의

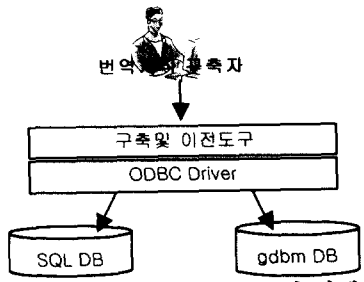

그림 3. 구축및 이전도구의 설계

MS SQL 서버 DB로의 이전을 관리한다.

구축및 이전도구는 ODBC함수를 이용해 SQL Server 접근 및 gdbm DB사이의 이전을 구현하였다.

6. 결론및 향후과제

자동번역 시스템에서 사용하는 번역지식은 원어와 대상어에 대한 사전을 포함하는 매우 방대한 양이다. 이러한 번역용 지식을 효과적으로 관리하지 못하면, 자동번역 시스템의 개발과정과 번역시스템의 성능에 좋지않은 영향을 미친다.

본 논문은 영/한, 한/영 자동 번역 시스템의 개발 과정에 필요한 번역지식의 효과적인 관리 방안에 대하여 알아보았다. 먼저, 현재 자동 번역 시스템의 개발 환경을 알아보고, 그 문제점을 분석하였다. 분석한 문제점의 대부분은 gdbm이라는 단순 파일 시스템 중심으로 지식을 관리하고 있다는데서 기인하였다. 본 논문에서는 이러한 문제들을 해결하고, 번역모듈에는 영향을 미치지 않는 범위내에서 기존 gdbm DB를 수용하는 번역지식 관리 시스템을 제안하였다. 또한, 웹 브라우저를 이용한 관리도구의 개발과 통합시험 뿐만 아니라 색인및 초기 볼륨의 크기에 대한 고려, 운영및 성능에 대한 조정은 향후과제이다.

7. 참고문헌

[1] Philip A. Nelson and Jason Downs, *GNU dbm: a Database Manager*, Free Software Foundation Inc., 1999.

[2] Microsoft, SQL Server Introduction, 1998.

[3] Microsoft, *Open Database Connectivity*, http://msdn.microsoft.com/library/sdkdoc/dasdk/sdko4vcn.htm, 1998.

[4] Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of Database Systems*, The Benjamin/Cummings, 1989.

[5] Carlo Batini, Stefano Ceri, and Shamkant B. Navathe, *Conceptual Database Design*, The Benjamin/Cummings, 1992.