Chapter 3 – Collation

The ordering of information, its collation, is very important in being able to retrieve it. Look for the Three of Diamonds in a shuffled card deck and you’ll have to look through the entire deck to find it. Look for it in a deck sorted by card and then alphabetically by suit, and you will know that it’s just this side of deck center. That labeling and ordering of the cards is its system of collation. It’s not just how we file and find things but how we make sense of things as well, and when it comes to the Internet, where we have trillions of things to be filed and found, it’s the foundation of ours and every other culture’s intellectual future.

For the last few decades, cultures have been scrambling to establish and standardize systems of collation for their languages. It’s not just so they can find their documents in Yahoo or Google. We saw some of this in the last chapter where we were organizing language input with the four corners system, radicals and strokes, and rhymes. Collation is important wherever there is data in any form. It’s finding products on shopping sites, hotels in tourist destinations, and movie show times.

There is far more to this than just the putting things in alphabetical order. It’s the way we label things as well. When you have a picture of blue flowers, is it “flowers”, is it “blue”, or is it the “meadow” where they were growing? It’s the question of how we describe the Three of Diamonds as a card as well as its relation to the rest of the deck. These are tricky questions even in Latin based languages, but how are they managing in Myanmar where their written language has no word boundaries, or with Urdu, 70 million speakers who until recently had no clear system of alphabetization?

Horrible Math

Collation is the ordering of written information into a standard order. The most common form of collation in the west is alphabetization. Most alphabetic languages have a set alphabetic order although sometimes it can seem to be a bit arbitrary. For instance some, like German which has phone book and dictionary order, have more than one. French uses the same alphabetic order as English; however some letters have extra marks such as é, à, and ô, but are not considered distinct. In Swedish though, they are. They’re tacked on after “z”. Sometimes groups of letters are counted as single letters as in “sch” in German, which comes just after “s”. They are, as long as they are consistent, all valid systems of collation.

One of the most basic ways of collating things is numeric ordering where lists of numbers such as \{1, 23, -3, 10\} collate into ordered sets like \{-3, 1, 10, 23\} based on some rule, in this case something like \(X = (x, y) | x \leq y\). The numbers can represent anything. In the U.S. for instance, we have social security numbers that represent US workers. Cars have license numbers. ASCII is a collation sequence for English letters and symbols and has been used itself, usually with some modification to adjust for capitalization, as a collation sequence in ordering databases. Unicode is an even bigger collation
sequence of human written symbols. Although assigning things random numbers does constitute a
collation, it isn’t a very useful one. It’s better if the numbers reflect some aspect of the data that you’re
interested in, such as the ranking of sports teams.

Collation is not to be confused with classification or categorization, which is the grouping of things into
categories. Collation is only concerned with the ordering of things, although that ordering can be used to
create categories. For instance the Dewey Decimal System used in libraries is a categorization. It divides
book collections into categories. It does not innumerate individual books. Collations allow you to search
for things and define ranges of things, such as the first five elements of . . . , the largest element in . . . , or
the elements between A and B.

In ordering things each element of the set has to compare itself to the set as a whole to find its particular
place within the set. The important thing to realize is that it’s the comparison function itself that defines
the order. For instance, in numerical ordering each element looks at its neighbors and says “am I bigger
than the guy to my left?” If it is, then it swaps places. If they all keep this up, eventually each element
will reach a point of equilibrium where none of the elements can rise further. The set will be in ascending
order. Change the function, for instance reverse the direction of the comparison, and you will change,
after a great deal of jostling as each element climbs downward, the order of the set to descending order.

There are some rules that a function must satisfy in order for the resulting set to be considered a valid
collation. In mathematics a collation is considered to be a total order. In order to qualify as a total order
it must satisfy some conditions. First, the ordering rule must apply to all elements in the set, it must be
mathematically total. If you skip something, then it cannot be considered to be part of the resulting
collation. For instance you could create a collation of mammals from the set of animals. The collation is
only orders the set of mammals. From the point of view of the collation, it’s as if the other animals don’t
exist.

The ordering rule must consistent or mathematically transitive. If A comes before B and B comes before
C, A must therefore come before C. The rules can’t change somewhere in the middle of the set. Lastly,
the most abstract property it must have is antisymmetry. Antisymmetry means that the ordering rules
must apply to A and B or just to A itself, but it can’t apply to B and A. What this means is that the order
can only go in one direction, it can’t go backwards. 1 before 2 and 2 before 3, but never 3 before 2 or 1.
Satisfy these conditions and you have a certified bona fide collation.

**Apples and Apples**

When we are setting up a collation sequence, we must think about the different aspects of objects being
collated. What do they have in common? There are an infinite number of ways any object can be
described, but if we’re going to create a collation we must find a type of description that is common to all
objects within that sequence. Think back to Ancient Egyptian. We found that it was possible to create an
ordering table based on pictographic content, birds with birds and people with people. We also found that
we could order pictographs by sound of the words they represent.
Our choices of aspects of data that we use to build collations are based on the relationship between the information we have at the start of our queries and what we will want when we use our collation. We have a name, we want a phone number. We have a title, we want a book location. These relationships are obviously linear, but sometimes the linearity of the relationship isn’t so obvious.

Search engines build their collations based on keywords supplied by web documents or derived from their titles or content. These words are treated as individual atoms of information and are taken together to form a document key, a signature that identifies that document. For instance an auto repair shop might have a set of keywords like: auto, car, repair, lube, tire, oil, etc. It’s their association to particular documents that is the key that identifies the auto shop’s web page. Having one or more matching keywords in the search is the function that builds the initial collation. The results though are returned sorted by the computed weight, relevance, or rank of the relationship. It’s how well the keywords being searched for fit with the keywords associated with the document. Did you match one, two, or more keywords? We have two collation sequences at work here, the first producing the data for the second.

**Apples and Oranges**

Anyone who uses the web encounters keyword searches almost every day. They’ve become the ubiquitous method of finding things on the web. But sometimes the data we have isn’t exactly compatible with the data we are searching through. Can we enter “apple” into Baidu, a Chinese search engine, and get meaningful results? The Chinese word for apple is 蘋果. Are they the same? The simple answer is currently no. 蘋果 and “apple” are different sets of code points. Chinese speakers and English speakers both visualize the same thing when they see their written words, but to a computer they are just strings of code values and are completely different. For a database to service both languages, it would have to store both words. For a database to service the entire world’s languages it would have to store thousands of versions of each word.

This is a huge problem. Currently we avoid it by dividing up the world by major languages using different sites with different databases so we don’t have to store ten versions of each word. For instance there are English, French, German, Japanese, and Chinese Amazon.com sites.¹ The dominant versions of the languages that are spoken in the areas of the world that Amazon has chosen to focus its marketing. But this is hardly comprehensive. By various counts, there are at least 165 different spoken languages in current use within the United States alone. In Papua New Guinea there are

¹ Amazon.com, Amazon.fr, and Amazon.de
And what does this do to less spoken languages and regional dialects when the only way they can participate in the Internet is to adopt one of the dominant languages?

So we segregate the world’s data into separate sites or limit the content of our sites to a limited word set that we can keep multiple translations of. But how can this technique cover the thousands of languages humans speak or store the thousands of versions of each word or phrase and still deliver proper service to each? To be blunt it doesn’t. There is no current solution to this problem. However, where there are problems there is also opportunity.

Where the big companies try to service as many as they can, often poorly, regional providers can target the specific needs of specific locales. Localization is the counterpart of internationalization, the focus on the individual rather than blanketing the whole. There are, for instance, more than 40 Arabic search engines based in more than 14 countries each catering to regional needs not serviced by the global search engines. As you should know, Arabic is not a homogenous language but has diverse regional dialects that are very different from one another. Although these search engines are Arabic, they service distinct populations.

The more local you become the more specific your key terms can be in your collations and the more usable your database can be. On the other hand, you may have fewer people who can use it. Local search engines may lead you to local delicacies, but then not all your customers may understand that they found them when they see them.

This problem has been mitigated to a small extent by the marriage of translation programs and search engines. Google’s translation services claims to be able to translate 135 languages, including Klingon. If you’ve used these services though, you’ll know that they are not very accurate, and on occasion completely unreadable. The problem is that different languages do not have a one to one relationship. Words in one language are missing in others or have synonyms with subtle differences. Euphemisms often based on jokes or shared experiences often cannot be translated to other cultures without a great deal of thought and explanation.

Problems of localization become especially acute in regions like Europe, where there are many cultures close together without a single dominant language. Bureaucrats in the European Union sometimes encounter great difficulty in obtaining translations of important documents in their preferred language. Human translation is tedious and expensive. Machine translation is a seductive alternative and research is always ongoing.

However, in the case of collations, exact translation may not always be necessary. A “lossy” query may be acceptable. Keyword description of information is not very accurate to begin with. Think of the last novel you read and then try to think of all the keywords someone might use to look for it. It’s almost certain that you won’t think of them all. Libraries focus their queries by limiting the indexes that you can

---

2 Top 20 countries by number of languages. Vistawide World Languages and Cultures. http://www.vistawide.com/languages/20_countries_most_languages.htm


4 MOLTO, Multilingual Online Translation. http://www.molto-project.eu/
search against to certain types of words such as author or title. General purpose search engines though, do not have that luxury.

What if you could create a kind of a common key, one not tied to any particular language? You could translate your search terms to that intermediate language before the search is performed and then you could create a non-language specific search. The problem is that most people want documents to be in their own language. Sites in foreign languages are often not very useful, so language itself must be an important part of the search itself.

**Unicode Collation Algorithm**

It should be obvious now that there is more to sorting things than just putting them in alphabetical order. Ligatures and compound characters are composed of multiple code values that count as single characters so you can’t sort by just code value. Multiple characters, such as “ch” in Spanish, count as single characters in sorting so you can’t just sort by how the word looks on the screen. The value of case varies from language to language and, as we will see in the next section, the order of characters within a word can alter their collation sequence. And clearly the difficult of all, you may need collations based on mixed languages where you must reconcile two or more sets of language rules. Clearly we cannot, for instance, rely on the ordering of characters within Unicode to sort our data.

The Unicode collation algorithm was developed to address this problem. The algorithm is as complex as the problem it seeks to solve. It breaks the assessment of characters or character groups down into weighted “layers”, generally at least three deep, using language specific information from the Unicode Common Locale Data Repository or CLDR. CLDR entries are written in Local Data Markup Language and are loaded by the operating system depending on languages selected. Characters are given a computed weight based on data from the table, which can be used in sorting. In cases where multiple languages are involved, because it is their weights that will be sorted instead of the characters themselves, the data will be interleaved independent of the language.

For example, Unicode defines a Default collation ordering the, Default Unicode Collation Element Table (DUCET)\(^5\), however this is not sufficient to address the needs of the European Union and the European Free Trade Association. In response they are developing a more advanced rule standard, the European Ordering Rules (EOR)\(^6\). It has four layers. The first, which take precedence, orders the basic letters:

- **Latin:** a b c d e f g h i j k l m n o p q r s t u v w x y z þ
- **Greek:** α β γ δ ε ζ η θ ι κ λ μ ν ο π ρ σ τ υ ψ ω θ
- **Cyrillic:** а ä э ё е є й ж з с и й ъ ы ў ӱ ӳ ү ұ ф х ҳ һ ц џ

Layer two deals with diacritics:

\(^5\) ISO/IEC 14651:2007  
\(^6\) EN 13710
The others are ordered as variants of their base letters. Level three deals with the treatment of capitalization. Level four concerns punctuation and the treatment of whitespace.

UCA only works with Unicode data. Some databases will incur conversion costs from other character encoding schemes; others will just throw errors if non-Unicode data is used. What is important is that the encoding format of the data must be considered when selecting a collation scheme. Check the specifics of the collation scheme, what algorithm it uses, what language rules it supports, and does it support mixed languages, before selecting it.

Cultural Standardization

Up until now we’ve talked about standards expanding and evolving in an attempt to fit the world. This is only possible up to a point after which the cultures themselves must make an effort to fit in or be left behind. Up until the end of the last millennium Urdu, one of the two official languages of Pakistan and one of the 22 official languages of India, had no set dictionary order. There were at least eight, with many Urdu dictionaries being hand copied or even typewritten. Urdu typewriters and printing have been common for over a century and several Urdu keyboard standards have been in place for over a decade, but the effort to set a collation order, so important for information storage and retrieval, didn’t start until the turn of the millennium.

Part of the problem with any standard is that, like most major languages, Urdu itself has many variants. These are caused by geographic divisions or mixing with other local languages. While most Urdu speakers live in Pakistan or India, Urdu is spoken widely in the Middle East, Europe, and North America. Written Urdu uses Arabic characters in its alphabet, but is linguistically closely related to Persian and Hindustani. It’s because of the geographic spread of the language, with no true central homeland, that it has been difficult to create a single standard. But with the new millennium, Pakistan has stepped forward to bring Urdu to the Internet.

Starting in 1996, a national effort was initiated in Pakistan to implement a set of national coding standards for computing in Urdu, including collation. In 2001 the Urdu Zabta Takhti standard for character encoding and collation was adopted. UZT was based on a 256 character code page at a time when code pages were well on their way towards being replace by Unicode. As such, the first efforts lacked the depth necessary to become a comprehensive standard. And as we will see, collation in Urdu is highly
complex and a simplistic numerical ordering was clearly inadequate and work has continued on a replacement standard.

Urdu has many features that make it difficult to sort. It has many diacritics, such as the hemza from Arabic, which must be dealt with in the same manner as accents and umlauts are in European languages. However there is still debate as to which characters are letters or diacritics. In some cases this may even depend on their placement within the word. Dealing with the many diacritics requires Urdu to use a two pass sort. The first pass sorts the root characters, the second identifies and sorts the diacritics. Because Urdu uses the Arabic alphabet, Urdu code points are included in the Arabic section of Unicode. There is Urdu collation support in the CLDR and the Unicode Collation Algorithm does currently work, but debate continues about the details of the rules and continues to be updated to this day.

These decisions about collation standards were made in Pakistan and accepted by the Unicode consortium. But only 14% of Urdu speakers live there. Pakistani Urdu contains many elements of Pashto, Punjabi, and Sindhi that other speakers of Urdu might find foreign. Despite this, Urdu speakers in the future will be seeing this new sequence in their dictionaries and search engine results without any question of consent being raised. The ultimate effect will be to make Urdu and the perhaps the people who speak it too more homogeneous, but at a loss of regional dialect.

**What is a Word?**

There are conventions unconsciously adopted by the Western creators of the Internet which do not apply the world over. Most of the alphabetic languages of world use a space to denote the boundary between words. The ability to isolate and use individual words is for most Westerners an unconscious assumption. The design of our databases is founded on it. However, the Myanmar Language, spoken by over 32 million people and formerly known as Burmese, does not separate their words. To be clear, you may see spaces sometimes in modern text because some authors will insert them as a convenience; it is often in a haphazard manner. Burmese does not naturally use whitespace, which poses a big problem in a word-oriented Internet. In older text you will not see spaces at all.

<table>
<thead>
<tr>
<th>noun</th>
<th>verb</th>
<th>part.</th>
<th>noun</th>
<th>part.</th>
<th>adj.</th>
<th>part.</th>
<th>verb</th>
<th>part.</th>
<th>part.</th>
<th>part.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Four Eight Uprising</td>
<td>happen</td>
<td>When it occurred</td>
<td>people</td>
<td>count</td>
<td>r word</td>
<td>three thousand</td>
<td>approx</td>
<td>die</td>
<td>past tense</td>
<td>plural</td>
</tr>
</tbody>
</table>

When the Four Eight Uprising occurred, approximately three thousand people died.\(^7\)

\(^7\) Based on an example given in Wikipedia, http://en.wikipedia.org/wiki/Burmese_language
Burmese is classified as a mono-syllabic or isolating language and does not yet have a system of Romanization. An isolating language has a low morpheme, the smallest component in a word, to word ratio. Burmese words tend to be composed of single morphemes. An English example might be “boy”, which is composed of a single morpheme “boy”, a 1:1 morpheme ratio. The Russian word “protivopravitelstvennyi” consists of six morphemes, a 6:1 ratio.

To make things complicated, it’s not a pure isolating language. It does have compound words:

- \( \text{head} \rightarrow \text{pack} \rightarrow \text{hat} \)
- \( \text{language} \rightarrow \text{look, see} \rightarrow \text{building} \rightarrow \text{library} \)

These make it difficult for a programmatic parser to differentiate between the words “head” and “pack” or “library” because without word boundaries it could be read either way. The meaning is implied by the context. It also has loan words, foreign words which are spelled out phonetically:

- \( [\text{kun pju ta}] \rightarrow \text{computer} \)
- \( [\text{hsa ' ko ma ti}] \rightarrow \text{sub-committee} \)

Interestingly, Burmese doesn’t have adjectives. Instead it has verbs that carry the meaning “to be X”, where X is an English adjective. These verbs can modify a noun by means of the grammatical particles \( \_ \) or \( \_\_\_ \) in literary Burmese. They take the form be + verb + \( \_\_\_ \) + noun. This will be something we will revisit later when we discuss the semantic web.

The Myanmar Language, according to one dictionary, has over 32,283 different cataloged syllables and it is around these that efforts to create a collation sequence have progressed. The language has excellent alphabetic ordering, so if it can be broken into atomic syllables, then these could be ordered and would make a reliable key for collation. However, this is a problem that has not yet been satisfactorily solved. There are several approaches being researched. The most of them rely on the fact that syllables in the Myanmar Language follow a certain pattern, generally diagrammed like this: C(G)V((V)C). “C” is a consonant, each syllable starting with an onset consonant. “G” is a glide or semi-vowel. They’re like vowels, only shorter. “V” is a rhyme, the last part of the word that we usually latch on to when we rhyme words. Burmese also can in some cases have an extra rhyme and certain particular consonants tacked onto the back of the syllable.

There are three classes of approaches to slicing sentences apart: pattern or rule based, syllable matching, and neural net algorithms. All of these approaches have so far achieved levels of correctness above 95%. One complicating factor is that the Burmese character set is still in a state of flux with 9 new characters, mostly punctuation, being added in 2006. Shown below is an example of a rule based strategy proposed in 2008.\(^8\)

<table>
<thead>
<tr>
<th>Category</th>
<th>Name</th>
</tr>
</thead>
</table>

\(^8\) Drs. Maung, Mikami. A Rule-based Syllable Segmentation of Myanmar Text.
States 1, 3, 4, and 5 are possible end of syllable exit states. At best, systems like this can only generalize. There are always exceptions in any language and misspellings pose problems. Wouldn’t you hate it if your business name happened to have one of the syllables that couldn’t be parsed and no one could find you on the web? One of the important metrics used in judging these algorithms is how well they recover from unintelligible, misspelled, slang, new and made up words, all precisely the kinds of things marketing people love to add to every language.

Microformats

The collation of documents, especially in the Internet, use keywords provided in document metadata or derived from document content. These provide only a single one-dimensional collation, a general description of content. Sometimes though, documents are more than just a single thing. Often they are containers, composed of many elements that are themselves worthy of collation.

In traditional web documents, information is portrayed as pictures and text. There is no actual meaning in the data itself. It’s up to the viewer to decide if what they’re looking at is an address or a laundry list. You couldn’t expect to find bread recipes buried in blog posts because the blog site only provides generic blog site oriented keywords to the search engine, not blog post content information. The blog posts themselves could be almost anything.

However, if you could identify your posts as recipes or movie reviews in a systematic way, then the search engine could catalog them. Those blog posts would carry meaning as well as content. Then the search engine could create collations that just deal with recipes, news, resumes, or even human relationships. These collations would be smaller and faster, and could be kept in separate machines spreading the workload out, increasing reliability and response time. And they would make available information normally buried.
This is giving the information we see a context, a place within the larger scheme. XHTML and HTML, the language used to define web pages, allows for the embedding of metadata within their tags. Metadata is data that describes data. Microformats take advantage of this to insert their descriptions of the data being presented. For instance, contact information might be presented like this in a normal web page:9

```html
<div>
  <div>Joe Doe</div>
  <div>The Example Company</div>
  <div>604-555-1234</div>
  <a href="http://example.com/">http://example.com/</a>
</div>
```

With microformat markup this becomes:

```html
<div class="vcard">
  <div class="fn">Joe Doe</div>
  <div class="org">The Example Company</div>
  <div class="tel">604-555-1234</div>
  <a class="url" href="http://example.com/">http://example.com/</a>
</div>
```

The browser can identify this information as an address which can be selected and added to the user’s address book should they so choose.

At present there are a sizable number of microformats defined, however browser plugins are required to recognize them. Microformats are not a standard but are instead defined by a community of users led by Commerce.net, a non-profit organization, and they are not the only effort working along these lines. There is ContextObjects in Spans, or COinS and the Microdata proposal in HTML 5. What is important here is that there are major efforts in the works to identify and group data within documents in a standardized way. By classifying data we no longer have to rely on arbitrary keywords dependent on the nationality, spelling ability, and literacy, of the authors. If we can link documents and the objects within them to some abstract meaning, then we may be able to begin to create collations that are based on relationships between them rather than just arbitrary subjective labels.

**Ontologies**

In the beginning of the Web, Web 1.0, we saw documents described in HTML. They were static, connected together in a web using links which, when selected took you from page to page. The next generation web, our current generation, saw the introduction of user participation to the web. The web page became an interface with which the user could interact. Social-networking sites, blogs, video

sharing sites, mashups, and hosted services are all examples of Web 2.0 sites. But what will the Web 3.0 look like? Many feel it will be the “Semantic Web”.

In microformats we saw information being grouped and indentified, tied to common elements of abstract meaning. The vCard class, for instance groups information such as city, name, and phone number together as an address. A city, in addition to being a city, has the additional property of potentially being part of an address, a “has a” relationship. The Symantec Web takes this a step further. It supplements the simple markup text with descriptive data stored within the document in its own separate markup languages called RDF and OWL.\(^{10}\) A document may contain more descriptive data content than it does flat content. It’s from this descriptive data that collations are formed.

Elements of knowledge, like the city in an address or the address itself, are called “atoms”. They’re kernels of fact called classes, entities, or concepts, the term depending on the background of who you are reading. Relating these atoms to each other is “fact”. “The sky is blue”. Sky and blue are atoms and “is-a” is a class of relationship. This is called predicate logic. You can ask the data questions by stating a fact, but replacing one of the atoms with a variable such as “the sky is X”, which will return everything with the atom sky and an is-a relationship, including “the sky is blue”.

Because different relationships can connect to the same atoms, “the sky is blue” and “my VW is blue”, they form a web – a “universe of discourse” or “ontology”. Does the computer know what these relationships and atoms mean? No, it just keeps lists of them. Any meaning they have comes from human readable text, “definition” or “annotation”. Those definitions are stored in the text of the document and in the text of documents out on the web.

All the descriptive markup code in the document is there to identify the atoms and relationships within the document and to relate them to atoms defined outside on the web. Atoms out on the web are identified using a URI, or Uniform Resource Identifier. URL’s are subset of URI’s. So “blue” might have its own definition out on the web and when you talk about blue, you will be relating your knowledge to all the other discourse on blue. For instance, the definition for integers is at http://www.w3.org/2001/XMLSchema#integer and there is a definition for “iron” is at http://www.daml.org/2003/01/periodictable/PeriodicTable#Fe. If you’re into philosophy, you might say it’s a bit like a Platonic form.

In current semantic web implementations these relationships have to be flat lists or hierarchical tree structures, otherwise you can have circular paths which can cause problems with search functions. In the real world though, without some yet to be discovered very strict control, circular paths are inevitable and natural. For instance you might have relationships like “blue cars”, “pictures of cars”, “pictures with blue”. A query like “I want pictures of blue cars” could cause a logic loop and a Star Trek style explosion!

You might notice as well that the expression “pictures of cars” is in English. All current ontology mechanisms, like web 2.0 keywords, are language centric. They rely on keywords in a particular language. There can be French ontologies, Russian ontologies, and Chinese ontologies. Other than

\(^{10}\) Resource Description Framework and Web Ontology Language. OWL instead of WOL for ascetic reasons and because Owl in Winnie the Poo spell his name “Wol”.
creating a more complicated way of keeping keywords, have we really improved anything if we can’t relate them? Isn’t blue the same as blau or الازرق؟ And even if we both speak English, how can we know that the blue I’m referring to is the same as yours?

**Natural Language**

The semantic web is still in a very primitive state. Ontologies are discrete\(^\text{11}\) from one another, a bit more like individual XML databases. OWL isn’t really any better than the language Prolog, invented 40 years ago. Current implementations don’t use natural language, although they intend to eventually. They use complex query languages like SPARQL, a query language for ontologies, which is based on predicate logic with queries that look like:

```
PREFIX table: <http://www.daml.org/2003/01/periodictable/PeriodicTable#>
SELECT ?name
FROM <http://www.daml.org/2003/01/periodictable/PeriodicTable.owl>
WHERE
{
  ?element table:name ?name;
  table:atomicWeight ?weight.
}
ORDER BY DESC(?weight)
LIMIT 10
OFFSET 10
```

This is not a query method that can be commonly used. But, one of the promises of semantic web is the ability to use natural language, or at least something like natural language, to pose questions, as you would your local librarian, and expect back lucid answers.

In order to accomplish this, ultimately ontologies will have to be connected together into webs with references that branch outside themselves. Instead of just saying “blue”, you might reference another blue somewhere else on the web, a blue that’s like the blue you’re describing. It might be a color swatch in a standard blue in a color library, a picture of a field a blue flowers on a clear day, or a description of a “blue moon,” which isn’t really blue at all. This is where the “semantic” in the semantic web comes from. And blau or الازرق\(^\text{2}\) or الازرق can all refer to common references because in the mind of their authors, they all mean the same thing.

\(^{11}\) You can import ontologies into ontologies to create huge compound ontologies, but this is not conducive to quick searches.
One approach being tried in order to find commonality in expression is to break sentences into simple logical fragments. Consider the sentence below:

“Yesterday I saw Beowulf. I liked it very much. It’s a fantasy movie with lots of actions and good special effects.”

By following a few rules, such as “and” results in several statements and “much”, “good” translate as subclasses, this could be written in a pidgin form:

“Yesterday I see Beowulf. I like it. It is fantasy movie. It has much action. It has good special effects.”

Although somewhat ugly, this form is still readable by humans and could translate into most languages. It’s also more capable of being processed by computer. Of course words like “a few rules” only work for this one sentence. In the face of human language the task of creating a comprehensive library of parsing rules has so far proven to be difficult. And they wouldn’t be the same for any two languages. Remember the lack of adjectives in Burmese and verbs being used as adjectives? Even identifying the individual words, not just parsing them into simple sentences that can be evaluated has proven to be very difficult. And then consider the ambiguities posed by the words themselves. “Action” for instance can mean many things. What did the author mean by it? Each human word is a blur of linked meanings.

Despite the difficulty, the promise of creating cross language collations that can account for the meaning of documents as well as the normal arbitrary keywords is important, especially in areas of the world where large numbers of languages overlap. The European Union is the biggest investor in semantic web research. They spent over €300 million in 2010 for semantic web research and are expected to top €700 million in 2014. This is not just for development of the web itself, but the all important applications to exploit it as well.

Creating a document in the semantic web will involve explaining it in terms of its place within the web. You will have to relate it to similar items in order to explain its purpose and meaning. This could double or triple the cost of creating a properly indexed document. But this is the cost of creating the precise meanings that can help us cross national boundaries.
Conclusion

The finding of uncatalogued and unenumerated sections of the world is the work of the entrepreneur and the bureaucrat, but clever art of finding that point of commonality in any set of data in order to build a collation is the work of the information engineer as he or she works to piece the disparate parts of the world together.

As our population grows, so grows the complexity of our civilization and our need to organize. Efficiency and organization are the key to its survival. The information necessary for our day to day existence can no longer be passed from one person to the next, but must be gleaned from an expanding network of sources. Finding that information we need in a timely manner becomes more difficult. Where before we relied on collations of local businesses in our phone books, we now search the world and in some databases, the entire known universe, on the Internet.

The Internet is a vast database of documents, the largest filing cabinet in history. To find things in it requires the most complex systems of collation ever devised by humankind. Because this collation is world spanning it is also world unifying and, as we have seen, if any culture is to be part of that mainstream - it must be part of it. The danger that modern technology will leave any culture behind is just too great for that culture to risk. Although the Internet is flexible and it changes to accommodate as it attempts to catalogue literally everything, it’s not omnipotent. Cultures too must change if they are to fit in.

Entire civilizations are rewriting their languages, standardizing in order to fit in. Although they retain their local cultures, their localization, they are joining a larger culture whether they know it or not. They are joining the Internet culture.
Figure 1: Wikipedia’s front page. [http://www.wikipedia.org](http://www.wikipedia.org)

Figure 2: The distribution of Urdu within Pakistan and India. Urdu Center for Language Technology and Instructional Enrichment. Indiana University Bloomington.

Figure 3: Burmese sentence structure. Modified from an example in Wikipedia, Burmese Language. [http://en.wikipedia.org/wiki/Burmese_language](http://en.wikipedia.org/wiki/Burmese_language)


