Sveučilište u Zagrebu

FAKULTET ORGANIZACIJE I INFORMATIKE

Jurica Ševa

WEB NEWS PORTAL CONTENT PERSONALIZATION USING INFORMATION EXTRACTION TECHNIQUES AND WEIGHTED VORONOI DIAGRAMS

- DOKTORSKI RAD -

Varaždin, 2014

To Ilona: A single brushstroke can change the entire canvas. <3

Acknowledgements

The author would like to acknowledge the immense help of his mentors Prof. Mirko Maleković, PhD and Asst. Prof. Markus Schatten, PhD. Without them, this dissertation would probably have never been finished. Also, a big thank you goes to Prof. Franciska de Jong for her good spirit and helpful conversation and comments that made this dissertation a better one. Special thanks go to Prof. Robert Proctor, Prof. Stephen J Elliott and the entire BSPL Laboratory; their help proved to be invaluable for the initial steps in this entire endeavor. Also, the author would like to acknowledge the Fulbright Scholar Program as well as Croatian Science Foundation for giving him the chance to spend time focused solely on this research. Additionally, a great thank should be given here to a friend and colleague Tonimir Kišasondi, PhD, as well as the entire OSS Laboratory team; thank you for the discussions as well as the infrastructure without which this work would have never been finished. The author would also like to thank Prof. Dragutin Kermek for his understanding, advice and support. The author would especially like to thank his parents Dragoslav and Marija as well as his brother Tomislav and his wife Ivana for their understanding and support. They brought a lot of joy into his life! The last, but greatest thank you goes to Ilona. This dissertation is dedicated to you and your infinite patience.

Jurica Ševa

Appendix A List of figures

Appendix B List of equalities

Appendix C List of tables

TABLE OF CONTENTS

PART I: BONES

1. Introduction

The last decade has witnessed an explosion of information accessible through online information resources. According to [1], there are 295 Exabytes of data accessible through the Web interface. The amount of the available information leads to the absurdity of information crisis: finding information is no longer a problem because of its scarcity or lack of access; to the contrary, the quantity of available information leads to frequent problems in finding the information that is needed. The problem of information overproduction was first noticed in the early sixties of the twentieth century. This is also when the first efforts were made in solving the problem of accessing the requested information. These efforts were performed parallel in research domains, but with a common goal: to design computational models for handling the natural language to allow better information organization, clustering, storage, search and access. An overview of this research field, in the scope needed for this dissertation and the work presented in it, is given in section 3.2 and its subsections.

One of the key research efforts in the field of information extraction, SMART [2, p. 61] was an information extraction system, the results of which include many important concepts such as vector space modeling, Rocchio classification and more. One way of solving problems caused by information overproduction is through information personalization. This can be achieved by creating virtual user profiles based on the analysis of the fundamental (individual or clustered) user behavioral characteristics. Personalization can, in this context, be defined as a way of distinguishing (IT) needs of different users. Personalization is mostly used in the field of information retrieval ([3]–[14]), as the attempts of specific domain personalization are infrequent, but do exist. One should mention research efforts in creating user profiles ([11], [13], [15]–[18]). The literature shows that the problem of information personalization is an interdisciplinary field which combines the research efforts from fields such as artificial intelligence, information extraction, data mining, statistics, natural language processing and others. This work gives the essential techniques of extracting information from unstructured texts as well as an overview of the performance evaluation methods (and their accuracy).

Furthermore, the dissertation also provides an introduction to the basics of Voronoi diagrams, whose use is proposed as the personalization function, as it has not been reported so far in this context (to the author's best knowledge at the time of writing this dissertation). Voronoi diagrams present a mathematical formulation that allows the division of an n dimensional space into regions (called cells) with every cell created around a specified cell generator. The

generalization of Voronoi diagrams are of interest in this work as they accept correction parameters in determining to which generator (and subsequently cell) a single point belongs to (based on the previously chosen distance functions, e.g. Euclidean distance). A more formalized definition is given in chapter 5 and its (sub)sections.

The definition of information crisis states that the problem of getting or accessing the needed information does not lie in the fact that information is inaccessible but just the opposite; the vast number of information that users are surrounded with, makes it very hard (if not almost impossible for the inexperienced cybernaut[1]) to access them in desired time. Ever since IT technologies, with the development of the first personal computers, escaped the scientific laboratories, the number of information has been growing in a very fast way. This exceptionally fast growing rate has contributed to the enlargement of entropy (following the pattern of everything else in our universe) but this entropy is in direct collision with the needs of the modern man.

This discrepancy between the needs and the findings has best been seen in the last two decades with the development of the World Wide Web that made it possible to create, share and use information in the most creative ways. The Information has become the driving force of humanity and new concepts on how to use and reuse the same information became available. The first efforts that have laid the foundation stone of the Information Revolution with the invention of HTML markup language as a way of structuring, organizing and presenting information in the digital world, have slowly become extinct and are becoming replaced with the steps of evolution from simple Web1.0 practices to today's Web2.0 and tomorrow's Web3.0. The important question here is – are users still the driving force of this change or are they mere subjects to its own will?

And so, one comes to the problem at hand, the problem of meaning. If one wants the next step in the evolution to happen, one has to look at the meaning of information, slowly pushing users from the presentation layers to the higher levels that will allow them to see the objects and subjects that collaborate in the exchange of information. Although computational semantics is a field that started gaining in importance slowly in the last decades, it is only now, with the information explosion, that it has become really important. There are a lot of

[1] According to the free dictionary [124], a cybernaut is a computer user who uses the internet; someone who explores cyberspace

researchers working on the answer to one question: how does one make computers understand?

The problem with computational semantics is that it is a very multidisciplinary field and requires the input from experts from the field of philosophy, linguistics, computer science, mathematics, psychology and others, and that directly influences the pace of development. Currently, there are a couple of approaches to the question of how to derive meaning, with natural language processing being one of them. It starts with the question – how to translate a sentence in a way that the computer (meaning the underlying algorithms that make everything work) knows what version, of all the possible meanings a sentence can have, the user meant.

Content personalization available via digital resources has become an important research area with the beginning of content digitalization. The development of hypertext as the presentational level in the online environment has enhanced the production, availability as well as research efforts in the structuring, organization and personalization of digital content. Until now, several research directions have been developed to obtain a deeper knowledge from unstructured text in order to filter the available content. The goal of these efforts is to achieve greater accessibility and utilization of available resources. The methods of personalization can be divided into three main categories, depending on the method of filtering [19, p. 4]:

- content-based filtering with recommendations based on the previously visited content,
- collaborative filtering in which user groups with the same or sufficiently similar interests are formed and any recommendations of new content are based on the discovered group preferences, and
- rule-based filtering where recommendations are based on the answers given by the user.

The process of personalization is done in four phases [20]:

- data collection (e.g. through Web server logs),
- preprocessing of the collected data (consisting of data cleaning),
- data analysis (e.g. converting a text into a document-term matrix),
- recommending based on the results obtained during the analysis.

The presentation of textual content online is done using HTML structural language that divides the document into two parts: the descriptive part (a container for the information processing and metadata for an HTML document) and the content part (a container for the displayable content of an HTML document). The descriptive part of the document (e.g. META tags defined in the HTML language HEAD element) offers the possibility of giving an additional description to the document through the help of predefined tags, but does not give the opportunity to define the semantic value[2] of the document's content. The structure of HTML used as a presentation layer, available to format the content presented in a Web page, adds to the difficult task at hand. Due to the diversity of HTML structure used in presenting the information via the WWW (and additionally expanding with the use of CSS[3]) some additional steps in detecting and extracting the content parts are needed.

This chapter will give a brief overview of the state of the art as well as the motivation which led to this research and subsequently to this dissertation. The history of information retrieval will be given with focus on research milestones that defined the theoretical and practical foundations of this scientific field. Also, a brief overview of the research problem will be given in this chapter's only subsection.

One way of solving problems of information overproduction is the personalization of the information source, in our case, the WWW environment, by creating virtual profiles based on the analysis of user behavioral characteristics, with the goal of assigning importance values to information nodes[4] on an individual basis (1:1 personalization). The personalization is mostly used in the field of information retrieval ([3], [4], [8], [11]). In the review of the previous research, a few different approaches that were used in the personalization of the available content should be noted: ontological approaches ([4],[11]), contextual models ([21], [22]), and data mining ([23]–[25]). These approaches are most common in the reviewed literature.

---

[2] According to [125] semantic values "are entities assigned to expressions by theories in order to account for semantic features of languages, such as truth conditions and inferential connections"

[3] Cascading Style Sheets

[4] An HTML document with unstructured textual content accesible over the Internet (in our case, available through the selected web news portal)

For example, [26] deals with news personalization through user attitudes towards personalization technology and mentions several challenges in addressing news content personalization:

- *Conflicting reading objectives* where the focused information search conflicts with leisure news reading in the scope of personalized/recommended articles
- *Difficulty of filtering information to fit user interests* for various reasons, e.g. change of interests, differentiation between short and long term interests
- *Ways to generate the user profile* with different approaches of acquiring a set of user preferences (explicit vs. implicit user information accumulation)
- *Novelty of the information* with focus on differentiation between the available information known to users and the information that is either old and unknown to users or new and unknown to users
- *Depth of personalization* and the effect the personalization depth has on filtered and presented news items

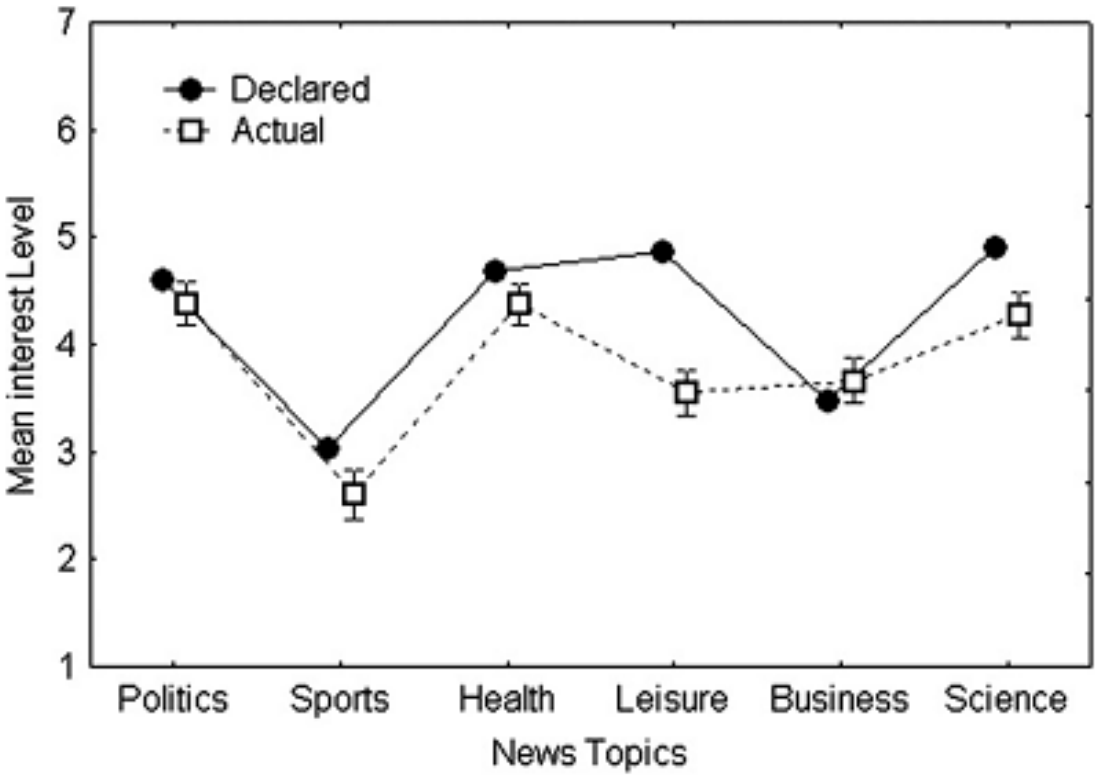

Figure 1: Declared vs. actual reading interests[26]

A study was organized with 117 participants (aged 20 to 70) with heterogenic backgrounds and with two steps. First, they were asked to do a questionnaire which was to collect data about their preferences. Their answers were compared to actual user interests collected via their reading habits. The results of the study are presented in Figure 1. They also state that "users have positive attitudes towards news personalization and are interested in receiving personalized news" [26, p. 492] and that "news personalization is beneficial by showing that users differ in their interest in different topics" [26, p. 492].

The literature review has also provided an insight to the lack of uniform term taxonomy used for annotation of information nodes (e.g. HTML Web pages of observed news portal(s)). The prevailing approach is to use annotations marking system based on user input. Heymann in [27, p. 54] states that "annotation practices are similar across systems for the most popular tags of an object, but often less so for less common tags for that object", the problem of synonyms is negligible with sufficient data [27, p. 53] and that annotations that are used by ordinary users and domain experts overlap in 52% of cases [27, p. 57]. Tagging systems carry with them a large amount of "information noise" due to the nature of individual information node identification that is directly tied to the user's domain knowledge of IT facilities.

As a potential solution to this perceived lack of uniformity between tagging systems, the use of existing taxonomies defined by the available Web directories has been recognized. A literature review of several available Web directories shows that the most mentioned and suitable Web directory is the ODP[5] Web directory as it provides the best hierarchical domain classification of information nodes. The use of ODP as a taxonomy is proposed in several papers ([4], [22], [27], [28]). Using the ODP taxonomy for the classification of information nodes allows also the determination of the domain membership[6]. This fact offers the definition of an information node through a membership value to specific domain(s) of the proposed Web directory. Given the complex structure of the ODP taxonomy (12 hierarchical levels of division, 17 categories at the first level) and a large number of potential categories, [28] suggests to use the ODP taxonomy for the classification of information nodes to level 6 [29]. With reference to the number of hierarchical levels that are recommended for use in analyzing the structure of the ODP, [28] also emphasizes the need for deep classification of

[5] Open Directory Project

[6] One or more ODP categories in to which an information node can be classified during the classification phase

documents. Other studies also state that the problem of personalization was primarily approached in the domain of information retrieval through a Web interface (web search) and that the personalization of the information available via Web portals is poorly understood and researched.

Expanding on the previously mentioned papers with [23] and [30], different possible data sources were identified: server log files, browsing history log files, clickstream applications, cookies and other. Data collected through one or more of these sources provide an insight into an individual user's movement within a defined time frame and information nodes (which can be used to define the user's navigational pattern as a directed node graph). In the reviewed literature, data collected in this fashion is used for information personalization based on the similar user's group and not at the user's individual level. This work approaches the problem of content personalization, in the domain of web news portals, via the individual approach where the available content is tailored according to previous user browsing preferences and knowledge models that are created based on the browsed data. The personalization itself is also content based and employs an analysis of visited content for the purposes of defining a set of recommendable content.

1.1 Research theme

The aim of this work is to test the existing methods, which are identified to be of importance for future steps (and presented in full in the following chapters of this work), and to improve these methods with the help of weighted Voronoi diagrams. The purpose of using Voronoi diagrams in this process is to achieve personalization at the individual level, differing from the standard clustering approaches. Using weighted Voronoi diagrams for personalization has so far not been reported in the literature and therefore represents a novel approach to the process of information personalization. Weighted Voronoi diagrams are suitable for this purpose as they allow for standardized distance metrics expanded with personalization parameters to be used.

The existence of a behavioral pattern associated with long-term and/or short-term data on the user's movement through information space provides for better filtering and personalization of the available information and is, among others, dealt with in [8], [23], [24]. Considering that the aim of this work is to show the ability to individual personalization, the potential use of weighted Voronoi diagram for the construction of virtual semantic profiles and

personalization information has been recognized. In the area of personalization and division of information space into logical units, Voronoi diagrams are used in [31], [32].

During the literature review phase several research problems were identified as being of interest to this work and they will be tackled in the remainder of this dissertation. The need for unified classification taxonomy was identified as, to the author's best knowledge, such a research effort yet to be realized. Due to the ODP's properties, it was selected as the appropriate basis for such classification. The work on Web news portals and the information available on them was also scarce, which led to the decision to focus the research domain on Web news portals. The practical application of this work will be presented in the future work but briefly described here; the practical idea is to allow news content accumulation from various sources and to create an individual user based personalization. Research efforts in the domain of personalization and/or recommendation systems were focused on clustering similar users and their preferences/browsing history data in clusters and on presenting personalization/recommendation for each cluster of users. This work takes the step towards a truly individual personalization/recommendation by using single user's browsing history data and recommending unseen/not recommended news articles. The generalization of Voronoi diagrams, weighted Voronoi diagrams, offers a way for such a personalization.

2. Detailed research proposal

This chapter will focus on the work presented in this dissertation and the methodology used during data collection and preparation phase. Also, it will define the goals as well as the hypothesis of this research along with the proposed personalization system overview.

The focus of this work is the possible use of the presentation part of each HTML document available through the selected news portals for the purposes of:

- defining the semantic value of the thematic domains and the value function of document attachment to one or more domains
- creating a hierarchical document attachment to the thematic domain and its sub domains
- implementation of the above generated hierarchies along with the attachment value function for the purposes of personalization of (newly) published content in an online environment
- classification of thematic units of interest to the individual user along with the personalized hierarchical presentation of newly published content.

Previous research agendas of content personalization are numerous and extending to several decades of concentrated research. Researches related to the personalization of online content fully began in the early nineties of the twentieth century. Despite numerous related works it can still be considered in its infancy because of the different approaches and a non-unified approach to content structure and presentation. Thus the problem of defining and creating user profiles (among others) is dealt with in [13], [15], [17], [20], [33], [34]. The personalization of online search results is dealt with, among others, in [3], [8], [11], [13], [14]. From the point of view of the applied methods of data analysis for profiling and/or definition of navigation/behavioral patterns one can mention the work presented in [24], [35]–[39]. The automatic classification/categorization of Web sites is dealt with in [40]–[45].

The semantic personalization (content based personalization) is determined by the quality of the categorization of unstructured text. The categorization is achieved by evaluating the content of online resources, and assigning a value function that projects each available document to one or more of the possible categories (thematic units). The development of the WWW environment has started efforts to create classifications of published and available content, primarily for the purposes of finding and accessing the required resources. For this

purpose, one of the first ways of organizing the available content was with the use of Web directories. Web directory is a way of organizing and categorizing available, online accessible content into a hierarchy or interconnected list of categories. In the meantime, the effective PageRank algorithm, presented in [46], was developed as the most popular and effective way of online information search. There are a number of Web directories that provide possible content classification taxonomies (Yahoo! Directory, World Wide Web Virtual Library, Starting Point Directory and others). ODP is the largest Web directory with open and public facilities and was until recently used as a part of the Google Directory. This system gives the possibility to use multi-level categorization, and thus standardization of content distribution in predefined categories.

2.1 Information sources

Previous studies in the field of information extraction techniques had as a result a collection of documents gathered through research efforts. The collections of documents are used as training data for new approaches in solving some of the current problems (content based information personalization being one of them) and offer a platform for the proposed solution performance testing (possible testing measures were mentioned in the previous chapters). The most famous collection of data so far is the TREC[7] collection with the collection itself being divided into data packages focused on a specific problem domain. Since 2011, an additional competition is available, called Web Track, which focuses on the task of improving the online information search task. Because of its size and the diversity of the collected data, this collection is most interesting for researchers in the field of information extraction/information retrieval.

Besides the TREC collection of documents, there are other collections like Time collection (collection of published categories in TIME Magazine in 1963), Cranfield collection (1,400 abstracts), Medlars collection (1,033 abstracts), Reuters-21,578 Text Categorization Test Collection (Reuters newswire published in 1987 with 21,578 documents), The 4 Universities Collection (WWW pages collected in January 1997 by World Wide Knowledge Base project at Carnegie Mellon University, 8,282 pages classified into seven categories).

[7] Text REtrieval Conference

2.2 Data Collection

Apart from using the already available collections, it is possible to create your own collection of documents with the help of Web scraping techniques, whose task is to produce copies of WWW accessible pages. This allows us to create an "image" for subsequent analysis. For the purposes of this research, a project was created in collaboration with Purdue University researchers Prof. Robert Proctor and Prof. Stephen Elliott during which, in experimental settings, data collection took place. Recognized hypothesis will be tested on the collected data. Data collection had 20 participants of different gender, age, origin and level of education. The goal of data collection was to capture data of both U/I usage as well as content-based interest of the participants in the domain of news portals. For this purpose, www.cnn.com portal was chosen as a representative system for several reasons:

- relatively quick content update
- standardized data structure and data display as well as
- coverage of information from a large spectrum of human interests providing opportunities to identify topics of interest for each participant.

During the data collection, 200 sessions of browsing data were collected, with each session lasting for 30 minutes, divided into 20 participants. The result of the research is insight into the history of the movement of each individual participant in a given portal with information on the use of standard I/O devices (keyboard and mouse). Data was collected via two tools: Mozilla Firefox SQLite file with individual participant browsing history (places.sqlite) as well as uLog Lite clickstream application.

The file places.sqlite is a SQLite file and consists of the following tables:

- moz_anno_attributes
- moz_annos
- moz_bookmarks
- moz_bookmarks_roots
- moz_favicons
- moz_historyvisits
- moz_inputhistory
- moz_items_annos
- moz_keywords
- moz_places

with the table data moz_places containing a browsing history and the table moz_historyvisits providing additional descriptive information.

uLog Lite system is a clickstream application that gathers information about user interaction with the computer based on the action with the U/I devices. The available data, accessible through CVS[8] files, is:

- Date - current date
- Time - current time (format hh: mm: ss)
- msec - current time in milliseconds
- Application - active application
- Window - the name of the active window
- Message - Event
- X - x-coordinates of pressure / release of the mouse button (in pixels)
- Y - y-coordinates of pressure / release of the mouse button (in pixels)
- Relative distance - the shortest distance between the pressure and release of the mouse button
- Total distance - actual distance between the pressure and release of the mouse button
- Rate - Ratio of distance: Total distance (a number between 0 and 1)
- Extra info - additional information

2.3 Data preparation

The preparation of the collected data was conducted in several phases:

- SQLite file content extraction
- CSV file content extraction
- Parse HTML structure to draw clear text

The files used in data collection had a common naming scheme to differentiate individual users, and furthermore, to distinguish each user session data. The naming scheme used in the study was shaped IDKorisnika_RedniBrojSesije, where IDKorisnika (possible values 1-20) identifies the user and RedniBrojSesije identifies the session number (1-10). The naming

[8] Comma-separated values

scheme is necessary to distinguish between individual users as well as to distinguish between individual user sessions. For the purposes of data preparation, a MySQL database was created in which, in the designated table, the original processed data was stored for further analysis.

2.3.1 Content extraction from individual places.sqlite file

The content of the SQLite file is prepared using a modified version of ff3histview[9]. The structure of the obtained data provides an insight into the visited Web sites (identified by a URL[10] address), date and time of the visit and the order of visited information nodes (thus providing the insight into the directed navigation graph). Thus, a complete insight into the user's movement within the online system is obtained.

2.3.2 Content extraction from CSV uLog Lite files

A file was parsed and extracted from the content of the uLog Lite files using a PHP script. The processed data gives information about user focus areas through his/her interaction with the I/O devices (keyboard, mouse), information about the current active application, the descriptive data for the I/O devices (mentioned above) and millisecond based time data for each, user invoked, event.

2.3.3 HTML document content extraction through HTML structure parsing

For each recognized valid visited online site, the site's content was extracted with focus on the content part itself (disregarding the various menus, related sites and other additional data). For this purpose, once more, a PHP script was developed that, based on the CSS class name (manually identified before), targeted only specific parts of the document. The extracted data is stored in a MySQL database, to its respective table. This data provides the basis for further analysis.

[9] http://blog.kiddaland.net/dw/ff3histview

[10] Uniform Resource Locator

2.4 Proposed personalization model in the domain of Web news portal(s)

The proposed model is based on the analysis of the work done in the areas of online resources personalization, information extraction techniques and areas of information visualization, more specifically weighted Voronoi diagrams.

2.4.1 Definition of words that define thematic units through information extraction techniques

The personalization of content essentially reflects the ability of the content classification into thematic units (information domains). A single thematic unit is comprised of content between which, to a greater or lesser extent, there is a semantic relationship/similarity. The semantic connection between the two available online resources assumes that a user will be interested in both resources (but not necessarily to the same extent). It is vital for the processes of personalization to find resources that are semantically related.

One approach to the classification of resources themselves as well as the connection between them is to use a predefined classification taxonomy. The above mentioned Web directories are one of the possible forms of such taxonomy and can be used in creating a universal, purely content based annotating scheme for the purposes of automatic annotation systems. Through such automatic annotation of the classification of the relevant thematic units, it is possible to create semantic links between all the available resources at the time of their creation.

This research recognized the potential of using information extraction techniques for the analysis of document content from unstructured text. tf-idf[11] method was identified as one of the two methods, whose functionality meets the needs of this research. The result of these analysis techniques is presented in chapter 2 of this dissertation.

From all possible and available classification taxonomies, ODP Web directory is identified as the most suitable for further work as it offers a hierarchical categorization scheme of thematic units. Through tf-idf analysis of resources that describe each of these categories, one can define a specific category term-document matrix that can be used for further implementation through the weighting values that describe a set of words or terms that best describe each category. As the ODP presents a hierarchical taxonomy, some overlapping matrices are expected. The justification for using the hierarchical structure of thematic units created in the

[11] Term frequency – inverse document frequency

ODP is given in [28], [47]–[50]. Based on the weighted values and applying the same techniques to the analysis of newly published online content, one can enable the automatic categorization of content.

NLTK[12] framework was selected for the implementation of thematic content analysis with the mentioned tf-idf technique. It is used primarily for modeling vector space, which is the basis of information extraction technique(s) used in this research.

2.5 Creating a virtual profile based on the contextual sequence of thematic units

The focus of the proposed model is the creation of a new system of content personalization in the online environment. The target groups are online systems that are classified as news Web portals. A Web portal is an online system that serves as a point of access to information in the WWW environment. Its task is the presentation of content from different sources and different thematic units. The presentation of content is done by using HTML language and is arbitrary for each of the portals. Heterogeneity of the presentation layer makes the data aggregation from multiple sources difficult. During the data collection, the information posted on the Web site www.cnn.com was collected as the only information source. The test system, as one of the results of this study, consists of the information from more than one news Web portals.

The basis of the proposed system to personalize the information is the creation of a virtual profile for each individual system user. Aside from the possible user's demographic data (provided by the user himself) the virtual profile consists of the semantic part that represents topics of interest for each individual user. In the proposed system, each thematic unit has a weight measure attached, based on the user's individual interests. Definition and creation of thematic units is given in the previous chapter.

This work proposes the creation of virtual profiles based on weighted Voronoi diagrams. Weighted Voronoi diagrams are, in the proposed system, used for the classification of users' interests during their browsing sessions. The Voronoi diagrams are created on the basis of the identified thematic units the users visited during their past browsing sections. This model allows the personalization of 1:1 instead of the current approach 1: N where the mechanism of

[12] Natural Language Toolkit

weighted Voronoi diagrams serve as the value function of user's interest to a specific thematic unit derived on the basis of the process defined in the previous chapter. The justification of employing the mechanisms of Voronoi diagrams is given in the previous chapters.

2.5.1 Model for tracking individual contextual navigation

To be fully able to make future recommendations of newly created content, the history of movement through the information space for each individual user has to be utilized. For that purpose it is required to enable user tracking for the sole purpose of storing the needed data about users' interests. For this purpose, three techniques are identified as viable sources:

- use of databases for data storage
- use of cookies as data storage
- use of clickstream applications

The most reliable source is the first mentioned, database storage systems are utilized as they provide a mechanism that is most reliable for future recommendations. With that in mind, proper anonymization techniques will be applied as the role of the system is not to be able to identify the individual user, but to be able to provide tracking and recommendation services.

2.6 System and personal information of interest model

Through content categorization and personalization, defined and presented in previous chapters, a system is defined and its functionality provides

- available content ranking via the system preferences based on all users (default system state for a new user)
- available content personalization for the individual user with the help of weighted Voronoi diagrams
- grouping content in predefined topic units available through overall system and
- identification of identical content (based on predefined topic units) published by two or more source information (different Web news portals)

When a new user registers in the system, a state defined in subchapter 2.5.1 represents the default state of preferences. This starting state can be modified by an individual selection of personal topics of interests (specific topic units). All categories included in the system have a weight value. The sources of information are added by the system development team and depend on the development of the parsing system of the presentation layer of a given individual information source (additional news Web portal).

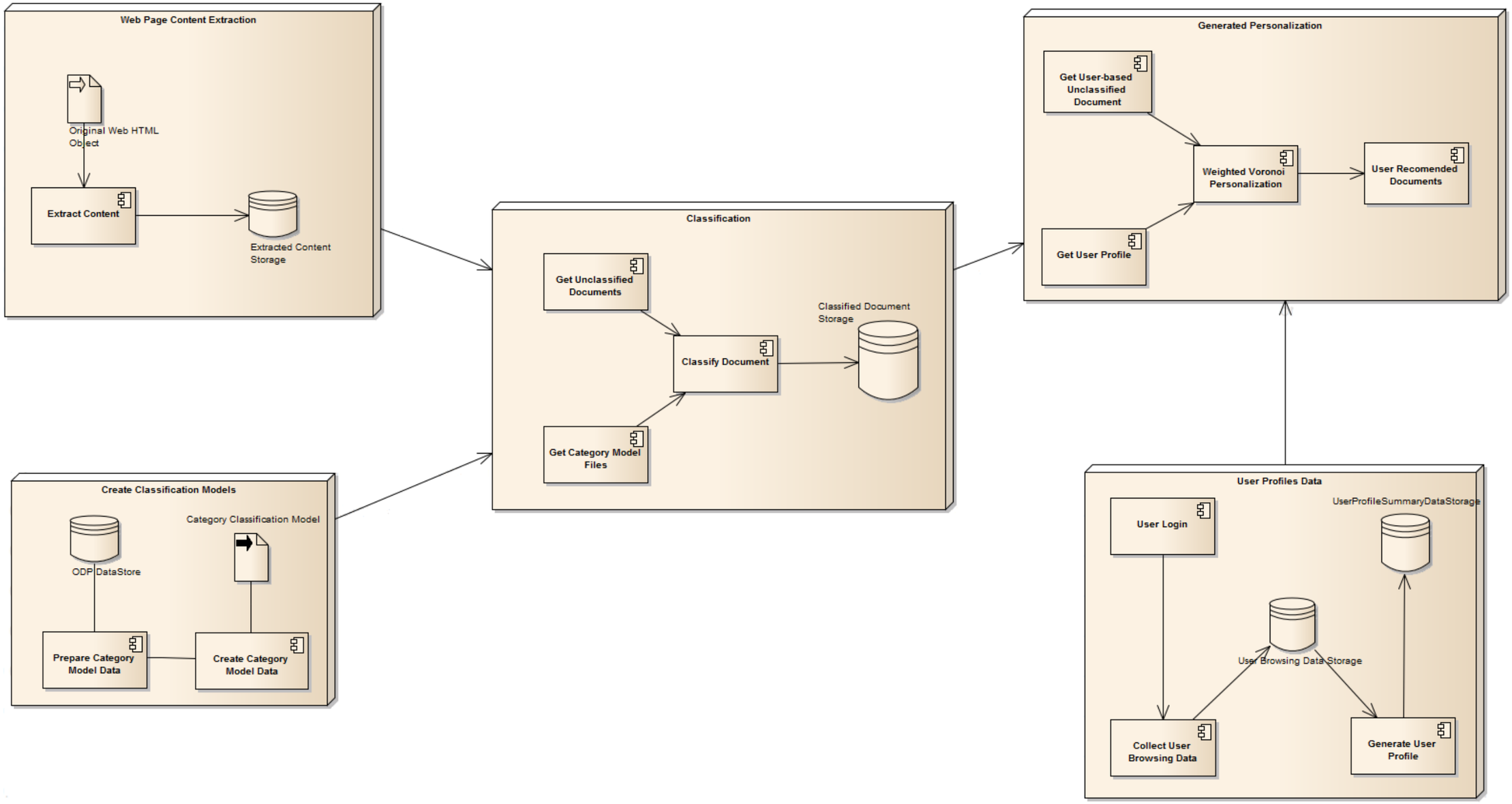

Figure 2: Proposed personalization system components overview

Given the characteristics of the vector space model and the application of computer calculation power required for the successful functioning of this system, the following potential problems, on a technical level, have been identified during the research:

- High dimensionality of the vector space defined in this way requires more calculation power. The same applies to the individual calculation of personalized, weighted Voronoi diagrams based virtual profiles
- The needed memory size for the storing of data necessary for the analysis phase and the tracking of users is large

Given the hypotheses, presented in the following chapter, two potential problems have been identified:

- inability to define a sufficiently detailed behavioral pattern would cause the system to misbehave and
- possible worse categorization of documents into the appropriate thematic units compared to the existing algorithms.

The proposed personalization system, presented in image 23, is defined with five individual modules, which produce the needed functionality based on the previously defined theoretical foundations.

The modules are as follows:

1. *Web page content extraction module*, with the goal of discovering new, system non-existent, news documents and extracting its content (e.g. news article) from the entire HTML object; the output of this module is the actual content viewed by system users.
2. *ODP category based classification tf-idf model creation module*, with the goal of producing the needed tf-idf models (including all the needed additional file system documents) based on the previously mentioned *gensim* framework
3. *Document classification module,* with the goal of classifying the output from 1 based on the output of 2; each previously unclassified document is paired with a set of 15 values in range [0,1] where each value corresponds to the relative value of similarity of the active document to a single category
4. *User profiling module,* with the goal of collecting user-centric browsing data (e.g. active URL and time descriptors (start time, end time) for each visited URL); the results of this module are used to represent user profiles based on the ODP-based

categorization, proposed in this dissertation, supported with the timing information (time spent on each of the ODP defined categories)

5. *Personalization module,* whose goal is to calculate user-based recommendations based on the methodology of weighted Voronoi diagrams with the results of 3 and 4 serving as modules input.

A detailed description of each module is given in the following chapters.

2.7 Research goals and hypothesis

The aim of the study is to create a virtual profile based on the semantic value of information of visited nodes (web pages formatted with HTML language) at the individual level. To this end, the following goals have been identified:

- G1: Content analysis of collected documents with tf-idf technique to obtain a set of key words that describe the individual node information
- G2: Assigning documents to one or more thematic units, taken from the ODP taxonomy, based on the results of G1 in order to create semantic descriptors for different thematic units
- G3: Identification of individual thematic access sequences (defined as behavioral pattern), using the results of G1 and G2, based on the data about the history of individual access to information nodes in the domain of web portals
- G4: Development of a model of semantic personalization of information nodes, accessible via a web portal, through the implementation of weighted Voronoi diagrams based on the data obtained in G2 and G3

Dissertation hypotheses are the following:

- *H1: Using newly created content categorization, based on the ODP structure taxonomy; it is possible to create a virtual contextual profile*. The confirmation of the hypothesis H1 will be conducted in the following two steps:
    - (H1.1) The application of ODP taxonomy will create a unified taxonomy categorization of the existing information nodes to one or more thematic units

- (H1.2) The descriptive system for creating virtual contextual profile will be defined by using the results from (H1.1)

- *H2: In the information space, in the domain of web portals, it is possible to extract a unique pattern for individual user navigation through the information space*. The presumption that H2 is true is based on the assumption that the sample, described through the previously defined unified hierarchical taxonomy, will be detailed enough to describe individual movement through information space. The input data describing the movement of the user through an information system are based on (H1) and represent contextual descriptors of visited information nodes along with time data.

- *H3: A new method for data personalization can be described by using weighted Voronoi diagrams*. Based on the unified taxonomy (H1) and contextual information using virtual profiles defined in (H2) as input data, weighted Voronoi diagrams are applied to personalize the content of newly available information.

3. Information extraction - history and models

This chapter will focus on information extraction, its history and information retrieval models used in this domain. A brief historical overview will be given along with the most important milestones achieved. Additionally, the field of Natural Language Processing will be presented in the scope of the information needed for this work. Porter stemming algorithm, to date, the most used stemmer, will also be presented. Finally, vector space modeling, as the method used in this work, will be introduced along with its most important concepts.

Information extraction is defined as the discovery of relevant information and includes a search and comparison between documents that are a part of the collection of documents. Some of the serious studies related to the search of the collections of documents were created in the sixties and the seventies of the twentieth century through the work done by G. Salton, C.J. van Rijsbergen and others. The creation of the WWW as the platform of information exchange has dramatically increased the amount of available information and the amount of information produced. With the introduction of hypertext, a new upswing of interest in the area of information extraction was initiated and it presents the environment in which techniques that follow are presented.

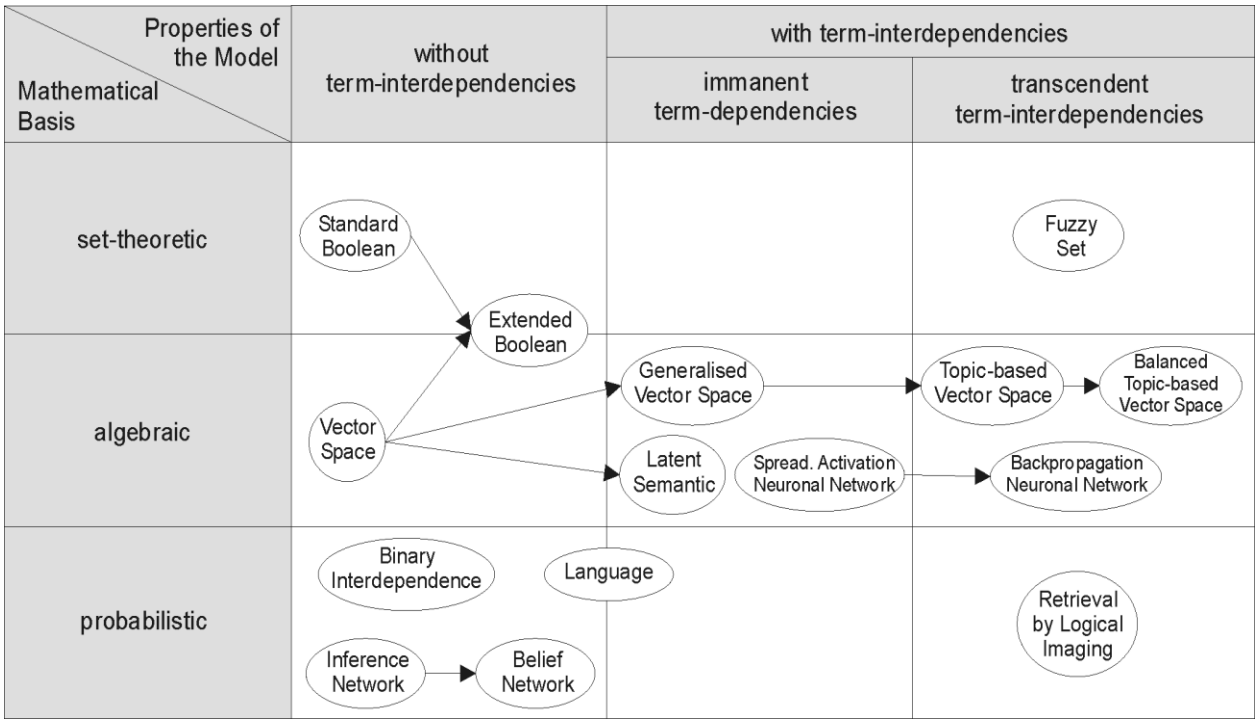

Figure 3: Information retrieval Models [51]

All existing models of information extraction, as shown in Figure 3, can be categorized, based on the mathematical theory on which they rely, as:

- algebraic models
- probabilistic models
- set-theoretic

Within these models there are models that allow the appreciation of linguistic features (recognition of synonyms, polysemy, language phrases, etc.) and those that do not allow such analysis. A complete classification is given in [52]. In this work the focus is put on vector space model. Vector space model was also created as a result of the previously mentioned SMART IR[13] system. VSM[14] allows words to be quantified in a single document and the document content to be represented as a point in space (vector in vector space). Such notation enables future processes of recommendation, personalization, information search and other approaches dealing with digital content to happen. Additional information about VSM is available in [53]–[56].

Information extraction is a subset of NLP[15] techniques that has the same role as the IR mentioned above. The only difference is the scope in which it is applied; whereas one applies IR on the whole document (a collection of an unstructured text), IE[16] is applied to "analyze only a small subset of any given text, e.g. those parts that contain certain 'trigger' words" [57] and it "focuses upon finding rather specific facts in relatively unstructured documents" [57]. Another definition states that IE is "any process which selectively structures and combines data which is found, explicitly stated or implied, in one or more texts" [58, p. 1].

This chapter will provide a brief overview of the basic techniques as listed in [57] and [58]. The main goal of information extraction is not to make a tool that "understands language" but instead to serve as a parser that recognizes linguistic patterns. From this point of view one can argue that natural language and its understanding are not really needed for these tasks but its

[13] Information retrieval

[14] Vector Space Model

[15] Natural Language Processing

[16] Information extraction

utilization, as stated in [59, p. 142], in combination with IE and VSM can improve results dramatically. As stated in [57] the main technique to solve these tasks include (but are not limited to) pattern matching, finite state automata, context-free parsing and statistical modeling. A brief overview of these techniques follows.

3.1 Information Extraction Techniques

The first technique mentioned is pattern matching. The simplest technique of pattern matching is simply matching strings of text. The way of achieving this is by using regular expressions (also called regex or regexp). In using regular expressions, the expression (defined as a rule in a list of rules or patterns) is the pattern and is made of a set of strings. Pattern rules are achieved by defining a set of patterns that are used as the comparison threshold. In addition to using a set of strings we can also use Boolean characters (e.g. (gray)|(grey) will match strings "gray" or "grey"), we can group items using parentheses and we can use quantifiers (?, +, *) and thus enrich our set of definitions or search terms.

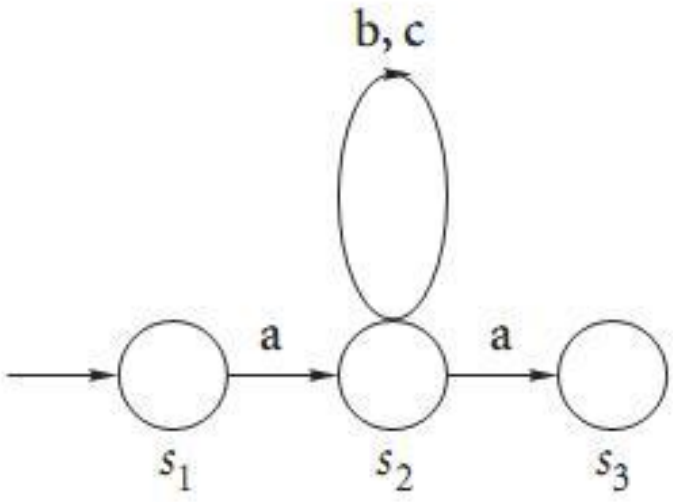

Figure 4: Finite state machine for a(b|c)*a [57]

As stated in [57], the regexp a(b|c)*a will represent a set of strings {aa,aba,aca,abba,abca,acba,acca,…} since we use the Boolean operator | (or). There is an

implementation of regexp pattern matching techniques in numerous programming and scripting languages such as Perl, JavaScript (among others), as well as command-line pattern matching programs (e.g. UNIX grep command).

The other approach found in literature is using Finite State Machine (a recommended project to look at is FASTUS - Finite State Automata-Based Text Understanding System [60]). A finite state machine can be defined as an abstract machine with finite states, transitions between them and actions for those transitions and is an abstraction of a real world process. The transitions between states are made in a series of discrete steps. In linguistics they can be used to describe the grammars of natural languages. Because of the nature of FSM[17], they can only be used to recognize or generate regular languages (where the number of next step possibilities is finite and depends on the current state, the input and the rules considering the current state). An example is shown in Figure 4. On the other hand, in [61] Chomsky argued that one cannot use finite state machines to describe language since language is a bit more complicated than that, because of its sentence richness and complication level. If we ignore his warning (if it is a warning) FSM can be used in two ways:

- for *string recognition* consuming character by character while making the transition to the next state
- for *string generation* by random (or rule based) transitions until it comes to a halt.

Figure 5: IE system modules [57]

It is once again suggested to look more into detail at the FASTUS approach (the operation after which closely follows the model presented in Figure 4) since it is an example of using FSM as a parser. More details about FASTUS can be found in [60]. The whole FASTUS

[17] Finite State Machine

system can be summarized as an FSM approach of extracting events from news. Although this approach (the FSM approach) is useful, its use is good up to a specific level of complexity.

One way of dealing with more complex phrases and rules is by using a context-free grammar. It can be used with FSM but with an addition of a stack, a memory tape. By giving FSM a chance to memorize the "context of the outer expression while the parser dives into the inner expression and then reinstates the outer context when the inner analysis is done " [57] (although stack memory is LIFO type of a memory – one can read from and write to the top of it). So, a context-free grammar can be defined as a grammar "in which clauses can be nested inside clauses arbitrarily deeply, but where grammatical structures are not allowed to overlap" [62]. This definition confirms the need for a specific memory for the operation to be able to work in this way. An example of this is the Cocke–Younger–Kasami (CYK) algorithm, presented as pseudo code:

let the input be a string S consisting of n characters: a1 ... an.

let the grammar contain r nonterminal symbols R1 ... Rr.

This grammar contains the subset Rs which is the set of start symbols.

let P[n,n,r] be an array of booleans. Initialize all elements of P to false.

for each i = 1 to n

for each unit production Rj -> ai

set P[i,1,j] = true

for each i = 2 to n -- Length of span

for each j = 1 to n-i+1 -- Start of span

for each k = 1 to i-1 -- Partition of span

for each production RA -> RB RC

if P[j,k,B] and P[j+k,i-k,C] then set P[j,i,A] = true

if any of P[1,n,x] is true (x is iterated over the set s, where s are all the indices for Rs) then

S is member of language

else

S is not member of language

As defined in [62] CYL algorithm "determines whether a string can be generated by a given context-free grammar and, if so, how it can be generated" meaning that it generates all possible sentences from a defined subsequence of words. The pseudo code works as follows:

a) first loop fills in the lexical categories associated with words
b) second loop (the entire triple loop) assigns non-lexical categories

In order for this to be possible (the triple nested loop) one has to have a good lexicon (with as many words as possible) and (complete) grammar rules which are used to identify grammar structures (e.g. a verb group or a noun group). Without those, ambiguity will arise. Further information about ambiguity and the way of dealing with it is available in [63] and [64], among others.

3.2 What is Natural Language Processing?

Natural Language Processing can, in a nutshell, be explained as a way of making the computer understand what the User meant (both in written and oral form). To give a more comprehensive definition one can quote [57] that states that NLP "is normally used to describe the function of software or hardware components in a computer system which can analyze or synthesize spoken or written language". The problem is not in the definition of NLP presented, but in the complexity of the phenomenon that is the target of NLP, namely: natural language. One of the examples is the word 'bank': it can be a financial institution, a river shore, relying on something etc. There are many similar examples that emphasize the main problems of NLP. The other problem is that language itself is a living entity that grows both in number of words and the possibilities of such words (starting with a single word being a slang word used in a specific area by specific population that can gain wide use if it is presented to the audience). The language itself is made up of the grammar (the logical rules of combining words of the language to make comprehensive sentences that have meaning) and the lexicon (the words and the situations of their usage).

NLP, in its battle for meaning, has a couple of tools available that make it easier for the algorithms to access, decompose and make sense of a sentence. Those tools will be presented after the listing as they can be found in [57]. They are:

- *Sentence delimiters and tokenizers* – detecting sentence boundaries and determining parts of sentences (based on punctuation characters)
- *Stemmers and taggers* – morphological analysis that links the word with a root form and word labeling giving information if a word is a noun, verb, adjective etc.
- *Noun phrase and name recognizers* – labeling the words with the noun phrase (e.g. adjective + noun) and recognizing names
- *Parsers and grammars* – recognizing well-formed phrase and sentence structures in a sentence

There are several linguistic tools that are used in text analysis. In general, the entire text is first broken down into paragraphs, paragraphs into sentences and sentences into individual words that are then tagged (to recognize parts of speech among other) before the parsing begins. The full suite of tools available are sentence delimiters, tokenizers, stemmers and parts of speech taggers, but they are not used in full in all situations.

The role of **sentence delimiters** and tokenizers is the determination of the scope of the sentences and identifying the members of a sentence. Sentence delimiters try to find the boundaries of a sentence, which can be a hard task since the usual sentence endings (e.g. period) can represent other meaning. They are usually created by using expression rules.

Tokenizers segment a list of characters into meaningful units that are called tokens. Creating tokens is language dependent as there are differences in how different languages mark word breaks (Latin based languages use white space as a word delimiter). They are usually created using rules, finite state machines, statistical models and lexicons.

Stemmers are used to find out the root form of a word. There are two types of stemmers: inflectional, that express the syntactic relations between words of the same part of speech (focus is on grammatical features such as present/past or singular/plural) and derivational, that try to bind different words to the same root (e.g. kind/unkind share the same root). They are supported by the use of rules and lexicons (they relate any form of a word to its root form).

Parts of speech taggers label the grammatical type of a word, assigning labels and deciding if a word is a noun, a verb, an adjective, etc. Since the same sentence can have more than one meaning, often POS[18] tag the same word with more than one label. POS can be rule based, that rely on linguistic knowledge to rule out tags that are syntactically incorrect, or stochastic, that rely on training data and assign tags based on frequency/probabilities (they are computed from the supplied training set that was matched by hand). Although POS are very useful in determining the structure and type of words in a sentence, some tasks require a more specific use of POS. At that point, noun phrase parsers and name recognizers are used. Their goal is to identify major constituents of a sentence (e.g. a name).

3.2.1 Porter stemming algorithm

Stemmers are used during the normalization process that is one of the foundations of creating an IR system. One of the most frequently used is Porter stemmer, created by Martin Porter and published in [65]. The role of this stemmer is to find the root version of the analyzed word and by this to normalize the document(s) that are being processed with the goal of improving the performance of the IR system. This algorithm is based on an approach that utilizes a fixed list of suffixes to stem as well as a list of rules that apply to each suffix during the stemming process. Another approach is to utilize a stem dictionary. The implementation in various programming languages can be found in [66]. The algorithm is divided into five steps and every word can be described in one of four forms:

$$CVCV\ldots C$$

$$CVCV\ldots V$$

$$VCVC\ldots V$$

$$VCVC\ldots C$$

where *c* denotes a consonant and *v* denotes a vowel. This can also be written as $[C](VC)^m[V]$, where (VC)m denotes subsequent number of VC pairs and is marked with *m*, called measure of the word. The overall rule of stemming (removing the suffix) is defined as:

[18] Parts of speech

(*condition*) S1 -> S2 [65, p. 214]

where S1 represents a given suffix and *condition* represents the stem before S1; if stem before S1 satisfies *condition* S1 is replaced by S2. This is presented as a list of possible transformations, based on the condition which is defined as the value of *m* plus additionally one of the following:

*S - the stem ends with S

v - the stem contains a vowel

*d - the stem ends with a double consonant

*o - the stem ends with cvc, where the second c is not W, X or Y

The combination of these rules makes the algorithm, depending on the word and its ending, to produce the final stem.

3.2.2 Why use Natural Language Processing?

"I never said she stole my money" - Someone else said it, but I didn't.

"I never said she stole my money" - I simply didn't ever say it.

"I never said she stole my money" - I might have implied it in some way, but I never explicitly said it.

"I never said she stole my money" - I said someone took it; I didn't say it was she.

"I never said she stole my money" - I just said she probably borrowed it.

"I never said she stole my money" - I said she stole someone else's money.

"I never said she stole my money" - I said she stole something of mine, but not my money.

Figure 6: "I never said she stole my money" example [67]

The last question in the introduction chapter is why use the NLP approach in extracting meaning form unstructured text? Part of the answer lies in the last sentence: the collection of

text on the WWW is still largely text based[19] and is in an unstructured form[20]. The second answer is that although one could use data mining techniques, they can only give us a version of statistical analysis of the text and ignore the meaning in the background. The third is a more philosophical answer that will start with the user and his individuality that data mining techniques results present. They are more concerned with the grey mass and not with the single user.

3.3 Document modeling and vector space model

The main way of summarizing and adapting unstructured text for later indexing and search is through giving a deeper "meaning" to the document itself. In HTML language (the basic presentation layer on the WWW) the document can be expanded through descriptive labels (e.g. META tag in HTML language) that represent the parameters whose values are checked during the search. While these parametric values are often predefined and structured, it is possible to use a descriptive label, called zones, which represent an unstructured abstract of the text as a way of mapping keywords and textual content (e.g. the title or the summary of the research paper is an example of the zone).

These approaches are still based on the postulates of Boolean logic where the search p is defined by the document collection d with the pair (p, d). A value in the interval *[0,1]* is added to the pair *(p,d)*. This form of evaluation of the document is often called a ranked Boolean approach. This approach does not take into account the possibility that the document that mentions a term often has more to do with a given search term or terms, and thus should be better ranked in the final results. Therefore, more sophisticated methods of creating the semantic value of the text and the analysis of similarity between documents, which are presented below, have been developed. They take into account the linguistic features of the analyzed text.

---

[19] with multimedia becoming more represented but the vast majority of presented information is still text based

[20] there are approaches on how to structure texts based on XML; OWL being one of more prominent as it is a standard one presented by W3C

XML - eXtensible Markup Language

OWL - Web Ontology Language

W3C - World Wide Web Consortium

One possible approach to the evaluation of the semantic value or the suitability of the document in comparison with the keyword(s) (for example through a query) is the vectorization of documents. This document representation is called the vector space model. The techniques of the evaluation of the document(s) in the collection, presented in this section, are ways of creating the vector space. Vectors themselves consist of dictionary elements (all words from the collection of documents being analyzed). The values of these components can be created in several ways and the most suitable for further research are presented in the subchapters that follow. The vector space model in the domain of information extraction is presented in [54] and fully defined in [53].

In this approach, each document is represented by *n* independent attributes derived from the name, summary or the content of the analyzed document. Each attribute has an associated value (0 denoting infinity), which indicates membership value of the attribute among the document attributes. Each document is represented by a vector (a set of attributes with associated weight values), for which the rules of commutativity, distributivity and associativity apply. The advantages of using the vector space model for the documents description are:

- it allows the representation of an object with multiple attributes (multi-dimensional spaces)
- attributes can be assigned a numerical weight value
- representation of documents as vectors allows the calculation of their mutual relations (e.g. the similarities/differences)
- document similarity calculation allows the creation of an ordered hierarchy of documents
- it allows dynamic adjustment of the results based on user feedback
- it creates an environment for the use of advanced methods of analysis, classification, categorization and personalization of documents in the collection of documents

The disadvantages of using vector space models are:

- multidimensional approach requires a greater amount of the necessary calculations
- semantic attribute values are lost by attributes extraction from the context

There are several approaches available in VSM, each with a goal of reducing the overall dimensionality of created vector space and improving the results of the returned documents (as presented in [68]):

- *tf-idf* where the number of occurrences of a word in a document is compared to the number of occurrences that the word has in the entire collection; this method is used in this work due to the nature of ODP
- *latent semantic indexing/analysis* is based on (tf-idf) matrix and tries to reduce its dimensions by singular value decomposition
- *probabilistic LSI/ aspect model* "models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of topics" [68, p. 2]
- *latent Dirichlet allocation* which "assumes that the words of each document arise from a mixture of topics. The topics are shared by all documents in the collection; the topic proportions are document-specific and randomly drawn from a Dirichlet distribution" [69, p. 2]

3.4 Word-document matrix

The word-document matrix ([59, p. 146], [70, p. 88]) is an *m*-dimensional matrix, where each column represents a single document (e.g. the weight value of a particular word in the document) and each row in the matrix represents each word that appears in the collection of documents which is analyzed through the word occurrence frequency in each document in the analyzed collection of documents. It is used as a way to view documents, or its contents, by the frequency of words relevant to the entire collection. The ways of assigning the word weight value to a particular document are different and the following chapters describe those most important for further work.

The matrix itself is created by text analysis from the document collection. Due to large amounts of different concepts that create the document, the matrix itself will contain mostly 0 values. To improve the content analysis one uses the term weighting methods which are based on the following assumptions [71, p. 81]:

- the words that appear in the document are more valuable for the document (they have higher weight values); this assumption is known as the local weight

- there is a difference between the commonly used words (everyday speech) and rarely used words (related to the analyzed document domain); this assumption is known as the global weight
- quantity of text in a document (document word count) affects the process of calculating document similarity.

3.4.1 Translating unstructured text into a text-document matrix

The process of conversion of the unstructured text for further processing using the methods of information extraction is called document indexing. With regards to the fact that the application domain of our information sources is the WWW interface, the steps are as follows:

1. remove HTML tags from HTML formatted text
2. tokenization
3. filtering
4. extraction of root of the words and
5. calculation of weight values.

During the first step, removing HTML markup language from document content, all HTML tags are removed from the original text/document. These include structural tags (e.g. HEAD, META, BODY, DIV, etc.), descriptive HTML tags as well as HTML tags whose task is to format the text itself (e.g. <P>,
, label styles, etc.). During the second step, tokenization, in the revised text (with HTML labels removed) all capital letters and punctuation marks are removed. During the filtering process the text is further analyzed and all the words with a big frequency (for example, a word that appears in all documents is not important in the semantic sense) as well as a word that appears in few documents (e.g. word that appears in only one document from the collection of documents being analyzed) are being removed. Thus the frequency of occurrence of words can be considered a value function for all the words within the document collection. As an additional tool that is used during this stage of purification and analysis of document content are stop words. The role of stop words is to define an additional set of words that do not carry semantic value within the observed set of documents and thus do not contribute to the results of the analysis. Their deletion from the content allows further analysis. Stop words can be taken from the existing collection of stop

words (for example through WordNet [72] lexical database) or can be separately defined for each collection of documents. Removing the root word from the remaining contents moves extensions from all the words that carry them and leaves only the root form of specific words. Only after the preparation of documents through the above mentioned steps we move to the step of calculating the weights (either local or global) for each of the remaining terms in the collection of documents.

3.4.2 Term frequency - inverse document frequency

tf-idf model is a continuation of the Boolean keyword mapping approach to digital collection(s) of documents. It starts from the fact that the document consists of a series of words (or phrases) that have different importance in the content of the text, thus affecting its semantic value. Two measures are used: term frequency and inverse document frequency.

Term frequency (tf) indicates the measure of the value of each term t in document d where the number of appearance of the term in the document is measured (this approach is also called bag of words; order of words does not matter). Of course, there are words within the document content that are not relevant to the document itself. Such words are generally not included in the analysis using predefined sets of stop words (mentioned previously).

To avoid the problem of attributing too much meaning to a given term, the inverse document frequency (idf) is used to model the frequency measures. If a word in the collection has a great tf value one can conclude that the level of importance in the whole collection is not crucial to the collection (e.g. the word ‘car’ in the collection of documents about cars). For this purpose it is necessary to know two values: N as the total number of documents in the collection being analyzed and dft as the number of documents in the collection which contain the word t. Then the value of the inverse document frequency obtained by

$$idf_t = \log (N / df_t)$$

Equality 1: Inverse document frequency

This resulting value raises the weight value of terms that are rarer in the collection and reduces the weight value of terms that are frequent in the collection. Finally, knowing these two values we can define tf-idf value as the product of tf and idf values of the document:

$$\text{tf-idf}(t, d, N) = \text{tf}(t, d) * \text{idf}(t, N),$$

Equality 2: tf-idf calculation

where *t* is the observed expression and *d* is a document from the collection of N documents. This measure assigns the word t in document d a value that is

1. the greatest where t is common in a small number of documents,
2. smaller when t is less common in d, or when it appears in many documents,
3. the smallest when t appears in all documents in d.

3.4.3 Document similarity calculation in the vector space model

The presentation of an analyzed unstructured text in the matrix format allows the document content using vector notation to be presented. In doing so, for each document in the collection, a separate vector is created whose components are elements from a set of terms generated through the text preparation steps defined in the previous chapters.

For two documents from the observed collection, D 1 and D 2, each consisting of m components, their vectors are denoted as $\vec{V}(d_1)$ and $\vec{V}(d_2)$. Their similarity is defined by

$$sim(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{\left|\vec{V}(d_1)\right| \left|\vec{V}(d_2)\right|}$$

Equality 3: Vector similarity

where $\vec{V}(d_1) \cdot \vec{V}(d_2)$ denotes the scalar product of two vectors and $\left|\vec{V}(d_1)\right| \left|\vec{V}(d_2)\right|$ the product of their Euclidean distance. The following are definitions of these two concepts. The scalar product of two vectors, $\vec{V}(d_1)$ and $\vec{V}(d_2)$, is defined as the sum product of their elements, mathematically expressed by

$$\vec{x} \cdot \vec{y} = \sum_{i=1}^{m} x_i y_i = x_1 y_1 + x_2 y_2 + \cdots + x_m y_m$$

Equality 4: Vector scalar products

while the Euclidean distance is given by

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{m} (\mathrm{x}_i - \mathrm{y}_i)^2}$$

Equality 5: Euclidean distance

.

This method of calculating the similarity between two documents in the collection makes it possible to eliminate any problem that may arise by comparing the two documents of different lengths.

4. Document retrieval and performance measures

This chapter gives an introduction to document retrieval and its use in different research domains as well as an overview of modeling techniques used in this work. Important performance measures, used in this work, will also be introduced. They will model the way in which later sections evaluate the research results.

4.1 Document retrieval

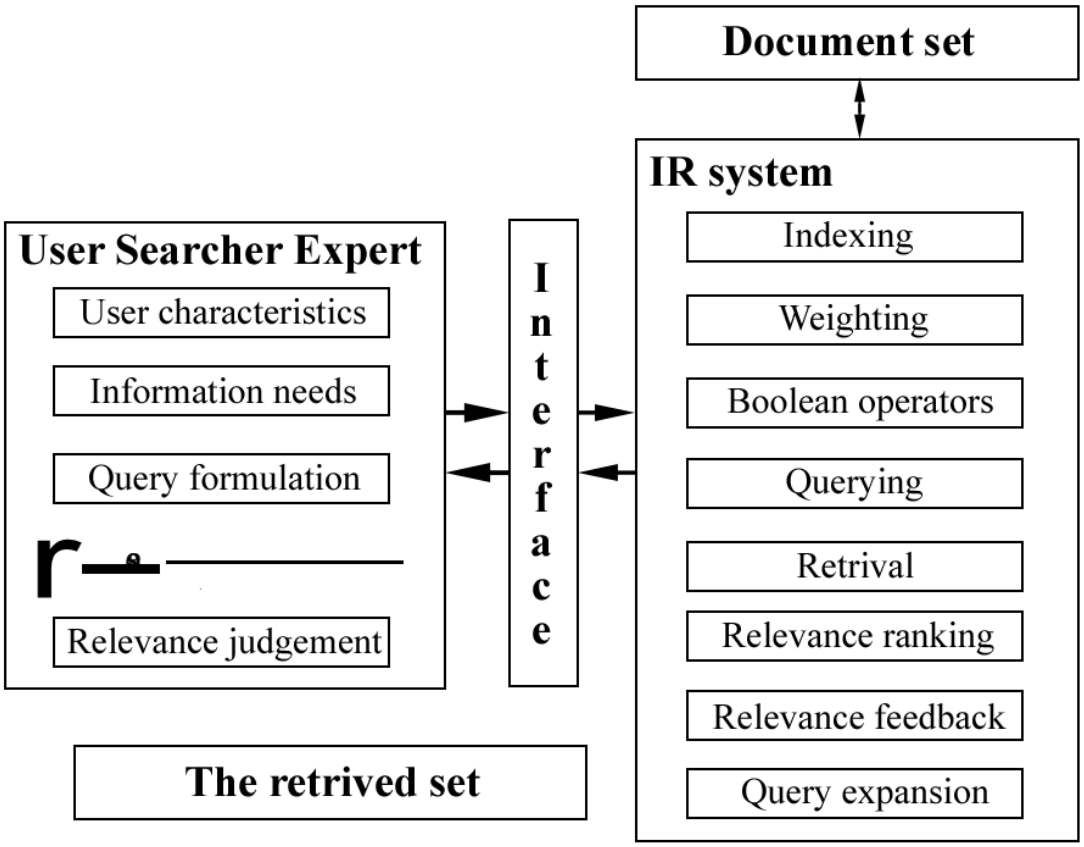

Figure 7: IR Process as found in [73, p. 23]

Document retrieval can be summarized as a method of accessing the wanted document based on a specific input. One can imagine a library as the most simple document retrieval place where all books are categorized into specific categories (e.g. a novel can be categorized by the country of origin, the type of novel, the genre, etc.). One can derive the simplest way of document retrieval from this example: keyword search where each of the documents is

described by a set of keywords and by performing the search, the keywords are matched and the results are shown.

This approach is a very simple and static one with no additional information about the relevance of the found information as it does not provide a hierarchy of any kind. The digital versions of libraries are search engines. The other approach is to use a full text search as a way of document retrieval where a document is searched (in NLP using the above mentioned NLP techniques) for specific phrases and is sorted based on the results of the full text analysis. This offers more accurate results that can be sorted by importance based on the keywords the user entered as important. Document approach is based on indexing and retrieval: the traditional approach that uses Boolean logic and the modern approach that uses term frequency explained in the tf-idf model. These approaches have never taken into consideration fuzzy logic set, since it gives more freedom but also demands more complex models of data collection, indexing and organization. More will be discussed further on.

4.1.1 Available methods/techniques

The core element of document retrieval is information retrieval. The definition stated in [74] defines IR as "defining retrieval models that accurately discriminate between relevant and non-relevant documents". Information retrieval is a user defined query that goes through the available set of documents and matches them to the terms of the search. As stated above, the comparison can be based on a set of defined keywords (defined by domain experts) or it can use the full text search approach. The returned results are sorted by relevance that can, but does not have to be related to the user's query (the user's happiness with the returned results is a topic for a different work and is a very delicate and complicated matter that deals with the concept of relevance of documents). The available models are shown in Figure 3. The Boolean model and the fuzzy retrieval will be covered. This will provide the needed theoretical models to fully understand how information retrieval works behind the scenes.

When dealing with full text search, to make things more practical, one uses the technique called indexing technology that basically takes a document and indexes it by all the words that occur in it. The result of this process is a dictionary and is presented in picture 4. The NLP tools mentioned in the introduction are used to make the dictionary more universal and more generic. The main parts of the dictionary, as seen in Figure 8, are:

- *Document count* is the number of documents in which a specific word occurs (and is used in the *tf-idf model*)

- *Total frequency count* is the number of times a specific word occurs in all the documents in the dictionary
- *Frequency* is the number of occurrences of a word in a specific document
- *Position* is the offset of the word occurrence in a specific document

User query is a method of search where a set of keywords entered by the user are compared to the dictionary and the results are shown (with more or less accuracy) based on the method of comparison (search). Of course, the need to get the most accurate results is always important and therefore there have been numerous attempts and developed techniques for such searches. The general overview is presented in Figure 3, and the most intriguing (at least for this author) is the set of methods based on fuzzy logic, since fuzzy logic can gradually assign the weight value of a specific item to a specific content and context.

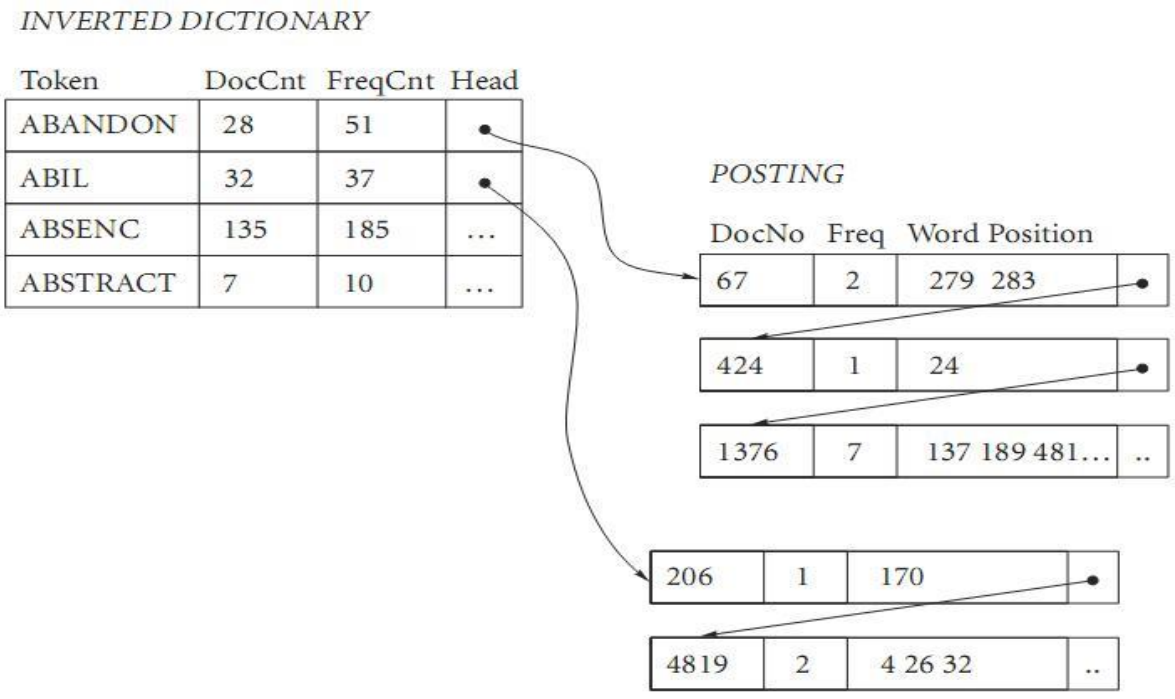

Figure 8: Dictionary sample [57]

One has to mention that fuzzy logic theory in general is a very specific theory that, in order to be implemented in the right way, needs to have a domain specific model on which it is based.

The basic search method is Boolean search (more about the method can be found in [75]), also known as the classical information retrieval model (there is also an extended Boolean logic based model), and it uses the general Boolean operators (AND, OR and NOT). The bases of the Boolean logic are Boolean truth tables. Retrieval is based on whether a document contains specific query terms or not. The drawbacks of the standard Boolean model are that the truth tables rules return a set of results that can either be too big or too small since it does not weigh the query terms in any way. As mentioned in [57], the traditional Boolean search has the following problems:

- *Large result set* made up of all the documents that satisfy query keywords
- Complex query logic for efficient queries
- *Unordered results set* nor ordered by relevance
- *Dichotomous retrieval* with the results set not having degrees of relevance
- Equal term weight for query terms

Although standard Boolean search can be sufficient for the experienced user, the inexperienced users will have problems when using it. For that reason the extended Boolean model, that gives the possibility of a ranked retrieval, has been developed. Using vectors to describe the query, term weights are taken into consideration and it looks for similarities between the query and the matched document. The weight mentioned is called term frequency–inverse document frequency. This weight measure lists a set of terms and calculates how important a term is to a specific document, which allows the extended Boolean method to use term weighting in its query processing. As mentioned in [76], the weight can be defined as

$$w_i = f_{x,j*\frac{Idf_x}{max_i Idf_x}}$$

Equality 6: Weight definition

with the weight vector

$$v_{d_j} = [w_{1,j}, w_{2,j}, \dots, w_{i,j}]$$

Equality 7: Weighted vector description

defining the weight of each query term in a specific document and Idf_x defining the inverse document frequency. Inverse document frequency measures the relative rarity of a term and is calculated by

$$Idf_x = \log\left(\frac{N}{n_x}\right) [57]$$

Equality 8: Inverse document frequency

where N is the number of documents in the collection and n_x is the number of documents that have the term x. Finally, by using two vectors, one describing the query and the other describing the comparing document, the similarity is calculated using the formula

$$sim(q,d) = \frac{\sum_x w_{x,d} * w_{x,q}}{\sqrt{\sum_x w_{x,d}^2} * \sqrt{\sum_x w_{x,q}^2}}$$

Equality 9: Document similarity

by introducing relevance in the query results one can, to some extent, manage the information crisis that was the building block of this idea.

There are numerous other attempts to solve this problem, and this paper will shortly mention only fuzzy logic attempt to deal with the relevance of a document. The first question is why bother with fuzzy logic in the field of IR? In the traditional Boolean logic, a proposition can be either true or false and their results can be derived, no matter how complicated they are, based on the truth tables. Fuzzy logic gained in importance and popularity in the last two decades with numerous papers and books written about dealing with fuzziness in various fields. Although this covers the needed foundations for this dissertation, the interested reader is advised to consult [73], [77]. Fuzzy logic was introduced as a way of dealing with the

weights of a document, which are the basic result of IR and are used to sort documents. As mentioned in [78] there are 3 types of weights:

- *relevance importance* - link between the weight of a keyword in the query and its presence in the document (high – high or low – low)
- *ideal weight* – similar weight of a keyword in both the query and the document presence
- *threshold weight* – a keyword with the weight at least as big as in the query

The useful property of a fuzzy set element is that its belonging to the set is in the interval of $[0, 1]$ and the functions that define this are called membership functions. As seen in picture 3, fuzzy set as an IR method is used when there is term dependence, meaning that one looks at the whole sentence (or part of it, assuming that logical parts of the sentence are separated by one of the grammatical delimiters, in this case punctuation marks) and determines structures that make a whole. This approach is the right way when talking about IR and extracting meaning since meaning extracted out of this context has no meaning at all. By the definition that can be found in [79] a fuzzy set is defined as a "set of ordered pairs $\tilde{A} = \{(x, \mu\tilde{A}(x)) \mid x \in X\}$, where $\mu: X \rightarrow [0; a] \subset ъ$, $a > 0$, is called a membership function (or the degree of compatibility or truth function), meaning the degree to which x belongs to Ã". One special case, the normalized fuzzy set, is the set where the membership function can be either 0 or 1. More general information about fuzzy set operations can be found in [80].

4.2 Performance measures

There are various evaluation measures listed below that evaluate the quality of a specific IR technique. In order to test a specific IR technique, one needs to test it on a collection of documents with a set of queries that are made of a set of relevance judgments on the mentioned document set.

There are test collections available as a part of evaluation series, most notably the Cranfield collection[21], Text Retrieval Conference[22] (TREC) document set, Reuters test collection[23], 20

[21] http://ir.dcs.gla.ac.uk/resources/test_collections/cran/

[22] http://trec.nist.gov/

[23] http://www.daviddlewis.com/resources/testcollections/reuters21578/

Newsgroups[24] test collection, just to name a few. Each of these collections is a collection of unstructured (natural) texts that can be used to assess the quality of a specific IR scheme. TREC is still popular and available today, with yearly tasks open to participate in. When talking about measuring the effectiveness of a specific IR scheme, one has to mention the difference between approaching ranked and unranked retrieval. The overall retrieval scheme is presented in Table 1. All formulas mentioned below are taken from [81]. More about various document retrieval evaluation (both ranked and non-ranked) can be found in [82] in chapter 7.

Table 1: Contingency table analysis of precision and recall [57]

	Relevant	Non-relevant	
Retrieved	a	b	a + b = m
Not retrieved	c	d	c + d = N - m
	a + c = n	b + d = N - n	a + b + c + d = N

Next, this paper will present the standard IR measures, precision, recall, fall-out, F-measure, average precision and mean average precision.

Precision (P) is defined as the fraction of the retrieved documents that are relevant:

$$Precision = \frac{\#(relevant\ items\ retrieved)}{\#(retrieved\ items)} = P(relevant|retrieved)$$

Equality 10: IR Precision measure

Recall (R) is defined as the fraction of the relevant documents that are retrieved:

[24] http://qwone.com/~jason/20Newsgroups/

$$Recall = \frac{\#(relevant\ items\ retrieved)}{\#(relevant\ items)} = P(retrieved|relevant)$$

Equality 11: IR Recall measure

Fall-Out is defined as the proportion of the non-relevant documents that are retrieved from all the non-relevant documents available:

$$Fall\ out = \frac{\#(non-relevant\ items\ retrieved)}{\#(non-relevant\ items)} = P(retrieved\ |non-relevant)$$

Equality 12: IR Fall-out measure

F-measure is defined as the weighted harmonic mean, known as F1, of precision and recall:

$$F = \frac{2 * P * R}{(P + R)}$$

Equality 13: IR F measure

Average precision takes into consideration the relevance of documents, giving them a higher score and is defined as:

$$AveP = \frac{\sum_{r=1}^{N}(P(r) * rel(\mathrm{r}))}{\#(relevant\ documents)}$$

Equality 14: Average precision

These efficiency measures are incorporated into the existing IR models presented in Figure 3. More detailed presentation of this topic is found in [57], [81] and [79], among others.

4.3 Text categorization

In the last two sections (and their respective subsections), the author went inside the document, presenting ways to go through the document content and build a structure of information of the content. This chapter will be a brief overview of the way to achieve the opposite: classifying a large number of documents and organizing them in a structure based on their content. An example of such classification is a provisory library system where each book is assigned to a general class with detailed classification following the main category (e.g., genre, country, century, author, publisher as the class elements). As mentioned in [83], text retrieval is typically concerned with specific, momentary information needs, while text categorization is more concerned with classifications of long-term interests. Understanding the data is the key to a successful categorization. One of the main problems with this field is that a successful categorization is more likely in a highly specialized field while a general categorization is more unlikely than likely since a general language knowledge transmitted into an understandable form by a computer is difficult to achieve (because of all the problem in the NLP, the complexity of language and the fact that language is a living entity that evolves over time not only in content but also in form). [57] lists categorization tasks and methods that are listed as follows:

- *Routing,* where an information provider sends articles to the user based on a user written query (one further step is a router that classifies all articles based on user's interests)
- *Indexing,* where each document in a collection is associated with one or more index terms from a vocabulary (there is an issue on how to provide for these classifiers since on a large scale document archive or feed the manual method would be too big a task)
- *Sorting,* where documents are sorted into (exclusive) categories but once again a high level of classification will need manual effort
- *Supplementation,* where a user defined (document based) keywords are matched with the overall system metadata (e.g. a scientific journal that classifies incoming articles into its own categories with the articles being described by author's keywords)
- *Annotation,* where one uses summaries to classify documents based on a preexisting scheme (although summaries have less text to work with, it is assumed that the summaries hold the relevant information to a specific point although there will be a specific amount of loss).

Now, since there is a need for human input (to define the scope, the levels of classification, the level of complexity, the underlying mechanisms of deduction etc.) there are a few factors (once again listed in [57]) that need to be taken into consideration while looking at the data:

- *Granularity* dealing with the level of division of the documents and the number of categories
- *Dimensionality* dealing with the number of features used for classification purposes (basically, the problem of defining a vocabulary for classification)
- *Exclusivity* dealing with the intersection between categories and how to define them
- *Topicality* dealing with the handling of documents dealing with multiple topics (and linked to exclusivity)

As with other language (or unstructured text) processing means, one has to be able to set the (starting) parameters and allow the algorithm to learn and improve over time. There are a few ways of doing that:

- Handcrafted rule based methods
- Inductive learning for text classification
- Nearest neighbor classification

4.3.1 Handcrafted rule based methods

The basis for this approach is using queries that go through a collection of indexed texts and compare them to the keywords of the query. This approach is the simplest one and can be efficient up to a specific number of categories and texts. The foundations of indexing must be done by a domain expert. An enhancement of this approach is using expert systems that use handwritten matching rules to recognize key concepts and assign documents with those values to specific categories. An example of such a system is Construe-TIS that uses Reuters news database and assigns (zero or more) labels to every new story in the database. A pattern defined as (gold (&n (reserve ! medal ! jewelry)) would detect the phrase gold but ignore phrases gold reserve, gold medal and gold jewelry [57]. Since the rules for this system were handcrafted by experts, the efficiency of categorization was quite high, which leads to the point that expert handcrafted system works on a very specialized domain with a specific size. Everything bigger would have problems in setting up the rules of recognition that would be as successful as the above mentioned system (recall of 94% and a precision of 84%).

4.3.2 Inductive learning for text classification

Once again, there is a need for the human input here, but only as feedback to the learning algorithm. The experts give a starting set of definitions and examples and let the program learn based on them. The core of this approach is the choice of the learning classifiers that are inputted manually until a sufficient number of examples exit for the program to analyze a new text automatically (new categories have to be inserted manually, nevertheless). The first approach is using statistical probability models, Bayes classifiers. Bayes rules show the relation between one conditional probability and its inverse and, assuming that classification elements can be extracted from a specific text, one can determine a set of labels (and subsequently categories) that the document belongs to. A basic Bayes rule is formulated as follows:

$$P(c_i|D) = \frac{P(D|C_i)P(C_i)}{P(D)},$$

Equality 15: Bayes rule

giving as the probability that a specific document belongs to a specific description class. As stated in [57] the "probability that a document belongs to a given class is a function of the observed frequency with which terms occurring in that document also occur in other documents known to be members of that class". What that basically means is that the documents that were given as a testing sample (e.g. user labeled) provide the program with the clues on how to approach new documents based on the knowledge about which terms to look for and their frequencies in a new document. This basic Bayes rule can be improved based on how the probabilities of terms given a class are computed offering the multinomial model and multivariate model.

Multinomial model (or bag of words) discounts the order of words and records the number of occurrences. With a training set made up of enough data one can calculate the frequencies in which terms occur in unclassified documents and see how they are associated to defined classes. The probability of a new document being in a class *X* would then have a specific term *Y* would then be:

$$P(\mathrm{Y}|X) = \frac{frequency\ of\ \text{"Y"}\ in\ known\ class\ X\ documents}{frequency\ of\ \text{"Y" in all classified documents}}$$

Equality 16: Multinomial model

.

Multivariate model uses binary vector representation of vocabulary word occurrences in the document without looking at the frequency of their occurrences. “The probability of a given document is then obtained by multiplying together the probabilities of all the components, including the probability of an absent component not occurring” [57, p. 127].

Another approach is to use linear classifiers. Linear classifiers use vectors called feature vectors to present feature values to the machine and they can be used as a mean of classification and machine learning. In text (and from that document) categorization, by using linear classifiers a document can either belong or not belong to a specific class. This approach uses vectors to describe a document either by defining the presence of a specific term in a document (by using Boolean notation – 0 or 1 values) or by using weights as a measure of the frequency of the term in the observed document and in the collection of documents (once again highlighting the difference between the probabilistic approach and future values approach). Once again, there is a vector for each category and each document is weighted and based on the “distance” to a class vector it can be classified in that category or not. There are a few ways of training linear classifiers, Rocchio algorithm [81, p. 269], Widrow-Hoff algorithm [57, p. 139] and Winnow algorithm [84, p. 129] being among more successful ones. The author will not go into detail about them in this work but can say that they deal with the way weights are measured and applied in the classification.

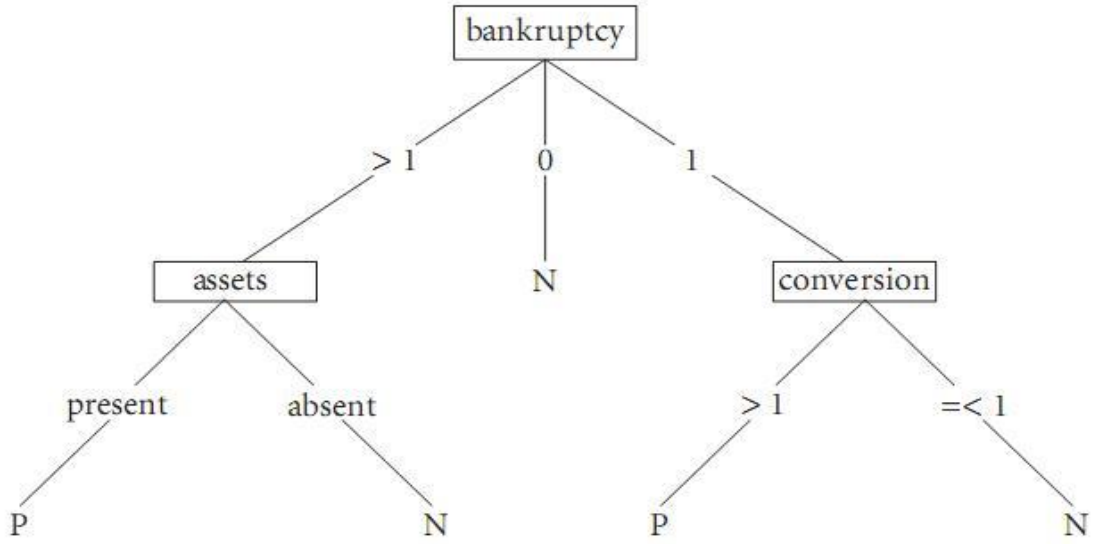

Figure 9: Bankruptcy decision tree

The third approach is by using decision trees and decision lists. As stated in [84, p. 5] "Decision trees, which specify the sequences of decisions that need to be made along with the resulting recommendation, are another popular means of expression.". In document categorization, such trees are comprised of leaves with categories having the possibility to overlap (more leaf nodes with same categories). Each category is represented by the tree path from the root of the tree. A sample is shown in Figure 9. When dealing with decision trees, one has to take into consideration that they work best with large data sets and that data must be in the attribute/value format. Decision lists are rule based lists that are a representation of Boolean functions, meaning that the rules are strictly ordered and can only contain Boolean conditions. The literature calls upon a tool called RIPPER [85], as an example of a decision list based text categorization tool. In a nutshell, the rules in the list are in the form: if f w1 ∈ D & ...& wn ∈ D then D ∈ C meaning that if a specific document D has labels $w_1,\ldots, w_n$ then it belongs to the class C.

4.3.3 Nearest neighbor algorithms

This family of algorithms has a different approach to learning by using rote learning ("Once a set of training instances has been memorized, on encountering a new instance the memory is searched for the training instance that most strongly resembles the new one." [84, p. 78]). The learning process works by the algorithm memorizing all the documents from the training set, selecting k documents for that set that are closed to the new document and from those

documents chooses and assigns categories to the new document. Classifiers assigned in such a manner are called k-NN classifiers and k-NN algorithms are defined as a method where "each new instance is compared with the existing ones using a distance metric, and the closest existing instance is used to assign the class to the new one." [84, p. 78]. In order to be able to do that (k closest neighbors have to be determined) one first has to define the metric to calculate the distances. One of the approaches is to use the Euclidean distance (the distance between two points in Euclidean space, Hamming distance ("if we have strings of the same length, we can define the distance between them as the number of positions that have different characters" [86, p. 148]) and by previously mentioned methods (the similarity between a user defined query and the document in the chapter about IR). Furthermore, after choosing k closest neighbors, one has to define the classes from k neighbors that will be used to describe a newly added document. One can simply rate the classes by frequency among chosen k neighbors or one can use a more sophisticated, weighted measure (the further a neighbor is from the document, the less value its category has in assignment). And the last choice is the number of neighbors taken into consideration that depends on two things ([57]):

- distance of classes in the feature space
- diversity of classes in the training data (more heterogeneous the larger the *k*)

5. Voronoi diagrams

In this chapter the author will give an overview of the needed mathematical foundations for the introduction and implementation of Voronoi diagrams with a detailed introduction of Voronoi diagram generalization, called weighted Voronoi diagrams. The reasons behind their implementation will also be given, in the scope of the proposed research area. The interested reader is advised to look into [87]–[89] for a detailed theoretical overview of Voronoi diagrams and their use in different scientific disciplines. A graphical representation of Voronoi diagram (in 2D space) is given in Figure 10.

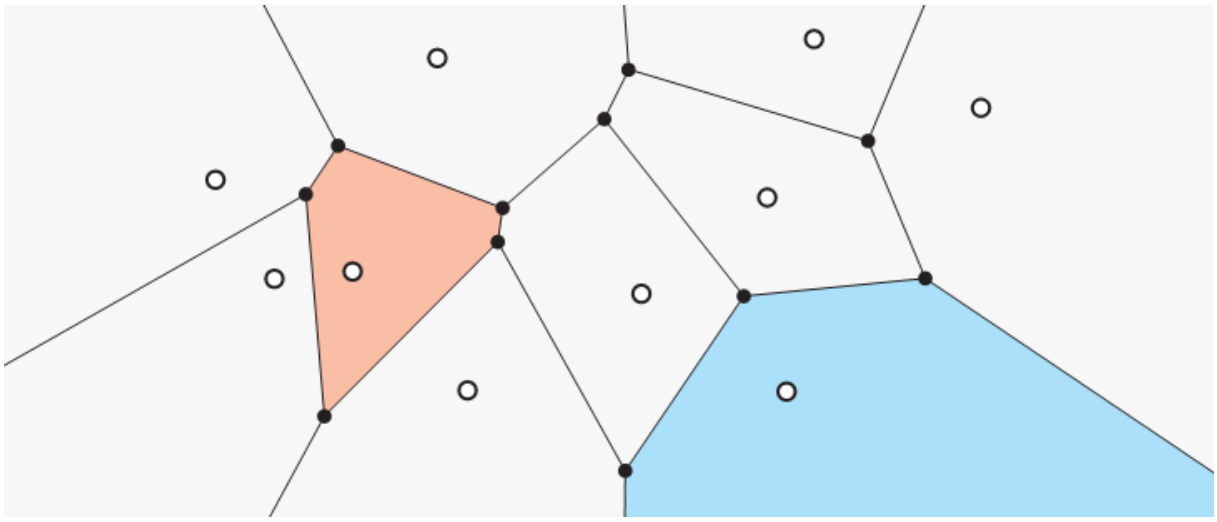

Figure 10: Voronoi diagrams for 10 sites [88, p. 99]

Voronoi diagram is a mathematical theory which presents a way of dividing space into a number of cells/regions with each cell/region having a generator (defined prior to defining the cells themselves). All the points in a single cell, except the ones on the bisectors, are, by some distance measure (usually Euclidean distance), closest to the cell generator. A generalization of Voronoi diagrams, weighted Voronoi diagrams, is of interest here as they allow calculating the modified distances between two points by adding weighting factors to the distance calculation. These weighting factors will be the basis for the personalized distance calculation between a cell generator (an average category point) and an unclassified point (information node descriptor).

The areas of Voronoi diagrams applications are many and include[90, p. 576]:

- Nearest neighbor search
- Facility location
- Largest empty circle
- Path planning
- Quality triangulations

Voronoi diagrams have been proposed, but under a different name, in the seventeenth century through the works of Descartes. In the nineteenth century a proper formulation was given by Dirichlet and Voronoy, where the Dirichlet space defined cases with two or three dimensions while Voronoy defined cases in m dimensions. The abstract idea of defining the Voronoi diagram is that each point in space is assigned at least to one of the Voronoi cell generator according to certain rules of allocation.

This work presents the implementation of the generalization of Voronoi diagrams for the personalization purposes of online available content of news Web portals. The categorization taxonomy is taken from the Web directory ODP [91]. This work uses Euclidean distance measure between each point and each cell generator as the allocation rule.

The properties of Voronoi diagrams follow. Voronoi polyhedron is convex. In Euclidean space, the object is convex if for every pair of points within the object, every point on the line segment connecting them is also located within the same object. Since the cell generator points are different, Voronoi polyhedron is not empty while all the Voronoi polyhedra are mutually exclusive except in their limits, called bisectors. It should be noted that, despite the fact that all parts of the Voronoi edges are parts of bisectors, bisectors do not always generate Voronoi edges. Voronoi edge is generated by the generator nearest to Voronoi polyhedra generator cell point, p_i. Generator p_i is the nearest generator for the p if and only if $V(p_i)$ contains the point p.

What follows is the formal definition of Voronoi diagrams in Euclidean space with m dimensions. All the definitions that follow in this chapter have been taken in its original form from [87].

Let $P = \{p_1, \dots, p_n\} \subset R^m$, where $2 \leq n \leq \infty$ i $x_i \neq x_j \; for \; i \neq j$. Let p = ($x_1$,..., x_m) define a single cell generator. Let p_i = (x_{i1},..., x_{im}) and p_j = (x_{j1}, ..., x_{jm}) define two points. The area

$$V(p_i) = \{p | \; \|p - p_i\| \leq \|p - p_j\| \; for \; j \neq i, j \in I_n\}$$

Equality 17: m-dimensional Voronoi polyhedron associated with point p_i

is called *m*-dimensional Voronoi polyhedron associated with point p_i while the set $V(P) = \{V(p_1), \dots, V(p_2)\}$ defines an *m*-dimensional Voronoi diagram. Area points of domination of point p_i over point p_j is defined with

$$H(p_i, p_j) = \|p - p_i\| \leq \|p - p_j\| \; for \; j \neq i$$

Equality 18: Points domination equation

and the bisection is defined as

$$b(p_i, p_j) = \|p - p_i\| = \|p - p_j\| \; for \; j \neq i,$$

Equality 19: Voronoi diagrams bisection

The distance between point p and cell generator p_i is defined as

$$d(p, p_i) = \|p - p_i\| = \sqrt{(x_1 - x_{i1})^2 +, \dots, +(x_m - x_{im})^2}$$

Equality 20: Distance between point and cell generator

In the event that p_i is the nearest (or one of the nearest) Voronoi polyhedra generator from point p, their relationship is defined as

$$\|p - p_i\| \leq \|p - p_j\| \text{ for j} \neq \text{i, i, j} \in I_n.$$

Equality 21: Nearest Voronoi cell definition

If that is the case, then point p is assigned to the cell with the generator p_i. Special types of Voronoi diagrams are weighted Voronoi diagrams, in which the generator is assigned a weight value that affects the allocation of points in space based on the cell generator. Their introduction follows.

5.1 Specializations of Voronoi diagrams: weighted Voronoi diagrams

A special type of Voronoi diagrams are weighted Voronoi diagrams, which, to each cell generator point p_i, assign a value that affects the allocation of points in space to that cell.

Let $P = \{p_1, \dots, p_n\} \subset R^m, (2 \leq n \leq \infty)$ be a set of points. A weight value that is based on the observed properties of the problem at hand can be added to each of the points. The set of weighting values are defined as

$$W = \{w_{i1}, \dots, w_{in_w}\}.$$

Equality 22: Weighting values set

In this case, the distance $d_w(p, p_i)$ is called the weighted distance. Dominace region of p_i over p_j is then defined as

$$Dom(p_i, p_j) = \left\{\{p\} \middle| d_w(p, p_i) \leq d_w(p, p_j)\right\}, j \neq i.$$

Equality 23: Dominance region

Okabe et al. in [87] define three types of weighted Voronoi diagrams. They will be introduced in this chapter and are as follows:

- multiplicatively weighted Voronoi diagrams,

- additively weighted Voronoi diagrams and
- compoundly weighted Voronoi diagrams.

They all rely on the previously introduced Voronoi diagrams properties and differ from the way weighting factors are calculated and used for cell allocation. The implementation of these generalized versions of Voronoi diagrams can be found in [31], [92], [93] respectively.

5.1.1 Multiplicatively weighted Voronoi diagrams

Multiplicatively weighted Voronoi diagrams give the weighted distances based on the weight values defined as:

$$d_{mw}(p, p_i) = \frac{1}{w_i}\|p - p_i\|, w_i > 0.$$

Equality 24: Multiplicatively weighted Voronoi diagram

Then the bisectors are defined as:

$$b(p_i, p_j) = \left\{p \middle| \left\|p - \frac{w_i^2}{w_i^2 - w_j^2}p_j + \frac{w_j^2}{w_i^2 - w_j^2}p_i\right\| = \frac{w_i w_j}{w_i^2 - w_j^2}\|p_j - p_i\|\right\}, w_i \neq w_j, i \neq j.$$

Equality 25: MWVD[25] bisectors

5.1.2 Additively weighted Voronoi diagrams

Additively weighted Voronoi diagrams define the weighting values as:

$$d_{aw}(p, p_i) = \|p - p_i\| - w_i.$$

Equality 26: Additively weighted Voronoi diagram

The dominance region behavior with this type of Voronoi diagram generalization depends on the values

[25] Multiplicatively weighted Voronoi diagrams

$$\alpha = \|p_i - p_j\| \text{ and } \beta = w_i - w_j \text{ where } w_i - w_j \geq 0.$$

Equality 27: AWVD[26] parameters

If $\propto > \beta$ the bisector is defined as the ratio

$$b(p_i, p_j) = \{p \| \|p - p_i\| - \|p - p_j\| = \beta\}, \propto > \beta, , i \neq j.$$

Equality 28: AWVD bisectors

If $0 < \propto < \beta$, the cell generator pi dominates the whole region. If $\propto = \beta$ then "the dominance region is the whole plane except for the half line radiating from pi in the direction from p_i to p_j" [87].

5.1.3 Compoundly weighted Voronoi diagrams

The third type of Voronoi diagrams generalization is the compoundly weighted Voronoi diagrams. This type of Voronoi diagrams calculate the distances based on the two previously mentioned Voronoi diagrams generalization. The distance is then defined as:

$$d_{mw}(p, p_i) = \frac{1}{w_{i1}} \|p - p_i\| - w_{i2}, w_{i1} > 0.$$

Equality 29: Compoundly weighted Voronoi diagrams

[26] Additively weighted Voronoi diagrams

6. On personalization using content, structure and user behavioral data

This chapter focuses on research efforts in the field of content based personalization of online content and introduces the needed mechanisms from web data mining, used in the preparation of the collected data for further analysis. An overview of the previous recommendation/personalization system is also given. Finally, a detailed overview of ODP with insight into the previous research efforts based on ODP is given. This chapter finalizes the needed theoretical framework of this dissertation.

Personalization has been discussed in brief at the beginning of this dissertation and will be elaborated in this chapter along with an overview of current state in the field of online content personalization with the emphasis put on research efforts in personalizing news resources. One can identify three transitions that have steered online users from static information sources towards dynamic (digital) sources:

- *content digitalization*, where the traditional information sources are slowly losing momentum (TV, radio, paper printed newspaper, etc.)
- *transition from static towards dynamic Web* where the roles have become mixed (the consumer became the creator/distributor of content)
- *change in the access device* (desktop, laptop, palm, mobile device, PDA readers, etc.)

In the meantime, the number of available channels through which one can access information has multiplied both in numbers (e-mail, RSS, Twitter, social networks, etc.) as well as in type of content presented through them (textual, multimedia). The exponential growth of available information has pushed the focus of researchers towards the possibilities of content personalization. With all this in mind, there are five challenges put in front of the challenge of (news) personalization [26]:

1. *Conflicting reading objectives*
2. *Difficulty of filtering information to fit user interests*
3. *Ways to generate the user profile*
4. *Novelty of the information*
5. *Depth of personalization*

In order to be able to discover useful patterns from any kind of data, data mining techniques have to be used. Three main steps in any data mining techniques (including Web mining or Web usage mining) are [24]:

1. *Pre-processing,* where data is gathered and any unsuitable data is removed from the data set
2. *Data mining,* where data mining techniques are applied to the cleaned data with the goal of pattern identification
3. *Post-processing,* where the results of step 2 are analyzed

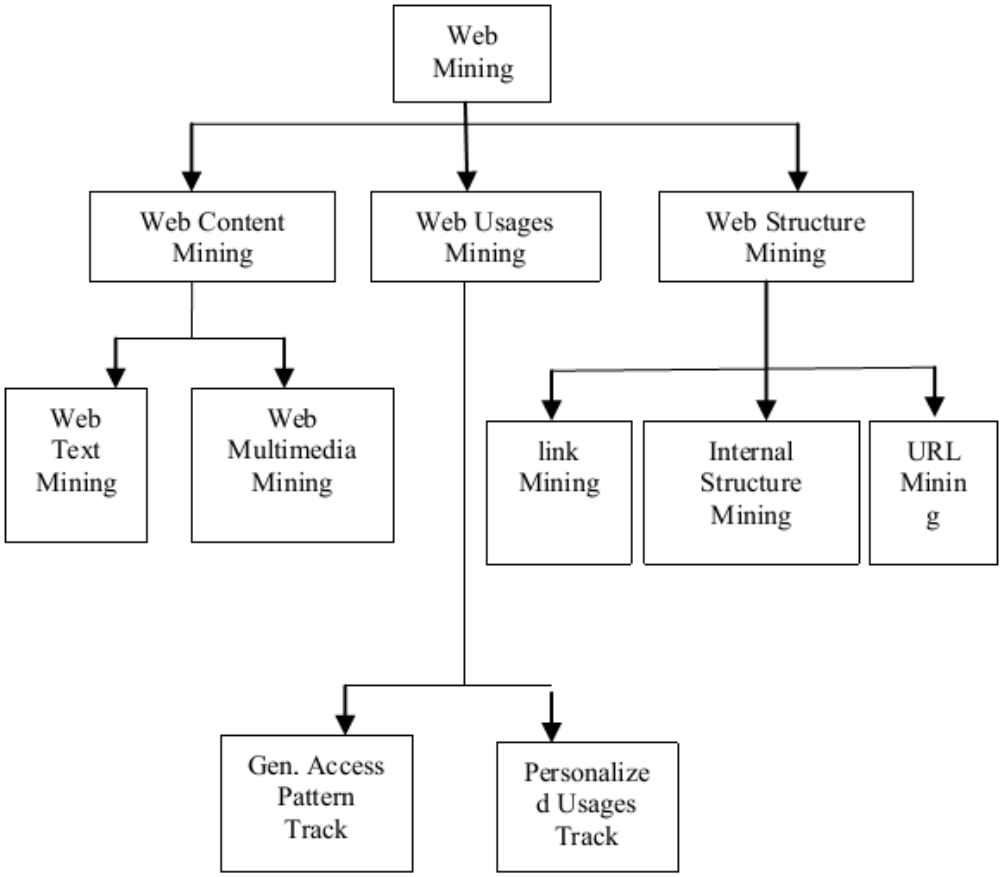

Figure 11: Web mining taxonomy [94]

Web mining presupposes the use of data mining techniques to "discover useful information or knowledge from the Web hyperlink structure, page content, and usage data"[24]. In [23], Web usage mining is defined as "the automatic discovery and analysis of patterns in clickstream

and associated data collected or generated as a result of user interactions with Web resources on one or more Web sites".

The available data sources that are used in obtaining the data needed for the analysis include data from Web server access logs, proxy server logs, browser logs, user profiles, registration data, user session or transactions, cookies, user queries, bookmark data, mouse clicks and scrolls (clickstream data) or any other data as the result of interaction [94]. These information sources are used in the subtask of Web usage mining. Aside from Web usage mining, textual (and multimedia in recent years) content found on the visited Web pages/resources can also be used (Web content mining), as well as additional information taken from the underlying visited pages structure (Web structure mining). The Web mining taxonomy is presented in Figure 11.

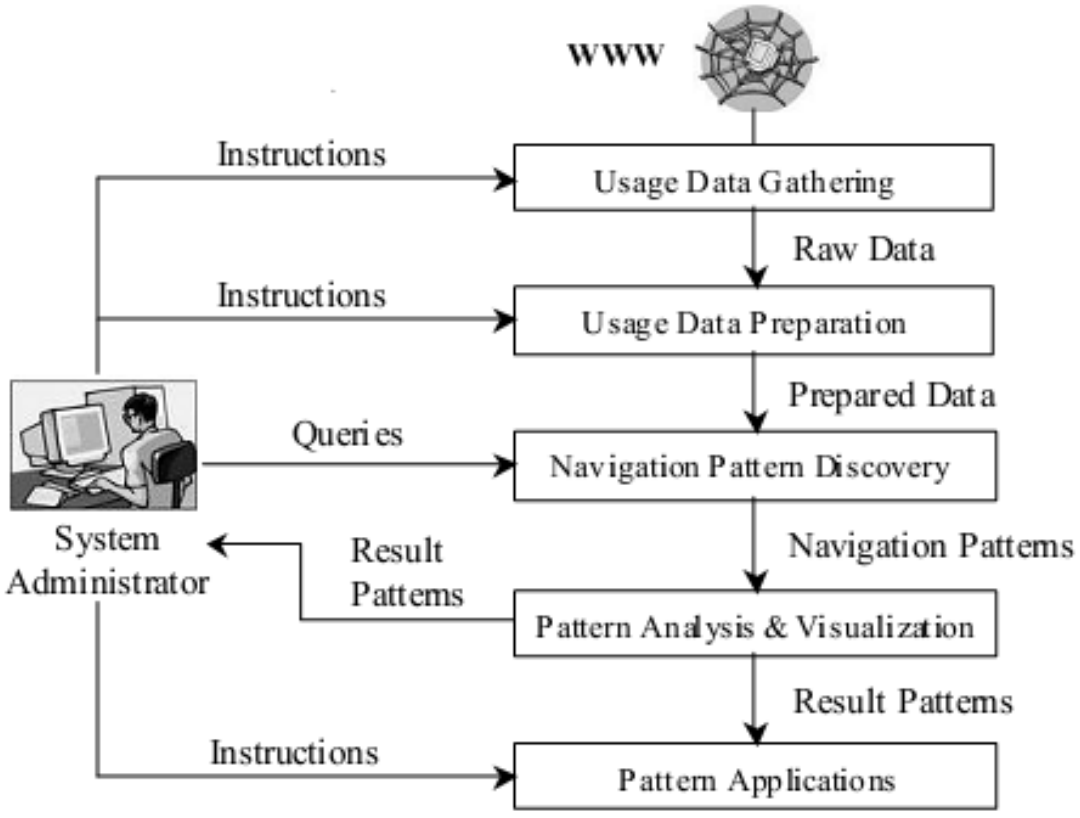

Figure 12: Generalized Web usage mining system [95]

The research into the possibility of using navigational pattern for content recommendation has a long history. The process of adapting the available information has been looked at from different angles and in different information domains. The previous research efforts consulted during the literature review phase will be presented in this chapter. The general architecture for usage-based Web personalization is presented in Figure 13.

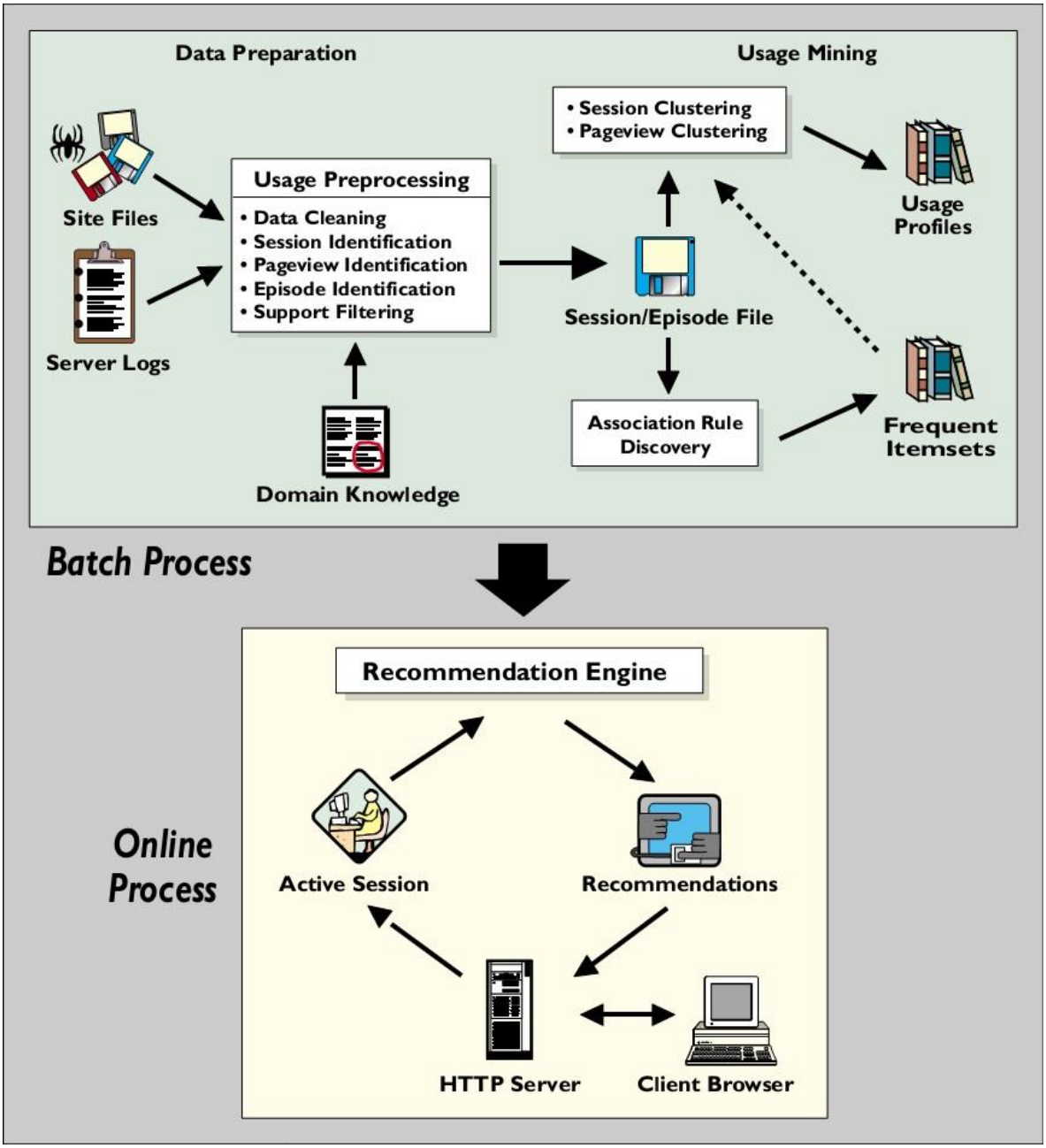

Figure 13: A general architecture for usage-based Web personalization [38]

The process is divided into two distinct processes: the batch process, whose purpose is to collect, preprocess and prepare the collected data and the online process that uses the gathered data and, based on it, serves as the recommendation engine.

A step by step overview of the same process is presented in Figure 12. In this chapter the author will give an overview of past research efforts in the process of collecting and preprocessing of used data as well as a way of using them in the identification of useful patterns/clusters. Also, an overview of related past personalization research projects is given in Table 2.

There are three main approaches for the personalization mentioned in the reviewed literature:

- *Collaborative filtering* (CF) where users with similar interests are grouped together divided into:
    - *User based filtering (most common, based on* [96]*)*
    - *Item based filtering*
    - *Model based filtering*
- *Content based filtering (CBF)* where items with similar content are grouped together
- *Hybrid approach* where different approaches are combined (most commonly CF and CBF)

6.1 Research efforts in the reviewed literature

Apart from the existing projects presented in Table 2, several additional research projects in the field of personalization will be presented next. They all differ on the methodology and type of personalization/recommendation. *WebSIFT* [97] uses the standard three steps in its personalization process (preprocessing, mining and analysis) and relies on using both available content and structure information about a specific Web site.

The needed data is taken from "the three server logs (access, referrer and agent), the HTML files that make up the site, and any optional data such as registration data or remote agent logs" [30]. The collected information is used to create a user session/profile. The information filtering based on both the user profile as well as contextual/structural information uses two algorithms to filter potentially interesting items: Beliefs with Mined Evidence (pages not

directly linked via hyperlink that bear similarities) and Beliefs with Contradicting Evidence (items that do not share specific item sets are considered not linked).

Table 2: Web Usage Mining projects in the reviewed literature [39]

Research projects ^	Data source			Data type				User		Site	
	Server	Proxy	Client	Structure	Content	Usage	Profile	Single	Multi	Single	Multi
GroupLens *	x		x			x	x		x		x
NewsDude *			x		x		x	x			x
Site Helper	x				x	x		x		x	
Letizia			x		x	x		x			x
Web Watcher		x			x	x			x		x
Krishnapuram	x					x			x	x	
Analog	x					x			x	x	
Mobasher	x			x		x			x	x	
Daily Learner *			x		x		x		x	x	
News@Hand *	x		x			x	x		x		x
NewsWeeder *	x					x	x	x			x
Google News *	x				x	x			x		x

^ research projects with focus on personalization

* applied in the domain of news

SEWeP ([98]–[100]) presents another web personalization framework that expands on web log data by including contextual and structural data of the observed Web pages. The proposed system is based on C-logs (concept logs) where standard web logs are enhanced with a categorization scheme of the observed document (URI). The categorization scheme is based on a predefined thesaurus. C-logs serve as the input to step 2, Web usage mining, of the three personalization steps. DBSCAN/THESUS system is used for the clustering of semantically enriched URIs. Data used for the evaluation was collected over the period of one year, divided into two distinctive sets, where each web page was semantically enriched with up to seven keywords which were then mapped to a specific category (each page assigned to a maximum of five categories).

EBOTS ([45], [101]) is a contextual IR system based on a premise that there are correlations between content of different texts and is therefore a pure context-based IR system. The system works based on domains, reference domains and correlation types (strong, weak and/or no correlation) [101]. Textual content is analyzed with tf/idf and the results are used for the domain creation that is followed by domain reference identification and definition of

correlation type. The evaluation was performed on the *classic3 dataset*[27] as well as the Time magazine news article dataset.

Hermes ([102]–[106]) represents a framework for building personalized news services using Semantic Web technologies, namely ontologies. The framework has four phases (*classification, knowledge base updating, news querying and results presentation*). In [102] efforts in news recommendation are categorized in semantic (Server for Adaptive News, YourNews, NewsDude) and non-semantic efforts (MyPlanet, SemNews, QuickStep). Hermes is an ontology based recommendation framework where ontologies are built from domain experts. It is both a system and a framework that allows additional expansions and is equipped with up to date semantic technologies. Recommendations themselves are given as a result of a query and are not based on user profiles. Ontologies are presented in OWL and SPARQL is used for querying. News preparation/analysis is based on advanced NLP techniques (tokenization, POS tagging, word sense disambiguation). [107] expands on Hermes framework by introducing user profiles (Ceryx framework) and two news similarity measures/methods: *Synset Frequency* (Inverse Document Frequency (SF-IDF)) where terms are substituted by WordNet synonym sets and *Semantic Similarity* which combines five existing similarity measures used for computing document similarities. [103] expands on Hermes with the introduction of *Concept Frequency - Inverse Document Frequency (CF-IDF)*, a version of tf-idf where concepts are used as input instead of terms. A needed requirement is the use of ontology as it is the source of concepts. Also, in this way the recommender system will provide better results.

peRSSonal [108]–[116] focuses on meta-portals (e.g. Google News[28], Yahoo[29]) and uses implicit user preferences data aggregation and dynamical user profiles updating. Each user profile is described with a positive (semantic analysis in user interests) and a negative vector (semantic analysis of items not in user interests). Documents in the system are organized in groups by their similarity (identical documents are documents that share the same facts but differ in their origin/source). Documents that are 16 or more hours apart cannot be in the same document group. The system captures HTML pages (via a web crawler; advaRSS, presented

[27] downloadable from ftp://ftp.cs.cornell.edu/pub/smart/

[28] http://news.google.com/

[29] http://news.yahoo.com/

in detail in [114],[111]) and extracts textual parts that are parsed (cleaned from HTML code), summarized, categorized and presented to the end user.

Table 3: ODP's RDF data dump structure [117]

Category	#Non-leaves	#Leaves	#Topics	#Terms	µC/N	Depth range
Adult	2085	5887	7972	2191	3,82	[2-11]
Arts	(3) 8344	(4) 38.525	(7) 46.869	(8) 28.595	(2,33) 5,62	([3-3]) [2-11]
Business	1977	9207	11.184	4311	5,66	[2-10]
Computers	(3) 1637	(3) 6478	(6) 8115	(6) 3730	(1,66) 4,96	([3-3]) [2-10]
Games	2775	8652	11.427	7013	4,12	[2-11]
Health	999	5534	6533	3276	6,54	[2-9]
Home	493	2026	2519	1667	5,11	[2-8]
Kids and Teens	837	3220	4057	3346	4,85	[2-11]
News	266	1583	1849	338	6,95	[2-7]
Recreation	1782	8770	10.552	3539	5,92	[2-10]
Reference	2747	8798	11.545	6650	4,2	[2-12]
Regional	87.262	210.130	297.392	48.429	3,41	[2-14]
Science	2205	9423	11.628	4965	5,27	[2-11]
Shopping	1173	4153	5326	3132	4,54	[2-10]
Society	4532	23.083	27.615	12.370	6,09	[2-12]
Sports	2329	15.446	17.775	7473	7,63	[2-10]
World	42.478	164.746	207.224	105.498	4,88	[3-14]
ODP	(7) 163.922	(7) 525.661	(14) 689.583	(13) 218.640	(1,86) 4,21	([3-13]) [2-14]
μ	9642,41	30.921,24	40.562,65	14.677,82	5,27	**[2,12-10,65]**
δ	22.321,01	60.125,68	81.986,75	26.316,08	1,13	[0,33-1,80]

µ, mean; *C/N*, children per node; *δ, standard deviation.*

Summarization of the analyzed text depends on [108]:

a) frequency of keywords in a sentence,
b) appearance of keywords in the title,
c) percentage of keywords in a sentence,
d) percentage of keywords in the text,
e) ability of keywords to represent a category,
f) ability of keywords to represent the choices and needs of a unique user or a category of users with the same profile.

Categorization is based on a pre-categorized document set and cosine similarity, dot product and term weights with two thresholds determining whether the document is to be regarded relevant/similar or not (cosine similarity between text and category denoted with T_{hr1} over 50% and difference of the cosine similarity between the highest ranked category and the rest of the categories T_{hr2} higher than 11%). The peRSSsonals profiling algorithm is extended through the work presented in [110] and is based on non-retrieval with the help of support vector machine method with standard TF-IDF weighting model at the basis of summarization process. The evaluation of peRSSonal is given in [109].

6.2 Open Directory Project data

As mentioned previously, ODP[30] is a Web directory. The original data comes in the RDF[31] format for both the structure and the entire content of the directory. Both content and structure files are freely available on the ODP Web page[32]. ODP itself was the first organized effort to classify Web pages manually into predefined categories and has, from its beginnings, relied on human editors and their manual efforts in classifying submitted Web pages.

The motivation in using ODP, as one of the possible taxonomies for further analysis, was drawn for the literature review of relevant research in the scope of research efforts in data personalization. The fact that the directory was manually edited and classified only added to the overall decision for using ODP in the process of reaching the above presented research goals and proving the defined hypotheses.

The data presented in ODP directory is divided into more than 590,000 categories with the number of sites listed in the directory exceeding 4.5 million. Additionally, besides the vast number of data it covers, it also presents a hierarchical categorization where each site belongs to one or more categories that are organized in (maximum) 13 category levels. This offers the possibility to use this information in further steps, as input into the Voronoi diagram calculation algorithms. For a detailed presentation of the data available in ODP RDF dump files, see Table 3. ODP database structure is presented in Figure 14.

---

[30] Open Directory Project

[31] Resource Description Framework

[32] http://rdf.dmoz.org/

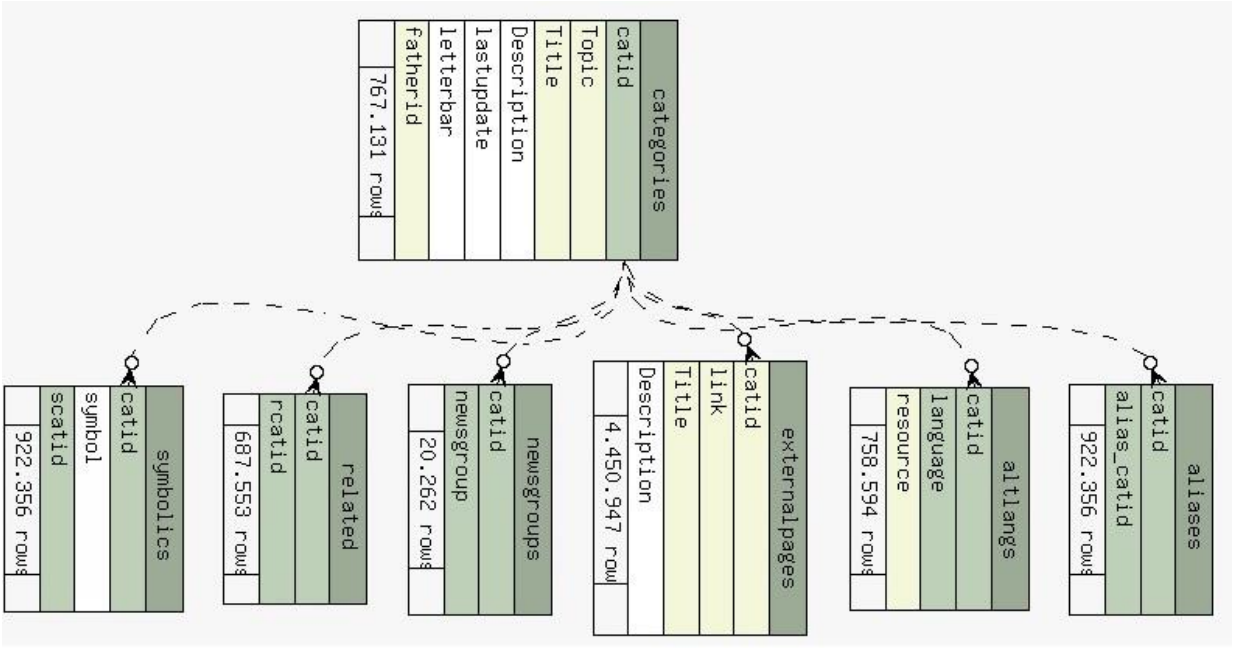

Figure 14: Open Directory Project MySQL structure

6.2.1 Open Directory Project in the reviewed literature

ODP has a long use, both practical as well as scientific, ever since it got into the spotlight at the end of the twentieth century. A proper introduction to the directory can be found in [118]. There are several ways how the data available in ODP was or can be utilized for the purposes of accumulating and extracting "hidden" knowledge from it. These research efforts are shown in this subchapter of this dissertation.

[119] extends on ODP data use by developing *Verb Extraction Engine (VEE)*, made up of two modules: *Relevant Document Searcher (RDS)* and *Verb Extractor (VE)*. The architecture is presented in Figure 15. Both modules serve to identify (*noun, verb*) pairs that best describe a specific ODP category. RDS searches for documents, relevant to an ODP category, simultaneously from four different sources: ODP body, SERPs[33], click logs and ConceptNet[34]. The second phase, VE, extracts verbs from documents identified as relevant in the previous stage.

[33] Search engine results pages

[34] http://conceptnet5.media.mit.edu/

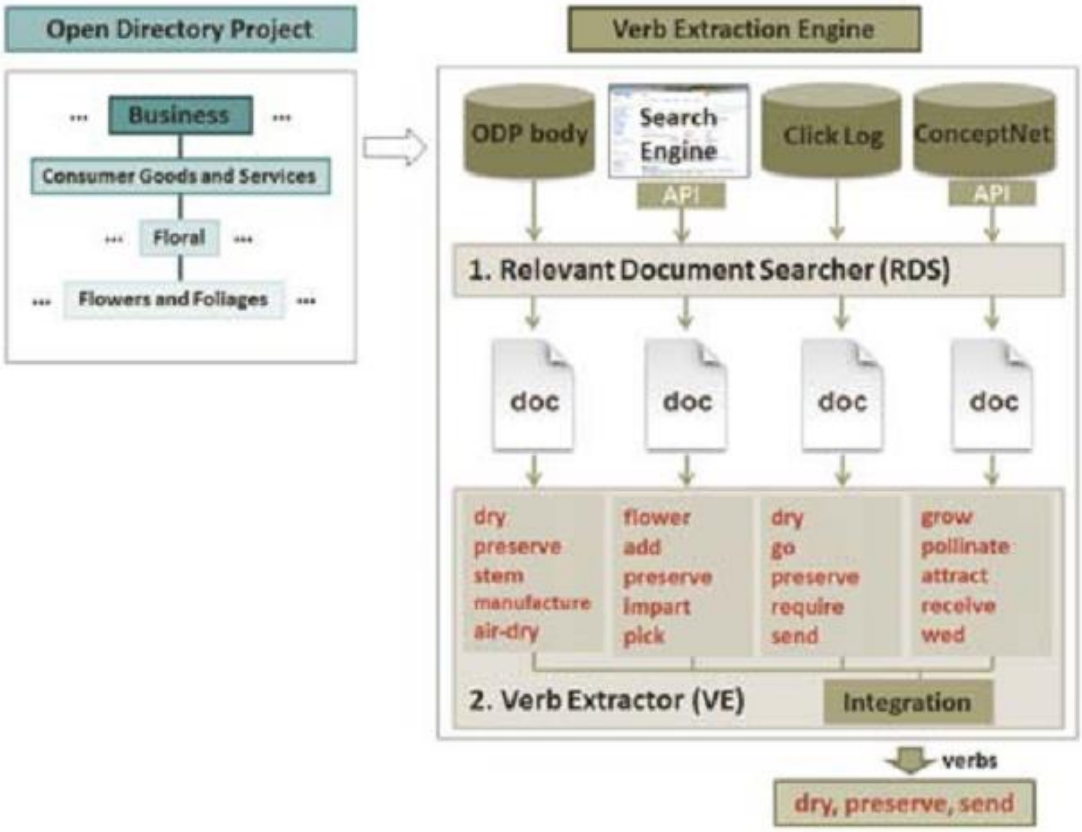

Figure 15: VEE architecture [119]

As stated in [119] VE follows these steps:

1) choosing sentences that include at least one term of the given category;
2) tokenizing chosen sentences;
3) tagging POS for each token (based on Stanford Natural Language Processing software[35]);
4) stemming to identify various forms of identical verbs (based on Porter stemmer);
5) dropping predefined stop verbs with high frequency and little value in representing user interests;
6) calculating relevance scores for all extracted verbs

ODP data was parsed based on the following three rules, which minimized the total number of the observed categories to 3,143:

[35] http://nlp.stanford.edu/software/

- removing useless categories that were not associated with any topic;
- excluding leaf categories in the original hierarchical structure of ODP;
- excluding categories with less than 50 web pages classified in the subtree rooted at themselves;

[5] focuses on employing (manually entered) metadata about topical categorization in improving the search results and focuses on ODP as it presents the biggest manually annotated categorization scheme available. The approach builds on PageRank as well as Personalized PageRank algorithms by utilizing user profiles created manually before performing the search tasks. Once again, the vectorization of user profile and documents is used to help calculate needed distance/similarity measures (e.g. minimum tree distance, graph shortest path). [5] proves that using ODP taxonomy based user profiles in sorting/filtering the returned results improves the results and confirms the usability of ODP (or any other taxonomy of its kind) in personalization efforts. Research results overview is available in Table 4. Previously, the work presented in [120] also dealt with using ODP data to personalized search efforts, with the difference being the scope of ODP data being utilized in the personalization effort. Whereas [5] focuses on utilizing ODP's subtopics, [120] focuses on utilizing the 16 top levels and then combining their vectors against the query.

Table 4: Effects of using ODP metadata on search

Algorithm	Ambiguous Queries	Semi-ambiguous Queries	Clear queries	Average/Algorithm
ODP Search	2,09	2,29	2,87	2,41
Personalized ODP	3,11	3,41	3,13	3,22
Google Search	2,24	2,79	3,27	2,76
Personalized Goog	2,73	3,15	3,74	3,2

[121] approaches a different research issue when it comes to using ODP data in the personalization: text classification using a hierarchical taxonomy, called deep classification, and identifies three main approaches (big-bang, top-down and narrow-down approach) in hierarchical text classification. The research proposes the use of combined Local and Global information for this purpose. For the experimental stage, data document belonging to the *World* and *Regional* categories as well as categories with one document were filtered out from the ODP. The document in the remaining set, identified as not written in English, have also been filtered out in later stages of data preparation.

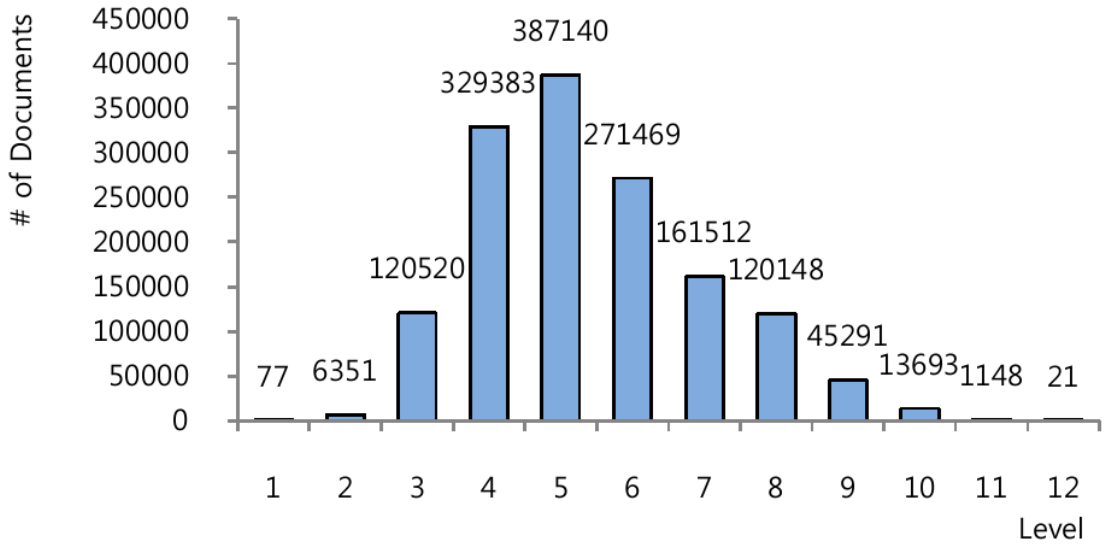

Figure 16: Level distribution of documents after filtering [121]

The proposed deep classification occurs in two stages: search stage, where relevant categories for active document are selected and classification stage where the active document is classified in one or more categories. Search stage once again utilizes the vector representation of documents (document based strategy) or categories (category based strategy) and compares the active documents vector with the selected search categories vector by utilizing top k results. The remaining documents (approx. 1.3 million) distribution, with regard of the level they can be accessed from, is shown in Figure 16. F1 measures are used as the comparison method with state of the art techniques mentioned in the research.

[22] looks on how to "expand features of the pages and ads by classifying them to a well-organized hierarchical taxonomy of topics" by enhancing the Rocchio classifier. Once again ODP taxonomy is utilized for this goal with top three levels of ODP being used. tf-idf weighting scheme is used with Euclidean distance or Cosine similarity as distance measures. The authors introduce the Category tag as "a form of meta-information used to describe the category of a page" and differentiate between tags and hierarchical categories. Tags are defined as user defined keywords that are used to create a personal classification. Hierarchical categories are defined as hierarchically defined tags where each category/tag has its own weight assigned to it which reflects tags priority in the hierarchical categorization.

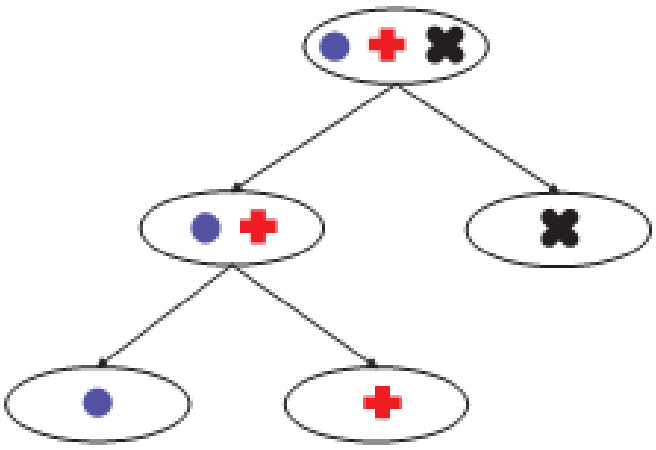

Figure 17: Nonlinearity problem through “circle”, “plus” and “x” classes [122]

The process of assigning these weights consists of three steps:

1. extracting documents from top *k* categories based on Rocchio classifier
2. calculating distances between the chosen *k* categories
3. calculating comparative distance score to determine the representative category whose score implies “how far category c_i is placed away from the other top *k* categories on the hierarchical taxonomy”

The used dataset was collected from BlogSpot.com and WordPress.com from the blog portals and CNN.com, Reuters.com, USNews.com and USAToday.com as representatives of online news media.

[122] observes, through *Refined Experts*, the problems of successfully adopting the use of taxonomies. The authors define three main disadvantages in using ODP (or ODP-like) taxonomy for classification as:

- “increasing sparsity of training data at deeper nodes in the taxonomy”;
- “error propagation where a mistake made high in the hierarchy cannot be recovered”;
- “increasingly complex decision surfaces in higher nodes in the hierarchy”;

In [122], through the proposed solution, the authors focus on the latter two disadvantages in the hierarchical categorization through a defined term taxonomy with the aim of improving macroF1 measure. The proposed solution to error propagation is presented under *Refinement*

where cross-validation between the predicted distributions and actual distribution of correctly classified documents is used. The second problem addressed in this work, nonlinearity, manifests itself in the fact that the top levels are not specific enough. An example is presented in Figure 17. The proposed solution to this problem is to use a combination of classifiers where the classification starts at the leaf nodes and the results are applied on the higher level. Refinement combined with the above mentioned solution provides a bottom-up approach for the mentioned problems and is the core of Refined experts solution.

Table 5: Research results - Refined Experts vs. Refinement vs. Baseline model [122]

	Baseline		Refinement		Refined Experts	
	Macro	Micro	Macro	Micro	Macro	Micro
Adult	0,167	0,39	0,171	0,421	**0,181**	**0,44**
Computer	0,228	0,252	0,233	0,278	**0,27**	**0,321**
Game	0,376	0,417	0,43	0,486	**0,468**	**0,561**
Health	0,341	0,443	0,373	0,509	**0,401**	**0,54**
Home	0,372	0,43	0,397	0,459	**0,405**	**0,51**
Kids & Tee	0,288	0,385	0,302	0,451	**0,324**	**0,514**
Reference	0,351	0,571	0,378	0,631	**0,436**	**0,688**
Science	0,3	0,35	0,315	0,396	**0,355**	**0,485**
Shopping	0,332	0,309	0,366	0,368	**0,421**	**0,439**
Average	0,306	0,394	0,329	0,444	**0,362**	**0,5**
			(7,6 %)	(12,7%)	**(18,4%)**	**(26,8%)**
Overall	0,302	0,365	0,326	0,414	**0,361**	**0,468**
			(7,9%)	(13,2%)	**(19,6%)**	**(28%)**

The proposed solution was tested on ODP taxonomy where 1.7 million documents were left after filtering and they were divided among 173,000 categories. Nine out of 15 categories were chosen to cover the topics. The classification was done solely based on the content of documents. The focus was to compare the proposed solution, via the macroF1 measure, with a base classifier (Hierarchical Support Vector Machine model). Research results can be found in Table 5.

Another approach in using ODP data and its taxonomy is the use of symbolic links. [91] defines a symbolic link as "a hyperlink which makes a direct connection from a webpage along one path through a directory to a page along another path". A visual representation can be seen in Figure 18. As the motivation for their work, [91] states that "symbolic links are used in nearly every web directory" but "very little research has been done to understand how symbolic links are used". Symbolic links, in the scope of ODP, can be used for shortcuts (bringing the user to a deeper level of the directory), backlinks (bringing the user to a higher

level of the directory), multi-classification links (allowing users to skip between different categories in one step) and cross-references. The results of their research show that "shortcuts and backlinks account for a very small percentage (<3%) of the total number of symbolic links in the entire directory" with backlinks being very rare (51 from 748,205 symbolic links present) without backlinks to the root/top-level category. Category *News* in ODP has the smallest number of shortcuts and backlinks which draws a conclusion "that news aficionados do not wish to skip through their topic, but rather prefer a more lengthy and thorough treatment of the subject". The vast majority of symbolic links in ODP are multi-classification links (more than 97% of all links). Regarding the case of the beginning and end of a multi-classification links, the results show that 89.06% multi-classification links connect pages within the same category while 10.94% connect pages between different categories, which leads to the conclusion that they are primarily used for local connectivity rather than global connectivity. Once again, multi-classification links in the *News* category distance themselves regarding these results; 83.64% of multi-classification links in this category provide a global connectivity. The results are presented in Figure 19.

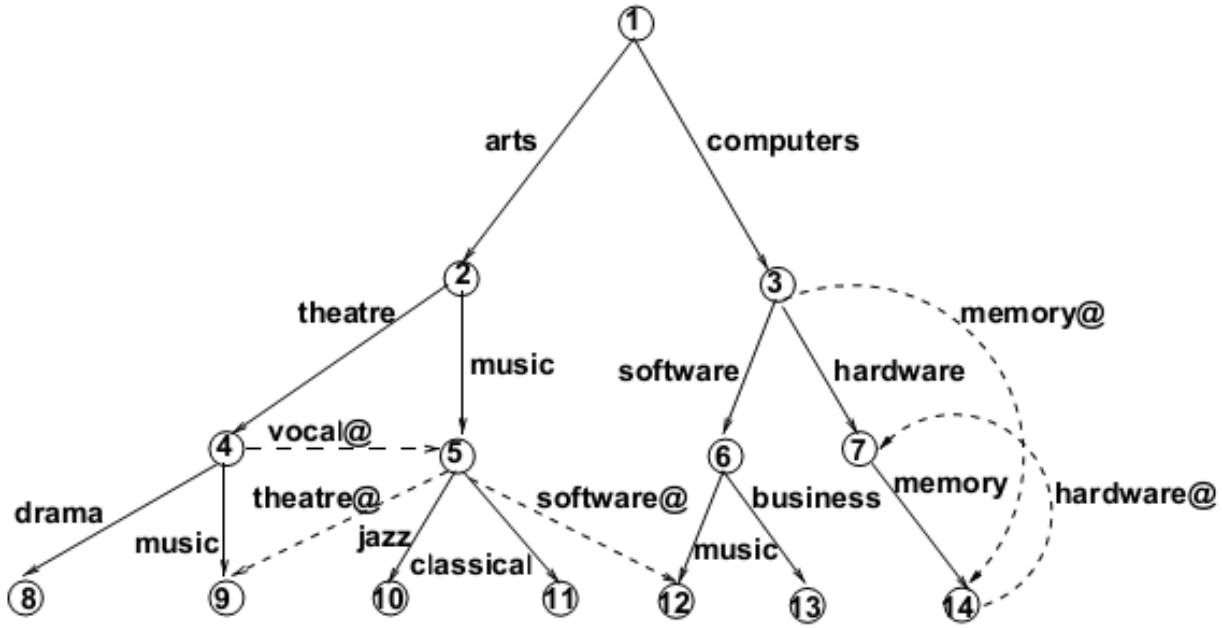

Figure 18: Symbolic link example [91]

[29] approaches the problem of automatically annotating ODP with keywords. They present the *KeyGen* algorithm made of *PageParse* (keyword extraction, using a combination of local and global weighting scores) and *Support* (merging keyword sets and eliminating keywords with a low support) algorithm. PageParse stage takes all the keywords from a document

collection and assigns a weight to each keyword based on the HTML tag the keyword was extracted from. Further weights are assigned according to local weights (term frequency of a keyword in a single document; tf weighting scheme), global weights (keyword term frequency in the entire collection; idf weighting scheme) and normalization factor (eliminating the discrimination based on the length of the document; cosine based measure), expressed through:

$$w_{ij} = tf_{ij} * idf_i * \cos j. \text{[29]}$$

Equality 30: PageParse weighting scheme

Support algorithm is a recursive algorithm based on the results of PageParse stage. Its focus is on defining the optimal set of keywords to describe a document. Once again, in each iteration, keywords with weights below user defined threshold are eliminated from the set. The evaluation was done on documents taken from the first six level of ODP. This produced 4,634,247 unique (and stemmed) keywords that represent 78,312 unique topics.

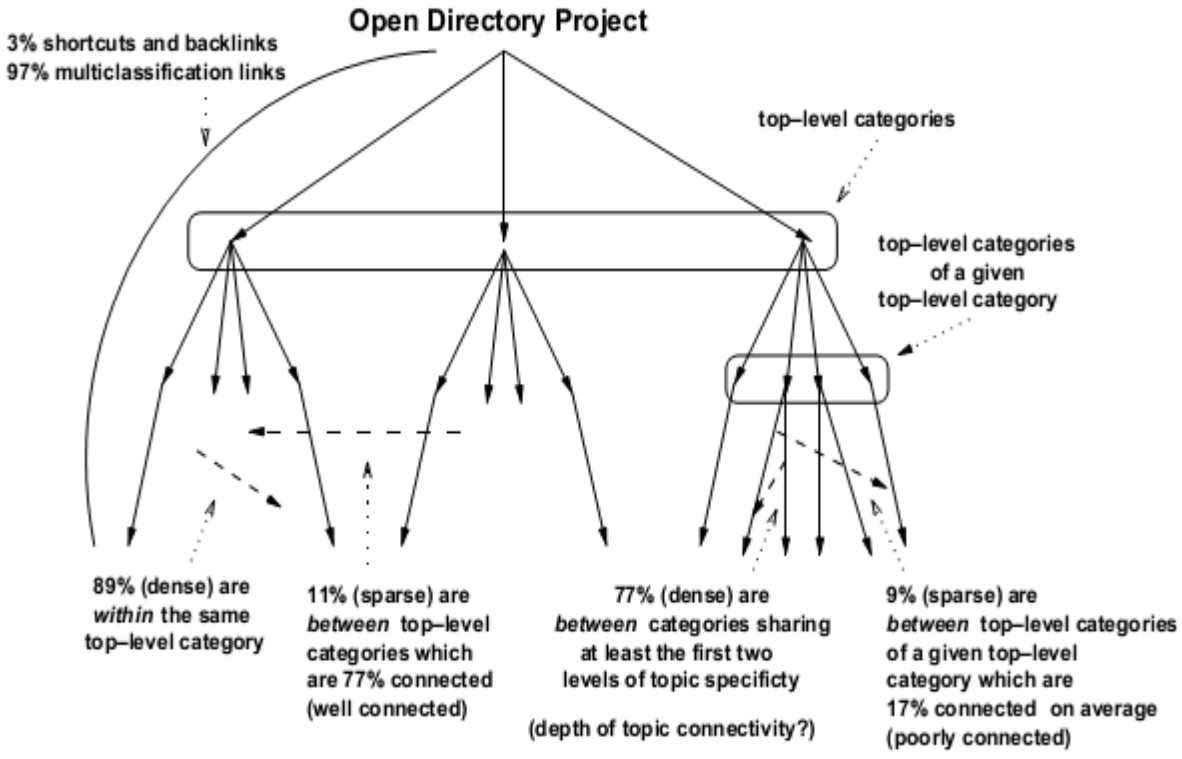

Figure 19: Graphical summary of the results presented in [91]

On average, every topic was described with 1,689 keywords. Automatic classification was performed based on the previously retrieved keyword sets. *Accuracy*, defined as 0 in precision, was taken as the measure of accuracy while the classification itself was done via the Nearest Neighbor algorithm. The classification was based on the documents in the first two levels of ODP and yielded 85.42% of correctly classified instances.

PART II: MEAT

7. ODP-based universal taxonomy (H1)

This chapter deals with the first hypothesis of this thesis defined as:

***H1**: Using newly created content categorization, based on the ODP structure taxonomy; it is possible to create a virtual contextual profile.*

ODP data, whose taxonomy is used for creating a universal classification scheme, is presented in more detail in chapter 6.2. Here is an overview of the algorithms used in preparing the available data and the results of the analysis. These results will be used for confirming H1.

ODP data, available in its original form through DMOZ's data dump files, was converted into MySQL available data with the help of an open source tool *suckdmoz*[36]. The database scheme created by this tool is presented in Figure 14, available in chapter 6.2. Two database tables are created and are especially interesting for further analysis: *dmoz_categories* and *dmoz_externalpages*. They offer a list of all the available categories with their descriptions and all the web pages available in ODP along with their descriptions respectively.

The overall process of creating a universal taxonomy can be divided into the following steps:

1. Data preparation,
2. Indexing
3. Similarity evaluation
4. Model evaluation

7.1 Data preparation

As mentioned, the data used for the analysis is freely obtainable from the ODP Web page as RDF XML files, both with content as well as structure. This raw data was prepared and converted into a MySQL database using an open source solution DMOZ RDF parser into MySQL[37]. The mentioned application is available under the GNU General Public License. It is a PHP based solution that converts the data from the RDF XML dump files into a MySQL database whose structure is presented in Figure 14. The scheme is generated directly from the

[36] http://sourceforge.net/projects/suckdmoz/

[37] http://sourceforge.net/projects/suckdmoz/

database using SchemaSpyGUI[38], also available under the GNU General Public License. In this way the ODP RDF XML dump is ready for processing and further analysis.

Raw data, available through ODP database dump, has to be prepared for further data analysis. For these purposes, the following algorithms have been implemented via Python programming language and its extensions, especially using *NLTK* and *gensim*[39] modules. NLTK offers a direct way for manipulating human written language and offers a set of tools to prepare the data for further classification and distance measures. For these purposes the *gensim* python module was selected.

Table 6: dmoz_externalpages descriptive statistics

Category	Mean	G mean	H mean	Median	Minimum	Maximum	Std. Dev.
Arts	27825,22	10967,562	2328,765047	12733	363	83627	28754,62038
Business	30273,63	14830,582	3887,92267	24076	690	74398	26174,46996
Computers	14848,63	6444,2559	2216,869561	9263	641	33965	13885,45221
Games	6973,556	2781,4008	687,5059104	3363	158	19378	6684,762825
Health	9063,3	4441,0812	1300,093798	7786	267	23736	7604,697154
Home	4278,667	2500,227	1262,848847	3462	401	10556	3588,044299
Kids_and_Teens	5090,222	2623,6446	997,2904809	3492	261	13903	4653,305128
News	1715,4	1110,7681	817,488778	1117	382	5260	1797,030284
Recreation	12662,22	6162,3965	1767,717937	10653	420	25588	9801,308799
Reference	7061,6	4059,1491	1779,131993	6646,5	431	24540	6653,31692
Regional	96214,75	8794,1429	11,60080827	48770	1	297601	108375,7547
Science	12906,67	7211,4362	2591,228124	10445	527	30313	10255,94563
Shopping	11731	5004,0888	1441,563914	6728,5	341	38328	12108,67765
Society	24166,3	9731,4154	1039,100972	25866	213	50972	18043,33091
Sports	12751,56	4676,4932	518,2088431	10133	70	27233	10363,97915

The basic statistical overview of ODP data, used in further processing steps (tables *dmoz_category* and *dmoz_externalpages* respectively), follows. Both tables show standard averages (Mean, G mean, H mean), the number of categories and external pages, respectively (Minimum, Maximum). The data in both tables is relatively scattered, which makes the features used in further steps more complicated.

[38] http://schemaspygui.sourceforge.net/

[39] http://radimrehurek.com/gensim/index.html

The data available from ODP's database needs to be prepared for further use. For that purpose data available through ODP database dump was preprocessed as follows:

- Two data categories were removed from the database; data for both category "Adult" as well as "World" were not taken into consideration for universal taxonomy. After that there were 15 possible top categories to classify the data into
- The remaining data, for each category, was divided into levels, where the main 15 categories represent the top level (database field depthCategory = 1). The depth detection was based on two approaches:
    - delimiter based (in our case delimiter is "/")
    - Bottom-up approach, based on parent-child relationship, using *fatherid* column.
- Database entries whose field 'Description', both in *dmoz_categories* as well as *dmoz_externalpages*, was without a value were taken out of further analysis as their content vectors would not carry any semantic value. All rows in the column *filterOut*, both in *dmoz_externalpages* as well as *dmoz_categories* database tables, that were not taken into consideration for further processing were marked with the value '-1'.

Table 7: dmoz_category descriptive statistics

Category	Mean	G mean	H mean	Median	Min	Max	Std. dev.
Arts	5143,222	1819,088	296,2204	3055	41	17046	5701,782
Business	1552	916,2621	286,3015	1662,5	48	3220	1072,725
Computers	1067,5	541,2	198,1151	712	45	2521	930,3882
Games	1550,556	490,3906	78,38482	689	16	4703	1607,302
Health	1222	556,5392	123,6296	1283	23	3096	987,813
Home	392	242,1427	89,55552	309,5	19	780	275,7656
Kids_and_Teens	713,4444	333,2703	89,56899	524	14	2000	673,5003
News	100,8	74,649	52,42281	84	19	239	74,59866
Recreation	1303,222	659,1548	191,2766	833	32	2926	1092,42
Reference	1216	579,857	147,4483	1070,5	22	3319	1068,502
Regional	26284,17	3106,846	87,59706	9396	10	98928	34185,89
Science	1565,778	858,1718	274,4334	2005	52	3402	1108,029
Shopping	863,5	499,3929	178,8792	613	35	1835	647,1019
Society	2858,9	1182,434	188,6574	2272,5	30	5926	2377,969
Sports	2288,333	860,8057	156,2241	2308	25	5690	2057,057

Using the above mentioned filtering steps, the data reduction, for tables dmoz_categories and dmoz_externalpages, was as follows:

Table 8: Available data after filtering

Database table	Rows Before reduction	Rows After reduction	% of rows left
dmoz_externalpages	4 592 105	2 637 412	~57%
dmoz_categories	763 378	496 007	~65%

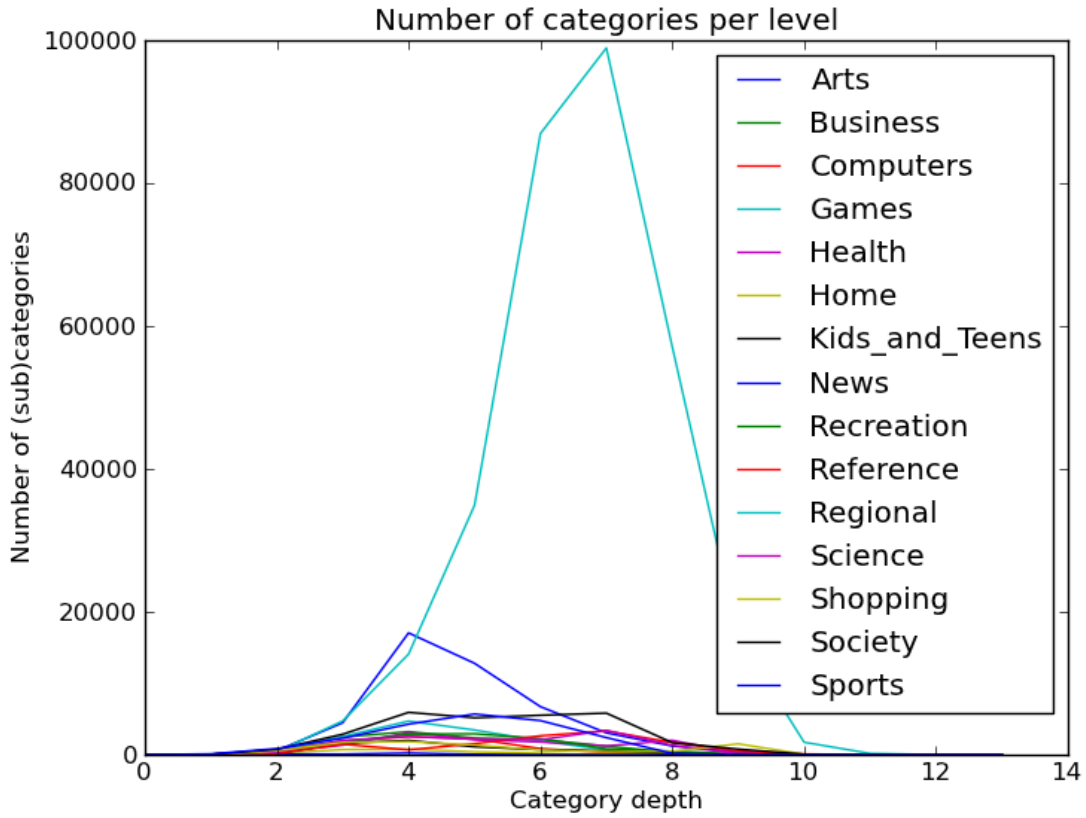

Figure 20: Number of categories per depth level

The distribution of the number of possible subtopics, for each of the 15 main categories identified as valuable for further analysis, which can be used for future classification, can be found in Figure 20. Figure 21 shows the distribution of the number of available pages, once

again for 15 identified categories, per depth level. Both images show that the category "Regional" stands out both in the table *dmoz_categories* as well as *dmoz_externalpages*.

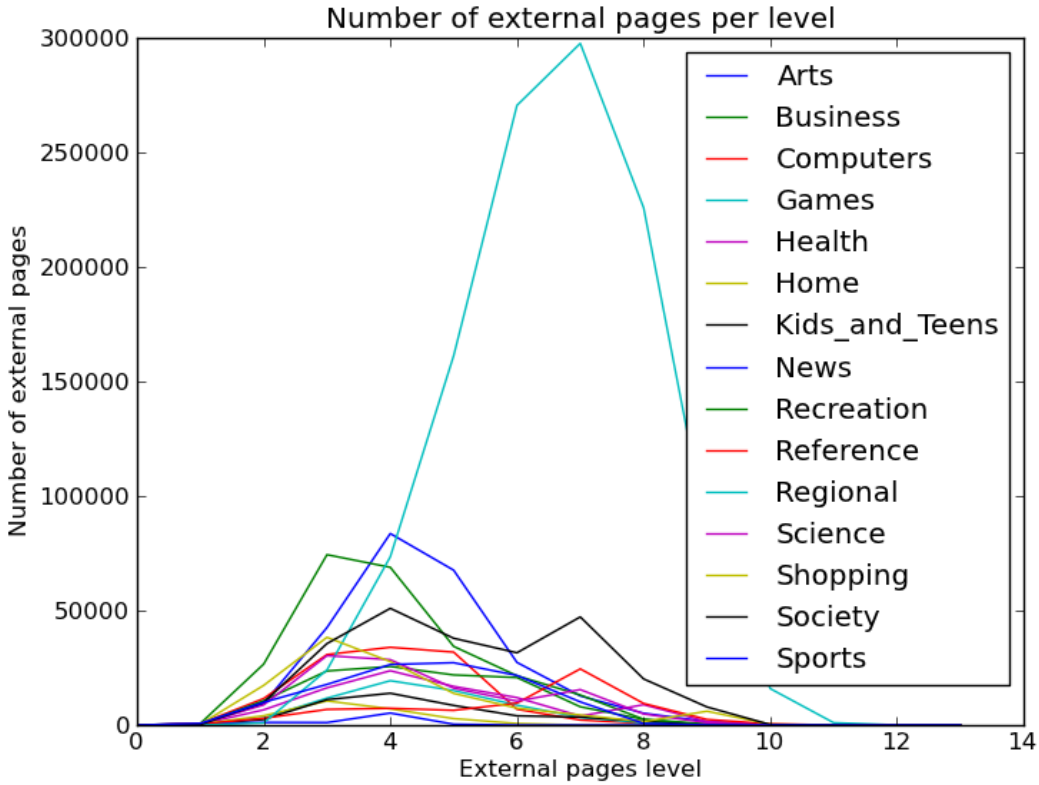

Figure 21: Number of external pages depth levels

7.1.1 Indexing (text features extraction)

The pseudo code for this process (indexing), modified from [123], reads as follows:

a) Read in a document, returned as a row from database query in our case (until last row)
b) Split individual document into tokens (defined by a delimiter, e.g. space between words)
c) Remove:
    a. special HTML code
    b. punctuation signs
    c. HTML tags

d. male/female first names
e. single alphanumeric characters
f. remove stop words

d) Stem the resulting tokens

These steps ensure that the final dimensionality of the data model will be significantly reduced. The result of this process, presented in Table 9, follows.

Table 9: Tokenizing and stemming results example

Sentence	***Processed results***	
Original sentence	This category covers all collectible card games (known as CCGs) such as Magic, Mythos, Star Wars, etc. It also includes games of a similar nature, even if they aren't strictly speaking collectible - XXXenophile, the newer INWO expansions, etc. As a rule of thumb, any game that involves professionally-printed cards (i.e. not standard playing cards) should be in here	
Tokenized sentence	['This', 'category', 'covers', 'all', 'collectible', 'card', 'games', '(', 'known', 'as', 'CCGs', ')', 'such', 'as', 'Magic', ',', 'Mythos', ',', 'Star', 'Wars', ',', 'etc.', 'It', 'also', 'includes', 'games', 'of', 'a', 'similar', 'nature', ',', 'even', 'if', 'they', 'are', "n't", 'strictly', 'speaking', 'collectible', '-', 'XXXenophile', ',', 'the', 'newer', 'INWO', 'expansions', ',', 'etc.', 'As', 'a', 'rule', 'of', 'thumb', ',', 'any', 'game', 'that', 'involves', 'professionally-printed', 'cards', '(', 'i.e.', 'not', 'standard', 'playing', 'cards', ')', 'should', 'be', 'in', 'here', '.']	
Removed stop words	['category', 'covers', 'collectible', 'card', 'games', '(', 'CCGs', ')', 'Magic', 'Mythos', 'Star', 'Wars', 'etc.', 'includes', 'games', 'nature', "n't", 'strictly', 'speaking', 'collectible', '-', 'XXXenophile', 'newer', 'INWO', 'expansions', 'etc.', 'rule', 'thumb', 'game', 'involves', 'professionally-printed', 'cards', '(', 'i.e.', 'standard', 'playing', 'cards', ')']	
WordNet Lemmatization	*Lancaster stemmer*	*Porter stemmer*
['category', 'cover', 'collectible',	['categ', 'cov', 'collect', 'card',	['categori', 'cover', 'collect',

'card', 'game', '(', 'CCGs', ')', 'Magic', 'Mythos', 'Star', 'Wars', 'etc.', 'includes', 'game', 'nature', "n't", 'strictly', 'speaking', 'collectible', '-', 'XXXenophile', 'newer', 'INWO', 'expansion', 'etc.', 'rule', 'thumb', 'game', 'involves', 'professionally-printed', 'card', '(', 'i.e.', 'standard', 'playing', 'card', ')']	'gam', '(', 'ccgs', ')', 'mag', 'mytho', 'star', 'war', 'etc.', 'includ', 'gam', 'nat', "n't", 'strictly', 'speak', 'collect', '-', 'xxxenophile', 'new', 'inwo', 'expand', 'etc.', 'rul', 'thumb', 'gam', 'involv', 'professionally-printed', 'card', '(', 'i.e.', 'standard', 'play', 'card', ')']	'card', 'game', '(', 'CCG', ')', 'Magic', 'Mytho', 'Star', 'War', 'etc.', 'includ', 'game', 'natur', "n't", 'strictli', 'speak', 'collect', '-', 'XXXenophil', 'newer', 'INWO', 'expans', 'etc.', 'rule', 'thumb', 'game', 'involv', 'professionally-print', 'card', '(', 'i.e.', 'standard', 'play', 'card', ')']
Sentence reduction:	*~47%*	

Based on the testing data, returned as the results of the previously defined SQL query, the overall reduction of the number of words is approximately 47%, which shows that by using the steps in the presented pseudo code one can achieve a significant dimension reduction for further analysis.

7.1.2 Proposed classification models and data preparation

The data available in ODP is categorized in 17 main categories, 15 of which were taken into consideration. Furthermore, each category data is described in different depth levels, where each level proposes a more detailed tagging possibility. The prepared data serves as input for the evaluation modes. All models, as the weighting schemes are concerned, use previously defined evaluation metrics and are based on tf-idf model. Several different approaches in creating and testing the prepared models have been implemented with results shown in latter sections of this chapter. A brief overview follows, with detailed overview including evaluation results presented in individual chapters.

The input to each of the different models tested in this hypothesis is textual data prepared with the steps presented above. Each of the approaches has two main models that are taken into account:

- *Level based model*, where each model is created based on the root category and additionally depth level of the category with level 2 being the initial level. These models are furthermore expanded upon with grouping and percentage schemes.

- *Limit based model,* where each model is based on a number of documents used in its creation. These models are furthermore expanded upon with grouping scheme.

Grouping schemes utilized in model creation are based on the structure of the original ODP data and are as follows:

- General grouping, where a single document in model is represented by a single document in the database
- CATID grouping, where a single document in model is represented by all documents with the same CATID value
- FATHERID grouping, where a single document in model is represented by all documents with the same FATHERID value

Models are created based on two approaches:

1. Percentage schemes utilized in model creation are:
    - 25% models
    - 50% models
    - 75% models
    - 100% models

2. Limit schemes utilized in model creation are:
    - 1,000 documents
    - 2,500 documents
    - 5,000 documents
    - 7,500 documents
    - 10,000 documents
    - 20,000 documents

Both percentage as well as limit schemes use the grouping schemes and are evaluated based on them. The purpose of different grouping and model schemes is to test:

1. if ODP is a good source for context based personalization model and

2. if there are differentiations in classic IR measurements between different model creation techniques

Model creation utilizes gensim and transformations that are available in the gensim module and were previously introduced. Gensim supports several vector models that are used in information retrieval and offers their implementation in python language. tf-idf has been used in this work. Model creation process is as follows (with the code presented below):

i. prepare input data, taken from the ODP database, as described in chapter 7.1.1
ii. create dictionary, with the list of all tokens/words taken from the database
iii. create corpora
iv. create vector model representation

The difference between models is defined in the first step. Files, created as the result of this stage, are then used in testing and model evaluation.

7.1.2.1 Evaluation process

In order to evaluate the created models and determine the possibility of using ODP as a universal taxonomy, two testing schemes were devised. The overall steps for model testing are the same for both testing schemes and follow the next steps:

1. Get n sample documents from ODP database
2. Prepare the sampled documents (following the steps described in 7.1.1)
3. Get active testing model files
4. Calculate similarity of each sample document versus prepared similarity index with the following constraints
    a. Get top 1,000 similar documents from model for active document
    b. Filter out documents with similarity value below a defined threshold
5. Evaluate results, with the following two approaches:
    a. To test the overall classification quality, a set of documents from category X was tested against all created models in order to get the most similar model/category (with the results stored in a MySQL database table)
    b. To test the best data organization inside a category, a set of documents from category X was tested against all created models for category X. As the classification results measures, standard IR measures Precision, Recall and F1

were calculated based on the implementation available from *scikit-learn* Python module (with the results stored in a MySQL database table)

These evaluation approaches answered the two research question regarding the utilization of ODP as the basis for the proposed universal taxonomy.

7.2 Module overview

The purpose of this module is to create a universal taxonomy for the purposes of content classification. This module presents the offline part of the proposed system. The tf-idf model data is stored and used as needed. The data for this purpose is retrieved from the ODP database, presented in 7.1 and is stored locally. The ODP data is unstructured and requires additional processing before it can be used as the foundation for the categorization. The overall functionality of this module is presented in Figure 22.

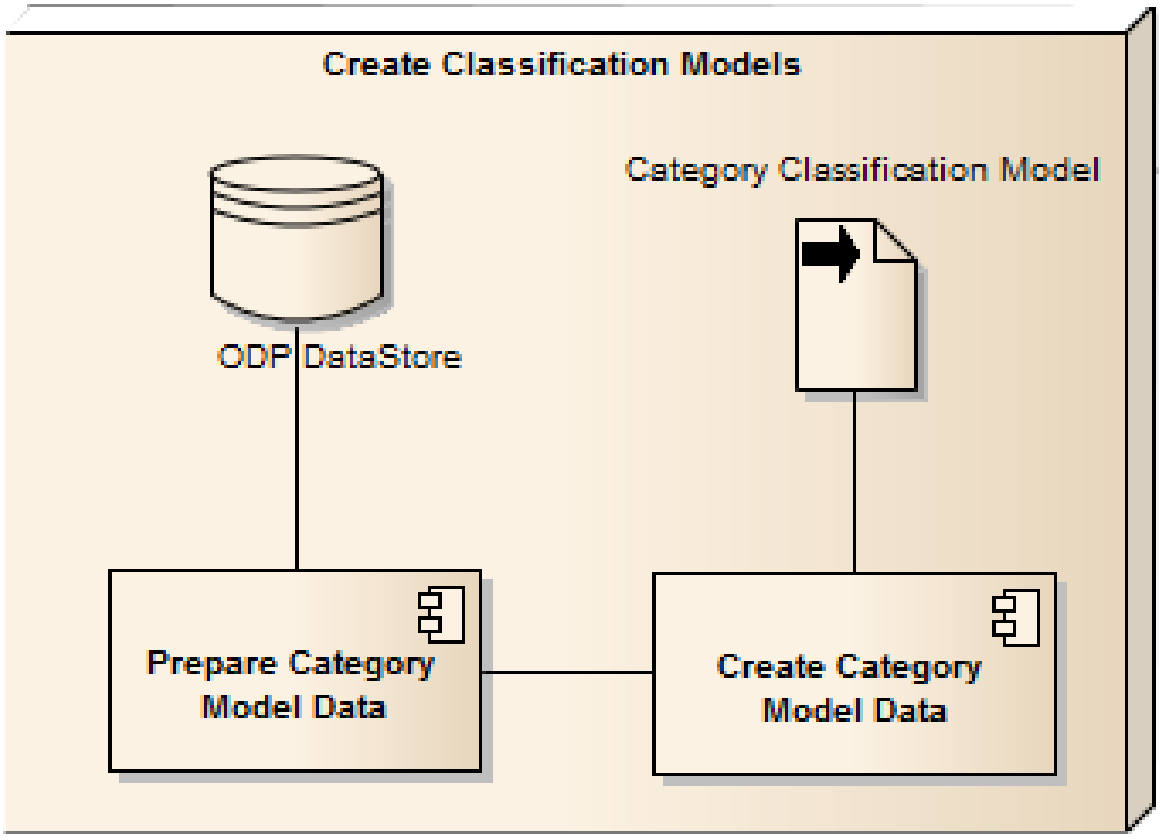

Figure 22: Create Classification Model module

The functionality is defined in four elements

a) *OPD DataStore*, as presented in the previous chapter

b) *Prepare Category Model Data*, whose main goal is to filter out data from a) that match the set criteria (namely, data belonging to the set category)

c) *Create Category Model Data*, that makes all the necessary conversion on the selected data and calculates the needed *gensim* files for further similarity calculations

d) *Category Classification Model*, defined as locally stored files that are used in further classification of new items

The two main steps are defined in steps b) and c) and their decomposition follows. Create Category Model Data receives, as input, raw data from ODP DataStore and performs the necessary actions to prepare the data for further processing. These steps are presented in image 27 and include:

- Get documents from DataStore that fit the selected criteria (a set main category)
- Limit the number of documents based on the defined model limit (as explained in 7.1.2)
- Group documents (as defined in 7.1.2)
- Clean prepared documents

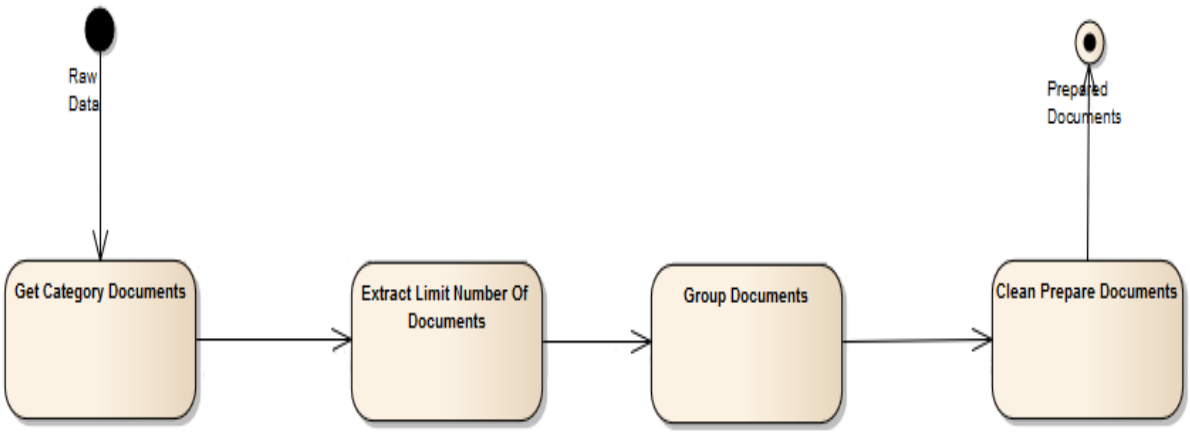

Figure 23: Prepare data for categorization

The most important activity in the above presented steps is the cleaning process, where the raw, unstructured data available in the ODP database dump is conformed to the available format and made available for further processing. The activity is decomposed with the help of the sequence diagram presented in Figure 24.

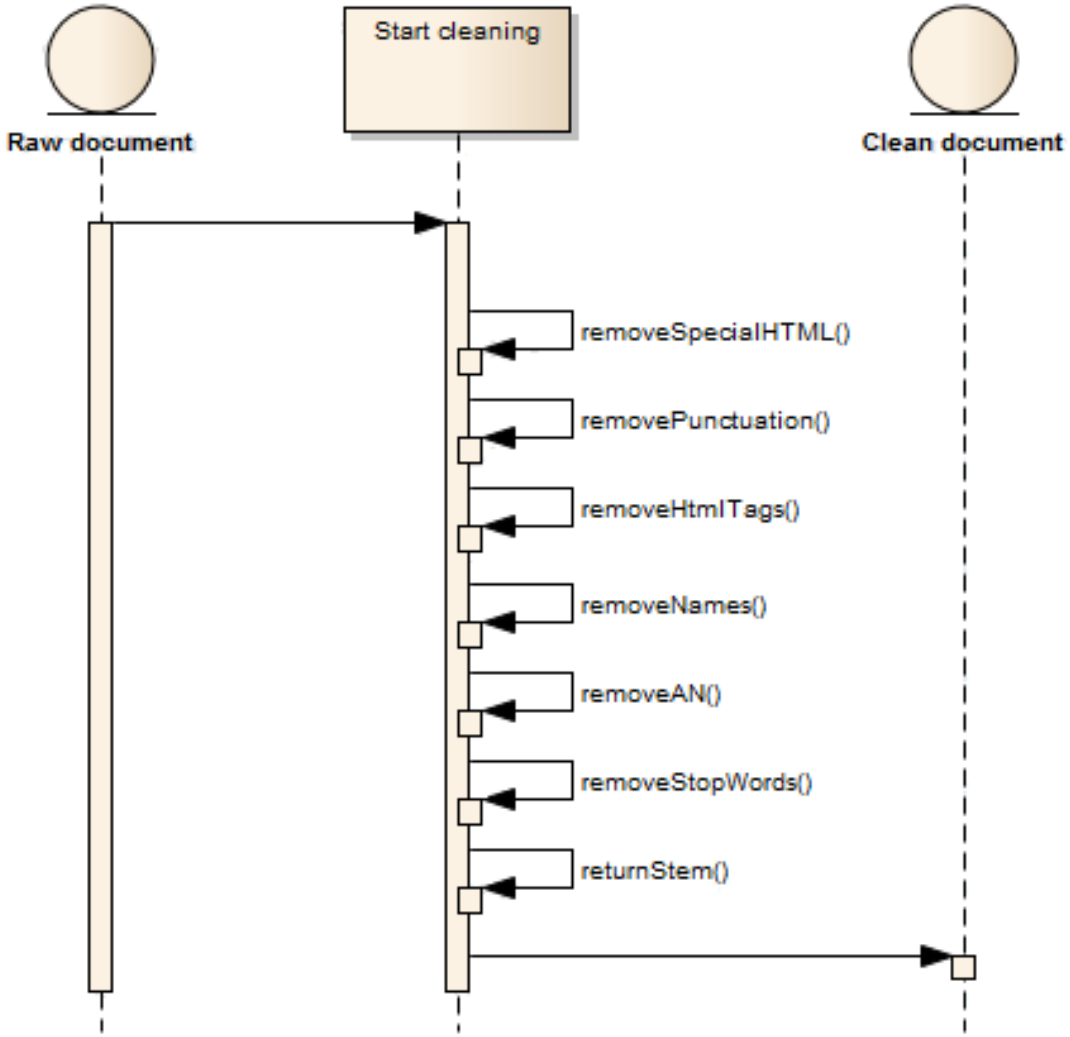

Figure 24: Document cleaning process

7.3 Evaluation results (H1.1.)

This chapter deals with the hypothesis H1.1 of this thesis defined as:

H1.1. *The application of ODP taxonomy will create a unified taxonomy categorization of the existing information nodes to one or more thematic units*

The experiments were focused on testing two different parts of the first hypothesis of the dissertation:

i. the justification of using ODP as the basis for the universal taxonomy (presented in 7.3.1)
ii. the dissemination of the labeling process through different devised model creation schemes (presented in 7.1.2)

7.3.1 Category classification scheme

The created classification models are differentiated based on two criteria:

- Number of original documents from each category taken into consideration in the creation of models
- Documents grouping scheme based on a hierarchical connection between two distinct documents, describing the same category, from the ODP dataset.

The purpose of these evaluations is to determine the overall quality of the ODP based contextual models for the proposed universal taxonomy. The goals are to determine the following:

1. The overall quality of the ODP based models for the proposed universal taxonomy
2. The differences between the previously defined models
3. The selection of the optimal model scheme for further steps

For these purposes a simple testing scheme was derived and implemented where, based on *n* extracted documents form category *X*, a set of documents was evaluated against every created model. The evaluation results, defined as the overall cumulative similarity value returned through the *gensim* module, were extracted and stored in a MySQL database for further processing; the stored data tested what model, based on the overall sum of all similarity values, was returned as the most similar model and therefore what category was shown as the most similar one from all 15 possible categories. The overview results are shown in Figure 25.

These thresholds provide a better understating of ODP as a classifier and the possibility of using detailed ODP based information for the classification process. Preliminary analysis shows differences between the two described approaches in creating classification models. A more detailed result analysis will determine which of the approaches is the best for further steps. When it comes to the proposed grouping schemes, the grouping based on CATID value showed the worst results with the lowest number of pairs with the came (category, mostSimilarCategory) values.

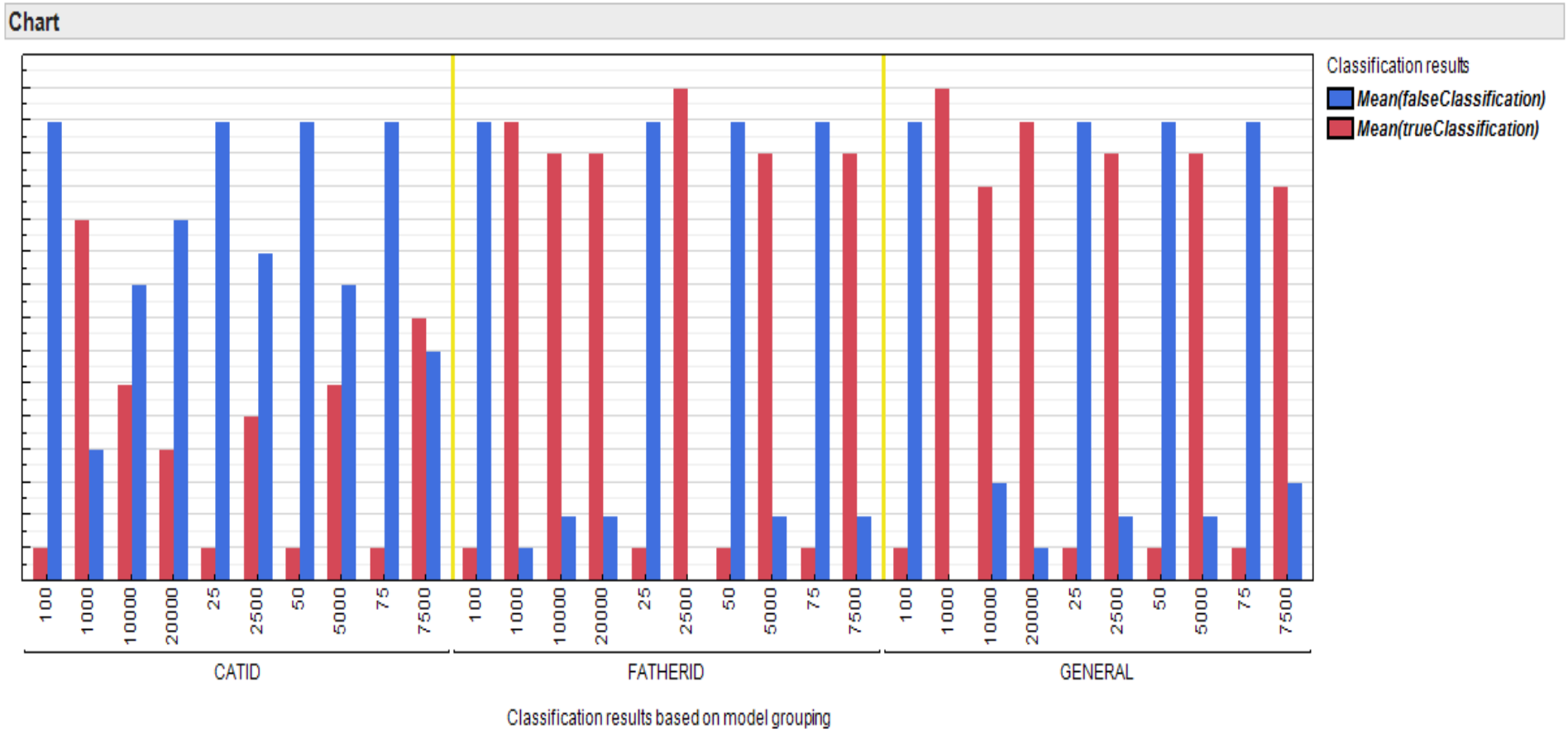

Figure 25: Categories similarity overview

When it comes to the proposed and tested content models, the overview showed that the percentage models are behaving subpar and actually, due to the different number of documents they are made of, insert more noise than good data into the models. Models based on the limited amount of data in them provide far better results but also, depending on the number of documents used to create specific model(s), based on the given results, it is best to use limit models where documents are grouped in FATHERID column. Hence, the data showed that when it comes to basic classification between models and categories, the percentage models are not to be used in further steps.

7.3.2 Deep classification and labeling process

Once an unclassified document is categorized, as explained in the previous chapter, the system proceeds to the next step. The goal of this process is to finely tune the classification and to apply automatic labels to the active document, as well as to test the quality of ODP as a possible labeling scheme. For that purpose, different model creation schemes as well as tests have been devised; all based on the standard precision, recall and F1 measures as explained in chapter 4.2.

As mentioned earlier, three different data grouping schemes have been deployed in model creation:

1. GENERAL, where model input data had a cardinality of 1:1, where each row in the created model corresponds to an individual row data from dmoz_externalpages.
2. CATID, where model input data had a carnality of 1:n, where rows sharing *dmoz_externalpages.catid* values correspond to a single document in the created model and
3. FATHERID, where model input data had a carnality of 1:n, where rows sharing *dmoz_externalpages.fatherid* values correspond to a single document in the created model

Along with the overall grouping schemes, the tested models were also divided based on the percentage of data taken into consideration as well as the number of documents taken into consideration during the model creation process. The models have been introduced in detail in chapter 7.1.2.

To measure the quality of the created models as a possible labeling scheme, three different measurement practices have been devised:

- *absolute measure*, where the most similar document, from all returned, is taken into consideration; evaluation measures are calculated based on 1:1 evaluation
- *relative measure*, where all returned documents either with similarity value equal to 1 or the id searched for are returned; evaluation measurements are calculated based on 1:M evaluation
- *exclusive measure*, where the analysis is done on the top 10% of the returned documents and the similarity is measured based on the overlapping input/output document IDs; evaluation measurements are calculated based on 1:N evaluation.

The evaluation results will be presented next.

7.3.2.1 Absolute similarity measures results

The first line of deep classification analysis is the most restrictive approach: from all returned documents (with similarity higher than the set threshold), take the most similar document. If the input and output document's ID values are equal, then the classification can be considered successful.

Figure 26 presents the results of this comparison. As the results show, even with strict rules (e.g. testing the returned similarity results only on the first returned row) the ODP based models can be used for deep classification/labeling. As the above graph shows, there are differences between different model creation approaches.

As far as the implemented grouping schemes are concerned, there are obvious differences between the three proposed approaches. The data and analysis results show that the best approach is the approach where no grouping scheme (row GENERAL in Figure 26) is implemented and every ODP document is represented as an individual element of the created model. The other two approaches, where each element of the model is created based on all original documents sharing either CATID or FATHERID values (rows CATID and FATHERID in Figure 26, respectively), the returned values are worse than no grouping scheme.

Figure 26: Absolute similarity measures

As the models are also divided based on the number of documents used in the created models, the results show that the proposed percentage models (columns 25, 50, 75 and 100 in Figure 26) are returning poor results as far as recall and F1 measures are concerned, while the precision results are satisfactory.

The limit based approaches (columns 1,000, 2,500, 5,000, 10,000 and 20,000) proved to be better approaches as far as the model content selection is concerned. The percentage models also show, naturally, that smaller models perform worse than bigger models (with 25% < 50% < 75% < 100%) but only 100% models are on par with the biggest level model (based on 20,000 documents). Taking into consideration the needed memory and time calculation sizes, it is easy to conclude that the best approach would be the biggest level model instead of the biggest percentage model.

Although this approach is the strictest one, the results are promising as far as deep classification/labeling is concerned. The results are used as a guideline in the practical implementation of this system as far as similarity calculation is concerned.

7.3.2.2 Relative similarity measures results

Additional labeling approaches have been devised with relative similarity and one of the approaches is presented here. While the previously mentioned approach, absolute similarity, takes into consideration only the first (the most similar) document and calculates performance and correctness measures based on those results, this approach is based on top *n* documents (with the default *n* value set at 1,000; e.g. 1,000 most similar documents).

The purpose of these tests is to see the quality of deep classification based on ODP in terms of both the correct results as well as the false results that have maximum similarity values (equals to the value 1.0).

Figure 27: Relative similarity measures

The difference from the previously presented testing is in the way it summarizes the results of the classification. The steps are as follows:

i. Prepare the input document (as presented in 7.1.1)
ii. Compare the prepared document to the selected model similarity index
iii. Get all documents from calculated similarities with maximal similarity (similarity = 1.0) or with matching id value to the input document
iv. Append *(inputDocumentID, returnedDocumentID)* to the results array
v. Calculate IR measures

The purpose of this process is to see how much the false classification influences the results of the absolute measures. Once again, the research results show that:

- Grouping approaches show similar results as the absolute calculations; FATHERID based models show the worst results, followed by CATID. Once again, GENERAL grouping scheme showed the best results
- As far as the content of the models, of the four proposed percentage models, only 100% models show similar results to the limit based models; as the limit based models are much smaller than the best percentage model (size/number of documents) the results suggest the use of GENERAL based, limit models

Graphical presentation of the results is presented in Figure 27.

7.3.2.3 Exclusive similarity measures results

The third set of tests for testing ODP as a potential deep classifier was devised based on *n* % (currently set to *n = 10*) of returned similarity results. The accepted similar documents are only the ones that, in the defined *n %,* have the same input and output document identifications.

The steps are as follows:

i. Prepare the input document (as presented in 7.1.1)
ii. Compare the prepared document to the selected model similarity index
iii. Get n % documents from calculated similarities with matching output id value to the input document

iv. Append *(inputDocumentID, returnedDocumentID)* to the results array

v. Calculate IR measures

The results, presented in Figure 28, show the following:

- Grouping approaches show similar results as the absolute and relative similarity calculations; once again, FATHERID based models show the worst results, followed by CATID and GENERAL grouping scheme; once again, with the grouping in mind, GENERAL grouping is proven to be the best way to combine documents and create models
- Content wise, the models that are based on the set limit are once again better in evaluation as the models that are based on document number percentage; limit models are suggested by the results for further use.

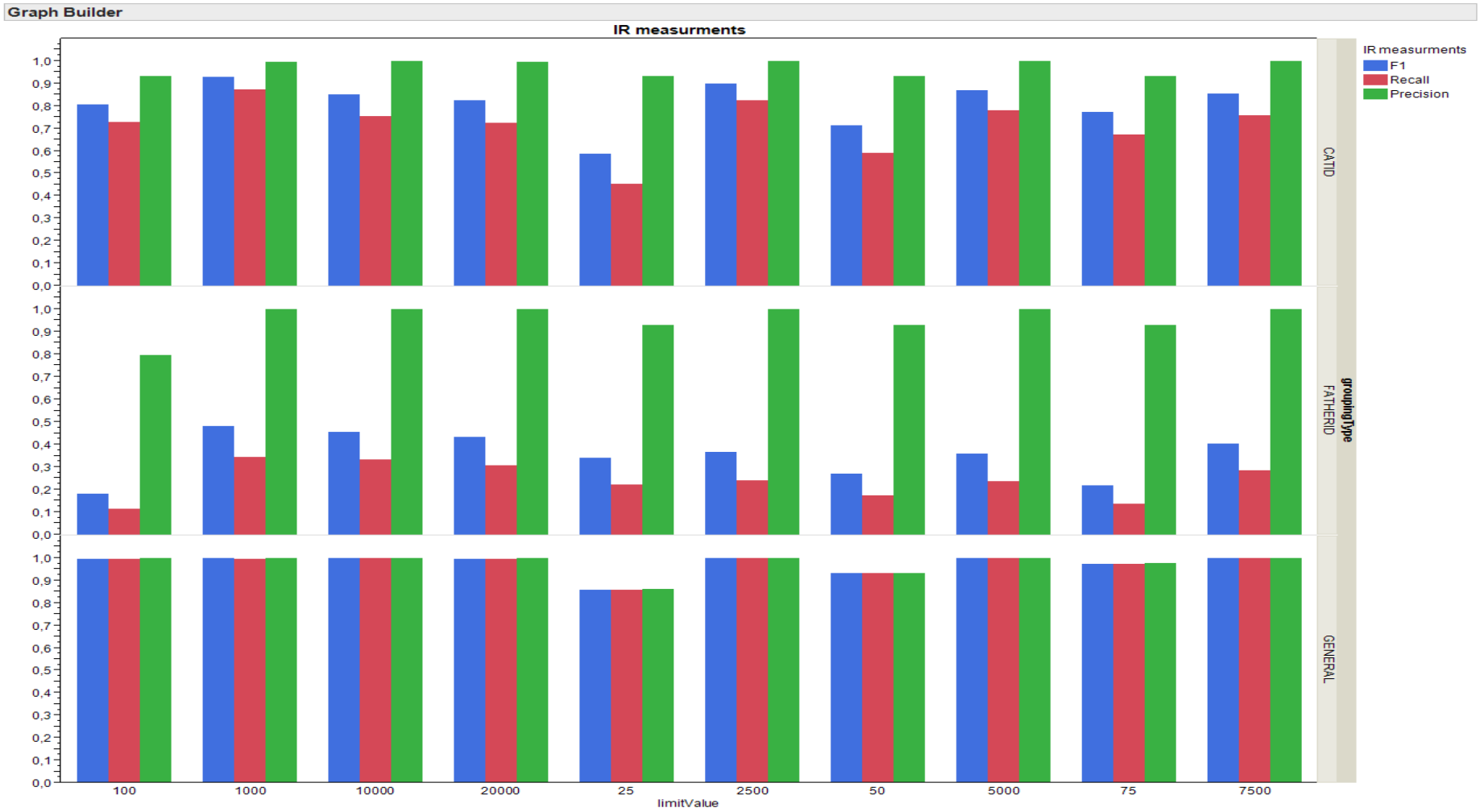

Figure 28: Exclusive similarity measures

7.4 Proposed system for creating virtual contextual profile (H1.2)

This chapter deals with the hypothesis H1.2 of this thesis defined as:

H1.2. *The descriptive system for creating virtual contextual profile will be defined by using the results from (H1.1)*

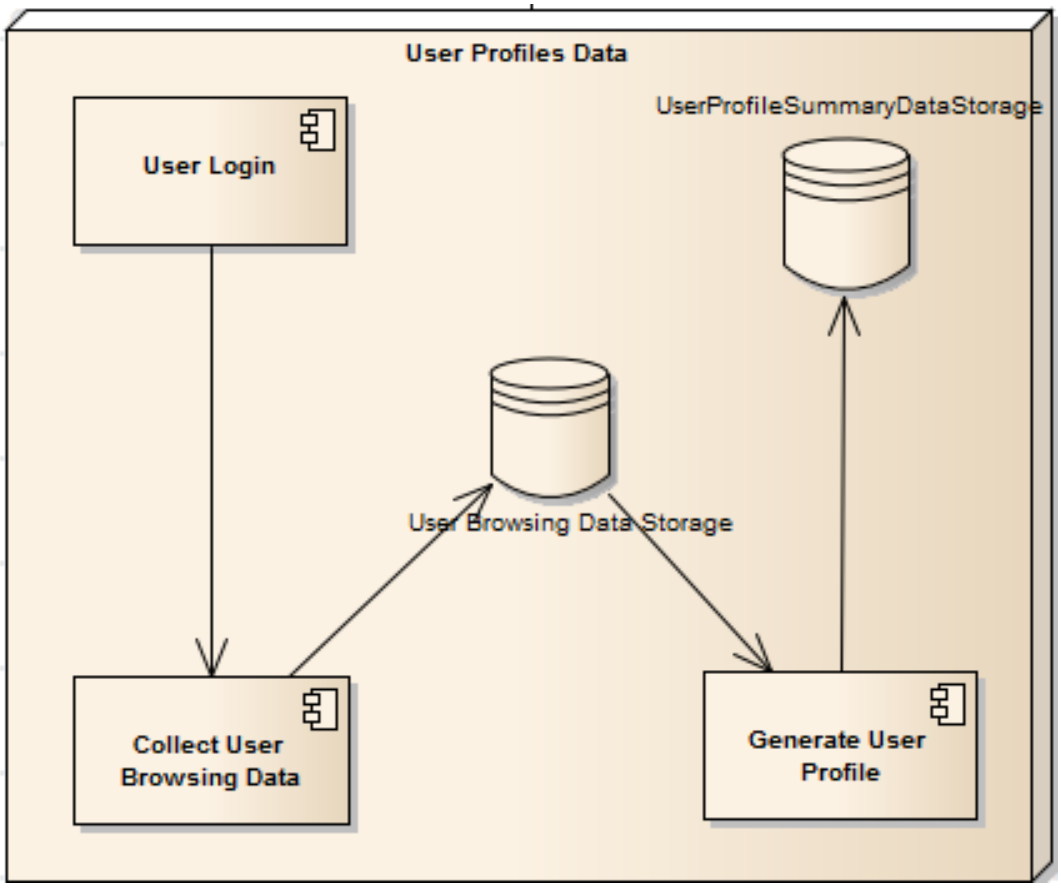

Figure 29: Virtual Contextual Profile Module overview

Contextual user profiles are based on the context of single user interests which is gathered from user browsing history data. The proposed system for creating virtual contextual profiles is defined based on the results presented in 7.3 (and its subchapters). The sub model is presented in Figure 29: Virtual Contextual Profile Module overview. All the needed elements of the sub model are defined here:

Definition. Let $U = \{u_1, u_2, \ldots u_n\}$ denote all user sessions in the collected data.

Definition. Let $L = \{l_1, l_2, \ldots, l_m\}$ denote all available web links in the proposed system.

Definition. Let $S_n = \{s_{n1}, s_{n2}, \ldots, s_{ny}\}$ denote all session data for user n, with K = {l1,l2,…,lk; k ≤ m}; $S_{ni} \subseteq K$, i=1,...,k; $K \subseteq L$

Definition. Let $AS = \{S_1, S_2, \ldots, S_n\}$ denote all recorded sessions across all users.

A single user, u_i, is described by a set of sessions. Each single session is described by a set of visited links. A set of sessions represents all links made available by the personalization system and visited by user u_i.

Each available link is described as:

$$l_n = \{w_1, w_2, w_3, \ldots, w_{15}\},$$

Equality 31: Individual link weight description

where w_1 through w_{15} represent weights for each category, based on the previously described categorization scheme and created tf-idf models.

User interests are defined as:

$$u_i = (W_1, W_2, \ldots, W_{15})$$

Equality 32: Definition of user interests

where each W_i denotes the average weight for category i, with average weights defined as the average value over set of links, calculated as:

$$W_i = \frac{\sum_1^k w_i}{k}$$

Equality 33: Average user interest in category i

where k denotes the number of documents over which the average weight for category i is calculated. This represent the average weight that category i has for the user n (hence, the average interest in category i for user n).

7.5 Result analysis

The above mentioned experiments were conducted with the goal of testing both categorization and deep classification processes and the quality of ODP-based tf-idf models as the classifiers. Based on the above presented results, ODP based classification has proven to be a quality way of horizontal and vertical classification and can be used in additional steps that are going to be presented in the next chapter(s) of this dissertation.

The overall goal was to determine if ODP and taxonomy based on ODP can sever as the foundation for the proposed unified classification and to see if it is possible to use it as a labeling tool. Furthermore, additional tests have been made in determining what modeling approach produces best results, as the amount of documents and their grouping are concerned. For this purpose several models, differing on the number of documents in the model, as well as the way documents were grouped together were devised (presented in section 7.1.2).

When it comes to the classification process first the overall adequacy of ODP as the proposed taxonomy was evaluated by testing if an original document will be classified in the originating category or if it will be label as a member of some other category (Figure 25). The results show that, as far as the overall classification quality of ODP is concerned, this approach gives very good results. From the proposed models, when it comes to the way of grouping documents together, the best results provide the GENERAL grouping model where each model equals to one document in the model.

When it comes to deep classification, which is used to determine the difference between different category models regarding the amount of documents used as category models input. The results show that limit models perform better than percentage models. This can be explained with the fact that each category model, when constructed via the limit method, has the same amount of data compared with percentage models which have different amount of documents used as basis for respective category model.

Additionally, a model of generating contextual user profiles, based on ODP and the created taxonomy is also presented as well as implemented in ShevaVIRT[40].

[40] https://github.com/deakkon/SemanticVIRT

8. Pattern extraction evaluation (H2)

This chapter deals with the second hypothesis of this thesis defined as:

H2: *In the information space, in the domain of web portals, it is possible to extract a unique pattern for individual user navigation through the information space.*

The data was collected from two sources for the following reasons: uLogLite data adds an entry to the collection for every interaction and the Mozilla Firefox browsing history data adds a new entry to its dataset when a new link is opened in the browsing window. Using the available information from both sources, it is easy to reconstruct how the user travelled from point A to point B in the web site. First, simple statistical measures will be done that will give the averages for the data. The total number of web pages visited (based on the data collected from Firefox history data) is 4,807 web pages dispersed in 200 sessions (20 users with ten sessions each). Overall, the users visited 4,807 web pages with 2,524 of unique visits. The average statistical information gathered form that data give as the mean value at 24.5, standard deviation value at 14.24 and median value at 21. Based on the uLogLite collected data one can identify pairs of user actions as presented in Table 10.

Table 10: Identified Actions Pairs

Starting action	Ending action	Pairs to
Logging started	Logging stopped	*Beginning/End of session data collection*
Left mouse button pressed	Left mouse button released	*Left mouse button click*
Mouse wheel	Mouse wheel stopped	*Mouse wheel*
Right mouse button pressed	Right mouse button released	*Right mouse button click*
Middle mouse button pressed	Middle mouse button released	*Middle mouse button click*
Key pressed	-	*Key pressed*
Left mouse button doubleclick	-	*Left mouse button click*

Based on pairs of action denoting start and end of each action, this is identified: 11,014 left mouse button usages, 6,229 mouse wheel uses, 3,546 keyboard pressings, 253 right mouse button uses, 448 middle mouse button uses and 48 left mouse button double-click action (which will be identified as single left mouse button click in further analysis). The basic statistical data shows the mean value at 55.07, standard deviation at 49.014 and median at 46

for left mouse button use and mean value at 31.145, standard deviation at 20.16 and median value at 37 for mouse wheel use. Other identified actions will not be taken into account in the development of the model since their participation in the data did not prove to be significant enough. The main focus will be in the two user actions that are used most often. Apart from the type of action, each log input is additionally marked with log time information, the title of the application and window causing the log input, the position on the X and Y screen coordinates of the action and relative and total travelled distance. By using the available information one can calculate additional behavioral information for each user, which will be the building block for user behavioral model. First, action duration and idle times from the logger time information were calculated. By combining time information for consecutive X pressed and X released actions (x denoting one of the actions presented in Table 10) and by calculating the time difference between them one gets the duration of each action and the position in the timeline (a value between 0 and 30 minutes, converted to milliseconds). Using the timeline offers the possibility to track user behavior through time, and follow his interaction both with the screen and the content. The opposite time is the time users spent doing nothing and engaging with the content presented on the screen. It is called idle time. The idle time was derived once again from the uLog Lite collected data and was measured as the time between actions. This was calculated by determining a difference in time between the neighboring end actions and starting actions. The data is presented in Table 11. All data is in milliseconds.

Table 11: User actions over time with no user grouping

Action	Sum of time spent	Mean	Standard deviation	Median
Left mouse button pressed	11501733	1044.283	3458.695	140
Mouse wheel	125279849	20057.61	17980.06	13836.5
Idle time	143302633	7794.117	37302.43	2117.5

The data shown in the tables above does not take into account the user/session. Instead, it is based on the assumption gathered from one user. This data gives a rough picture of how a "single user" interacted with the content on the www.cnn.com portal. To get more detailed information it is dissected on the user/session level. Comparing the information gathered for

each user gives more insight in how the user behaved, how to quantify the difference between them and how specific those differences are. The data shows that an average user had the mean value at 550.7, standard deviation at 382.3775 and median value at 489 for left mouse button use and mean value at 311.45, standard deviation at 164.7989 and median value at 389 for mouse wheel use. When one looks at the data per user basis, it shows differences in averages between users and this gives a better understanding of what an "average" user did per action basis. Next, the same is done for the time data, as shown in Table 12.

Table 12: Cumulative user actions over time on action basis with user grouping

Action	Sum of time spent	Mean	Standard deviation	Median
Left mouse button pressed	11501733	575086.7	1495647	105122
Mouse wheel	125279849	6263992	3606488	6643680
Idle time	143302633	7165132	3742212	7681433

Next, look at the average values of each action taken per user basis derived from the cumulative session values of each action. As an addition, the average number of links each user visited during his/her ten browsing sessions will be presented. This information will give a better understanding of the individual behavior and will determine the differences between each user. There are sessions in which the user has 0 actions of a specific type and the time dispersion has not been taken into consideration. The data is shown in table 4. The actions are coded with: 1 Number of Left mouse button pressed; 2 Average duration of action 1 in milliseconds; 3 Number of Mouse wheel; 4 Average duration of action 2 in milliseconds; 5 Average total distance traveled in pixels; 6 Average relative distance traveled in pixels; 7 Visited Links; 8 Average idle time in milliseconds.

The data presented in Table 13 is based on the data gathered through each individual session and is the foundation of determining the future behavior model. One can see that user behavior differs on each of the selected behavioral measurements and can be quantified. That can create and present a model for future use. The differences in behavior can also be seen in figure 1, showing screen focus area over the timeline.

Table 13: Average User behavioral data

Measure / User	1	2	3	4	5	6	7	8
1	84.5	*200.813*	39.3	*16880.86*	105344.4	35823.32	14.9	1015640
2	63.2	*143.4826*	24.5	*18936.39*	93817.7	32069.46	22.7	1242000
3	36.9	*1009.799*	43.5	*27382.02*	102922.1	29714.12	18	605972.7
4	69.3	*730.241*	30.8	*13782.55*	68812.8	26384.14	34.4	403031.9
5	52.7	*149.3074*	41.1	*21877.67*	62948.93	30554	29.3	604093.6
6	60.7	*672.5585*	39.6	*18861.65*	102074.5	37642.99	45.2	848514.9
7	22.9	*267.5546*	13.7	*17949.48*	33357.38	14887.84	29.6	460101.6
8	39.2	*168.3468*	44.2	*24620.8*	84950.63	29240.11	14.1	647962.7
9	21.2	*205.8443*	38.5	*25417.76*	44143.65	17489.7	9.7	863990
10	24.4	*223.2541*	45.4	*25405.6*	41395.54	20480.52	13.3	688292
11	41.7	*1273.005*	12.4	*30838.31*	48325.03	21411.4	12.4	1203201
12	120.4	*734.5897*	46	*14463.44*	80385.23	36280.59	30.1	849102
13	98.2	*329.7332*	49.8	*15605.23*	135521.9	46367.21	36.8	1391352
14	124.5	*5481.118*	0.4	*737.2727*	154575.4	73550.44	26.6	847994.6
15	45.1	*171.5299*	36.3	*21614.48*	69606.79	26005.79	18.4	1031661
16	5.2	*121.0769*	5.2	*17912.36*	9287.894	4009.219	16.7	122716.4
17	122.4	*704.0123*	50.5	*11822.09*	67032.09	32431.59	16.6	437349.6
18	55.1	*216.9927*	41.6	*22133.27*	70064.78	33825.77	30.5	874858
19	4.1	*143.6098*	4.8	*15404.96*	11628.33	3628.032	44.5	42322.33
20	9.7	*169.9485*	15.3	*23205.19*	15632.43	7220.322	21.4	167035.3

8.1 User behavior model

Following the initial statistical analysis, the user model behavior attributes are identified and they will be used in this model. Identified as the information used to differentiate between users, a vector model is presented where each user space is defined through the use of six vectors.

These vectors are numerical representations of the following data, collected and calculated from the collected data:

- *Action taken*
- *Duration*
- *Total traveled distance*
- *Relative traveled distance*
- *Coordinates of the action*
- *Idle time*

The preliminary results of the data collected have shown there are differences in user behavior and that those differences can be quantified just by looking at the averages of the most usual I/O devices interaction. The data has been observed in two ways. First, it was assumed that all the data was collected from a single user. That provided an overlook on what an "average" user would look like in terms of number of actions taken, time it took to make those actions and number of visited nodes. This information was recognized as the most valuable information that gives an overlook of the pure interaction between the user and the screen. The reasoning in the background of what made the users behave in this manner has not been taken into account and will be the subject of further studies and data analysis. The second approach was to look at the data from the individual participant's point of view. The same information was observed to see if there are any differences that can be quantified to include in this behavioral model. The data showed that the quantified differences can be used to show the individuality of user and his behavior.

8.2 User time, category interest and weighting scheme patterns

After the descriptive statistics depicting the basic I/O devices usage, based on the available data (explained in more detail in 2.2 and prepared as explained in 2.3), its use in preparing and extracting time and interest based on user profile is presented in this chapter.

The user profiles are prepared and extracted from the previously prepared data from two available data sources. Both ODP based taxonomy as well as user browsing history data are combined and further processed for the purposes of creating and evaluating the hypothesis

H3. The evaluation measure used in defining the quality of the extracted patterns is Euclidean distance. The pattern creation process is defined in more detail in 7.4.

8.2.1 Methodology

Three different patterns are extracted from the available data:

- *User time patterns*
- *User category interest pattern*
- *User weighting pattern*

All the above mentioned patterns are based on ODP classification and are calculated based on the browsing history data, which has been previously recategorized. Each visited node has been updated with three 15 coordinate points, each describing one of the above mentioned patterns. As presented previously, user browsing session(s) were recorded over a period of 30 minutes, during which information about activated links as well as timing information regarding the duration of stay at each activated link has been recorded. This data is further used in user profiling.

A *time pattern* for user u_i is defined as:

$$T_i = (t_{i1}, t_{i2}, \dots, t_{i15})$$

Equality 34: Time pattern

with t_{ij} defined as the sum of time user *i* spent on category *j* divided by the duration of session. Values of t_{ij} are$[0,1]$.

A *user category interest pattern* is defined as

$$C_i = (c_{i1}, c_{i2}, \dots, c_{i15})$$

Equality 35: User category interest pattern

with c_{ij} defined as the sum of times user *i* visited a node from category *j* divided by the number of activated nodes during his session(s).. Values of c_{ij} are[0,1].

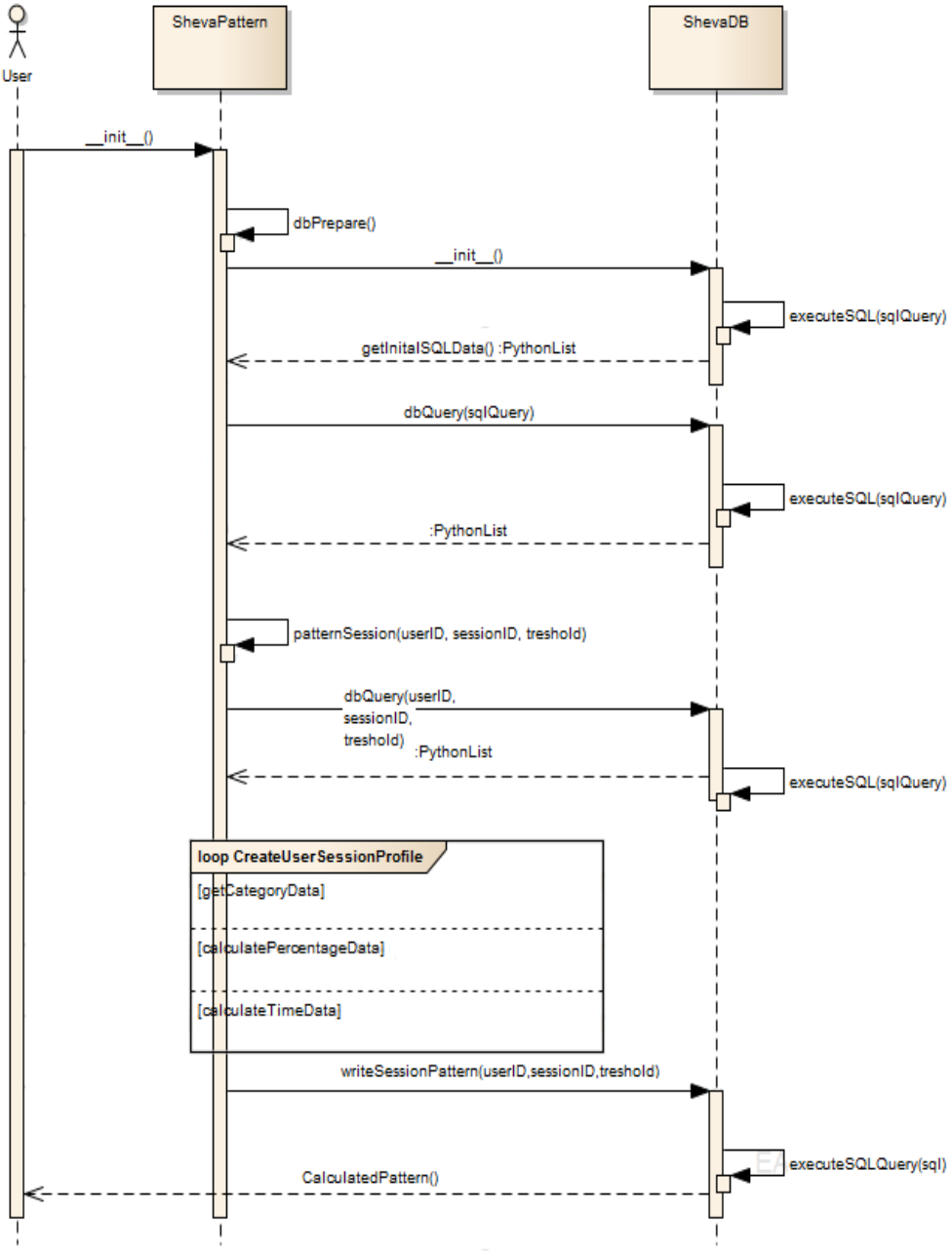

Figure 30: User Pattern Creation Sequence Diagram

A *user weighting pattern* is defined as

$$W_i = (w_{i1}, w_{i2}, \dots, w_{i15})$$

Equality 36: User weighting pattern

with w_{ij} defined as the sum of tf-idf model weights assigned to user *i* visited for category *j* divided by the number of activated nodes during his session(s).. Values of w_{ij} are $[0,1]$.

The proposed recommender system uses user weighting patterns as the basis for its recommendation, with additional information being supplied by user time patterns and user category interest patterns. A detailed graphical overview of the pattern creation process is presented in Figure 30. Time and category interest patterns are used in weighted Voronoi diagrams as respective personalization parameters. The main pattern, the basis for the proposed personalization model, is the user weighting pattern. Next, the analysis results based on user weighting patterns are presented.

8.3 Result analysis

In this analysis, user weighting patterns were evaluated based on the cumulative difference between overall user context profiles. Euclidean distance was chosen as the defining similarity metric.

Euclidean distance is defined as the distance between two points, X $(x_1,x_2,\dots,x_n)$ and Y $(y_1,y_2,\dots,y_n)$ calculated as defined in Equality 5.

For this purpose, two types of generated patterns have been calculated and assessed:

- *difference between sessions for an individual user* and
- *difference between sessions between cumulative user sessions*.

Both evaluations are subject to a different *threshold* parameter value; *threshold* parameter affects the overall unclassified document classification process by filtering out model elements (model documents) with the similarity below the set *threshold* parameter. This affects the outcome of the similarity calculations by excluding documents based on the

calculated similarity. The results of evaluation based on seven different threshold values (.0, .05, .1, .15, .2, .25, .3 respectively) are presented next.

Based on the generated individual session data, the following can be defined about the quality of user patterns. Graphical presentation is available in Figure 31. The evaluation follows these steps:

1. *get all userid data*
2. *split userid data in useridsessionid data*
3. *get useridsessioniddata and compare it with userid data*

Although the purpose of this dissertation is not the definition and testing of the differences/similarities between two distinct users (based on distinct user data evaluation) and between a single user (based on unique users browsing session evaluation), the results will give insight in the future work brought on by the discovered problem areas in this field.

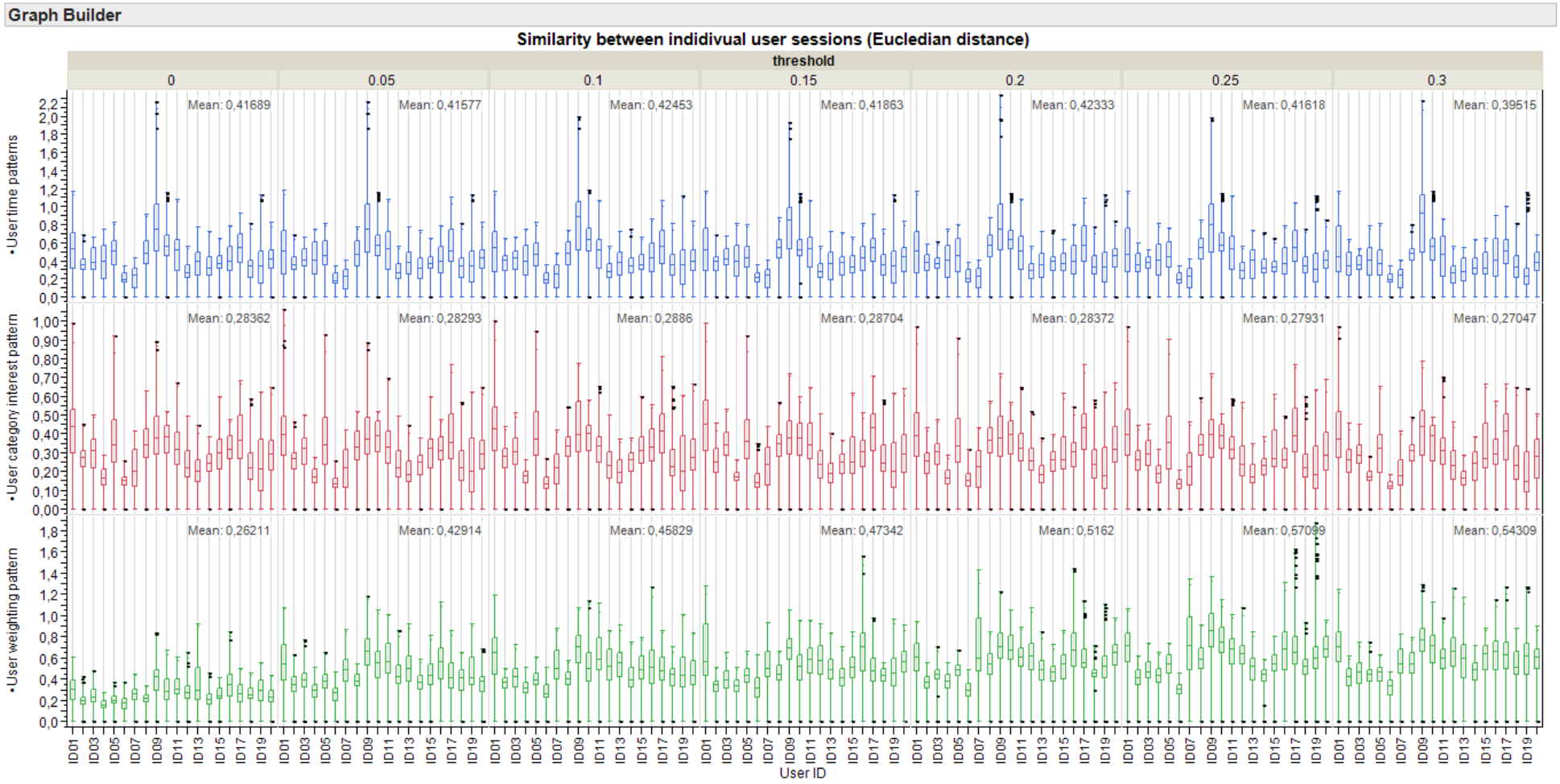

Figure 31: In user session Euclidean distance

8.3.1.1 Individual user browsing session evaluation

Individual user sessions were evaluated based on the previously defined Euclidean distance between numerical descriptors of session based visited documents. For each user, three descriptors were defined. The results are presented in Figure 31.

Time pattern analysis takes into account the amount of time spent on the set of visited documents, each belonging to a single category over the session duration. In the experimental setup, each session's duration was defined as 30 minutes. There are 15 possible categories a document belongs to. This offers an aggregate time session pattern defined in Equality 34., The statistical mean value of calculated Euclidean distances was chosen as the defining measure. The results show that the time structure of the visited links is unchanged over different threshold settings. Users 6 and 7 have the most similar results, over different session, while the majority of users are in the range of 0.35 and 0.45. The most different browsing sessions are for user number 9. The overall mean values, across different threshold parameter setting, are not fluctuating.

Category interest pattern takes into account the number of times a document from a specific category has been visited over the course of user browsing sessions. The category pattern is defined in Equality 35. As with the time pattern series, the results also show that the overall interest in news from the specific category for individual user does not change over the course of different sessions. Once again, the most cohesive user sessions concerning the utilized categorization scheme are the same as with the time patterns; users 6 and 7 have the least different individual sessions while user 9 has the most diverse range of interests. Once again, the set *threshold* value does not influence the created patterns as the mean values are constant over all different threshold settings.

User weighting pattern takes into account document weights assigned to each individual document during the categorization stage. It shows the average weight per category for all the visited documents in individual browsing session. The results show that the smallest difference between all browsing sessions for each user is with the lowest *threshold* value (0) and they increase as the *threshold* value increases. This confirms the assumption that the best results provided by the system are when the most documents from the categorization model are taken into consideration.

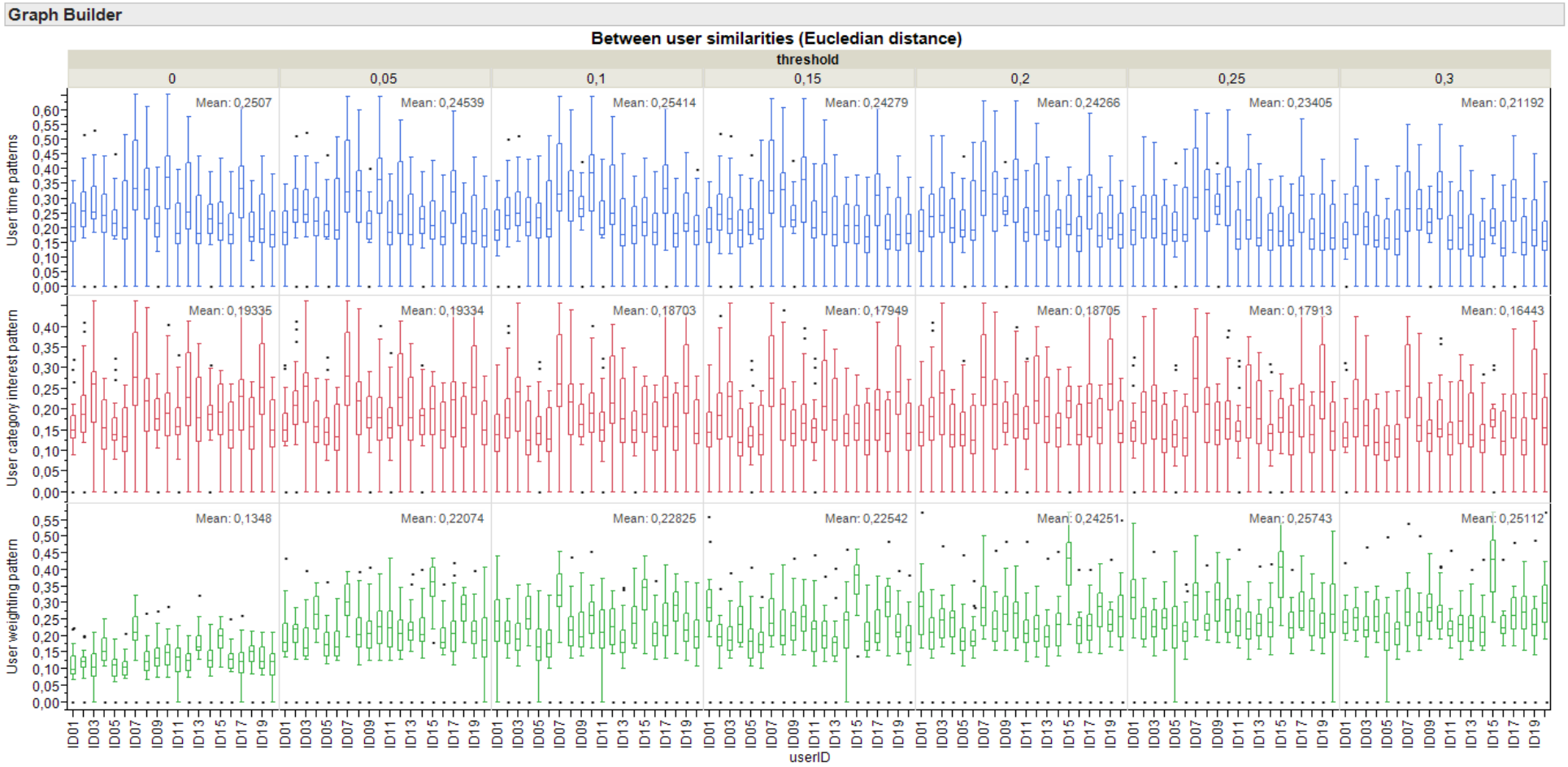

Figure 32: Cumulative user session differences based on Euclidean distance

8.3.1.2 Distinct user session evaluation

The second analysis in this module takes into consideration the differences between the cumulative users sessions, aggregated over all browsing session data gathered for each user. Once again, the system works with three different patterns, presented in 8.2.1.

The aggregated user session is derived from the average user session where the inputs are aggregated data for each session. The results show that the differences for time and category interest patterns do not fluctuate over different *threshold* parameter. As far as the user weighting pattern is concerned the smallest difference between user interests is with the smallest *threshold* value, and the largest difference is with the biggest *threshold* value, although the differences between *mean* values are not significantly large. The results are presented in Figure 32.

This shows that by increasing the threshold values it is possible to get better single user descriptions. Although the similarities and/or differentiations between users does not affect the general 1:1 personalization/recommendation of newly classified articles it does show that, even on a small set of participants in our experiment, it is possible to identify a single user based solely on his interaction with the system.

8.4 Result analysis

The overall goal of this chapter was to identify the available patterns once can extract from data gathered via uLogLite as well as SQLite Mozilla Firefox database files, for the purpose of proving H2. Additionally, extracted patterns were submitted to testing in order to see whether they can be used in differentiating between single users that participated in constructed experiment.

Three patterns were identified of use in this work, all of them based on the way single participant used the I/O devices available (keyboard, mouse) and the way he/she moved through the information space presented in the target web news portal.

The performed tests show:

a) It is possible to extract three distinct patterns gathered by the collection, namely:
    I. *time patterns*, which show the time dynamic of participants dynamic with the chosen web news portal

II. *category interest pattern*, which shows participants interests in information from each of the defined classification categories defined as a ratio between articles read belonging to an individual category and all visited articles (hence, a percentage value) and

III. *weighting pattern,* which shows the average tf-idf based weight assigned to the interest participant showed for each individual category.

b) Individual participant tends to access articles from similar categories over different browsing sessions as shown in Figure 31. There are differences between different sessions for a single participant. Those differences increase with the threshold value which indicates that for the proposed recommendation session and it's recommendation process a threshold of 0 is the best option.

c) When aggregated in a single descriptor, as done in Figure 32, the differences between different uses do exist but they are, for each set threshold, smaller than the differences for individual participant sessions. Although this does not affect the proposed 1:1 recommendation system, it is a clear indication that the pattern descriptions need refinements.

9. Voronoi diagrams implementation and application (H3)

This chapter deals with the third hypothesis of this thesis defined as:

H3: *A new method for data personalization can be described by using weighted Voronoi diagrams.*

An overview of the Voronoi diagrams was given in chapter 5. An overview of the proposed weighted Voronoi diagrams based on personalization is presented in Figure 33.

Definition. Let P = {p_1, p_2, p_n} be all documents in the system and p_j = (w_1,w_2,…,w_{15}) where w_i represents the weight *w* assigned by the categorization module to category *i* and P represents a set of documents.

Definition. Let VP = {vp_1, vp_2,…,vp_m} be all documents that have been visited by the user, where $VP \subseteq P$

Definition. Let NVP = (nvp_1,nvp_2,…,nvp_o), $nvp_i \in (P \setminus VP)$ be all non-user based personalized documents with NVP representing a relative complement of P and VP.

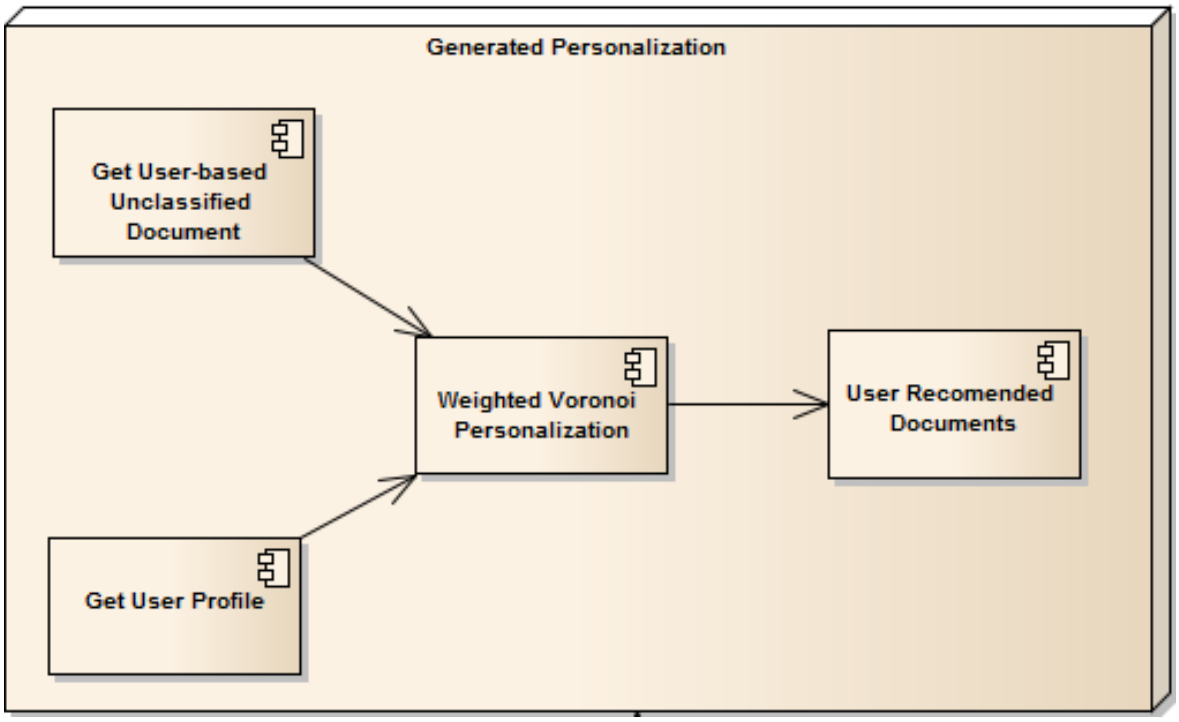

Figure 33: Personalization module overview

The module consists of four main parts, which will be presented next.

Get User-based unclassified documents module retrieves documents that have been unevaluated for the user *i.*

Get user profile module extracts user i's time, category interest pattern and weighting pattern, and prepares the data from the extracted patterns for further use in Weighted Voronoi personalization module. The details about specific user based patterns used in this work are presented in 10.2.

Weighted Voronoi personalization module calculates the Euclidean distance between each document from Get User-based unclassified documents module and weighting pattern, adjusted by the data from time pattern or/and category interest pattern, depending on the type of weighted Voronoi diagrams utilized.

Additively weighted Voronoi diagrams implementation are computed by calculating a defined distance measure (e.g. Euclidean distance) between a point in space (e.g. a news article weighting points) and all defined cell generators modified by subtracting a defined personalization factor (e.g. time and/or category interest value for news article category).

Multiplicatively weighted Voronoi diagrams are computed by calculating a defined distance measure (e.g. Euclidean distance) between a point in space (e.g. a news article weighting points) and all defined cell generators modified by dividing the calculated value by a defined personalization factor (e.g. time and/or category interest value for news article category).

Compoundly weighted Voronoi diagrams are computed by calculating a defined distance measure (e.g. Euclidean distance) between a point in space (e.g. a news article weighting points) and all defined cell generators modified by dividing the calculated value by a defined personalization factor (e.g. time and/or category interest value for news article category). That value is then reduced by a value of the second personalization factor (e.g. time personalization factor value when dividing with category interest personalization factor and vice versa).

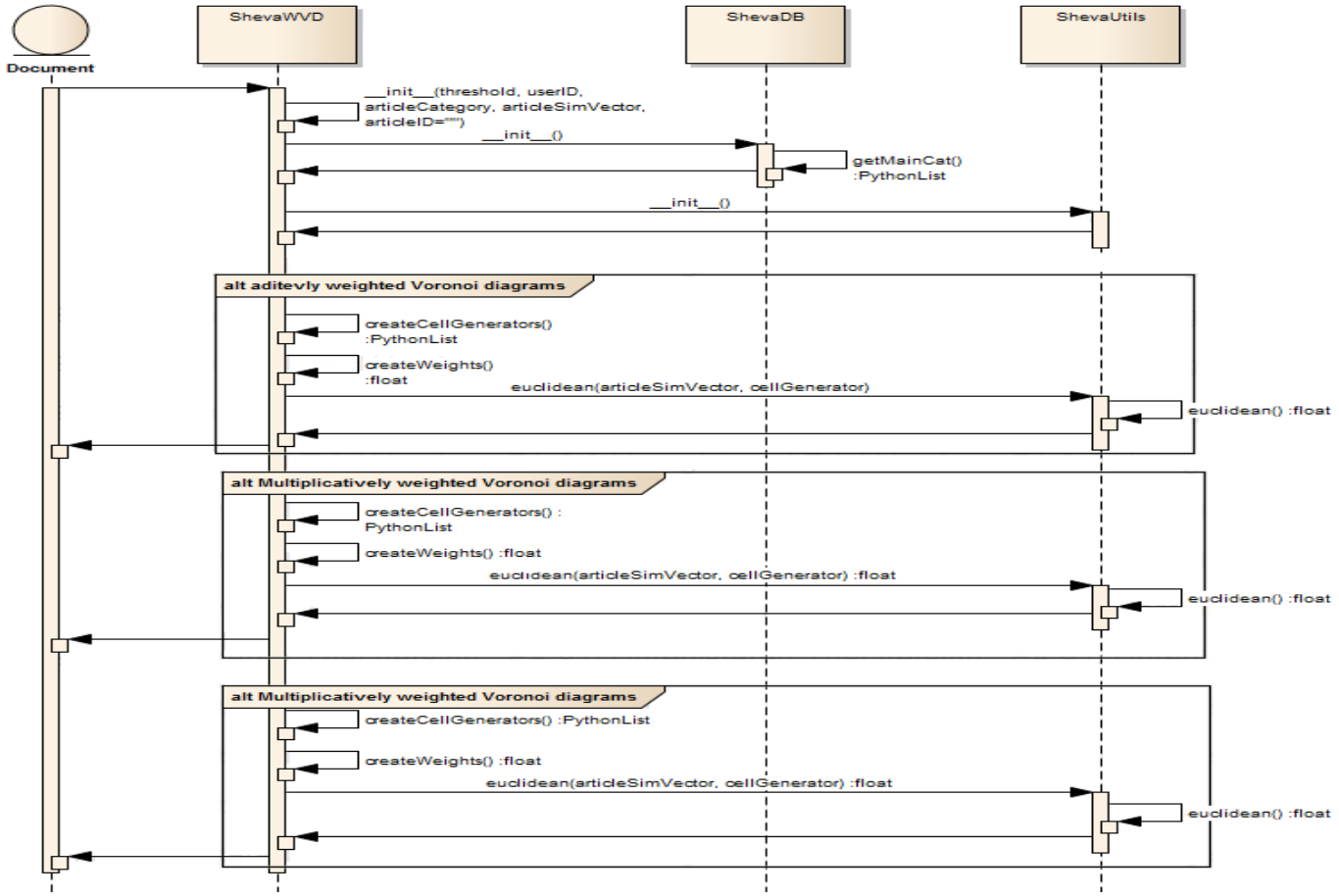

Figure 34: Weighted Voronoi diagrams sequence diagram

9.1 User pattern data preparation methodology

The system deals with four different types of data that define the input for the module presented in Figure 33: the previously mentioned *user-defined, session specific patterns* depicting average weights derived from all visited documents, *time defining pattern* depicting the relative amount of time spent on documents from single category, *category interest defining pattern* depicting the number of document from a specific category visited during browsing sessions and single document weighting values compared with the previously mentioned patterns. The overview of personalization/recommendation process is depicted in Figure 34.

Definition. Let $UP_i = \{w_{i1}, w_{i2}, \dots, w_{i15}\}$ be the user pattern, based on the previously presented categorization module, where w_{ij} presents the average weights for category *j* for user *i*.

Definition. Let $CG_i = (cg_{i1}, \dots, cg_{i15})$. We call these points Voronoi cell generators for user *i*. The detailed steps are presented in Figure 35.

Definition. Let $d_k = (nvp_i, cg_j);\ nvp \in NVP_k, cg_j \in CG_k, 1 \leq j \leq 15$ represent the Euclidean distance between a non-personalized document *i* and cell generator point *j* for user *k*. Then $\min(d_k)$ represents the recommended category for document nvp_i based on preferences for user *k*.

9.1.1 Extracting personalization factors

Definition. Let $D = (d, c);\ d \in NVP$ represent the document data and the document category assigned during the classification. Let $T_i = (t_{i1}, t_{i2}, \dots, t_{i15})$ represent user based time usage pattern. Let $C_i = (c_{i1}, c_{i2}, \dots, c_{i15})$ represent user *i* category interest pattern.

Then, user *i* timing factor is defined as t_{ij}, where j = d and category interest factor is defined as c_{ij} where *j* = d. Values t_{ij} and c_{ij} are called user *i* personalization factors.

Time and category interest data are used as the modification criteria for weighted Voronoi diagrams.

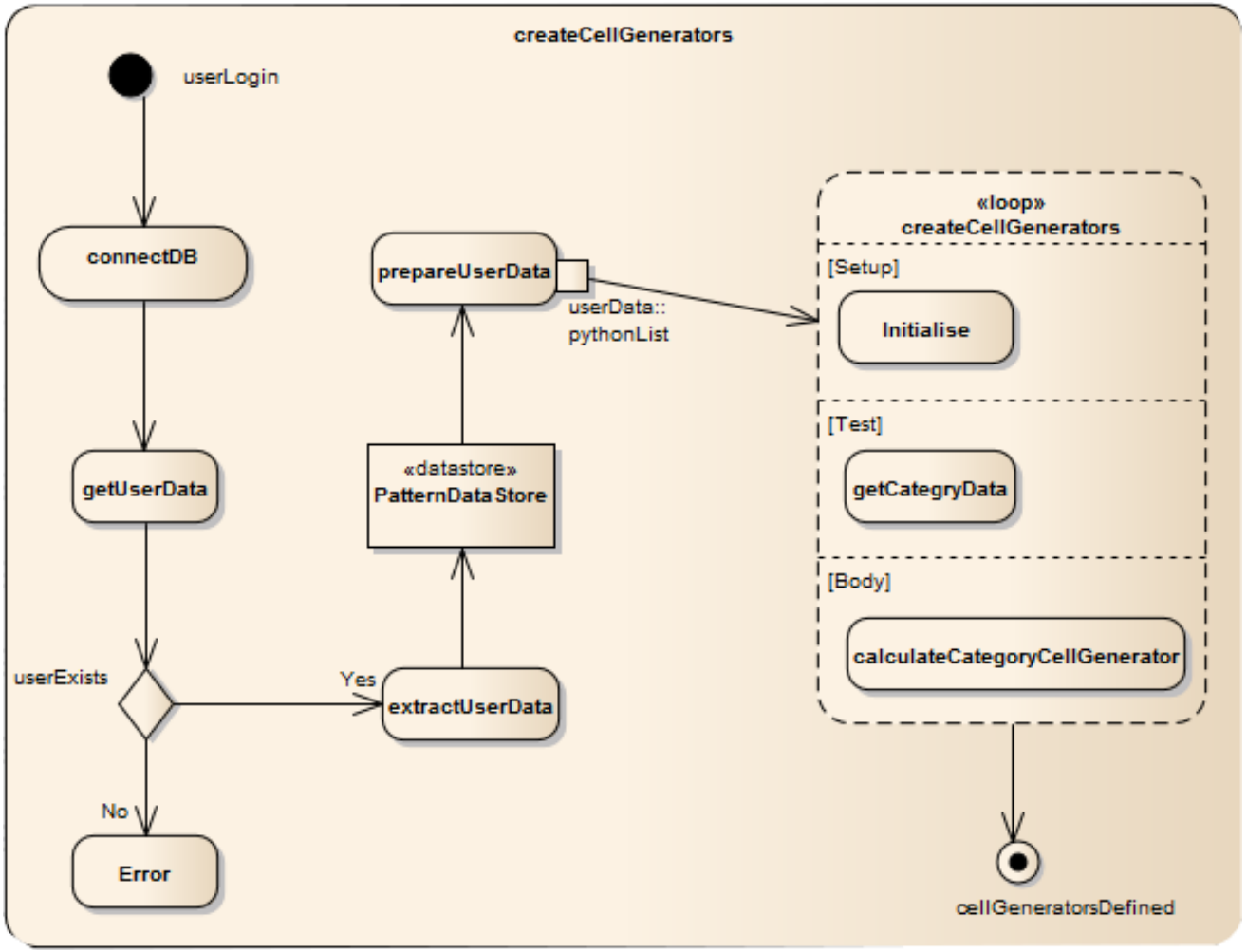

Figure 35: WVD cell generation activity diagram

They are generated from two different data sources:

- *Original document classification*, added to an individual document during the classification stage where each document is described by the newly assigned category and
- *Accumulated user-specific browsing data*, defined with two different patterns: time usage pattern and category interest pattern where each of the parameters defines specific interest information

9.1.1.1 Difference between personalization factors

The generated recommendations depend on the version of the weighted Voronoi diagrams and its inputs. As the weighting factors both time spent on a specific document assigned category as well as the category interest are used.

All defined weighted Voronoi diagrams are calculated in two manners:

a. based on category time usage pattern data
b. based on category interest pattern data

This offers two values per weighted Voronoi diagram. To test whether a different weighting parameter affects the results of different weighted Voronoi diagrams, a simple test is devised. The objective of this test is to compare ranked documents for a dummy user, with his patterns set to 1. The results are shown in Figure 36.

The generated sequence of recommended articles, based on three implemented weighted Voronoi diagrams, has been evaluated using standard IR techniques as well as the already used Euclidean distance measure. As the purpose of this test was to see if there are significant differences between using different weighting schemes for the same weighted Voronoi diagram implementation, the optimal results would show that the generated sequences, ordered by calculated similarity, have the same documents in the same order. The results shown in image below show exactly that. The IR measures are returned with a value as 1 and the Euclidean distance of the returned proposed articles is 0 (meaning no difference). These results are somewhat expected as they depend on the personalization patterns that are calculated based on the available browsing data; both interest as well as time patterns are similar if generated on the same browsing data. As weighted Voronoi diagrams use two

personalization factors, it is suggested from these results that the order in which the above mentioned pattern data is used in the calculation does not affect the final outcome, e.g. recommendations.

9.2 Analysis methodology

Three different weighted Voronoi diagrams have been implemented and analyzed in this work. The theoretical and implementation details have been discussed in chapters 5 and 8 respectively.

Their implementations differ in terms of implementing the above mentioned and defined personalization factors as well as the cell generator scheme utilized in the recommendation process. The results of their implementations are tested with standard IR test, precision, recall and F1 scores, as suggested by the reviewed literature ([2], [71] respectively). These measures were introduced in detail in chapter 4.2.

As far as the recommendation process is concerned, two implementations have been devised; differing by the way Voronoi cell generators are utilized. In the first implementation, user weighting profile is assumed to be the single cell generator that describes the area of user interests. Each document is then compared by this single generator and values are rearranged accordingly, from minimum distance to maximum distance. This approach gives a list of documents that have similarities to calculated user preferences.

The second approach implements a cell generator process: user browsing space is broken down into 15 cells depicting 15 major categorization areas, each cell being represented by a 15 dimensional point. The full definition is available in chapter 9.1. An article is then compared to all of the generated cells and put into one of the appropriate categories. The resulting recommendation list is then returned to the user with documents belonging to the most important category being returned as first, the second interest category as second and so on. Next, both recommendation schemes, implemented through the three defined weighted Voronoi diagrams will be tested and the results will be presented. This will conclude this chapter.

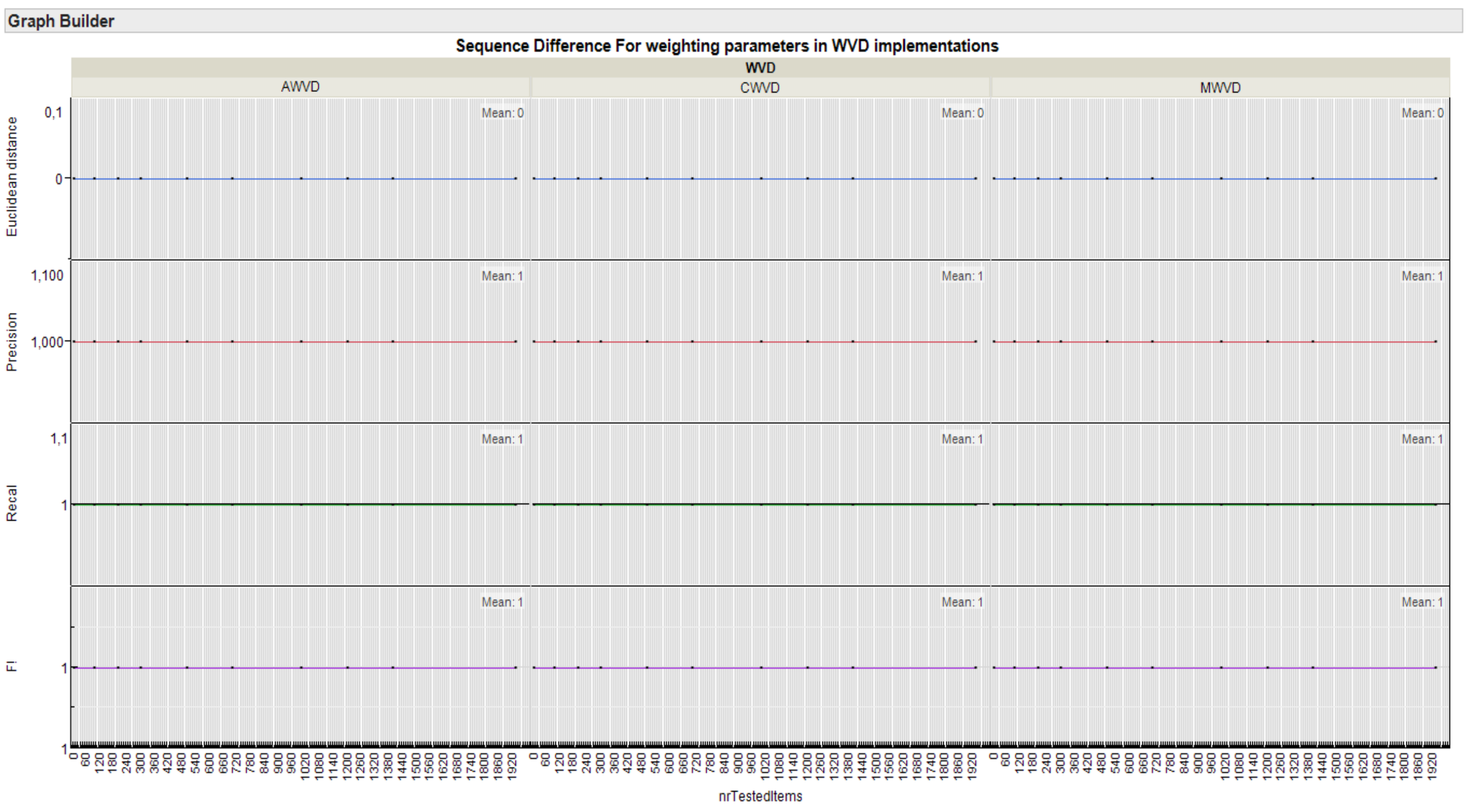

Figure 36: Generated recommendations for different weighting parameters

9.3 Result analysis

The available data is tested on two devised recommendation generation processes in two different manners. The purpose of these tests is to see how well the proposed recommendation approach, implemented through weighted Voronoi diagrams, operates on a set of collected articles.

Two different testing sets are prepared:

- Individual user visited articles, taken from his sessions
- Articles not visited by the said individual user but collected during browsing sessions (visited by other experiment participants).

Data sets are tested on two previously mentioned weighted Voronoi diagrams implementations, varying in the process of cell generation, and the results are compared.

Several testing measures are devised for these purposes:

- *Precision/recall/F1 measures*, for the documents from each user specific browsing sessions; the purpose is to see if the proposed recommendation/personalization approach would classify visited documents into two recommended or non-recommended sets of articles
- *Euclidean distance* between actual user profile (generated as presented in chapter 8) and virtual user profile generated in the same manner but based on documents not included in his recorded browsing sessions.

9.3.1 Single cell generated results

The goal of this approach is to test whether weighted Voronoi diagrams can be used as recommender algorithms, to begin with. The analysis is based on a set of visited documents and it tests if all visited documents will be recommended for each specific user. The visited documents are taken from the recorded browsing sessions. Mapping is done based on document ID values.

The results show that this approach generates recommendations perfectly aligned with the observed browsing sessions. Almost all documents visited by a specific user are classified as recommended to that user. There is a slight difference between the weights used as

personalization parameters: time based personalization shows slightly inferior results as shown in Figure 37.

This suggests that, in practical environments, category ratio based personalization scheme would perform better than time ratio personalization scheme. Time based personalization is also more difficult to utilize and will be the subject of future research in the development of this approach. This also suggests that further studies of weighted Voronoi diagrams as personalization/recommendation scheme is well advised as the results show that their outcome was very good.

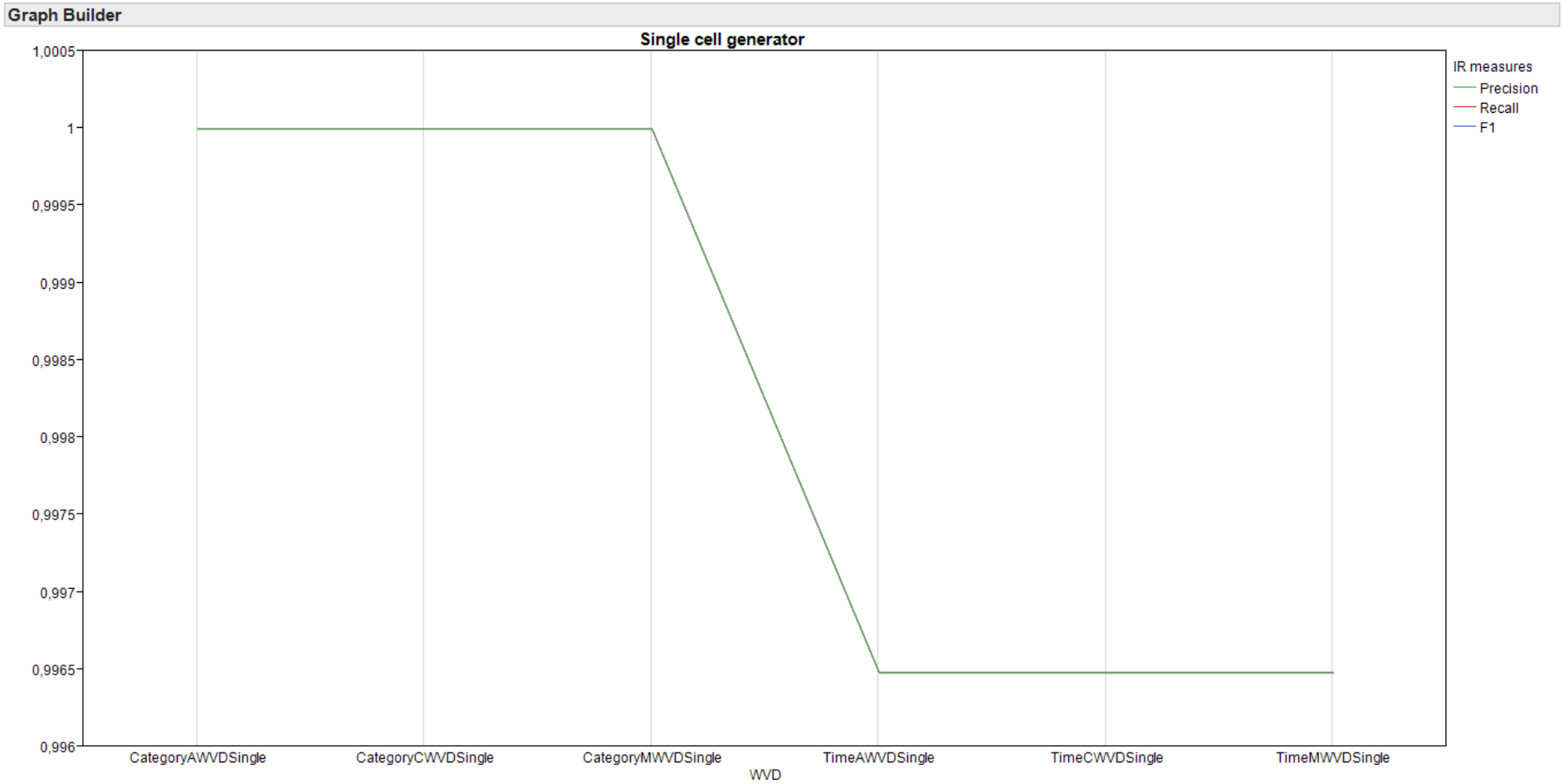

Figure 37: Single cell generator recommendations on visited documents

9.3.2 Multiple cell results

The second approach to generate a list of recommendations is by using multiple cell generators. The way of generating multiple cells is defined as CG_i in the introduction section of this chapter. This approach calculates the Euclidean distance between each cell generator and active document, arranges the document(s) into one of the available categories and returns a ranked list of documents retrieved as important to the user. Although the outcome of this process is different from the previously presented approach, namely single cell generator, the underlying mechanisms of calculating Euclidean distances between a document and the user profile are the same for all three types of weighted Voronoi diagrams analyzed in this section.

The results of this approach, presented in Figure 38, are identical to the results of the previously presented single cell generator approach (due to the fact that the underlying calculation mechanisms are identical). Once again it is shown that time patterns are less suitable for recommendation generation than category interests pattern, although by an ignorable margin. The real power of this approach will present itself in the future research, dealing with ranked personalization/recommendation.

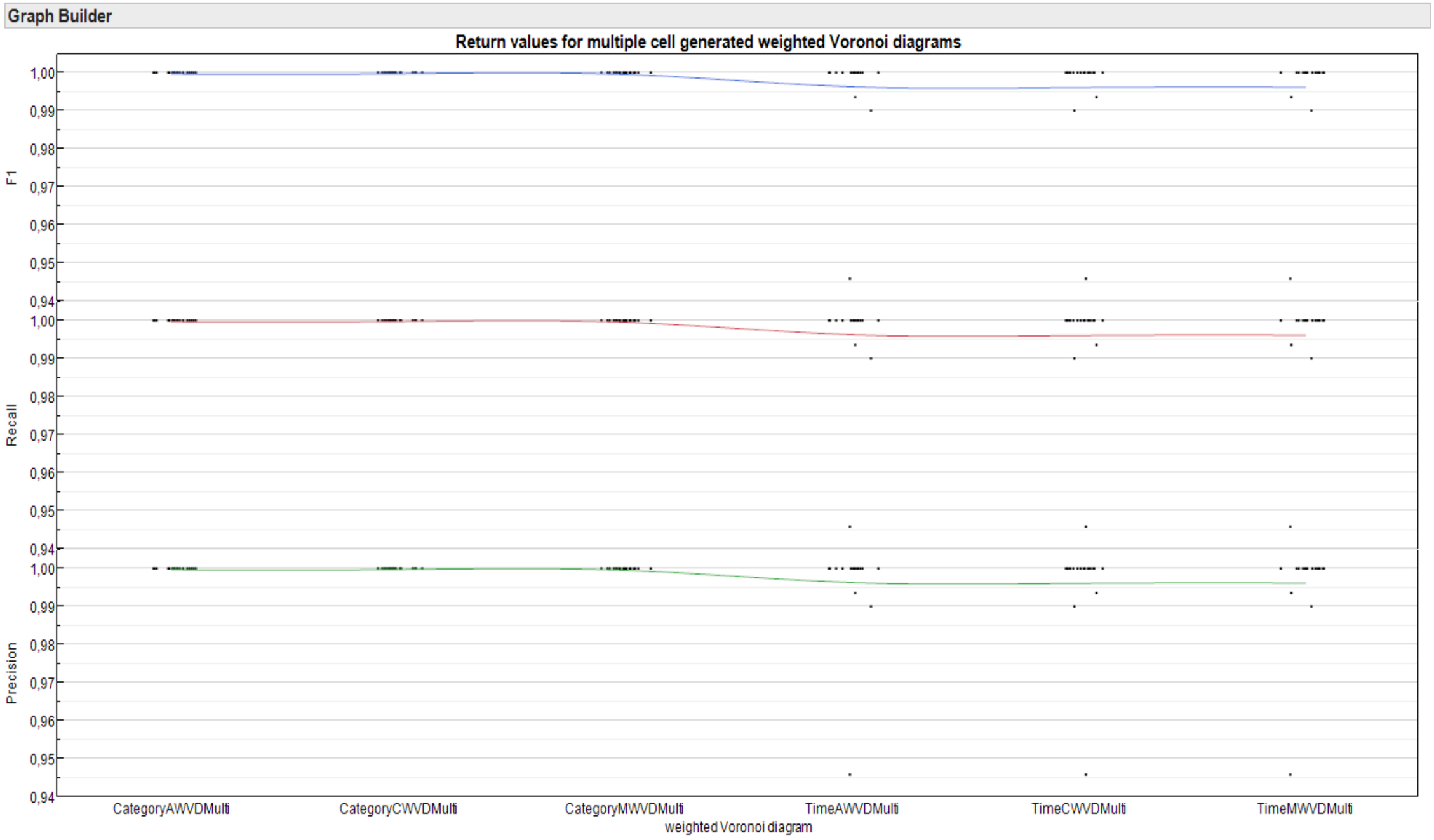

Figure 38: Multiple cell generator recommendations on visited documents

10. Conclusion and future work

The main goal of this dissertation is to research and present a novel approach for accessing and personalizing/recommending online accessible textual content available through known web news portals. There were several approaches possible for this research area, and the content-based personalization was chosen. For this purpose, several research areas have been combined in order to successfully prove the set hypothesis.

This dissertation had several goals and hypotheses as defined in subsection 2.7. The main focus of this work is the domain of web news portals, with an overall goal to research and present a new and/or improved personalization/recommendation framework. The contribution of this dissertation is as follows:

- Creating and presenting a universal taxonomy/classification scheme, based on the data of ODP
- Defining a way of extracting information about user preferences from user browsing history data
- Implementing the use of weighted Voronoi diagrams as the proposed personalization scheme

The main part of the work done during this research was focused on creating and evaluating different ODP based models for the purposes of creating the proposed unified taxonomy and testing different classification schemes. The need for a unified taxonomy/classification of information nodes (e.g. news articles) is derived from different classification schemes each web news portal employs. The foundation for the creation of content-based models, and the method employed in that phase, is given in chapters 3 and 4. The main problem in creating the classification models was the unstructured data presented in ODP and developing different grouping schemes, both vertical as well as horizontal, for the basis of this classification scheme. Not all results have been presented in this dissertation as the number of the devised testing schemes surpasses the scope of this dissertation. It has to be said that vertical models (defining a model as all pages in main category *Y* from root level (depth 2) and level n) have shown to have subpar results in comparison to vertical models.

Between all vertical models, percentage based as well as limit based models, the limit based models have over performed in comparison to the percentage based models. Other advantages

of using the limit based modes are performance measures; due to smaller sizes and smaller number of documents used in model creation, computer resources needed, as well as time needed for the comparison and the process of classification.

Additionally, due to the nature of created models, additional tests were made with different lower bound similarity levels, ranging from 0 (meaning, all documents that have any similarity greater than 0 are regarded as similar documents) to 0.3 (documents with similarity values of 0.3 and above are regarded as similar) in steps of 0.05. These tests had the following goals:

- Testing whether different similarity threshold have any impact on IR measures
- Testing whether different similarity measure impact calculated differences between users and their sessions

The results are shown in Figure 31 and Figure 32 respectively. The data shows that, while looking at the difference between single users browsing sessions, setting a higher threshold contributes to more noise to the data in that it lowers the differences for two personalization patterns (time and percentage weighting patterns; values ranging from 0.41 – 0.39 and 0.28 – 0.27 respectively) and increases the calculated difference for user weighting pattern (values range 0.26 – 0.54). While comparing the differences between distinct users who participated in this experiment, the same approach provides with similar results with time and percentage patterns, decreasing with bigger similarity threshold values and simultaneously increasing weighting patterns values. As the focus of this work is individual personalization, based on the test and results presented above, the author concluded that the best similarity threshold is the smallest one as the results for that value provide the smallest difference for sessions of individual user(s) and helps in the creation of better personalization patterns which serve as input to weighted Voronoi diagrams.

When it comes to user interest patterns, three patterns are identified and proposed for further use:

- User time patterns
- User category interest pattern
- User weighting pattern

They serve as the input for the weighted Voronoi diagrams that are proposed as the personalization algorithm with user weighting pattern depicting the weighted values through all the articles the user read during his sessions (grouped in 15 identified and proposed categories). Additionally, user behavior is described with the help of time spent on news articles classified as most similar to single available categories (user time pattern) and the interest in a specific category (user category interest pattern). Apart from the already mentioned results in determining the differences between different users and user sessions themselves, these patterns were also tested to see if, when it comes to their use in weighted Voronoi diagrams, there are any differences in the order of their input. The results showed that the differences are practically nonexistent meaning that the order of their use is the same (there are no big differences between those two personalization patterns). The way to extract those patterns from the available user browsing history data is shown in section 8.2 which satisfies and expands on the set hypothesis regarding user behavior patterns.

Finally, the work done on weighted Voronoi diagrams, as the third part of this research, introduces a new personalization approach for the purposes of individual personalization and presents a step away from the traditional approaches (e.g. user clustering and clustered based recommendations). Weighted Voronoi diagrams, as used in this dissertation, rely on preclassified items as well as additional personalization factors based on which they can calculate recommendations/personalization for each individual user. The main goal of this hypothesis was to show how weighted Voronoi diagrams can be used in achieving this goal, which was done in chapter 8 of this dissertation. Several tests were performed additionally to see how well the developed algorithms perform here, based on the previously visited document for each individual user. In these cases, each document was tested to see whether or not it would be recommended for the individual user for all three different weighted Voronoi diagrams implemented. Voronoi diagrams rely on cell generators to determine where a new point would belong to in the proposed classification and subsequently in the individual user interest set. For this purpose two cell generator ways were implemented:

- Single cell generators, where a new document is compared to the generated user browsing profile
- Multiple cell generators, where user browsing profile was divided into 15 different points, each depicting a user interest for single category

The results show that multiple based cell generators perform slightly better than single cell generators and that category interest pattern based recommendations perform almost perfectly.

10.1 Future work

During the work done in this dissertation, and especially the practical implementation of it (which resulted in freely available framework ShevaVIRT[41]), several future research directions presented themselves.

First, further improvement of the text cleaning process and venture into the NLP side of research, which is the one the author finds very interesting. Especially, focus on using n-gram notation and extracting concepts (not just words) from both ODP as well as Web pages to improve the quality of both classification models as well as document classification alone.

Additionally, research direction that this work can be pushed into next is to test the same data with different VSM techniques (some of them listed in section 3.3; most of them are already implemented in the above mentioned framework, but due to time constraints the tests themselves were left for later stages of this work) as tf-idf can be improved upon. Latent Dirichlet allocation, as the predominant classification modeling approach, is the one that is of interest as it allows for further dimensionality reduction, although it is still unclear if the results will be improved on the current results due to the ODP structure and input.

One path not traveled during this research was to cluster ODP similar nodes together in different manners besides the ones implemented (CATID and FATHERID connections in a hierarchical approach); the availability of symbolic links, which are available through dmoz data dump, would make a good addition to the already produced models and approaches. This approach has already been taken in IR community, but not to a great extent (as well as the overall use of ODP).

Last, the practical implementation of this research is missing; which will be the first step following the acceptance and (hopefully) dissertation defense.

[41] https://github.com/deakkon/SemanticVIRT

11. Literature

[1] J. Stewart, “BBC News - Global data storage calculated at 295 exabytes.” .

[2] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender systems: an introduction*. Cambridge University Press, 2011, p. 335.

[3] D. Vallet and I. Cantador, “Personalizing web search with folksonomy-based user and document profiles,” *Adv. Inf. Retr.*, pp. 420–431, 2010.

[4] S. Sendhilkumar and T. Geetha, “Personalized ontology for web search personalization,” in *Proceedings of the 1st Bangalore Annual Compute Conference*, 2008, pp. 1–7.

[5] P. A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschütter, “Using ODP metadata to personalize search,” in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’05*, 2005, pp. 178–185.

[6] P. Morville and J. Callender, *Search Patterns*, First edit., vol. 12, no. 3. Sebastopol, CA, USA: O’Reilly Media, 2010, p. 193.

[7] O. Hoeber, “Web Information Retrieval Support Systems: The Future of Web Search,” *2008 IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, pp. 29–32, Dec. 2008.

[8] A. Micarelli, F. Gasparetti, F. Sciarrone, and S. Gauch, “Personalized Search on the World Wide Web,” pp. 195–230.

[9] H. Chen, A. L. Houston, R. R. Sewell, and B. R. Schatz, “Internet browsing and searching: User evaluations of category map and concept space techniques,” *J. Am. Soc. Inf. Sci.*, vol. 49, no. 7, pp. 582–603, 1998.

[10] S. S. Ormstad and J. Isojärvi, “Information retrieval for health technology assessment: standardization of search methods.,” *Int. J. Technol. Assess. Health Care*, vol. 26, no. 4, pp. 359–61, Oct. 2010.

[11] A. Sieg, B. Mobasher, and R. Burke, “Web search personalization with ontological user profiles,” *Proc. Sixt. ACM Conf. Conf. Inf. Knowl. Manag. - CIKM ’07*, p. 525, 2007.

[12] N. Matthijs, “Personalizing web search using long term browsing history,” in *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011, pp. 25–34.

[13] K. Goenka, I. B. Arpinar, and M. Nural, “Mobile web search personalization using ontological user profile,” *Proc. 48th Annu. Southeast Reg. Conf. - ACM SE ’10*, p. 1, 2010.

[14] P. N. Bennett, F. Radlinski, R. W. White, and E. Yilmaz, “Inferring and using location metadata to personalize web search,” *Proc. 34th Int. ACM SIGIR Conf. Res. Dev. Inf. - SIGIR ’11*, p. 135, 2011.

[15] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli, “User profiles for personalized information access,” in *The adaptive web*, 2007, pp. 54–89.

[16] A. Kobsa, “Generic user modeling systems,” *User Model. User-adapt. Interact.*, vol. 11, no. 1–2, pp. 49–63, 2001.

[17] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, “Discovery and evaluation of aggregate usage profiles for web personalization,” *Data Min. Knowl. Discov.*, vol. 6, no. 1, pp. 61–82, 2002.

[18] P. Brusilovsky and E. Millán, “User models for adaptive hypermedia and adaptive educational systems,” in *The adaptive web*, 2007, pp. 3–53.

[19] M. Eirinaki and M. Vazirgiannis, “Web mining for web personalization,” *ACM Trans. Internet Technol.*, vol. 3, no. 1, pp. 1–27, Feb. 2003.

[20] O. Nasraoui, “World Wide Web Personalization,” in *Encyclopedia of Data Mining and Data Warehousing*, John Wang, Ed. Idea Group, 2005, pp. 1235–1241.

[21] Z. Jrad, M.-A. Aufaure, and M. Hadjouni, “A Contextual user model for Web personalization,” in *Web Information Systems Engineering–WISE 2007 Workshops*, 2007, pp. 350–361.

[22] J.-J. Lee, J.-H. Lee, J. Ha, and S. Lee, “Novel web page classification techniques in contextual advertising,” *Proceeding Elev. Int. Work. Web Inf. data Manag. - WIDM ’09*, p. 39, 2009.

[23] B. Mobasher, “Data mining for web personalization,” *Adapt. Web*, pp. 90–135, 2007.

[24] B. Liu, *Web data mining: exploring hyperlinks, contents, and usage data*, 2nd Editio. Springer Verlag, 2011, p. 642.

[25] E. Frias-Martinez, S. Y. Chen, and X. Liu, “Survey of Data Mining Approaches to User Modeling for Adaptive Hypermedia,” *IEEE Trans. Syst. Man Cybern. Part C (Applications Rev.*, vol. 36, no. 6, pp. 734–749, Nov. 2006.

[26] T. Lavie, M. Sela, I. Oppenheim, O. Inbar, and J. Meyer, “User attitudes towards news content personalization,” *Int. J. Hum. Comput. Stud.*, vol. 68, no. 8, pp. 483–495, Aug. 2010.

[27] P. Heymann, A. Paepcke, and H. Garcia-Molina, “Tagging human knowledge,” in *Proceedings of the third ACM international conference on Web search and data mining - WSDM ’10*, 2010, pp. 51–60.

[28] G.-R. Xue, D. Xing, Q. Yang, and Y. Yu, “Deep classification in large-scale text hierarchies,” *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR ’08*, p. 619, 2008.

[29] C. Christophi, D. Zeinalipour-Yazti, M. D. Dikaiakos, and G. Paliouras, “Automatically Annotating the ODP Web Taxonomy,” in *Proc. 11th Panhellenic Conf. Informatics (PCI’07)*, 2007, pp. 397–408.

[30] R. Cooley, B. Mobasher, and J. Srivastava, “Data preparation for mining world wide web browsing patterns,” *Knowl. Inf. ...*, vol. 1, no. 1, pp. 5–32, 1999.

[31] R. Reitsma, S. Trubin, and E. Mortensen, “Weight-proportional Space Partitioning Using Adaptive Voronoi Diagrams,” *Geoinformatica*, vol. 11, no. 3, pp. 383–405, Mar. 2007.

[32] S. I. Trubin, “Information Space Mapping with Adaptive Multiplicatively Weighted Voronoi Diagrams,” Oregon State University, 2006.

[33] H.-R. Kim and P. K. Chan, “Learning implicit user interest hierarchy for context in personalization,” *Appl. Intell.*, vol. 28, no. 2, pp. 153–166, Jun. 2007.

[34] F. Carmagnola, F. Cena, and C. Gena, “User modeling in the social web,” in *KnowledgeBased Intelligent Information and Engineering Systems*, vol. 4694, Springer, 2007, pp. 745–752.

[35] C. Wei, W. Sen, Z. Yuan, and C. Lian-Chang, “Algorithm of mining sequential patterns for web personalization services,” *ACM SIGMIS Database*, vol. 40, no. 2, p. 57, Apr. 2009.

[36] R. Feldman, Y. Kinarl, Y. Lindell, O. Liphstat, M. Rajman, Y. Schler, and O. Zamir, “Text Mining at the Term Level,” in *Principles of Data Mining and Knowledge Discovery*, J. M. Żytkow and M. Quafafou, Eds. Springer Berlin Heidelberg, 1998, pp. 65–73.

[37] M. Levene and J. Borges, “Data Mining of User Navigation Patterns,” in *Web Usage Analysis and User Profiling*, B. Masand and Myra Spiliopoulou, Eds. Springer Berlin Heidelberg, 2000, pp. 92–112.

[38] B. Mobasher, R. Cooley, and J. Srivastava, “Automatic personalization based on Web usage mining,” *Commun. ACM*, vol. 43, no. 8, pp. 142–151, 2000.

[39] J. Srivastava, R. Cooley, M. Deshpande, and P. Tan, “Web Usage Mining : Discovery and Applications of Usage Patterns from Web Data,” vol. 1, no. 2, pp. 12–23, 2000.

[40] K. Golub and M. Lykke, “Automated classification of web pages in hierarchical browsing,” *J. Doc.*, vol. 65, no. 6, pp. 901–925, 2009.

[41] R. Gupta and L. Ratinov, “Text Categorization with Knowledge Transfer from Heterogeneous Data Sources.,” in *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, 2008, pp. 842–847.

[42] I. Antonellis, C. Bouras, and V. Poulopoulos, “Personalized news categorization through scalable text classification,” in *Proceedings of the 8th Asia-Pacific Web conference on Frontiers of WWW Research and Development - APWeb’06*, 2006, pp. 391–401.

[43] P. Brzeminski and W. Pedrycz, “Textual-based clustering of web documents,” *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 12, no. 6, pp. 715–743, 2004.

[44] E. Baykan, M. Henzinger, L. Marian, and I. Weber, “Purely URL-based topic classification,” in *Proceedings of the 18th international conference on World wide web - WWW ’09*, 2009, pp. 1109–1110.

[45] C. Bayrak and H. Joshi, “Learning Contextual Behavior of Text Data,” *Fourth Int. Conf. Mach. Learn. Appl.*, pp. 299–304, 2005.

[46] S. Brin and L. Page, “The anatomy of a large-scale hypertextual Web search engine,” *Comput. Networks ISDN Syst.*, vol. 30, no. 1–7, pp. 107–117, Apr. 1998.

[47] A. Plangprasopchok and K. Lerman, “Constructing folksonomies from user-specified relations on flickr,” in *Proceedings of the 18th international conference on World wide web - WWW ’09*, 2009, pp. 781–790.

[48] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, “Learning to classify short and sparse text & web with hidden topics from large-scale data collections,” *Proceeding 17th Int. Conf. World Wide Web - WWW ’08*, pp. 91–100, 2008.

[49] P. Singh, T. Lin, E. Mueller, and G. Lim, “Open Mind Common Sense: Knowledge acquisition from the general public,” in *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, Springer Berlin Heidelberg, 2002, pp. 1223–1237.

[50] D. Davidov, E. Gabrilovich, and S. Markovitch, “Parameterized generation of labeled datasets for text categorization based on a hierarchical directory,” in *Proceedings of the*

27th annual international conference on Research and development in information retrieval - SIGIR '04, 2004, p. 250.

[51] W. Contributors, "Information retrieval," *Wikipedia, The Free Encyclopedia.*, 2011. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Information_retrieval&oldid=582521337. [Accessed: 15-Dec-2013].

[52] D. Kuropka, *Modelle zur Repräsentation natürlichsprachlicher Dokumente: Information-Filtering und -Retrieval mit relationalen Datenbanken*, vol. 10. Logos Verlag, 2004.

[53] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, vol. 22, no. 1. McGraw-Hill, 1986, pp. xv, 448.

[54] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975.

[55] G. Salton, C. S. Yang, and C. T. Yu, "A Theory of Term Importance in Automatic Text Analysis," *J. Am. Soc. Inf. Sci.*, vol. 36, no. 1, pp. 33–44, 1975.

[56] G. Salton, *A theory of indexing*. Society for industrial and applied mathematics, 1975.

[57] I. Moullinier and P. Jackson, *Natural language processing for online applications: Text retrieval, extraction and categorization*. John Benjamins Publishing Co., 2002, p. 237.

[58] J. Cowie and W. Lehnert, "Information extraction," *Commun. ACM*, vol. 39, no. 1, pp. 80–91, 1996.

[59] P. D. Turney and P. Pantel, "From Frequency to Meaning : Vector Space Models of Semantics," vol. 37, pp. 141–188, 2010.

[60] J. Hobbs, D. A. Sr, J. Bear, D. I. Sr, and W. Tyson, "Fastus: A system for extracting information from natural-language text," SRI International, 1992.

[61] N. Chomsky, *Syntactic structures*, 2nd editio. Walter de Gruyter, 2002, p. 117.

[62] W. Contributors, “Context-free grammar,” *Wikipedia, The Free Encyclopedia.*, 2010. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Context-free_grammar&oldid=583916378. [Accessed: 15-Nov-2013].

[63] K. Baker, A. Franz, and P. Jordan, “Coping with ambiguity in knowledge-based natural language analysis,” *Proc. FLAIRS-94*, 1994.

[64] K. L. Baker, A. M. Franz, P. W. Jordan, T. Mitamura, and E. H. Nyberg, “Coping with ambiguity in a large-scale machine translation system,” in *Proceedings of the 15th conference on Computational linguistics - COLING ’94*, 1994, vol. 1, pp. 90–94.

[65] M. F. Porter, “An algorithm for suffix stripping,” *Progr. Electron. Libr. Inf. Syst.*, vol. 40, no. 3, pp. 211–218, 2006.

[66] W. Contributors, “Stemming,” *Wikipedia, The Free Encyclopedia.* [Online]. Available: http://en.wikipedia.org/w/index.php?title=Stemming&oldid=584206974. [Accessed: 13-Dec-2013].

[67] W. Contributors, “Natural language processing,” *Wikipedia, The Free Encyclopedia.* [Online]. Available: http://en.wikipedia.org/w/index.php?title=Natural_language_processing&oldid=582178777. [Accessed: 16-Dec-2013].

[68] D. Blei, A. Ng, and M. Jordan, “Latent dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[69] J. Lafferty and D. Blei, “Correlated Topic Models,” in *Advances in Neural Information Processing Systems 18*, 2005.

[70] T. Roelleke, *Information Retrieval Models: Foundations and Relationships*, vol. 5, no. 3. 2013, pp. 1–163.

[71] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*. Boston, MA: Springer US, 2011.

[72] C. Fellbaum, *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press, 1998, p. 423.

[73] E. Sanchez, *Fuzzy logic and the semantic web*. Elsevier Science, 2006, p. 496.

[74] M. Baziz, M. Boughanem, Y. Loiseau, and H. Prade, "Fuzzy logic and ontology-based information retrieval," in *Fuzzy Logic: A Spectrum of Theoretical & Practical Issues*, P. P. Wang, D. Ruan, and E. E. Kerre, Eds. Springer Berlin Heidelberg, 2007, pp. 193–218.

[75] W. Contributors, "Standard Boolean model," *Wikipedia, The Free Encyclopedia*. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Standard_Boolean_model&oldid=585564970. [Accessed: 16-Dec-2013].

[76] W. Contributors, "Extended Boolean model," *Wikipedia, The Free Encyclopedia*. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Extended_Boolean_model&oldid=560948403. [Accessed: 18-Jun-2013].

[77] U. Straccia, "A fuzzy description logic for the semantic web," *Fuzzy Log. Semant. Web*, 2006.

[78] S. Zadrozny and K. Nowacka, "Fuzzy information retrieval model revisited," *Fuzzy Sets Syst.*, 2009.

[79] S. Dominich, *The Modern Algebra of Information Retrieval*, 1st editio. Springer, 2008, p. 327.

[80] W. Contributors, "Fuzzy set operations," *Wikipedia, The Free Encyclopedia*. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Fuzzy_set_operations&oldid=544313182. [Accessed: 08-Mar-2013].

[81] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*, 1st editio., no. c. Cambridge: Cambridge University Press, 2008, p. 496.

[82] C. J. van Rijsbergen, *Information retrieval*, vol. 26, no. 4. London: Butterworths, 1979.

[83] D. D. Lewis, “An evaluation of phrasal and clustered representations on a text categorization task,” *Annu. ACM Conf. Res. Dev. Inf. Retr.*, 1992.

[84] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical machine learning tools and techniques*, Third edit. Elsevier, 2011, p. 665.

[85] W. Cohen, “Fast effective rule induction,” *ICML*, pp. 115–123, 1995.

[86] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern infromation retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999, p. 544.

[87] A. Okabe, B. Boots, K. Sugihara, and S. N. S. CHIU, *Spatial tessellations: concepts and applications of Voronoi diagrams*, 2nd editio. John Wiley & Sons Inc, 2000, p. 683.

[88] S. Devadoss and J. O’Rourke, *Discrete and computational geometry*. Princeton, New Jersey: Princeton University Press, 2011, p. 270.

[89] F. Aurenhammer, “Voronoi Diagrams,” pp. 201–290, 2000.

[90] S. S. Skiena, *The Algorithm Design Manual*, 2nd ed. London: Springer London, 2008.

[91] S. Perugini, “Symbolic links in the Open Directory Project,” *Inf. Process. Manag.*, vol. 44, no. 2, pp. 910–930, Mar. 2008.

[92] R. Reitsma and S. Trubin, “Information space partitioning using adaptive Voronoi diagrams,” *Inf. Vis.*, vol. 6, no. 2, pp. 123–138, May 2007.

[93] R. Reitsma, S. Trubin, and S. Sethia, “Adaptive Multiplicatively Weighted Voronoi Diagrams for Information Space Regionalization,” in *Proceedings of the Information Visualisation, Eighth International Conference on (IV 2004)*, 2004, pp. 290–294.

[94] B. Singh and H. K. Singh, “Web Data Mining research: A survey,” *2010 IEEE Int. Conf. Comput. Intell. Comput. Res.*, pp. 1–10, Dec. 2010.

[95] W.-C. Hu, H.-J. Yang, C. Lee, and J. Yeh, “World Wide Web Usage Mining Systems and Technologies,” *Syst. Cybern. INFORMATICS*, vol. 1, no. 4, pp. 53–59, 2003.

[96] H. L. Borges and A. C. Lorena, “A Survey on Recommender Systems for News Data,” in *Smart Information and Knowledge Management: Advances, Challenges, and Critical Issues*, Edward Szczerbicki and Ngoc Thanh Nguyen, Eds. Springer Berlin Heidelberg, 2010, pp. 129–151.

[97] R. Cooley, P. Tan, and J. Srivastava, “WebSIFT: The web site information filter system,” in *Proceedings of the Web Usage Analysis and User Profiling Workshop*, 1999, vol. 8.

[98] M. Eirinaki, M. Vazirgiannis, and I. Varlamis, “SEWeP: using site semantics and a taxonomy to enhance the Web personalization process,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’03*, 2003, pp. 99–108.

[99] S. Paulakis, C. Lampos, M. Eirinaki, and M. Vazirgiannis, “Sewep: a web mining system supporting semantic personalization,” in *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases - PKDD ’04*, 2004, pp. 552–554.

[100] M. Eirinaki, C. Lampos, S. Paulakis, and M. Vazirgiannis, “Web personalization integrating content semantics and navigational patterns,” *Proc. 6th Annu. ACM Int. Work. Web Inf. data Manag. - WIDM ’04*, p. 72, 2004.

[101] K. Elleithy, Ed., *Advances and Innovations in Systems, Computing Sciences and Software Engineering*. Dordrecht: Springer Netherlands, 2007.

[102] F. Frasincar, J. Borsje, and F. Hogenboom, “Personalizing News Services Using Semantic Web Technologies,” in *E-Business Applications for Product Development and Competitive Growth: Emerging Technologies*, I. Lee, Ed. IGI Global, 2010, pp. 261–289.

[103] F. Goossen, W. IJntema, F. Frasincar, F. Hogenboom, and U. Kaymak, “News personalization using the CF-IDF semantic recommender,” in *Proceedings of the International Conference on Web Intelligence, Mining and Semantics - WIMS ’11*, 2011.

[104] W. IJntema, F. Goossen, F. Frasincar, and F. Hogenboom, “Ontology-based news recommendation,” in *Proceedings of the 1st International Workshop on Data Semantics - DataSem ’10*, 2010.

[105] K. Schouten, P. Ruijgrok, J. Borsje, F. Frasincar, L. Levering, and F. Hogenboom, “A semantic web-based approach for personalizing news,” in *Proceedings of the 2010 ACM Symposium on Applied Computing - SAC ’10*, 2010, pp. 854–861.

[106] A. Verheij, A. Kleijn, F. Frasincar, D. Vandic, and F. Hogenboom, “Querying and ranking news items in the hermes framework,” in *Proceedings of the 27th Annual ACM Symposium on Applied Computing - SAC ’12*, 2012, pp. 672–679.

[107] M. Capelle, F. Frasincar, M. Moerland, and F. Hogenboom, “Semantics-based news recommendation,” in *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics - WIMS ’12*, 2012.

[108] C. Bouras, V. Poulopoulos, and V. Tsogkas, “PeRSSonal’s core functionality evaluation: Enhancing text labeling through personalized summaries,” *Data Knowl. Eng.*, vol. 64, no. 1, pp. 330–345, Jan. 2008.

[109] C. Bouras, V. Poulopoulos, and V. Tsogkas, “Evaluating PeRSSonal : a medium for personalized dynamically created news,” in *Proceedings of the IADIS International Conference on WWW/Internet*, 2008, pp. 211–218.

[110] C. Bouras and V. Tsogkas, “Noun retrieval effect on text summarization and delivery of personalized news articles to the user’s desktop,” *Data Knowl. Eng.*, vol. 69, no. 7, pp. 664–677, Jul. 2010.

[111] G. Adam, C. Bouras, and V. Poulopoulos, “Efficient extraction of news articles based on RSS crawling,” *2010 Int. Conf. Mach. Web Intell.*, pp. 1–7, Oct. 2010.

[112] C. Bouras, G. Tsichritzis, and V. Tsogkas, “Caching news channels on the user’s desktop,” in *Proceedings of the IADIS International Conference Applied Computing*, 2009, pp. 35–42.

[113] C. Bouras and V. Poulopoulos, “Dynamic user context web personalization in meta-portals,” in *The IEEE symposium on Computers and Communications*, 2010, pp. 925–930.

[114] G. Adam, C. Bouras, and V. Poulopoulos, “Utilizing RSS Feeds for Crawling the Web,” *2009 Fourth Int. Conf. Internet Web Appl. Serv.*, pp. 211–216, 2009.

[115] C. Bouras, V. Poulopoulos, and P. Silintziris, “Personalized News Search in WWW: Adapting on User’s Behavior,” *2009 Fourth Int. Conf. Internet Web Appl. Serv.*, pp. 125–130, 2009.

[116] C. Bouras and V. Poulopoulos, “Enhancing meta-portals using dynamic user context personalization techniques,” *J. Netw. Comput. Appl.*, vol. 35, no. 5, pp. 1446–1453, Sep. 2012.

[117] P. Kalinov and A. Sattar, “Building a Dynamic Classifier for Large Text Data Collections,” vol. 104, no. January, 2010.

[118] C. Sherman, “Humans Do It Better: Inside the Open Directory Project.,” *Information Today, Inc.* [Online]. Available: http://www.infotoday.com/online/OL2000/sherman7.html. [Accessed: 13-Oct-2013].

[119] S. So, J.-H. Lee, D. Jung, J. Ha, and S. Lee, “Extending Open Directory Project to represent user interests,” *Proc. 27th Annu. ACM Symp. Appl. Comput. - SAC ’12*, p. 354, 2012.

[120] T. H. Haveliwala, “Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search,” *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 784–796, Jul. 2003.

[121] H.-S. Oh, Y. Choi, and S.-H. Myaeng, “Combining global and local information for enhanced deep classification,” *Proc. 2010 ACM Symp. Appl. Comput. - SAC ’10*, p. 1760, 2010.

[122] P. N. Bennett and N. Nguyen, “Refined experts,” in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’09*, 2009, pp. 11–18.

[123] M. Greenwood, “Implementing a vector space document retrieval system,” *dcs.shef.ac.uk*.

[124] F. clipart collection. WordNet 3.0, “cybernaut.” [Online]. Available: http://www.thefreedictionary.com/cybernaut. [Accessed: 09-Dec-2013].

[125] J. Dever, “Semantic Value,” *Elsevier Encyclopeda of Language and Linguistics*. Elsevier, pp. 137–142, 2006.