Scientific Journal of Impact Factor (SJIF): 4.14

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 3, Issue 12, December -2016

Study on Wikipedia Mining Using an Open-Source Toolkit

Rashmi S. Dharwadkar[1], Dr. Mrs. Neeta A. Deshpande[2]

ME Scholar, Dept. of Computer Engineering, D Y Patil College of Engineering, Akurdi, Pune, MH, India[1]
Associate Professor, Dept. of Computer Engineering, D Y Patil College of Engineering, Akurdi, Pune, MH, India[2]

Abstract: *The online encyclopedia Wikipedia is a vast, regurly evolving tapestry of interlinked articles. For developers and researchers it represents a huge multilingual database of concepts and semantic relations, a potential resource for natural language processing and many other research areas. This paper introduces the Wikipedia Miner toolkit, an open-source software system that allows researchers and developers to accommodate Wikipedia's rich semantics into their own applications. The toolkit creates databases that contain summarized versions of Wikipedia's content and structure, and includes a Java API to provide access to them. Wikipedia's articles, categories and redirects are represented as classes, and can be regularly searched, browsed, and iterated over. Advanced features include parallelized processing of Wikipedia dumps, machine-learned semantic relativity measures and annotation features, and XML-based web services. Wikipedia Miner is calculated to be a platform for sharing data mining techniques.*

Keywords: *Weka machine, Miner toolkit, MapReduce, Berkeley DB, big data.*

I. INTRODUCTION

The online encyclopedia Wikipedia is a vast, constantly evolving tapestry of richly interlinked textual information. To a growing community of researchers and developers it is an ever-growing source of manually defined concepts and semantic relations. It constitutes an unparalleled and largely untapped resource for natural language processing, knowledge management, data mining, and other research areas.

Those who wish to draw on Wikipedia as a source of machine-readable knowledge have two options. They can either base their work on secondary structures that others have extracted from it, such as Freebase [1] and Yago [2], or they can start from scratch and build their own algorithms to mine Wikipedia directly. The first approach is the easiest. However— as this special issue demonstrates—new innovations and mining techniques are introduced regularly, potentially rendering obsolete pre-built resources that rely on the current state of the art. Furthermore, unless they are studiously maintained, such resources forego one of Wikipedia's greatest strengths: its propensity to grow rapidly and keep abreast of world events. The second option—working directly from the source—allows researchers to continue to innovate and find new ways to mine knowledge from Wikipedia. Moreover, any new information added by Wikipedians flows directly through. Wikipedia's entire content is readily available under a Creative Commons license, with regular releases in the form of large XML and HTML dumps.[1] unfortunately, substantial effort is needed to mine these dumps: they are enormous and replete with cryptic markup.

This paper introduces a third option: to share algorithms and code rather than secondary resources. We introduce Wikipedia Miner, an open-source toolkit that allows its users to sidestep the laborious effort needed to mine Wikipedia's riches. It also provides a platform for sharing mining techniques, and for taking advantage of powerful technologies like the distributed computing framework Hadoop [3] and the Weka machine learning workbench [4].

The paper is structured as follows. Section II provides a broad overview of the toolkit. The following two sections elab-orate on two of its unique features: Section III describes and evaluates its algorithms for measuring semantic relatedness using Wikipedia as background knowledge, and Section IV does the same for detecting and disambiguating Wikipedia topics when they are mentioned in documents. The application of these features is illustrated in Section V, which demonstrates how a simple thesaurus browser and document annotator can be constructed with minimal code. Section VI reviews related work, including alternative and complementary resources for mining Wikipedia, and projects that apply the toolkit to various research problems. The paper concludes with a discussion of Wikipedia Miner's features and limitations, and points out directions for future development. Before continuing, it may be valuable to clarify some of

the terminology that will be used in the remainder of the paper. Because Wikipedia articles are typically homogenous—they are each dedicated to describing a single topic—they can be treated the same as descriptors in a thesaurus or concepts in an ontology. Consequently we use the terms article, topic and concept interchangeably throughout the paper. Articles are referred to in textual documents by words or phrases that we call terms or labels; again, we use these interchangeably. Where a label may refer to multiple concepts (i.e. it is ambiguous), we refer to these concepts as senses.

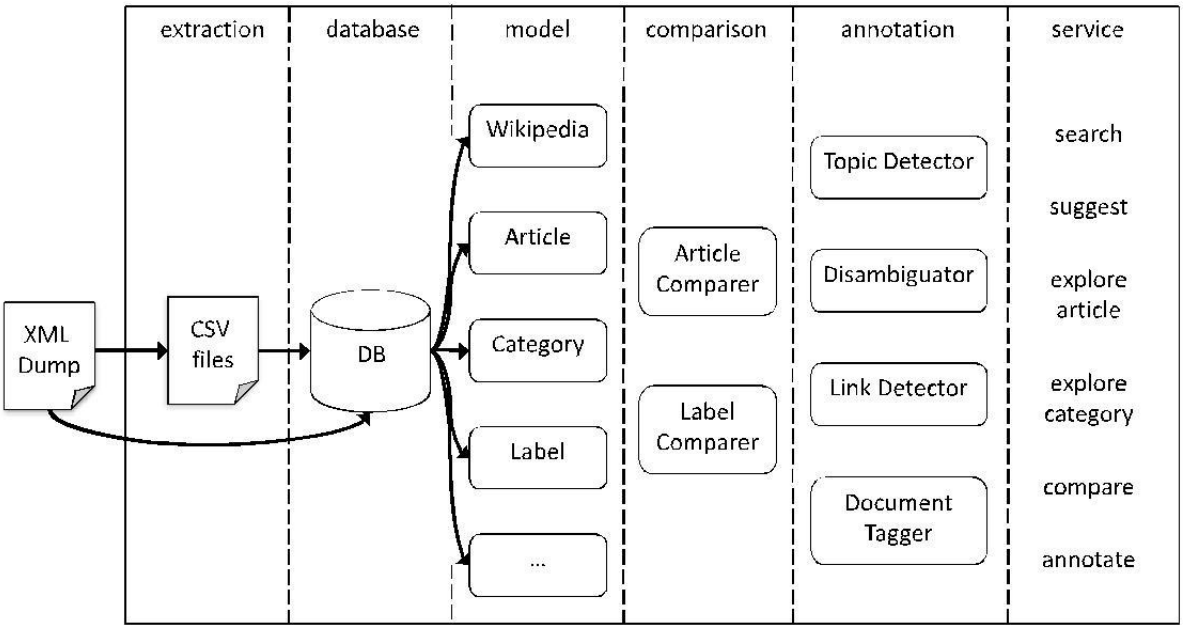


Figure 1. Architecture of the Wikipedia Miner toolkit.

II. THE WIKIPEDIA MINER TOOLKIT

Figure 1. Illustrates the overall architecture of the Wikipedia Miner toolkit, which is implemented entirely in Java. The figure begins on the left with a single large XML file containing the full content of a particular edition of Wikipedia, obtained directly from the Wikimedia Foundation, excluding revision history, background discussion and multimedia content. This file is fed into a run-once extraction process (found within the extraction package described in Section 2.1), which produces a series of flat-file summaries of Wikipedia's structure.

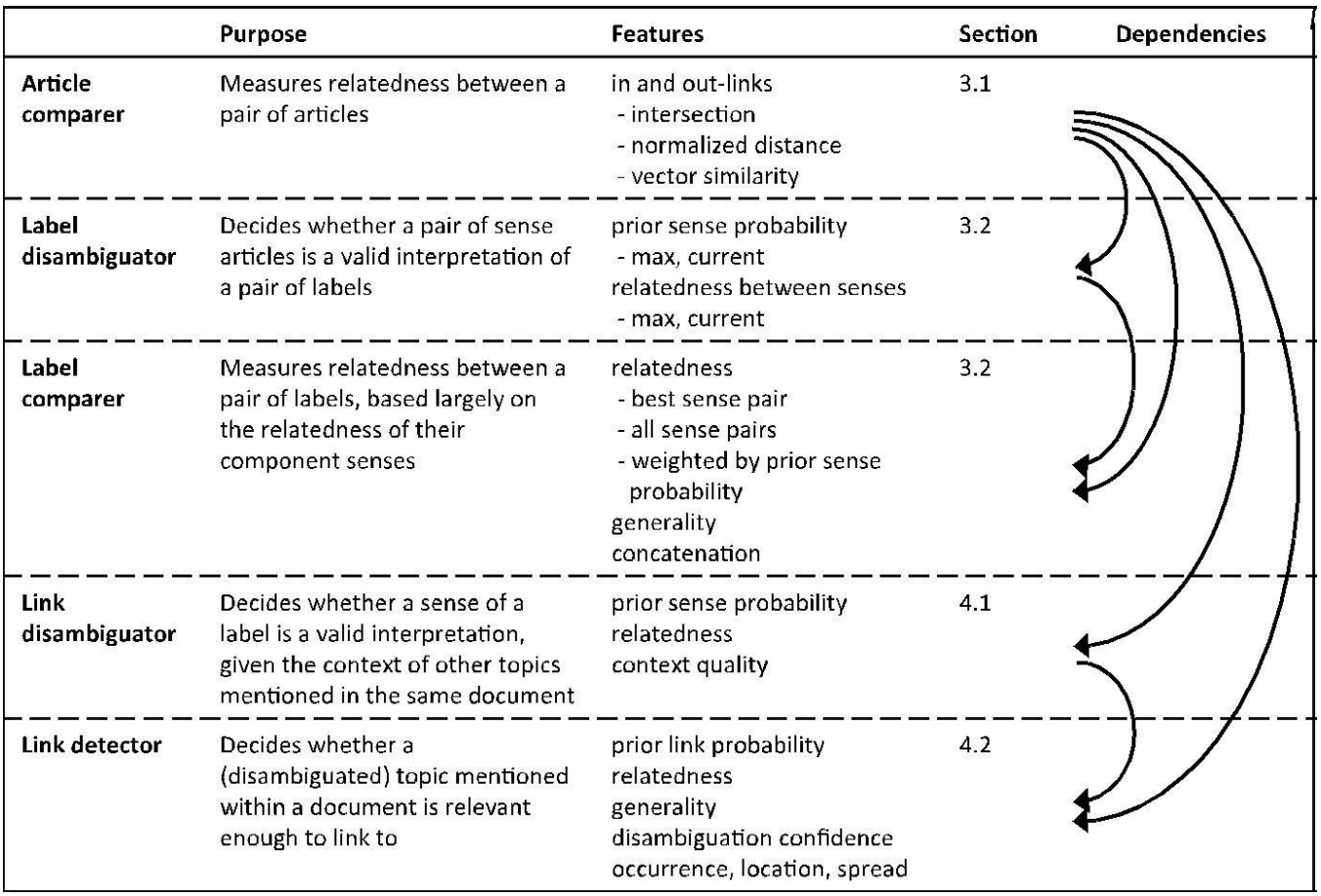
These summaries are simply delimited text files, and developers could construct programs to read them directly. This would, however, require significant computer time and memory—the link-graph summary alone occupies more than 1 GB. Instead, both the summaries and the original XML dump are fed into a database environment, managed by the database package described in Section 2.2, so that they can be indexed persistently. As Section 2.3 explains, the model package of the toolkit simplifies access to the stored data by wrapping it with easy to understand, thoroughly documented classes, such as Wikipedia, Article and Category.

Using the features described up to this point requires a significant commitment from the user: one has to download Wikipedia in its entirety and invest many machine hours preprocessing it. The toolkit's final component is the service package, a suite of web services that allows casual users to explore its functionality. Section 2.6 gives a brief overview of this package.

2.1. Extraction

The toolkit's extraction package is responsible for gathering summary data from Wikpedia's XML dumps. We will not describe the process or the data it produces in detail, as users are unlikely to interact with it directly.

The extraction process exploits Hadoop, an open source implementation of Google's proprietary GFS file system [5] and MapReduce technology [6]. The former implements a reliable file system distributed across clusters of machines, while the latter supports a programming paradigm that greatly simplifies sharing work between multiple machines. Combining the two yields a powerful platform for processing "big data" that is endorsed by Internet giants Yahoo, Facebook, Twitter, and many others. This Hadoop-powered extraction process is extremely scalable. Given a cluster of 30 machines, each with two 2.66 GHz processors and 4 GB of RAM, it processes the latest versions of the full English Wikipedia—3.3 million articles, 27 GB of uncompressed markup in a little over 2.5 hours.[3] The process scales roughly linearly with the size of Wikipedia and the number of machines available.



	Purpose	Features	Section	Dependencies
Article comparer	Measures relatedness between a pair of articles	in and out-links - intersection - normalized distance - vector similarity	3.1	
Label disambiguator	Decides whether a pair of sense articles is a valid interpretation of a pair of labels	prior sense probability - max, current relatedness between senses - max, current	3.2	
Label comparer	Measures relatedness between a pair of labels, based largely on the relatedness of their component senses	relatedness - best sense pair - all sense pairs - weighted by prior sense probability generality concatenation	3.2	
Link disambiguator	Decides whether a sense of a label is a valid interpretation, given the context of other topics mentioned in the same document	prior sense probability relatedness context quality	4.1	
Link detector	Decides whether a (disambiguated) topic mentioned within a document is relevant enough to link to	prior link probability relatedness generality disambiguation confidence occurrence, location, spread	4.2	

Figure 2. Machine-learned classifiers used within the toolkit.

2.2. Storage, indexing and caching

The database package provides persistent, appropriately indexed access to the summarized data and original markup. It depends heavily on Berkeley DB JE [29], an open-source Java-based storage engine maintained by Oracle. Users interact with the database via the model package that wraps it (Section 2.3), so we will not elaborate on its content.

The performance of this file-based database is a bottleneck for many applications. Although Berkeley DB caches data to memory if it is accessed repeatedly, it will be very slow for applications that require millions of lookups; such as the semantic relatedness and annotation experiments that follow. Fortunately, the toolkit allows any of its databases to be cached to memory in their entirety, greatly reducing access time. Users can choose which databases are cached, depending on the tasks that need optimizing and the amount of memory available. They can also specify articles that should not be cached, in which case the toolkit does not waste time retrieving them from disk, but instead behaves as if these articles do not exist. Common strategies are to avoid articles that are extremely short or receive few links from other articles.

Wikipedia contains many of these: they would require significant space to cache and are likely to be of limited use. Users can also specify whether databases should be cached directly, prioritizing speed, or in compressed form, prioritizing space.

III. MODELING

The main access point to the toolkit is the model package, which abstracts away from the data to provide simplified, object-oriented access to Wikipedia's content and structure. Figure 3. gives an overview of its most important classes, along with their inheritance hierarchy and some selected properties and methods.

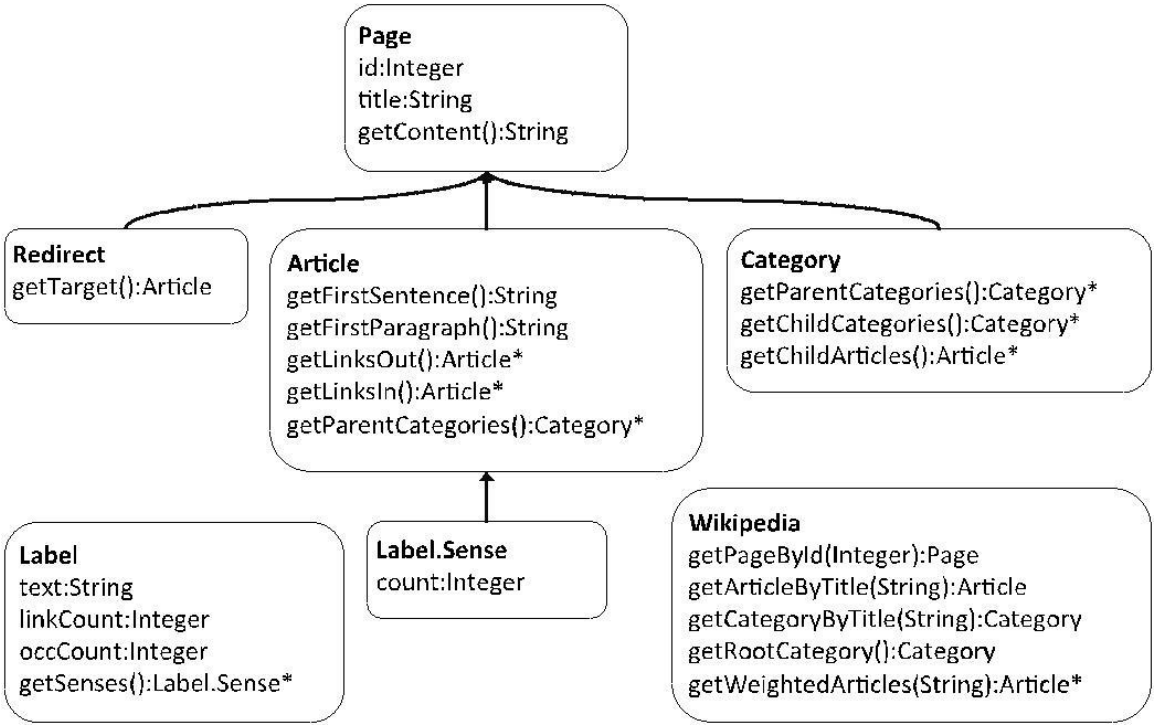


Figure 3. Classes, properties and methods from the *model* package.

3.1. Wikipedia

Wikipedia itself is, of course, one of the more important objects to model. This class provides the central point of access to most of the toolkit's functionality. Among other things, here users can gather basic statistics about the encyclopedia, or access the pages within it through iteration, browsing, and searching.

3.2. Page

All of Wikipedia's content is presented on pages of one kind or another. The toolkit models every page as a unique id, a title, a type, and some content expressed as Media Wiki markup. More specific functionality depends on the type of page.

3.3. Article

Articles supply the bulk of Wikipedia's informative content. Each article describes a single concept or topic, and their titles are succinct, well-formed phrases that can be used as descriptors in ontologies and thesauri. For example, the article about domesticated canines is entitled Dog, and the one about companion animals in general is called Pet. Articles follow a predictable layout, which allows the toolkit to provide short and medium length definitions of concepts by extracting the corresponding article's first sentence and paragraph.

Once a particular article is identified, related concepts can be gathered by mining the articles it links to, or those that link to it. However, many of the links do not correspond to semantic relations, and it is di cult to separate useful links from irrelevant ones. Section 3.1 describes how this is resolved by considering links in aggregate, rather than individually.

Articles often contain links to equivalent articles in other language versions of Wikipedia. The toolkit allows the titles of these pages to be mined as a source of translations.

3.4. Redirect

Redirects are Wikipedia pages whose sole purpose is to connect articles to alternative titles that correspond to synonyms and other variations in surface form. For example, dogs, canis lupus familiars, and domestic dog redirect to the article entitled Dog. Redirects may also represent more specific topics that do not warrant separate articles, such as male dog and dog groups. The toolkit allows redirects to be mined for their intended target, and articles to be mined for all redirects that refer to them.

3.5. Category

Almost all Wikipedia's articles are organized within one or more categories, which can be mined for hyponyms, holonyms and other broader, more general, topics. Dog, for example, belongs to the categories domesticated animals, cosmopolitan species, and scavengers. If a topic is broad enough to warrant several articles, the central article may be paired with a category of the same name: the article dog is paired with the category dogs. This equivalent category can be mined for more parent categories (canines) and subcategories (dog breeds, dog sports). Child articles and other descendents (puppy, fear of dogs) can also be mined for hypernyms, meronyms, and other more specific topics.

All Wikipedia's categories descend from a single root. The toolkit uses the distance from the root to a particular article or category to provide a measure of its generality or specificity. According to this measure, dog is more specific than carnivores, which has the same specificity as omnivores and is more specific than animals.

3.6. Label

Wikipedia provides several structural elements that associate articles with terms or surface forms that can be used to denote them. The most obvious elements are article titles and redirects: the article about dogs is given the title Dog and has about 30 redirects, including canis familiaris, domestic dog and man's best friend. The links made from other Wikipedia articles to this one provide additional surface forms, because authors tailor anchor text to suit the surrounding prose. A scientific article may contain a link to Dog that is labeled with the anchor text canis lupus familiaris, while a more informal article may refer to doggy.

By default, labels are indexed and searched without modifying them in any way. They already encode many of the desired variations in letter case (Dog and dog), pluralism (dogs), and punctuation (US and U.S.), so automatic term conflation is often unnecessary and may introduce erroneous matches—returning digital on-screen graphic (or DOG) as a match to dog, for example. When modification is desirable, the toolkit provides several text processors—case-folders, stemmers, and punctuation cleaners—to re-index the labels. Users can also develop and apply their own text processors.

IV. COMPARISON

The comparison package for measuring relatedness, both between pairs of concepts and between pairs of terms. The annotation package for detecting and disambiguating concepts when they are mentioned in textual documents. Both packages rely heavily machine-learned classifiers, which are listed in Figure 2. The figure also summaries the features these classifiers draw on, the chain of how the output of one classifier flows on to the next, and the sections in the paper that discuss them in detail.

The toolkit's comparison package contains algorithms for generating semantic relatedness measures, which quantify the extent to which different words or concepts relate to each other. According to these algorithms, dog is 84% related to canine, 72% related to pet, and 66% related to animal. These measures have a wide range of applications—including word-sense disambiguation [7], spelling correction [8], and document clustering [9]—because they allow terms and concepts to be compared, organized, and perhaps even reasoned about.

The package contains two main classes: Article Comparer, which measures relatedness between pairs of Wikipedia articles, and Label Comparer, which measures relatedness between pairs of terms and phrases. Section 3 explains how these classes work and evaluates their performance.

V. ANNOTATION

The annotation package provides tools for identifying and tagging Wikipedia topics when they occur in textual documents. Documents are processed in five main steps: cleaning them, detecting terms (including phrases) that could refer to topics, resolving ambiguous terms to create a functional mapping from term occurrences to topics, predicting the salience of each topic, and marking up the document with references to these topics. The toolkit is intended to be modular, so separate classes handle each step.

The annotation process begins by feeding a document into a Document Preprocessor. This abstract class is responsible for identifying regions of the document that should not be altered and from which Wikipedia topics should not be mined. The toolkit provides code for processing HTML and MediaWiki markup, and allows users to develop new preprocessors.

Preprocessors produce Preprocessed Documents. These are copies of the original document that distinguish markup from content, so that the latter can be manipulated and tagged without invalidating the former. Processed documents keep track of text that is ineligible for tagging but may be useful for understanding or disambiguating content, such as the content of HTML title and meta tags and the anchor text of existing links.

The Topic Detector gathers all labels in the document whose prior link probability (Section 2.3.6) exceeds a configurable threshold. This acts as a rough filter to discard terms and phrases that are rarely used within Wikipedia articles to refer to other Wikipedia articles.

The labels gathered by the topic detector are often ambiguous in that they refer to multiple articles—that is, multiple concepts. To resolve ambiguity, each candidate concept from each detected label is fed in turn into a Disambiguator, which produces a probability that the concept is relevant for that label in the context of the given document. This disambiguator uses machine learning, and is trained using the article-to-article links that Wikipedia already contains. Each existing link provides one positive example, namely its chosen destination, and several negative examples, namely the destinations that have been chosen for this link text in other articles but not this one.

Many of the detected topics will be questionable. At this stage, no effort has been made to distinguish those that are central to the document from ones that are only mentioned in passing. For example, the topic Dog has the same status in a document about pet registration as it does in one that describes the weather as “raining cats and dogs”. In the latter document, the family movie Cats & Dogs is considered no less important than the topic Rain. The Topic Weighter abstract class provides a blueprint for classes that are responsible for identifying the importance or relevance of topics within documents. Currently the toolkit provides only one topic weighter: the Link Detector. This class is based on the idea that every existing Wikipedia article is an example of how to separate relevant topics ones that authors chose to link to from irrelevant ones. For each topic in a new document, the link detector calculates a weight based on how well it fits the model of what Wikipedians would choose to link to if the document were a Wikipedia article. Section 4.2 gives details of this algorithm.

By this point, many users of the toolkit will have achieved what they need: a list of Wikipedia topics for any given document, weighted by their relevance to it. This list could be used as a concept-based representation of the document in applications like categorization, clustering, retrieval, and so on.

For other applications, the detected topics should be injected back into the original document. To achieve this, the Pre-processed Document and the list of detected Topics can be fed into a Document Tagger, which produces a new version of the original document, marked up to identify the topics. The tagger assumes that tags should not be nested, and resolves colli- overlapping mentions of topics. Imagine, for example, that a document mentioning “cats and dogs” was processed and Cat, Dog, and Cats & Dogs were all given to the tagger as relevant topics. The tagger decides whether to create a single tag that refers to the family movie or two tags that refer to each animal separately.

Like the preprocessor, the tagger can be tailored to generate different styles of markup. HTML Document Tagger creates HTML links to Wikipedia, while Media Wiki Document Tagger identifies links using standard Media Wiki markup. Users can also develop their own document taggers.

VI. SERVICE

The service package provides web-based access to a subset of the toolkit's functionality, via REST-style XML-over-HTTP web services. These services are hosted as a publicly available demonstration we strongly recommend readers try them out and can be redeployed by anyone who hosts the toolkit. The services are briefly described below. Further details (and hands-on experience) are available at the toolkit's website.

The explore Article service takes either the title or unique id of an article, and returns details such as textual definitions, alternative labels, in- and out-links, and representative image icons. The explore Category provides similar functionality for categories. The search service matches queries against Wikipedia's label vocabulary breaking complex queries into their component terms as necessary—and lists the different senses each label can refer to. The suggest service takes a set of article ids (such as a selection of the senses returned by the search service) and returns lists of related articles, organized by the categories they belong to.

The compare service takes a pair of terms, a pair of article ids, or a set of ids. It returns measures of how strongly the terms or articles relate to each other, and can optionally return details of how ambiguous terms have been interpreted, and lists of articles and textual snippets that help to explain the relation.

The annotate service takes a snippet of HTML or Media Wiki markup, or a web-accessible URL, and returns the markup augmented with links to the relevant Wikipedia articles. It can optionally return details of how relevant each topic is to the rest of the document

CONCLUSION

In his paper introduces the Wikipedia Miner toolkit, an open-source software system that allows researchers and developers to integrate Wikipedia's rich semantics into their own applications. The toolkit creates databases that contain summarized versions of Wikipedia's content and structure, and includes a Java API to provide access to them. Wikipedia's articles, categories and redirects are represented as classes, and can be efficiently searched, browsed, and iterated over.

REFERENCES

[1] K. Bollacker, R. Cook, P. Tufts, Freebase: A shared database of structured general human knowledge, in: 22nd National Conference on Artificial Intelli-gence (AAAI), Vancouver, Canada, 2007, pp. 1962–1963.

[2] F.M. Suchanek, G. Kasneci, G. Weikum, Yago: A large ontology from Wikipedia and WordNet, Web Semantics: Science, Services and Agents on the World Wide Web 6 (3) (2007) 203–217.

[3] T. White, Hadoop: The Definitive Guide, second edition, O'Reilly Media/Yahoo Press, 2010.

[4] I.H. Witten, E. Frank, M.A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, third edition, Morgan Kaufmann, Burlington, MA, 2011.

[5] S. Ghemawat, H. Gobioff, S. Leung, The Google file system, ACM SIGOPS Operating Systems Review 37 (5) (2003) 29–43.

[6] J. Dean, S. Ghemawat, MapReduce: Simplified data processing on large clusters, Communications of the ACM 51 (1) (2008) 107–113.

[7] U.S. Kohomban, W.S. Lee, Learning semantic classes for word sense disambiguation, in: 43rd Annual Meeting of the Association for Computational Linguistics, Ann Arbor, MI, 2005, pp. 34–41.

[8] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, E. Ruppin, Placing search in context: The concept revisited, ACM Transactions on Information Systems 20 (1) (2002) 116–131.

[9] A. Huang, D. Milne, E. Frank, I.H. Witten, Clustering documents using a Wikipedia-based concept representation, in: 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Bangkok, Thailand, 2009, pp. 628–636.

[10] O. Medelyan, C. Legg, Integrating Cyc and Wikipedia: Folksonomy meets rigorously defined common-sense, in: Wikipedia and Artificial Intelligence: An Evolving Synergy, Chicago, IL, 2008.

[11] O. Medelyan, D. Milne, Augmenting domain-specific thesauri with knowledge from Wikipedia, in: New Zealand Computer Science Research Student Conference, Christchurch, New Zealand, 2008.

[12] O. Medelyan, D. Milne, C. Legg, I.H. Witten, Mining meaning from Wikipedia, International Journal of Human-Computer Studies 67 (9) (2009) 716–754.

[13] O. Medelyan, I.H. Witten, D. Milne, Topic indexing with Wikipedia, in: Wikipedia and Artificial Intelligence: An Evolving Synergy, Chicago, IL, 2008.

[14] V. Nastase, D. Milne, K. Filippova, Summarizing with encyclopedic knowledge, in: 2nd Text Analysis Conference, National Institute of Standards and Technology, Gaithersburg, MA, 2009.

[15] R. Navigli, S.P. Ponzetto, BabelNet: Building a very large multilingual semantic network, in: 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 2010, pp. 216–225.

