Automatic Augmentation of Translation Dictionary with Database Terminologies in Multilingual Query Interpretation

Hodong Lee and **Jong C. Park**
Computer Science Division and AITrc
Korea Advanced Institute of Science and Technology
373-1 Gusung-dong, Yusong-gu, Daejon 305-701, South KOREA
{hdlee,park}@nlp.kaist.ac.kr

Abstract

In interpreting multilingual queries to databases whose domain information is described in a particular language, we must address the problem of word sense disambiguation. Since full-fledged semantic classification information is difficult to construct either automatically or manually for this purpose, we propose to disambiguate the senses of the source lexical items by automatically augmenting a simple translation dictionary with database terminologies and describe an implemented multilingual query interpretation system in a combinatory categorial grammar framework.[1]

ip-ta	person	body	color	...
	Mary ...	oy-twu ...	kal-sayk ...	... ...

sin-ta	person	foot	status	...
	John ...	sin-bal ...	nalk-ta ...	... ...

sa-ta	person	object	status	...
	John Mary Manny ...	ca-tong-cha sin-bal ko-yang-i ...	nalk-ta nalk-ta nulk-ta ...	... ...

Table 1: Sample Database

1 Introduction

In interpreting multilingual queries to databases with domain information such as objects, table names, and attribute names that are described in a particular language, we must address the problem of word sense disambiguation. For example, if we wish to interpret a query in English to a database with domain information described in Korean, lexical items in English must be disambiguated to the matching senses in Korean. This problem is similar to that of lexical selection in machine translation domain (Lee et al., 1999; Palmer et al., 1999), except that the target is different in the sense that one is a formal query language and the other is another natural language. This difference prompts us to make use of database information, such as domain database objects, table names, and attribute names, instead of the general semantic classifications (Palmer et al., 1999) for disambiguating the senses of lexical items in the query. Example queries are shown below:

(1) (a) Which shoes does Mary buy?
(b) Who *wears* a brown coat?
(c) Who *wears old* shoes and buys an *old* car?

Query 1a shows a query made up of unambiguous words having a unique target interpretation. But in 1b, *wears* may have several interpretations in Korean such as 'ip-ta', 'ssu-ta', 'sin-ta', and 'tti-ta' (cf. Table 3). And *old* in query 1c also contains several senses[2]. If we assume a simple database made up of tables such as 'ip-ta' (to put on the body), 'sin-ta' (to put on the foot), and

[1] This work was supported by the Korea Science and Engineering Foundation (KOSEF) through AITrc.

[2] We notate Korean alphabets in Yale form.

'sa-ta' (buy) in Table 1, *wears* in 1b can be disambiguated by a lexical item 'coat' and its target 'oy-twu', since 'oy-twu' only appears in the table as related to 'ip-ta'. And *wears* in 1c is also restricted by 'shoes', but 'shoes' appears in the table as related to 'sin-ta' and 'sa-ta'. As shown, these senses can be disambiguated with the translation dictionary. Since 'sa-ta', or 'buy', is not registered in the translation dictionary, it is simply discarded. *old* in a query 1c can be interpreted into 'nalk-ta' (not new) and 'nulk-ta' (not young) because it appears in the same table entries for 'sa-ta'. Since it is difficult to disambiguate the senses only with database information, we may utilize co-occurrence information between the collocated words such as (old,shoes) and (old,car) (Park and Cho, 2000; Lee et al., 1999).

In this paper, we propose a disambiguation method with the database information and co-occurrence information (Park and Cho, 2000; Palmer et al., 1999) for the interpretation of natural language queries (Lee and Park, 2001) in multilingual query interpretation. Although we propose to construct the system without an intermediate representation language, we show that our Combinatory Categorial Grammar (CCG) framework is compatible with the approaches with an intermediate representation (Nelken and Francez, 2000; Androutsopoulos et al., 1998; Klein et al., 1998). We also discuss the advantages and disadvantages of these two approaches.

The rest of the paper is organized as follows. A brief introduction to CCGs and natural language database interfaces (NLDBs) will be shown in Section 2. We show the translation process with and without an intermediate representation using CCG in Section 3. The proposed system with multilingual translation is described in Sections 4 and 5.

2 Related Work

In this paper, we propose to interpret natural language queries in English and Korean with CCGs and argue that word selection problem must be resolved for multilingual query interpretation.

Rule					Rule Name (Symbol)
X/Y		Y	$\rightarrow$	X	F/W Application ($>$)
Y		$X\backslash Y$	$\rightarrow$	X	B/W Application ($<$)
X	$conj$	X	$\rightarrow$	X	Coordination ($< \phi^n >$)
X/Y		Y/Z	$\rightarrow$	X/Z	F/W Composition ($> B$)
$Y\backslash Z$		$X\backslash Y$	$\rightarrow$	$X\backslash Z$	B/W Composition ($< B$)
X/Y		$Y\backslash Z$	$\rightarrow$	$X\backslash Z$	F/W Crossed Comp. ($> B_x$)
X			$\rightarrow$	$T/(T\backslash X)$	F/W Type Raising ($> T$)
X			$\rightarrow$	$T\backslash(T/X)$	B/W Type Raising ($< T$)

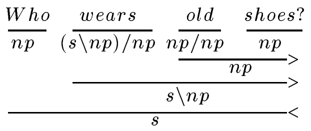
Table 2: CCG Rules for Korean

2.1 Combinatory Categorial Grammar

Combinatory Categorial Grammars (CCGs) are combinatory extensions to the categorial grammars (Steedman, 2000). CCGs are among the lexicalized grammars, such as linear indexed grammars and tree adjoining grammars, and are generally known to provide a wide linguistic coverage and a way of processing sentences incrementally.

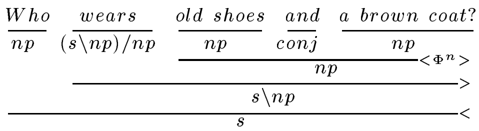
Table 2 shows the CCG reduction rules proposed for Korean (Park and Cho, 2000). (Steedman, 2000) suggested the reduction rules for English which include backward crossed composition and backward substitution. We adopt this rule set for the processing of the queries in English.

(2)



Example 2 shows a syntactic derivation for an example query using CCG. Transitive verbs like 'wears' are assigned the category (s\np)/np, which receives a phrase of category *np* on its right (the second *np* and the directionality is indicated by the slash /, that is, to the right) and then receives another *np* on its left (the first *np* and the directionality is indicated by the backslash \, that is, to the left), to give rise to the phrase of category *s*. Such a computation is done by simple function application. Example 3 shows the CCG derivation for a query with coordination.

(3)



In addition to function application utilized in examples 2 and 3, CCGs use rules for a limited set of combinators such as **B** (function composition), **T** (type raising), and **S** (function substitution) to model natural language. The reader is referred to (Steedman, 2000) for further details.

2.2 Multilingual Database Interfaces

There have been many proposals for NLDBs since the 1960's[3]. In this section, we review some of the more recent ones. (Androutsopoulos et al., 1998; Nelken and Francez, 2000) focus on queries in English with temporal expressions, with a specialized semantic representation language that can handle temporality. Examples are shown below.

(4)
(a) Did any flight circle while runway 2 was open?
(b) Which companies serviced BA737 in 1990?
(c) During which years did Mary work in marketing?

The system in (Klein et al., 1998) interprets noun phrase queries such as 5 in German:

(5) Ersatzzeiten wegen Kindererziehung
(Exemption times because of child raising)

While the system can analyze noun phrases with various adverbial phrases, it is not reported to handle more complex noun phrase queries such as those with subordinate or coordinate constructions.

None of these work deals with multilingual issues. Nor is there much related work in the field of NLDBs. (Thompson and Mooney, 1999) presents a system that automatically constructs the lexicon for NLDBs, in various languages such as English, Spanish, Japanese, and Turkish, which represents the lexical entries with a pair of the phrases and the corresponding semantic representation in first-order logic. Since the semantic representation for lexical items is determined using the frequency of the general terms of the semantic representation in the corpus made up of the query sentences annotated by their logical representation, the system makes it difficult to incorporate various linguistic considerations on natural language.

3 Translation with CCG

In this section, we discuss the translation with and without an intermediate language. The translation based on CCG can derive the target database language expressions/queries such as SQL, TSQL, and QUBE, as well as expressions in intermediate representation languages. We show the translation into both languages with examples (Nelken and Francez, 2000).

3.1 Indirect vs. Direct Translation

Most NLDBs use an intermediate representation which does not make use of expressions that correspond directly to real database objects. The intermediate representations are usually notated as logic expressions such as a quasi-logical form (Klein et al., 1998) and a customized language (Androutsopoulos et al., 1998; Nelken and Francez, 2000). These representations provide a way to translate indirectly to the target database languages.

For example, query 6a is translated into 6b with the intermediate representation L_{Allen} (Nelken and Francez, 1999; Toman, 1996), and into 6c with the SQL/Temporal expressions (Nelken and Francez, 2000).

(6)
(a) During which years did Mary work in marketing?
(b) $year(I) \wedge \exists J(work(mary, marketing, J) \wedge J \subseteq past \wedge J \subseteq I$
(c)
```
NONSEQUENCED VALIDTIME
SELECT DISTINCT a0.c1 AS c1
FROM work' AS a1,year' AS a0
WHERE VALIDTIME(a0) contains VALIDTIME(a1)
AND a1.c1 = 'mary' AND a1.c2 = 'marketing'
AND PERIOD(TIMESTAMP 'beginning', TIMESTAMP
'now') contains VALIDTIME(a1)
```

The translation using an intermediate representation has several advantages, including (a) the availability of an independent linguistic front-end, (b) the separation of domain dependent knowledge from the system engine, and (c) the relative easiness of augmenting the system with an extra inference module for disambiguation (cf. Androutsopoulos et al., 1995). The points (a) and (b) indicate the separation of domain-dependent resources such as lexicon, database mapping information, and other knowledge bases. (c) arises from the modularity of the translation process.

[3] The reader is referred to (Androutsopoulos et al., 1995) for a survey.

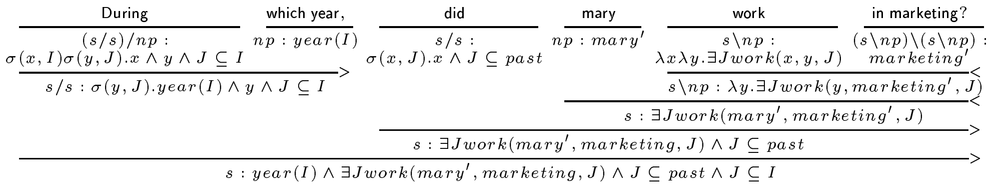

During
which year,
did
mary
work
in marketing?
(s/s)/np : σ(x, I)σ(y, J).x ∧ y ∧ J ⊆ I
np : year(I)
s/s : σ(x, J).x ∧ J ⊆ past
np : mary′
s\np : λxλy.∃Jwork(x, y, J)
(s\np)\(s\np) : marketing′
>
s/s : σ(y, J).year(I) ∧ y ∧ J ⊆ I
<
s\np : λy.∃Jwork(y, marketing′, J)
<
s : ∃Jwork(mary′, marketing′, J)
>
s : ∃Jwork(mary′, marketing, J) ∧ J ⊆ past
>
s : year(I) ∧ ∃Jwork(mary′, marketing, J) ∧ J ⊆ past ∧ J ⊆ I

Figure 1: A Derivation of Example 6a to an Intermediate Representation

When we use an intermediate language, we do not need to concern ourselves with the syntactic details of the target query language during the mapping process, so that we can pay more attention to the differences in syntax between the two source languages (i.e. English and Korean), making the resulting interpretation more reliable. In addition, the use of an intermediate language gives rise to a more flexible query interpretation system as the queries can be translated into multiple target query languages without further processing at the stage of the source query interpretation. However, the use of the same intermediate language for source query languages such as English and Korean that are known to have very different linguistic characteristics makes it difficult to capture subtle differences between the queries of the different source languages unless the intermediate language is quite expressive. And much of the expressiveness of the intermediate language for the translation of the queries in one language may not be what is needed in the translation of the queries in the other.

The translation without an intermediate representation has a simpler and more straightforward process. And there is no extra effort on development of a formal intermediate representation which is difficult to ensure the full coverage on linguistic expressiveness and the soundness of the proposed formalism. Nevertheless, the three points mentioned above are thought to be difficult to overcome in this approach. However, the points (a) and (b) can be equally achieved by separating domain-dependent elements from the query processing module using lexicalized grammars such as CCG. In this case, the construction of a domain-dependent lexicon can be a problem, but it can be resolved to some extent with an automatic construction method. The point (c) is difficult to address, since the translation without an intermediate representation usually is done in a single module. The inference module, however, can be complemented by disambiguation using co-occurrence information (Park and Cho, 2000) and disambiguation of domain-dependent word senses with consideration for the context-dependent information such as information structure (Steedman, 2000). (Nelken and Francez, 2000) use an intermediate representation because the compositional construction of formulae during parsing becomes easier. However, we show that database queries can be interpreted compositionally during parsing without such an intermediate representation through direct translation.

3.2 Translation to an Intermediate Representation

While our approach does not make use of an intermediate representation, the CCG framework itself allows queries to be interpreted into an intermediate representation. Figure 1 shows the translation process from the query 6a to the form 6b which is in L_{Allen}. Since we are only showing the possibility of translation, we use an example from (Nelken and Francez, 2000). In Figure 1, we slightly modified the semantics in (Nelken and Francez, 2000; Nelken and Francez, 1999) for the convenience of translation. And for the same reason, we devised the operator $\sigma(x, I)$ where x is an argument and I represents a time interval variable.

3.3 Translation to a Target Language

Figure 2 shows the translation process from the query 6a to SQL/Temporal expression 6c, also indicating the need for post-processing. For in-

Word	Relation	Collocation	Word sense	Target words
wear	object	coat,glasses	put on	입다(ip-ta), 쓰다(ssu-ta), 신다(sin-ta), 차다(cha-ta)
		expression	express	띠다(tti-ta), 짓다(cis-ta)
old	modifiee	man,book	not young	늙다(nulk-ta), 노령의(nonyen-uy)
		shoe,car	not new	낡다(nalk-ta), 헐다(hel-ta)

Table 3: Part of Word Disambiguation Knowledge for ‘wear’ and ‘old’

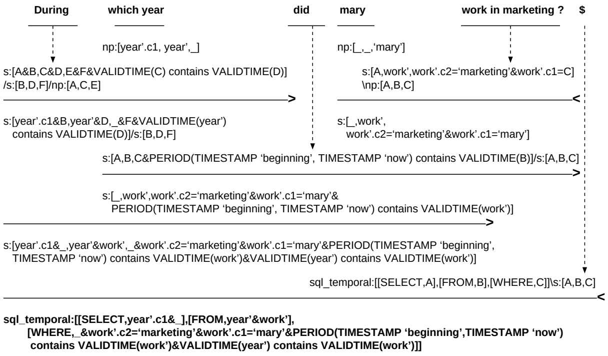

During
which year
did
mary
work in marketing ?
$
np:[year’.c1, year’,_]
np:[_,_,‘mary’]
s:[A&B,C&D,E&F&VALIDTIME(C) contains VALIDTIME(D)]
/s:[B,D,F]/np:[A,C,E]
s:[A,work’,work’.c2=‘marketing’&work’.c1=C]
\np:[A,B,C]
>
<
s:[year’.c1&B,year’&D,_&F&VALIDTIME(year’)
contains VALIDTIME(D)]/s:[B,D,F]
s:[_,work’,
work’.c2=‘marketing’&work’.c1=‘mary’]
s:[A,B,C&PERIOD(TIMESTAMP ‘beginning’, TIMESTAMP ‘now’) contains VALIDTIME(B)]/s:[A,B,C]
>
s:[_,work’,work’.c2=‘marketing’&work’.c1=‘mary’&
PERIOD(TIMESTAMP ‘beginning’, TIMESTAMP ‘now’) contains VALIDTIME(work’)]
>
s:[year’.c1&_,year’&work’,_&work’.c2=‘marketing’&work’.c1=‘mary’&PERIOD(TIMESTAMP ‘beginning’,
TIMESTAMP ‘now’) contains VALIDTIME(work’)&VALIDTIME(year’) contains VALIDTIME(work’)]
sql_temporal:[[SELECT,A],[FROM,B],[WHERE,C]]\s:[A,B,C]
<
sql_temporal:[[SELECT,year’.c1&_],[FROM,year’&work’],
[WHERE,_&work’.c2=‘marketing’&work’.c1=‘mary’&PERIOD(TIMESTAMP ‘beginning’,TIMESTAMP ‘now’)
contains VALIDTIME(work’)&VALIDTIME(year’) contains VALIDTIME(work’)]]

Figure 2: A Derivation of Example 6a to a Target Language

stance, in Figure 2, multiply occurring constants and the uninstantiated variable ‘_’ must be discarded. Additionally, ‘&’ in the result of Figure 2 must be mapped to ‘AND’ and additional information such as ‘NONSEQUENCED VALIDTIME’ and ‘DISTINCT’ must be added for the generation of complete target results as in 6c.

4 Multilingual Translation

Source word disambiguation is an important problem in both of the approaches mentioned in the previous section because the problem of lexical selection arises equally. We propose a method to translate and disambiguate the source queries to the appropriate target database information in a direct translation approach.

4.1 Word Sense Disambiguation and Target Mapping

Our method to disambiguate the source queries is based on the semantic features of the lexical items. In lexical selection methods using the semantic features and their syntactic relations (Palmer et al., 1999; Copestake and Sanfilippo, 1993), the lexicon is designed with semantic type-features constructed from the semantic classifications of a language for the collocated verb-object and modifier-modifiee relations. We also consider these two syntactic relations, but we do not adopt the general semantic classifications that are hard to construct automatically. For this, we would need the additional mapping information to the domain databases. So we designed a method with the database information which can play the role of semantic classifications in the restricted database domain.

In query 1b, the meaning of ‘wears’ is ‘to put on the body’, but in 1c, its meaning is ‘to put on the foot’. The meaning of ‘old’ in 1c is ‘not new’, but that in the phrase ‘the oldest man’ is ‘not young’. Table 3 shows word senses and their candidate target words of ‘wears’ and ‘old’ (Lee et al., 1999). We can disambiguate the senses of ‘wears’ with information in the database, like the sample database shown in Table 1, annotated in the lexical entries. But ‘old’ in 1c cannot be disambiguated with the database information alone because the values of the ‘old’ can occur in the same table attributes as shown in the sample database (Table 1). For this problem, we can think of two disambiguation methods.

- Use of additional semantic type-features based on the semantic classifications
- Use of co-occurrence information between the collocated words

In the first method, the source queries are disambiguated during parsing, but this method requires the semantic classification information. And the semantic features from the classifications generate many lexical entries, since all the senses for a given lexical item have to be accounted for. As a result, we can expect that the increase in the number of lexical entries may also cause the increase in the loss of both the space and processing time of the system.

The second method needs co-occurrence information, but no additional lexical entries. And this method also requires an additional disambiguation process after the parsing to extract information on the collocated words. However, since co-occurrence information between the words can be automatically extracted from a general-purpose corpus, the construction of this information is thought to be relatively straightforward, compared to the construction of the semantic classifications. (Park and Cho, 2000; Lee et al., 1999) proposed to use the co-occurrence information during parsing and lexical selection.

For example, in 1c, 'wears' is disambiguated into 'sin-ta' for the semantics of 'shoes' and the collocated words 'old' and 'shoes' is extracted during the parsing. Then the disambiguation module selects the preferred sense of 'old' through the computation of the similarity for the co-occurrence information. As a result, 'old' is correctly disambiguated into the target 'nalk-ta'.

4.2 Representation of Lexical Entry

In a CCG framework, all the levels of information, such as syntax, semantics, and discourse, are integrated into the categorial lexicon as lexical entries. The following shows example lexical entries of a CCG for English.

(7) (a) lex(coat,np:[_,‘입다’,body=‘외투’];_).
(b) lex(coat,np:[_,‘사다’,clothes=‘외투’];_).
(c) lex(wears,(s:[A,B,C];wear@B;D;E\np:[A,_,_];D)/np:[_,B,C];E).
(d) lex(old,np:[A,sin-ta,status=‘낡다’&C];old˜C;E/np:[A,B,C];E).

The lexical entry consists of a lexical item and its CCG category. The CCG category is a pair of the syntactic and semantic information that are interwoven in the following way. Elementary CCG (syntactic) categories include np and s, and CCG categories are recursively defined as either X/Y or $X\backslash Y$, where X and Y are also CCG categories, including elementary categories. Each elementary CCG (syntactic) category X is augmented with an appropriate semantic information Y and word disambiguation information Z so that the resulting form $X : Y;Z$ is a CCG category (Steedman, 1996). In our proposal, the semantic information is replaced with a suitable fragment of SQL, with slots corresponding to SELECT, FROM, and WHERE clauses in SQL, bracketed by '[' and ']'. For example, in entry 7a, 'coat' is assigned the syntactic category 'np' and the semantic information which encodes the fact that the database attribute 'body' has the value '외투' (oy-twu, meaning 'coat') in the table for '입다' (ip-ta, meaning 'to put on body'). '입다' is described in FROM clause of SQL and 'body=외투' in WHERE clause. In entry 7b, it shows other 'coat' instances in the database table '사다' (sa-ta, meaning 'buy'). In entries 7c and 7d, the verb 'wears' and the adjective 'old' are taken to add information in form of $X@Y$ and $X\tilde{}Y$ for the disambiguation of their senses. $X\tilde{}Y$ provides a template for co-occurrence information.

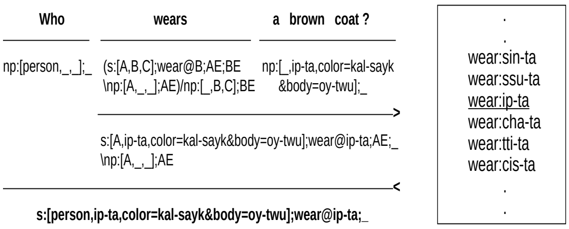

Who
wears
a brown coat ?
np:[person,_,_];_
(s:[A,B,C];wear@B;AE;BE
\np:[A,_,_];AE)/np:[_,B,C];BE
np:[_,ip-ta,color=kal-sayk
&body=oy-twu];_
>
s:[A,ip-ta,color=kal-sayk&body=oy-twu];wear@ip-ta;AE;_
\np:[A,_,_];AE
<
s:[person,ip-ta,color=kal-sayk&body=oy-twu];wear@ip-ta;_
.
.
wear:sin-ta
wear:ssu-ta
wear:ip-ta
wear:cha-ta
wear:tti-ta
wear:cis-ta
.
.

Figure 3: A Derivation of the Query 1c and a Portion of the Translation Dictionary

4.3 Translation Process

Figure 3 shows a derivation of the query 1c and a relevant portion of the translation dictionary. This derivation does not show the binding with SQL syntax. In the final step of the derivation, the syntactic information is combined by a backward application with the category sql:[SELECT A,FROM B,WHERE C]\s:[A,B,C];_. And the exhibited portion of the translation dictionary shows the list of pairs of a word and its target word. Using this information, after the derivation in Figure 3, semantic checking is performed with the tagged information, that is, 'wear@ip-ta'. This tagging is compared with the translation dictionary for the correct sense disambiguation. Through this process, the results that have a matching pair in the translation dictionary are confirmed as the desired result, and the others are discarded. Because the result in Figure 3 has the

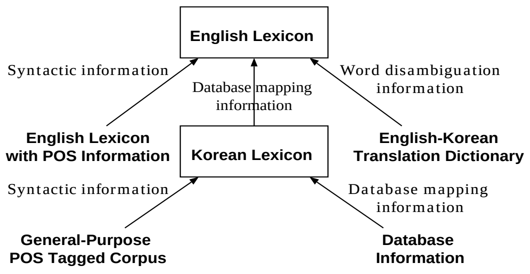

English Lexicon
Syntactic information
Database mapping information
Word disambiguation information
English Lexicon with POS Information
Korean Lexicon
English-Korean Translation Dictionary
Syntactic information
Database mapping information
General-Purpose POS Tagged Corpus
Database Information

Figure 4: Resources for the Lexicon

correct pair in Table 3, it is selected as the right result. The resulting SQL statement is shown below:

(8) SELECT person
FROM ip-ta
WHERE color=kal-saik and body=oy-twu

In response to the SQL statement 8, the answer ‘Mary’ is produced from Table 1.

4.4 Construction of the Lexicon from Available Resources

We construct an English lexicon for the multilingual query from several linguistic resources such as an English lexicon with only POS information, a Korean lexicon for the mapping information and an English-Korean translation dictionary. In our system, the English-Korean translation dictionary is needed in two processes. The first is the process of adding word sense information to the lexical items in English and the second is the process of checking for the senses of the given source word. The Korean lexicon is used for the mapping into the database and the English lexicon with POS tag is used for extracting syntactic categories and syntactic relations between the words. Figure 4 shows the needed information resources for the English and Korean lexicons. The Korean lexicon is constructed by a tool in a semi-automatic manner (Lee and Park, 2001). The lexicon construction tool constructs the Korean lexicon using information from a general-purpose corpus and domain specific database information.

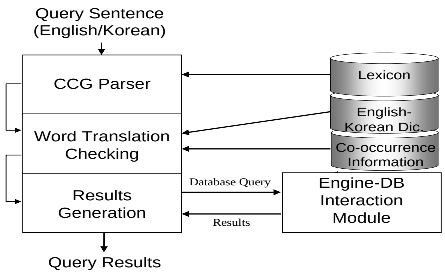

Query Sentence (English/Korean)
CCG Parser
Word Translation Checking
Results Generation
Lexicon
English-Korean Dic.
Co-occurrence Information
Database Query
Results
Engine-DB Interaction Module
Query Results

Figure 5: The Structure of the Query Processing Engine

5 Implementation

Figure 5 shows the structure of the engine that processes multilingual queries. The database is on the home appliance domain in e-commerce. It contains objects for appliance information such as category, name, maker, price, size, other features and so forth. We have populated the database with information from Korean shopping mall websites. Two queries are shown below:

(9) (a) Who *makes* a flat-screen TV set?
(b) SELECT maker FROM product
WHERE name=‘평면사각’ and category=‘TV’

(10) (a) 제일 작은 용량의 냉장고를 사고 싶은데, 가격은 얼마인가요?
I want to buy a refrigerator of the smallest capacity, but what is its price?
(b) SELECT price FROM product WHERE size IN
(SELECT min(size) FROM product WHERE category=‘냉장고’)

The query processing engine is implemented on the UNIX using SICStus Prolog. The word translation checking module performs disambiguation using the English-Korean dictionary (cf. Figure 3) and co-occurrence information. The Korean lexicon contains about a million number of lexical entries, but the English lexicon is comparatively much smaller, and still under construction.

The system can process diverse linguistic expressions in English such as coordination, unbounded dependencies, and gapping etc. The system can also process diverse expressions in Korean including subject ellipsis, noun phrases, numerical expressions, coordination, and subordination where the performance of the system for the queries in Korean is reported in (Lee and Park, 2001).

6 Conclusion

In the paper, we proposed a method to disambiguate the source lexical items of queries with database information such as the objects, table names and attribute names. Since information about the interpreted candidates and the collocated words is extracted during parsing, the implemented query interpretation system can extract the results in a straightforward manner.

Since full-fledged semantic classification information is difficult to construct either automatically or manually in a reliable manner, we proposed to dispense with it and instead to utilize information that can be extracted automatically from the available resources such as the database information, a simple translation dictionary and other linguistic resources.

References

I. Androutsopoulos, G. D. Ritchie, and P. Thanisch. 1995. Natural Language Interfaces to Databases - An Introduction. *Natural Language Engineering*, 1(1):29–81.

I. Androutsopoulos, G. D. Ritchie, and P. Thanisch. 1998. Time, Tense and Aspect in Natural Language Database Interfaces. *Natural Language Engineering*, 4(3):229–276.

A. Copestake and A. Sanfilippo. 1993. Multilingual lexical representation. In *Proceedings of the AAAI Spring Symposium: Building Lexicons for Machine Translation*.

A. Klein, J. Matiasek, and H. Trost. 1998. The treatment of noun phrase queries in a natural language database access system. In *COLING-ACL'98 workshop on the computational treatment of nominals*, pages 39–45.

H. Lee and J. C. Park. 2001. Translating Natural Language Queries into Formal Language Queries with Combinatory Categorial Grammar. In *International Conference on Computer Processing of Oriental Languages*. (to appear).

H. A. Lee, J. C. Park, and G. C. Kim. 1999. Lexical Selection with a Target Language Monolingual Corpus and an MRD. In *Proceedings of International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 150–160.

R. Nelken and N. Francez. 1999. A semantics for temporal questions. In *Proceedings of Formal Grammar*, pages 131–142.

R. Nelken and N. Francez. 2000. Querying Temporal Databases Using Controlled Natural Language. In *COLING*, pages 1076–1080.

M. Palmer, D. Egedi, C. Han, F. Xia, and J. Rosenzweig. 1999. Constraining Lexical Selection Across Languages Using Tree Adjoining Grammars. In *TAG+3 Workshop Proceedings*, CSLI volume.

J. C. Park and H. J. Cho. 2000. Informed Parsing for Coordination with Combinatory Categorial Grammar. In *COLING*, pages 593–599.

M. Steedman. 1996. *Surface Structure and Interpretation*. Number 30 in Linguistic Inquiry Monographs. MIT Press.

M. Steedman. 2000. *The Syntactic Process*. MIT Press.

C. A. Thompson and R. J. Mooney. 1999. Automatic Construction of Semantic Lexicons for Learning Natural Language Interfaces. In *AAAI/IAAI*, pages 487–493.

D. Toman. 1996. Point vs. Interval-based Query Languages for Temporal Databases. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART PODS*, pages 58–67.