NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

POSTGRADUATE STUDIES PROGRAM

DIPLOMA THESIS

Mining Experts in Wikipedia

Ilias K. Panagiotopoulos

Supervisor: **Dimitrios Gunopulos,** Associate Professor NKUA

ATHENS

SEPTEMBER 2012

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εξόρυξη Ειδικών στη Wikipedia

Ηλίας Κ. Παναγιωτόπουλος

Επιβλέπων: **Δημήτριος Γουνόπουλος,** Αναπληρωτής Καθηγητής ΕΚΠΑ

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2012

DIPLOMA THESIS

Mining Experts in Wikipedia

Ilias K. Panagiotopoulos
R.N.: M1108

SUPERVISOR: **Dimitrios Gunopulos,** Associate Professor NKUA

EXAMINATION COMMITTEE:
Dimitrios Gunopulos, Associate Professor NKUA
Manolis Koubarakis, Professor NKUA

September 2012

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εξόρυξη Ειδικών στη Wikipedia

Ηλίας Κ. Παναγιωτόπουλος
Α.Μ.: Μ1108

ΕΠΙΒΛΕΠΩΝ: **Δημήτριος Γουνόπουλος,** Αναπληρωτής Καθηγητής ΕΚΠΑ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:
Δημήτριος Γουνόπουλος, Αναπληρωτής Καθηγητής ΕΚΠΑ
Εμμανουήλ Κουμπαράκης, Καθηγητής ΕΚΠΑ

Σεπτέμβριος 2012

ABSTRACT

The rapid growth of web 2.0 applications is evident during the last years. A part of this category is Wikipedia, an online encyclopedia which covers a wide range of topics, like traditional encyclopedias. The main innovation of Wikipedia is that its content may change dynamically by users who edit its articles. However, this possibility gives us insight into assuming that the entire content cannot be considered fully reliable. Consequently, it is useful to segregate Wikipedia users in terms of expertise into a topic. The goal of this thesis is the development of a flexible expert extraction system, which is based on a graph that models the interactions between users. A set of attributes is extracted from the graph, which contributes to the construction of a metric that ranks users in terms of their expertise. In the end, an experimental evaluation of the system is presented.

SUBJECT AREA: Data Mining

KEYWORDS: Wikipedia, Expert Finding, Social Network Analysis,
Collaborative Knowledge Systems, Controversies

ΠΕΡΙΛΗΨΗ

Η ραγδαία ανάπτυξη των web 2.0 εφαρμογών είναι εμφανής τα τελευταία χρόνια. Στην κατηγορία αυτή ανήκει και η Wikipedia, μία online εγκυκλοπαίδεια που καλύπτει ένα μεγάλο εύρος θεμάτων, όπως οι παραδοσιακές εγκυκλοπαίδειες. Η βασική καινοτομία της Wikipedia είναι ότι το περιεχόμενό της μπορεί να αλλάζει δυναμικά από τους χρήστες που συντάσσουν τα άρθρα της. Ωστόσο, η δυνατότητα αυτή μας προτρέπει να υποθέσουμε ότι δεν μπορεί να θεωρηθεί πλήρως αξιόπιστο ολόκληρο το περιεχόμενο της. Συνεπώς, είναι χρήσιμο να διαχωρίσουμε τους χρήστες της Wikipedia ως την εμπειρογνωμοσύνη σε κάποιο θέμα. Σκοπός της παρούσας εργασίας είναι η ανάπτυξη ενός ευέλικτου συστήματος εξαγωγής ειδικών, το οποίο βασίζεται σε ένα γράφο που μοντελοποιεί τις αλληλεπιδράσεις μεταξύ των χρηστών. Από το γράφο αυτό εξάγεται ένα πλήθος γνωρισμάτων, το οποίο συμβάλλει στην κατασκευή μίας μετρικής, σύμφωνα με την οποία οι χρήστες κατατάσσονται ως προς την εμπειρογνωμοσύνη τους. Στο τέλος παρουσιάζεται η πειραματική αξιολόγηση του συστήματος.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Εξόρυξη Δεδομένων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Βικιπαιδεία, Εύρεση Ειδικών, Ανάλυση Κοινωνικών Δικτύων, Συνεργατικά Συστήματα Γνώσης, Αντιπαραθέσεις

ACKNOWLEDGEMENTS

Firstly, I would like to thank my advisor, Dr. Dimitrios Gunopulos, for providing me with such a wonderful and rewarding research experience. Without his support, patience, and constructive comments, this thesis would not be a reality. I would also like to thank my family for everything they have done to support me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION TO WIKIPEDIA

1.1. Overview of Wikipedia

Wikipedia [1] is a free, collaborative and multilingual Internet encyclopedia, founded in January 2001 by Jimmy Wales and Larry Sanger, and supported by the non-profit Wikimedia Foundation. It mainly consists of articles that have been authored collaboratively by volunteers around the world. The main innovation of Wikipedia, which has been considered both a source of strength and weakness, is that the vast majority of articles can be edited by anyone with access to the site. This possibility has led to the existence of about 100,000 regular active contributors. As of January 2012, Wikipedia consists of editions in 283 languages and has become the largest and more popular reference work on the Internet. Its name was coined by Sanger, who combined words *wiki* (a technology for collaborative website creation) and *encyclopedia*.

Originally, Wikipedia articles were authored by volunteers with expertise to the respective topics, but later the presence of large body of un-academic content has led to Wikipedia's rapid growth, to the likes of other prominent websites (YouTube, MySpace, Facebook). Wikipedia has also been praised as a news source due to the fast update of articles concerning recent events.

An important matter of Wikipedia is the verifiability and neutral point of view in its articles. Many critics accuse it of systematic bias and inconsistencies due to many factors, such as preference to the popular culture. Another remarkable issue is the existence of vandalism phenomena in popular articles, although the opposite view considers these phenomena short-term. A research [2], held by scientific journal *Nature*, performed a comparison between science articles of *Wikipedia* and *Encyclopedia Britannica*, to show that they shared a similar rate of serious errors. Therefore, one thing is for sure; Wikipedia has become a major point of interest over the last few years.

1.2. History of Wikipedia

Initially, Wikipedia started as a complementary project of *Nupedia* [3], a free online English-language encyclopedia project, whose articles were authored and reviewed according to a formal process. Nupedia project was run on March 9, 2000, under the ownership of Bomis, Inc, a web portal company. Company's main figures were Jimmy Wales, CEO, and Larry Sanger, editor-in-chief. Nupedia was originally licensed under its own Open Content License [4], but later it switched to the GNU Free Documentation License [5]. Wales is credited with defining the goal of making a publicly editable encyclopedia, while Sanger considered the strategy of wiki use to satisfy this goal. Wikipedia was formally launched on January 15, 2001, as a single English-language edition at *www.wikipedia.com*, and announced by Sanger on the Nupedia mailing list. Its policy of neutral point-of-view was codified in its initial months and was similar to Nupedia's non-biased policy.

The first contributors of Wikipedia came from Nupedia, while it supported a web search engine indexing. By the end of 2001, it grew to approximately 20,000 articles and 18 language editions. By late 2002, it had been extended to 26 language editions, 46 by the end of 2003 and 161 by the final days of 2004. In the first two years, Nupedia and Wikipedia coexisted until the former's servers were taken down permanently in 2003, and its text was incorporated into Wikipedia. On September 9, 2007 the English version of Wikipedia passed the two-million article mark, becoming the largest encyclopedia ever assembled. In August 2009 it reached three-million articles, although the growth of the edition, in terms of the numbers of articles and of contributors, peaked around early 2007. In figure 1 and figure 2 we present the article number and its growth in the English Wikipedia, respectively.

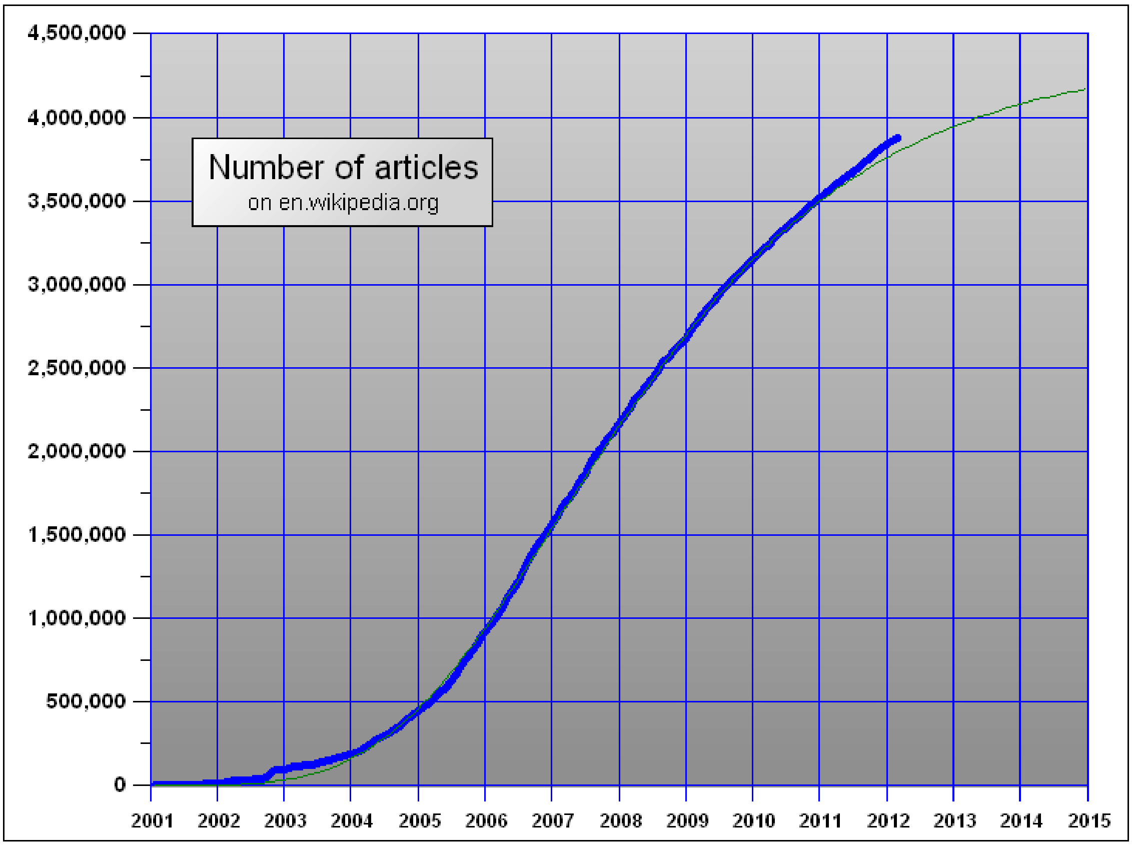


Figure 1: Number of articles in the English Wikipedia (in blue)

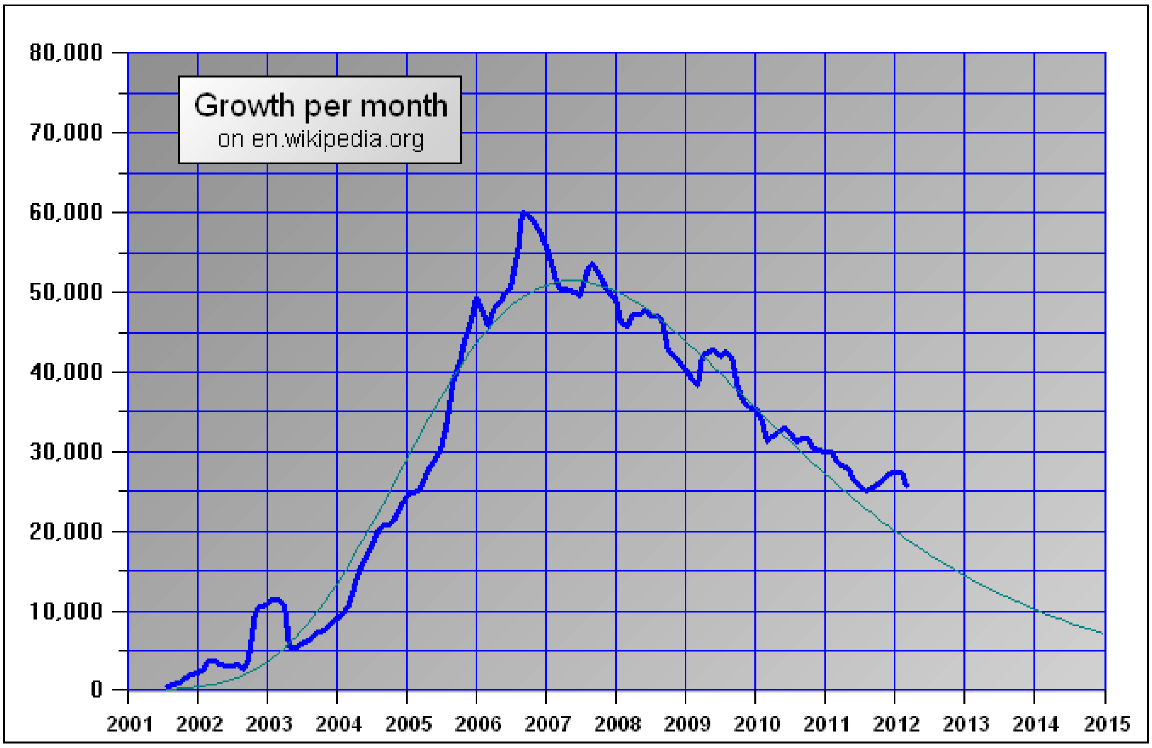


Figure 2: Growth of the number of articles in the English Wikipedia (in blue)

1.3. Structure of Wikipedia

It is obvious that a large-scale project like Wikipedia requires an organization model. Wikipedia articles that have similar subjects are organized into groups, called *Categories*. In extent to this approach, categories may form a higher lever category. Categories are normally found at the bottom of an article page. Clicking the category name brings up a category page listing the articles (or other pages) that have been added to that specific category. Furthermore, it is possible that there may also be a section listing the *subcategories* of that category. Consequently, the subcategorization feature yields a tree-like structure to Wikipedia that facilitates easy navigation.

The top level category of Wikipedia is *Category:Contents*. It contains subcategories with various types of encyclopedic content, content for easy navigation and pages concerning the maintenance of the encyclopedia.

The encyclopedic content consists of:

- **Articles:** All articles organized by various category systems.
- **Featured content:** Articles, pictures and other media selected by the community as being Wikipedia's finest.
- **Glossaries:** Alphabetical lists explaining technical terms related to some field.
- **Lists:** Encyclopedic content in list or tabular form
- **Timelines:** Graphical representation of a chronological sequence of events.

The navigation content consists of:

- **Books**
- **Categories**
- **Indexes:** Alphabetical list of all articles related to a specific topic.
- **Outlines:** Hierarchical list of the most important articles for a specific topic.
- **Portals:** A page highlighting a particular subject.

The maintenance and help content consists of:

- **Help**
- **Administration**

The category system of Wikipedia aims to provide links to all articles in a hierarchy of categories, which readers can browse, knowing essential, defining characteristics of a topic, and quickly find sets of articles on topics defined by those characteristics. Every article should belong to at least one category.

Categories are organized as overlapping trees which are formed by creating links between inter-related categories. Any category may contain subcategories and may belong to more than one parent category. As already mentioned, there is one top-level category, *Category:Contents*, which contains all other categories. Because of this structure, every category apart from the top one must be a subcategory of at least one other category. There are two main kinds of category:

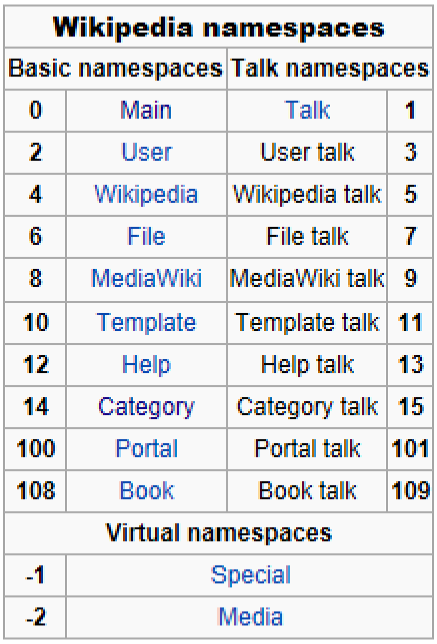
- **Topic categories:** They are named after a topic (usually sharing a name with the Wikipedia article on that topic). For instance, *Category:Greece* contains articles relating to the topic Greece.

- **Set categories:** They are named after a class (usually in the plural). For instance, *Category:Islands of Greece* contains articles whose subjects are islands of Greece.

These two types can sometimes be combined, to create a set-and-topic category (such as *Category: Voivodeships of Poland*, which contains articles about particular voivodeships as well as articles relating to voivodeships in general).

Except from belonging to at least one category, a Wikipedia page has also a specific namespace. A Wikipedia namespace is a set of Wikipedia pages, whose names begin with a particular prefix recognized by the MediaWiki software (following by a colon), or in the case of the main namespace have no such prefix. For instance, the user namespace consists of all pages with names beginning "User:". The encyclopedia articles belong to the main namespace, so they have no prefix. There are currently 22 namespaces in Wikipedia; ten basic namespaces, each with a corresponding talk namespace and two virtual namespaces (listed in table 1).

Table 1: Wikipedia namespaces



Wikipedia namespaces			
Basic namespaces		Talk namespaces	
0	Main	Talk	1
2	User	User talk	3
4	Wikipedia	Wikipedia talk	5
6	File	File talk	7
8	MediaWiki	MediaWiki talk	9
10	Template	Template talk	11
12	Help	Help talk	13
14	Category	Category talk	15
100	Portal	Portal talk	101
108	Book	Book talk	109
Virtual namespaces			
-1	Special		
-2	Media		

Wikipedia's basic namespaces and their functions are the following:

- **Main namespace** (no prefix): It consists of all encyclopedia articles, lists, disambiguation pages and encyclopedia redirects. It is also referred as "mainspace".
- **Project namespace / Wikipedia namespace** (prefix **Wikipedia:**): It contains many types of pages connected with the Wikipedia project itself; information, policy, essays, processes, discussion, etc. Its prefix can be shortened to **WP:** and there are many short redirects in the namespace written with capital letters for ease of access.
- **Portal namespace** (prefix **Portal:**): It consists of reader-oriented portals that help readers find articles for a specific subject and may contain links to encourage contributions to relevant Wikipedia projects.

- **User namespace** (prefix **User:**): It includes user pages and other pages created by individual users for their own personal use. Pages in this namespace can be viewed and modified by others.
- **File namespace / Image Namespace** (prefix **File:**): It contains file description pages for image, video or audio files, with links to the files themselves.
- **MediaWiki namespace** (prefix **MediaWiki:**): It consists of interface texts, such as links and messages that appear on automatically generated pages. Pages in this namespace are permanently protected.
- **Template namespace** (prefix **Template:**): It contains template pages, intended to be transcluded or substituted onto other pages to insert standard text or boxes, such as info-boxes and navigation-boxes.
- **Category namespace** (prefix **Category:**): It contains category pages, which display a list of pages and subcategories, added to a specific category, and optional additional text.
- **Book namespace** (prefix **Book:**): It consists of entries for Wikipedia books, collections of articles about one theme, used to generate downloadable files of printable documents.
- **Help namespace** (prefix **Help:**): It consists of pages which provide help in using Wikipedia and its software, both for users and editors.

The basic namespaces are also referred as “subject spaces”, in contrast to “talk spaces”.

Each of the basic namespaces has a corresponding talk namespace. The talk namespaces are defined by adding talk: to the normal prefix. Most of the pages in the talk namespaces are used for discussion of changes to the respective page in the associated namespace. In addition, pages in the user talk namespace are used to leave messages for a specific user.

The virtual namespaces are the following:

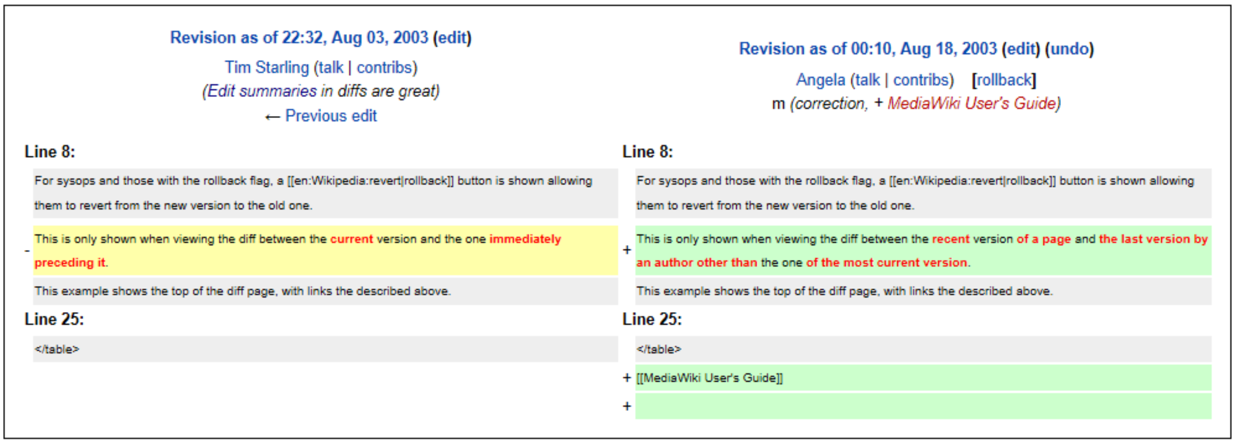
- **Special namespace** (prefix **Special:**): It consists of pages created by the software on demand. These pages can be linked as usual, except when they have parameters, when the full URL must be given like an external link.
- **Media namespace** (prefix **Media:**): It is used to link directly to a file, rather than to the file description page. It does not work on redirects, since the link must be to the file’s true name.

1.4. Editing in Wikipedia

The most important aspect of Wikipedia is the editing operation. This editing model, based on wiki technology, is the essential difference between Wikipedia and traditional encyclopedias. In particular, every article may be edited either by a logged on user or by an anonymous user. It must be noted that different language editions modify this policy; for instance, English Wikipedia allows only to registered users to create a new article. According to this approach, no article is owned by its creator or any other editor. Rather, all articles are agreed on by consensus. By default, any edit that has been applied to an article becomes available immediately. As a result, an article might contain errors, bias or untruthful facts after an edit action, until they are removed or corrected by a potential different editor.

Due to the availability of the edit action, a Wikipedia page may have many versions, each of which is called *revision*. The total of all revisions of a Wikipedia page is called *history*. The history of a page is accessed by clicking the “history” tab at the top of the page and enlists all the revisions applied on the page. A revision of a page contains the changes (data insertion or deletion in any means) applied on the page after the user’s edit, including the date and time (in UTC - Coordinated Universal Time format), the username or the IP address of the user and the *edit summary*. The edit summary of a page is a brief explanation of the respective edit to it. History is available for viewing by any user. Moreover, any user can check the differences between two revisions, in order to keep track of changes. The differences between two versions of a page are found after a *diff* operation. In figure 3, there is a revision comparison example. It is remarked that unchanged text is black on grey background (only parts before and after text are shown). Paragraphs that have changed are highlighted in yellow on the old version side and green on the new version side (table 2). Where whole paragraphs have been removed or inserted, the other side is blank (white). Additionally, removed text is shown in red on the old version, while new text is shown in red on the new version.

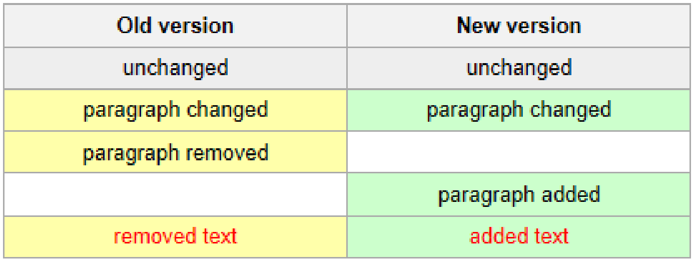
It is evident that any editable page on Wikipedia can be modeled as a graph, which has the authors/editors of the page as nodes and the revisions applied among the page’s versions as edges. In particular, users participate into the revisions of articles and therefore, the relationships between users can be exploited in graph structures.



	Revision as of 22:32, Aug 03, 2003 (edit) Tim Starling (talk | contribs) *(Edit summaries in diffs are great)* ← Previous edit		Revision as of 00:10, Aug 18, 2003 (edit) (undo) Angela (talk | contribs) [rollback] m *(correction, + MediaWiki User's Guide)*
	Line 8:		**Line 8:**
	For sysops and those with the rollback flag, a [[en:Wikipedia:revert|rollback]] button is shown allowing them to revert from the new version to the old one.		For sysops and those with the rollback flag, a [[en:Wikipedia:revert|rollback]] button is shown allowing them to revert from the new version to the old one.
-	This is only shown when viewing the diff between the **current** version and the one **immediately preceding it**.	+	This is only shown when viewing the diff between the **recent** version **of a page** and **the last version by an author other than** the one **of the most current version**.
	This example shows the top of the diff page, with links the described above.		This example shows the top of the diff page, with links the described above.
	Line 25:		**Line 25:**
	</table>		</table>
		+	[[MediaWiki User's Guide]]
		+	

Figure 3: Revision comparison example

Table 2: Color key for revision comparison



Old version	New version
unchanged	unchanged
paragraph changed	paragraph changed
paragraph removed	
	paragraph added
removed text	added text

The editing nature of Wikipedia has sometimes led to edit warring. It is phenomenon where editors who disagree about the content of a page repeatedly override each other’s contributions, instead of trying to resolve the disagreement via discussion. Edit warring creates animosity between editors, making it harder to reach a consensus. Users involved in edit wars risk being blocked or even banned.

In case of edit warring, Wikipedia applies the three-revert rule, known as 3RR. The rule is defined by Wikipedia as follows:

An editor must not perform more than three reverts on a single page within a 24-hour period. Undoing another editor's work – whether in whole or in part, whether involving the same or different material each time – counts as a revert. Violations of the rule normally attract blocks of at least 24 hours. Any appearance of gaming the system by reverting a fourth time outside the 24-hour slot is likely to be treated as a 3RR violation.

A “page” means any editable page on Wikipedia, including talk and project space. A “revert” means any edit that reverses the actions of other editors, in whole or in part, whether involving the same or different material. It can involve as little as one word. A series of consecutive saved revert edits by one user with no intervening edits by another user counts as one revert.

3RR applies per person, not per account. This means that reverts made by multiple accounts operated by one editor count together. Editors violating 3RR for the first time will usually be blocked for 24 hours. Even without a 3RR violation, an administrator may still act if they believe that a user's behavior constitutes edit warring, and any user may report edit warring with or without 3RR being breached. The rule is not an entitlement to revert a page for a specific number of times. If an editor violates 3RR by mistake, they should reverse their most recent reversion. Administrators may take this action into consideration and decide not to block in such cases; for instance, if the user is not a habitual edit warrior and is genuinely trying to rectify their own mistake.

There are actions that are not considered to be reverts for the 3RR:

- Reverting your own actions (“self-reverting”)
- Reverting edits to pages in your own user space, as long as user page guidelines are respected.
- Reverting actions performed by banned users, their sockpuppets and by tagged sockpuppets of indefinitely blocked accounts.
- Reverting obvious vandalism, such as page blanking and adding offensive language.
- Removal of clear copyright violations or content that unquestionably violates the non-free content policy.
- Removal of other content that is clearly illegal in the U.S. state of Florida (location of Wikipedia servers).
- Removal of libelous, biased, unsourced or poorly sourced contentious material violating the policy on biographies of living persons.
- Considerable leeway is given to editor reverting to maintain the quality of a featured article while it appears on the main page.

Another problem that rises due to the editing operation is the vandalism phenomena, which are any addition, removal or change of content in a deliberate attempt to undermine the integrity of Wikipedia. Typical vandalism examples are adding irrelevant obscenities and crude humor to a page, illegitimately blanking pages, and inserting obvious nonsense into a page.

There are various types of vandalism:

- **Abuse of tags:** It includes bad-faith placing of non-content tags or other tags on pages that do not meet such criteria and baseless removal of policy related tags.
- **Malicious account creation:** It includes the account creation with usernames that contain deliberately offensive or disruptive terms.
- **Avoidant vandalism:** It includes removing vandalism-related tags in order to conceal deletion candidates or avert deletion of such content.
- **Illegitimate blanking:** It includes the removal (full or partial) of a page's content without any (important) reason or replacing entire pages with nonsense.
- **Copyrighted material:** It includes the upload or use of material in ways which violate Wikipedia's copyright policies after having been warned.
- **Edit summary vandalism:** It includes the creation of offensive edit summaries in attempt to leave a mark that cannot be easily expunged from the record (often combined with malicious account creation).
- **Hidden vandalism:** It includes any form of vandalism that uses embedded text, which is not visible during viewing the article but visible during editing.
- **Image vandalism:** It includes the upload of inappropriate images on pages or use of images in a disruptive way.
- **Link vandalism:** It includes the addition or change of internal or external links to disruptive, irrelevant or inappropriate targets that are disguised via mislabeling.
- **Illegitimate page creation:** It includes the creation of new pages with the sole intent of malicious behavior (blatant ad pages, personal attack pages, hoaxes).
- **Illegitimate page lengthening:** It includes the addition of very large amounts of bad-faith content to a page, in order to make the page impossible to load or load abnormally slowly, or to provoke machine crashing.
- **Page-move vandalism:** It includes changing the names of pages to disruptive, irrelevant or inappropriate names (available operation for some user groups).
- **Silly vandalism:** It includes the addition of profanity, graffiti or patent nonsense to pages.
- **Sneaky vandalism:** It includes adding plausible misinformation to articles (minor alteration of facts or addition of plausible-sounding hoaxes), hiding vandalism (making two bad edits and only reverting one), simultaneously using multiple accounts or IP addresses to vandalize, abuse of maintenance and deletion templates, or reverting legitimate edits with the intent of hindering the improvement of pages.
- **Spa external linking:** It includes the addition or the continuation of adding spam external links after a warning. A spam external link is added to a page in order to promote websites, products or interests of a user, instead of improving the page editorially.

- **Talk page vandalism:** It includes illegitimate deleting or editing other users' constructive comments.
- **Template vandalism:** It includes the modification of the wiki language or text of a template in a disruptive manner.
- **User and user talk page vandalism:** It includes unwelcome, illegitimate edits to another person's user page.
- **Vandalbots:** They are scripts or robots that attempt to vandalize or add spam to a mass of pages.

Vandalism is prohibited in Wikipedia. While editors are encouraged to warn and educate vandals, administrators may block them at once. Upon the discovery of vandalism, a user may use revert onto the vandalizing edits. Then he may warn the vandalizing editor and notify the administrators if the vandalizing editor persists despite warnings. The administrators should intervene to protect content and prevent further disruption by blocking the malicious users from editing. When warranted, accounts whose main or only use is obvious vandalism or other forbidden activity may be blocked even without warning.

1.5. Reliability of Wikipedia

The editing model of Wikipedia yields an important matter to be examined, which is its reliability. The reliability of Wikipedia (mostly the English-language edition) is assessed in many ways, using analysis of historical patterns, as well as strengths and weaknesses, induced by the editing process. Several studies have been conducted in order to assess the reliability (some of them are presented in the next paragraphs). Moreover, a number of reliability criteria have been defined for various assessments. The most important of these are below:

- Accuracy of information provided within articles
- Appropriateness of the images provided with the article
- Appropriateness of the writing style and focus of the articles
- Susceptibility to, and exclusion and removal of, false information
- Comprehensiveness, scope and coverage within articles and in the range of the articles
- Identification of reputable third-party sources as citations
- Stability of the articles
- Susceptibility to editorial an systematic bias
- Quality of writing

In December 2005, a study was conducted by the journal Nature [2], comparing the accuracy of a sample of articles from Wikipedia and Encyclopedia Britannica [6]. The comparison had been evaluated by academic reviewers who remained anonymous. The results of the study showed that the average Wikipedia article contained 4 errors or omissions, while the average Britannica article contained 3. Additionally, only 4 serious errors were detected both in Wikipedia and Britannica. Finally, the study came to the conclusion that both encyclopedias share the same accuracy, although Wikipedia's articles have a poor structure.

In June 2006, Roy Rosenzweig, professor specializing in American History, performed a comparison of the Wikipedia biographies of 25 Americans to the corresponding biographies found on Encarta and American National Biography Online [7]. He concluded that Wikipedia is accurate at a very high level concerning names, dates and events in U.S. history. However, he stated that articles fail to distinguish important details from trivial ones, as well as that the best references are not provided.

From December 2005 to May 2006, a web-based survey [8] took place by Larry Press, professor of Information Systems at California State University, Dominguez Hills, which analyzed the rate of accuracy and completeness of Wikipedia articles. Fifty people were responsible for the assessment of the articles, from which thirty-eight agreed about Wikipedia’s accuracy, while twenty-three agreed about its completeness. Eighteen people compared the article they reviewed with the corresponding one in Encyclopedia Britannica. The results showed that the six of them preferred Britannica, seven of them Wikipedia, while the remaining five considered them equal. Furthermore, eleven people stated that Wikipedia is substantially complete, compared to seven for Britannica. However, it must be remarked that the selection of the participants was not random, while the criteria inviting the participants are not clear.

In October 2007, Australian magazine PC Authority [9] published an article concerning the accuracy of Wikipedia. It compared Wikipedia’s content with encyclopedias Britannica and Encarta, assisted by experts. The evaluation showed that Wikipedia was comparable to the other encyclopedias, topping the chemistry field.

In April 2008, British computing magazine PC Plus [10] conducted a comparison between the English Wikipedia and the DVD editions of World Book Encyclopedia and Encyclopedia Britannica, in order to assess the coverage of random subjects in each of them. It concluded that the there is good quality in all three encyclopedias. Particularly, it showed that the vast majority of Wikipedia articles provide valuable and accurate information.

2. THESIS MOTIVATION AND RELATED WORK

2.1. Thesis motivation

The rapid growth of Web 2.0 applications such as wikis, blog and social tagging software has led Web users to easily edit, review and publish content collaboratively. One of the largest and most popular Web 2.0 examples is Wikipedia, with an enormous number of articles in many languages. Wikipedia accommodates millions registered users and is one of the most high-ranked websites according to Alexa.com.

Wikipedia can be viewed as a large knowledge repository, which is produced by users who edit the articles. A recent study [11] analyzed its degree of topic diversity. In particular, it was found that Wikipedia is a comprehensive encyclopedia at a very high level. Although its organizing model is not similar to that of the traditional encyclopedias, it provides an equally valid and useful structure. Lately, several projects are running to ensure that important topics receive the appropriate coverage. For instance, *WikiProject Physics* consists of several participants who are contributing to physics-related articles. The project keeps track of missing or obsolete articles, as well as articles which must be reviewed by experts.

However, the interactions among users in articles, modeled through the editing scheme of Wikipedia, give us insight into assuming that the entire content of articles (including their revision history) cannot be considered fully reliable. This point of view can be supported by the vandalism phenomena that appear sporadically in controversial articles. Some malicious users deliberately add, remove or modify the content of them, in attempt to undermine the integrity of Wikipedia. Surprisingly, vandalism is usually detected very quickly and revised back. The existence of co-ordination between Wikipedia users is a main reason that resolves the problem effectively.

The editing nature of Wikipedia raises an important matter to be examined; how to extract experts from Wikipedia users. Expert extraction is a complex problem and is applied in many information filtering systems, like recommendation systems. Concerning Wikipedia, it is useful to detect users that have expertise in certain topics. In this way, when a newly created article is inserted into Wikipedia, the appropriate experts in the topic, to which the article belongs, are capable of evaluating it about its reliability, consistency and neutral point of view. The motivation for this thesis is to design a system for effective expert extraction.

2.2. Related work

The problem of expert mining in Wikipedia is a new research field. Most of the studies that have conducted on Wikipedia are focused on other aspects, like conflict and coordination between authors [12][13][14][15][24], topical coverage [11] and revision behavior of users [16][17]. In this section we present the studies that are related with expert extraction in Wikipedia.

2.2.1. Expert finding using profiles

The study in [18] is based on construction of expert profiles for each individual user that can be later retrieved through queries describing the topic of expertise in which experts are explored. Several algorithms have been applied for the profile construction. Firstly, their proposed naïve approach uses standard Information Retrieval techniques considering the expert profiles as standard documents and indexing them using an inverted index. So, it is possible to create a query representing the topic in which experts are required and query the index in order to extract a ranked list of people using a TF-IDF similarity measure.

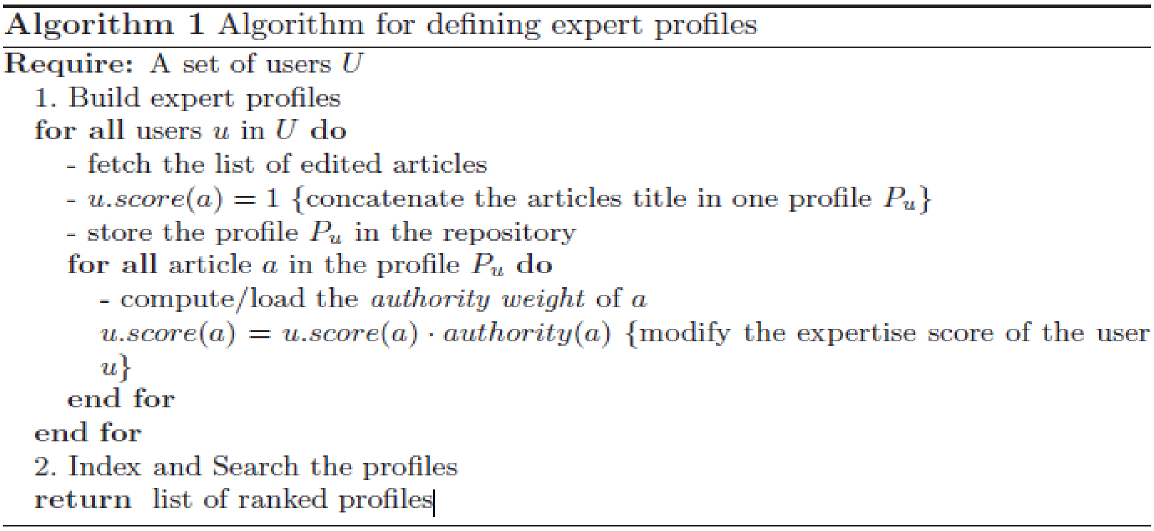
A more sophisticated method takes advantage of the relations between documents, assuming that the most linked documents have a higher authority and they represent at a higher level the expertise of its authors than other less linked documents. In other words, the author of a highly linked/cited article is an expert on the article's topic. So, authors have different weights, depending on the authority of the content they produce. The authority weights of the articles can be computed with popular link analysis algorithms, like PageRank [19] or HITS [20]. However, it must be remarked that this approach is restricted concerning the case where the links in Wikipedia do represent popularity of topics rather than authority of pages as in the Web.

The following notations are used:

- $u.score(i)$ represents how good is the item i as representative of the expertise of user u
- $u.rsv$ denotes the score of a user u as returned by an expert search system for a given query. This metric is used for ranking of the retrieved experts
- $h_{WinN(i)}$ denotes the sub-article of the article h composed of a window of text of size N around the link to the article i

In table 3, the algorithm for defining expert profiles is presented:

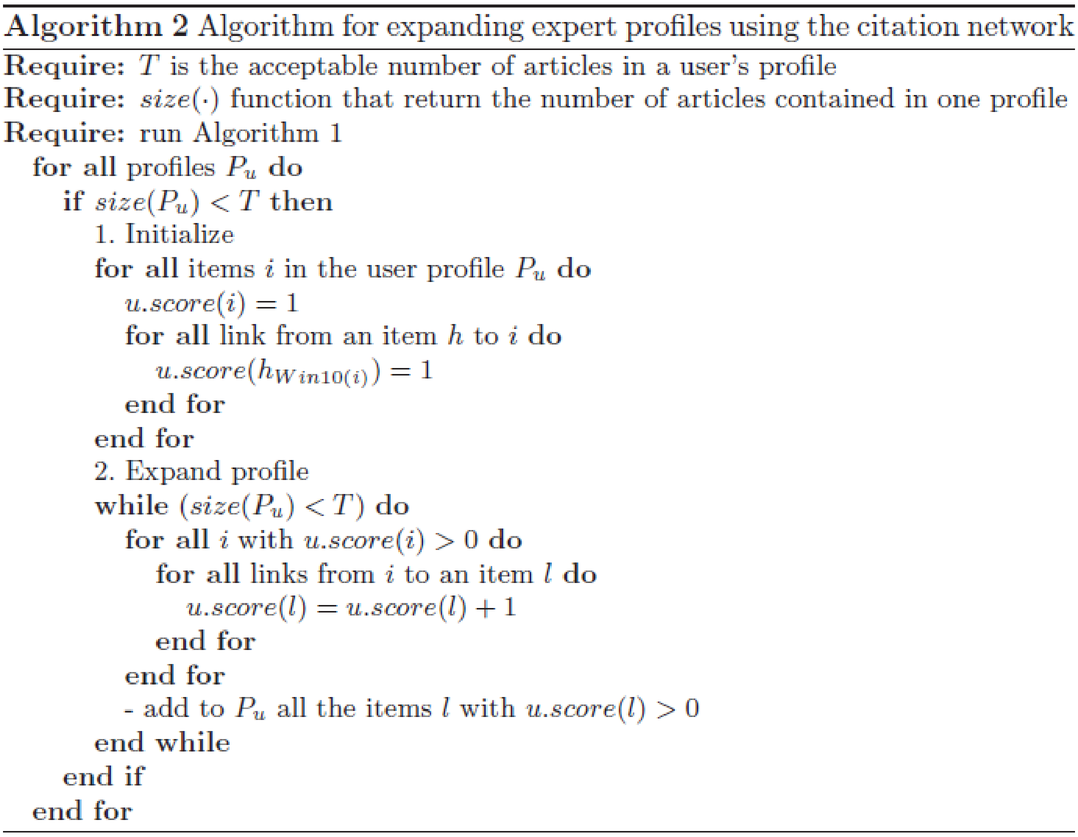
Table 3: Algorithm for defining expert profiles



```
Algorithm 1 Algorithm for defining expert profiles
Require: A set of users $U$
  1. Build expert profiles
  for all users $u$ in $U$ do
    - fetch the list of edited articles
    - $u.score(a) = 1$ {concatenate the articles title in one profile $P_u$}
    - store the profile $P_u$ in the repository
    for all article $a$ in the profile $P_u$ do
      - compute/load the authority weight of $a$
      $u.score(a) = u.score(a) \cdot authority(a)$ {modify the expertise score of the user $u$}
    end for
  end for
  2. Index and Search the profiles
  return list of ranked profiles
```

A more sophisticated approach of the algorithm makes further use of the link structure that exists between the Wikipedia articles. It is possible that the methodology described previously in this section may produce small expert profiles because the corresponding users have participated in the creation of only few articles. To deal with this problem, the profile is expanded using the citation network assuming that the users know something about what they cite. As a result, an expert profile is constructed with topic weights based on Wikipedia's link structure. In table 4, the described algorithm is presented. Initially, it considers a profile that contains the edited articles. Then, for all the items in the profile, it firstly considers the ingoing links. It inserts into the profile topics extracted from a window of N words before and after the anchor of the citing documents. The algorithm also inserts the linked articles with a lower weight into the profile, until the latter is big enough.

Table 4: Algorithm for expanding expert profiles using the citation network



Algorithm 2 Algorithm for expanding expert profiles using the citation network

```
Require: $T$ is the acceptable number of articles in a user's profile
Require: $size(\cdot)$ function that return the number of articles contained in one profile
Require: run Algorithm 1
  for all profiles $P_u$ do
    if $size(P_u) < T$ then
      1. Initialize
      for all items $i$ in the user profile $P_u$ do
        $u.score(i) = 1$
        for all link from an item $h$ to $i$ do
          $u.score(h_{Win10(i)}) = 1$
        end for
      end for
      2. Expand profile
      while $(size(P_u) < T)$ do
        for all $i$ with $u.score(i) > 0$ do
          for all links from $i$ to an item $l$ do
            $u.score(l) = u.score(l) + 1$
          end for
        end for
        - add to $P_u$ all the items $l$ with $u.score(l) > 0$
      end while
    end if
  end for
```

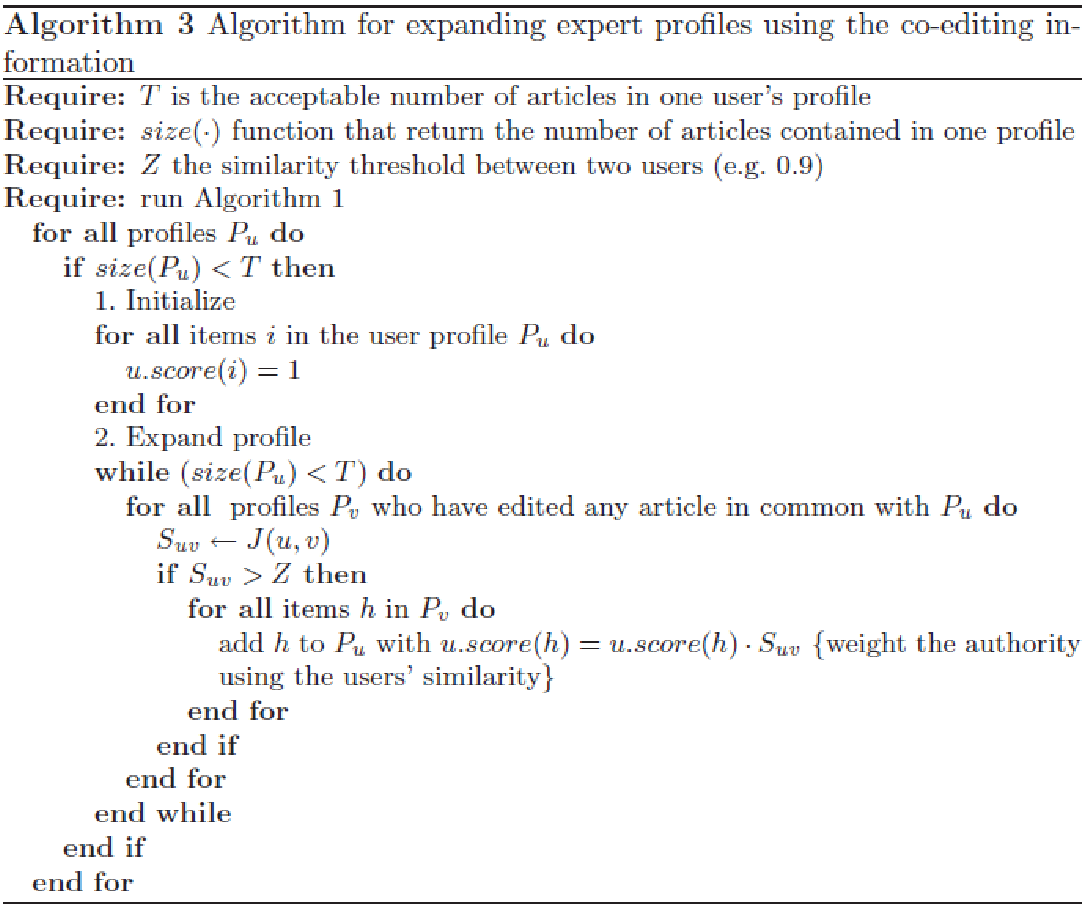
Another strategy that can be adopted uses collaborative filtering techniques. The definition of a measure of similarity between users expands the expert profile including topics of expertise or very similar users and the list of retrieved experts with those similar to the retrieved ones in a pseudo-relevance feedback fashion. The similarity measure that is used is the standard Jaccard measure

$$J(a,b) = \frac{|P_a \cap P_b|}{|P_a \cup P_b|},$$

where P_a, P_b are the sets of articles edited by a user a and a user b, respectively.

The algorithm in table 5 initializes the score for all items in the user profile to one. If required, the profile is expanded adding the articles edited by the most similar users with a score proportional to the users' similarity. The algorithm in table 6 needs to run algorithm in table 3 initially, in order to extract a first list of experts. Then, for each extracted expert, it searches for similar profiles adding them to the results with a Retrieval Status Value (used by the system to rank the experts proportional to the users' similarity).

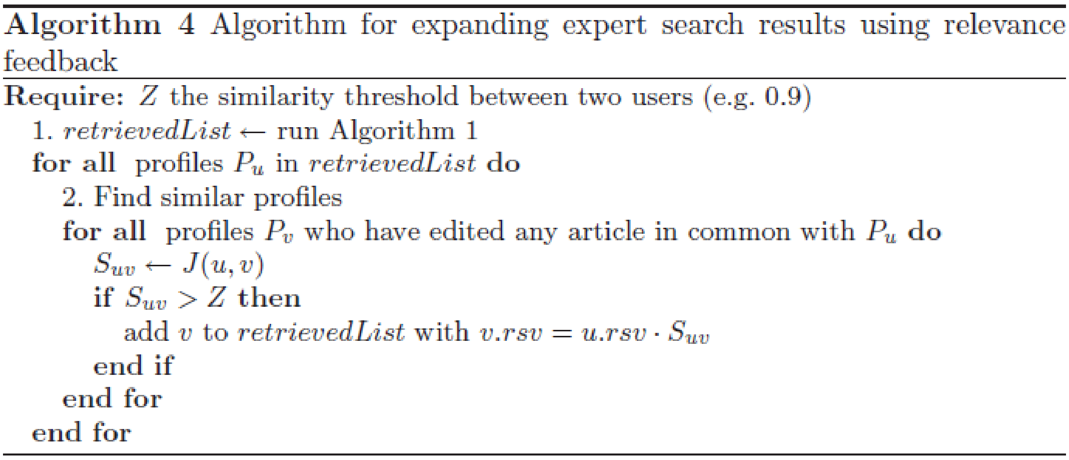
Table 5: Algorithm for expanding expert profiles using the co-editing information



Algorithm 3 Algorithm for expanding expert profiles using the co-editing information

```
Require: $T$ is the acceptable number of articles in one user's profile
Require: $size(\cdot)$ function that return the number of articles contained in one profile
Require: $Z$ the similarity threshold between two users (e.g. 0.9)
Require: run Algorithm 1
  for all profiles $P_u$ do
    if $size(P_u) < T$ then
      1. Initialize
      for all items $i$ in the user profile $P_u$ do
        $u.score(i) = 1$
      end for
      2. Expand profile
      while $(size(P_u) < T)$ do
        for all  profiles $P_v$ who have edited any article in common with $P_u$ do
          $S_{uv} \leftarrow J(u, v)$
          if $S_{uv} > Z$ then
            for all items $h$ in $P_v$ do
              add $h$ to $P_u$ with $u.score(h) = u.score(h) \cdot S_{uv}$ {weight the authority
              using the users' similarity}
            end for
          end if
        end for
      end while
    end if
  end for
```

Table 6: Algorithm for expanding expert search results using relevance feedback



Algorithm 4 Algorithm for expanding expert search results using relevance feedback

```
Require: $Z$ the similarity threshold between two users (e.g. 0.9)
  1. $retrievedList \leftarrow$ run Algorithm 1
  for all  profiles $P_u$ in $retrievedList$ do
    2. Find similar profiles
    for all  profiles $P_v$ who have edited any article in common with $P_u$ do
      $S_{uv} \leftarrow J(u, v)$
      if $S_{uv} > Z$ then
        add $v$ to $retrievedList$ with $v.rsv = u.rsv \cdot S_{uv}$
      end if
    end for
  end for
```

Expert extraction becomes more effective with use of semantics. The defined knowledge taxonomies contribute to detection of correct experts, while ontologies provide support in disambiguating multi senses topics. The author proposes the use of the Yago ontology [21], a combination of notions from Wordnet and Wikipedia, in order to model the expertise and to identify knowledge areas. Therefore, the ontology is able to define better the expert profiles. For instance, knowing that "Macintosh computer" is a subclass of “Computer” supports the system when there are no results for the query “Find an expert on Computer”. In that case, the system can proceed one step further, searching for experts in relative subcategories. It is remarked that this relationship is bidirectional. For example, knowing that “Eclipse” is a “Java tool” provides the possibility to assume that an expert on Eclipse will be an expert (with score proportional to the number of children of the class “Java tool”) on Java tools, too. Additionally, the proposed system uses the algorithm JIGSAW [22] for word sense disambiguation between different topics of expertise. This algorithm performs a calculation for similarity between each candidate meaning for an ambiguous word and all the meanings in its context defined as words with the same POS tag in the same sentence. Similarity is calculated as inversely proportional to path length between concepts in the WordNet IS-A hierarchy. In this case, it is assumed that the suitable meaning belongs to a similar/same concept as words in the context belong to. Furthermore, the system can use co-occurrence statistics to improve further the quality of profiles. For a user profile P_u, topics are disambiguated looking at the context in the related articles. For instance, if user u is an expert in topic “Jaguar” and it is found that word “Car” often co-occurs with word “Jaguar” in the articles which are considered for his profile, then topic “Car” may be added to the expertises of u with the final goal of disambiguation.

2.2.2. Expert finding using link pattern analysis

The goal of the work in [23] is not to find experts in Wikipedia, but to find experts according to their journal publications, given a query proposal. However, the idea of using Wikipedia as the background knowledge source can be applied in various expert finding systems. More specifically, the proposed system has a module that maps terms from publications to Wikipedia pages, since Wikipedia database has vast knowledge diversity and well-defined structure. This mapping procedure is supported by Google, which have the ability to search the query string in the specific web site. The system exploits the link structure of Wikipedia to build the Wikipedia elements of the concept terms. This link structure includes:

1. The Wikipedia page title
2. The Wikipedia categories which contain (1)
3. The Wikipedia categories which contain (2) as a child node
4. The Wikipedia categories which contain (3) as a parent node

In figure 4 a Wikipedia element example is presented. In particular, the Wikipedia element is extended by the page “Back Propagation”. According to Wikipedia’s page/category relationship structure, the term “Back Propagation” is included in the “Neural Network” category. Additionally, “Neural Network” is included in “Information, knowledge and uncertainly” and “Machine Learning” parent categories level.

However, it must be remarked that the link structure of Wikipedia faces some limitations. For instance, some of the Wikipedia categories are the internal tag of Wikipedia, like “Articles with unsourced statements since July 2007”. These internal categories constitute the noise of the Wikipedia element. This noise is removed manually by the system.

The category scope is a key feature of Wikipedia’s link structure. It is noted that the scopes of the Wikipedia categories are unbalanced. For example, the “Neural Network” category contains 17 pages, while the “Artificial Intelligence” category contains 35 pages. This example indicates that the “Neural Network” expertise domain is more specific that the “Artificial Intelligence”. Therefore, the unbalanced nature of scopes is used as the ranking feature in expert finding system.

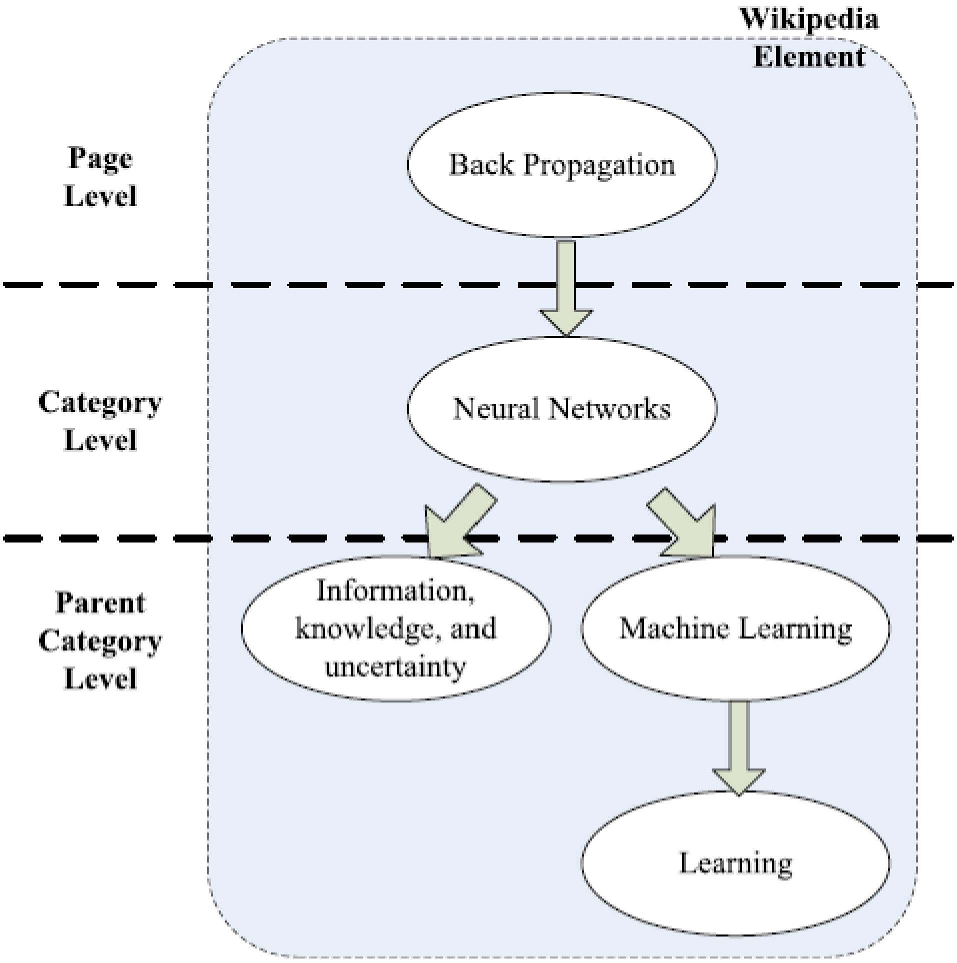


Figure 4: Wikipedia element example

Wikipedia plays an important role both in the expertise indexing process and the expert searching and ranking process. The index process depends heavily on the structure of Wikipedia element. Therefore, the concept terms are indexed by the page level, the category level and the parent/child category level. Similarly, in the search process, the system searches the experts whose expertise match in these 3 levels. The experts get different scores, based on the match levels.

The system ranks the experts according to the similarity difference between their expertise Wikipedia elements from expertise profile and the Wikipedia elements of the query proposal. The following concepts are adopted:

1. If two Wikipedia elements (the expertise of the expert and the expertise of the proposal) match in the page level, then the ranking score will be bigger than a match in the category level.
2. If two Wikipedia elements match in the category level, then the ranking score will be bigger than a match in the parent/child category level.

3. If two Wikipedia elements match in the category level, then the ranking score will cross the inverse of the times of the pages in match category.
4. If two Wikipedia elements match in the parent/child category level, then the ranking score will cross the inverse of the subcategory in match category.

2.3. System overview

Our system is largely based on the work of U. Brandes, P. Kenis, J. Lerner and D. van Raaij in [24]. They propose methods to analyze collaboration networks that encode the edit interactions among users who contribute to a Wikipedia page or a set of pages. Particularly, they introduce the concept of the edit network, which is a graph associated with a Wikipedia page, consisting of the authors of the page and their relationships after the performed edit actions. The edit network is derived from the revision history of a Wikipedia page, where the edit interactions between authors are encoded appropriately.

The main reason of the edit network use is that its graph structure models satisfyingly the relationships between users. Each Wikipedia page can be modeled as a social network, in which every user may be considered as a unique entity, while the dyadic ties between them represent the relationships and interactions. Therefore, it is essential to extract new knowledge and information, studying the patterns and implications of these relationships. Through the edit network, we can extract useful attributes, which help us define a metric, in order to rank Wikipedia users according to their expertise in the article. The edit network and the ranking metric are described in detail in chapter 3. In figure 5 we present the architecture of our system.

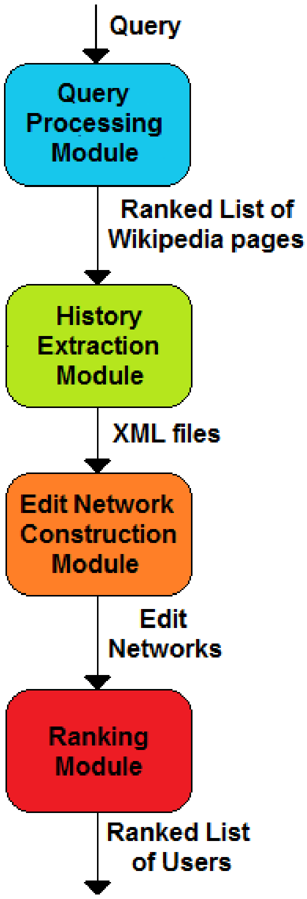

Query
Query Processing Module
Ranked List of Wikipedia pages
History Extraction Module
XML files
Edit Network Construction Module
Edit Networks
Ranking Module
Ranked List of Users

Figure 5: System Architecture

Our proposed system is called WEM (Wikipedia Expert Miner) and consists of the following basic modules:

1. **Query processing module:** This module takes as input a query for the Wikipedia database, in order to provide as output a ranked list of Wikipedia pages, which serve as results. It is possible that a search can produce only one result, which could be a Wikipedia article or a Wikipedia category.
2. **History extraction module:** For every Wikipedia page that belongs to the search results, a XML file is created, which contains all the revisions, performed on the specific Wikipedia page, and revision history is built in a format that the next module understands. It is possible to include the revisions of more than one Wikipedia page in the file. The XML file for a page has the following structure:

   ```
   <page>
      <title></title>
      <id></id>
      <revision>
          <id></id>
          <timestamp></timestamp>
          <contributor>
              <username></username>
              <id></id>
          <contributor>
          <text></text>
      </revision>
      <revision>
   </page>
   ```

   The `page` element contains other XML elements related with the Wikipedia page. More specifically, it includes `title`, `id` (the identification number of the page), and many instances of the `revision` element. The `revision` element includes `id` (the identification number of the revision), `timestamp` (the time point of the revision), `text` (the full text version of the page in the present revision) and `contributor` (info about the contributor of the revision). In case of a logged in user, the `contributor` element includes `username` and `id` (the identification number of the user), as presented above, while in case of an anonymous user it only includes `ip` (the ip address of the user).
3. **Edit network construction module:** This module builds the edit network for all the Wikipedia pages, represented in the XML files. Then, several attributes of the edit network are computed for the ranking process.
4. **Ranking module:** The final module of our system provides a ranking of Wikipedia users according to a ranking metric, which is based on the attributes of the edit

network. The ranking metric is computed for each user in the edit network and then all users are sorted in descending order.

3. EDIT NETWORK

3.1. Introduction to edit-network

In this chapter, we describe the concept of the edit-network [24] which is used for knowledge extraction from Wikipedia pages. The edit-network is a useful graph structure that represents the interactions among users in a specific Wikipedia page. As mentioned before in chapter 1, every Wikipedia page can be modeled as a social network, in which every user may be considered as a unique node, while the dyadic ties between them correspond to their relationships. The edit-network follows a similar strategy and maintains a list of useful attributes, in order to extract new knowledge and information, by studying the patterns and implications of these relationships.

The edit-network for a Wikipedia page p is a directed graph $G=(V,E,A)$, which represents the interactions among users in the specific page. It consists of the following components:

1. The nodes V of the graph (V,E) represent the authors that have at least one revision on page p.
2. The directed edges $E \subseteq V \times V$ of the graph (V,E) reflect the edit interactions among authors. It is defined that a particular pair of authors $(u,\upsilon) \in V \times V$ is in E, if u performs one of the following three actions, with respect to υ.
   a. u *deletes* text, which has been written by υ (delete action).
   b. u *undeletes* text, which has been deleted by υ and has been written by a potential different author w(undelete action)
   c. u *restores* text, which has been written by υ and has been deleted by a potential different author w (restore action)

   It is obvious that edges, connecting an author with himself, are allowed, since there is a possibility that an author may revise text written by himself.
3. A is a set of weighted attributes on nodes and edges (explained in the following sections).

3.2. Attributes of the edit-Network

The attributes on nodes and edges of the edit-network represent the amount of text, which Wikipedia users add, delete or restore. This amount is measured by the number of words. In addition, it must be noted that in case of deletion the original authors of the text are stored. In case of restoration, the edit-network stores both the original authors and deleters. Furthermore, the edit-network keeps track of the edit-actions timepoint by indexing the attributes with the revision number. It is assumed that a history of a given page is a sequence of revisions $R=(r_1,r_2,...,r_N)$, ordered by increasing timestamps $1,2,...,N$.

3.2.1. Basic attributes

The edit-network defines as basic attributes the attributes, which have to be computed by the network construction algorithm and cannot be computed by other attributes. In table 7 and table 8 the basic attributes on edges and nodes are presented, respectively.

Table 7: Basic attributes on edges

Basic Attributes on Edges For each timepoint $i \in \{1,2,...,N\}$ and each pair of authors $(u,\upsilon) \in V \times V$,	
$delete_i(u,\upsilon)$	denotes the number of words deleted by u in revision r_i and written by υ at an earlier revisions r_j ($j < i$).
$undelete_i(u,\upsilon)$	denotes the number of words restored by u in revision r_i, deleted by υ in revisions r_j ($j < i$), and written by a potential different author w in revisions r_ℓ ($\ell < j < i$).
$restore_i(u,\upsilon)$	denotes the number of words restored by u in revision r_i, written by υ in revisions r_j ($j < i$), and deleted by a potential different author w in revisions r_ℓ ($\ell < j < i$).

It is obvious that $delete_i(u,\upsilon)$, $undelete_i(u,\upsilon)$ and $restore_i(u,\upsilon)$ are equal to zero, if u is not the author of revision r_i.

Table 8: Basic attributes on nodes

Basic Attributes on Nodes For each timepoint $i \in \{1,2,...,N\}$ and each author $u \in V$,	
$add_i(u)$	denotes the number of words that are added by u in revision r_i.
$authorship_i(u)$	denotes the number of words in revision r_i that have been authored by u, i.e., all words that have been added to the text by u in a revision r_j, $j \leq i$ and that are still there in r_i.

It is noted that if u is not the author of r_i, then $add_i(u)$ is zero. However, even in this case $authorship_i(u)$ might be greater than zero. Furthermore, it always holds that $add_i(u) \leq authorship_i(u)$, since, at timepoint i, u is the author of at least those words that he added in revision r_i, and it holds that $authorship_i(u) \leq \sum_{j=1}^{i} add_j(u)$, since, at timepoint i, u can only be the author of those words that he added before or in revision r_i.

3.2.2. Derived attributes

The derived attributes of the edit-network are the attributes, which have to be computed by the basic attributes. In table 9 and table 10 the derived attributes on edges and nodes are presented, respectively.

Table 9: Derived attributes on edges

Derived Attributes on Edges For edges $(u,\upsilon)\in E$,	
$delete(u,\upsilon)=\sum_{i=1}^{N} delete_i(u,\upsilon)$	denotes the number of words deleted by u and written by υ over all timepoints $i\in\{1,2,...,N\}$.
$undelete(u,\upsilon)=\sum_{i=1}^{N} undelete_i(u,\upsilon)$	denotes the number of words restored by u and deleted by υ over all timepoints $i\in\{1,2,...,N\}$.
$restore(u,\upsilon)=\sum_{i=1}^{N} restore_i(u,\upsilon)$	denotes the number of words restored by u and written by υ over all timepoints $i\in\{1,2,...,N\}$.
$revise(u,\upsilon)=delete(u,\upsilon)+undelete(u,\upsilon)$	denotes how much u undoes υ's edits over all timepoints $i\in\{1,2,...,N\}$, with respect to number of words.

It must be noted that large values on attributes $delete(u,\upsilon)$ and $undelete(u,\upsilon)$ indicate a negative relationship from u to υ. More specifically, if u deletes a lot of text written by υ, then u apparently disagrees with υ's contributions to the article. Similarly, if u undeletes a lot of text, which has been previously deleted by υ, then u disagrees with υ concerning the removal of this text from the article. On the other hand, large values on attribute $restore(u,\upsilon)$ indicate a positive relationship from u to υ, since u defends υ's contributions against deletion. In addition, it must be mentioned that attribute $revise(u,\upsilon)$ is a measure of how much u disagrees with υ, while attribute $restore(u,\upsilon)$ is a measure of how much u disagrees with υ.

Table 10: Derived attributes on nodes

Derived Attributes on Nodes For each author $u\in V$,	
$delete_i(u)=\sum_{\upsilon\in V} delete_i(u,\upsilon)$	denotes the total number of words deleted by u in revision r_i and written by different authors.
$restore_i(u)=\sum_{\upsilon\in V} restore_i(u,\upsilon)=$ $\sum_{\upsilon\in V} undelete_i(u,\upsilon)$	denotes the total number of words restored by u in revision r_i and written by different authors (or deleted by different authors).

$add(u)=\sum_{i=1}^{N} add_i(u)$	denotes the total number of words written by u in all revisions of the page history.
$delete(u)=\sum_{i=1}^{N} delete_i(u)$	denotes the total number of words deleted by u in all revisions of the page history.
$restore(u)=\sum_{i=1}^{N} restore_i(u)$	denotes the total number of words restored by u in all revisions of the page history.
$activity(u)=add(u)+delete(u)+restore(u)$	denotes the total action (addition, restoration and deletion) of u in all revisions of the page history, with respect to number of words. It is called *edit-activity* and is a measure of *involvement*.
$netadded(u)=add(u)+restore(u)-delete(u)$	denotes the total number of words by which u increased the length of the text. It is called *net-amount of added words* and indicates how an author contributes to a page.
$netaddedratio(u)=netadded(u)/activity(u)$	denotes the ratio between the *net-amount of added words* and the *edit-activity* of u
$revisor(u)=\sum_{\upsilon\in V} revise(u,\upsilon)$	denotes the number of words that u deletes after they have been added, or restores after they have been deleted. It is called *u's degree as a revisor* and is a measure of the *undo-activity* of u.
$revised(u)=\sum_{\upsilon\in V} revise(\upsilon,u)$	denotes the number of words that have been written by u, before they have been deleted, and the number of words that have been deleted by u, before they have been restored. It is called *u's degree as being revised* and is a measure of how much u's edits are undone later.

Firstly, it must be mentioned that the attributes $add(u)$, $delete(u)$ and $restore(u)$ indicate u's role as being provider of new content, someone who removes content or someone who defends content against deletion, respectively. Furthermore, it holds that $authorship_N(u)\le add(u)$ at last timepoint N, since u can only be author of those words that he added. It is assumed that u is not the author of a word that he restores. Rather, the original author of the word before the deletion increases his $authorship$ by one. Normally, $authorship_N(u)$ will be smaller than $add(u)$, since words written by u might be deleted in next revisions. In addition, if the attribute $netadded(u)$ is positive, then u tends to increase the text by adding new words or restoring deleted text. If it is negative, then u tends to decrease the length of the text by deleting parts of it. In all cases, the absolute value of $netadded$ is always bounded by $activity$. As a result, the ratio $netaddedratio(u)$ lies between minus one and plus one. If $netaddedratio(u)$ is equal to minus one, then u dedicates all his activity to deletion of text, and if $netaddedratio(u)$ is plus one, then u dedicates all his activity to either adding or restoring text. Moreover, it holds by their definition that attributes $revisor(u)$ and $revised(u)$ are bounded from above by $activity(u)$. Authors that present $revisor \approx activity$ adopt a *reactive* behavior, since they are mostly concentrated on undoing changes made by others. On the other hand, authors with $revised \approx activity$ do not succeed in making permanent edits, since these are mostly undone by others afterwards.

3.3. Computing the edit-network

This section describes in detail everything that is important for the construction of the edit-network. In particular, it presents the conventions that have been made for the text processing, the input/output and data structures of the edit network, as well as the construction algorithm.

3.3.1. Text-processing conventions

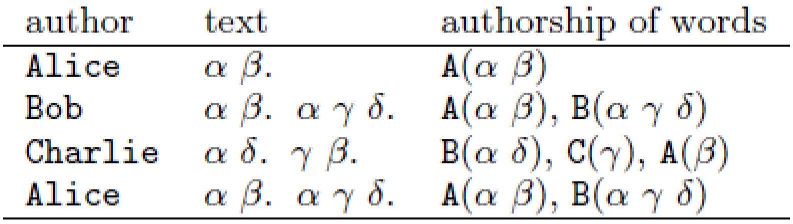
In this section, we mention the conventions concerning the text processing for the construction of the edit-network. More specifically, we give insight into detecting cut and pasted text to a different location, edits that are reverts and how duplicated text is treated.

Firstly, it is evident that the granularity of authorship is on the word level, i.e. each word has exactly one author and different words may have different authors. An important aspect of the text processing is if word ordering should be taken into consideration. In case of ordering, let us assume that an author restructures a Wikipedia page by cutting and pasting large part of the text to different places. This scenario will be considered as a massive deletion of text and addition of a newly created one. If word ordering is not taken into account, then it would be impossible to determine authorship of duplicated words. This problem is solved with the assumption that words are assembled to sentences, each of which represents one statement, a fact or a claim. In particular, the whole text is modeled as an unordered set of sentences, which in turn are modeled as ordered lists of words. According to this assumption, moving a complete sentence to another position, duplicating a complete sentence or deleting a duplicated sentence are cases that do not yield a change to the text. However, it must be noted that two words within the same sentence may have different authors. For example, if an author modifies a sentence partially by adding some words to it, then he becomes only the author of the newly added words and not of the previous ones.

An important aspect of text processing is also to determine the boundaries of the sentences, which are defined using punctuation and capitalization. After the split of the text into sentences, punctuation and capitalization can be ignored, since they do not affect the construction algorithm. The last point taken into consideration is the detection of reverts in a Wikipedia page, i.e. revisions that set back the page to an earlier version. For example, if a user u deletes the whole content of a page in revision r_i and another user υ restores it in revision r_{i+1} to the version in revision r_{i-1}, then user υ should not be credited as the author of the whole text. Rather, authorship of all words rolls back to the version of revision r_{i+1}, while it is stored that υ performed an undelete action to the text deleted by u.

In order to provide a better understanding of the revert concept, table 11 presents an example of four revisions and the resulting authorship of words determined by the aforementioned conventions. In this example, Greek letters denote words and periods delimit sentences. It is evident that the third revision is interpreted in the way that *Charlie* interchanged the first and second sentence, deleted word γ in sentence $\alpha\ \gamma\ \delta$, and changed word α in sentence $\alpha\ \beta$ to γ. The interchange of the two sentences is established by the fact that sentence $\alpha\ \gamma\ \delta$ and sentence $\alpha\ \delta$ have a common subsequence of length two, so they constitute the most similar pair of sentences. After the third revision, *Charlie* is the author of γ, *Alice* is the author of α and δ. In addition, *Charlie* has deleted one word of *Alice* (word α from sentence $\alpha\ \beta$), one word from *Bob* (word γ from sentence $\alpha\ \gamma\ \delta$) and has added word γ in sentence $\gamma\ \beta$. The fourth revision is considered a revert, in which *Charlie*'s edits are undone. Therefore, *Alice* deleted *Charlie*'s word γ, restored her own word α, and restored *Bob*'s word γ, setting attribute $undelete(Alice, Charlie) = 2$.

Table 11: Example of four revisions on a page



author	text	authorship of words
Alice	$\alpha\ \beta.$	A($\alpha\ \beta$)
Bob	$\alpha\ \beta.\ \ \alpha\ \gamma\ \delta.$	A($\alpha\ \beta$), B($\alpha\ \gamma\ \delta$)
Charlie	$\alpha\ \delta.\ \ \gamma\ \beta.$	B($\alpha\ \delta$), C(γ), A(β)
Alice	$\alpha\ \beta.\ \ \alpha\ \gamma\ \delta.$	A($\alpha\ \beta$), B($\alpha\ \gamma\ \delta$)

3.3.2. Input-Datastructures-Output

This section describes the auxiliary data structures that contribute to the successful construction of the edit-network, as well as the input and output of the construction algorithm. The first term that must be defined is that of revision. A *revision* of a Wikipedia page is a tuple of the form $r = (time, author, text)$, where $time$ denotes the exact timestamp of the revision (given in the form of date), $author$ denotes the username of the actor, if he has been logged in during the revision, or his IP-address, if the revision has been done anonymously, and $text$ is the complete text of the page, after the revision. The construction algorithm gets as input the history of the page $R = (r_1, r_2, ..., r_N)$, which includes all the revisions, ordered by increasing timestamps.

During the processing of history, each revision r_i, $i = 1, 2, ..., N$ will successively be augmented by an unordered set $S_i = \left\{ s_{i1}, s_{i2}, ..., s_{il_i} \right\}$ of sentences s_{ij}. Each sentence $s_{ij} = \left(w_{ij1}, w_{ij2}, ..., w_{ijk_{ij}} \right)$ is an ordered list of pointers to words w_{ijh}. Each word w is modeled as triple of the form $w = (charseq(w), author(w), deleter(w))$, where $charseq(w)$ denotes the character sequence of word w, $author(w)$ is a pointer to the author who has written the word, and $deleter(w)$ is a pointer to the author who has deleted the word. In case the word has not yet been deleted or has been undeleted afterwards, then the $deleter$ -variable is set to null.

The construction algorithm performs the instantiation of a new word $w = (charseq(w), author(w), deleter(w))$, if the word is newly added. In particular, if a complete sentence $s_{ij} = \left(w_{ij1}, w_{ij2}, ..., w_{ijk_{ij}} \right)$ is copied from revision r_i to revision r_{i+1}, then no new word objects are instantiated. Rather, the set of sentences S_{i+1} contains a sentence that is the identical list of pointers as s_{ij}. Consequently, while processing the history, the edit-network $G=(V,E,A)$ is successively build up. During the processing of revision $r_i=(time_i, author_i, text_i)$, author $u=author_i$ is inserted into V (if not already in), and attributes $add_i(u)$, $delete_i(u,\upsilon)$, $undelete_i(u,\upsilon)$ are updated (these attributes are initialized to zero).

3.3.3. Edit-network construction algorithm

In this section, we present the algorithm for the edit-network construction [25]. It consists of two basic steps; the first step implements the processing of the first revision and acts as an initialization step, while the second one implements the processing of the subsequent revisions. It is remarkable that the second step is divided into two cases; the first case handles revisions that are not reverts, while the second one handles the reverts. At the end, we present in table 12 the algorithm in pseudo-code.

3.3.3.1. Processing the first revision

Let us consider $r_1=(time_1, author_1, text_1)$ the first revision of the Wikipedia page. The actor $u=author_1$ is inserted into the set of authors V. Firstly, the text of revision $text_1$ is divided into sentences. Duplicated sentences are removed and the distinct ones split into words (i.e. character sequences delimited with whitespace). For each character sequence, a new instance w of a word is created, with $author(w)$ pointing to u, and attribute $add_1(u)$ is incremented by one. For each sentence a list of pointers to its words is created and is inserted into the set of sentences S_1.

3.3.3.2. Processing subsequent revisions

Let us assume that revisions $r_1, r_2, ..., r_i$, $i \geq 1$ are already processed and $r_{i+1}=(time_{i+1}, author_{i+1}, text_{i+1})$ is the next the revision. For the remainder of this section, we consider that $u=author_{i+1}$. Firstly, actor u is inserted into V (if not already in). Thereafter, $text_{i+1}$ is compared with $text_j$, $j=i,i-1,...,1$ for equality. If there is some $j=i,i-1,...,1$, so that it holds $text_{i+1}=text_j$, then revision r_{i+1} is a revert. In equality, revision r_{i+1} is not a revert.

3.3.3.2.1. Handling non-revert revisions

If revision r_{i+1} is not a revert, then its text $text_{i+1}$ is compared with the set of sentences S_i of revision r_i, in order to determine which words have been copied, added or deleted. The first step is the initialization of a temporary set of sentences. Let S'_{i+1} be the set of sentences determined from $text_{i+1}$, as in section 3.3.3.1. Thereafter, an empty set of sentences S_{i+1} is created that will be filled in the following steps. After set S_{i+1} is filled, then the temporary set S'_{i+1} is discarded.

- **Handle sentences copied from** r_i

For each sentence $s \in S_i, S'_{i+1}$, create a list of pointers to words identical to s and insert it into S_{i+1}. As a result, these words have the same authors in revision r_i. Mark sentence s as processed both in sets S_i, S'_{i+1}. No words are added and no edges induced by copied sentences.

- **Process successively the most similar pairs of sentences**

While there are still unprocessed sentences in S_i and in S'_{i+1}, let (s, s') be the pair of unprocessed sentences $s \in S_i$ and $s' \in S'_{i+1}$ with the longest common subsequence. At this point we have made our own assumption; we consider that a pair of sentences has the longest common subsequence if their common subsequence is greater or equal to 75% of the minimum length between the two sentences. At this step sentence s' is considered as a slightly changed version of sentence s, so it must be determined which words have been copied, deleted or added between s and s' . Mark s and s' as processed and compute a shortest edit-script from s and s'.

An edit-script is a sequence$\left((w_1, a_1), (w_2, a_2), ..., (w_k, a_k)\right)$ of pairs $\left(w_j, a_j\right)$, where w_j denotes the word and a_j denotes the *edit-action* for the specific word. The edit-action can be either *NONE*, *DELETE* or *ADD*. If a_j is *NONE*, then word w_j is found in both sentences s and s'. If a_j is *DELETE*, then word w_j is found in sentence s, but not in sentence s', and if a_j is *ADD*, then word w_j is found in sentence s', but not in sentence s. The order of the edit-script is important for keeping track words that have been moved. For instance, if only one word w has been moved from the beginning of the a sentence to the end, then the first pair in the corresponding edit-script is $(w, DELETE)$ and the last one is (w, ADD). It is evident that all words labeled by *NONE* and *DELETE* constitute the sentence s, while all words labeled by *NONE* and *ADD* constitute the sentence s'.

After the edit-script construction, create an empty sentence $s*$ and traverse the edit-script from the start to the end. If the current pair in the edit-script is $(w, NONE)$, then create a pointer to w and add it to the end of the sentence $s*$. Notice that the author of w remains constant. If the current pair in the edit-script is $(w, DELETE)$, then let υ be the author of w. Increment attribute $delete_{i+1}(u, \upsilon)$ by one and let the $deleter$ -variable of w point to υ (no pointer is added to sentence $s*$). If the current pair in the edit-script is (w, ADD), then increment attribute $add_{i+1}(u)$ by one, create a new instance of a word from the character sequence w, set its $author$ -variable to u, and add a pointer to that word at the end of the sentence $s*$. After all pairs of the edit-network have been processed, insert sentence $s*$ into set S_{i+1}.

- **Process deleted sentences**

After the above two steps, there are sentences in set S_i, which are still unprocessed. For each such sentence $s = (w_1, w_2, ..., w_k)$ traverse all words w_h, $h = 1, 2, ..., k$, increment $delete_{i+1}(u, author(w_h))$ by one and set the $deleter$ -variable of w_h point to u.

- **Process added sentences**

Similarly, there are sentences in set S'_{i+1}, which are still unprocessed. For each such sentence $s=(w_1,w_2,...,w_k)$ create a list of pointers $s*$, traverse all words w_h, $h=1,2,...,k$, increment $add_{i+1}(u)$ by one, create a new instance of a word from the sequence of characters w_h, set its $author$ -variable to u, and add a pointer to that word at the end of the sentence $s*$. When all words of s have been processed, insert $s*$ into set S_{i+1}.

3.3.3.2.2. Handling revert revisions

Let $text_{i+1}$ equal $text_j$ and let $j<i+1$ be the largest index with this property. Create a set of sentences S_{i+1} by copying all lists of pointers of S_j. Thereby, all authors in revision r_{i+1} are exactly the same as in revision r_j. It is evident that no words are added by a revert, so attribute $add_{i+1}(u)$ is not increased. The next step would be to update the attributes $delete_{i+1}(u,\cdot)$, $undelete_{i+1}(u,\cdot)$ and $restore_{i+1}(u,\cdot)$.

To achieve this, sets S_{i+1} and S_i are compared in the same way sets S'_{i+1} and S_i are compared in section 3.3.3.2.1. The steps of processing the copied and deleted words are exactly the same. The only steps that are slightly changed are those of processing the added words. In a case of a revert, these words are not added, but they are restored. For any restored word w let υ be the author of w and let υ' be the user who deleted w at some timepoint between j and $i+1$ (this author is pointed by the $deleter$ -variable of w). Increase the attributes $restore_{i+1}(u,\upsilon)$ and $undelete_{i+1}(u,\upsilon')$ by one.

Table 12: Edit-network construction algorithm

```
EDIT_NETWORK_CONSTRUCTION
Input: History $R=(r_1,r_2,...,r_N)$
Output: Edit-Network $G=(V,E,A)$

// Processing the First Revision
$u \leftarrow author_1$;
insert($u$,$V$);
sentenceList←splitIntoSentences($text_1$);
sentenceList←removeDuplicates(sentenceList);
$S_1 \leftarrow \{\ \}$;
for each sentence in sentenceList
   createSentenceStruct(s);
   wordList←splitIntoWords(sentence);
   for each word in wordlist
      createWordStruct(w);
      w.charseq←word;
      w.author←u;
      w.deleter←null;
      insert(w,s);
   end for
   insert(s,$S_1$);
```

```
end for

// Processing Subsequent Revisions
revertPos←-1;
for each i=1 to N-1
   $u \leftarrow author_{i+1}$;
   if($u \notin V$)
     insert($u$ , $V$);
   end if
   for each j=1 to i
      if($text_{i+1} = text_j$)
         revertPos=j;
      end if
   end for
   if(revertPos=-1)
      sentenceList←splitIntoSentences($text_{i+1}$);
      sentenceList←removeDuplicates(sentenceList);
      $S'_{i+1} \leftarrow \{\ \}$;
      for each sentence in sentenceList
         createSentenceStruct(s);
         wordList←splitIntoWords(sentence)
         for each word in wordlist
            createWordStruct(w);
            w.charseq←word;
            w.author←u;
            w.deleter←null;
            insert(w,s);
         end for
         insert(s, $S'_{i+1}$);
      end for
      $S_{i+1} \leftarrow \{\ \}$;
      for each s in $S_i$
         if(s.processed=false)
            for each s′ in $S'_{i+1}$
               if(s′.processed=false)
                  if(s=s′)
                     insert(s, $S_{i+1}$);
                     s.processed=true;
                     s′.processed=true;
                  end if
               end if
          end for
         end if
      end for
      for each s in $S_i$
         if(s.processed=false)
            for each s′ in $S'_{i+1}$
               if(s′.processed=false)
                  if(lcs(s,s′)>=3/4*min{s,s′})
                     s* ← $\{\ \}$
                     for each w in s
                         for each w′ in s′
                             if(w.charseq=w′.charseq)
                                insert(w,s*);
```

```
                              remove(w,s);
                              remove(w′,s′);
                           end if
                       end for
                   end for
                   for each w in s
                       $\upsilon \leftarrow$ w.author;
                       $delete_{i+1}(u,\upsilon)++$;
                       w.deleter $\leftarrow u$;
                   end for
                   for each w′ in s
                       $add_{i+1}(u)++$;
                       createWordStruct(w″);
                       w″.charseq$\leftarrow$w′.charseq;
                       w″.author$\leftarrow$u;
                       w″.deleter$\leftarrow$null;
                       insert(w″,s*);
                   end for
                   insert(s*, $S_{i+1}$);
                   s.processed=true;
                   s′.processed=true;
               end if
            end if
        end for
      end if
   end for
   for each s in $S_i$
      if(s.processed=false)
          for each w in s
              $delete_{i+1}(u, w.author)++$;
              w.deleter$\leftarrow u$;
          end for
          s.processed=true;
      end if
   end for
   for each s′ in $S'_{i+1}$
      if(s′.processed=false)
          s* $\leftarrow \{\ \}$
          for each w′ in s′
              $add_{i+1}(u)++$;
              createWordStruct(w″);
              w″.charseq$\leftarrow$w′.charseq;
              w″.author$\leftarrow$u;
              w″.deleter$\leftarrow$null;
              insert(w″,s*);
          end for
          insert(s*, $S_{i+1}$);
          s′.processed=true;
      end if
   end for
end if
else
  j$\leftarrow$revertPos;
  $S_{i+1} \leftarrow S_j$
  for each s in $S_i$
```

```
    if(s.processed=false)
      for each s’ in $S_{i+1}$
        if(s’.processed=false)
          if(s=s’)
            s.processed=true;
            s’.processed=true;
          end if
        end if
      end for
    end if
  end for
  for each s in $S_i$
    if(s.processed=false)
      for each w in s
        $delete_{i+1}(u, w.author)++$;
        w.deleter ← $u$;
      end for
      s.processed=true;
    end if
  end for
  for each s’ in $S_{i+1}$
    if(s’.processed=false)
      for each w’ in s’
        $\upsilon$ ← w’.author;
        $\upsilon'$ ← w’.deleter;
        $restore_{i+1}(u,\upsilon)++$;
        $undelete_{i+1}(u,\upsilon')++$;
      end for
      s’.processed=true;
    end if
  end for
end if
```

4. RANKING EDIT NETWORK USERS

4.1. Overview of ranking

After the construction of the edit network and the computation of basic and derived attributes, we can now rank Wikipedia users in terms of their expertise. Initially, we should define an indicator to be used as our ranking metric. A high quality metric should consider each user's contribution to a page, as well his positive and negative relationships with other users.

Concerning expertise in a specific topic, we use the category system of Wikipedia. In particular, we assume that if a user has high ranking in most of the articles that belong to a Wikipedia category, which represents a topic, then he has high ranking in the topic too. In the following sections we describe in detail our proposed metric.

4.2. Ranking for a Wikipedia page

The metric that ranks users for a Wikipedia page i is the following:

$$absoluteRank_i(u) = \left[a \cdot \frac{totalRevisor(u)}{totalRevisor(u) + totalRevised(u)} + (1-a) \cdot \frac{authorship_N(u)}{add(u)} \right] \cdot authorship_N(u)$$

We remind that:

- $totalRevisor(u)$ is the number of words that u deletes after they have been added, or restores after they have been deleted.
- $totalRevised(u)$ is the number of words that have been written by u, before they have been deleted, and the number of words that have been deleted by u, before they have been restored
- $authorship_N(u)$ denotes the number of words in revision r_N that have been authored by u
- $add(u)$ is the total number of words written by u in all revisions of the page history.

The variable a is a weight parameter that is used to give weights to the two terms of the metric. We consider that $a = 0.5$, i.e. we assume that the two fractions have the same influence on ranking procedure.

The first fraction of our metric represents the consistency of a user in terms with other users, considering both his role as revisor and as being revised. It is evident that when this fraction has high value, then the user is more reliable for the specific page, because his edits seem to persist after revisions.

The second fraction of our metric represents the absolute consistency of a user. $authorship_N(u)$ is a very useful attribute for our system, since it denotes the number of words that have been remained constant after all the revisions. However, it is important to find the percentage of consistency for the user. This can be computed by dividing

$authorship_N(u)$ with the total number of words that have been authored by him (edited and non-edited), i.e. $add(u)$.

It is possible that several users may exist with similar values in the above fractions, but they may have different contribution to page's growth. In order to distinguish them, we multiply each fraction with $authorship_N(u)$, if the latter is not zero.

4.3. Ranking for a Wikipedia category

The metric that ranks users for a Wikipedia category c is the following:

$$categoryRank_c(u) = \sum_{i=0}^{N} absoluteRank_i(u) / (N + 1 - M)$$

where

- $absoluteRank_i(u)$ is the rank of author u in page i of the category c
- N is the number of Wikipedia pages in the category
- M is the number of Wikipedia pages, which author u has edited

We must remark that the category ranking metric should consider a user's expertise among the various articles that belong to the category. Therefore, a metric that computes for user only the sum of the rank scores for each Wikipedia page in the category is not suitable for qualitative analysis, since it does not take user's knowledge diversity into account. This problem can be solved by dividing the sum with the amount of pages with which user has not interacted via editing. It is obvious that the denominator is incremented by one, in order to cope with the case in which user has taken part in editing all the pages of the Wikipedia category.

4.4. Ranking for top K Wikipedia results

After a search process in the Wikipedia database there is a possibility that more than one Wikipedia page or category might serve as a satisfying result. This situation mainly appears when the search key-words are imprecise or when they do not correspond to an already created page. For this reason, we propose a modification of the ranking metric, considering the top K Wikipedia pages.

In this modification, we should take into consideration the ranking of the results. More specifically, each result has weight $1/k$ (where k is the number of pages used for extracting experts), multiplied with $b/2^{i-1}$, where i denotes the rank of the result and b is a scaling factor. We can consider the search results as probabilities (after a search a user is more interested in the first results, so they have higher probability to be clicked), so it can be assumed that the sum of all weights must be equal to one. Therefore:

$$p_1 + p_2 + ... + p_k = 1 \quad \Rightarrow$$

$$\frac{b}{k} + \frac{b/2^1}{k} + ... + \frac{b/2^{k-1}}{k} = 1 \quad \Rightarrow$$

$$b \cdot \left(1 + 1/2 + ... + 1/2^{k-1}\right) = k \quad \Rightarrow$$

$$b = \frac{k}{\frac{1 - \left(1/2\right)^{k}}{1/2}}$$

Applying the described modification, the metric that ranks users for a Wikipedia search s that returns at least k results is the following:

$$topK - Rank_s(u) = \sum_{i=1}^{k} w(i) \cdot resultRank_i(u)$$

where:

- $resultRank_i(u) = \begin{cases} absoluteRank_i(u) & if \ \ result \ \ i \ \ is \ \ a \ \ Wikipedia \ \ page \\ categoryRank_i(u) & if \ \ result \ \ i \ \ is \ \ a \ \ Wikipedia \ \ category \end{cases}$
- $w(i) = \frac{b/2^{i-1}}{k}$ is the weight of result with rank i

4.5. Ranking synopsis

It is evident that the search results of Wikipedia play a major role in the selection of the appropriate ranking metric. When the search returns only one Wikipedia page, then the Wikipedia page metric is applied in our proposed system. Similarly, when the search result is a Wikipedia category, consisting of pages, then the Wikipedia category metric is used. The general case appears when there are more than one results, where the top-k metric is used.

5. EXPERIMENTAL EVALUATION

5.1. Overview of experimental evaluation

In this chapter, we conduct a series of experiments on sets of articles from the English version of Wikipedia. Our goal is to collect a considerable amount of results, in order to evaluate the performance of our proposed model.

For evaluation purposes, we consider a baseline metric, which is the total addition attribute $add(u)$ of each user in the constructed edit-networks. We name this metric TA Rank (Total Addition Rank). Moreover, we constructed another metric, based on the HITS link analysis algorithm. More specifically, we assumed that each user is recognized by his authority score. There are three main differences between our approach and the traditional algorithm. The first difference is that the original authority score for every user u is the value of total addition $add(u)$ attribute. The second difference is that the authority score of every user u is computed in each iteration step by adding the $restore(\upsilon,u)$ values and subtracting the $delete(\upsilon,u)$ and $undelete(\upsilon,u)$ values by other users υ. Similarly, the hub score of every user u is computed in each iteration step by adding the $restore(u,\upsilon)$ values by other users υ and subtracting the $delete(u,\upsilon)$ and $delete(u,\upsilon)$ values by other users υ. We name this metric HITS Rank.

5.2. Dataset

The dataset used for our experiments was gathered through the export page of Wikipedia (*http://en.wikipedia.org/wiki/Special:Export*). This page takes as input a wiki page or a set of pages and exports the text and the editing history, wrapped in XML format. The produced XML files constitute the input to our system. It is important to note that full history exports are limited to 1000 revisions. However, we are strongly confident to believe that this limitation does not seem to affect largely our system's performance, since this maximum value is suitable for the construction of a reliable edit network.

We used 5 different datasets in our experiments. The first one contains the revisions made on the Wikipedia article *'Energy 52'* (~ 50 revisions). The second one is bigger and contains the revision history of the article *'Jim Gray'* (~ 300 revisions), while the third one is the largest and contains the revision history of article *'Nelson Mandela'* (~1000 revisions). The fourth dataset consists of the revision histories of all articles that belong to the Wikipedia category *'Uninhabited islands of Greece'*. Last but not least, we conducted an experiment under the case, in which Wikipedia returns top-K results in terms with a search key. In particular, we used keywords "Cryptography algorithms" in Wikipedia's search box. We collected the top-10 search results (table 13), which are used for expert extraction.

Table 13: Top-10 Wikipedia results for query "Cryptography algorithms"

TOP-10 WIKIPEDIA RESULTS FOR KEYWORDS "CRYPTOGRAPHY ALGORITHMS"	
1	MD5
2	Block cipher
3	Cryptography

4	RSA (algorithm)
5	Whirlpool (cryptography)
6	MD6
7	Key (cryptography)
8	Cellular message encryption algorithm
9	ElGamal encryption
10	Yarrow algorithm

5.3. Implementation

Our proposed system is implemented in the programming language Java on Netbeans 6.8, a platform framework for Java applications.

5.4. Experiments and results

In tables 14, 15 and 16 we present the top-20 experts for articles “Energy 52”, “Jim Gray” and “Nelson Mandela” respectively. Firstly, it is obvious that TA Rank is not a reliable metric, since it does not show similar behavior with our proposed metric. The reason for this discrepancy is the tendency of Wikipedia users to edit incorrect text written by other users. In other words, writing more text does not necessary grant more expertise in a topic. What does matter though is the absolute consistency of a user’s text, which is represented successfully by the $authorship_N(u)$ attribute and used in our metric. However, this feature is not the only one to be taken into consideration. Our proposed metric also computes how much a user is affected by other users and how much he affects the other through the $totalRevisor(u)/(totalRevisor(u)+totalRevised(u))$ fraction, i.e. his relative consistency. This is the main reason that we did not choose our metric value not to depend solely on $authorship_N(u)$.

Concerning the HITS Rank, we can presume that it is not reliable either. Another obvious disadvantage is that it does not seem to be robust, since there are many users with zero score. This phenomenon is more pronounced in articles with a large number of revisions (table 16). We used several articles with 1000 revisions, in which HITS Rank showed similar behavior. We believe that the main reason for this phenomenon is the fact that HITS is affected more by the outgoing and the ingoing links of a user (represented by the $delete$, $undelete$ and $restore$ attributes) than by the text the user writes.

In tables 17 and 18 we present the top-20 experts for Wikipedia category “Uninhabited islands of Greece” and search keywords “Cryptography algorithms”. We can conclude that the three metrics act differently from each other. The phenomena that appeared in previous datasets are also present in the current ones, for the aforementioned reasons.

Table 14: "Energy 52" Wikipedia article results

ARTICLE		ABSOLUTE RANK		HITS RANK		TA RANK	
		Author	Score	Author	Score	Author	Score
Energy 52	**1**	Dawnseeker2000	31.882502	Alakasam	0.29478595	83.146.62.97	276
	2	Alakasam	27.481482	Yobot	0.0	Nick Arcade	270
	3	WOSlinker	0.5	WOSlinker	0.0	Bobby1011	260
	4	ZéroBot	0.5	ZéroBot	0.0	Tsnosaj	260
	5	Avg	0.5	United states of ecstasy	0.0	Jaknudsen	128
	6	Bluebot	0.5	Aspects	0.0	199.74.81.53	109
	7	Lashuto	0.4848485	Fallschirmjäger	0.0	The Ronin	85
	8	Wickethewok	0.48181817	MelonBot	0.0	Derek R Bullamore	85
	9	Spamdingel	0.31904763	78.86.142.193	0.0	Dawnseeker2000	73
	10	84.168.235.34	0.2881356	MrMPS	0.0	Filipao	72
	11	Energyfiftytwo	0.27272728	Thijs!bot	0.0	Spamdingel	45
	12	199.74.81.53	0.2627551	Cydebot	0.0	Energyfiftytwo	37
	13	FrescoBot	0.26	70.30.65.110	0.0	Alakasam	30
	14	24.187.191.142	0.25	Robert Moore	0.0	Bearcat	28
	15	Bobby1011	0.24757281	Avg	0.0	84.168.235.34	26
	16	Janadore	0.24418604	Flabot	0.0	Janadore	25
	17	Bearcat	0.21186441	Bluebot	0.0	OOODDD	20
	18	Tsnosaj	0.2013889	Bruce1ee	0.0	FrescoBot	12
	19	Jaknudsen	0.1891892	Allen3	0.0	71.202.233.160	11
	20	Filipão	0.17391305	138.38.32.84	-0.079985105	24.187.191.142	2

Table 15: “Jim Gray” Wikipedia article results

ARTICLE		ABSOLUTE RANK		HITS RANK		TA RANK	
		Author	**Score**	**Author**	**Score**	**Author**	**Score**
Jim Gray	**1**	SimonLyall	198.24043	Rillian	0.19323374	CrazyGlu	750
	2	Rillian	96.39685	MrPrada	0.011116079	66.167.48.87	381
	3	CrazyGlu	86.36253	Liujiang	0.0	Gazpacho	355
	4	66.167.48.87	48.20614	EmausBot	0.0	Ken Birman	338
	5	Donnacarnes	41.834846	Jamesscottbrown	0.0	216.231.44.221	283
	6	El chepi	29.025864	Wikinstone	0.0	66.42.13.76	274
	7	Bigmantonyd	26.290337	Δ	0.0	Delirium	259
	8	Neilc	14.295767	Duncan.Hull	0.0	Donnacarnes	242
	9	86.44.134.208	11.510047	92.112.109.126	0.0	SimonLyall	219
	10	64.140.251.226	9.820147	64.140.115.190	0.0	Sakhalinrf	198
	11	192.38.109.188	6.4727273	RjwilmsiBot	0.0	El chepi	182
	12	99.140.214.16	2.0	ArthurBot	0.0	Peterhoneyman	179
	13	MrPrada	1.5213474	131.107.0.77	0.0	Jokestress	165
	14	Mikeblas	0.5	Why why why why why	0.0	Neilc	125
	15	EmausBot	0.5	131.107.0.73	0.0	64.140.251.226	111
	16	92.112.109.126	0.5	Jpbowen	0.0	Alan smithee	105
	17	RjwilmsiBot	0.5	58.11.71.169	0.0	Psantora	103
	18	ArthurBot	0.5	Igbrown	0.0	72.152.112.158	102
	19	131.107.0.73	0.5	Jgemmell	0.0	Rillian	101
	20	Rilak	0.5	Valepert	0.0	Robert Merkel	99

Table 16: “Nelson Mandela” Wikipedia article results

ARTICLE		ABSOLUTE RANK		HITS RANK		TA RANK	
		Author	**Score**	**Author**	**Score**	**Author**	**Score**
Nelson Mandela	**1**	Dewet	120.157715	Jdavidb	0.0	152.163.101.8	3723
	2	Gurubrahma	60.623634	Phase1	0.0	Jezzyka	3345
	3	Jezzyka	54.756195	MONGO	0.0	198.54.202.2	2258
	4	70.19.47.217	52.641006	213.166.17.25	0.0	Dewet	2524
	5	210.9.138.5	38.48577	Ravidreams	0.0	Sam Francis	2496
	6	Ferdinand Pienaar	25.283834	65.162.60.101	0.0	Ezeu	1972
	7	DJ Clayworth	23.817362	JdforresterBot	0.0	217.139.59.213	1595
	8	Magister Mathematicae	17.93932	Lt-wiki-bot	0.0	Vzbs34	1528
	9	68.82.115.29	17.620797	Paul August	0.0	157.161.45.137	1329
	10	198.54.202.242	17.01138	JoanneB	0.0	203.129.33.225	1284
	11	Carolynparrishfan	16.801199	TheRingess	0.0	62.171.194.40	1278
	12	192.30.202.13	15.337795	68.111.39.112	0.0	209.232.158.20	1239
	13	209.192.83.3	15.033075	Doc glasgow	0.0	Mav	1216
	14	Elf-friend	14.777694	FranksValli	0.0	62.171.194.8	1121
	15	217.139.59.213	13.355627	FreplySpang	0.0	168.209.98.35	1095
	16	Swissjames	13.355627	195.209.85.3	0.0	Elf-friend	1089
	17	Canuckguy	13.257783	Aude	0.0	Wizzy	1072
	18	Hottentot	12.930892	195.252.67.254	0.0	Kingal86	1061
	19	Benw	10.122368	YurikBot	0.0	24.91.225.143	1033
	20	Ezeu	9.865271	159.101.45.161	0.0	Nirvana2013	1010

Table 17: “Uninhabited Islands of Greece” Wikipedia category results

CATEGORY		ABSOLUTE RANK		HITS RANK		TA RANK	
		Author	Score	Author	Score	Author	Score
Uninhabited Islands of Greece	1	Nipsonanomhmata	90.81684	PKT	0. 0595076	Nipsonanomhmata	3723
	2	Cplakidas	12.8361025	Cydebot	0.04906508	LukasPietsch	3345
	3	Future Perfect at Sunrise	6.400086	Download	0.014976065	Khoikhoi	2258
	4	Wicki2009	4.5121775	EmausBot	0.009443346	Future Perfect at Sunrise	2524
	5	Download	3.47191	OgreBot	0.00887009	Cplakidas	2496
	6	Nefasdicere	2.1322043	66.41.70.118	0.006758097	213.46.217.102	1972
	7	131.111.185.88	2.1056244	Krenakarore	0.0060922736	Wicki2009	1595
	8	Cydebot	1.8255646	Elkost	0.0060363133	Letus	1528
	9	LukasPietsch	1.6467681	Lightbot	0.004483544	78.172.23.84	1329
	10	El Greco	1.1178061	128.130.115.5	0.0029132704	Hittit	1284
	11	Woohookitty	0.9098342	WinstonSmith147	0.0022103253	122.100.160.7	1278
	12	WinstonSmith147	0.73229825	Volcanoguy	0.0014489385	Nefasdicere	1239
	13	Elkost	0.7303371	32X	0.0013967352	El Greco	1216
	14	201.171.196.176	0.59354395	Ferengi	4.1748714E-4	Magioladitis	1121
	15	Pumpie	0.5879809	145.94.72.203	1.8939257E-4	Baristarim	1095
	16	Khoikhoi	0.53919077	Winner 42	6.187125E-5	Politis	1089
	17	99.140.177.247	0.5060978	ArgGeo	0.0	Leandros	1072
	18	EmausBot	0.45730516	98.154.22.168	0.0	83.79.139.237	1061
	19	128.130.115.5	0.42897594	C messier	0.0	Pumpie	1033
	20	Letus	0.42897594	Ashershow1	0.0	Dimadick	1010

Table 18: “Cryptography algorithms” search key results

SEARCH KEY		ABSOLUTE RANK		HITS RANK		TA RANK	
		Author	**Score**	**Author**	**Score**	**Author**	**Score**
Cryptography Algorithms	**1**	Mitch Ames	11040.263	RNAasaurus	0.9762708	Matt Crypto	680539.8
	2	Nageh	6947.584	Stybn	0.29132044	Ww	519213.78
	3	Mesoderm	5968.5503	134.58.253.57	0.19421363	210.2.171.210	443633.22
	4	Ww	5917.803	CWii	0.18065235	Nageh	440330.0
	5	141.154.216.98	3221.48	Skintigh	0.05833783	Mitch Ames	440330.0
	6	Mangojuice	2986.8254	74.112.174.10	0.04566767	Mesoderm	290708.88
	7	Matt Crypto	2787.7688	130.215.29.46	0.022667043	Mangojuice	240559.9
	8	SCCC	2180.5955	TreasuryTag	0.020474833	Myria	181026.78
	9	Phr	2015.0903	Angela	0.0	Phr	129626.586
	10	Cyde	1397.2327	Rast	0.0	70.53.126.44	123395.49
	11	Anna512	973.4387	62.16.227.130	0.0	202.177.155.148	111058.45
	12	71.80.26.81	952.41376	62.218.66.106	0.0	85.100.30.63	106103.61
	13	Bryan Derksen	699.3801	Splintercellguy	0.0	163.151.0.253	104502.05
	14	198.161.246.2	575.56	Royboycrashfan	0.0	Davidgothberg	101991.2
	15	Feezo	501.53336	Porges	0.0	66.214.162.103	101804.875
	16	129.54.8.45	498.48392	Luís Felipe Braga	0.0	81.215.239.164	101098.73
	17	Pgan002	459.45764	Furrykef	0.0	Feezo	99246.914
	18	GDallimore	450.4865	Alexav8	0.0	Intgr	92261.97
	19	Wavelength	410.65506	David Eppstein	0.0	58.107.15.178	79327.47
	20	SDC	391.3945	Dchristle	0.0	Oli Filth	77926.09

6. CONCLUSION AND FUTURE WORK

In this thesis we presented a simple and flexible expert extraction system for the Wikipedia users. Our proposed system is largely based on the concept of the edit network, a graph structure that represents the interactions between users for a Wikipedia page. The edit network is enriched with a set of attributes, which helps us define a reliable metric for ranking users according to their expertise.

The main advantages of our system are ease of implementation and scalability. It can be extended in order to support expert extraction for more general topics than the topic, in which experts are explicitly requested through a search query. This functionality is feasible if we use the categorization model of Wikipedia, in which each Wikipedia category is a subcategory of at least another Wikipedia category (except from the top level category). We plan to implement this feature in future work. Another promising avenue to follow is to use semantic analysis for better search results in the query processing module. The default search feature of Wikipedia does not always provide the best results for a query. Therefore, the adaptation of semantic analysis into the search query could improve the quality of the results, and thus the performance of our system.

Last but not least, we should remark that our approach can also be applied to other systems that maintain revision history in XML format. A perfect example is Discogs [26], a website and database of information about audio recordings, including commercial releases, promotional releases, and bootleg or off-label releases. More precisely, each release is represented like a Wikipedia page, i.e. it maintains the edits that have previously been applied to it. The revision history of releases can be used to construct the edit network, and thus extract attributes that help us rank users in terms of expertise into a single release, a music project, or even a music genre.

GLOSSARY

3RR	Three Revert Rule
HITS	Hyperlink-Induced Topic Search
POS	Part-Of-Speech
TA	Total Addition
UTC	Coordinated Universal Time
WEM	Wikipedia Expert Miner
XML	Extensible Markup Language

REFERENCES

[1] Wikipedia, http://www.wikipedia.org/

[2] Jim Giles, “Internet encyclopedias go head to head”, Nature Journal, December, pg. 900, 2005.

[3] Nupedia, http://www.nupedia.org/

[4] Open Content License, http://opencontent.org/opl.shtml

[5] GNU Project, http://www.gnu.org/

[6] Encyclopædia Britannica, http://www.britannica.com/

[7] Roy Rosenzweig, “Can History Be Open Source? Wikipedia and the Future of the Past”, the Journal of American History 93, June 2006, pg. 117–146.

[8] Larry Press, “Survey of Wikipedia accuracy and completeness”, in California State University, Los Angeles, 2006.

[9] Stuart Andrews, “Wikipedia Uncovered”, PC Authority magazine, 2007.

[10] Simon Williams, “Wikipedia Vs Encyclopedia: A Question of Trust? Are online resources reliable or should we stick to traditional encyclopedias”, PC Plus Magazine, issue 268, April 21, 2008.

[11] Alexander Halavais, Derek Lackaff, “An Analysis of Topical Coverage of Wikipedia”, Journal of Computer-Mediated Communication 13, 2008, pp. 429-440.

[12] Ba-Quy Vuong, Ee-Peng Lim, Aixin Sun, Minh-Tam Le, Kuiyu Chang, “On Ranking Controversies in Wikipedia: Models and Evaluation”, In Proc. of the 1st ACM International Conference on Web Search and Data Mining (WSDM 2008), Stanford, CA, USA, Feb 2008.

[13] Bongwon Suh, Ed H. Chi, Bryan A. Pendleton, Aniket Kittur, “Us vs. Them: Understanding Social Dynamics in Wikipedia with Revert Graph Visualizations” IEEE Symposium on Visual Analytics Science and Technology (VAST ‘07), pp. 163 – 170, 2007.

[14] Aniket Kittur, Bongwon Suh, Bryan A. Pendleton, Ed H. Chi, “He says, she says: Conflict and Coordination in Wikipedia”, In Proc. of the SIGCHI conference on Human factors in computing systems, ACM, New York, NY, USA, pp 453 – 462, 2007.

[15] Fernanda B. Viégas, Martin Wattenberg, Jesse Kriss, Frank van Ham, “Talk before you type: Coordination in Wikipedia”, In Proc. of the 40th Hawaii International Conference on System Sciences, vol. 40, 2007.

[16] Ulrik Brandes, Jürgen Lerner, “Revision and Co-revision in Wikipedia”, Proc. Intl. Workshop Bridging the Gap Between Semantic Web and Web 2.0, 4th Europ. Semantic Web Conf. (ESWC ‘07), 2007.

[17] Ulrik Brandes, Patrick Kenis, Jürgen Lerner, Denise van Raaij, “Is Editing More Rewarding than Discussion? A Statistical Framework to Estimate Causes of Dropout from Wikipedia”, Proc. 1st Intl. Workshop Motivation and Incentives on the Web (Webcentives '09, co-located with WWW2009), 2009.

[18] Gianluca Demartini, “Finding Experts Using Wikipedia”, In Proceedings of the Workshop on Finding Experts on the Web with Semantics, 2007.

[19] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd, “The PageRank Citation Algorithm: Bringing Order to the Web”, Technical Report, Stanford University, Stanford, CA, 1998.

[20] John Kleinberg, “Authoritative Sources in a Hyperlinked Environment”, Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.

[21] Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum, “Yago: A Core of Semantic Knowledge”, In Proc. Of the 16th International Conf. on World Wide Web, pages 697-706, 2007.

[22] Giovanni Semerano, Marco Degemmis, Pasquale Lops, Pierpaolo Basile, “Combining Learning and Word Sense Disambiguation for Intelligent User Profiling”, Twentieth International Joint Conference on Artificial Intelligence, 2007.

[23] Kai-Hsiang Yang, Chu-Yu Chen, Hahn-Ming Lee, Jan-Ming Ho, “EFS: Expert Finding System Finding System Based on Wikipedia Link Pattern Analysis”, In Proc. of the 2008 IEEE International Conference on Systems, Man and Cybernetics (SMC), Singapore pp. 12 – 15, October 2008.

[24] Ulrik Brandes, Patrick Kenis, Jürgen Lerner, Denise van Raaij, “Network Analysis of Collaboration structure in Wikipedia”, In Proc. 18th Int. World Wide Web Conf. (WWW2009), 2009.

[25] Ulrik Brandes, Patrick Kenis, Jürgen Lerner, Denise van Raaij, “Computing Wikipedia Edit-Networks”, 2009.

[26] Discogs, http://www.discogs.com/