Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Theophrastus: On demand and real-time automatic annotation and exploration of (web) documents using open linked data

Pavlos Fafalios, Panagiotis Papadakos *

Institute of Computer Science, FORTH-ICS, Greece
Computer Science Department, University of Crete, Greece

ARTICLE INFO

Article history:
Received 30 July 2013
Received in revised form
4 July 2014
Accepted 29 July 2014
Available online xxxx

Keywords:
Semantic annotation
Linked open data
Entity mining and exploration
Biodiversity

ABSTRACT

Theophrastus is a system that supports the automatic annotation of (web) documents through entity mining and provides exploration services by exploiting Linked Open Data (LOD), in real-time and only when needed. The system aims at assisting biologists in their research on species and biodiversity. It was based on requirements coming from the biodiversity domain and was awarded the first prize in the Blue Hackathon 2013. Theophrastus has been designed to be highly configurable regarding a number of different aspects like *entities of interest*, *information cards* and *external search systems*. As a result it can be exploited in different contexts and other areas of interest. The provided experimental results show that the proposed approach is efficient and can be applied in real-time.

1. Introduction

The purpose of the Blue Hackathon 2013[1] Workshop was to gather developers in order to create and apply new ideas targeting a sustainable and healthier future. Specifically, this event tried to showcase the potential of Open Data and APIs that facilitate access to marine environmental, hydrobiological and oceanographical data. It was a hands-on workshop whose overall purpose was to find out what existing tools can do, potentially when applied to new disciplines and types of literature, rather than developing new tools and techniques. The event was organized and facilitated by *Agro-Know Technologies* and hosted by the *Hellenic Centre for Marine Research (HCMR)* in Heraklion, Crete.

A number of different challenges were suggested along two basic dimensions:

- The use of environmental and bibliographic data and measurements that are of interest to scientists and researchers.
- Innovative educational applications and quests available to Thalassocosmos[2] educators, in order to raise awareness on environmental issues.

This work focuses on the first dimension of the challenges. During the event we had extensive discussions with marine and biology scientists. According to them, the main problem their community faces is that the process of identifying and gathering related information about a species is time consuming and cumbersome. Specifically, the bottlenecks of this process were reported to be:

- **Species identification difficulties**. It is not easy to locate, identify and summarize all referenced species in an information source.
- **Lack of exploration services**. There is no central point that can provide exploration of similar species, genus, etc.
- **Diversity of sources**. Biodiversity data are distributed in a number of heterogeneous repositories.
- **Disengagement from the initial task**. Due to the diversity of sources, the marine biologist has to locate information in different places. As a result he/she has to move his/her focus on other information sources instead of the original paper, web page, etc.
- **Ambiguous synonyms**. As a consequence of the vast 250 years bibliography, the same names have been given to different species, leading to wrong information.

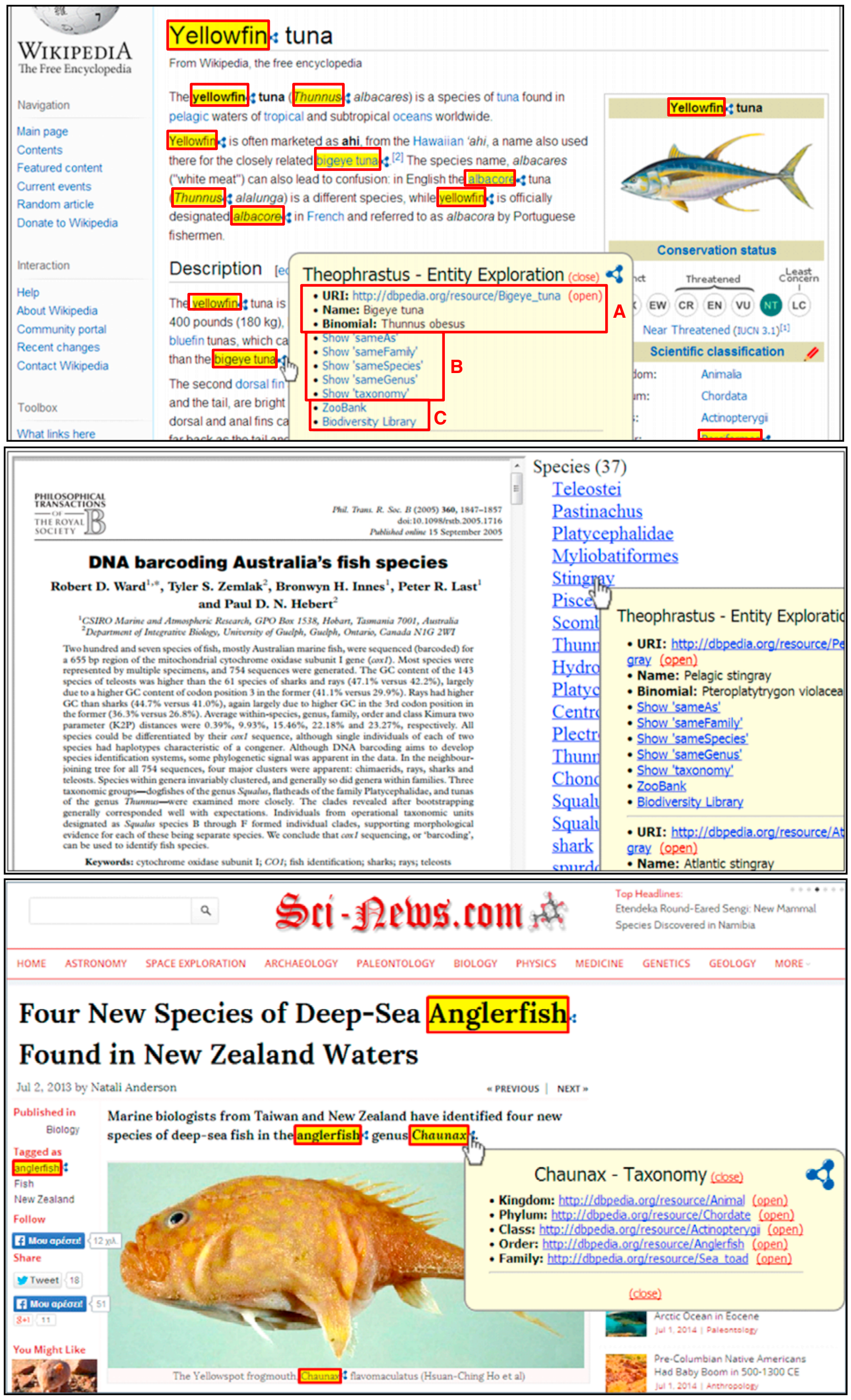
In order to remove the aforementioned bottlenecks and assist biologists with their research about species and biodiversity, we

* Corresponding author at: Institute of Computer Science, FORTH-ICS, Greece. Tel.: +30 6976749835.
E-mail addresses: fafalios@ics.forth.gr (P. Fafalios), papadako@gmail.com, papadako@ics.forth.gr (P. Papadakos).

1 http://wiki.agroknow.gr/agroknow/index.php/BlueHackathon2013.

2 The Thalassocosmos complex is the largest research, technology and entertainment center in the Mediterranean.

http://dx.doi.org/10.1016/j.websem.2014.07.009
1570-8268/© 2014 Elsevier B.V. All rights reserved.



WIKIPEDIA
The Free Encyclopedia

Navigation: Main page, Contents, Featured content, Current events, Random article, Donate to Wikipedia

Interaction: Help, About Wikipedia, Community portal, Recent changes, Contact Wikipedia

Toolbox: What links here

Yellowfin tuna

From Wikipedia, the free encyclopedia

The yellowfin tuna (Thunnus albacares) is a species of tuna found in pelagic waters of tropical and subtropical oceans worldwide.

Yellowfin is often marketed as ahi, from the Hawaiian 'ahi, a name also used there for the closely related bigeye tuna.[2] The species name, albacares ("white meat") can also lead to confusion: in English the albacore tuna (Thunnus alalunga) is a different species, while yellowfin is officially designated albacore in French and referred to as albacora by Portuguese fishermen.

Description

The yellowfin tuna is ... 400 pounds (180 kg), ... bluefin tunas, which ca... than the bigeye tuna ... The second dorsal fin ... and the tail, are bright ... dorsal and anal fins ca...

Yellowfin tuna — Conservation status: EW, CR, EN, VU, NT, LC — Threatened — Least Concern — Near Threatened (IUCN 3.1)[1] — Scientific classification: Animalia, Chordata, Actinopterygii

Theophrastus - Entity Exploration (close)
- URI: http://dbpedia.org/resource/Bigeye_tuna (open)
- Name: Bigeye tuna
- Binomial: Thunnus obesus

A

- Show 'sameAs'
- Show 'sameFamily'
- Show 'sameSpecies'
- Show 'sameGenus'
- Show 'taxonomy'

B

- ZooBank
- Biodiversity Library

C

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY B

Phil. Trans. R. Soc. B (2005) 360, 1847–1857
doi:10.1098/rstb.2005.1716
Published online 15 September 2005

DNA barcoding Australia's fish species

Robert D. Ward[1,*], Tyler S. Zemlak[2], Bronwyn H. Innes[1], Peter R. Last[1] and Paul D. N. Hebert[2]

[1]CSIRO Marine and Atmospheric Research, GPO Box 1538, Hobart, Tasmania 7001, Australia
[2]Department of Integrative Biology, University of Guelph, Guelph, Ontario, Canada N1G 2W1

Two hundred and seven species of fish, mostly Australian marine fish, were sequenced (barcoded) for a 655 bp region of the mitochondrial cytochrome oxidase subunit I gene (cox1). Most species were represented by multiple specimens, and 754 sequences were generated. The GC content of the 143 species of teleosts was higher than the 61 species of sharks and rays (47.1% versus 42.2%), largely due to a higher GC content of codon position 3 in the former (41.1% versus 29.9%). Rays had higher GC than sharks (44.7% versus 41.0%), again largely due to higher GC in the 3rd codon position in the former (36.3% versus 26.8%). Average within-species, genus, family, order and class Kimura two parameter (K2P) distances were 0.39%, 9.93%, 15.46%, 22.18% and 23.27%, respectively. All species could be differentiated by their cox1 sequence, although single individuals of each of two species had haplotypes characteristic of a congener. Although DNA barcoding aims to develop species identification systems, some phylogenetic signal was apparent in the data. In the neighbour-joining tree for all 754 sequences, four major clusters were apparent: chimaerids, rays, sharks and teleosts. Species within genera invariably clustered, and generally so did genera within families. Three taxonomic groups—dogfishes of the genus Squalus, flatheads of the family Platycephalidae, and tunas of the genus Thunnus—were examined more closely. The clades revealed after bootstrapping generally corresponded well with expectations. Individuals from operational taxonomic units designated as Squalus species B through F formed individual clades, supporting morphological evidence for each of these being separate species. We conclude that cox1 sequencing, or 'barcoding', can be used to identify fish species.

Keywords: cytochrome oxidase subunit I; CO1; fish identification; sharks; rays; teleosts

Species (37)
Teleostei
Pastinachus
Platycephalidae
Myliobatiformes
Stingray
Pisce
Scomb
Thunn
Hydro
Platyc
Centr
Plectr
Thunn
Chond
Squal
Squal
shark

Theophrastus - Entity Exploratio
- URI: http://dbpedia.org/resource/Pe gray (open)
- Name: Pelagic stingray
- Binomial: Pteroplatytrygon violacea
- Show 'sameAs'
- Show 'sameFamily'
- Show 'sameSpecies'
- Show 'sameGenus'
- Show 'taxonomy'
- ZooBank
- Biodiversity Library

- URI: http://dbpedia.org/resource/At gray (open)
- Name: Atlantic stingray

Sci-News.com

Top Headlines: Etendeka Round-Eared Sengi: New Mammal Species Discovered in Namibia

HOME ASTRONOMY SPACE EXPLORATION ARCHAEOLOGY PALEONTOLOGY BIOLOGY PHYSICS MEDICINE GENETICS GEOLOGY MORE

Four New Species of Deep-Sea Anglerfish Found in New Zealand Waters

Jul 2, 2013 by Natali Anderson « PREVIOUS | NEXT »

Published in Biology
Tagged as anglerfish, Fish, New Zealand
Follow
Share
You Might Like

Marine biologists from Taiwan and New Zealand have identified four new species of deep-sea fish in the anglerfish genus Chaunax.

The Yellowspot frogmouth Chaunax flavomaculatus (Hsuan-Ching Ho et al)

Chaunax - Taxonomy (close)
- Kingdom: http://dbpedia.org/resource/Animal (open)
- Phylum: http://dbpedia.org/resource/Chordate (open)
- Class: http://dbpedia.org/resource/Actinopterygii (open)
- Order: http://dbpedia.org/resource/Anglerfish (open)
- Family: http://dbpedia.org/resource/Sea_toad (open)

(close)

Arctic Ocean in Eocene, Jul 1, 2014 | Paleontology
Pre-Columbian Native Americans Had Baby Boom in 500-1300 CE, Jul 1, 2014 | Anthropology

Fig. 1. Theophrastus system indicative screenshots over html (top and bottom) and pdf files (middle).

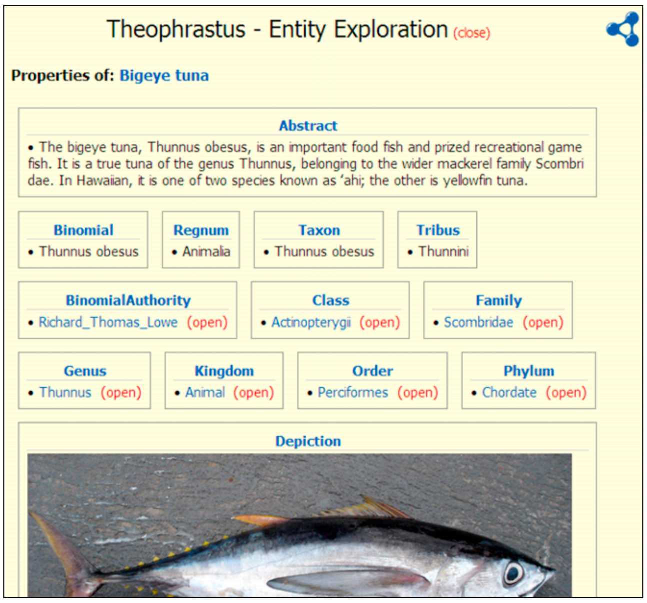
designed and implemented Theophrastus.[3,4] The system supports the automatic annotation of the original information source and provides exploration services through entity mining and exploitation of Linked Open Data (LOD) [1]. These services are provided in real-time and only when needed, by using available Semantic Web technologies like SPARQL, as well as mining tools. Theophrastus was voted and awarded the first prize in the Blue Hackathon 2013.

Fig. 1 depicts three indicative screenshots of Theophrastus. In all examples, the entities of interest concern fish species (including species genera and family names). The top screenshot shows the annotation of the Wikipedia page of Yellowfin tuna after clicking over a bookmarklet.[5] The identified species have been annotated with yellow background and the RDF icon next to each one of them. By clicking on the icon the user can, at real-time,

[3] A deployment of the system and more information can be found at http://www.ics.forth.gr/isl/Theophrastus.

[4] Greek philosopher Theophrastus continued Aristotle's ichthyological research. He wrote a treatise on amphibious fish and offered the first systemization of the botanical world.

[5] A bookmarklet is a bookmark stored in a web browser that extends the browser's functionality.


Theophrastus - Entity Exploration (close)

Properties of: Bigeye tuna

Abstract
• The bigeye tuna, Thunnus obesus, is an important food fish and prized recreational game fish. It is a true tuna of the genus Thunnus, belonging to the wider mackerel family Scombri dae. In Hawaiian, it is one of two species known as 'ahi; the other is yellowfin tuna.

Binomial
• Thunnus obesus

Regnum
• Animalia

Taxon
• Thunnus obesus

Tribus
• Thunnini

BinomialAuthority
• Richard_Thomas_Lowe (open)

Class
• Actinopterygii (open)

Family
• Scombridae (open)

Genus
• Thunnus (open)

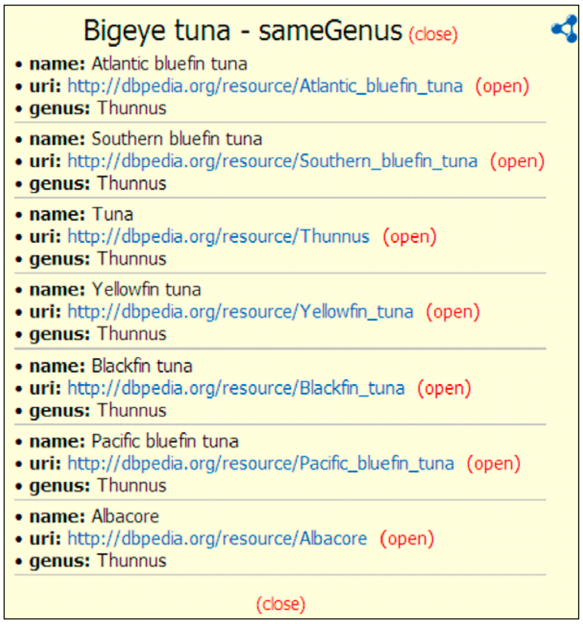
Kingdom
• Animal (open)

Order
• Perciformes (open)

Phylum
• Chordate (open)

Depiction

Fig. 2. Exploration of species' properties.


Bigeye tuna - sameGenus (close)

• **name:** Atlantic bluefin tuna
• **uri:** http://dbpedia.org/resource/Atlantic_bluefin_tuna (open)
• **genus:** Thunnus

• **name:** Southern bluefin tuna
• **uri:** http://dbpedia.org/resource/Southern_bluefin_tuna (open)
• **genus:** Thunnus

• **name:** Tuna
• **uri:** http://dbpedia.org/resource/Thunnus (open)
• **genus:** Thunnus

• **name:** Yellowfin tuna
• **uri:** http://dbpedia.org/resource/Yellowfin_tuna (open)
• **genus:** Thunnus

• **name:** Blackfin tuna
• **uri:** http://dbpedia.org/resource/Blackfin_tuna (open)
• **genus:** Thunnus

• **name:** Pacific bluefin tuna
• **uri:** http://dbpedia.org/resource/Pacific_bluefin_tuna (open)
• **genus:** Thunnus

• **name:** Albacore
• **uri:** http://dbpedia.org/resource/Albacore (open)
• **genus:** Thunnus

(close)

Fig. 3. Exploration of same genus species.

semantically explore the corresponding species. In this example the user explores the species bigeye tuna. Specifically, the pop-up window displays: (A) some basic information regarding the bigeye tuna, e.g. its URI and its binomial (scientific name), (B) a set of information cards, e.g. other species that belong to the same family or genus, its taxonomy, etc., and (C) a set of external systems (e.g. the Biodiversity Heritage Library[6]) that the user can visit and query for the corresponding species. The user can browse all this information; for instance by clicking on the URI of an object the user can inspect all the properties of the corresponding species that exist in the LOD (Fig. 2), and by clicking the `sameGenus` link the user can instantly browse other species that belong to the same genus (Fig. 3). Note that all this information is derived at real-time and, as described later on, can be configured by the user.

The middle part of Fig. 1 shows an analyzed scientific paper (PDF file). The identified species are displayed in the right sidebar and the user can explore a species by simply clicking on it. Notice that for the detected species stingray, there are more than one matching species (the species Pelagic stingray and Atlantic stingray).

Finally, at the bottom part of Fig. 1, a scientific news web page has been analyzed and annotated. In this example, the user inspects the information card of the Chaunax species taxonomy.

We should stress that we have paid particular attention to the configurability of the system. The user/administrator can easily configure `Theophrastus` for satisfying the needs of different user communities, by defining a number of different *entities of interest*, *information cards* and *external search systems*, as well as by specifying how to find related resources and how to browse them. For instance, scientists in an organization may want to inspect and explore additional categories of entities in the document, e.g. *water areas*, *countries*, etc. In addition, different communities/users may want to link and enrich the identified species with resources from different sources; one may want to attach images from DBpedia [2], while others might want to include papers that describe the genome of the species. Since the needs of a community constantly change, `Theophrastus` can be dynamically configured without requiring redeployment (e.g. for updating the list of fish species, for specifying a new information card, etc.). This enhanced configurability allows `Theophrastus` to be applied to different contexts and other areas of interest.

In brief, the process supported by `Theophrastus` is sketched in Fig. 4. Initially, entity mining is performed over the contents of a (web) document, which are then annotated with the identified entities according to the specified entities of interest. Afterwards, the user can explore these entities by exploiting the available semantic information (LOD) through the defined information cards, and by querying external search systems.

2. The `Theophrastus` system

In this section we present thoroughly the main concepts of the system (Section 2.1), the process of entity mining and annotation of the source (Section 2.2), and the semantic exploration of the detected entities (Section 2.3). Later on, we discuss configuration issues (Section 2.4), experimental results (Section 2.5), and finally the limitations of the proposed approach (Section 2.6).

2.1. Main concepts

A basic concept is the notion of *entities of interest*, which can be thought of as *categories*. For example, in the case of the marine domain the entities of interest include fish species, water areas, countries, etc. Every category is associated with a *knowledge base (KB)* (accessible through a *SPARQL endpoint*) and a *SPARQL template query*. This template query explicitly specifies how to find resources in the underlying KB matching the name of an entity, and what information to retrieve for each such resource. Part A in the top screenshot of Fig. 1 shows a matching resource and some basic information regarding the fish bigeye tuna.

For each category the user can specify one or more *information cards*. An information card is a *piece of information* regarding a resource that exists in the underlying KB. For example, regarding a fish species, an information card could be its *taxonomy* (classification), its *predatory fishes*, available *synonyms*, species that belong to the *same family* or *genus*, etc. Every information card is associated with a *card name* and a *SPARQL template query*. This template

6 http://www.biodiversitylibrary.org.

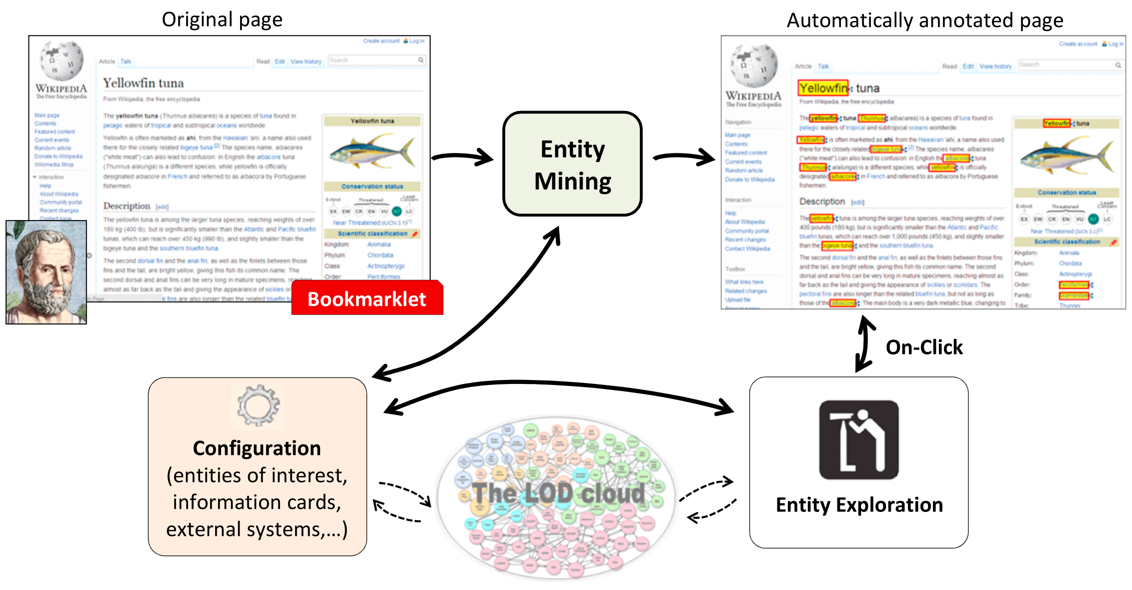

Original page
Automatically annotated page
Entity Mining
Bookmarklet
On-Click
Configuration (entities of interest, information cards, external systems,...)
The LOD cloud
Entity Exploration

Fig. 4. The process supported by the `Theophrastus` system.

specifies the information that must be obtained from the underlying KB regarding a particular resource. Part B in the top screenshot of Fig. 1 corresponds to a set of information cards.

Finally, for each category the user can define *external search systems*, queryable through the HTTP protocol. For example, regarding species one can use ZooBank[7] which contains descriptions and sketches of species. Each external search system is associated with a *system name* and a *template URL*. The template URL specifies how to query and access the external search system for a given entity name. In the top screenshot of Fig. 1, part C corresponds to a set of external search systems.

2.2. Real-time entity mining and annotation

Regarding the process of entity mining and annotation, `Theophrastus` at first reads the contents of the requested document. Afterwards, it exploits a named entity recognition tool for identifying entities in the document's contents, in a similar way as [3] does for the snippets of search results. Currently, `Theophrastus` does not process RDFa[8] or Microdata that may be embedded in a web page; it only identifies entities of interest based on its current configuration. In the case of a web page, the detected entities are annotated in the original contents (top and bottom screenshots of Fig. 1). Otherwise, in the case of a PDF file, they are displayed in the right sidebar (middle screenshot of Fig. 1).

`Theophrastus` can use any named entity recognition service that takes as input a text and returns a list of entity names. Currently, it uses Gate Annie [4], which is a ready-made information extraction system supporting both gazetteers (lists of entity names) and Natural Language Processing (NLP) functions. In our setting, it takes as input the contents of a web page or a PDF file. The output is a set of entity lists, one list for each category of identified entities. Gate Annie internally "cleans" the contents by removing useless text (like the HTML tags of a Web page). We have extended Gate Annie (as described in Section 2.4), in order to be able to create new categories and update existing ones using gazetteers. As a result, its functionality can be adapted according to the needs of different user communities, rendering also the `Theophrastus` system configurable and extensible.

7 http://www.zoobank.gr.
8 http://www.w3.org/TR/xhtml-rdfa-primer/.

2.3. On demand semantic exploration of entities

The user is able to semantically explore a detected entity at real-time and on demand. Specifically, when the user clicks on the small RDF icon on the right of a detected entity (or the name of the entity in the case of PDF files), the system at real-time gathers the related information by exploiting LOD and shows it in a pop-up window (Fig. 1). Notice that `Theophrastus` derives this information dynamically according to its configuration. In this way, the users not only get useful information about an entity without having to disengage from their initial task, but they can also start *browsing* the entities that are linked to the entity of choice.

In case the user requests the exploration of an identified entity that belongs to more than one of the supported categories (e.g. the named entity "argentina" is both a *country* and a *fish genus*), we inform the user through a popup window. The user can disambiguate the named entity by selecting the appropriate category and the system then retrieves the corresponding semantic information.

It should be noted that the proposed approach of exploiting LOD to gather information related to the mined entities is more dynamic, affordable and feasible than an approach that has its own KB of entities and facts. Consequently, the proposed system can be easily utilized by interested parties, assuming a minimum cost of configuring the system, as explained next.

2.4. Configuring Theophrastus

The system is deployed using the configuration stored in a properties file. The user can either change the system's configuration through this file or dynamically configure it through a configuration page, even while the system is running. Below we analyze the different configuration aspects.

Specifying how to find related resources. As described in Section 2.1, the user can define how to semantically match an identified entity and find related resources by giving a *SPARQL endpoint* and a *SPARQL template query* for each category of entities. Fig. 5 shows a SPARQL template query for the category Fish (having defined DBpedia's SPARQL endpoint as the underlying KB). The query retrieves information about fishes in DBpedia with a label that contains the name of the selected fish. In this specific example, for each matching fish the query returns its *URI*, *label* (in English) and

```
SELECT DISTINCT ?URI ?Name ?Binomial WHERE {
 ?URI a <http://dbpedia.org/ontology/Fish> .
 ?URI rdfs:label ?Name
    FILTER(langMatches(lang(?Name), 'EN'))
    FILTER(regex(str(?Name),'<ENTITY>','i')) .
 OPTIONAL {
   ?URI <http://dbpedia.org/property/binomial> ?Binomial
 } }
```

Fig. 5. Example SPARQL template query for finding related resources.

```
SELECT DISTINCT ?name ?uri ?genus WHERE {
  <THE_URI> <http://dbpedia.org/property/genus> ?genus .
  ?uri <http://dbpedia.org/property/genus> ?genus .
  ?uri <http://dbpedia.org/property/name> ?name }
```

Fig. 6. Example SPARQL template query for declaring an information card.

```
SELECT DISTINCT ?propertyName ?propertyValue
WHERE {
 { <THE_URI> ?propertyName ?propertyValue
        FILTER(langMatches(lang(?propertyValue),'EN'))
 } UNION {
   <THE_URI> ?propertyName ?propertyValue
        FILTER(!isLiteral(?propertyValue))
        FILTER(!isBlank(?propertyValue)) } }
```

Fig. 7. Example SPARQL template query for specifying how to browse a resource.

binomial. Notice that the SPARQL template query contains the character sequence <ENTITY>, which is used for gathering information about a specific entity. Specifically, the system reads the template query of the entity's category and replaces each occurrence of <ENTITY>with the entity's label name. The marked part A of the top example of Fig. 1 shows the response of the SPARQL query for the fish bigeye tuna.

Specifying the information cards. The user is able to declare an *information card* by giving a *card name* and a *SPARQL template* query. The SPARQL template query contains the character sequence THE_URI; when a user clicks the information card regarding a particular matching resource (i.e. entity), the system reads the corresponding template query and replaces each occurrence of THE_URI with the URI of the resource. The corresponding SPARQL query is executed and the response is shown in the pop-up window.

Fig. 6 shows a SPARQL template query for declaring an information card regarding the category Fish, that uses the DBpedia's SPARQL endpoint. This query retrieves all fishes that belong to the same genus of a particular fish. Fig. 3 shows the results of the query over the *Bigeye tuna*'s URI http://dbpedia.org/resource/Bigeye_tuna.

Specifying how to browse a resource. In addition, the user can specify a SPARQL template query for browsing over the properties of a resource. Like in the case of an information card, the SPARQL template query contains the character sequence THE_URI. When a resource is clicked, each occurrence of THE_URI in the template query is replaced with the URI of the resource.

Fig. 7 shows such a SPARQL template query. The query consists of two parts: the first part retrieves all English literal properties of the resource, while the second part retrieves all related resources that have a URI. Note that the user is free to define another language for showing the literal properties, or write a *federated* SPARQL query [5] that queries diverse (distributed) KBs.

Specifying external search systems. External search systems, queryable through the HTTP protocol, can be added by providing a *system name* and a *template URL*. The template URL contains the

```
SELECT DISTINCT str(?label) WHERE {
  ?uri rdf:type <http://dbpedia.org/ontology/Fish> .
  ?uri rdfs:label ?label FILTER(lang(?label)='it') }
```

Fig. 8. Example SPARQL query for retrieving a list of Italian fish names from DBpedia.

character sequence <*ENTITY_NAME* >which on each request is replaced by the name of the entity. By clicking over an external search system, the user is redirected to the results of the corresponding system for this specific entity (the name of the entity is passed as a GET parameter).

Supporting a new category of entities. As previously noted, we have automated the procedure of adding a new category of entities in Gate Annie. Thereby, we can easily configure the entity names that are interesting for the application at hand (e.g. Linked Data of a particular RDF class). Specifically, we are able to add a new category of entities by giving a *category title* and a *list of words/phrases*. The list can be imported in the system by running a SPARQL query over a KB accessible via a SPARQL endpoint. For example, we can run the SPARQL query of Fig. 8 (which returns a list of Italian fish names) at DBpedia's SPARQL endpoint and thereby offer the ability to annotate and explore Italian fish names in a (web) document. In this case, a gazetteer file containing the entity names of the corresponding category is created and loaded in Gate Annie.

Likewise, the user can create a complex SPARQL query that describes a set of entities with some specific characteristics, or he/she can use federated SPARQL queries and gather entities from multiple KBs. As a result, the system will annotate and provide exploration services over entities that belong to this complex category of entities.

Formulation of SPARQL queries. The construction of SPARQL queries can be facilitated by a number of tools [6,7], that do not require any advanced knowledge of SPARQL. In addition, there are NLP approaches that guide users in formulating queries in a language seemingly akin to English and translate them to SPARQL [8].

2.5. Experimental results

Below we report some experimental results regarding the efficiency of the annotation and exploration services.

Document analysis timings. In general, the time for analyzing a document highly depends on the size of the document and the number of entities found. The main tasks to perform are: (a) the document is downloaded and its contents are read, (b) entity mining is performed over the contents of the document and (c) the original document is annotated. We measured the time needed to perform the aforementioned tasks over a collection of 500 Wikipedia pages regarding fish species. The results are shown in Fig. 9. On average, each page of the collection had 27.7 entities and a size of 125 kB. The main observation from the results is that more than 75% of pages were analyzed in real-time (less than 1 s). The worst case was a 757 kB page with 485 entities which needed almost 9 s to be analyzed.[9] Regarding the annotation of PDF files, analyzing a single-column scientific paper of 16 pages (6000–8000 words) needs around 8 s. Specifically, 6.5 s are needed for downloading and reading the contents, 1.5 s for performing entity mining and only a few ms for displaying the sidebar with the mined entities.

[9] The dataset and the results are available at http://www.ics.forth.gr/isl/Theophrastus/EvalResults.zip.

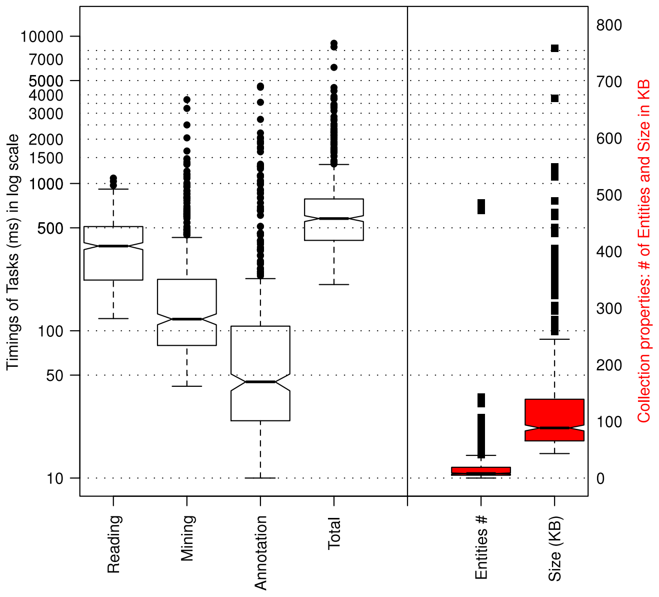


Fig. 9. Experimental results for fetching, mining and annotating a collection of 500 documents.

Entity exploration timings. The time needed for the exploration of an entity is highly dependent on the SPARQL template query and the corresponding SPARQL endpoint, since it is queried at real-time. In the conducted experiments we found that the more resources (i.e. entities) a category of entities contains in the underlying KB, the more time is required for retrieving the relevant information. For example, DBpedia's endpoint currently contains about 15,000 entities of type Fish (`rdf:type dbpedia-owl:Fish`). Finding related resources and retrieving information regarding the name of a Fish (query shown in Fig. 5) requires about 2 s. On the other hand, the same query about an Animal requires about 7 s (DBpedia contains about 145,000 entities of `rdf:type dbpedia-owl:Animal`). In the case of an information card or for browsing a resource, where the `URI` of an entity is known (queries shown in Figs. 6 and 7), the results are returned almost instantly ($\simeq$ 0.5 s) due to less string comparisons. We should note though that the above timings are indicative. Since the underlying SPARQL endpoints are public and queried at real-time, their response times may vary according to their payload at the time of request.

2.6. Discussion

The previously reported experimental results showed that the described services are provided in real-time (i.e. the user has to wait just a few seconds). The major bottleneck is that the existing publicly available online KBs like DBpedia are not optimized for efficiency. Since they are free to be queried by anyone and do not support multiple concurrent requests in order to avoid system overloading, they mainly serve demonstration purposes. Moreover, in the case of an entity belonging to a category with millions of entities, finding related resources can be expensive. In the near future, as such technologies mature and get used in applications, we expect that the aforementioned problems will be handled.

In the meantime, to increase the efficiency and the reliability of the entity exploration service, we could adopt a *caching mechanism* or we could *index* a part of the underlying KB. Furthermore, as proposed in [9], we could keep a local copy of data that hardly changes and offer a hybrid query execution approach for improving the response time and reducing the load on the endpoints, while keeping some of the results fresh. In addition, the underlying KBs may not be publicly available, or a *dedicated Warehouse* could be constructed that will only serve a particular application, like the marineTLO-based warehouse [10] for the marine domain. The KBs or the Warehouse could also be *distributed* in many servers, taking advantage of load balancing techniques [11]. Such approaches could highly improve the response times and the served throughput, however with the cost of losing the freshness of the results.

3. Related work

There are many systems and tools that offer a kind of automatic annotation of (Web) documents tailored for life sciences. There is also a majority of general purpose named entity extraction tools and services that can be exploited for many domains. Below, we first review the most related annotation systems used in the life science domain (Section 3.1), then we present several related *LOD-based* named entity extraction tools (Section 3.2), and finally we report the main differences with our approach (Section 3.3).

3.1. Annotation tools for the life science domain

Reflect [12] is a web application and service that annotates biochemical entities (in particular *genes*, *proteins* and *small molecules*) in web pages and allows the user to further explore an identified entity. By clicking the annotated term, a small popup window showing summary information is displayed, which gives a concise summary of the most important features of the selected entity (e.g. synonyms, database identifiers, 2D and 3D structures, etc.), as well as links to commonly used databases. Reflect can be extended for adding further entity types. For each entity type the Reflect dictionary needs a list that maps each synonym to an identifier. In addition, in order to create the popup content for a single entity it needs the address of a web service.

OnTheFly [13] is a service which automatically annotates document files. After submitting the file to the service, the system returns an annotated HTML version of the document. *Gene*, *protein* and *chemical* names are highlighted. By clicking on them, a popup window containing relevant information about the entity such as sequence, organism, formula for chemicals, etc., appears. The user can select the preferred organism whose protein aliases will be used for the tagging and network generation.

Domeo annotation toolkit [14] is a collection of software components that enables users to create, share and curate ontology-based annotations for online documents. It supports fully automated, semi-automated, and manual *biomedical* annotation with full representation of the provenance of annotations, as well as personal or community annotations with authorization and access control. Annotations are represented using the Annotation Ontology (AO) RDF model [15]. However, Domeo is currently being extended to also support the Open Annotation Data Model [16]. Its user interface is an extensible web component which enables direct biomedical annotation of HTML and XML documents. Domeo performs entity mining and accesses ontologies as well as other automated markup facilities via web service calls. In addition, it provides bibliographic reference lookup and identification through PubMed web services, and also supports curation of the text mining results.

Utopia documents [17] is a desktop application for reading and exploring PDF files like scientific papers. By exploiting domain-specific ontologies and plugins, it links both explicit and implicit information (of biological or chemical interest) embedded in the articles to online resources. Utopia Documents allows editors and authors to annotate terms with definitions from online resources (e.g. Wikipedia), and allows readers to easily find these definitions. It also transforms static tables and figures into dynamic, interactive objects and simplifies the process of finding related articles by automatically linking references to their digital online versions. Via its plugins it has access to a wealth of bioinformatics data: each plugin uses appropriate client libraries to access web-service endpoints and other remotely accessible resources, such as relational databases and RDF stores.

Whatizit [18] is a text processing system that allows a user to perform text-mining tasks. Whatizit identifies *molecular biology terms* and links them to related (publicly available) databases. The identified terms are wrapped with XML tags that carry additional information, such as the keys to the databases where relevant information is kept. Any vocabulary can be integrated into Whatizit as a pipeline and also several vocabularies can be integrated in a single pipeline. Examples of already integrated vocabularies are *Swissprot*, the *Gene Ontology*, the *NCBI's taxonomy* and *Medline Plus*.

The **NCBO Annotator** [19] is an ontology-based web service for annotating textual biomedical data with biomedical ontology concepts, thus facilitating data integration and translational scientific discoveries. The NCBO Annotator provides access to almost two hundred ontologies from BioPortal and UMLS and is an alternative to manual annotation through the use of a concept recognition tool. The annotator is not limited to the syntactic recognition of terms, but also leverages the structure of the ontologies to expand annotations. Such annotations allow unstructured free-text data to become structured and standardized, and also contribute to create a biomedical Semantic Web that facilitates data integration.

3.2. LOD-based named entity extraction

DBpedia spotlight [20] is a REST API tool for annotating mentions of DBpedia resources in the text, providing a solution for linking unstructured information sources to the LOD. It discovers and returns entities that are found in text, ranks them depending on how relevant they are with the text content, and links them to URIs from DBpedia. Regarding configurability, users can provide whitelists (allowed) and blacklists (forbidden) of resource types for annotation. The available types are derived from the class hierarchy provided by the DBpedia Ontology. Interesting resources can be constrained using a SPARQL query.

AlchemyAPI [21] is an NLP service that provides a scalable platform for analyzing web pages, documents and tweets along with APIs for integration. The retrieved entities are also ranked based on their importance in the given text. In addition, the named entity extractor is able to disambiguate the detected entities, link them to various datasets on the LOD and resolve co-references.

Calais [22] is a toolkit that allows incorporating semantic functionality within a blog, content management system, website or application. The OpenCalais Web Service automatically creates semantic metadata for the submitted content. Using NLP, machine learning and other methods, it analyzes a document, finds the entities within it and gives them a score based on their text relevance. In addition, it supports automatic connection to the LOD.

AIDA [23] is a framework and online tool for entity detection and disambiguation. Given a natural-language text, AIDA maps mentions of ambiguous names onto entities registered in the YAGO2 KB [24]. It accepts the plain text, HTML as well as semi-structured inputs like tables, lists, or short XML files. AIDA is centered around collective disambiguation exploiting the prominence of entities, similarity between the context of the mention and its candidates, and the coherence among candidate entities for all mentions.

Wikimeta [25] is an NLP semantic tagging and annotation system that allows incorporating semantic knowledge within a document, website or content management system. It links each detected named entity with an entity in DBpedia based on a disambiguation process. The datasets used to train the NLP tools are derived from Wikipedia.

3.3. Proposed approach

The `Theophrastus` system applies a different LOD-based approach, focused on *configurability*. Specifically, it exploits the dynamic and open nature of LOD for configuring the entities of interest, as well as for specifying how to link, enrich and browse the identified entities and the related semantic information. This enhanced configurability allows the user to dynamically configure the system even while it is running (through a configuration page). On the contrary, existing systems require the reconfiguration and redeployment of the underlying service, which can be a laborious task. For instance, users cannot easily control and change the returned information for the identified entities (e.g. show only properties useful for a particular application, restrict properties to a specific language, etc.).

In addition, the described system provides exploration and browsing services to the users, by exploiting a number of available semantic repositories and the expressive power of SPARQL. These services are offered on-demand and at real-time over fresh and ever-changing information that exists in different sources. The system does not index semantic information like other existing LOD-based approaches and does not use any kind of cache. It only indexes lists of entities (gazetteers) regarding the supported categories of entities. This makes the system *lightweight* and *portable*, providing its services over *fresh* and *inter-linked* information.

4. Conclusion

In this work we presented `Theophrastus`, a system for marine biologists that annotates and explores on demand and at real-time (web) documents with entities described in sources of the Semantic Web. For being configurable, we showed how the existing LOD can be exploited for this purpose, making the LOD accessible to the end users. As a result, the tool can be used in a plethora of contexts (e.g. web browsing, pdf reading) and in different domains. Since the supported process takes place at query time, this approach can tackle the constant evolution of vocabularies, terminologies and entity lists. We showed the efficiency of the approach and found out that the major bottleneck is the reliability and performance of online SPARQL endpoints, due to their immature technology. We expect this limitation to be removed in the near future.

Regarding the extension of this work, we plan to provide the described annotation services over RDFa, Microdata, and the W3C Open Annotation Community Standard [16], allowing publishers to automatically enrich their documents with LOD. Moreover, it would be interesting to show how the identified entities are connected in general and study approaches for ranking the matching URIs. We also plan to mine not only correct entities but entire conceptual models that describe and relate the identified entities and other external entities.

Acknowledgments

This work was partially supported by the *iMarine* Project (FP7 Research Infrastructures, 2011–2014) and *MUMIA* COST action (IC1002, 2010–2014).

References

[1] C. Bizer, T. Heath, T. Berners-Lee, Linked data-the story so far, Int. J. Semant. Web Inf. Syst. 5 (3) (2009) 1–22.
[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: a nucleus for a web of open data, in: The Semantic Web, Springer, 2007, pp. 722–735.
[3] P. Fafalios, I. Kitsos, Y. Marketakis, C. Baldassarre, M. Salampasis, Y. Tzitzikas, Web searching with entity mining at query time, in: Procs of the 5th Information Retrieval Facility Conference, Vienna, Austria, July 2012.

[4] GATE ANNIE. https://gate.ac.uk/ie/annie.html.
[5] SPARQL 1.1 Federated Query, W3C Recommendation, 21 March 2013. http://www.w3.org/TR/sparql11-federated-query/.
[6] O. Ambrus, K. Möller, S. Handschuh, Konduit VQB: a visual query builder for SPARQL on the social semantic desktop, in: Workshop on Visual Interfaces to the Social and Semantic Web, 2010.
[7] A. Russell, P.R. Smart, D. Braines, N.R. Shadbolt, Nitelight: a graphical tool for semantic query construction, in: Semantic Web User Interaction Workshop, April 2008.
[8] M.T. Enrico Franconi, Paolo Guagliardo, Quelo: a NL-based intelligent query interface, in: Procs of the Second Workshop on Controlled Natural Languages, 2010.
[9] J. Umbrich, M. Karnstedt, A. Hogan, J.X. Parreira, Hybrid SPARQL queries: fresh vs. fast results, in: The Semantic Web—ISWC 2012, Springer, 2012, pp. 608–624.
[10] Y. Tzitzikas, C. Alloca, C. Bekiari, Y. Marketakis, P. Fafalios, M. Doerr, N. Minadakis, T. Patkos, L. Candela, Integrating heterogeneous and distributed information about marine species through a top level ontology, in: Procs of the 7th Metadata and Semantic Research Conference, MTSR'13, Thessaloniki, Greece, November 2013.
[11] V. Cardellini, M. Colajanni, P.S. Yu, Dynamic load balancing on Web-server systems, IEEE Internet Comput. 3 (3) (1999) 28–39.
[12] E. Pafilis, S.I. O'Donoghue, L.J. Jensen, H. Horn, M. Kuhn, N.P. Brown, R. Schneider, Reflect: augmented browsing for the life scientist, Nature Biotechnol. 27 (6) (2009) 508–510.
[13] G.A. Pavlopoulos, E. Pafilis, M. Kuhn, S.D. Hooper, R. Schneider, Onthefly: a tool for automated document-based text annotation, data linking and network generation, Bioinformatics 25 (7) (2009) 977–978.
[14] P. Ciccarese, M. Ocana, T. Clark, Open semantic annotation of scientific publications using domeo, J. Biomed. Semantics 3 (Suppl. 1) (2012) 1–14.
[15] P. Ciccarese, M. Ocana, L.J. Garcia Castro, S. Das, T. Clark, An open annotation ontology for science on Web 3.0, J. Biomed. Semantics 2 (Suppl. 2) (2011) S4.
[16] R. Sanderson, P. Ciccarese, H. Van de Sompel, Open annotation data model. W3C Community Draft, 8, 2013.
[17] T.K. Attwood, D.B. Kell, P. McDermott, J. Marsh, S. Pettifer, D. Thorne, Utopia documents: linking scholarly literature with research data, Bioinformatics 26 (18) (2010) i568–i574.
[18] D. Rebholz-Schuhmann, M. Arregui, S. Gaudan, H. Kirsch, A. Jimeno, Text processing through Web services: calling Whatizit, Bioinformatics 24 (2) (2008) 296–298.
[19] C. Jonquet, N. Shah, C. Youn, C. Callendar, M.-A. Storey, M. Musen, NCBO annotator: semantic annotation of biomedical data, in: International Semantic Web Conference, 2009.
[20] P.N. Mendes, M. Jakob, A. García-Silva, C. Bizer, Dbpedia spotlight: shedding light on the Web of documents, in: Procs of the 7th International Conference on Semantic Systems, ACM, 2011, pp. 1–8.
[21] Alchemyapi. http://www.alchemyapi.com/.
[22] Opencalais. http://www.opencalais.com/.
[23] M.A. Yosef, J. Hoffart, I. Bordino, M. Spaniol, G. Weikum, AIDA: an online tool for accurate disambiguation of named entities in text and tables, Proc. VLDB Endow. 4 (12) (2011) 1450–1453.
[24] J. Hoffart, F.M. Suchanek, K. Berberich, E. Lewis-Kelham, G. De Melo, G. Weikum, YAGO2: exploring and querying world knowledge in time, space, context, and many languages, in: Procs of the 20th International Conference Companion on World Wide Web, ACM, 2011, pp. 229–232.
[25] Wikimeta. http://www.wikimeta.com/.