Cross-Language Hybrid Keyword and Semantic Search

David W. Embley[1], Stephen W. Liddle[2], Deryle W. Lonsdale[3], Joseph S. Park[1], Byung-Joo Shin[4], and Andrew Zitzelberger[1]

[1] Department of Computer Science,
[2] Information Systems Department,
[3] Department of Linguistics and English Language,
Brigham Young University, Provo, Utah 84602, U.S.A.
[4] Department of Computer Science & Engineering
Kyungnam University, Kyungnam, Korea

Abstract. The growth of multilingual web content and increasing internationalization portends the need for cross-language information retrieval. As a solution to this problem for narrow-domain, data-rich web content, we offer ML-HyKSS: **M**ulti**L**ingual **Hy**brid **K**eyword and **S**emantic **S**earch. The key component of ML-HyKSS is a collection of linguistically grounded conceptual-model instances called extraction ontologies. Extraction ontologies can recognize keywords and applicable semantics; when coupled with cross-language mappings at the conceptual level, they enable cross-language information retrieval and query processing. Our experimental results are promising, yielding good results for cross-language information retrieval with contrasting languages, application content, and cultures.

1 Introduction

With the growth of multilingual web content, it is becoming increasingly important to enable users whose native language is A to find useful information in web pages written in a language B, which they do not know. As an example, consider a person in the USA who wishes to send flowers for a friend's funeral in France and thus wishes to enter the search-engine query "address for funeral of Mrs. Gabrielle DUPIEREUX of Braine-le-Comte" or who wishes to send flowers in place of making a condolence call to a family in Korea and thus wants to search for "mortuary location and interment date for Minhaegyeong's father".

For these queries and many others like them, answers are on the web, but in a language the user does not know. Furthermore, keyword search, though helpful, is insufficient. The keywords "funeral", "Mrs. Gabrielle DUPIEREUX", and "Braine-le-Conte" might be sufficient to return a page with the address—although, of course, with translations "funérailles" for "funeral" and "Madame" for "Mrs." Ideally, however, the search engine should return the address highlighted within the identified source page. Here, "address" is a meta-word rather than a keyword, and should therefore be recognized semantically. In the Korean example, none of the words in the query is likely to be helpful for keyword search—even when

translated. Many of them, however, when translated would semantically relate to a Korean obituary, which would contain information about the hospital for the condolence call, the day of the burial, and the relatives.

These queries call for CLIR (Cross-Language Information Retrieval) [1, 2]. The typical approach to CLIR consists of query translation followed by monolingual retrieval, where systems perform query translation with machine-readable, bilingual dictionaries and machine translation. Our approach to CLIR differs significantly, in that we translate queries only after having semantically conceptualized them and after having separated semantic and keyword components. Semantic conceptualization requires a linguistically grounded conceptual-model instance as its key component, which limits the approach to applications that are data-rich and narrow in scope.

We call our CLIR engine ML-HyKSS (**M**ulti-**L**ingual **Hy**brid **K**eyword and **S**emantic **S**earch). Like search engines, ML-HyKSS assumes the existence of an indexed document collection. Indexes for ML-HyKSS, however, are not just for keywords, but also for recognized semantic concepts. In our prototype implementation, we use Lucene for keyword indexing and extraction ontologies [3, 4], which are linguistically grounded conceptual models, for semantic indexing. In Section 2 we describe extraction ontologies and explain how we semantically index a document collection. ML-HyKSS processes a query for a single language, as we also explain in Section 2, by applying extraction ontologies to the query itself to extract semantic constraints and to isolate non-semantic keywords resulting in a formal conceptualized query. ML-HyKSS then locates relevant documents by matching constraints and keywords of the formal conceptualized query with the semantic and keyword indexes. For cross-language query processing ML-HyKSS transforms a formal conceptualized query Q_A in language A to a formal conceptualized query Q_B in language B. ML-HyKSS can then apply Q_B to the already indexed language-B document collection. Answer transformation is an inverse mapping. In Section 3 we describe the mapping architecture, which explains both query and answer translation. We present experimental results in Section 4 and conclude in Section 5 by summarizing our contributions:

1. Development of cross-language query translation at the conceptual level as an effective alternative to the more traditional language-level translation.
2. Implementation of a prototype system showing the viability of cross-language hybrid keyword and semantic search over diverse languages and applications.

Our earlier work on multilingual ontologies focused on architecture [5]. The work here includes hybrid keyword and semantic search queries—a significant step beyond the types of queries proposed earlier—and an implementation of conceptual-level query transformation. This paper establishes the viability of hybrid query processing and conceptual-level query translation.

2 Query-Processing with Extraction Ontologies

An *extraction ontology* [3, 4] is a conceptual model augmented linguistically to enable information extraction. (To make the paper self-contained we briefly

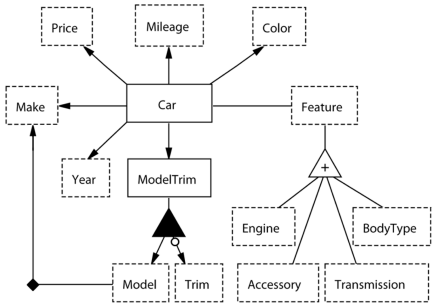

Fig. 2. Car Ad Model Instance.

```
Price
   internal representation: Double
   external representations: \$[1-9]\d{0,2},?\d{3}
      | \d?\d [Gg]rand | ...
   context keywords: price|asking|obo|neg(\.|otiable)| ...
   ...
   units: dollars|[Kk] ...
   canonicalization method: toUSDollars
   comparison methods:
      LessThan(p1: Price, p2: Price) returns (Boolean)
      external representation: (less than | <
         | under | ...)\s*{p2} | ...
      ...
   output method: toUSDollarsFormat
   ...
end

Make
   ...
   external representation: CarMake.lexicon
   ...
```

Fig. 3. Car Ad Data Frames.

introduce extraction ontologies in Section 2.1.) Extracted information constitutes a database structured with respect to the schema induced by a specified conceptual-model instance. When coupled with links into the document collection for each fact extracted, the database constitutes a semantic index. Then with the semantic index in place as well as a standard keyword index, ML-HyKSS can process hybrid semantic/keyword free-form queries[5] (see Section 2.2).

2.1 Extraction Ontologies

The primary components of an extraction ontology are object sets, relationships sets, constraints, and linguistic recognizers. Figure 2 shows a conceptual-model instance for a car-ad application with its object and relationship sets and its constraints, and Figure 3 shows part of two linguistic recognizers—one for *Price* and one for *Make*. Together they constitute an extraction ontology.

The conceptual foundation for an extraction ontology is a restricted fragment of first-order logic. Each object set (denoted graphically in Figure 2 by a rectangle) is a one-place predicate. Each predicate has a *lexical* or a *non-lexical* designation: lexical predicates (denoted by dashed-border rectangles) restrict domain-value substitutions to be literals in domain sets, and non-lexical predicates (denoted by solid-border rectangles) restrict substitutions to be object identifiers that represent real-world objects. Each n-ary relationship set (denoted graphically by a line with n object-set connections) is an n-place predicate. Black diamonds on relationship sets denote prepopulated, fixed relationship sets. Black triangles denote aggregation groupings of relationship sets in an is-part-of hierarchy. Constraints are of three types: (1) referential integrity including optional/mandatory participation (with optional participation denoted by an "o" on the optional side) (2) functional (denoted by edges with arrowheads) and (3) generalization/specialization constraints (denoted by a white triangle).

[5] The hybrid search systems we are aware of require structured queries rather than free-form queries (e.g., [6, 7]), require a keyword-based structured query language (e.g., [8]), or require queries that are an extension of formal conjunctive queries (e.g., [9, 10]). None is multilingual.

Like Buitelaar et al. [11], we linguistically ground ontologies, turning a conceptual specification into an extraction ontology. Each object set has a *data frame* [4], which is an abstract data type augmented with linguistic recognizers that specify textual patterns for recognizing instance values, applicable operators, and operator parameters. Figure 3 shows part of the data frames for the object sets *Price* and *Make* in Figure 2. Although any kind of textual pattern recognizer is possible, our implementation supports regular expressions as exemplified in *Price* and lexicons as exemplified in *Make* or combinations of regular expressions and lexicons. Each relationship set may also have a data-frame recognizer. Relationship-set recognizers reference and depend on data-frame recognizers for each connected object set. In addition, relationship sets may be prepopulated with with a fixed set of relationships that can provide additional context to aid in linguistic grounding. Thus, in Figure 2, the *Make* "Honda" would be additional context information for *Model* "Accord" in the *Car Ad* ontology.

In a data frame, the **internal representation** clause indicates how the system stores extracted values internally, and the **external representation** clause specifies the instance recognizers. The textual distance of matches from *context keywords* helps determine which match to choose for ambiguous concepts within an ontology. The string "25K", for example, could be a *Mileage* or a *Price* but would be interpreted as a *Price* in close context to words such as *asking* or *negotiable*. A **units** clause expresses units of measure or value qualifications that help quantify extracted values. A **canonicalization method** converts an extracted value and units to a unified internal representation. Once in this representation, **comparison methods** can compare values extracted from different documents despite being represented in different ways. These methods can correctly confirm, for example, that "$9,500" is less than "12 grand." The **external representation** clause within a method declaration recognizes text indicating method applicability. The recognizer for the *LessThan* method in Figure 3, for example, identifies comparison phrases in queries like "under 12 grand". The *p2* within curly braces denotes the position in the regular expression of the expected appearance of a *Price* for parameter *p2*. The **output method** is responsible for displaying internally-stored values to the user in a readable format.

2.2 Extraction-Ontology-Based Query Processing

Before query processing begins, ML-HyKSS preprocesses a document collection and creates a keyword index (with Lucene) and a semantic index (with extraction ontologies). ML-HyKSS applies extraction ontologies to text documents to find instance values in the documents with respect to the object and relationship sets in the ontology. The extraction process uses the linguistic recognizers in data frames and the constraints of the conceptual-model structure along with several heuristics to extract instance values. Past work shows that extraction ontologies perform well in terms of precision and recall for the extraction task when documents are rich in recognizable constants and narrow in ontological breadth [3, 4]. ML-HyKSS returns its semantic index as RDF triples that contain, in addition to the extracted data values, internal values converted to appropriate internal

representations by data-frame canonicalization methods, standardized external string representations obtained by data-frame output methods, and information identifying the document and location within the document of extracted text.

To explain and illustrate how ML-HyKSS processes queries, we consider as a running example, the query "Hondas in 'excellent condition' under 12 grand". When ML-HyKSS applies the extraction ontology in Figures 2 and 3 to this query, the data frames recognize "Hondas" and "under 12 grand" as constraints on *Make* and *Price* respectively. Given the nodes *Make* and *Price* in the conceptual-model graph in Figure 2, ML-HyKSS generates a formal SPARQL query in a straightforward way: it generates joins over the edges *Car-Make* and *Car-Price* and also appropriate selection conditions in *FILTER* statements, allowing constraint satisfaction to be *OPTIONAL*, as an open-world assumption requires.

Query generation for expected queries over expected ontologies is straightforward. Like keyword-only queries in which users query for pages that satisfy all, or as many of the keywords as possible, we expect that for hybrid queries users wish to maximize the semantic constraints that are satisfied. Thus, we generate conjunctive queries and always allow constraint satisfaction to be optional. Further, since we expect most ontologies for ML-HyKSS applications to have a simple acyclic structure (or acyclic after excluding prepopulated fixed relationship sets like *Model-Make* in Figure 2), ML-HyKSS can generate queries like it does for the running example in a straightforward way: join over ontology edges that connect identified nodes, and filter conjunctively on identified conditions. For cycles that lead to multiple paths between the same two nodes, such as if the conceptual-model instance in Figure 2 had a second edge between *Car* and *Feature* denoting features that could be added (as opposed to ones the car already has), we expect that relationship-set data-frame recognizers would be able to identify clues in queries that would select one path or another, or in the absence of identified clues, interact with the user to disambiguate the query. Also, like typical search engines, we provide advanced search capabilities for users who wish to pose queries that involve disjunctions and negations.

To add keyword search, making query processing hybrid, ML-HyKSS removes, in addition to stopwords, all meta-words denoting application concepts and all phrases denoting semantic constraints, except equality constraints. For our running example, ML-HyKSS thus removes "in" as a stopword and "under 12 grand" as an inequality constraint, but not "Hondas", which denotes the equality comparison constraint $Make = Honda$. It would also have removed "price" or "cost" as meta-words if they had appeared. Note that it is the plural "Hondas", which is a keyword, even though the *Make* data-frame recognizer converts the plural to a singular for its use in semantic equality comparisons. The keywords for our running example are thus "Hondas" and the phrase "excellent condition".

ML-HyKSS uses its keyword and semantic indexes to identify documents that satisfy its semantic constraints and contain its keywords. It then ranks returned documents by the simple linear interpolation formula: $keywordScore \times keywordWeight + semanticScore \times semanticWeight$ where the scores measure the degree of match with a document and the weights measure the relative contribution

I understood: show me Vehicle.Make and Vehicle.Price where Vehicle.Make = Honda and Vehicle.Price < $12,000
Keywords: Hondas "excellent condition" Advanced Search

Rank	Document	Keywords	Make	Price
1	2002 Honda Accord LX Sedan(highlighted)	excellent condition(1)	Honda	$4,995
2	2002 Honda Accord LX Sedan(highlighted)	excellent condition(1)	Honda	$4,995
3	1997 Honda Accord EX(highlighted)	hondas(1)	Honda	$3,200
4	1993 Honda Accord White(highlighted)	excellent condition(1)	Honda	$3,100
5	2002 Honda Accord!!! 106k 5,995(highlighted)		Honda	$5,995

Fig. 3. Query Result.

2002 Honda Accord LX Sedan - $4995 (Orem)

miscategorized
prohibited
spam/overpost
best of craigslist

Date: *****-**-**, **:**** MST
Reply to: see below

Excellent condition! 158k miles with 2.3L Vtec engine, automatic transmission, new inspections and timing belt, power windows, locks, and mirrors, tilt wheel and cruise control, air conditioning, nice Kenwood CD stereo system. Restored title.

Fig. 4. Highlighted Result.

of keywords and semantic constraints in the query. Figure 3 shows the top-five ranked results when the ML-HyKSS processes the running query over a collection of car ads from craigslist.com. Figure 4 shows the result of clicking highlighted for the top-ranked document.

3 Cross-Language Query Processing

Similar to the work of Dorr et al. [12], we adopt a star or pivot or interlingua-based architecture for cross-language query processing with a central language-agnostic ontology A. Each application has a single central ontology, together with, for each language/locale[6] L an extraction ontology and an L-*to*-A/A-*to*-L mapping specification. Omitting its linguistic grounding, the conceptual-model instance in Figures 2 and 3 is a language-agnostic central ontology.[7] Language-agnostic ontologies are not extraction ontologies because we do not ground them linguistically. They do, however, have full conceptual-model instances with all of an application's object and relationship sets and constraints, and with **internal representation** declarations, **units** declarations, and **method** signatures. The obituaries ontology in Figure 5 is a second example, which we use as we continue to explain how cross-language hybrid query processing works.

[6] For any language, different locales may use different terminology and thus warrant different extraction ontologies—e.g., British English vs. American English.

[7] Although we could use any real or artificial language to represent a language-agnostic ontology, similar to [13], we use English.

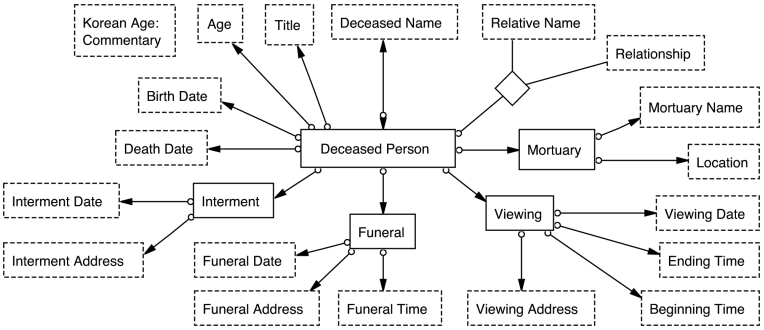

Fig. 5. Obituary Ontology.

For narrow-domain, data-rich applications, we expect extraction ontologies for different languages/locales to be similar, but not identical. In French obituaries, for example, titles such as *Madame* and *Monsieur* regularly appear but are rare in English. As we added French, we therefore added the object set *Title*, which was not part of the original English ontology. In Korea, friends and families make condolence calls usually at the hospital where the deceased body lies in wait between death and interment. Thus, as we added Korean, we also added the information for condolence calls—*Mortuary* and its *Name* and *Location*. Since users may wish to query about all concepts in all languages/locales, we propagate additions to all existing extraction ontologies as well. We thus keep all ontologies structurally synchronized—structurally identical.

Cross-language mappings are compositional through the central ontology. For extraction-ontology languages L_1 and L_2 and central language-agnostic ontology A, the L_1-*to*-L_2 mapping is the composition of the L_1-*to*-A and A-*to*-L_2 mappings. Thus, adding a new language to the system is a linear pay-as-you-go endeavor. Further, because data-rich, narrow-domain ontologies are largely alike in all languages and cultures, and because cross-language lexical resources have been developed over the years and are readily available on the web, the cost of adding a new language/locale to the system is much less than the cost of developing extraction ontologies and mappings from scratch.

Mappings are of several types, and each has its pecularities and its linguistic resources to aid in rapid development:

– *Lexicons.* Lexicon mappings substitute one word by another, or one word by a small number of others. For common concepts such as colors (e.g., for car ads in Figure 2), corresponding translations are available in cross-language dictionaries. Interestingly, these mappings are not one-to-one as many might expect (e.g., "blue" in Korean is 파랑색 and 파란색 and 청색). Concept words not usually in dictionaries such as makes and models of cars are often readily available in lists on the web (e.g., pull-down menus in `http://paruvendu.fr` contain all French make/model combinations, and tabs in `http://www.encar.com` lead to Korean makes and models). Further, identity translations to English are common for many of these words.

- *Units and Measures.* ISO standard conversion formulas for units and measures are commonly available and coding them is straightforward. In our prototype implementation for car ads and obituaries, we use kilometers for mileage, integers for car years and person ages, julian-calendar specifications for dates, and a 24-hour clock for time.
- *Currency.* Because services exist that directly convert amounts in one currency to amounts any other currency, mappings for currency conversions should be an exception to the star-architecture composition rule. However, since the service[8] we were able to conveniently use for our prototype implementation only has conversion rates for US Dollars, we also implement currency conversion as a composition.
- *Transliteration.* Like direct conversion among currencies, transliteration mappings are likely to be best if they map directly from one language to another. We, however, are unable to find a general transliteration service. In our current application, we only need transliteration to and from Korean for English and French and thus use a Hangul/Latin-Language transliterator.[9]
- *Keywords.* Since keywords can be any word or quoted phrase, we use a general translation service, the Bing translator[10] for our implementation. As it should be, the translation is direct, not indirect through the central ontology.
- *Commentary.* Ontologies may contain free-form commentary to explain unfamiliar concepts, such as Korean age. In our implementation, we use the Bing translation service, which translates the Korean explanation for *Age*:

  > 한국의 나이 계산법은 갓 태어난 아이에게 한 살을 부여한다. 그 이후로는, 나이가 새해에 변경된다. 이 방법으로는 12월 31일에 태어난 아이는 다음날에 나이가 두 살로 변경된다. 이 나이 계산법은 공식적으로(법적으로) 사용되지 않으며, 일상적으로는 한국에서 널리 통용되고 있다.

  to:

  > Korean age reckoning is a newborn child was one year old. Since then, the age is changed in the new year. In this way, children born on December 31, the next day is changed to two years of age. In this age of reckoning is not used officially (and legally) on a daily basis, and widely accepted in Korea.

  Translations are usually understandable but not ideal. When important, human translators can provide better translations in "pay-as-you-go" fashion.

By making use of structural correspondences and defined mappings, query transformation from language L_1 to language L_2 can occur at the "deep" conceptual level, as opposed to the more common textual "surface" level. When an extraction ontology interprets a query, it generates an internal representation of the query before generating a formal SPARQL query. The internal representation consists of (1) an acyclic join path in the conceptual-model graph connecting all nodes relevant to the query, (2) an identification of lexical object sets on the path, (3) for each constraint, its Boolean method along with canonicalized values for identified parameters, and (4) keywords and keyword phrases. For our running

[8] `http://raw.github.com/currencybot/open-exchange-rates`
[9] `http://sori.org/hangul/conv2kr.cgi`
[10] `http://api.microsofttranslator.com/V2/Http.svc/Translate`

example—"Hondas in 'excellent condition' under 12 grand"—the join path from Figure 2 is {*Car-Price*, *Car-Make*}, the lexical object sets are *Price* and *Make*, the constraints are *LessThan(x, 12000)* and *Equal('Honda')*, and the keywords are *Hondas* and the phrase *'excellent condition'*. Since all extraction ontologies for an application are structurally identical, the transformed join paths, identified lexical object sets, and methods are immediate. For values and keywords, the mappings provide the transformation. Thus the internal representation for the running query in French consists of referenced lexical object sets *Marque* and *Prix*, constraints *Marque*="Honda" and *Prix* < 9148 €, and keywords *Hondas* and '*excellent état*'. For Korean the internal representation consists of object sets 가격 and 제조사, constraints 제조사 = 혼다 and 가격 < 1340만원, and keywords 혼다 and '좋은 상태가'. When value mappings are one-many, we generate disjunctions; thus, if an English query asks for a blue car, the Korean constraint becomes the disjunction (색상 = 파랑색 ∨ 색상 = 파란색 ∨ 색상 = 청색). Since all internal representations map directly to hybrid SPARQL and keyword queries in the same way, ML-HyKSS immediately obtains and executes the transformed query.

For answer values returned, we use the mappings to transform values and keywords back into the original language. For the query in the introduction, "mortuary location and interment date for Minhaegyeong's father" the answer returned is "Soonchunhyang, Hospital, Hannam, Seoul" on "November 2nd". The date is a units compositional mapping from 11월 2일 to the julian date 306 (the year 2011 assumed) and then to "November 2nd", written as a standard date without the year in English as specified in the **output method** of the English extraction ontology. Mortuary name and location are direct Hangul-to-Latin transliterations (not quite correct, as "Hannam" should be "Hannamdong"). For the query in the introduction, "address for funeral of Mrs. Gabrielle DUPIEREUX of Braine-le-Comte", the answer returned is "l'église Saint-Géry de Braine-le-Comte". Note that neither names nor addresses require any translation for French-to-English.

Advanced queries are also possible in ML-HyKSS. For example, the filled in form in Figure 6 lets a user augment the running query to find either a Honda or a Toyota, but not an Accord, that is red or yellow and still under 12 grand and in 'excellent condition'. HyKSS generates the form from the structure of the extraction ontology along with its declared methods. (Thus, to enable advanced-form capabilities, nothing need be done beyond declaring an extraction ontology.) ML-HyKSS initializes the form with values obtained from a user query and writes them in the entry blanks of the form according to the ontology's **output method**s—thus "Honda" appears as the *Make* while "Hondas" appears as a keyword, and "$12,000" appears for the "<" *Price* operator (not "12 grand"). After initialization, a user can click on OR for *Make* and enter "Toyota" and on OR for *Color* and enter "red" in one field and "yellow" in the other. Clicking on the NOT checkbox for *Model* and entering "Accord" adds the exclusion. Interestingly, users can enter data values without concern for format because ML-HyKSS invokes its data-frame recognizers to interpret user entries.

Like mappings, extraction ontologies also have linguistic resources to aid in rapid development. For example, to quickly obtain a dictionary for French given

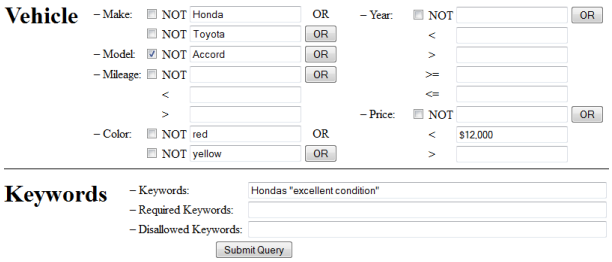

Fig. 6. Advanced Form Query. (Generated from the *Vehicle* extraction ontology we used in our experimental work—not from the extraction ontology in Figures 2 and 3.)

names, we took the list from a book of suggested French baby names.[11] Further, since extraction ontologies are often similar, many regular-expression recognizers can be readily adopted or adapted. For **context keywords**, WordNet synsets are useful. For example, instead of "interment date" a user might say "burial date". The WordNet synset for "interment" is {"burial", "entombment", "inhumation", "interment", "sepulture"}, which includes "burial" and thus, ML-HyKSS will recognize the query as asking for the interment date in the extraction ontology in Figure 5. In general, we can use techniques for automatic query enhancement [14] to automatically populate **context keywords** clauses.

4 Experimental Results

Cross-language query-processing accuracy depends on (1) extraction accuracy in all languages when initially indexing the semantics in a document collection with respect to an application ontology and (2) cross-language query transformation so that nothing is lost or spuriously added in the transformation.

4.1 Extraction Accuracy

To check extraction accuracy for French, we gathered 500 car ads from online sites in France and Canada. The car ads in www.craigslist.fr appear as free-form French, but the rest are mostly semi-structured. For obituaries, we gathered 1500 obituaries from several different online sites in France, Belgium, Canada, and Switzerland—all free-form, but largely conforming to informal conventions. Similarly, for Korean, we gathered 430 car ads from 13 different sites (all semi-structured), and 502 obituaries from 36 different sites (all free-form). We randomly selected about 100 of each of the four combinations to constitute validation and blind test sets (respectively 20 and 80 of the 100) and used the rest for training—training in the sense that we looked at many of them as we linguistically grounded our ontologies.

Tables 1 and 2 show the results. The car ads domain is ontologically narrow, and accordingly our extraction ontologies perform quite well on this domain.

[11] `http://www.journaldesfemmes.com`

Table 1. Car Ad within-language Extraction Results.

		Make	Model	Year	Price	Color	Mileage
French	Recall	87%	76%	96%	89%	82%	98%
	Precision	65%	67%	90%	95%	47%	92%
Korean	Recall	99%	99%	100%	100%	100%	95%
	Precision	99%	99%	100%	100%	100%	95%

Table 2. Obituary within-language Extraction Results.

		Title	Name	Death	Funeral			Mortuary	Relative Name
				Date	Date	Time	Place	Name	& Relation
French	Recall	76%	42%	80%	69%	43%	38%	N/A	—
	Precision	99%	63%	88%	70%	30%	83%		
Korean	Recall	N/A	97%	97%	50%	50%	100%	99%	97%
	Precision		97%	97%	100%	100%	67%	94%	94%

Precision and recall for Korean car ads are high because these ads mostly have a regular structure, allowing our Korean expert to quickly tune the extraction ontology. Results for French car ads are lower both because French ads are more free-form and because we were not able to spend enough time with our French expert tuning the extraction ontology (as one consequence, we did not finish relative-name extraction—hence the empty cell in Table 2). Overall, the numbers are in line with what we have come to expect in this domain [4]. The obituaries domain is much broader and extraction is more challenging—particularly for names and places. Even so, our Korean expert was able to quickly tune the extraction ontology, and performance for most concepts was remarkably high. French extraction was hampered by greater variability and complex sentence structures. For example, there are only 187 names in our Korean surname lexicon, compared with 228,429 in our French surname lexicon, which partially explains the relatively high performance for Korean name extraction. Most Korean obituaries do not mention a funeral, and in our test set there were only two examples of funerals. Our extractor did well with one of the two declared funerals, so the 50% recall we report for funeral date and time represents only one missed concept for the entire corpus. Another cultural difference is that the mortuary name is important in a Korean obituary because that is where friends and family make condolence calls. In contrast, French obituaries refer to viewings, funerals, and interments, much like typical English obituaries. Since mortuary name does not appear in the French obituary ontology, and title does not appear in the Korean ontology, we mark their cells as "N/A". Performance for concepts not listed in Table 2 (e.g. viewing, interment, birth date) is similar to the performance for concepts that are listed.

4.2 Cross-Language Query Accuracy

Obtaining a query set for hybrid keyword/semantic search is not as straightforward as it might seem. Search-engine users learn quickly to adapt their queries

Table 3. Cross-Language Query Transformation Results.

	Recall			Precision		
Car Ad Queries	σ	π	κ	σ	π	κ
French-to-English	77%	86%	100%	81%	90%	74%
Korean-to-English	98%	100%	100%	93%	99%	52%

to the capabilities of a search engine. Users quickly learn not to ask queries like our running example when they see that a top returned result is something far removed from what they want, like grand pianos in 'excellent condition'. Nevertheless, it is possible to explain, and in our case even show how they work, and ask subjects to imagine queries that should work.

To obtain query sets, we asked the students in two senior-level database classes to generate two car-ad queries they felt an earlier demo version of a free-form query processor (not ML-HyKSS) interpreted correctly and two queries they felt the the demo version misinterpreted, but should have interpreted correctly. The students generated 137 syntactically unique queries, of which 113 were suitable for testing ML-HyKSS. To obtain Korean and French queries, we faithfully translated 50 of these 113 into each language.

Table 3 shows the results of interpreting the queries in their respective languages and transforming the internal representation of each query, as understood, into the internal representation of the query in English. In the table the σ columns are for generated database selection operations (Boolean conditions such as *Price* $<$ \$12,000 and *Make* = "Honda" for our running query), the π columns are for generated database projection operations (object sets referenced from which results are returned such as *Price* and *Make* for our running query), and the κ columns are for keywords and keyword phrases ("Hondas" and "'excellent condition"' for our running query). We remark, as explained earlier, that σ and π translations are always correct because ML-HyKSS translates them at the conceptual level by matching methods and object sets respectively, which are necessarily in a one-to-one correspondence. Thus, the less than perfect σ and π results all come from inaccurate within-language query interpretation. It is significant that no σ and π translation errors can occur since these errors are common in cross-language information retrieval systems that translate at the textual language level and then apply information retrieval techniques.

The lower recall and precision for French conditionals (σ) points to a need for better recognizers. For example, we missed recognizing "une 1990 ou moins récente" as the conditional *Year* $\leq$ 1990. Better recognizers, along with more complete synonym sets for ontological concepts, would also increase the recall for requested French results (π). Conditional recognition failures also account for some of the lower keyword (κ) precision, especially for French, as words in missed semantic conditionals remain as possible keywords. Expanded stopword lists in French would remove spurious keywords like "list" and "want". Stopwords in Korean make little sense because most of the standard English-like stopwords are prefixes and suffixes and become part of characters themselves. An attempt to remove them after translation often fails because translations themselves are

often poor; e.g., 인, which in our query should translate as "which is"—both English stopwords—instead was translated as "inn" (or "hotel").

5 Concluding Remarks

We have demonstrated the viability of ML-HyKSS—a conceptual-model-based, cross-language, hybrid keyword and semantic search engine. ML-HyKSS indexes both the semantics and the keywords of a data-rich document collection for an application written in language L_1 and allows users to search the collection with free-form queries in language L_2. While not exhaustive in coverage, our prototype demonstrates its ability to work with a diversity of languages and applications. How well ML-HyKSS performs depends on how well it identifies and interprets an application's semantics in a document collection and in user free-form queries. Experimental results show that ML-HyKSS is able to identify and index the semantics in data-rich, narrow-domain French and Korean documents with an average F-measure of about 90% for semi-structured documents (car ads) and about 75% for unstructured documents (obituaries). Interpretation of French and Korean queries have average F-measures of about 94% for identifying semantic constraints, 87% for identifying referenced concepts of interest, and 77% for identifying keywords. Non-semantic keyword translations are often less than ideal, but since ML-HyKSS translates queries at the conceptual level across language-diverse but structurally identical ontologies, its semantic translations are necessarily correct. Hence, results returned are surprisingly accurate.

Although we have accomplished much (built an ML-HyKSS prototype system, enabled advanced-form hybrid search for non-conjunctive queries, struggled through and resolved issues with international encodings, dealt with diverse languages and locals, learned how to leverage language resources and international standards to mitigate the construction of extraction ontologies), more can still be done. As future work, we can see how to enrich semantic-constraint recognition for superlative queries (e.g., "cheapest car") and aggregate queries (e.g., "average age at death") by including additional types of recognition phrases in data-frame methods. And we see how to enrich identification of referenced concepts for wh-queries by adding keywords for persons ("who"), places ("where"), things ("what"), and time ("when"). We can already say "why" answers hold by giving immediate access to highlighted facts in semantically indexed documents, and we see how to add "why" for inferred facts by adding reasoning chains grounded in facts in indexed documents. Also, we see that automating the construction of interlingual mappings and linguistic-grounding components of extraction ontologies from available language resources would be valuable.

Acknowledgements We are grateful for the work of Tae Woo Kim and Rebecca Brinck in annotating Korean and French document sets.

References

1. J. Olive, C. Christianson, and J. McCary, editors. *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Ex-*

ploitation. Springer, 2011.
2. C. Peters, M. Braschler, and P. Clough. *Multilingual Information Retrieval: From Research to Practice*. Springer, 2012.
3. D.W. Embley, D.M. Campbell, Y.S. Jiang, Y.-K. Ng, R.D. Smith, S.W. Liddle, and D.W. Quass. A conceptual-modeling approach to extracting data from the web. In *Proceedings of the 17th International Conference on Conceptual Modeling (ER'98)*, pages 78–91, Singapore, November 1998.
4. D.W. Embley, S.W. Liddle, and D.W. Lonsdale. Conceptual modeling foundations for a web of knowledge. In D.W. Embley and B. Thalheim, editors, *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*, chapter 15, pages 477–516. Springer, Heidelberg, Germany, 2011.
5. D.W. Embley, S.W. Liddle, D.W. Lonsdale, and Y. Tijerino. Multilingual ontologies for cross-language information extraction and semantic search. In *Proceedings of the 30th International Conference on Conceptual Modeling (ER 2011)*, pages 147–160, Brussels, Belgium, October/November 2011.
6. P. Castells, M. Fernandez, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowedge and Data Engineering*, 19(2):261–272, February 2007.
7. R. Bhagdev, S. Chapman, F. Ciravegna, V. Lanfranchi, and D. Petrelli. Hybrid search: Effectively combining keywords and ontology-based searches. In *Proceedings of the 5th European Semantic Web Conference (ESWC'08)*, pages 554–568, Tenerife, Canary Islands, Spain, June 2008.
8. J. Pound, I.F. Ilyas, and G. Weddell. Expressive and flexible access to web-extracted data: A keyword-based structured query language. In *Proceedings of the 2010 International Conference on Management of Data (SIGMOD'10)*, pages 423–434, Indianapolis, Indiana, June 2010.
9. H. Wang, T. Tran, and C. Liu. CE^2: Towards a large scale hybrid search engine with integrated ranking support. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM'08)*, pages 1323–1324, Napa Valley, California, October 2008.
10. L. Zhang, Q. Liu, J. Zhang, H. Wang, Y. Pan, and Y. Yu. Semplore: An IR approach to scalable hybrid query of semantic web data. In *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC'07)*, pages 652–665, Busan, Korea, November 2007.
11. P. Buitelaar, P. Cimiano, P. Haase, and M. Sintek. Towards linguistically grounded ontologies. In *Proceedings of the 6th European Semantic Web Conference (ESWC'09)*, pages 111–125, Heraklion, Greece, May/June 2009.
12. B.J. Dorr, E. Hovy, and L. Levin. Machine translation: Interlingual methods. In K. Brown, editor, *Encyclopedia of Language and Linguistics*. Elsevier, 2nd edition, 2004.
13. D.W. Lonsdale, A.M. Franz, and J.R.R. Leavitt. Large-scale machine translation: An interlingua approach. In *Seventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-94)*, pages 525–530, Austin, Texas, May/June 1994.
14. C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys*, 44(1), January 2012.