

# Multilingual Ontologies for Cross-Language Information Extraction and Semantic Search

David W. Embley<sup>1</sup>, Stephen W. Liddle<sup>2</sup>, Deryle W. Lonsdale<sup>3</sup>, and Yuri Tijerino<sup>4</sup>

<sup>1</sup> Department of Computer Science,

<sup>2</sup> Information Systems Department,

<sup>3</sup> Department of Linguistics and English Language,  
Brigham Young University, Provo, Utah 84602, U.S.A.

<sup>4</sup> Department of Applied Informatics,  
Kwansei Gakuin University, Kobe-Sanda, Japan

**Abstract.** Valuable local information is often available on the web, but encoded in a foreign language that non-local users do not understand. Can we create a system to allow a user to query in language  $L_1$  for facts in a web page written in language  $L_2$ ? We propose a suite of multilingual extraction ontologies as a solution to this problem. We ground extraction ontologies in each language of interest, and we map both the data and the metadata among the language-specific extraction ontologies. The mappings are through a central, language-agnostic ontology that allows new languages to be added by only having to provide one mapping rather than one for each language pair. Results from an implemented early prototype demonstrate the feasibility of cross-language information extraction and semantic search. Further, results from an experimental evaluation of ontology-based query translation and extraction accuracy are remarkably good given the complexity of the problem and the complications of its implementation.


## 1 Introduction

Many users, especially those traveling abroad or doing business in multiple countries and cultures, would like to be able to query foreign-language sites on the web in their own language. An ideal app would allow users to pose queries in their own language, run these queries against foreign-language sites, and return results in their own language. A user  $U$ , for example, who speaks only English, may wish to enquire about nearby restaurants while visiting Osaka, Japan. Using an iPhone,  $U$  may wish to pose a query to find a “BBQ restaurant near the Umeda station, with typical prices less than \$40.” The app should rewrite  $U$ ’s inquiry in Japanese, access Japanese web pages to find restaurants that satisfy the criteria, respond with answers in English, and allow  $U$  to tap on answers to obtain more detail in English. Figure 1 gives actual answers retrieved from the web for this sample query. Figure 2 shows an interface with the query in a type-in text field, the English version of the answers retrieved, and a list of additional

available information about the restaurants. If  $U$  then checks the check-box for one or more of these restaurants (e.g., the checked box for Shin-YakinikuYa) and clicks on [PaymentMethod](#), the additional information in Figure 3 appears.

店名	住所	ジャンル	予算
新焼肉屋	梅田1-10-19	焼肉	2000
肉屋	梅田1-11-29	焼肉	3000
美味焼肉	梅田2-30-22	焼肉	1500
焼肉屋	梅田3-19-28	焼肉	3000
焼き焼き	梅田2-18-26	焼肉	1000

Fig. 1. Results Extracted from Japanese Web Pages.

BBQ restaurant near the Umeda station, with typical prices less than \$40. 

	Name	Address	Cuisine Type	Price Range
<input checked="" type="checkbox"/>	Shin-YakinikuYa	1-10-19 Umeda	BBQ	\$15-30
<input type="checkbox"/>	NikuYa	1-11-29 Umeda	BBQ	\$30-50
<input type="checkbox"/>	OishiiYakiniku	2-30-22 Umeda	BBQ	\$15-30
<input type="checkbox"/>	YakinikuYa	3-19-28 Umeda	BBQ	\$30-50
<input type="checkbox"/>	Yakiyaki	2-18-26 Umeda	BBQ	\$5-15

More details: [Hours of Operation](#), [Payment Method](#), [Rating](#), [Serving Style](#), [Tipping Protocol](#)

Fig. 2. English Query over Japanese Data with Results Translated to English.

**More Details** (Shin-YakinikuYa, 1-10-19 Umeda, BBQ, \$15-30)

**Payment Method:** Cash, MasterCard, Visa

[Back to Results](#)

Fig. 3. Payment Method Information.

Although within-language information extraction and semantic search is a common research topic (e.g., [Sar08, TAC06]), much less effort has been devoted to cross-language information extraction and query processing, where the user's query and the information sources are not in the same language (e.g., [Gre98]).

The U.S. government<sup>5</sup>, the European Union<sup>6</sup>, and Japan<sup>7</sup> all have initiatives to help further the development and evaluation of multilingual and crosslinguistic information retrieval and information extraction systems. Of course, companies interested in web content and market share are also working on ways to provide multilingual access to the Internet. Many of the existing crosslinguistic efforts involve a scenario that includes a hybrid of variously configured extraction and machine-translation technologies [KHF<sup>+</sup>01]. Such approaches are complicated by the status of efficient, accurate machine-translation engines, as yet another ongoing research effort. One group mitigates this problem by directly annotating web pages with conceptual vectors in an interlingua representation [FRS<sup>+</sup>10] to assure direct extraction against queries in any language. The use of an interlingua [LFL94] also represents the central paradigm for translating between languages in several machine-translation systems [DHL04]. The use of conceptual ontologies in this type of work is fairly common (see, for example, [MDL<sup>+</sup>06]).

To address the multitude of problems in cross-language information extraction and semantic search, we propose here ML-OntoES (MultiLingual Ontology Extraction System). ML-OntoES is a conceptual-modeling approach to crosslinguistic information processing based on extraction ontologies. An extraction ontology is a linguistically grounded conceptual model capable of populating its schema with facts extracted from web pages [ECJ<sup>+</sup>99,ELL11]. Extraction ontologies also extract information from free-form user queries, match the information with ontological conceptualizations, and generate formal queries over populated schemas [AME07]. The key idea that makes ML-OntoES work is the mapping of each language-specific extraction ontology to and from a central, language-agnostic ontological conceptualization of a narrow domain of interest. The basic premise draws on machine translation through interlinguas, but our application of this notion to extraction ontologies is new.

To illustrate our approach consider the user query in Figure 2. ML-OntoES “translates,” “extracts,” and “translates again” as follows: we (1) apply an English restaurant extraction ontology to match the query to a conceptual model, (2) use pre-determined mappings through a central language-agnostic conceptual model to a Japanese restaurant extraction ontology, (3) extract both requested facts and ontologically related facts from Japanese web sites with the Japanese restaurant extraction ontology, (4) map returned results (e.g., Figure 1) and related results through the central language-agnostic conceptualization back to the English restaurant extraction ontology, and (5) display results and links to additional information (e.g., Figures 2 and 3).

The contributions of this work include: (1) development of an architecture with a central language-agnostic ontological conceptualization for cross-language information extraction and semantic search (Sections 2.1–2.2) (2) specification of mapping types to and from the central conceptualization along with scalable, pay-as-you-go ways to establish both mappings and new language-specific

<sup>5</sup> See <http://trec.nist.gov>.

<sup>6</sup> See <http://www.clef-campaign.org>.

<sup>7</sup> See <http://research/nii.ac.jp/ntcir>.

extraction ontologies (Section 2.3), and (3) implementation of prototypes demonstrating proof-of-concept feasibility and providing encouraging results for cross-language query translation and extraction accuracy (Sections 3.1–3.2).

## 2 Architecture

In this section we propose an architecture for ML-OntoES and emphasize how this proposal provides the feature set and scalability required to support rich multilingual interactions. We begin by describing extraction ontologies (Section 2.1) and multilingual ontologies (Section 2.2). Then we discuss the multilingual mappings that connect different languages and locales in a meaningful way, thus making a multilingual ontology useful for a variety of information processing tasks, supporting users in their native locales (Section 2.3).

### 2.1 Extraction Ontologies

In general, *ontology* is the study of reality. More specifically, *an ontology* is an expression of a particular model of reality, including a specification of concepts, relationships among concepts, and constraints that exist in the model. An *extraction ontology* is an ontology that has enough information in the model to be able to drive the process of extracting concepts and relationships from some source document such as an HTML page or a PDF document.

Figure 4 gives the conceptual-model component of an extraction ontology that describes aspects of the *Restaurant* concept that an international traveler might be interested in exploring, such as price range of meals, menu items available, hours of operation, payment methods accepted, and tipping protocols.

The notation of Figure 4 conforms to OSM (Object-oriented Systems Modeling) [EKW92]. Names written in rectangles constitute concepts (*object sets*) of the ontology. Solid borders denote nonlexical concepts (e.g., *Restaurant* and *Rating* in Figure 4), while dashed borders indicate lexical concepts (e.g., *Address* and *Geo Location*). Lines between concepts denote relationship sets, and arrow heads mark functional associations. For example, in Figure 4 a *Rating* has at most one *Agency*, one *Value*, and one *Scale*, but an *Agency* may give many *Ratings* to multiple *Restaurants*. A triangle represents a generalization/specialization (ISA) relationship between object sets. For example, *Beverage* is a generalization that has two specializations: *Alcoholic Beverage* and *Non-Alcoholic Beverage*. The half solid dot on *Alcoholic Beverage* is an *object-set object* that represents the *Alcoholic Beverage* object set itself, so that by connecting *Regulations* to the object-set object we mean that regulations apply to the whole set of *Alcoholic Beverages* as a collection, not individually to each member of the collection.

The conceptual model in Figure 4 is only one part of the restaurant extraction ontology, namely the conceptual structure. The other part of the extraction ontology is a collection of *data frames* that describe the individuals, contextual clues, and keywords associated with—or signaling the presence of—concepts in

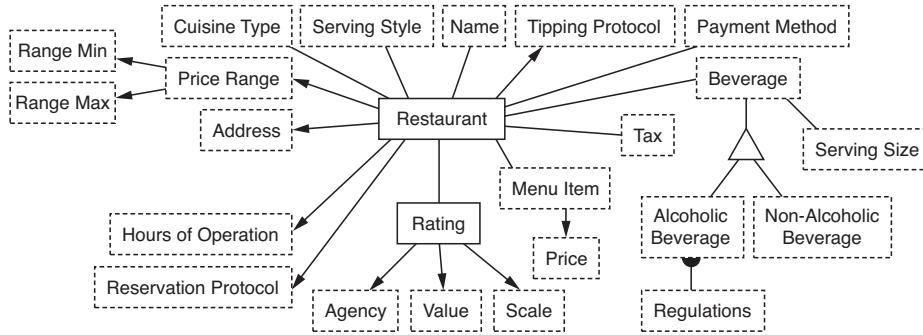


Fig. 4. Conceptual-Model Component of a Restaurant Extraction Ontology.

the ontology. We use a variety of techniques to encode data frames. For relatively narrow domains, lexicons can simply list the corresponding terms (e.g., *Payment Method*'s data frame could be a lexicon containing the names of credit card companies and other terms such as “personal check” or “cash”). For richer concepts we use regular expressions (e.g., *Price* would be difficult to enumerate, but a simple regular expression such as `\d+\.\d\d` can represent a large set of prices in a compact way). Contextual clues are also important for the data extraction process, and we again use lexicons and regular expressions to specify contextual details. For example, “\$” is a strong signal that a *Price* concept follows, especially if it matches one of the regular expressions of *Price*.

It is common to use OWL, the Web Ontology Language, to describe the details of an ontology. Numerous tools leverage the OWL standard for ontology creation and use. We use OSM because we have built a data extraction system, OntoES, around the OSM structure and the data-frame extensions that support data extraction. OntoES extracts data from ontologically narrow application domains with relatively high precision and recall, using ontology specifications that are robust with respect to different web sites or changes in document structure within the target domains. In order to interoperate with other tools and systems, OntoES is able to generate an OWL version an OSM populated conceptual model with RDF data instances queryable with SPARQL.

## 2.2 Multilingual Ontologies

In ML-OntoES, a *multilingual ontology* localized to  $n$  contexts  $\{C_1, \dots, C_n\}$  is an  $n+1$ -tuple  $\mathcal{O} = (\mathcal{A}, \mathcal{L}_1, \dots, \mathcal{L}_n)$ , where  $\mathcal{A}$  is a language-agnostic ontology representing concepts and facts in the ontology from a language-agnostic perspective, and each  $\mathcal{L}_i$ ,  $1 \leq i \leq n$ , is a localization<sup>8</sup> of  $\mathcal{A}$  to one of the  $n$  contexts.  $\mathcal{A}$  is

<sup>8</sup> We use “localization” rather than “language” because even within the same language there may be local variants we wish to capture (e.g., Australia uses both a different measurement system and a different currency than the U.S. even though both languages are English).

an extraction ontology that consists of a set of structural concepts (e.g., object sets, relationship sets, data frames) and facts (e.g., objects, relationships) that describe a domain of interest in a language-agnostic way. Each localization is a 4-tuple  $\mathcal{L}_i = (\mathcal{C}_i, \mathcal{O}_i, \mathcal{M}_{\mathcal{A} \rightarrow \mathcal{L}_i}, \mathcal{M}_{\mathcal{L}_i \rightarrow \mathcal{A}})$ , where  $\mathcal{C}_i$  is a local context label,  $\mathcal{O}_i$  is an extraction ontology,  $\mathcal{M}_{\mathcal{A} \rightarrow \mathcal{L}_i}$  is a set of mappings from  $\mathcal{A}$  to  $\mathcal{L}_i$ , and  $\mathcal{M}_{\mathcal{L}_i \rightarrow \mathcal{A}}$  is a set of mappings from  $\mathcal{L}_i$  to  $\mathcal{A}$ . Each concept in  $\mathcal{O}_i$  must map to a single concept in  $\mathcal{A}$ , but concepts in  $\mathcal{A}$  may map only partially to concepts in  $\mathcal{O}_i$  (i.e.,  $\mathcal{M}_{\mathcal{A} \rightarrow \mathcal{L}_i}$  is surjective, while  $\mathcal{M}_{\mathcal{L}_i \rightarrow \mathcal{A}}$  is injective).

The key idea of the ML-OntoES architecture is that each localized ontology maps to a central language-agnostic representation and vice versa. This “star architecture” avoids the  $n^2$  complexity of mapping each localized ontology to all other localizations, and instead provides a nearly linear scaling. Adding another localization involves constructing the localized extraction ontology ( $\mathcal{O}_i$ ) along with mappings ( $\mathcal{M}_{\mathcal{L}_i \rightarrow \mathcal{A}}$  and  $\mathcal{M}_{\mathcal{A} \rightarrow \mathcal{L}_i}$ ) to and from  $\mathcal{A}$ . In the process, it may be necessary to adjust  $\mathcal{A}$  so that all concepts in  $\mathcal{O}_i$  are represented directly in  $\mathcal{A}$ , and this may in turn require adjusting some of the mappings for other localizations. But since most mappings are trivial, the expected case is a linear effort required to add an additional localization to  $\mathcal{O}$ —indeed, sublinear since many language resources exist to aid in constructing the mappings.

It is customary to identify language and culture contexts by spoken language and country name such as German/Switzerland (de-CH) or Spanish/Guatemala (es-GT). But in general there could be many contexts associated with a given language/country pair, such as Swiss German/Switzerland in contrast to High German/Germany, or even tourist Spanish/Mexico versus business Spanish/Mexico. The concepts chosen for a particular localization may vary for many reasons. Ultimately, the precise meaning of “context” is defined by the author of the localization who expresses a selected set of ideas in a particular language. In our definition, we only need to note that a context has a chosen label,  $\mathcal{C}_i$  (though conventional locale labels such as “en-US” or “de-CH” could easily be used where appropriate). As a convention, we may replace  $i$  with  $\mathcal{C}_i$  when referring to elements of  $\mathcal{O}$ . For example, the English/U.S. localization  $\mathcal{L}_i$  could be designated  $\mathcal{L}_{en-US} = (\text{“en-US”}, \mathcal{O}_{en-US}, \mathcal{M}_{\mathcal{A} \rightarrow \mathcal{L}_{en-US}}, \mathcal{M}_{\mathcal{L}_{en-US} \rightarrow \mathcal{A}})$ .

For our running example, Figure 4 shows the English/U.S. (en-US) localization and Figure 5 shows the Japanese/Japan (ja-JP) localization. The language-agnostic component  $\mathcal{A}$  (not shown) is similar to these two.  $\mathcal{A}$  includes *Geo Location* (地理座標) from Figure 5 and *Range Min/Range Max* from Figure 4. Concept labels in  $\mathcal{A}$  can be written in any language or symbol system the ontology developer finds most useful. For example, the concept for *Geo Location* could be written 地理座標, *Geo Location*,  $C_1$ , or anything else the developer chooses.

### 2.3 Multilingual Mappings

Because the ML-OntoES architecture ( $\mathcal{O} = (\mathcal{A}, \mathcal{L}_1, \dots, \mathcal{L}_n)$ ) includes a central language-agnostic component ( $\mathcal{A}$ ) together with multiple localizations, mappings between  $\mathcal{A}$  and the various localizations are key to our approach. These

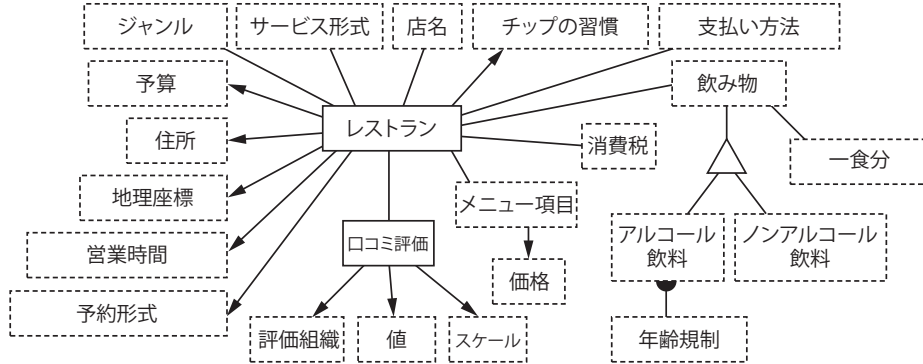


Fig. 5. Japanese Localization of the Restaurant Ontology Figure 4.

mappings fall into three categories: *Structural Mappings* that resolve differences among conceptualizations with standard schema integration techniques, *Data-Instance Mappings* that maintain correspondences among data instances, and *Commentary Mappings* that require standard language-to-language translation.

### Structural Mappings

Structural (schema) mappings between  $\mathcal{A}$  and  $\mathcal{L}_i$  are usually straightforward. For the applications we target, we anticipate most of them to be direct from  $\mathcal{L}_i$  to  $\mathcal{A}$  and partial from  $\mathcal{A}$  to  $\mathcal{L}_i$ . Fundamentally, this is because applications such as restaurants, items for sale, hotel and airline reservations, and many more all basically include the same concepts in the same relationship to one another. However, as guided by our earlier extensive work on schema mapping [XE06], we allow a full array of 1:1, 1:n, n:1, and n:m mappings along with operators such as split, merge, select, union, Booleanization, deBooleanization, skolemization, and lexicalization that carry data into structural variations.

Because it is so common to have identical structure, only with a little ingenuity were we able to provide illustrations. To illustrate that a concept in  $\mathcal{A}$  sometimes does not appear in some localization, we assumed that *Geo Location* does not appear in  $\mathcal{L}_{en-US}$  (as indicated by the gap below *Address* in Figure 4), but does appear in  $\mathcal{A}$ . And, to illustrate a non-1:1 mapping, we assumed that  $\mathcal{L}_{ja-JP}$  has no *Range Min* and *Range Max*, but rather just the more typical “budget” amount. Then, via a complex mapping, the system is able to convert the 予算 values in Figure 1 to the the *Price Range* values in Figure 2.

### Data-Instance Mappings

Data-instance mappings encode lexical-level snippets of instance information that are largely self-contained in nature and whose lexicalizations tend to be fairly direct across various languages. Various types of language resources serve

to mediate these differences. Thus, given some existing language-specific extraction ontologies,  $\mathcal{L}_1, \dots, \mathcal{L}_n$  and their mappings to and from  $\mathcal{A}$ , we can quickly assemble a new language-specific extraction ontology  $\mathcal{L}_{new}$  for ML-OntoES and mappings to and from it and  $\mathcal{A}$ . We identify four types of data-instance mappings: *Scalar Units Conversions*, *Lexicon Matching*, *Transliteration*, and *Currency Conversions*.

*Scalar Units Conversions.* One type of data-level correspondence assures conversion between items that are expressed with respect to some scale, for example measurements such as temperature, weight, length, volume, speed, and altitude. Different fixed scales exist for measuring such items, and these scales may vary by locale: much of the world uses the metric system, for example, whereas the U.S. for the most part does not. Conversion routines between measurement units and their associated values are straightforward to implement. A 5-3/8 oz. wine carafe will always have that measurement, and the value of an ounce is constant over time, as is its metric equivalent. We can thus store measurement values and associated units in a language- and locale-agnostic resource and convert it to any other format via simple arithmetic when developing and using a localized ontology. A wide range of such measurements exists across cultures and languages, of course. So the specification of conversion factors between such “exotic” measures (e.g. a stone for weight in English) may be necessary when localizing an ontology, but ML-OntoES supports this functionality.

*Lexicon Matching.* Another level of data mapping, this one more directly tied to language, has to do with the lexicon. Each language expresses concepts in its own combination of words, phrases, and other expressions. Often these terms correspond fairly closely, though this point has been debated among linguists. In cases where the correspondence or mapping is fairly direct, we can simply maintain a list of the crosslinguistic mappings. So, for example, the English word “meal” is a fairly close translation equivalent to the French word “repas”. Of course, there are word sense ambiguities: the English word “meal” in fact has several other senses, including one that means finely ground grain. This may complicate the storage of such correspondences, but for the types of data-rich web application domains we envision, the problem is not nearly as intractable as comprehensive modern dictionaries would suggest. In fact, several available technologies, such as termbase systems, software localization, lexical databases, and statistical machine translation assist in developing and maintaining crosslinguistic correspondences of this type, and the process scales well [LMN95]. ML-OntoES allows us to specify lexical information at this granularity in our ontologies and use them for finding and extracting data.

*Transliteration.* An even lower level of data mappings is necessary when considering a crosslinguistic context—that of transliteration. Proper nouns such as people’s names, place names, and company names generally do not vary much across languages, though language differences in terms of phonetics (i.e. individual sounds) and phonology (e.g. syllable structure and allowable phonetic sequences) are observable. For example, the name of Muhammad Ghadaffi has no less than 39 variant spellings in published English sources, and the surname



of President Bill Clinton has been rendered in more than 6 ways into Arabic in newswire. Tracking and identifying all of the proper nouns in any language is an important task, and various machine learning techniques and comprehensive language resources exist for identifying and cataloging them for any one language. Much more substantial, however, is the task of doing this across languages. While maintaining a crosslinguistic lexicon of proper names is possible, we are also able to take advantage of character conversion and phonetic transcription tools, perhaps with fuzzy matching, to compute these correspondences on-the-fly. The restaurant names in Figure 1, for example, were converted for Figure 2 by transliteration, and tools exist for automatic Kanji-to-English transliteration.<sup>9</sup>

*Currency Conversions.* Because of the evanescent nature of prices, referring to a price with an ontology, particularly a language-agnostic one, is best accomplished by storing the raw extracted value from the web page in question, rather than with respect to some idealized universal standard, which in this case does not exist. We are then confronted with the issue of converting this amount to other languages/locales when the user requires the price in another currency. The task would appear to be difficult, given the temporal variance of the conversion. Fortunately, because this need is so prevalent today, several web services are available that given a date, an amount, and source and target currencies, provide a conversion for the values in question. This precludes the need for developing and maintaining a conversion protocol. Since ML-OntoES supports web services, we are able to execute these conversions at query or retrieval time. When developing a language-specific ontology and retrieving associated information, we can call a currency conversion web service to compute the appropriate value.

### Commentary Mappings

Beyond these representational issues that impact how we specify and use correspondences at a linguistic level, there are larger-scale mismatches across cultures that must be addressed. For example, restaurants in different countries may have widely divergent requirements that customers need to be aware of, especially customers from outside the culture: tipping practices, how meals are structured, and even dress codes and the allowableness of pets on the premises. This type of information is best kept as short free-form notes or *commentaries* that are stored in  $\mathcal{A}$  and are available for scrutiny and elaboration by developers of language-specific ontologies. For example, the reservation protocol of a typical U.S. fine-dining restaurant might be described as, “*Reservations highly recommended, especially on Friday, Saturday, and holiday evenings.*”

When moving from one localization to another, translating commentary such as this can be quite valuable. Since there are web services that provide automatic natural-language translation (e.g. `translate.google.com` or `babelfish.yahoo.com`), it is possible simply to submit the commentary to a web service and request a particular language translation. Unfortunately, even though automatic translation technology has improved considerably in recent years, the quality of

<sup>9</sup> For example, see <http://nihongo.j-talk.com/kanji> and <http://www.romaji.org>.

automatic translation still varies immensely, and human review with correction generally gives the best results.

Since commentary is written with respect to a local culture and language, there cannot be a language-agnostic version of commentary. Thus, the nature of the problem dictates that commentary mappings (translations) must be provided for each additional localization added to a multilingual ontology. For example, if we start our restaurant ontology by creating a Japanese version and then adding an American localization, the ontology author of the American localization must translate commentary from the Japanese localization into English, and any new commentary from the American version into Japanese. In the worst case, this creates  $n^2$  mappings (where  $n$  is the number of localizations), but again since we have automatic translation services readily available, we get a base-level automatic translation essentially for free.

Nonetheless, high-quality mappings of natural-language commentary do require significant effort, often from a multidisciplinary, multilingual team. But as many web sites demonstrate, when the community receives significant value from a shared resource, it is possible to elicit from the community the team needed to create, enhance, and maintain that resource. Prominent examples include Wikipedia articles, Amazon book reviews, and TripAdvisor travel recommendations. We envision a “pay-as-you-go” approach where the system creates initial translations automatically, and experts from the community incrementally supply improved translations. Crucially, this does not adversely impact our extraction because it is not directly used for extraction purposes.

### 3 Evaluation

To show the feasibility and practicality of cross-language query processing, we describe an implemented early prototype of ML-OntoES and give some results of testing the prototype on independent-user-provided queries in Japanese, Chinese, and English (Section 3.1). To show the accuracy of cross-language information extraction and query processing, we give results for an initial Japanese/English cross-language application we have implemented for the car-ad domain (Section 3.2).

#### 3.1 Results from an Early Prototype

Based on extraction ontologies, we have developed a preliminary system, called *Pijin* [GT10]. *Pijin* accepts free-form, natural-language queries from mobile phone users in English, Japanese, and Chinese; maps queries to a restaurant extraction ontology; and reformulates them as form-based web queries to query four Japanese restaurant recommendation web services: Hotpepper, LivedoorGourmet, Tabelog, and Gournavi. These services return results in Japanese. *Pijin* also makes use of GPS information, Google maps, and other web services to provide “mashed up” recommendations to users.

For the experiment, we asked five subjects not involved with the project to make 100 queries for each language (Chinese, English, and Japanese). The subjects were asked to write free-form, natural-language queries that could be used to inquire about restaurants where they might like to eat. Typical of many, one of the queries posed was: クーポンのあるヤキニク屋さん、東京駅の近くに、予算は5000円 (loosely translated, “find me a BBQ restaurant that offers coupons near Tokyo station and my budget is under 5000 yen). Pijin interprets this query using the free-form query processor described in [AME07] and composes the web service query: `station=東京駅&coupon=0&food=焼肉&maxBudget=5000` which it then rewrites for each specific web-service API. This query produces a list of restaurants near Tokyo Station that offer menus priced under 5000 yen.

The system was able to correctly interpret and translate to interface form queries 79% of the Japanese queries, 72% of the English queries, and 69% of the Chinese queries. Pijin, for example, was unable to recognize and reformulate as a form query the Japanese query: アルコルの種類が多い居酒屋 (loosely translated: find me a bar that provides a wide variety of alcoholic beverages). Although Pijin recognizes 居酒屋 (“bar”) and correctly maps it to the restaurant genre, it can do nothing for アルコルの種類が多い (a “wide variety of alcoholic beverages”) because none of the web services has a parameter to accept this kind of search criterion.

### 3.2 Cross-Language Query Translation and Extraction Accuracy

To experimentally verify the feasibility of cross-language information extraction, we began with OntoES (our current data-extraction system) and made modifications to allow it to behave in accord with ML-OntoES. We call the version we implemented ML-OntoES'. For ML-OntoES' we added UTF-8 encoding, which immediately enabled us to build extraction ontologies in multiple languages and to process free-form queries in multiple languages. We were then faced with the task of constructing extraction ontologies for some domain in some natural language other than English. We chose the car-ads domain and the Japanese language—car ads because it is a challenging domain for information extraction and free-form query processing, but also because we have been able to make OntoES perform successfully in this domain, and Japanese because both the other languages, French and Spanish, for which we have near-native language abilities are too much like English for the testing we wished to do.

Having chosen a test domain and test language, we then took our existing car-ads extraction ontology and replaced the English concept recognizers with Japanese concept recognizers. To simplify crosslinguistic extraction, we limited the extraction ontology to six basic lexical concepts: *Make*, *Model*, *Price*, *Year*, and *Transmission*. To make ML-OntoES' work multilingually, we used the English car-ads extraction ontology as our language-agnostic extraction ontology (as well as its English localization) and we made the Japanese car-ads extraction ontology correspond 1-1 both structurally and for data instances. To make it correspond structurally for data instances, we extracted Japanese instances using

	ExE	ExJ	QiE	QiJ	QrEE	QrJJ	QrEJ	QrJE
Precision	.95	.89	.97	.92	.96	.84	.75	.87
Recall	.94	.91	.94	.82	.95	.84	.72	.85

**Fig. 6.** Experimental Results.

Japanese regular-expression recognizers, but immediately converted the resulting values to English (e.g., H12年 to the year 2000, 日産 to Nissan, and with the exchange rate \$1.00=JPY82.3). These conversions allowed us to further process data, converting it to RDF, and enabling us to query it with SPARQL, for queries generated by our already implemented free-form query engine [AME07]. Thus, we were able to interpret and process free-form queries such as H12年より新しい、50万円未満の車を探して and 全ての白い日産の車、価格、年式及び走行距離を見せてください which fared comparably to the English parsed queries to produced generic queries of the form: `Year, >, 2000; Price, <, 6050` and `Make, =, Nissan; Color, =, White`.

With ML-OntoES', implemented as explained, we conducted an experiment and obtained the precision and recall results in Figure 6 for Extraction in English on English car ads (ExE), Extraction in Japanese on Japanese car ads (ExJ), Query interpretation for free-form English queries (QiE), Query interpretation for free-form Japanese queries (QiJ), Query results for English queries on English car ads (QrEE), Query results for Japanese queries on English car ads (QrJJ), Query results for English queries on Japanese car ads (QrEJ), and Query results for Japanese queries on English car ads (QrJE). For the experiment, we used a collection of 100 English car ads taken from `craigslist.com` and 30 Japanese car ads taken from `Goo-Net.com`. For queries, we used 200 English free-form car-ad queries, which we had previously gathered from students in two senior-level database courses, and we manually translated 50 of them to Japanese free-form queries (see examples in the previous paragraph). We computed precision and recall for English and Japanese car ads by counting all the matches and mismatches the ML-OntoES' recognizers labeled for each of the six car-ad attributes in the two collection of car ads and taking an average over the individual attributes. For query interpretation, we counted the matches and mismatches for each ML-OntoES-generated constraint (e.g., `Year, >, 2000`). And, for query results we counted the number of car ads ML-OntoES' selected that were relevant and irrelevant over the respective document collections.

One of the interesting characteristics of the application we encountered was the problem of multiple years of interest in Japanese car ads. In addition to using 年式 and 製造年, which both translate as “model year,” most Japanese car ads on `Goo-Net.com` report 車検 (“shaken year”), which is a required and expensive smog, safety, and registration certification that can be transferred to new owners if it has not expired. As a further complication, what would be the year 2008 in an English localization, would be written as 平成20/2008年式 or H20/2008年 in the Japanese localization, where the first number preceded by

either 平成 or H represents the year 20 of Heisei, the current Japanese Imperial period. The second number, 2008, followed by 年, the Kanji for “year,” is its Julian-year equivalent. We overcame this difficulty partially by tuning ML-OntoES’ to recognize a single instance of year from both model year and *shaken* year, and to some extent, for the *shaken* year by recognizing the specific keyword for *shaken*. Application characteristics like these show some of the subtleties of implementing multilingual extraction and semantic search systems.

## 4 Conclusions

Our proposed multilingual architecture (ML-OntoES), with its central language-agnostic ontology and pay-as-you-go incremental design, along with our proof-of-concept prototypes and our initial cross-language extraction and query results, support the conclusion that cross-language information extraction and semantic search can be successful. Our results (reported in harmonic-mean F-measures) indicate the following: We can accurately extract in multiple languages: English ( $F = .94$ ) and Japanese ( $F = .90$ ). We can accurately interpret queries in multiple languages: English ( $F = .95$ ) and Japanese ( $F = .87$ ). We can query sites written in one language with queries written in another language: English query against Japanese source ( $F = .73$ ) and Japanese query against English source ( $F = .86$ ). The accuracy of these results is somewhat lower than we would like. We expect, however, that with the addition of accurate schema and data-instance mappings to and from localized extraction ontologies and a central language-agnostic ontology we can increase the accuracy. Currently within-language query accuracy indicates that this is achievable: English query against English source ( $F = .95$ ) and Japanese query against Japanese source ( $F = .84$ ).

As for future work, we intend to complete the transformation of ML-OntoES’ to ML-OntoES, and we intend to experiment with many different domains and several more languages. The results we have for car-ads English extraction reported here are consistent with previous results for the car-ads domain [ECJ<sup>+</sup>99]. And, since we have applied English extraction ontologies to a few dozen other domains with consistently good results (F-measures typically between .80 and .95), we can be reasonably confident that similar results to those reported here are possible. We must, of course, add and make use of instance mappings as defined here, so that we can boost the accuracy of cross-language information extraction and semantic search.

## References

- [AME07] M. Al-Muhammed and D.W. Embley. Ontology-based constraint recognition for free-form service requests. In *Proceedings of the 23rd International Conference on Data Engineering (ICDE’07)*, pages 366–375, Istanbul, Turkey, April 2007.
- [DHL04] Bonnie J. Dorr, Eduard H. Hovy, and Lori S. Levin. *Machine Translation: Interlingual Methods*. Encyclopedia of Language and Linguistics. 2nd edition, 2004.

- [ECJ<sup>+</sup>99] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-model-based data extraction from multiple-record web pages. *Data & Knowledge Engineering*, 31(3):227–251, 1999.
- [EKW92] D.W. Embley, B.D. Kurtz, and S.N. Woodfield. *Object-oriented Systems Analysis: A Model-Driven Approach*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.
- [ELL11] 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.
- [FRS<sup>+</sup>10] Achille Falaise, David Rouquet, Didier Schwab, Hervé Blanchon, and Christian Boitet. Ontology driven content extraction using interlingual annotation of texts in the OMNIA project. In *Proceedings of the 4th Workshop on Cross Lingual Information Access*, pages 52–60, 2010.
- [Gre98] G. Grefenstette, editor. *Cross-Language Information Retrieval*. Kluwer, Boston, 1998.
- [GT10] Z. Geng and Y.A. Tijerino. Using cross-lingual data extraction ontology for web service interaction – for a restaurant web service. In *2010 Workshop on Cross-Cultural and Cross-Lingual Aspects of the Semantic Web*, Shanghai, China, November 2010. Submitted.
- [KHF<sup>+</sup>01] Judith Klavans, Eduard Hovy, Christian Fuh, Robert E. Frederking, Doug Oard, Akitoshi Okumura, Kai Ishikawa, and Kenji Satoh. Multilingual (or Cross-lingual) Information Retrieval. In Eduard Hovy, Nancy Ide, Robert Frederking, Joseph Mariani, and Antonio Zampolli, editors, *Multilingual Information Management: Current Levels and Future Abilities*, volume XIV–XV of *Linguistica Computazionale*. Insituti Editoriali e Poligrafici Internazionali, Pisa, Italy, 2001. ISSN 0392-6907.
- [LFL94] Deryle W. Lonsdale, Alexander M. Franz, and John R. R. Leavitt. Large-scale Machine Translation: An Interlingua Approach. In *Proceedings of the Seventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-94)*, pages 525–530. Gordon and Breach Science Publishers, 1994.
- [LMN95] Deryle Lonsdale, Teruko Mitamura, and Eric Nyberg. Acquisition of Large Lexicons for practical Knowledge-Based MT. *Machine Translation*, 9:251–283, 1995.
- [MDL<sup>+</sup>06] Craig Murray, Bonnie J. Dorr, Jimmy Lin, Jan Hajič, and Pavel Pecina. Leveraging Reusability: Cost-effective Lexical Acquisition for Large-scale Ontology Translation. In *Proceedings of the Association for Computational Linguistics*, pages 945–952, 2006.
- [Sar08] S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.
- [TAC06] J. Turmo, A. Ageno, and N. Català. Adaptive information extraction, July 2006.
- [XE06] L. Xu and D.W. Embley. A composite approach to automating direct and indirect schema mappings. *Information Systems*, 31(8):697–732, December 2006.