Collective Intelligence in Action

Satnam Alag

SAMPLE CHAPTER

MANNING

Collective Intelligence in Action

by Satnam Alag

Chapter 1

brief contents

Part 1

Gathering data for intelligence

Chapter 1 begins the book with a brief overview of what collective intelligence is and how it manifests itself in your application. Then we move on to focus on how we can gather data from which we can derive intelligence. For this, we look at information both inside the application (chapters 2 through 4) and outside the application (chapters 5 and 6).

Chapter 2 deals with learning from the interactions of users. To get the ball rolling, we look at the architecture for embedding intelligence, and present some of the basic concepts related to collective intelligence (CI). We also cover how we can gather data from various forms of user interaction. We continue with this theme in chapter 3, which deals with tagging. This chapter contains all the information you need to build tagging-related features in your application. In chapter 4, we look at the various forms of content that are typically available in a web application and how to derive collective intelligence from it.

Next, we change our focus to collecting data from outside our application. We first deal with searching the blogosphere in chapter 5. This is followed by chapter 6, which deals with intelligently crawling the web in search of relevant content.

1 Understanding collective intelligence

This chapter covers

- The basics of collective intelligence
- How collective intelligence manifests itself in web applications
- Building user-centric applications using collective intelligence
- The three forms of intelligence: direct, indirect, and derived

Web applications are undergoing a revolution.

In this post-dot-com era, the web is transforming. Newer web applications trust their users, invite them to interact, connect them with others, gain early feedback from them, and then use the collected information to constantly improve the application. Web applications that take this approach develop deeper relationships with their users, provide more value to users who return more often, and ultimately offer more targeted experiences for each user according to her personal need.

Web users are undergoing a transformation.

Users are expressing themselves. This expression may be in the form of sharing their opinions on a product or a service through reviews or comments; through sharing and tagging content; through participation in an online community; or by contributing new content.

This increased user interaction and participation gives rise to data that can be converted into intelligence in your application. The use of collective intelligence to personalize a site for a user, to aid him in searching and making decisions, and to make the application more sticky are cherished goals that web applications try to fulfill.

In his book, *Wisdom of the Crowds,* James Surowiecki, business columnist for *The New Yorker*, asserts that "under the right circumstances, groups are remarkably intelligent, and are often smarter than the smartest people in them." Surowiecki says that if the process is sound, the more people you involve in solving a problem, the better the result will be. A crowd's *collective intelligence* will produce better results than those of a small group of experts if four basic conditions are met. These four basic conditions are that "wise crowds" are effective when they're composed of individuals who have diverse opinions; when the individuals aren't afraid to express their opinions; when there's diversity in the crowd; and when there's a way to aggregate all the information and use it in the decision-making process.

Collective intelligence is about making your application more valuable by tapping into *wise crowds*. More formally, collective intelligence (CI) as used in this book simply and concisely means

To effectively use the information provided by others to improve one's application.

This is a fairly broad definition of collective intelligence—one which uses all types of information, both inside and outside the application, to improve the application for a user. This book introduces you to concepts from the areas of machine learning, information retrieval, and data mining, and demonstrates how you can add intelligence to your application. You'll be exposed to how your application can learn about individual users by correlating their interactions with those of others to offer a highly personalized experience.

This chapter provides an overview of collective intelligence and how it can manifest itself in your application. It begins with a brief introduction to the field of collective intelligence, then goes on to describe the many ways it can be applied to your application, and finally shows how intelligence can be classified.

1.1 What is collective intelligence?

Collective intelligence is an active field of research that predates the web. Scientists from the fields of sociology, mass behavior, and computer science have made important contributions to this field. When a group of individuals collaborate or compete with each other, intelligence or behavior that otherwise didn't exist suddenly emerges; this is commonly known as *collective intelligence*. The actions or influence of a few individuals slowly spread across the community until the actions become the norm for the community. To better understand how this circle of influence spreads, let's look at a couple of examples.

In his book *The Hundredth Monkey*,[1] Ken Keyes recounts an interesting story about how change is propagated in groups. In 1952, on the isolated Japanese island of Koshima, scientists observed a group of monkeys. They offered them sweet potatoes; the monkeys liked the sweet potatoes but found the taste of dirt and sand on the potatoes unpleasant. One day, an 18-month-old monkey named Imo found a solution to the problem by washing the potato in a nearby stream of water. She taught this trick to her mother. Her playmates also learned the trick and taught it to their mothers. Initially, only adults who imitated their children learned the new trick, while the others continued eating the old way. In the autumn of 1958, a number of monkeys were washing their potatoes before eating. The exact number is unknown, but let's say that out of 1,000, there were 99 monkeys who washed their potatoes before eating. Early one sunny morning, a 100th monkey decided to wash his potato. Then, incredibly, by evening *all* monkeys were washing their potatoes. The 100th monkey was that *tipping point* that caused others to change their habits for the better. Soon it was observed that monkeys on other islands were also washing their potatoes before eating them.

As users interact on the web and express their opinions, they influence others. Their initial circle of influence is the group of individuals that they most interact with. Because the web is a highly connected network of sites, this *circle of influence* grows and may shape the thoughts of everybody in the group. This circle of influence also grows rapidly throughout the community—another example helps illustrate this further.

In 1918, as the influenza flu pandemic spread, nearly 14 percent of Fiji's population died in just 16 days. Nearly one third of the native population in Alaska had a similar fate; it's estimated that worldwide, nearly twenty-five million people died of the flu. A pandemic is a global disease outbreak and spreads from person to person. First, one person is affected, who then transmits it to another and then another. The newly infected person transmits the flu to others; this causes the disease to spread exponentially.

In October 2006, Google bought YouTube for $1.65 billion. In its 20 months of existence, YouTube had grown to be one of the busiest sites on the Internet, dishing out 100 million video[2] views a day. It ramped from zero to more than 20 million unique user visits a day, with mainly *viral marketing*—spread from person to person, similar to the way the pandemic flu spreads. In YouTube's case, each time a user uploaded a new video, she was easily able to invite others to view this video. As those others viewed this video, other related videos popped up as recommendations, keeping the user further engaged. Ultimately, many of these viewers also became submitters and uploaded their own videos as well. As the number of videos increased, the site became more and more attractive for new users to visit.

[1] http://en.wikipedia.org/wiki/Hundredth_Monkey

[2] As of September 2006

Whether you're a budding startup, a recognized market leader, or looking to take an emerging application or web site to the next level, harnessing information from users improves the perceived value of the application to both current and prospective users. This improved value will not only encourage current users to interact more, but will also attract new users to the application. The value of the application further improves as new users interact with it and contribute more content. This forms a self-reinforcing feedback loop, commonly known as a *network effect*, which enables wider adoption of the service. Next, let's look at CI as it applies to web applications.

1.2 CI in web applications

In this section, we look at how CI manifests itself in web applications. We walk through an example to illustrate how it can be used in web applications, briefly review its benefits, see how it fits in with Web 2.0 and can be leveraged to build user-centric applications.

Let's expand on our earlier definition of collective intelligence. Collective intelligence of users in essence is

- *The intelligence that's extracted out from the collective set of interactions and contributions made by your users.*
- *The use of this intelligence to act as a filter for what's valuable in your application for a user*—This filter takes into account a user's preferences and interactions to provide relevant information to the user.

This filter could be the simple influence that collective user information has on a user—perhaps a rating or a review written about a product, as shown in figure 1.1—or it may be more involved—building models to recommend personalized content to a user. This book is focused toward building the more involved models to personalize your application.

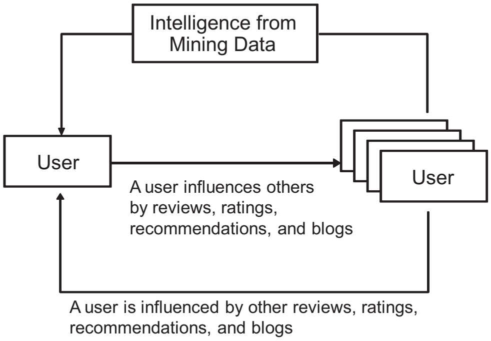

Intelligence from Mining Data
User
User
A user influences others by reviews, ratings, recommendations, and blogs
A user is influenced by other reviews, ratings, recommendations, and blogs

Figure 1.1 A user may be influenced by other users either directly or through intelligence derived from the application by mining the data.

As shown in figure 1.2, there are three things that need to happen to apply collective intelligence in your application. You need to

1. Allow users to interact with your site and with each other, learning about each user through their interactions and contributions.
2. Aggregate what you learn about your users and their contributions using some useful models.
3. Leverage those models to recommend relevant content to a user.

Let's walk through an example to understand how collective intelligence can be a catalyst to building a successful web application.

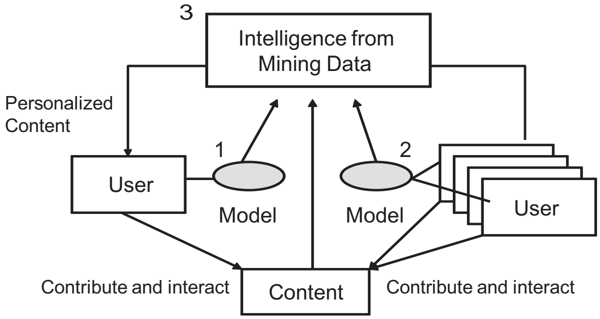


Figure 1.2 Three components to harnessing collective intelligence. 1: Allow users to interact. 2: Learn about your users in aggregate. 3: Personalize content using user interaction data and aggregate data.

1.2.1 Collective intelligence from the ground up: a sample application

In our example, John and Jane are two engineers who gave up their lucrative jobs to start a company. They're based in Silicon Valley and as is the trend nowadays, they're building their fledgling company without any venture capital on a shoestring budget leveraging open source software. They believe in fast-iterative agile-based development cycles and aren't afraid to release beta software to gain early feedback on their features .[3] They're looking to build a marketplace and plan to generate revenue both from selling ad space and from sharing revenue from sold items.

In their first iteration, they launched an application where users—mainly friends and family—could buy items and view relevant articles. There wasn't much in terms of personalization or user interaction or intelligence—a plain vanilla system.

Next, they added the feature of showing a list of top items purchased by users, along with a list of recently purchased items. This is perhaps the simplest form of applying collective intelligence—*providing information in aggregate to users.* To grow the application virally, they also enabled users to email these lists to others. Users used this to forward interesting lists of items to their friends, who in turn became users of the application.

In their next iteration, they wanted to learn more about their users. So they built a basic user profile mechanism that contained explicit and implicit profile information. The explicit information was provided directly by the users as part of their accounts—first name, age, and so on. The implicit information was collected from the user interaction data—this included information such as the articles and content users viewed and the products they purchased. They also wanted to show more relevant articles and content to each user, so they built a content-based *recommendation engine* that analyzed the content of articles—keywords, word frequency, location, and so forth to correlate articles with each other and recommend possibly interesting articles to each user.

Next, they allowed users to generate content. They gave users the ability to write about their experiences with the products, in essence writing reviews and creating their list of recommendations through both explicit ratings of individual products

[3] Note that beta doesn't mean poor quality; it just means that it's incomplete in functionality.

and a "my top 10 favorite products" list. They also gave users the capability to rate items and rate reviews. Ratings and reviews have been shown to influence other users, and numerical rating information is also useful as an input to a collaborative-based recommendation engine.

With the growing list of content and products available on the site, John and Jane now found it too cumbersome and expensive to manually maintain the classification of content on their site. The users also provided feedback that content navigation menus were too rigid. So they introduced dynamic navigation via a *tag cloud*—navigation built by an alphabetical listing of terms, where font size correlates with importance or number of occurrences of a tag. The terms were automatically extracted from the content by analyzing the content. The application analyzed each user's interaction and provided users with a personalized set of tags for navigating the site. The set of tags changed as the type of content visited by the users changed. Further, the content displayed when a user clicked on a tag varied from user to user and changed over time. Some tags pulled the data from a search engine, while others from the recommendation engine and external catalogs.

In the next release, they allowed the users to explicitly tag items by adding free text labels, along with saving or *bookmarking* items of interest. As users started tagging items, John and Jane found that there was a rich set of information that could be derived. First of all, users were providing new terms for the content that made sense to them—in essence they built *folksonomies*.[4] The tag cloud navigation now had both machine-generated and user-generated tags. The process of extracting tags using an automated algorithm could also be enhanced using the dictionary of tags built by the users. These user-added tags were also useful for finding keywords used by an ad-generation engine. They could also use the tags created by users to connect users with each other and with other items of interest. This is collective intelligence in action.

Next, they allowed their users to generate content. Users could now blog about their experiences, or ask and respond to questions on message boards, or participate in building the application itself by contributing to wikis. John and Jane quickly built an algorithm that could extract tags from the unstructured content. They then matched the interests of users—gained from analyzing their interaction in the applications—with those of other users to find relevant items. They were soon able to learn enough about their users to personalize the site for each user, and to provide relevant content—targeting niche items to niche users. They could also target relevant advertisements based on the user profile and context of interaction.

They also modified the search results to make them more relevant to each user, for which they used the user's profile and interaction history when appropriate. They customized advertising by using keywords that were relevant to both the user and the page content.

To make the application stickier, they started aggregating and indexing external content—they would crawl a select list of external web sites to index the content and present

[4] Folksonomies are classifications created through the process of users tagging items.

links to it when relevant. They also connected to sites that tracked the blogosphere, presenting the users with relevant content from what others were saying in blogs.

They also clustered users and items to find patterns in the data and built models to automatically classify content into one of many categories.

The users soon liked the application so much that they started recommending the application to their friends and relatives and the user base grew *virally*. In our example, after a couple of years, John and Jane retired to Hawaii, having sold the company for a gigantic amount, where they waited for the next web revolution… Web 3.0!

These in essence are the many ways by which collective intelligence will manifest itself in your application, and thus more or less the outline for this book. Table 1.1 summarizes the ways to harness collective intelligence in your application. Each of these is discussed throughout the book.

Table 1.1 Some of the ways to harness collective intelligence in your application

Techniques	Description
Aggregate information: lists	Create lists of items generated in the aggregate by your users. Perhaps, *Top List* of items bought, or *Top Search Items or List of Recent Items*.
Ratings, reviews, and recommendations	Collective information from your users influences others.
User-generated content: blogs, wikis, message boards	Intelligence can be extracted from contributions by users. These contributions also influence other users.
Tagging, bookmarking, voting, saving	Collective intelligence of users can be used to bubble up interesting content, learn about your users, and connect users.
Tag cloud navigation	Dynamic classification of content using terms generated via one or more of the following techniques: machine-generated, professionally-generated, or user-generated.
Analyze content to build user profiles	Analyze content associated with a user to extract keywords. This information is used to build user profiles.
Clustering and predictive models	Cluster users and items, build predictive models.
Recommendation engines	Recommend related content or users based on intelligence gathered from user interaction and analyzing content.
Search	Show more pertinent search results using a user's profile.
Harness external content	Provide relevant information from the blogosphere and external sites

John and Jane showed us a few nice things to apply to their site, but there are other benefits of applying collective intelligence to your application. Let's look at that next.

1.2.2 *Benefits of collective intelligence*

Applying collective intelligence to your application impacts it in the following manner:

- *Higher retention rates*—The more users interact with the application, the stickier it gets for them, and the higher the probability that they'll become repeat visitors.
- *Greater opportunities to market to the user*—The greater the number of interactions, the greater the number of pages visited by the user, which increases the opportunities to market to or communicate with the user.
- *Higher probability of a user completing a transaction and finding information of interest*—The more contextually relevant information that a user finds, the better the chances that he'll have the information he needs to complete the transaction or find content of interest. This leads to higher click-through and conversion rates for your advertisements.
- *Boosting search engine rankings*—The more users participate and contribute content, the more content is available in your application and indexed by search engines. This could boost your search engine ranking and make it easier for others to find your application.

Collective intelligence is a term that is increasingly being used in the context of Web 2.0 applications. Let's take a closer look at how it fits in with Web 2.0.

1.2.3 CI is the core component of Web 2.0

Web 2.0 is a term that has generated passionate emotions, ranging from being dismissed as marketing jargon to being anointed as the new or next generation of the Internet. There are seven principles that Web 2.0 companies demonstrate, as shown in table 1.2.[5]

Table 1.2 Seven principles of Web 2.0 applications

Principle	Description
The network is the platform	Companies or users who use traditional licensed software have to deal with running the software, upgrading it periodically to keep up with newer versions, and scaling it to meet appropriate levels of demand. Most successful Web 2.0 companies no longer sell licensed software, but instead deliver their software as a service. The end customer simply uses the service through a browser. All the headaches of running, maintaining, and scaling the software and hardware are taken care of by the service provider seamlessly to the end user. The software is upgraded fairly frequently by the service provider and is available 24 x 7.
Harnessing collective intelligence	The key to the success of Web 2.0 applications is how effectively they can harness the information provided by users. The more personalized your service, the better you can match a user to content of her choice.
Hard-to-replicate data as competitive advantage	Hard-to-replicate, unique, large datasets provide a competitive advantage to a company. Web 2.0 is *data* and *software* combined. One can't replicate Craigslist, eBay, Amazon, Flickr, or Google simply by replicating the software. The underlying data that the software generates from user activity is tremendously valuable. This dataset grows every day, improving the product daily.

[5] Refer to Tim O'Reilly's paper on Web 2.0.

Table 1.2 Seven principles of Web 2.0 applications *(continued)*

Principle	Description
The perpetual beta	Web 2.0 companies release their products early to involve their users and gain important feedback. They iterate often by having short release cycles. They involve the users early in the process. They instrument the application to capture important metrics on how a new feature is being used, how often it's being used, and by whom. If you aren't sure how a particular feature should look and have competing designs, expose a prototype of each to different sets of users and measure the success of each. Involve the customers and let them decide which one they like. By having short development cycles, it's possible to solicit user feedback, incorporate changes early in the product life cycle, and build what the users really want.
Simpler programming models	Simpler development models lead to wider adoption and reuse. Design your application for "hackability" and "remixability" following open standards, using simple programming models and a licensing structure that puts as few restrictions as necessary.
Software above the level of a single device	Applications that operate across multiple devices will be more valuable than those that operate in a single device.
Rich user experience	The success of AJAX has fueled the growing use of rich user interfaces in Web 2.0 applications. Adobe Flash/Flex and Microsoft Silverlight are other alternatives for creating rich UIs.

It is widely regarded that harnessing collective intelligence is the key or core component to Web 2.0 applications. In essence, Web 2.0 is all about inviting users to participate and interact. But what do you do with all the data collected from user participation and interaction? This information is wasted if it can't be converted into intelligence and channeled into improving one's application. That's where collective intelligence and this book come in.

Dion Hinchliffe, in his article, "Five Great Ways to Harness Collective Intelligence," makes an analogy to the apocryphal Einstein quote that compound interest was the most important force in the universe. Similarly, web applications that effectively harness collective intelligence can "benefit" in much the same way—harnessing collective intelligence is about those very same exponential effects.

NOTE *Collective intelligence is the heart of Web 2.0 applications.* It's generally acknowledged that one of the core components of Web 3.0 applications will be the use of artificial intelligence.[6] There's debate as to whether this intelligence will be attained by computers reasoning like humans or by sites leveraging the collective intelligence of humans using techniques such as collaborative filtering. Either way, having the dataset generated from real human interactions will be necessary and useful.

In order to effectively leverage collective intelligence, you need to put the user at the center of your application, in essence building a user-centric application.

[6] http://en.wikipedia.org/wiki/Web_3.0#An_evolutionary_path_to_artificial_intelligence

1.2.4 Harnessing CI to transform from content-centric to user-centric applications

Prior to the user-centric revolution, many applications put little emphasis on the user. These applications, known as *content-centric* applications, focused on the best way to present the content and were generally static from user to user and from day to day. User-centric applications leverage CI to fundamentally change how the user interacts with the web application. User-centric applications make the user the center of the web experience and dynamically reshuffle the content based on what's known about the user and what the user explicitly asks for.

As shown in figure 1.3, user-centric applications are composed of the following four components:

- *Core competency*—The main reason why a user comes to the application.
- *Community*—Connecting users with other users of interest, social networking, finding other users who may provide answers to a user's questions.
- *Leveraging user-generated content*—Incorporating generated content and interactions of users to provide additional content to users.
- *Building a marketplace*—Monetizing the application by product and/or service placements and showing relevant advertisements.

The user profile is at the center of the application. A part of the user profile may be generated by the user, while some parts of it may be learned by the application based on user interaction. Typically, sites that allow user-generated content have an abundance of information. User-centric sites leverage collective intelligence to present relevant content to the user.

Figure 1.4 shows a screenshot of one such user-centric application—LinkedIn,[7] a popular online network of more than 20 million professionals.[8] As shown in

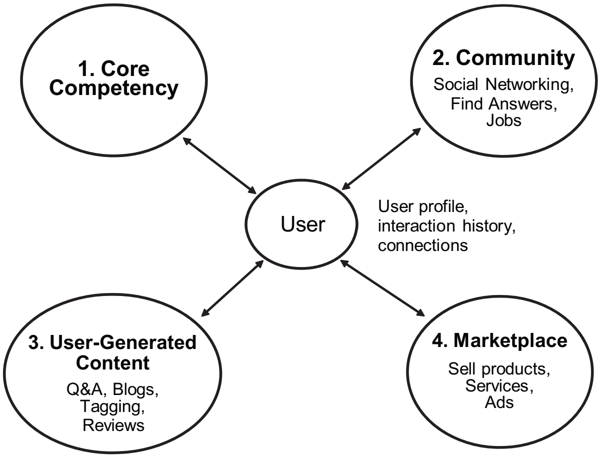
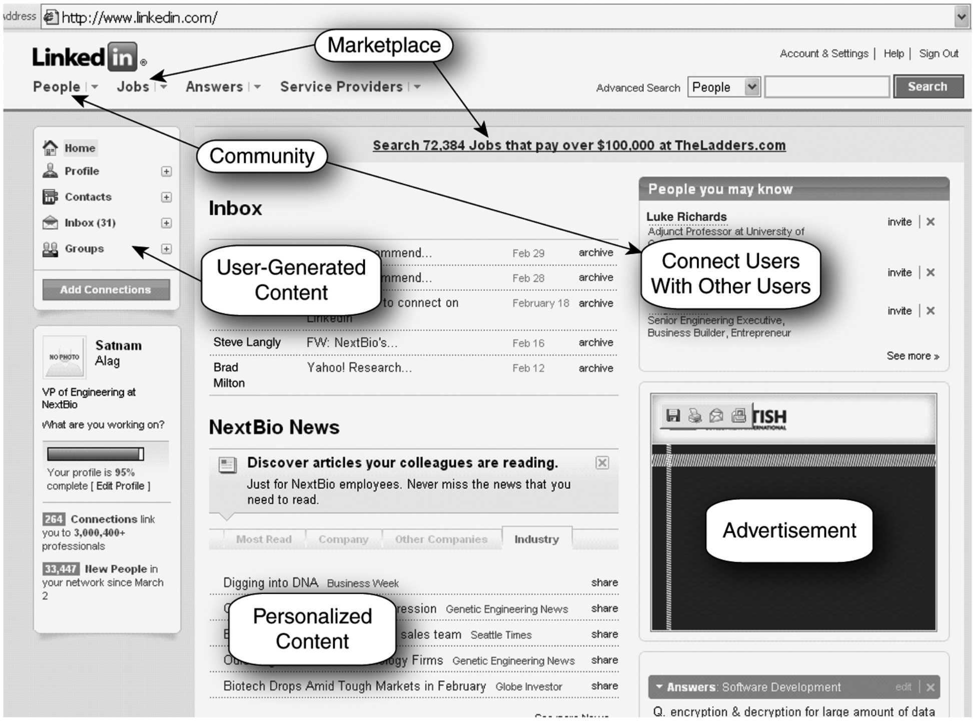

1. Core Competency
2. Community
Social Networking, Find Answers, Jobs
User
User profile, interaction history, connections
3. User-Generated Content
Q&A, Blogs, Tagging, Reviews
4. Marketplace
Sell products, Services, Ads

Figure 1.3 Four pillars for user-centric applications

[7] http://www.linkedin.com/static?key=company_info&trk=ftr_abt

[8] As of May 2008



Search 72,384 Jobs that pay over $100,000 at TheLadders.com

Inbox

	...mmend...	Feb 29	archive
	...mmend...	Feb 28	archive
	...to connect on LinkedIn	February 18	archive
Steve Langly	FW: NextBio's...	Feb 16	archive
Brad Milton	Yahoo! Research...	Feb 12	archive

People you may know

Luke Richards
Adjunct Professor at University of ... invite | ✕

invite | ✕

Senior Engineering Executive, Business Builder, Entrepreneur invite | ✕

Satnam Alag
VP of Engineering at NextBio
What are you working on?

Your profile is 95% complete [Edit Profile]

264 Connections link you to 3,000,400+ professionals

33,447 New People in your network since March 2

NextBio News

Discover articles your colleagues are reading.
Just for NextBio employees. Never miss the news that you need to read.

Digging into DNA Business Week share
...ression Genetic Engineering News share
...sales team Seattle Times share
...ogy Firms Genetic Engineering News share
Biotech Drops Amid Tough Markets in February Globe Investor share

Answers: Software Development
Q. encryption & decryption for large amount of data

Figure 1.4 An example of a user-centric application—LinkedIn (www.linkedin.com)

the screenshot, the LinkedIn application leverages the four components of user-centric applications:

- *Core competency*—Users come to the site to connect with others and build their professional profiles.
- *Community*—Users create connections with other users; connections are used while looking up people, responding to jobs, and answering questions asked by other users. Other users are automatically recommended as possible connections by the application.
- *User-generated content*—Most of the content at the site is user-generated. This includes the actual professional profiles, the questions asked, the feed of actions—such as a user updating his profile, uploading his photograph, or connecting to someone new.
- *Marketplace*—The application is monetized by means of advertisements, job postings, and a monthly subscription for the power-users of the system, who often are recruiters. The monetization model used is also commonly known as *freemium*[9]—basic services are free and are used by most users, while there's a charge for premium services that a small minority of users pay for.

[9] http://en.wikipedia.org/wiki/Freemium_business_model

For user-centric applications to be successful, they need to personalize the site for each user. CI can be beneficial to these applications. So far in this section, we've looked at what collective intelligence is, how it manifests itself in your application, the advantages of applying it, and how it fits in with Web 2.0. Next, we'll take a more detailed look at the many forms of information provided by the users.

1.3 Classifying intelligence

Figure 1.5 illustrates the three types of intelligence that we discuss in this book. First is explicit information that the user provides in the application. Second is implicit information that a user provides either inside or outside the application and is typically in an unstructured format. Lastly, there is intelligence that's derived by analyzing the aggregate data collected. This piece of derived intelligence is shown on the upper half of the triangle, as it is based on the information gathered by the other two parts.

Data comes in two forms: structured data and unstructured data. Structured data has a well-defined form, something that makes it easily stored and queried on. User ratings, content articles viewed, and items purchased are all examples of structured data. Unstructured data is typically in the form of raw text. Reviews, discussion forum posts, blog entries, and chat sessions are all examples of unstructured data.

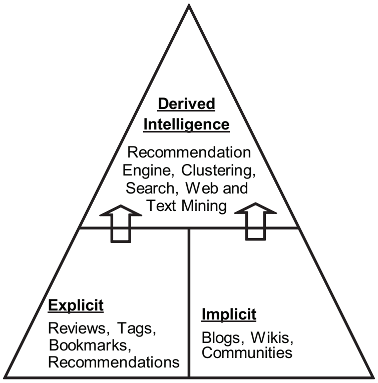

Derived Intelligence
Recommendation Engine, Clustering, Search, Web and Text Mining
Explicit
Reviews, Tags, Bookmarks, Recommendations
Implicit
Blogs, Wikis, Communities

Figure 1.5 Classifying user-generated information

In this section, we look at the three forms of intelligence: explicit, implicit, and derived.

1.3.1 Explicit intelligence

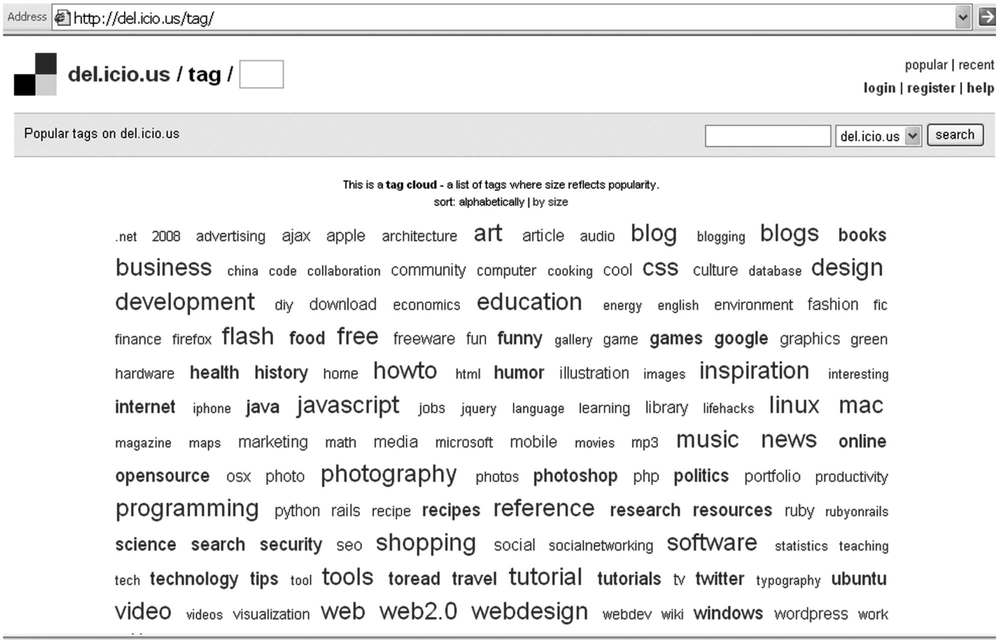
This section deals with explicit information that a user provides. Here are a few examples of how a user provides explicit information that can be leveraged.

REVIEWS AND RECOMMENDATIONS

A recommendation made by a friend or a person of influence can have a big impact on other users within the same group. Moreover, a review or comments about a user's experience with a particular provider or service is contextually relevant for other users inquiring about that topic, especially if it's within the context of similar use.

TAGGING

Adding the ability for users to add tags—keywords or labels provided by a user—to classify items of interest such as articles, items being sold, pictures, videos, podcasts, and so on is a powerful technique to solicit information from the user. Tags can also be generated by professional editors or by an automated algorithm that analyzes content. These tags are used to classify data, bookmark sites, connect people with each other, aid users in their searches, and build dynamic navigation in your application, of which a tag cloud is one example.



Address http://del.icio.us/tag/

del.icio.us / tag /

popular | recent
login | register | help

Popular tags on del.icio.us

del.icio.us search

This is a **tag cloud** - a list of tags where size reflects popularity.
sort: alphabetically | by size

.net 2008 advertising ajax apple architecture art article audio blog blogging blogs books
business china code collaboration community computer cooking cool css culture database design
development diy download economics education energy english environment fashion fic
finance firefox flash food free freeware fun funny gallery game games google graphics green
hardware health history home howto html humor illustration images inspiration interesting
internet iphone java javascript jobs jquery language learning library lifehacks linux mac
magazine maps marketing math media microsoft mobile movies mp3 music news online
opensource osx photo photography photos photoshop php politics portfolio productivity
programming python rails recipe recipes reference research resources ruby rubyonrails
science search security seo shopping social socialnetworking software statistics teaching
tech technology tips tool tools toread travel tutorial tutorials tv twitter typography ubuntu
video videos visualization web web2.0 webdesign webdev wiki windows wordpress work

Figure 1.6 This tag cloud from del.icio.us shows popular tags at the site.

Figure 1.6 shows a tag cloud showing popular tags at del.icio.us, a popular bookmarking site. In a tag cloud, tags are displayed alphabetically, with the size of the font representing the frequency of occurrence. The larger the font of the tag, the more frequently it occurs.

VOTING

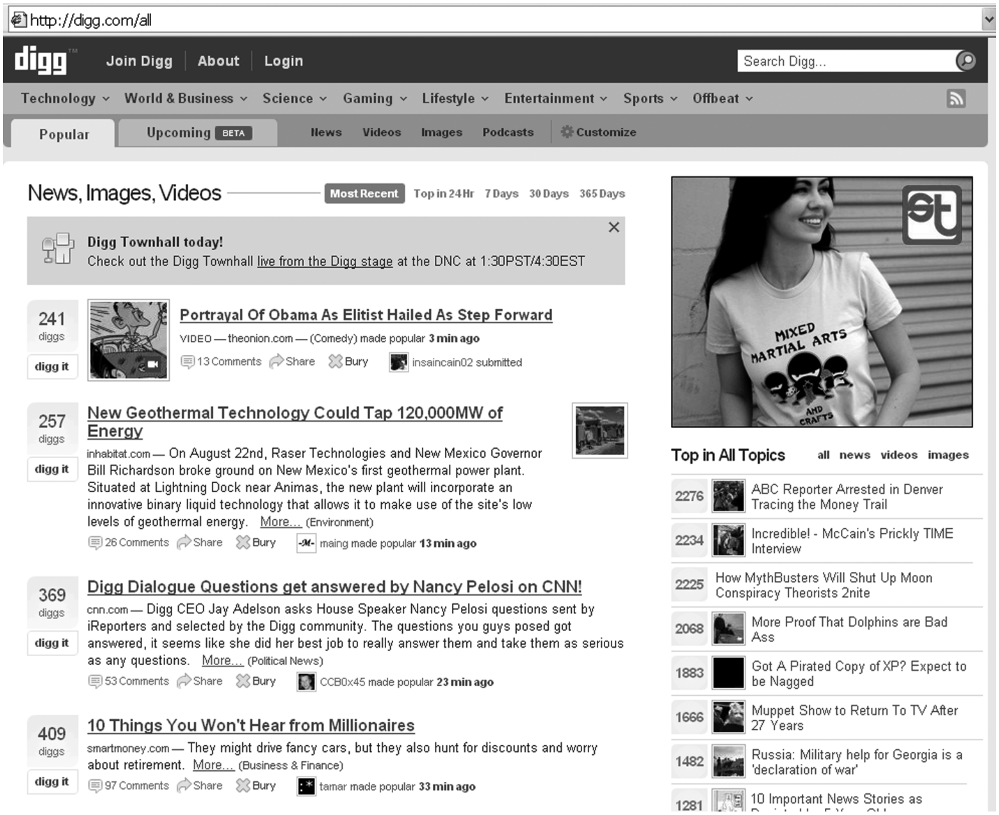
Voting is another way to involve and obtain useful information from the user. Digg, a web site that allows users to contribute and vote on interesting articles, leverages this idea. Every article on Digg is submitted and voted on by the Digg community. Submissions that receive many votes in a short period tend to move up in rank. This is a good way to share, discover, bookmark, and promote important news. Figure 1.7 is a screenshot from Digg.com showing news items with the number of Diggs associated with each.

1.3.2 *Implicit intelligence*

This section deals with indirect information that a user provides. Here are a few examples of how a user provides this information.

Information relevant to your application may appear in an unstructured free-form text format through reviews, messages, blogs, and so forth. A user may express his opinion online, either within your application or outside the application, by writing in his blog or replying to a question in an online community. Thanks to the power of search engines and blog-tracking engines, this information becomes easily available to others and helps to shape their opinions.

You may want to augment your current application by aggregating and mining external data. For example, if your area is real estate applications, you may want to augment your application with additional data harvested from freely available external



Digg Townhall today!
Check out the Digg Townhall live from the Digg stage at the DNC at 1:30PST/4:30EST

241 diggs digg it
Portrayal Of Obama As Elitist Hailed As Step Forward
VIDEO — theonion.com — (Comedy) made popular 3 min ago
13 Comments Share Bury insaincain02 submitted

257 diggs digg it
New Geothermal Technology Could Tap 120,000MW of Energy
inhabitat.com — On August 22nd, Raser Technologies and New Mexico Governor Bill Richardson broke ground on New Mexico's first geothermal power plant. Situated at Lightning Dock near Animas, the new plant will incorporate an innovative binary liquid technology that allows it to make use of the site's low levels of geothermal energy. More... (Environment)
26 Comments Share Bury maing made popular 13 min ago

369 diggs digg it
Digg Dialogue Questions get answered by Nancy Pelosi on CNN!
cnn.com — Digg CEO Jay Adelson asks House Speaker Nancy Pelosi questions sent by iReporters and selected by the Digg community. The questions you guys posed got answered, it seems like she did her best job to really answer them and take them as serious as any questions. More... (Political News)
53 Comments Share Bury CCB0x45 made popular 23 min ago

409 diggs digg it
10 Things You Won't Hear from Millionaires
smartmoney.com — They might drive fancy cars, but they also hunt for discounts and worry about retirement. More... (Business & Finance)
97 Comments Share Bury tamar made popular 33 min ago

2276 ABC Reporter Arrested in Denver Tracing the Money Trail
2234 Incredible! - McCain's Prickly TIME Interview
2225 How MythBusters Will Shut Up Moon Conspiracy Theorists 2nite
2068 More Proof That Dolphins are Bad Ass
1883 Got A Pirated Copy of XP? Expect to be Nagged
1666 Muppet Show to Return To TV After 27 Years
1482 Russia: Military help for Georgia is a 'declaration of war'
1281 10 Important News Stories as

Figure 1.7 Screen shot from Digg.com showing news items with the number of diggs for each

sites, for example, public records on housing sales, reviews of schools and neighborhoods, and so on.

Blogs are online journals where information is displayed in reverse chronological order. The blogosphere—the collection of blogs on the net—is huge and growing fast. As of August 2008, Technorati, a private company that tracks blogs, was tracking 112.8 million blogs. With a new blog being created virtually every second, the blogosphere is an important source of information that can be leveraged in your application. People write blogs on virtually every topic.

Next, let's look at the third category of intelligence, which is derived from analyzing the data collected.

1.3.3 *Derived intelligence*

This section deals with information derived from the data you collect from users. Here are a few examples of techniques and features that deal with derived intelligence.

DATA AND TEXT MINING

The process of finding patterns and trends that would otherwise go undetected in large datasets using automated algorithms is known as *data mining*. When the data is in the form of text, the mining process is commonly known as *text data mining*. Another related field is *information retrieval*, which deals with finding relevant information by analyzing the content of the documents. Web and text mining deal with analyzing unstructured content to find patterns in them. Most applications are content-rich. This content is indexed by search engines and can be used by the recommendation engine to recommend relevant content to a user.

CLUSTERING AND PREDICTIVE ANALYSIS

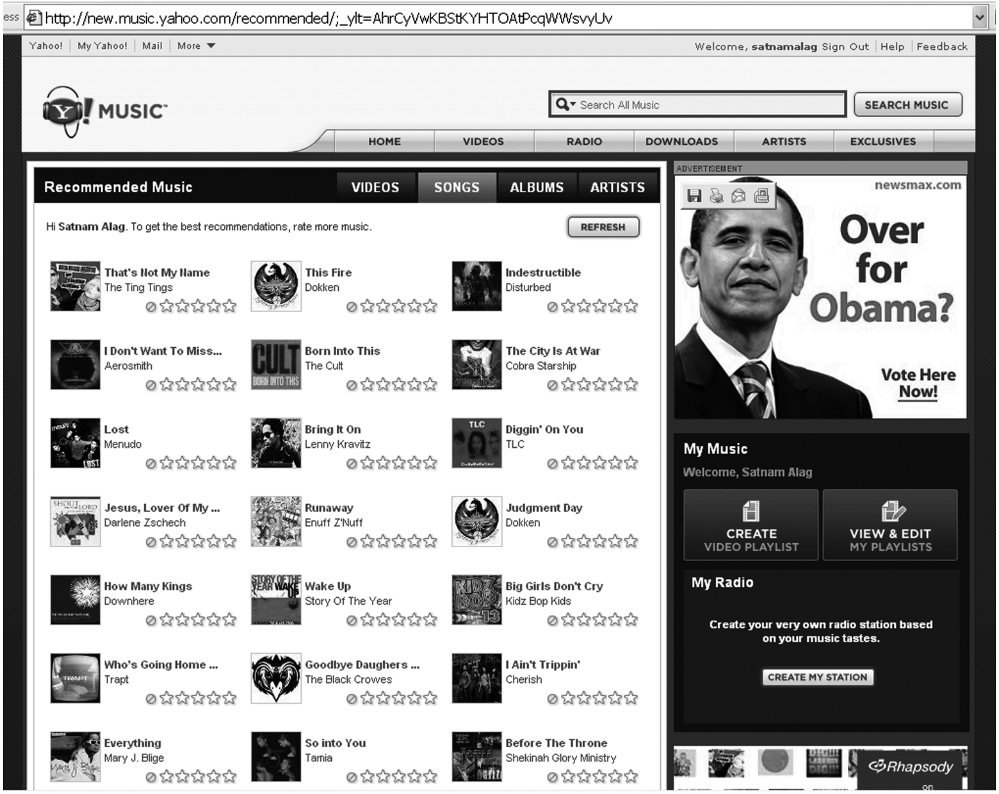
Clustering and predictive analysis are two main components of data mining. Clustering techniques enable you to classify items—users or content—into natural groupings. Predictive analysis is a mathematical model that predicts a value based on the input data.

INTELLIGENT SEARCH

Search is one of the most commonly used techniques for retrieving content. In later chapters, we look at *Lucene*—an open source Java search engine developed through the Apache foundation. We look at how information about the user can be used to customize the search through intelligent filters that enhance search results when appropriate.

RECOMMENDATION ENGINE

A recommendation engine offers relevant content to a user. Again, recommendation engines can be built by analyzing the content, by analyzing user interactions (collaborative approach), or a combination of both. Figure 1.8 shows a screenshot from Yahoo! Music in which a user is recommended music by the application.



http://new.music.yahoo.com/recommended/;_ylt=AhrCyVwKBStKYHTOAtPcqWWsvyUv

Welcome, **satnamalag**

Hi **Satnam Alag**. To get the best recommendations, rate more music.

That's Not My Name The Ting Tings	This Fire Dokken	Indestructible Disturbed
I Don't Want To Miss... Aerosmith	Born Into This The Cult	The City Is At War Cobra Starship
Lost Menudo	Bring It On Lenny Kravitz	Diggin' On You TLC
Jesus, Lover Of My ... Darlene Zschech	Runaway Enuff Z'Nuff	Judgment Day Dokken
How Many Kings Downhere	Wake Up Story Of The Year	Big Girls Don't Cry Kidz Bop Kids
Who's Going Home ... Trapt	Goodbye Daughters ... The Black Crowes	I Ain't Trippin' Cherish
Everything Mary J. Blige	So into You Tamia	Before The Throne Shekinah Glory Ministry

Welcome, Satnam Alag

Create your very own radio station based on your music tastes.

Figure 1.8 Screenshot from Yahoo! Music recommending songs of interest

Recommendation engines use inputs from the user to offer a list of recommended items. The inputs to the recommendation engine may be items in the user's shopping list, items she's purchased in the past or is considering purchasing, user-profile information such as age, tags and articles that the user has looked at or contributed, or any other useful information that the user may have provided. For large online stores such as Amazon, which has millions of items in its catalog, providing fast recommendations can be challenging. Recommendation engines need to be fast and scale independently of the number of items in the catalog and the number of users in the system; they need to offer good recommendations for new customers with limited interaction history; and they need to age out older or irrelevant interaction data (such as a gift bought for someone else) from the recommendation process.

1.4 Summary

Collective intelligence is powering a new breed of applications that invite users to interact, contribute content, connect with other users, and personalize the site experience.

Users influence other users. This influence spreads outward from their immediate circle of influence until it reaches a critical number, after which it becomes the norm. Useful user-generated content and opinions spread virally with minimal marketing.

Intelligence provided by users can be divided into three main categories. First is direct information/intelligence provided by the user. Reviews, recommendations, ratings, voting, tags, bookmarks, user interaction, and user-generated content are all examples of techniques to gather this intelligence. Next is indirect information provided by the user either on or off the application, which is typically in unstructured text. Blog entries, contributions to online communities, and wikis are all sources of intelligence for the application. Third is a higher level of intelligence that's derived using data mining techniques. Recommendation engines, use of predictive analysis for personalization, profile building, market segmentation, and web and text mining are all examples of discovering and applying this higher level of intelligence.

The rest of this book is divided into three parts. The first part deals with collecting data for analysis, the second part deals with developing algorithms for analyzing the data, and the last part deals with applying the algorithms to your application. Next, in chapter 2, we look at how intelligence can be gathered by analyzing user interactions.

1.5 Resources

"All things Web 2.0." http://www.allthingsweb2.com/component/option,com_mtree/Itemid,26/

Anderson, Chris. *The Long Tail: Why the Future of Business Is Selling Less of More.* 2006. Hyperion

Hinchliffe, Dion. "The Web 2.0 Is Here." http://web2.wsj2.com/web2ishere.htm

"Five Great Ways to Harness Collective Intelligence." January 17, 2006, http://web2.wsj2.com/five_great_ways_to_harness_collective_intelligence.htm

"Architectures of Participation: The Next Big Thing." August 1, 2006, http://web2.wsj2.com/architectures_of_participation_the_next_big_thing.htm

Jaokar, Ajit. “Tim O’Reilly’s seven principles of web 2.0 make a lot more sense if you change the order.” April 17, 2006, http://opengardensblog.futuretext.com/archives/2006/04/tim_o_reillys_s.html

Kroski, Ellyssa. “The Hype and the Hullabaloo of Web 2.0.” http://infotangle.blogsome.com/2006/01/13/the-hype-and-the-hullabaloo-of-web-20/

McGovern, Gerry. “Collective intelligence: is your website tapping it?” April 2006, New Thinking, http://www.gerrymcgovern.com/nt/2006/nt-2006-04-17-collective-intelligence.htm

“One blog created ‘every second’.” BBC news, http://news.bbc.co.uk/1/hi/technology/4737671.stm

“Online Community Toolkit.” http://www.fullcirc.com/community/communitymanual.htm

O’Reilly, Tim. “What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software.” http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

“The Future of Technology and Proprietary Software.” December 2003, http://tim.oreilly.com/articles/future_2003.html

“Web 2.0: Compact Definition?” October 2005, http://radar.oreilly.com/archives/2005/10/web_20_compact_definition.html

Por, George. “The meaning and accelerating the emergence of CI.” April 2004, http://www.community-intelligence.com/blogs/public/archives/000251.html

Surowiecki, James. *The Wisdom of Crowds.* 2005. Anchor

Web 3.0. Wikipedia, http://en.wikipedia.org/wiki/Web_3.0#An_evolutionary_path_to_artificial_intelligence

JAVA/WEB

Collective Intelligence IN ACTION

Satnam Alag

Free ebook
SEE INSERT

> “Use these CI techniques to extract valuable data from your applications.”
>
> —FROM THE FOREWORD BY
> Richard MacManus, ReadWriteWeb

On the internet, harnessing the collective wisdom of users can be a key to success. A new category of programming techniques lets you discover the valuable patterns, inter-relationships, and individual profiles—the collective intelligence—locked in the data people leave behind as they traverse and interact on the web.

Collective Intelligence in Action starts with the principles of the subject and ideas for building more interactive sites. It then follows a running example to develop an immediately useful Java-based CI toolkit. You'll learn to mine both your own site and the wider web to uncover trends and make practical predictions and recommendations. Along the way, you'll work with numerous helpful APIs and open source toolkits that substantially reduce development effort. This book is written for Java web developers.

What's Inside

- Reusable code for intelligent
  - search
  - recommendations
  - predictions
- Web crawling and text analysis using Lucene and Nutch
- Machine learning using WEKA
- How to implement the Java Data Mining (JDM) standard

About the Author

Satnam Alag is currently the vice president of engineering at NextBio. He was formerly the Chief Software Architect at Rearden Commerce and has a PhD from UC Berkeley.

> “It's technical, it's theoretical—but most importantly, it's practical.”
>
> —Taran Rampersand
> KnowProse.com

> “Harness the untapped power of your imagination.”
>
> —John Tyler
> UBS Investment Bank

> “Learn practical, hands-on, machine learning.”
>
> —Robi Sen, Twin Technologies

> “This is the right book on collective intelligence. I wish I'd had it a few years ago.”
>
> —Jérôme Bernard
> Elastic Grid LLC

> “I recommend this book for any developer of social networking sites.”
>
> —Sopan Shewale
> TWIKI.NET-Enterprise WIKI

Online access to the author, code samples, and (for owners of this book) a free ebook at www.manning.com/alag or www.manning.com/CollectiveIntelligenceinAction

ISBN-13: 978-1933988313
ISBN-10: 1933988312
54499

9 781933 988313

MANNING $44.99 / Can $44.99 [INCLUDES eBOOK]