Unit 1

Introduction

Introduction

- Main interest is on building web applications that do take into account the input and behavior of every user in the system, over time, as well as any other potentially useful information that may be available.
- We focus on developing web applications that have a learning capacity.
- **Examples:**
  - Consider a web application to order food, and every Wednesday you order fish. You'd have a much better experience if, on Wednesdays, the application asked you "Would you like fish today?" instead of "What would you like to order today?". Asking a question that's based on the user's prior selections introduces a new kind of interactivity between the website and its users.
  - Web search engines such as Google.

Introduction

- **Building recommendations:** Interaction of an intelligent web application with a user may adjust due to the input of other users that are somehow related to each other. For example, If your dietary habits match closely those of Mr. John, the application may recommend a few menu selections that are common for John but that you never tried.
- **Social networking site:** A social networking site could offer a fact-checking chat room or electronic forum. By fact checking, we mean that as you type your message, there’s a background check on what you write to ensure that your statements are factually accurate and even consistent with your previous messages. This functionality is similar to spelling and grammar checking, rather it checks a set of facts that could be general truths, your own beliefs about a particular subject, or simple personal facts. Websites with such functional behavior are inference capable.

Example of Intelligent Web Applications

- A turning point in the history of the intelligent web was the advent of search engines. **Google Inc**. has grown, in less than 10 years, from a startup to a dominant player in the technology sector due primarily to the success of its link-based search and secondarily to a number of other services such as Google News and Google Finance.
- The online retailer **Amazon** was one of the first online stores that offered recommendations to its users based on their shopping patterns. Let’s say that you purchase a book on JavaServer Faces and a book on Python. As soon as you add your items to the shopping cart, Amazon will recommend additional items that are somehow related to the ones you’ve just selected. In addition, during your next visit to the Amazon website, the same or other related items may be recommended.

Example of Intelligent Web Applications

- Another intelligent web application is **Netflix**, which is the world's largest online movie rental service. It has ability to provide users with an easy way to choose movies, from an expansive selection of movie titles. At the core of that ability is a recommendation system called Cinematch. Its job is to predict whether someone will enjoy a movie based on how much she liked or disliked other movies. Cinematch analyzes each customer's movie-viewing habits and recommends other movies that the customer might enjoy.
- The company **PredictWallStreet** collects the predictions of its online users for a particular stock or index in order to spot trends in the opinions of the traders and predict the value of the underlying asset through the aggregation of predictions from online users.

Basic Elements of Intelligent Applications

- Applications such as Wikipedia or other public portals are different from applications such as Google search, Google Ads, Netflix Cinematch, and so on. Applications of the first kind are **collaborative platforms** that facilitate the aggregation and maintenance of collective knowledge whereas applications of the second kind are **intelligent applications** and they generate abstractions of patterns from a body of collective knowledge.
- Elements that are required in order to build an intelligent web application are:
  - **Aggregated content:** Aggregated content is a large amount of data pertinent to a specific application. The aggregated content is dynamic rather than static, and its origins as well as its storage locations could be geographically dispersed. Each piece of information is typically associated with, or linked to, many other pieces of information.

Basic Elements of Intelligent Applications

- **Reference structures:** These structures provide one or more structural and semantic interpretations of the content. For example, this is related to what people call folksonomy – the use of tags for annotating content in a dynamic way and continuously updating the representation of the collective knowledge to the users. Reference structures about the world or a specific domain of knowledge come in three big flavors: dictionaries, knowledge bases, and ontologies.
- **Algorithms:** This refers to a layer of modules that allows the application to harness the information, which is hidden in the data, and use it for the purpose of abstraction (generalization), prediction, and (eventually) improved interaction with its users. The algorithms are applied on the aggregated content, and sometimes require the presence of reference structures.

Basic Elements of Intelligent Applications

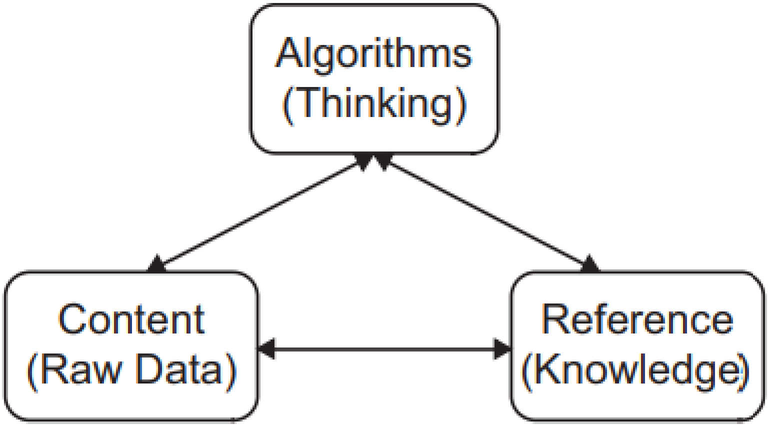

Fig: The triangle of intelligence: the three essential ingredients of intelligence applications

- It’s prudent to keep these three components separate and build a model of their interaction that best fits your needs.

What applications can benefit form intelligence?

- The ingredients of intelligence can be found across a wide spectrum of applications. Some common applications are given below.
- **Social networking sites:**
  - Social networking sites contain all three ingredients of intelligence.
  - These sites aggregate contents, so the first ingredient for building intelligence is readily available. The second ingredient is also present in those sites (for example, content categorization). Finally, most social networking sites are able to recommend to their users new friends and new postings that may be of interest. In order to do that, they rely on advanced algorithms for making predictions and abstractions of the collected data.

What applications can benefit form intelligence?

- **Mashups:**
  - Mashups are interactive web applications that draw upon content retrieved from external data sources to create entirely new and innovative services.
  - Mashups aggregate content form external sources from different locations to create unique presentation based on your interaction with the application.
  - All mashups are not intelligent. In order to build intelligent mashups, we need the ability to reconcile differences or identify similarities of the content that we try to collage. In turn, the reconciliation and classification of the content require one or more reference structures.
  - A number of algorithms can identify what elements of the reference structures are contained within the various pieces or how content has been retrieved from different sites should be categorized for viewing purposes.

What applications can benefit form intelligence?

- **Portals:**
  - Portals and in particular news portals are gateways to content that’s distributed throughout the internet or, in the case of a corporate network, throughout an intranet.
  - The best example in this category is Google News. This site gathers news stories from thousands of sources and automatically groups similar news stories under a common heading that are available by default, such as Business, Health, World, Sci/Tech, and so on. You can even define your own categories and determine what kind of stories are of interest to you.
  - The underlying theme is aggregated content coupled with a reference structure and a number of algorithms that can perform the required tasks automatically or, at least, semi-automatically.

What applications can benefit form intelligence?

- **Wikis:**
  - A wiki is a repository of knowledge that’s accessible online. For example, Wikipedia.
  - These sites are clearly content aggregators. A lot of these sites, due to the page creation workflow, have a built-in structure that annotates the content.
  - Different algorithms can be used for advanced search, clustering, and other analytical techniques.
- **Media-sharing sites:**
  - The unique feature of media-sharing sites (such as YouTube) is that most of their content is in binary format – video or audio files.
  - These sites have aggregated content. The content is typically categorized and algorithms can help us extract value from that content.

What applications can benefit form intelligence?

- **Online gaming:**
  - Massive multiplayer online games have all the ingredients required to create intelligence in the game.
  - They have ample aggregated content and reference structures that reflect the rules, and they can certainly use the algorithms to introduce new levels of sophistication in the game.
  - Characters that are played by the computer can assimilate the input of the human players so that the experience of the game as perceived by the humans becomes more entertaining.

How can I build intelligence in my own application?

- **Two prerequisites of building an intelligent application:**
  - **Review of your functionality** – What are your users doing with your application? How does your application add consumer or business value? The importance of these questions will vary depending on what your application does. Nevertheless, these specific questions should help you identify the areas where an intelligent component would add most value to your application.
  - **Data –** For every application, data is either internal to an application or external. First examine your internal data. You might also want or need to obtain more data from external sources.
- **Get more data from the web:**
  - In many cases, your own data will be sufficient for building intelligence that's relevant and valuable to your application.

How can I build intelligence in my own application?

- But in some cases, providing intelligence in your application may require access to external information. Some enabling technologies for obtaining external data from the Web are given below.
- **Crawling and Screen Scraping:**
  - Crawlers (also known as **spiders**) can roam the internet and download content that's publicly available. Typically, a crawler would visit a list of URLs and attempt to follow the links at each destination. Once the crawler has visited a page, it stores its content locally for further processing.
  - Screen scraping refers to extracting the information that's contained in HTML pages. Screen scraping itself can benefit from the intelligent techniques. For example, restaurant search engine can employ intelligent techniques during screen scraping and help to automatically categorize the reviews (which are plain text in natural language) and ratings from people who ate there and access the ranking of the restaurant.

How can I build intelligence in my own application?

- **Website syndication:**
  - Website syndication is another way to obtain external data and it eliminates the burden of revisiting websites with your crawler.
  - Website syndication is a form of syndication in which content is made available from one website to other sites. Most commonly, websites are made available to provide either summaries or full renditions of a website's recently added content.
  - There are three common feed formats: RSS (RDF (Resource Description Framework) Site Summary) 1.0, RSS (Really Simple Syndication) 2.0, and Atom.
- **Web services:**
  - A web service is a set of open protocols and standards that allow data to be exchanged between different applications or systems.

How can I build intelligence in my own application?

- Web services can be used by software programs written in a variety of programming languages and running on a variety of platforms to exchange data via computer networks.
- Any software, application, or cloud technology that uses standardized web protocols (such as HTTP) to connect, interoperate, and exchange data messages – commonly XML (Extensible Markup Language) – across the internet is considered a web service.
- Web services allow programs developed in different languages to connect with one another by exchanging data over a web service between clients and servers.
- A client invokes a web service by submitting an XML request, which the service responds with an XML response**.**
- **RESTful** Web Services are basically REST (REpresentational State Transfer) Architecture based Web Services. In REST Architecture everything is a resource and a resource is accessed by a common interface using HTTP standard methods.

Machine learning, data mining, and all that

- What exactly Intelligence mean?
  - Artificial intelligence
  - Machine learning
  - Data mining
  - Soft computing (set of algorithms including neural networks, fuzzy logic, genetic algorithms etc.)

Eight fallacies of intelligent applications

- **Fallacy #1: Your data is reliable**
  - You should always examine whether the data that you'll work with can be trusted before you start considering specific intelligent algorithm.
  - **List of the things that can go wrong with your data:**
    - The data that you have available during development may not be representative of the data that corresponds to a production environment. For example, you may want to categorize the users as "tall," "average," and "short" based on their height. If the shortest person in your development data is six feet tall, you're running the risk of calling someone short because they're "just" six feet tall.
    - Your data may contain missing values. Typically, you either leave the missing values as missing or you fill them in with some default or calculated value. Both conditions can lead to unstable implementations.

Eight fallacies of intelligent applications

  - Your data may change. The database schema may change or the semantics of the data in the database may change.
  - Your data may not be normalized. For example, measurement of weight of people should be the same for all individuals – in pounds or kilograms, not a mix of measurements in pounds and kilograms.
  - Your data may be inappropriate for the algorithmic approach that you have in mind.

- **Fallacy #2: Inference happens instantaneously**
  - Computing a solution takes time, and the responsiveness of your application may be crucial. You shouldn't assume that all algorithms, on all datasets, will run within the response time limits of your application. You should test the performance of your algorithm within the range of your operating characteristics.

Eight fallacies of intelligent applications

- **Fallacy #3: The size of data doesn’t matter**
  - The more data you have, the more intelligent your application can be.
- **Fallacy #4: Scalability of the solution isn’t an issue**
  - Don’t assume that your solution is scalable. Some algorithms are scalable and others aren’t. For example, not all clustering algorithms can run in parallel to find groups of similar headline news among billions of titles.
- **Fallacy #5: Apply the same good library everywhere**
  - When you’re holding a hammer, everything looks like a nail.
  - Make sure that you test thoroughly your favorite solution in new areas of application. In addition, it’s recommended that you examine every problem with a fresh perspective; a different problem may be solved more efficiently or more expediently by a different algorithm.

Eight fallacies of intelligent applications

- **Fallacy #6: The computation time is known**
  - Certain applications have a large variance in solution times for a relatively small variation of the parameters involved.
  - Typically, people expect that, when we change the parameters of a problem, the problem can be solved consistently with respect to response time.
  - A seemingly innocuous change in the data can lead to significantly different solution times.
- **Fallacy #7: Complicated models are better**
  - Always start with the simplest model. Then gradually try to improve your results by combining additional elements of intelligence in your solution. KISS (Keep It Simple Stupid)
- **Fallacy #8: There are models without bias**
  - The choice of the models that you make and the data that you use to train your learning algorithms introduce a bias.