

An Open-Source Toolkit for Mining Wikipedia

David Milne

Department of Computer Science, University of Waikato

Private Bag 3105, Hamilton, New Zealand

+64 7 856 2889 (ext. 6038)

d.n.milne@gmail.com

ABSTRACT

The online encyclopedia Wikipedia is a vast repository of information. For developers and researchers it represents a giant multilingual database of concepts and semantic relations; a promising resource for natural language processing and many other research areas. In this paper we introduce the Wikipedia Miner toolkit: an open-source collection of code that allows researchers and developers to easily integrate Wikipedia's rich semantics into their own applications.

The Wikipedia Miner toolkit is already a mature product. In this paper we describe how it provides simplified, object-oriented access to Wikipedia's structure and content, how it allows terms and concepts to be compared semantically, and how it can detect Wikipedia topics when they are mentioned in documents. We also describe how it has already been applied to several different research problems. However, the toolkit is not intended to be a complete, polished product; it is instead an entirely open-source project that we hope will continue to evolve.

Categories and Subject Descriptors

I.2.7 [Artificial Intelligence]: Natural Language Processing – *text analysis*

General Terms

Algorithms, Documentation.

Keywords

Wikipedia, Toolkit, API, Open Source Software, Data Mining, Knowledge Management.

1. INTRODUCTION

For the general public, Wikipedia represents a vast source of knowledge. To a growing community of researchers and developers it also represents a huge, constantly evolving collection of manually defined concepts and semantic relations. It is a promising resource for natural language processing, knowledge management, data mining, and other research areas.

This paper describes Wikipedia Miner,¹ an open-source toolkit that allows developers and researchers to painlessly integrate the rich semantics encoded within Wikipedia into their own projects. The toolkit is a mature product with state-of-the-art functionality. Although these features were developed by the author, the toolkit has been released as an open-source project. It has already been

applied to a wide array of research problems, and is free to evolve and be used in any way that the research community sees fit.

The remainder of the paper is structured as follows. Section 2 provides an overview of the toolkit and elaborates on some of its more important functions. The use of these features is made more concrete in Section 3, which demonstrates how a simple thesaurus browser can be constructed with a minimal amount of code. Section 4 describes related work, including alternative and complementary resources for mining Wikipedia, and examples that apply this 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.

2. THE WIKIPEDIA MINER TOOLKIT

The purpose of this work is to allow developers and researchers to easily explore and draw upon the content of Wikipedia. While this information is already readily available,² the problem is extracting and accessing useful information from it in a scalable and timely manner. The current English Wikipedia dump, without revision history or background discussion, includes approximately six million pages. Its useful semantic features are buried under 20 GB of cryptic markup.

The Wikipedia Miner toolkit includes PERL scripts for processing Wikipedia dumps, and extracting summaries such as the link graph and category hierarchy. All of the scripts scale in linear time, and can flexibly split the data where necessary in case of memory constraints. All but one of the summaries can be extracted within a day or two on modest desktop hardware. Only the link-likelihood statistics (Section 2.4.1) take longer, and these are entirely optional. The script to extract them requires approximately ten days, but can be easily parallelized to run across multiple machines. Finally, several pre-summarized versions of Wikipedia are available from the toolkit's website. The entire extraction process can be avoided unless one requires a specific edition of Wikipedia that we have not provided.

After running these scripts (or downloading the pre-prepared data), developers can construct programs to read the resulting summaries—which are simply delimited text files—directly. This would, however, require significant time and memory; the link-graph summary alone is still almost 1 GB. Instead the toolkit communicates with a MySQL database, so that the data can be indexed persistently and accessed immediately, without waiting for anything to load.

¹ Code, data and online demonstrations of the Wikipedia-Miner toolkit are available at <http://wikipedia-miner.sourceforge.net>

² Wikipedia's entire content is released every month or so as html and xml dumps at <http://download.wikimedia.org>

The last component of the toolkit is a documented Java API, which abstracts away from the data to provide simplified access to Wikipedia. The following sections describe some of its more important functions: modeling Wikipedia in an object oriented fashion; comparing terms and concepts semantically; and cross-referencing documents with relevant topics. Section 3 revisits this API to demonstrate its ease of use.

2.1 Modeling Wikipedia

In this section we describe some of the more important classes for modeling Wikipedia's structure and content, and explain how they correspond with the elements of a traditional thesaurus. These classes are shown in Figure 1, along with their inheritance hierarchy and some selected properties and methods.

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

Articles provide 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 non-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 fairly predictable layout, and consequently the toolkit can provide short and medium length definitions of concepts by extracting the first sentence and first paragraph from the articles' content.

Once a particular article is identified, related concepts can be gathered by mining the articles it links to, or the ones that link to it. Unfortunately many of these individual links do not correspond to semantic relations, and it is difficult to separate useful links from irrelevant ones. Section 2.3 describes how this is resolved by considering links in aggregate, rather than individually.

The anchor texts of the links made to an article provide a source of synonyms and other variations in surface form. The article about *dogs*, for example, has links from anchors like *canis familiaris*, *man's best friend*, and *doggy*. As we will see in Section 2.2, these anchors provide our best means of searching Wikipedia. The sheer number of links made to an article is also useful. They provide a sense of how well-known the concept is, because obscure and unknown concepts are unlikely to be referred to by other articles.

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; the article about *dogs* links to (among many others) *chien* in the French Wikipedia, *haushund* in German, and 犬 in Chinese.

Redirects are pages whose sole purpose is to connect an article to alternative titles. Like incoming anchor texts, these correspond to synonyms and other variations in surface form. The article entitled *dog*, for example, is referred to by redirects *dogs*, *canis lupus familiaris*, and *domestic 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 of the redirects that refer to them.

Categories: Almost all of 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 descendants (*puppy*, *fear of dogs*)

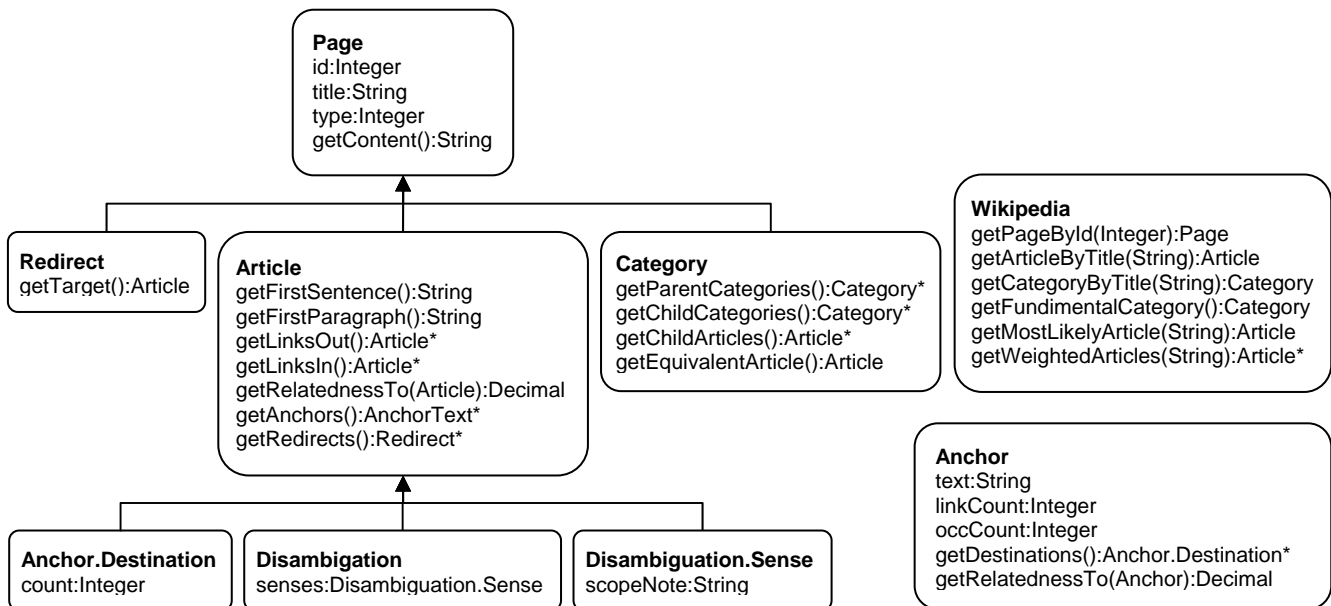


Figure 1: A sample of classes, properties and methods available in the Wikipedia-Miner toolkit.

can also be mined for hypernyms, meronyms, and other more specific topics.

All of Wikipedia’s categories descend from a single root called *Fundamental*. The toolkit uses the distance between a particular article or category and this root to provide a measure of its generality or specificity. According to this measure *Dog* has a greater distance than *carnivores*, which has the same distance as *omnivores* and a greater distance than *animals*.

Disambiguations: When multiple articles could be given the same name, a specific type of article—a disambiguation—is used to separate them. For example, there is a page entitled *dog* (*disambiguation*), which lists not only the article on domestic dogs, but also several other animals (such as *prairie dogs* and *dogfish*), several performers (including *Snoop Doggy Dogg*), and the Chinese sign of the zodiac. The toolkit separates these articles, which correspond to senses of the title term, from other links. Each of these sense pages have an additional scope note; a short phrase that explains why it is different from other potential senses.

Anchors, the text used within links to Wikipedia articles, are surprisingly useful. As described earlier, they encode synonymy and other variations in surface form, because people alter them to suit the surrounding prose. A scientific article may refer to *canis familiaris*, and a more informal one to *doggy*. Anchors also encode polysemy: the term *dog* is used to link to different articles when discussing pets, star signs or the iconic American fast food. Disambiguation pages do the same, but link anchors have the advantage of being marked up directly, and therefore do not require processing of unstructured text. They also give a sense of how likely each sense is: 76% of *Dog* links are made to the pet, 7% to the Chinese star sign, and less than 1% to *hot dogs*.

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

2.2 Searching Wikipedia

The Wikipedia Miner toolkit indexes pages so that they can be searched efficiently. The most common scenario for searching is to return an article or set of articles that could refer to the given term. When searching for *dog*, then the expected result is the set of all articles that could reasonably be given that title.

A common approach in the literature is to search over page titles, and resolve different page types by following the links implied by them. According to this scheme, matching articles are used directly, redirects are resolved to their target articles, and disambiguation pages are mined for the different senses they list. In practice, Wikipedia is not so cleanly machine-readable. Disambiguation pages are deceptively difficult to process automatically, since they are written in free text and often list items that are merely associated with the target term rather than senses of it. Page titles can also present difficulties, because they often include additional scope information—e.g. *Dog (zodiac)* or *Dog (domestic)*—that must be identified and stripped out.

We have had more success with indexing pages by the links that are made to them. We have already explained how these anchors encode synonymy in much the same way as redirect titles, except they are far more plentiful and very rarely include additional scope information. We have also explained how anchors encode polysemy directly—with no unstructured text—and provide additional statistics to separate obvious, helpful senses from obscure ones.

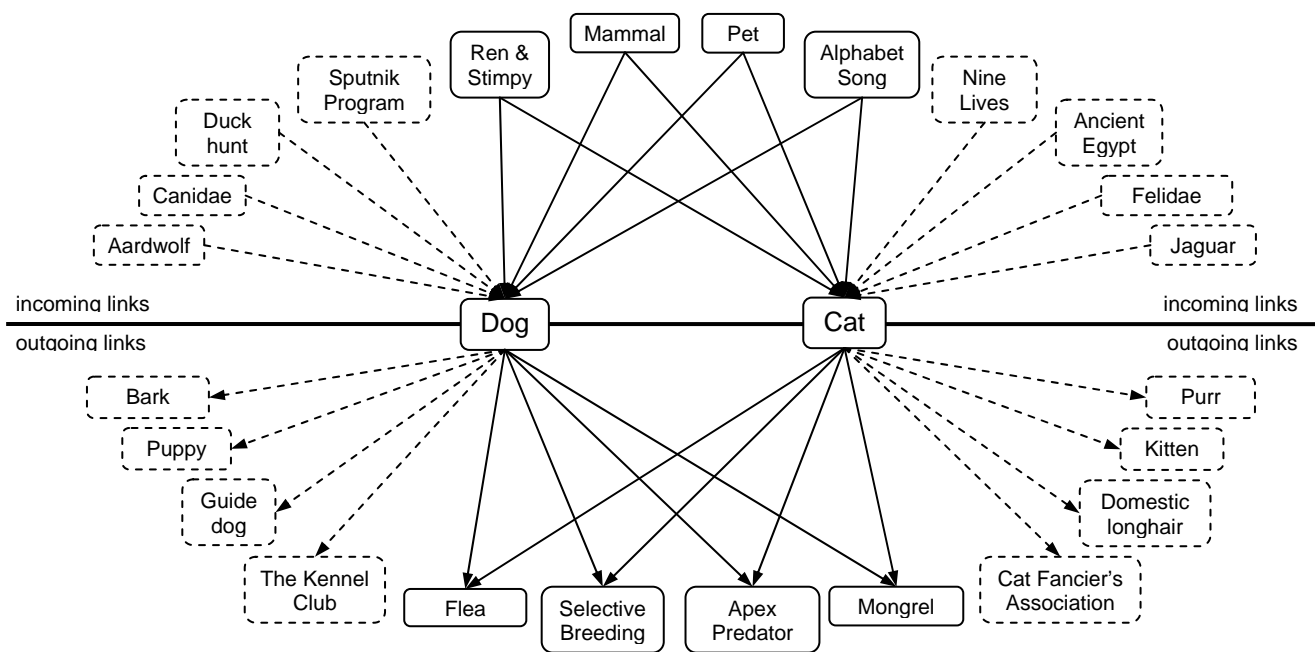


Figure 2: Obtaining a semantic relatedness measure between *Dog* and *Cat* from Wikipedia links.

By default, anchor texts 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 pages. It is also fairly simple for users to develop and apply their own text processors.

2.3 Computing Semantic Relatedness

The Wikipedia Miner toolkit includes algorithms for generating semantic relatedness measures, which quantify the extent to which different words or concepts relate to each other. According to the toolkit, *dog* is 100% related to *canis familiaris*, 47% related to *domestic animal*, and 19% related to *animal*. These measures have a wide range of applications—particularly for natural language processing, knowledge management, and data-mining—because they allow terms and concepts to be compared, organized, and reasoned with.

Wikipedia Miner’s relatedness measures are generated using the hyperlinks made between Wikipedia’s articles. These articles reference each other extensively, and at first glance the links appear to be promising semantic relations. *Dog* links to broader concepts like *mammal* and *pet*, to narrower topics such as *working dog* and *Chihuahua*, and to related concepts like *domestication* and *dog breeds*. Unfortunately it also contains links to many irrelevant concepts, such as *inch*, *color*, and *suntan lotion*. An individual link between two Wikipedia articles cannot be trusted to represent any particular semantic relation. To separate useful relations from irrelevant links, the links must be considered in aggregate.

The details of our approach (and an evaluation of it) are described more thoroughly by Milne and Witten (2008a). Figure 2 gives a rough impression of how it works by comparing the article about *dogs* with another about *cats*. If we gather the links made by these articles, shown in the bottom half of the diagram, we see some overlap to indicate that the two concepts are related; both link to *selective breeding*, *flea*, and around 40 other shared concepts (only a small sample of links are shown). There are links to that are unique to each of the concepts (*puppy* and *bark*, or *kitten* and *purr*) to indicate that they are not related, or at least not the same thing. The same process can be applied to the links that are made to each article, shown on the top half of the diagram. With some normalization, the proportion of overlapping and distinct links gives us a measure of relatedness—in this case, 77%.

The measure of relatedness described so far is between concepts; you can only compare an article with another article. It is more common to compare words or phrases instead, and the toolkit supports this by allowing anchors to be compared to other anchors. To do so it disambiguates the anchors by choosing one particular article (or sense) to represent each anchor. The algorithm for disambiguation balances the need to choose the pair of senses that most strongly relate to each other, with a preference for choosing the more obvious or likely senses. Again, the details are discussed by Milne and Witten (2008a).

2.4 Understanding and Augmenting Documents

For any given document, it is probable that Wikipedia knows at least something about the topics discussed within it and could add additional information. Figure 3, for example, shows a short news article in which several Wikipedia topics can be identified. These could be made into links to the appropriate articles—or definitions mined from them, as shown for *Labrador*—so that users could easily investigate the topics further. Connecting resources to Wikipedia in this way can improve how people digest the information within them (Csomai and Mihalcea 2007).

The detected topics can also improve how the document is modeled and understood by automatic systems. These systems—search engines, recommender systems, and so forth—typically represent documents as bags of the words they contain. By consulting Wikipedia, they could instead draw on the concepts these words represent. This is illustrated on the right of the figure, which graphs some of the detected concepts and the relations between them. In building this graph we resolve many of the problems that would otherwise degrade retrieval.

Synonymy has been resolved, so that it doesn’t matter whether the document (or the users who search for it) talks of *satellite guided positioning technology*, *global positioning system*, or *GPS*—we know that they are all the same thing. Polysemy has also been resolved. The document below discusses *president*, which according to Wikipedia could refer to any one of hundreds of different leaders of countries and organizations. From the graph we know that we could only be talking about the title in general, or the specific president of Russia. By navigating the relationships of meaning between the topics, the main threads of discussion can be identified: there is a cluster of topics relating to the dog, another to the president, and another to the technology. More formal reasoning can be made available by taking the (trivial) step of connecting to Wikipedia-derived ontologies such as those described in Section 4.1. All of this adds up to a machine readable representation of the document that is extremely informative.

The challenge is to detect these topics and create the appropriate links accurately and efficiently. The toolkit provides proven algorithms to do just that. This section briefly explains the three steps involved; *candidate selection* (gathering the terms and phrases to be considered for linking), *disambiguation* (deciding where a link should go to) and *detection* (deciding whether a link should be made at all). Further details about the algorithm and its evaluation can be found in Milne and Witten (2008b), and an online demonstration is available on the toolkit’s website.

2.4.1 Candidate Selection

The process for detecting topics in a document starts by gathering all n-grams within it, and consulting the anchor vocabulary to find out if Wikipedia knows about them. If anything, this vocabulary is too comprehensive. We could, for example, match *be* to the linguistic component used to link the subject of a sentence with a predicate, and *the* to the grammatical concept of an article. To discard such unhelpful terms, we use the probability that the n-gram would be made a link if it were found in a Wikipedia article; *be* and *the* are mentioned very often in Wikipedia, but rarely used as anchors.

2.4.2 Link Disambiguation

Once candidate terms and phrases have been identified, the anchor vocabulary is again consulted to find out what they mean. This is where we encounter the problem of ambiguity; the anchor statistics tell us that there are many things that could be referred to as *president*. We have developed a machine-learning approach to disambiguation that uses the links found within Wikipedia articles for training. For every link, someone has manually—and probably with some effort—selected the correct destination to represent the intended sense of the anchor. This provides millions of ground-truth examples to learn from.

The two main features of the disambiguation classifier are commonness and relatedness. The system is predisposed towards more common senses (e.g. the *president of the United States* rather than the *president of the Maryland State Senate*), but also considers how strongly each sense relates to the surrounding unambiguous terms, which in this case include *Russian president* and *Vladimir Putin*. A third feature—quality of context—is used to adjust the balance between commonness and relatedness from document to document.

The disambiguation classifier does not actually choose the best sense for each term. Instead it considers each sense independently, and produces a probability that it is valid. If strict disambiguation is required (e.g. when choosing the destination for a link), we select the sense that has the highest probability. If more than one sense may be useful (e.g. when representing the document as a graph), we gather all senses that have a higher probability of being valid than not.

2.4.3 Link Detection

The previous two steps produce a set of associations between terms in the document and the Wikipedia articles that describe them. The final task is to choose which of these topics are

relevant enough to the story to be worth linking to. Wikipedia provides millions of examples of how to do this, since every article has been manually cross-referenced with other articles. We have developed a machine learned topic detector that uses Wikipedia articles for training.

To train the link detector, a Wikipedia article is stripped of markup and automatically processed according the previous two steps. This results in a set of automatically identified Wikipedia articles, which provide training instances for a classifier. Positive examples are the articles that were manually linked to, while negative ones are those that were not. Features of these articles—and the places where they were mentioned—are used to inform the classifier about which topics should and should not be linked.

Some of the features are generated from the Wikipedia topics themselves. One feature is the extent to which a topic relates to other candidates; in Figure 3 the reader is more likely to find *Vladimir Putin* interesting than *news agency*. Another feature is generality, since it is more useful for the reader to provide links for specific topics that they may not know about, rather than general ones that require little explanation.

The previous steps also generate features. The link probability feature—essentially a prior probability of link-likelihood—used to perform the initial candidate selection is again used here. Also, the disambiguation classifier does not just produce a yes/no judgment as to whether a topic is a valid sense of a term. It also gives a probability or confidence in this answer, which is used as a feature to give the senses that we are most sure of a greater chance of being linked.

The remaining features are based on the locations where topics are mentioned: the n-grams from which they were mined. Frequency is an obvious choice, since the more times a topic is mentioned, the more link-worthy it is. First occurrence, last

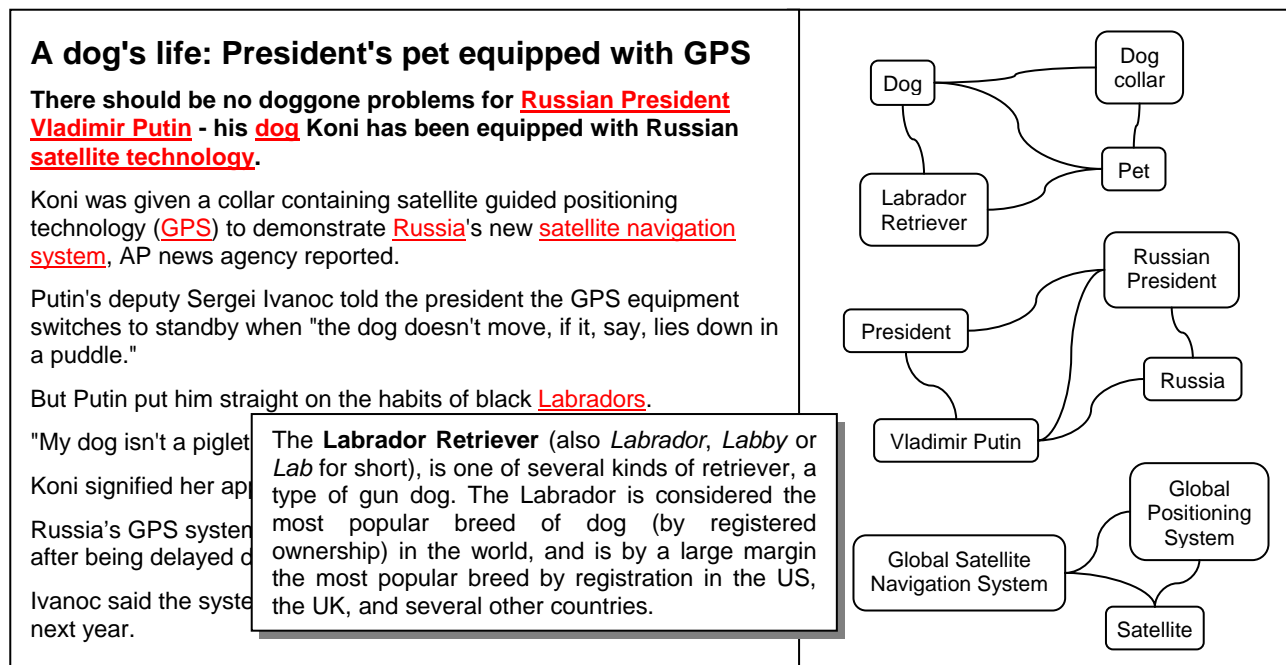


Figure 3: A news article augmented with Wikipedia topics.

occurrence, and the distance between them are also helpful, since important topics tend to be discussed in introductions, conclusions, and consistently throughout documents.

After training on several hundred Wikipedia articles, the topic detector can be run to identify link-worthy topics in other documents. The toolkit provides methods for preparing and tagging both HTML documents and MediaWiki markup, and makes it easy for users to modify how other formats are processed.

3. WRITING APPLICATIONS WITH WIKIPEDIA MINER

In this section we demonstrate how to write a simple thesaurus browser using the Wikipedia Miner toolkit. The application described here, with code and truncated output displayed in Figure 4, searches Wikipedia to locate different senses of the term *Dog*, and elaborates on the most likely one.

The first line of code creates a new instance of Wikipedia and connects it to a pre-prepared MySQL database. Line 2 queries this instance to get a list of articles which represent the different senses of *Dog*. The second argument of this method call is an optional text processor. As described in Section 2.2, the list of senses returned by this call is obtained by investigating all the times the word *dog* is used as a link anchor. The proportion of

links that go to each of the candidate senses is used to sort them, so that the most likely sense—the domestic pet—appears first. Line 6 stores this sense in a new variable and 7 outputs a short, plain-text definition.

Lines 8-10 output the different anchors that refer to the article about *Dogs*, which correspond to synonyms in a thesaurus. 11-14 output the different links to equivalent articles in other language versions of Wikipedia, which correspond to translations. The remaining code mines the links within the *Dog* article for related topics. Not all of these links represent useful semantic relations, so lines 15-19 sort the articles according to their semantic relatedness to *Dog*. Lines 20-22 output the result.

Of course, there are many more features of the toolkit that we could take advantage of. We could use the article’s redirects to gather more synonyms, and navigate the article and category links surrounding it to gather more related topics. These could be classified as broader, narrower, or associated topics by inspecting the hierarchical relationships between them, or clustered using semantic relatedness measures. We could even run the topic detection process described in Section 2.4 over a predefined set of documents, and restrict the thesaurus to the subset of Wikipedia that is relevant to them. However, we have already done a impressive job with just 22 lines of code. With a little more work to provide input and selection facilities, we would have a useful—and very large scale—thesaurus browser.

<pre>1 Wikipedia w = new Wikipedia("dbServer", "dbName") ; 2 SortedVector<Article> ss = w.getWeightedArticles("Dog", null) ; 3 System.out.println("Senses for Dog:") ; 4 for(Article s: ss) 5 System.out.println(" " + s.getTitle()) ; 6 Article dog = ss.first() ; 7 System.out.println(dog.getFirstSentence()) ; 8 System.out.println("Synonyms: ") ; 9 for (AnchorText at:dog.getAnchorTexts()) 10 System.out.println(" " + at.getText()) ; 11 System.out.println("Translations: ") ; 12 HashMap<String, String> ts = dog.getTranslations() ; 13 for (String lang:ts.keySet()) 14 System.out.println(" " + lang + ", " + ts.get(lang)) ; 15 SortedVector<Article> rts = new SortedVector() ; 16 for (Article rt:dog.getLinksOut()) { 17 rt.setWeight(rt.getRelatednessTo(dog)) ; 18 rts.add(rt, false) ; 19 } 20 System.out.println("Related Topics: ") ; 21 for (Article rt: rts) 22 System.out.println(" " + rt.getTitle()) ;</pre>	<p>Senses for Dog:</p> <ul style="list-style-type: none">- Dog- Dog (zodiac)- Hurricane Dog (1950)- Hot dog... <p>The dog (<i>Canis lupis familiaris</i>) is a domestic subspecies of the wolf, a mammal of the Canidae family of the order Carnivora.</p> <p>Synonyms:</p> <ul style="list-style-type: none">- Canis familiaris- Mans best friend- Doggy... <p>Translations:</p> <ul style="list-style-type: none">- Chien (fr)- Haushund (de)- 犬 (zh)... <p>Related Topics:</p> <ul style="list-style-type: none">- Dog Breed- American Kennel Club- Poodle- Pet...
--	---

Figure 4: Java code and truncated output of a simple thesaurus browser.

4. RELATED WORK

This section describes related work on mining Wikipedia. It reviews alternative and complementary resources that the reader may wish to investigate, and summarizes the various research problems that Wikipedia Miner has so far been applied to.

4.1 Alternative and complementary resources

The software system that is most directly related to our work is the Java Wikipedia Library (JWPL)³ produced by Darmstadt University (Zetch *et al.* 2008). The architecture of this toolkit is very similar: it consists of a database containing Wikipedia's summarized content and structure, and a Java API which provides access to it. Most of the API's functionality is also much the same: articles, categories and redirects are represented as objects, which can be searched, browsed, and iterated over efficiently. A recent addition to JWPL allows the content of articles to be parsed for more fine-grained information. This is an advantage over Wikipedia Miner, which currently does little with article content except to make it available as original markup and plain text. Additionally JWPL directly supports several languages, including German, Czech and Ukrainian. While the techniques used by Wikipedia Miner are language-independent in theory, it has only been tested in English—support for other languages is currently in development. However, the anchor statistics, semantic relatedness measures, topic detection and disambiguation features—essentially everything described from Section 2.2 onwards—are unique to Wikipedia Miner.

Another related system is Wikipedia-Similarity,⁴ produced by EML Research (Ponzetto and Strube 2007). This software is more concerned with computing semantic relatedness than with modeling Wikipedia. It communicates with MediaWiki, leaving most of the overhead of serving Wikipedia to the original software that produced it. A library of Perl and Java code provides access to Wikipedia pages, and produces semantic relatedness measures. Unfortunately the libraries are less comprehensive than either of the other toolkits, and the semantic relatedness measures it derives are less accurate than Wikipedia Miner's (Milne and Witten 2008a).

The last piece of software available for mining Wikipedia directly is the WikiPrep PERL script produced by Evgeniy Gabrilovich. This produces much the same data—category hierarchy, anchor statistics, etc—as Wikipedia Miner's scripts do, and additionally feeds this information back to produce an augmented version of the original Wikipedia dump. The use of these files, however, is entirely left to the user.

If one requires a thesaurus or an ontology derived from Wikipedia—rather than direct access to the original structure—then there are several options. DBpedia,⁵ FreeBase⁶ and Yago⁷

are three large-scale ontologies that have been partially or fully derived from Wikipedia. The Wikipedia Lab⁸—a special interest group centered on the University of Tokyo—has produced several resources, including a thesaurus and a bilingual (Japanese–English) dictionary.

4.2 Case studies and applications

The Wikipedia Miner toolkit has already been applied to several research problems. Obviously it has been used to calculate semantic relatedness measures (Milne and Witten 2008a) and for link detection (Milne and Witten 2008b); these are integral components of the toolkit. In this section we describe some of the research that has directly drawn upon this work.

Section 2.1 identified several similarities between the structure of Wikipedia and the relations expressed in thesauri and ontologies. Milne *et al.* (2006) investigate the extent to which these similarities held for Agrovoc, a hand-built thesaurus for the agriculture domain. They found that Wikipedia had much to offer despite its relatively chaotic structure. This work was continued by Medelyan and Milne (2008), who used semantic relatedness measures and anchor destination statistics to map descriptors in Agrovoc to articles in Wikipedia with an average accuracy of 92%. The matched articles augment the thesaurus with extended definitions and new synonyms, translations, and other relations. A similar mapping strategy is used by Medelyan and Legg (2008) to perform large scale alignment between Wikipedia and the common-sense ontology Cyc. Their ultimate aim is to create a resource combining the principled ontological structure of the latter with the messier but vastly more abundant information in the former. An evaluation on 10,000 manually mapped terms provided by the Cyc Foundation, as well as a study with six human subjects, shows that performance of the mapping algorithm compares with the efforts of humans.

In Section 2.3 we identified Wikipedia's potential for improving how documents are modeled and understood. Milne and Witten (2007) used an earlier, less sophisticated method of topic detection to improve how documents are retrieved. Specifically, they use Wikipedia to understand and expand upon queries, and to help guide the user as they evolve their queries interactively. The resulting system was capable of lending assistance to almost every query issued to it; making their entry more efficient and improving the relevance of the documents they return.

Medelyan and Milne (2008) use Wikipedia as a vocabulary for describing the main topics in a document. They employed a predecessor of the work described in Section 2.3 to identify relevant Wikipedia topics, and used machine learning over manually indexed documents to build a model of which topics best describe what a document is about. An analysis of indexing consistency shows that the algorithm performs as well as the average human. Huang *et al.* (2008) employ similar techniques to improve how documents are clustered. They use the same process as above to identify relevant Wikipedia concepts and build concept-based representations of each document. The techniques described in Section 2.2 are then used to build a similarity metric between documents. The resulting clustering algorithm is able to group related documents together, regardless of textual overlap.

³ The Java Wikipedia Library is available at <http://www.ukp.tu-darmstadt.de/software/jwpl/>

⁴ Wikipedia-Similarity can be downloaded from <http://www.eml-research.de/nlp/download/wikipediasimilarity.php>

⁵ <http://www.dbpedia.org>

⁶ <http://www.freebase.com>

⁷ <http://www.mpi-inf.mpg.de/~suchanek/downloads/yago>

⁸ <http://www.wikipedia-lab.org>

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a highly functional toolkit for mining the vast amount of semantic knowledge encoded in Wikipedia. Given the hundreds of papers that have been published in the last few years—Medelyan *et al.* (2008) provide a comprehensive survey—the research community surrounding Wikipedia is clearly in a healthy state. Unfortunately it is also somewhat fragmented and repetitive. We hope that Wikipedia Miner will allow developers and researchers to avoid re-inventing the wheel, and instead invest their time on the novel aspects of their work.

The toolkit already offers valuable, state-of-the-art functionality that is unique among alternative resources. We acknowledge that there are gaps: it makes little use of the textual content of Wikipedia, and none at all of its templates or revision history—both of which have been fruitful in related research. Our aim in releasing this work open source is not to provide a complete polished product, but rather a resource for the research community to collaborate around and continue building together.

Feel free to download the code and apply it to your own research problems. More importantly, please join the project if you feel inclined, and help us to identify and resolve its shortcomings. Wikipedia itself grew, and continues to grow, out of a community of collaborative authors. We—the researchers and developers who benefit from this shared effort—would do well to follow the same model.

6. REFERENCES

- [1] Csomai, A. and Mihalcea, R. (2007) Linking Educational Materials to Encyclopedic Knowledge. In *Frontiers in Artificial Intelligence and Applications 158*, IOS Press.
- [2] Huang, A., Milne, D., Frank, E. and Witten, I.H. (2008) Clustering documents with active learning using Wikipedia. In *Proc. of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, Pisa, Italy.
- [3] Medelyan, O. and Legg, C. (2008) Integrating Cyc and Wikipedia: Folksonomy meets rigorously defined common-sense. In *Proc. of the first AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI'08)*, Chicago, I.L.
- [4] Medelyan, O., Legg, C., Milne, D. and Witten, I. H. (2008) Mining meaning from Wikipedia. Department of Computer Science, University of Waikato Working Paper 07/2008 www.cs.waikato.ac.nz/pubs/wp/2008/uow-cs-wp-2008-11.pdf (Accessed February 2009).
- [5] Medelyan, O. and Milne, D. (2008) Augmenting domain-specific thesauri with knowledge from Wikipedia. In *Proc. of the NZ Computer Science Research Student Conference (NZCSRSC 2008)*, Christchurch, New Zealand.
- [6] Medelyan, O., Witten, I.H., and Milne, D. (2008) Topic Indexing with Wikipedia. In *Proc. of the first AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI'08)*, Chicago, I.L.
- [7] Milne, D., Medelyan, O. and Witten, I. H. (2006). Mining Domain-Specific Thesauri from Wikipedia: A case study. In *Proc. of the International Conference on Web Intelligence (WI 2006)*, Hong Kong.
- [8] Milne, D., Witten, I.H. and Nichols, D.M. (2007). A Knowledge-Based Search Engine Powered by Wikipedia. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM'07)*, Lisbon, Portugal.
- [9] Milne, D. and Witten, I.H. (2008a) An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proc. of the first AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI'08)*, Chicago, I.L.
- [10] Milne, D. and Witten, I.H. (2008b) Learning to link with Wikipedia. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM'08)*, Napa Valley, California.
- [11] Ponzetto, S.P. and Strube, M. (2007) An API for Measuring the Relatedness of Words in Wikipedia. In *Companion Volume of the Proc. of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, Prague, Czech Republic.
- [12] Zesch, T. and Müller, C. and Gurevych, I. (2008) Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proc. of the Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco.