A Platform for Scalable, Collaborative, Structured Information Integration

Kurt Bollacker, Patrick Tufts, Tomi Pierce, Robert Cook
Metaweb Technologies, Inc.
631 Howard St.
San Francisco, California 94105

Abstract

Freebase is as collaboratively created and edited database of general, structured information intended for broad public use. It is designed to scale to a large number and diversity of users and data. Both an AJAX/Web based user interface and an HTTP/JSON based API are provided to allow use by and collaboration between both humans and software. In particular, these interfaces allow for an "emergent workflow" of collaborative, distributed data structuring and entity reconciliation, even if individual users are not intentionally cooperating in such information integration tasks.

Introduction

The World Wide Web has become the de facto system for dissemination and organization of information. While ideally all knowledge on the Web would be consistently and uniformly structured for easy access, the actual Web varies widely in its structure, representation, and quality. This inconsistency often makes access of the Web difficult for the purposes of sophisticated data analysis, search, organization, and even display. Some of the specific problems include:

- Intrinsically structured information is "trapped" in unstructured documents, unable to be easily read by automated processes.
- There is a dearth of canonical references to entities (e.g. no URIs for people). There are often multiple, unconnected representations of the same real-world entity.
- There is often a lack of separation of meaning from presentation. Most unstructured and many structured data sources mix semantics with display information.
- Often, there exists a heterogeneous representation of information across data sources and even within a single data source across time.
- Web-based information is often presented without the explicit context of other, potentially helpful information sources. This is a condition arising from the one-way nature of Web links.

Although most Web pages lack the intrinsic features to be a good foundation for addressing these problems, there have been a few large scale research efforts to add such features or create augmenting data systems to the Web as a whole. One of the best known is the Semantic Web (Berners-Lee, Hendler, & Lassila 2001); a framework toward adding rich, deep, structured semantics to Web pages and information repositories. The Semantic Web has evolved to include a set of standard formats, protocols, and organizational conventions that allow the creation of ontological structures and the organization of Web-based information using those structures. The hope of this effort is that "meaning" rather than keywords can be used to index and organize information. While not originally created to be so, the Wikipedia (Wikipedia 2007), is another major contributor in efforts toward general Web based information structuring. This collaboratively generated encyclopedia has inspired research efforts in such directions as entity extraction (Auer & Lehmann 2007), search disambiguation (Strube & Ponzetto 2006)(Gabrilovich & Markovitch 2007), and semantic structuring (Krotzsch, Vrandecic, & Volkel 2005). There have also been commercial services that allow users to create Web-based data that is intrinsically structured (e.g. (GoogleBase 2007)(DB 2007)), but these usually do not emphasize tools for entity reconciliation, data sharing, and collaborative editing between data sets.

In much of the research toward bringing structure to the Web, there is a common view that solving large scale information integration, entity extraction and data reconciliation problems requires systems that automatically perform structuring, extraction, and reconciliation tasks over large, messy, already existing data sets. In this view, the workflow of information structuring tries to extract, impose, or augment structure on the existing Web, but the original data on the Web itself is considered immutable because control of such data is highly distributed and uncoordinated. The Semantic Web does provide mechanisms to feed structure back to original Web data, but its decentralized nature brings challenges of uneven adoption, implementation difficulty for non-technical users, and query performance issues arising from its inherently distributed structure.

Freebase

Addressing some of the same large scale information integration problems that the Semantic Web and other similar efforts, we have built Freebase, a collaborative database of structured general human knowledge. The features of Freebase are designed to facilitate high "collaborative density" among its users in the organization, representation, and integration of large, diverse data sets. In the abstraction of Dataspaces (Franklin, Levy, & Maier 2005), Freebase may be thought of as a DSSP with features to increase "Semantic Integration" while allowing a range of near and far "Administrative Proximity". Freebase has an AJAX-based client for human users and a JSON/HTTP-based API for machine "users" so both can work together both tightly and asynchronously to create and edit complex, evolving, structured data. Specifically, Freebase consists of the following components and features:

- **A Graph-Shaped Data Store:** This is a scalable, tuple store with some built-in query planning and optimization capabilities, which allow deep, naively constructed, tree-shaped queries to be satisfied quickly. This assists users unskilled in query optimization in building high performing systems and algorithms using Freebase. Also, similar in flavor to the reversion tools in many wiki systems, the Freebase data store has an integrated versioning mechanism that supports complete reversion of all database edits, thus allowing "undo" of large, complex operations to any degree. The graph store also supports fine-grained attribution of all data. Together, these two features allow capabilities such as fixing damage caused by malicious and/or defective users and algorithms.
- **A Large Data Object Store (LOB):** This is a store of large data objects such as text documents, images, sound files, and software. LOB objects are indexed and annotated in the graph store, and are accessed through the same API as the graph store.
- **A Public, HTTP-Based API:** The primary method of machine access to Freebase is through its public HTTP-based API. Queries and answers are formulated using the "Metaweb Query Language" (MQL) that we have developed for this purpose, and which adheres to the Javascript Object Notation (JSON) syntax (Flanagan 2007). A novel feature of the Freebase API is that all writes as well as reads are performed using MQL. MQL is "query by template" language designed for ease of use, and scalability on a graph store that is simultaneously being written to and read from. This is a different focus than other, more expressive (but traditionally read-only) graph/RDF query languages (e.g. SPARQL,SeRQL,RQL). MQL is well suited to use over continuously evolving, refactoring, and updating data sets. Important MQL features include mixing structural data matching with approximate string matching of literals, cursors, and flat semantics of all data, which makes mixing of data and metadata easy. Because of the intrinsic versioning of the graph store, MQL also supports retrieval of data that has been deleted or replaced, through the use of appropriate filtering clauses.
- **An Easy-To-Use Web UI:** Human users of Freebase can use its Web interface (found at www.freebase.com) to search, browse, edit, and organize data at a human scale. This UI is AJAX-heavy and is intended to guide (especially non-technical) users toward structuring data and entity reconciliation by making these tasks as easy or easier than simple data entry. Novel features include typed entity suggestion through an interactive auto-completion mechanism, type-based filtering of search results, and data cleaning at time of entry.
- **A Substantial Seed Data Set:** An emphasis has been placed on on the early seeding of Freebase with data sets of interest to the general population, rather than those that are highly esoteric and specialized. This hopefully will result in greater heterogeneity of structure and content, that is more representative of the world's sum of general knowledge. The current data in Freebase consists of millions of concepts (topics) and tens of millions of relationships between those topics. Areas of initial seeding include popular culture (e.g. films, music, books, sports, television), location information (restaurants, geolocations, businesses), scholarly information (linguistics, biology, astronomy), and general knowledge (Wikipedia). While this data is already useful, we are making efforts for it to grow quickly over time in both quantity and density of relationships.
- **A Creative Commons License:** One of the problems with using and redistributing data on the Web is the restrictions and ambiguity of the copyright on that data. To avoid this problem in Freebase, by default and with few exceptions, all users who contribute to Freebase agree to a *Creative Commons Attribution* (CC-by) license(Berry & Moss 2005) for their data.
- **A Lightweight Typing System**: Pervasive in the UI, API, and data itself is the Freebase typing system. It is a loose collection of structuring mechanisms and conventions, rather than a rigid systems of ontologies and representations. The type system supports collaborative design of types and properties, and there is no intrinsic canonical world view of all knowledge. Conflicting and contradictory types and properties may exist simultaneously in order to reflect users differing opinions and understanding.

The Type System in Freebase

The Freebase type system consists of very few moving parts. Its features are designed to be simple and easily mapped onto the ontological structures of other systems, consisting of only the following kinds of objects:

- **Topic:** This is an object representing a discrete entity. A topic may be specific and concrete, (e.g. Bill Clinton or The City of Vancouver, Canada) or an abstract concept (e.g. The number PI, Zoroastrianism). In Freebase, each topic is given exactly one globally unique identifier (GUID) which refers only that topic. While a topic may have many names and may be used in many different contexts in Freebase, each topic should represent one and only one concept or entity in the world.

- **Literal:** A literal in Freebase is a simple scalar string, numeric value, boolean, or timestamp. All other data types are built using explicit structure.
- **Type:** A type is an object that is used to semantically group topics. A topic associated with a type is considered to be an instance of that type. Examples of types include Film Actor, Person, Airport, and Programming Language. Topics may have multiple types which may be added or removed over time. (e.g. A person is elected president in the USA may have the US President type added.) Like a topic, a type represents exactly one meaning, even if it has multiple names and usages. There is no notion of a type hierarchy in Freebase. Instead, types are created and "mixed in" as needed independently, because each confers different expected properties. For example, "Film Actor" does not inherit from "Person" because some actors may not be people (e.g. dogs, robots), but have the same properties (e.g. Films, Film Awards) as human actors.
- **Property:** A property can be thought of as a flavor of attribute of a topic in Freebase. Properties may be literals (e.g. a name, a length, a SKU number) or relationships to another topic (e.g. is a parent of, is contained by, has sequel). A topic using a property is considered to have an instance of that property. Every property has an "Expected Type", which is an indicator that the topic at other end of the property's link is likely to be of the expected type.
- **Schema:** Each type has collection of zero or more properties, known as the schema of that type. If a topic is an instance of a type, this is an indication that the properties in the schema of that type are appropriate for application to that topic. For example, if the topic Danny Hillis is an instance of the type Person which has a schema with property Date of Birth, then it is expected that Danny Hillis may have an instance of the Date of Birth property.

Rather than ontological correctness or logical consistency, Freebase's type system is designed for collaborative creation of structure. Types and properties may be created by any user to add structured data to topics in Freebase. As an example of collaborative workflow emerging from use of the Freebase type system, differences of opinion can be explicitly represented as incompatible types and properties. One user can create a property that asserts a specific positive relationship between two entities. If another user disagrees and no compromise can be found, then this second user can create a simultaneous property that asserts the negative relationship. Since all data in Freebase have fine grained attribution, third parties may simply filter query results by the version of reality they wish to pay attention to.

Using Freebase To Create Structure

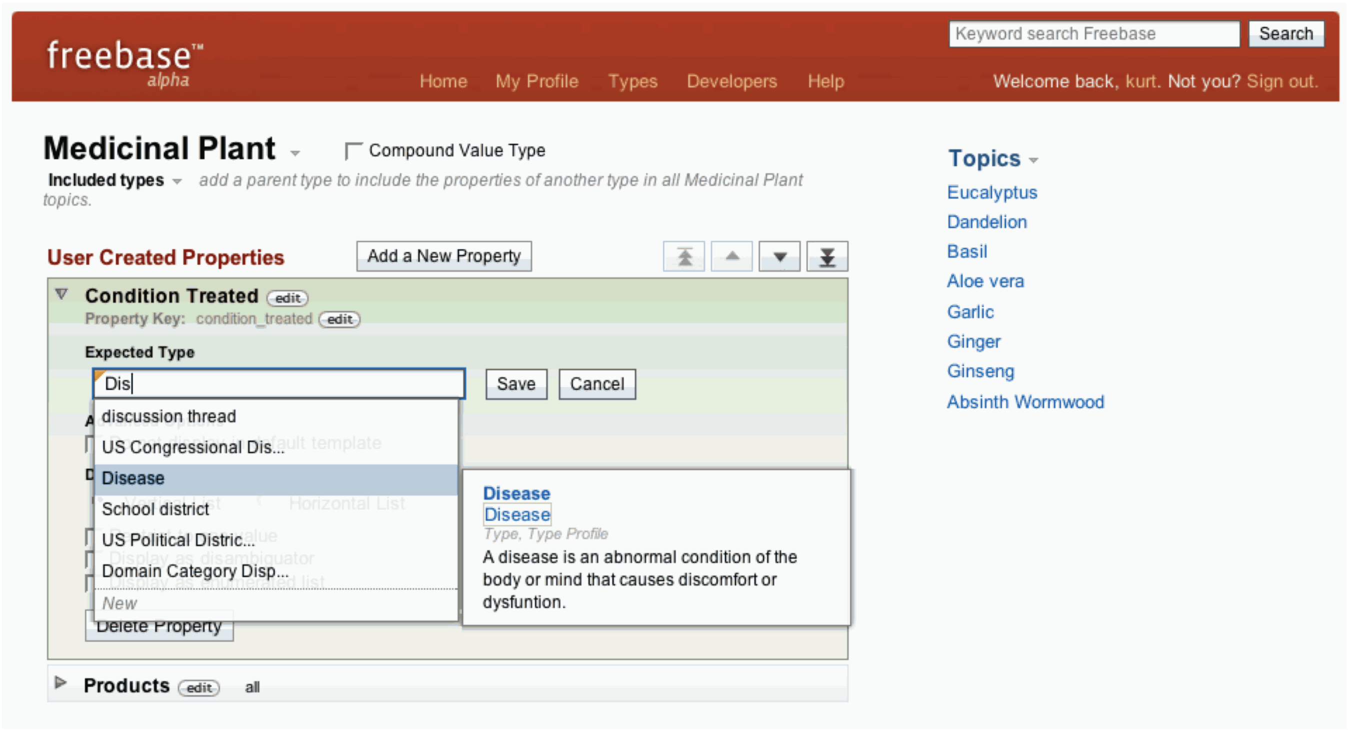
The use of Freebase is intended to be intrinsically collaborative between both human users and automated processes. With a much finer grain of object than documents (as in Wikipedia), users may more easily work on pieces of a complex data set without accidentally stepping on each other's toes. The access features of Freebase are the Web UI for interactive human use and the API for automated processes and third party access applications.

Structure creation and information integration in Freebase are intended to be a natural part of this usage of Freebase rather than a task for which a specific user or agent must take responsibility. It is expected that each user will do only as much as is needed for his/her own purposes, but that collectively over time, the result will be overall increased and improved data structure. Unstructured data will become structured, errors will be fixed, and new, hopefully better, structure will be derived from old. The initial Freebase toolset supports a number of features that are intended to create this "emergent workflow" of integration, even if the users are not explicitly trying to perform such integration.

Adding Structure Interactively

Freebase's AJAX-heavy, Web browser based user interface is designed for data browsing, entry, and refactoring in the context of many other users performing similar actions. All write operations are immediately visible, making tightly interactive data entry and structure creation possible. Some workflow examples of Freebase usage through the UI that result in information integration include:

- **Creation and editing of types and properties:** As an example, Figure 1 shows a user editing the schema for the "Medicinal Plant" type. Here the expected type for the "Condition Treated" property is being set to "Disease". Notice how the auto-completion suggests "Disease" as a choice, including a description of what the type itself. This highly contextual display of information is one of the key interactive features that reduces the creation of redundant entities and thus helps the reconciliation process.
- **Creating and editing structured instance information:** Users can create new entities (topics), give these entities types, and create property instances that are part of the schemas of those types. Figure 2 shows a user adding a "Condition Treated" relationship from the Medicinal Plant topic "Dandelion". Auto-completion suggests the match of "Dyspepsia" to the user. The user may also create a new entity with the desired name. In either case, the entity at the end of this link is automatically typed with the expected type if needed. This is an example of how the Freebase Web UI extracts information from the user implicit intention to add to the overall explicit structure of the data without burdening the user with extra effort.
- **Marking topics for reconciliation and cleaning:** Unlike in some databases, entities in Freebase are intended to be explicitly canonicalized. That is, there should be only one GUID in Freebase representing each real world entity, topic, or concept. Entity canonicalization in Freebase is organized as topic reconciliation through a "mark and sweep" process. Using the Web UI, users can mark a topic as a suggestion for future processing of that topic. For example, a user can mark a topic as being a duplicate entity of another, creating a relationship between those topics having the meaning "should be merged with". Later, humans and/or algorithms can then "sweep"



add a parent type to include the properties of another type in all Medicinal Plant topics.

Property Key: condition_treated

A disease is an abnormal condition of the body or mind that causes discomfort or dysfuntion.

Figure 1: A screenshot of a user editing property in a type schema.

through these marks to actually perform the merges. This process thus allows for users who have insufficient permissions to perform a merge to request one. Also, multiple users/algorithms may mark the merge, thus allowing a form of voting on an operation for controversial operations. This mark and sweep process can also be used to mark entities for deletion (e.g. spam or copyright violations) or for other gardening tasks as the need arises. Since the users can create their own types and properties, they can create their own set of mark relationships and their own sweep processes. Figure 3 shows a user marking a musical artist topic for merge with another topic about the same artist (but was spelled differently).

Using The Freebase API

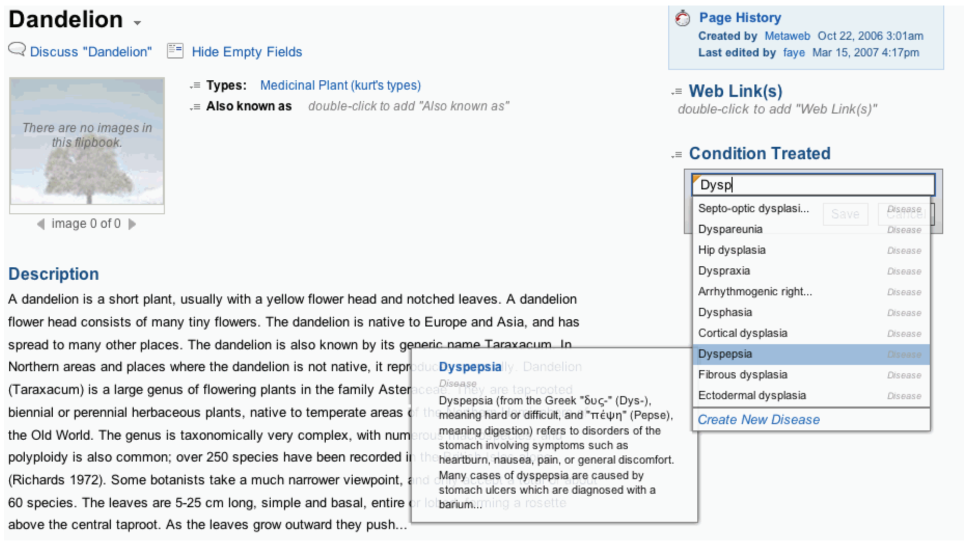
While our efforts so far have been focused on collaborative features in Its Web-based user interface, Freebase also has an API that supports highly expressive structured queries. Through the API, MQL queries in JSON syntax are submitted via HTTP with responses returned also in JSON. MQL is best thought of as a "query by template" language in that the shape of the MQL query result is defined by the shape of the query, excepting the cardinality of (sub)results and some administrative directives. We chose JSON to represent our query results rather than the standard RDF due to the greater ease in building Web application using JSON. We hope to have an RDF export facility in the near future.

A simple example of a MQL query:

```
{
  "query":[{
    "album":[],
    "name":"The Police",
    "type":"/music/artist"
  }]
}
```

asks for the albums of a musical artist. The result is the same shape as the query:

```
{
  "q1": {
    "status": "/mql/status/ok",
    "result": [
      {
        "album": [
          "Outlandos d'Amour",
          "Reggatta de Blanc",
          "Zenyatta Mondatta",
          "Ghost in the Machine",
          "Synchronicity",
          "Every Breath You Take:
           The Singles",
          "Greatest Hits",
          "Every Breath You Take:
           The Classics",
          "Their Greatest Hits",
          "Can't Stand Losing You",
          "Roxanne '97"
        ],
        "type": "/music/artist",
```


Dandelion

Discuss "Dandelion" Hide Empty Fields

Types: Medicinal Plant (kurt's types)

Also known as double-click to add "Also known as"

There are no images in this flipbook.

image 0 of 0

Description

A dandelion is a short plant, usually with a yellow flower head and notched leaves. A dandelion flower head consists of many tiny flowers. The dandelion is native to Europe and Asia, and has spread to many other places. The dandelion is also known by its generic name Taraxacum. In Northern areas and places where the dandelion is not native, it repr... (Taraxacum) is a large genus of flowering plants in the family Aster... biennial or perennial herbaceous plants, native to temperate areas ... the Old World. The genus is taxonomically very complex, with num... polyploidy is also common; over 250 species have been recorded i... (Richards 1972). Some botanists take a much narrower viewpoint, ... 60 species. The leaves are 5-25 cm long, simple and basal, entire ... above the central taproot. As the leaves grow outward they push...

Dyspepsia
Disease
Dyspepsia (from the Greek "δυς-" (Dys-), meaning hard or difficult, and "πέψη" (Pepse), meaning digestion) refers to disorders of the stomach involving symptoms such as heartburn, nausea, pain, or general discomfort. Many cases of dyspepsia are caused by stomach ulcers which are diagnosed with a barium...

Page History
Created by Metaweb Oct 22, 2006 3:01am
Last edited by faye Mar 15, 2007 4:17pm

Web Link(s)
double-click to add "Web Link(s)"

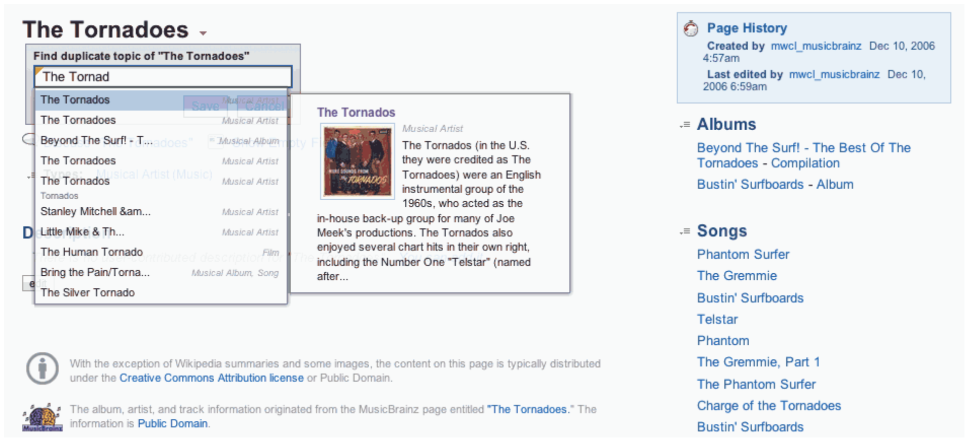
Condition Treated

Dysp

Septo-optic dysplasi...	Disease
Dyspareunia	Disease
Hip dysplasia	Disease
Dyspraxia	Disease
Arrhythmogenic right...	Disease
Dysphasia	Disease
Cortical dysplasia	Disease
Dyspepsia	Disease
Fibrous dysplasia	Disease
Ectodermal dysplasia	Disease

Create New Disease

Figure 2: A screenshot of a user creating a link between entities.


The Tornadoes

Find duplicate topic of "The Tornadoes"

The Tornad

The Tornados	Musical Artist
The Tornadoes	Musical Artist
Beyond The Surf! - T...	Musical Album
The Tornadoes	Musical Artist
The Tornados Tornados	Musical Artist
Stanley Mitchell &am...	Musical Artist
Little Mike & Th...	Musical Artist
The Human Tornado	Film
Bring the Pain/Torna...	Musical Album, Song
The Silver Tornado	

The Tornados
Musical Artist
The Tornados (in the U.S. they were credited as The Tornadoes) were an English instrumental group of the 1960s, who acted as the in-house back-up group for many of Joe Meek's productions. The Tornados also enjoyed several chart hits in their own right, including the Number One "Telstar" (named after...

With the exception of Wikipedia summaries and some images, the content on this page is typically distributed under the Creative Commons Attribution license or Public Domain.

The album, artist, and track information originated from the MusicBrainz page entitled "The Tornadoes." The information is Public Domain.

Page History
Created by mwcl_musicbrainz Dec 10, 2006 4:57am
Last edited by mwcl_musicbrainz Dec 10, 2006 6:59am

Albums
Beyond The Surf! - The Best Of The Tornadoes - Compilation
Bustin' Surfboards - Album

Songs
Phantom Surfer
The Gremmie
Bustin' Surfboards
Telstar
Phantom
The Gremmie, Part 1
The Phantom Surfer
Charge of the Tornadoes
Bustin' Surfboards

Figure 3: A screenshot of a user marking two topics as duplicates to be merged.

```
        "name": "The Police"
      }
    ]
  },
  "status": "200 OK"
}
```

MQL's type system is considered to be plain data like any other in Freebase. A query asking for the properties of the "Musical Artist" type would look like:

```
{
  "query":[{
    "id":"/music/artist",
    "properties":[],
    "type":"/type/type"
  }]
}
```

Freebase would give the result:

```
{
  "q1": {
    "status": "/mql/status/ok",
    "result": [
      {
        "type": "/type/type",
        "id": "/music/artist",
        "properties": [
          "/music/artist/origin",
          "/music/artist/active_start",
          "/music/artist/active_end",
          "/music/artist/genre",
          "/music/artist/label",
          "/music/artist/home_page",
          "/music/artist/member",
          "/music/artist/album",
          "/music/artist/contribution",
          "/music/artist/track",
            ]
          }
        ]
    },
    "status": "200 OK"
}
```

The Freebase Web UI runs entirely in the user's browser, accessing Freebase through its API, as any other client application would. The existence of the Web UI demonstrates some of the capabilities that Freebase applications can have.

Summary and Future Work

The Freebase service is being built as a practical platform for collaborative creation and evolution of structured data. The Freebase Web UI and API support a number of interactive workflows that allow personal needs to result in adding globally usable structure that can be immediately used by others.

We are offering Freebase as a free public service to all users and have built a Web UI that facilitates the structuring of data and reconciliation of entities. However, as of the time of this paper, we are still in an "alpha" testing phase. Readers interested in participating in this phase are encouraged to contact the authors or visit `www.freebase.com` to request an account. We will continue to improve Freebase by adding data and user interface features, but we recognize that there is still much work to be done. We hope that increased usage will progress Freebase toward becoming a "structured data commons" for much of the world's public data. We encourage those who are interested in building public data services that access the data in Freebase to create applications and holders of sharable data sets that may be of public interest to include them in Freebase. Researchers in areas such as entity extraction and reconciliation, data mining, Semantic Web, information retrieval, ontology creation and analysis, and graph analysis can use the Freebase API to support their work.

References

Auer, S., and Lehmann, J. 2007. "what have innsbruck and leipzig in common? extracting semantics from wiki content". In *Proceedings of the European Semantic Web Conference (TO BE PUBLISHED).*

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. "the semantic web". *Scientific American* 284:34–43.

Berry, D. M., and Moss, G. 2005. "the libre culture manifesto".

DB, D. 2007. The dabble db database creation service. [on the internet. `http://www.dabbledb.com`].

Flanagan, D. 2007. *Developing Metaweb-Enabled Web Applications.* Metaweb Technologies, Inc. `www.freebase.com`.

Franklin, M.; Levy, A.; and Maier, D. 2005. From databases to dataspaces: A new abstraction for information management.

Gabrilovich, E., and Markovitch, S. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence.*

GoogleBase. 2007. Googebase. [on the internet. `http://base.google.com`].

Krotzsch, M.; Vrandecic, D.; and Volkel, M. 2005. "wikipedia and the semantic web - the missing links". In *Proceedings of Wikimania 2005, Frankfurt, Germany.*

Strube, M., and Ponzetto, S. P. 2006. WikiRelate! computing semantic relatedness using wikipedia. In *Proceedings of the Association For the Advancement of Artificial Intelligence.*

Wikipedia. 2007. Wikipedia: The free encyclopedia. [on the internet. `http://www.wikipedia.org`].