New Topological Approach to Vocabulary Mining and Document Classification

A. S. Salama and O. G. El-Barbary

Mathematics Department, Faculty of Science, Tanta University, Tanta, Egypt
Dr_salama75@yahoo.com, ualbarbari@su.edu.sa

Abstract: In This Paper We Investigate A New Framework For Vocabulary Mining And Document Classification That Derives From The Combination Of Rough Sets And Mixed Neighborhood System. The Framework Allows One To Use Mixed Neighborhood Approximations Even When The Documents And Queries Are Described Using Weighted. The Paper Also Explores The Real Life Applications Of Mixed Neighborhood System Approach. The Proposed Framework Supports The Systematic Study And Application Of Different Vocabulary Views In Information Retrieval.

[A. S. Salama and O. G. El-Barbary. **New Topological Approach TO Vocabulary Mining AND Document Classification.** *Life Sci J* 2014;11(5):84-91]. (ISSN:1097-8135). http://www.lifesciencesite.com. 11

Keywords: Vocabulary Mining; Generalized Rough Sets; Data Mining; Topological Spaces; Information Systems.

1. Introduction

In Recent Years, With The Astonishing Expansion Of The Internet, And The Increase In Hard Disk Capacities, Processing Power Of Computers And Bandwidth Of Network Connections, There Has Been Tremendous Growth In The Volume Of Electronic Text Documents Available On The Internet, Company-Wide Intranets, And Digital Libraries. In Information Retrieval The Challenge Is To Retrieve Relevant Texts In Response To User Queries. Information Retrieval Technology Has Matured To The Point That We Now Have Reasonably Sophisticated Operational And Research Systems. However, Increasing The Effectiveness Of Retrieval Algorithms Remains An Important And Actively Pursued Research Goal. Query Refinement, Where The Initial Query Is Modified To Yield A Potentially More Effective Query, Is An Important Part Of Information Retrieval. This Step Is Very Critical For Users Whose Queries Are Not Formulated Well Enough For An Effective Retrieval Run. One Alternative For Query Refinement, Referred To Here As Vocabulary-Based Query Refinement, Is To Exploit Knowledge Within A Vocabulary That Is Typically Domain Specific. A Second Approach Utilizes The Vocabulary In Documents Related To The Query Where The Related Documents May Be Identified Either Through Relevance Or Retrieval Feedback.

Several Families Of Statistical Information Retrieval Models Have Received Significant And Long-Term Attention, Such As The Boolean, Vector, Probabilistic And Fuzzy Families Of Models. The General Approach Is To Create Suitable Representations (Boolean, Weighted, Un Weighted, Etc.) For The Query And The Document And Apply A Suitable Retrieval Technique (Similarity Computation, Probability Of Relevance, Etc.) That Derives From The Adopted Model. Query Refinement In The Boolean Model May Occur By Either Changing The Query Operators Or Changing The Terms Or Both. At All Times The Integrity Of The Term-Operator Relationships With Respect To The User's Information Needs Must Be Maintained. In The Vector Model, Processes Such As Rocchio's And Ide's Feedback Offer Document-Based Query Refinement Options Researchers Have Also Investigated In Contrast, The Rough Set Model Offers A Tight Integration Between Retrieval And Vocabulary- Based Query Refinement. In Fact, Retrieval Operates Only After First Exploring Query Refinement. Characteristics Of The Domain Vocabulary, I.E., Terms And Relationships, Are Automatically Utilized To Refine The Query Representation Before Retrieval Begins. An Additional Advantage Is That The Model Also Automatically Allows The Natural Perturbations In Vocabularies To Influence Document Representations. In Essence, Rough Sets Offer An Approach Where The Domain's Vocabulary Can Be Automatically Mined Prior To Retrieval. Relationships Linking Terms Such As Synonymy, Near Synonymy Or Related Terms, Lexically-Related Terms, Specific And General Terms Can All Be Automatically Mined In Order To Strengthen Retrieval Effectiveness.

Our Research Goal Is To Explore The Application Of The Family Of Rough Set Models To Information Retrieval. Almost 10 Years Ago, Initial Efforts By One Of The Authors Demonstrated Some Of The Potential Of Rough Sets For Information Retrieval. Since Then The Area Of Rough Sets Has Matured Significantly With Many Exciting Advances Reported In The Literature. We Will Explore Further Developments And Their Potential For Information Retrieval. In Particular, We Aim To Determine If Current Extensions To The Model Will Strengthen Our Previous Applications Of Rough Sets To Retrieve.

There Are Many Reasons For The Study Of Granular Rough Set Theory [1]. The Practical Necessity And Simplicity In Problem Solving Are Perhaps Some Of The Main Reasons. When A Problem Involves Incomplete, Uncertain, Or Vague Information, It May Be Difficult To Differentiate Distinct Elements And One Is Forced To Consider Granules [2-9]. Although Detailed Information May Be Available, It May Be Sufficient To Use Granules In Order To Have An Efficient And Practical Solution. Very Precise Solutions May Not Be Required For Many Practical Problems. The Use Of Granules Generally Leads To Simplification Of Practical Problems. The Acquisition Of Precise Information May Be Too Costly, And Coarse-Grained Information Reduces Cost [10-14]. There Is Clearly A Need For The Systematic Studies Of Granular Rough Computing. It Is Expected That Granular Rough Computing Will Play An Important Role In The Design And Implementation Of Efficient And Practical Intelligent Information Systems. The Theories Of Rough Sets [15-20] And Neighborhood Systems Provide Convenient And Effective Tools For Granulation, And Deal With Some Fundamental Granulation Structures [21,22].

In The Rough Set Theory, One Starts With An Equivalence Relation. A Universe Is Divided Into A Family Of Disjoint Subsets. The Granulation Structure Adopted Is A Partition Of The Universe. By Weakening The Requirement Of Equivalence Relations, We Can Have More General Granulation Structures Such As Coverings Of The Universe. Neighborhood Systems Provide An Even More General Granulation Structure. For Each Element Of A Universe, One Associates It With A Nonempty Family Of Neighborhood Granules, Which Is Called A Neighborhood System. The Concept Of Neighborhood System Spaces Was Originally Introduced By Sierpenski And Krieger (See Any General Topology Course) For The Study Of A Generalization Of Topological Spaces. Yao [48] Used The Notion Neighborhood Systems For Granular Computing By Focusing On The Granulation Structures Induced By Neighborhood Systems. Zadeh [20] Studied The Relationships Between Fuzzy Sets And Information Granularity.

Mathematically, The Association Of Each Element With Such A Family Of Granules Is The Notion Of Neighborhood System Space. In Neighborhood System Space, Granules Are Called Neighborhoods And The Family Of Granules Which Is Associated With An Object X Is Called A Neighborhood System Of X And Is Denoted By NS (X).

For Data In An Information System, The

Acquisition Of Knowledge And Reasoning May Involve Vagueness, Incompleteness, And Granularity. In Order To Deal With The Incomplete And Vague Information On Classification, Concept Formulation, And Data Analysis, Researchers Have Proposed Many Methods Other Than Classical Logic, For Example, Rough Fuzzy Sets, Rough Set Theory And Its Generalizations [23-33], Computing With Words, Granular Computing, Formal Concept Analysis, Quotient Space Theory , And Computational Theory For Linguistic Dynamic Systems. The Advantage Of The Rough Set Method Is That It Does Not Need Any Additional Information About The Data, Like Probability In Statistics Or Membership In Fuzzy Set Theory. The Main Idea Of The Rough Theory Comes From Pawlak's Work [8,34]. Many Researchers Have Made Contributions To This Theory. Applications of The Rough Set And Fuzzy Set Theories Can Be Found In [33-34].

One Of The Nice Features Of Rough Set Theory Is That Rough Sets Can Tell Whether The Data Is Complete Or Not Based On The Data Itself. If The Data Are Incomplete, It Suggests More Information About The Objects Need To Be Collected In Order To Build A Better Classification Model. On The Other Hand, If The Data Is Complete, Rough Set Theory Can Also Determine Whether There Are More Than Enough Or Redundant Information In The Data And Find The Minimum Data Needed For A Classification Model [34]. This Property Of Rough Set Theory Is Very Important For Application Where Domain Knowledge Is Very Limited Or Data Collection Is Very Expensive Laborious Because It Makes Sure The Data Collected Is Just Good Enough To Build A Better Classification Model Without Sacrificing The Accuracy Of The Classification Model Or Wasting Time And Effort To Gather Extra Information About The Objects. Furthermore, Rough Set Theory Classifies All The Attributes Into Three Categories: Core Attributes Reduced Attributes And Dispensable Attributes. Core Attributes Have The Essential Information To Make The Correct Classification Of The Data Set And Should Be Retained In The Data Set; Dispensable Attributes Are The Redundant Ones In The Data Set And Should Be Eliminated; And Reduced Attributes Are In The Middle Between. Depending On The Combination Of The Attributes, In Some Cases, A Reduced Attribute Is Not Necessary, While In Other Situations It Is Essential [11, 28, 34].

2. Basic Document Retrieval System and Document Processing

In Document Retrieval, Some Processes Take Place Dynamically When The User Inputs Their Query, While Other Processes Take Place Off-Line In Advance And In Batch Mode And Do Not Involve Individual Users. These Static Processes Are Run On The Documents That Will Be Made Available In The

Retrieval System. These Will Be Explained First. Then, The Two Dynamic Processes, Query Processing And Matching, Will Be Presented. Figure 2.1 Provides A Simple, But Clear View Of The Relationship Between These Three Processes.

The First Two Steps In The Processing Of Documents Are Somewhat Mundane, But Necessary, And Can Be Considered As Batch Pre-Processing. These Are:

(1) Normalize Document Stream To A Predefined Format, Whereby Multiple External Formats (E.G. News Feeds, Web Pages, And Word Processed Documents) Are Standardized Into A Single Consistent Format. This Is An Essential Step (Much Akin To Data Clean-Up In Data Mining) As All Downstream Processes Rely On Receiving A Common Format They Can Recognize And Process. Preprocessing Is Particularly Vital For Systems With More Complex Processing Than Simple 'Characters Between White Spaces' Indexing.

(2) Break Document Stream Into Desired Retrievable Units, Whether This Is A Web Page, Chapter, Full Document, Paragraph, Etc. The Pointers Stored In The Inverted File Are To Whatever Unit Size Has Been Pre-Determined. Therefore, Document Retrieval Could In Fact Be Paragraph Retrieval, If The Indexable Unit Was Determined At This Stage To Be The Paragraph. From This Step Forward, The System Is Performing The Heart Of The Document Indexing Process.

(3) Identify Potential Indexable Elements In Documents. This Is A Key Decision Point That Dramatically Affects The Nature And Quality Of The Retrieval Performance. First, The Important Definition Needs To Be Made As To What Is A Term. Is It Any String Of Alphanumeric Characters Between Blank Spaces Or Punctuation? If So, Are Non Compositional Phrases Or Multi-Word Proper Names, Or Inter-Word Symbols Such As Hyphens Or Apostrophes Treated Differently (E.G. Are "Small Business Men" And "Small Business Men" The Same)? At This Stage, The System Requires A Set Of Rules To Be Executed Which Control What Actions Are Taken By The 'Tokenize' – The Algorithm Which Recognizes 'Indexable Terms'. IR Systems Vary As To Which Of These Processes They Perform, But The Most Frequently Used Processes Are:

(i) Delete Stop Words Via An Algorithm That Filters The Document's Potential Index-Able Elements Against A Stop Word List To Eliminate Terms That Are Deemed To Be Insignificant In Determining A Document's Relevance To A User's Request. The Original Objective In Using Stop Words Was To Save System Resources By Eliminating Those Terms That Have Little Value For Retrieval Performance. Although These Terms May Comprise Up To 40% Of The Tokens In A Document Set, Index Size Is Of Far Less Importance Today Due To Cheap Memory, But Their Omnipresence Renders Them Of Little Value To Retrieve. The Typical Word Classes That Are Marked As Stop Words Include The Function Word Classes And A Few More (I.E. Articles, Conjunctions, Interjections, Prepositions, Pronouns, And 'To Be' Verb Forms).

(ii) Stem Terms Of Removing Suffixes. In This Morphological Step, Some IR Systems Do Just Inflectional ('Weak') Stemming Which Only Changes The Subclass Within A Part-Of-Speech Category, I.E. Past Tense To Present Tense, While Others Also Do Derivational ('Strong') Stemming Which Removes Suffixes, Sometimes Recursively, That May Actually Change The Part Of Speech Of A Word. Use Of Stemming Will Result In Fewer Entries In An Index, Each Of Which Is Likely To Have Higher Frequency Counts Than If All Morphological Variants And Their Counts Are Used. The Initial Goal Of Stemming Was To Reduce The Storage Requirements Of The Inverted Index File By Reducing The Number Of Unique Words, But Stemming Has Remained In Use Even Today When Storage Is Not An Issue, Because It Improves Recall Of Relevant Documents. For Example, If A Query Includes "Analyze", The User May Well Want Documents Which Contain Analysis, Analyzing, Analyze, Or Analyzed. In Order For The System To Match On All These Variables, It Must Stem Both The Query And The Document Terms To Analyze. Obviously, Stemming May Negatively Impact Precision.

(iii) Bracket Noun Phrases, Usually By Means Of Regular Expressions Which Define The Part-Of-Speech Patterns Which Comprise A Noun Phrase (E.G. <ADJ NN> Or <NN NN>). This Is A Step That Can Negatively Affect Recall Of Retrieval Results By Either Excluding Documents When The Phrasal Expression In The Query Is Not Exactly The Same As The Index Entry Of A Document, Or Positively Affect Precision By Retrieving Only Documents That Include The Terms In The Desired Phrasal Expression.

(4) Produce An Inverted File Containing A Sorted Array Of All Indexable Terms (With Terms Defined As Referring To Either A Word Or A Phrase), Along With The Unique Identification Number Of Each Document In The Collection In Which The Term Occurs, A Link To Each Of These Documents, Weights For Each Term As Determined By The IR Model Being Implemented In The System [Which Will Be Described In The Next Section] And Optionally, The Within-Document Location Of The Term. More Sophisticated Systems May Include Further Information In The Inverted File, Such As Named Entity Category For Proper Names (I.E. PERSON,

ORGANIZATION, GEO-LOCATION, Etc.) But The Most Common Features Are Simply Termed, Document ID, And Weight.

2.2 Query Processing

The System's Internal Representation of the User's Question / Search Terms Is Typically Referred To As The Query. Most Of The Same Processes That Are Running On The Documents Are Also Run To Produce The Query, But There Are Some Unique Processes As Well. As Distinct From Document Processing, All Of The Query Processing Is Done In Real Time, While The User Awaits Their Documents. These Are:

1. Recognize Query Terms Vs. Special Operators, Such As "I Need Information About…" Which Do Not Convey The Topic Of The User's Information Need And Will Not Be Included In The Query Representation.

2. Tokenize Query Terms, A Process That Requires Similar Decisions As Were Described In The Document Processing Side – That Is Stop Word Deletion, Stemming, And Phrase Recognition.

3. Create Query Representation, Which Typically Follows Stop Word Removal And Stemming, And Which May Also Include Insertion Of Logical Operators Between Terms Requiring Co-Occurrence Or Simple Presence Of Only One Of The Arguments.

4. Expand Query Terms To Include Variant Terms That Refer To Or Relate To The Same Concept. These May Be Synonymous Terms That Are Found In An Electronic Thesaurus Such As Wordnet Or Terms That Are Highly Associated With The Query Term, Based On Co-Occurrence Statistics Preferably Computed On The Same Or A Similar Document Collection As The One On Which The Search Is Being Conducted. Query Expansion Relieves The User Of Needing To Generate All Conceptual Variants Of Their Search Terms And Is Likely To Improve Recall, But May Reduce Precision When The Erroneous Senses Of The Newly Introduced Terms Retrieve Irrelevant Documents. The Longer A Query Is, The Less Likelihood That Erroneous Senses Of Expanded Terms Will Have A Negative Impact, But Also The Less Likely That Expansion Will Contribute Much To The Retrieval Results.

5. Compute Query Term Weights. This Step Is Less Commonly Included In Document Retrieval Systems, Mainly Because It Is Difficult Both For Users To Know How To Assign Weights To Query Terms In A Way That Improves Retrieval Results, Or For Automatic Weighting, Since Queries Are Frequently So Short As To Give Little Evidence Of The Relative Importance Of The Query Terms As Most Terms Only Occur Once In A Single Query. Some NLP-Based Systems Have Positive Results From The Automatic Determination Of The 'Mandatory' Concept In A Query Which Is Then Assigned A Greater Weight .

2.3 Matching of Query to Documents

Once The Query Representation Is Produced, The Matching Process Begins. The Process Description Below May Be Easier To Follow If You Conceive Of Both The Query And The Documents As Vectors Of Terms, With Frequency Information Or Weights For Each Term In The Vector.

(1) Search Inverted File For Documents That Contain Terms In The Query. This Is Typically Done Using A Standard Binary Search. Each Document That Contains Any Of The Query Terms Becomes A Candidate For Retrieval.

(2) Compute Similarity Score Between Query And Each Candidate Document Using The Algorithm Prescribed By One Of The Four Document Retrieval Models Being Used. This Score Is Referred To As The Similarity Coefficient. The Scoring Mechanism For Each Of The Major Document Retrieval Models Will Be Detailed In The Next Section.

(3) Rank Order The Documents In Decreasing Order Based On The Scores Assigned Them By The Scoring Algorithm. This May Be Either Straightforward Ranking Based On The Similarity Coefficient, Or The System May Utilize Automatic Relevance Feedback Whereby The System Takes The Top N-Ranked Terms From The Top N-Ranked Documents As They Are Being Shown To The User, And Adds These Terms To The Query Representation And Reruns The Search With The Revised Query To Produce The Continuation Of The Ranked List Of Relevant Documents.

(4) Provide A List Of Perceived Relevant Documents To User Ranked By Similarity Score Between Query And Document. Systems That Utilize Other Sources Of Evidence Of The Value Of A Document To The Query, Such As Number Of Links From The Page/Document To Or From Other Pages/Documents, Would Integrate This Information And Produce A Potentially Different Ranked List.

(5) Allow For Query Modification By The User If User-Based Relevance Feedback Is Provided By The System. If So, Typically, The User Marks The Documents He/She Finds Relevant, Either Based On Just The Title And Brief Description Shown Them On The Initial List Or By Actually Reviewing The Full Document, Which They Can Link To From The Results Page.

(6) Perform Relevance Feedback Based On User's Input. The Algorithm For User-Based Relevance Feedback Is Typically The Same As That For Automatic Relevance Feedback As Described In Step 3 Above. The System Then Re-Runs The Search With The Revised, And Hopefully Improved Query And Produces A Revised Ranked List Of Documents.

The Relevance Feedback Loop Is Iterative And Can Be Performed As Many Times As The User Wants.

3. The Vector Space Model

The Vector Space Model [25] Is The Most Commonly Used Model In Document Retrieval Systems Today Due To Its Consistent, Proven Performance Across Multiple Implementations On Many Collections. Conceptually, In The Vector Space Model, A Document Is Represented By A Vector Of The Terms In The Document, And These Vectors Exist In Term Space, Which Is The Size Of All The Unique Terms In The Collection. Each Term Represents A Dimension In This Term Space And The Similarity Between A Query And A Document Is Measured By The Closeness Of The Query Vector And The Document Vector, Where Closeness Is Measured By The Angle Between The Two Vectors. Cosine Similarity Or The Inner / Dot Product Are Used To Compute The Angle Between Vectors. However, To Compute A Similarity Score Between Query And Documents In The Collection And Then Rank Order The Documents Based On Their Likely Relevance To The Query, It Is Typical To Use Weighted Vectors. The System Needs A Basis To Assign Weights To Both Query And Document Terms. While There Are Multiple Ways To Compute Such Weights, The Nearly Universal Way To Do This Is What Is Known As Turf / IDF – That Is Termed Frequency (Tf) Multiplied By Inverse Document Frequency (IDF) (Or Equivalently, Divided By Document Frequency). By Use Of This Weighting Scheme, The Vector Space Model Is Saying That The Best Indexing Terms Are Those That Occur With High Frequency In A Document (Tf) Relative To Their Occurrence In Other Documents In The Collection (IDF). The Tf Metric Is Considered An Indication Of How Well A Term Characterizes The Content Of A Document. Of Course, There Are Several Arguments That Might Be Made Against This View, Such As The Linguistic Phenomena Of Synonymy And Anaphora – Both Of Which Can Represent The Same Concept With Different Terms, Thereby Resulting In The Candidate Index Term Undercounting Conceptual Presence. The Idf, In Turn, Reflects The Number Of Documents In The Collection In Which The Term Occurs, Irrespective Of The Number Of Times It Occurs In Those Documents. Using These Two Metrics In Combination, A Query-To-Document Similarity Score Is Computed Between A Query And Each Document In The Collection. Based On These Similarity Scores, The Model Produces A Ranked List Of Documents In Terms Of Predicted Relevance To The Query.

Computing Tf And Idf Requires First Determining Which Features (E.G. Words, Phrases) Will Be Used In Representing Documents And Queries. It Might Be Of Interest To Linguists That Until Recently, Relatively Little Attention Was Paid To What These Features Were. While Noun Phrases Were Historically Used As Subject Headings In Library Catalogs And Controlled Vocabularies, Doing This Automatically Requires The Ability To Distinguish Noun Phrase Elements From Other Parts Of Speech, Which Was Beyond The State Of The Art When Document Retrieval Systems Were First Introduced. But, The Belief That Some Words Provide A Better Representation Of Documents And Queries Led To The Use Of A Stop Word List Which Excludes Closed Class Terms (E.G. Prepositions, Pronouns, Determiners) From Indexing, Thereby Leaving Nouns, Adjectives, Verbs, And Adverbs As The Feature Set. Advantages Of The Vector Space Model Are That, Distinct From The Boolean Model, It Allows Partial Matching Of Query And Document, And The Model's Easy Adaptability Via Adjustments To Its Parameters, Including Term-Weighting Schemes, Which Have Been Shown To Have A Major Impact On The Quality Of Retrieval Results. As Mentioned Above, The Vector Space Model's Performance Is Consistently Good With General Collections. The Disadvantages Of The Vector Space Model Include Its Weighting Schemes' Reliance On Information From Across The Database And The Need Therefore For Weights To Be Updated As The Database Changes. However, Research Has Shown That Less Frequent Updating Of Collection Figures Does Not Negatively Impact Performance Significantly If The Collection Is Large Enough.

4. Pawlak's Rough Set Model and Mixed Neighborhood System Model

In This Section, We Give An Exposition Of The Needed Definitions. Also, We Introduce The Notion Of Mixed Neighborhood Systems And A New Definition Of Accuracy Of The Approximations Of Sets Which Are Essential For Our Present Study.

Let U Be A Non-Empty Finite Set And R Be An Equivalence Binary Relation On U, Then The Lower And The Upper Approximations Of $X \subseteq U$ Are Defined Respectively As Follows:

$\underline{R}(X) = \{ X \in U \mid [X]_R \subseteq X \}, \overline{R}(X) = \{ X \in U \mid [X]_R \cap X \neq \emptyset \}$, Where $[X]_R$ Is The Equivalence Class Of X. Also, The Boundary, Positive And Negative Regions Of X Are Defined Respectively By $BON(X) = \overline{R}(X) - \underline{R}(X), POS(X) = \underline{R}(X)$ And $NEG(X) = U - \overline{R}(X)$.

The Accurate Measure Of A Subset $X \subseteq U$ Is Denoted By $A(X)$ And Is Defined By: $A(X) = \frac{|\underline{R}(X)|}{|\overline{R}(X)|}$, Where $|\overline{R}(X)| \neq 0$, Such That $|X|$ Is The Cardinality Of X. The Accuracy Measure Is Also Called The Accuracy Of The Approximation.

Now, We Are Going To Introduce A New Definition For The Accuracy Measure Of The Approximations In Pawlak Approximation Spaces.

Also, The Accurate Measure Of A Subset $X \subseteq U$ Is Denoted By $P(X)$ And Is Defined By:

$$P(X) = 1 - \frac{|BON(X)|}{|U|}.$$

In The Above Definition, It Is Obvious That $0 \leq P(X) \leq 1$. Moreover, If $P(X) = 1$ Then X Is Called R-Definable (Or R-Exact) Set. Otherwise, It Is Called R-Rough.

We Believe That Our Measure $P(X)$ Of The Accuracy Measure Of A Subset $X \subseteq U$ Is Accurate Than Pawlak's Measure $A(X)$ Since Our Measure Consider The Negative Region And Pawlak's Measure Does Not Consider It.

For Demonstrating The Above Idea Considers The Following Example.

Example 4.1 Let $U = \{T1, T2, T3, T4, T5\}$ Represent A Vocabulary Partitioned By The Equivalence Relation R Defined On U As Follows:

$R = \{(T1,T1), (T1,T4), (T2,T2), (T2,T3), (T3,T2), (T3,T3), (T4,T1), (T4,T4), (T5,T5)\}$. So, The Equivalence Classes Of R Are: $[T1]_R = [T4]_R = \{T1, T4\}$, $[T2]_R = [T3]_R = \{T2, T3\}$ And $[T5]_R = \{T5\}$. Hence, The Partition Induced By R Is $U/R = \{\{T1, T4\}, \{T2, T3\}, \{T5\}\}$. Let $X = \{T2, T4\}$ Be Any Document Of U . Thus $\underline{R}(X) = \emptyset$ And $\overline{R}(X) = \{T1, T2, T3, T4\}$. So We Have $A(X) = 0$ And $P(X) = 1/5$. Obviously, $P(X)$ Is Accurate Than $A(X)$ Since The Element Of The Set $NEG_R(X) = \{T5\}$ Is Surely Does Not Belong To X According To R. Furthermore, Let $Y = \{T1, T5\}$ Be Any Query Of U. So $\underline{R}(Y) = \{T5\}$ And $\overline{R}(Y) = \{T1, T4, T5\}$. Hence $A(Y) = 1/3$ And$\rho(Y) = 3/5$. Clearly, $P(Y)$ Is Accurate Than $A(Y)$ Since The Elements Of The Set $NEG_R(Y) = \{T2, T3\}$ Are Surely Do Not Belong To Y With Respect to$r$. Also, The Element Of $\underline{R}(Y) = \{T5\}$ Is Surely Belongs To Y According to$r$. Consequently, We Can Decide With Full Certainty That $T5 \in Y$ And $T2, T3 \notin Y$. Accordingly, The Accuracy Should Equal To $3/5$.

Let U Be A Non Empty Finite Vocabulary And $\mathcal{R}$ Be An Arbitrary Binary Relation On U, Then The Pair $\mathcal{K} = (U, \mathcal{R})$ Is Called A Generalized Approximation Space. The Right Neighborhood (Resp. Left Neighborhood) Of An Element $x \in U$ Is The Set $N_R(X) = \{ y \in U \mid x\mathcal{R}y \}$ (Resp. $N_L(X) = \{ y \in U \mid y\mathcal{R}x \}$). The Right Neighborhood System (Resp. Left Neighborhood System) Of An Element $x \in U$ Is The Class $NS_R(X) = \{ N_R(X) : x \in U\}$ (Resp. $NS_L(X) = \{ N_L(X) : x \in U\}$).

Example 4.2 Let $U = \{T1, T2, T3, T4, T5\}$ Be A Vocabulary And We Define A General Binary Relation $\mathcal{R} = \{(T1,T1), (T1,T2), (T2,T3), (T2,T5), (T4,T3), (T4,T4), (T5,T2), (T5,T4), (T5,T5)\}$ On U . Then We Have $N_R(T1) = \{T1, T2\}, N_R(T2) = \{T3, T5\}$, $N_R(T3) = \emptyset$, $N_R(T4) = \{T3, T4\}, N_R(T5) = \{T2, T4, T5\}, NS_R(T1) = \{\{T1, T2\}\}, NS_R(T2) = \{\{T3, T5\}\}, NS_R(T3) = \{\emptyset\}$, $NS_R(T4) = \{\{T3, T4\}\}$, And $NS_R(T5) = \{\{T2, T4, T5\}\}$. Also We Have $N_L(T1) = \{T1\}$, $N_L(T2) = \{T1, T5\}$, $N_L(T3) = \{T3, T4\}, N_L(T4) = \{T4, T5\}$, $N_L(T5) = \{T2, T5\}$, $NS_L(T1) = \{\{T1\}\}$, $NS_L(T2) = \{\{T1, T5\}\}$, $NS_L(T3) = \{\{T2, T4\}\}$, $NS_L(T4) = \{\{T4, T5\}\}$, And $NS_L(T5) = \{\{T2, T5\}\}$.

Let $\mathcal{K} = (U, \mathcal{R})$ Be A Generalized Approximation Space, Then The Mixed Neighborhood System Of An Element $x \in U$ Is The Class $NS_M(X) = \{ N_R(X), N_L(X): x \in U \}$. The Mixed Neighborhood Of An Element $x \in U$ Is Denoted By $N_M(X)$ Such That $N_M(X) \in NS_M(X)$.

Example 4.3 According To Example 2.2, The Mixed Neighborhood Systems Are Given By $NS_M(T1) = \{ \{T1, T2\}, \{T1\} \}, NS_M(T2) = \{ \{T3, T5\}, \{T1, T5\} \}, NS_M(T3) = \{ \emptyset, \{T2, T4\} \}, NS_M(T4) = \{ \{T3, T4\}, \{T4, T5\} \}$, And $NS_M(T5) = \{ \{T2, T4, T5\}, \{T2, T5\} \}$.

Let $\mathcal{R}$ Be An Arbitrary Binary Relation Defined On A Non-Empty Vocabulary U . Then The Right Interior Operator $Int_R: P(U) \longrightarrow P(U)$ And The Right Closure Operator $Cl_R: P(U) \longrightarrow P(U)$ Using Neighborhood System Are Defined Respectively As Follows: $Int_R(X) = \{X \in X \mid N_R(X) \subseteq X\}, Cl_R(X) = X \cup \{X \in U \mid N_R(X) \cap X \neq \emptyset\}$. The Left Interior Operator $Int_L: P(U) \longrightarrow P(U)$ And The Left Closure Operator $Cl_L: P(U) \longrightarrow P(U)$ Are Defined Respectively As Follows: $Int_L(X) = \{X \in X \mid N_L(X) \subseteq X\}$, $Cl_L(X) = X \cup \{X \in U \mid N_L(X) \cap X \neq \emptyset\}$.

The Lower And The Upper Approximations Of A Document X Of The Vocabulary U Using Right Neighborhood Systems Are Defined Respectively By: $\underline{\mathcal{R}}_R(X) = \{ X \in X \mid N_R(X) \subseteq X \}$, $\overline{\mathcal{R}}_R(X) = X \cup \{ X \in X^C \mid N_M(X) \cap X \neq \emptyset \}$. The Lower And The Upper Approximations Of A Document X Of U Using Left Neighborhood Systems Are Defined Respectively By: $\underline{\mathcal{R}}_L(X) = \{ X \in X \mid N_L(X) \subseteq X \}$, $\overline{\mathcal{R}}_L(X) = X \cup \{ X \in X^C \mid N_M(X) \cap X \neq \emptyset \}$. The Lower And The Upper Approximations Of A Document X Of U Using Mixed Neighborhood Systems Are Defined Respectively By: $\underline{\mathcal{R}}_M(X) = \{ X \in X \mid \exists N_M(X), N_M(X) \subseteq X \}$, $\overline{\mathcal{R}}_M(X) = X \cup \{ X \in X^C \mid \forall N_M(X), N_M(X) \cap X \neq \emptyset \}$.

The Boundary, Positive And Negative Regions Of Document X Using Right Neighborhood Systems Are Defined Respectively By: $\mathcal{B}_{\mathrm{R}}(\mathrm{X}) = \overline{\mathcal{R}}_{\mathrm{R}}(\mathrm{X}) - \underline{\mathcal{R}}_{\mathrm{R}}(\mathrm{X})$, $\mathrm{POS}_{\mathrm{R}}(\mathrm{X}) = \underline{\mathcal{R}}_{\mathrm{R}}(\mathrm{X}), \mathrm{NEG}_{\mathrm{R}}(\mathrm{X}) = \mathrm{U} - \overline{\mathcal{R}}_{\mathrm{R}}(\mathrm{X})$. The Boundary, Positive And Negative Regions Of Document X Using Left Neighborhood Systems Are Defined Respectively By: $\mathcal{B}_{\mathrm{L}}(\mathrm{X}) = \overline{\mathcal{R}}_{\mathrm{L}}(\mathrm{X}) - \underline{\mathcal{R}}_{\mathrm{L}}(\mathrm{X})$, $\mathrm{POS}_{\mathrm{L}}(\mathrm{X}) = \underline{\mathcal{R}}_{\mathrm{L}}(\mathrm{X}), \mathrm{NEG}_{\mathrm{L}}(\mathrm{X}) = \mathrm{U} - \overline{\mathcal{R}}_{\mathrm{L}}(\mathrm{X})$. The Boundary, Positive And Negative Regions Of Document X Using Mixed Neighborhood Systems Are Defined Respectively By: $\mathcal{B}_{\mathrm{M}}(\mathrm{X}) = \overline{\mathcal{R}}_{\mathrm{M}}(\mathrm{X}) - \underline{\mathcal{R}}_{\mathrm{M}}(\mathrm{X})$, $\mathrm{POS}_{\mathrm{M}}(\mathrm{X}) = \underline{\mathcal{R}}_{\mathrm{M}}(\mathrm{X}), \mathrm{NEG}_{\mathrm{M}}(\mathrm{X}) = \mathrm{U} - \overline{\mathcal{R}}_{\mathrm{M}}(\mathrm{X})$.

Let $\mathcal{K} = (\mathrm{U}, \mathcal{R})$ Be A Generalized Approximation Space, Then The Accuracy Of The Approximations Of A Document $\mathrm{X} \subseteq \mathrm{U}$ Using (Right, Left And Mixed) Neighborhood Systems Are Defined Respectively By: $\Sigma_{\mathrm{R}}(\mathrm{X}) = 1 - \frac{|\mathcal{B}_{\mathrm{R}}(\mathrm{X})|}{|\mathrm{U}|}$, $\Sigma_{\mathrm{L}}(\mathrm{X}) = 1 - \frac{|\mathcal{B}_{\mathrm{L}}(\mathrm{X})|}{|\mathrm{U}|}$, $\Sigma_{\mathrm{M}}(\mathrm{X}) = 1 - \frac{|\mathcal{B}_{\mathrm{M}}(\mathrm{X})|}{|\mathrm{U}|}$.

It Is Obvious That $0 \leq \Sigma_{\mathrm{R}}(\mathrm{X}) \leq 1, 0 \leq \Sigma_{\mathrm{L}}(\mathrm{X}) \leq 1$ And $0 \leq \Sigma_{\mathrm{M}}(\mathrm{X}) \leq 1$. Moreover, If $\Sigma_{\mathrm{R}}(\mathrm{X}) = 1$ Or $\Sigma_{\mathrm{L}}(\mathrm{X}) = 1$ Or $\Sigma_{\mathrm{M}}(\mathrm{X}) = 1$ Then X Is Called $\mathcal{R}$ - Definable (Or $\mathcal{R}$-Exact) Document . Otherwise, It Is Called $\mathcal{R}$-Rough Document .

Example 4.4 Let $\mathrm{U} = \{\mathrm{T1}, \mathrm{T2}, \mathrm{T3}, \mathrm{T4}, \mathrm{T5}\}$ Be A Vocabulary Set And $\mathcal{R} = \{(\mathrm{T1}, \mathrm{T2}), (\mathrm{T1}, \mathrm{T4}), (\mathrm{T2}, \mathrm{T2}), (\mathrm{T2}, \mathrm{T3}), (\mathrm{T2}, \mathrm{T4}), (\mathrm{T4}, \mathrm{T5}), (\mathrm{T4}, \mathrm{T3}), (\mathrm{T5}, \mathrm{T2}), (\mathrm{T5}, \mathrm{T5})\}$ Be Any Binary Relation On U . Thus We Get $\mathrm{NS}_{\mathrm{M}}(\mathrm{T1}) = \{\{\mathrm{T2}, \mathrm{T4}\}, \emptyset\}, \mathrm{NS}_{\mathrm{M}}(\mathrm{T2}) = \{\{\mathrm{T2}, \mathrm{T3}, \mathrm{T4}\}, \{\mathrm{T1}, \mathrm{T2}, \mathrm{T5}\}\}$, $\mathrm{NS}_{\mathrm{M}}(\mathrm{T3}) = \{\emptyset, \{\mathrm{T2}, \mathrm{T4}\}\}$, $\mathrm{NS}_{\mathrm{M}}(\mathrm{T4}) = \{\{\mathrm{T3}, \mathrm{T5}\}, \{\mathrm{T1}, \mathrm{T2}\}\}$, And $\mathrm{NS}_{\mathrm{M}}(\mathrm{T5}) = \{\{\mathrm{T2}, \mathrm{T5}\}, \{\mathrm{T4}, \mathrm{T5}\}\}$.

Accordingly, Table 1 Shows The Differences Among $\Sigma_{\mathrm{R}}(\mathrm{X})$, $\Sigma_{\mathrm{L}}(\mathrm{X})$ And $\Sigma_{\mathrm{M}}(\mathrm{X})$ For Some Documents Of The Vocabulary U.

Table 1: Differences Among The Measures $\sigma_r(X)$, $\sigma_l(X)$ And $\sigma_m(X)$

Document Set	$\sigma_r(X)$	$\sigma_l(X)$	$\sigma_m(X)$
$\{t1\}$	$4/5$	$3/5$	1
$\{t1, t2, t4, t5\}$	$3/5$	$4/5$	1
$\{t1, t2\}$	$2/5$	$2/5$	$4/5$
$\{t2, t3, t4, t5\}$	$4/5$	$3/5$	1

5. Conclusions

In This Paper, We Proved That The Approximations Based On Mixed Neighborhood Systems Are Accurate Than The Approximations Based On Either Right Neighborhood Systems Or Left Neighborhood Systems. By Using Both Of Them, We Mean Defining The Lower Approximation Of $\mathrm{A} \subseteq \mathrm{U}$ By $\underline{\mathcal{R}}_{\mathrm{R}}(\mathrm{A}) \cup \underline{\mathcal{R}}_{\mathrm{L}}(\mathrm{A})$ And The Upper Approximation By $\overline{\mathcal{R}}_{\mathrm{R}}(\mathrm{A}) \cap \overline{\mathcal{R}}_{\mathrm{L}}(\mathrm{A})$. Furthermore, We Believe That Our Definition Of The Accuracy Measure Of A Document $\mathrm{A} \subseteq \mathrm{U}$ Is Accurate Than Pawlak's Definition Since Our Definition Consider The Negative Region And Pawlak's Definition Does Not Consider It. Using Mixed Neighborhood Systems Open The Door About Many Applications In Finding The Attributes Missing Values And Topological Generalizations. Also, In The Domain Of Data Reduction And Data Mining This Approach Will Have High Voted [7,13-15,19]

References

1. Salton G., A. Wong And C. Yang, A Vector Space Model For Automatic Indexing, Communications Of The ACM", 18(11), 613–620 (1975).
2. Gupta S., K. Patnaik, Enhancing Performance Of Face Recognition Systems By Using Near Set Approach For Selecting Facial Features, J. Theor. Appl. Inform. Technol. 4 (5) (2008) 433–441.
3. Hassanien A., A. Abraham, J. Peters, G. Schaefer, C. Henry, Rough Sets And Near Sets In Medical Imaging: A Review, IEEE Trans. Info. Tech. Biomed. Volume 13 Issue 6, November 2009, Pages 955-968.
4. Pawlak Z., A. Skowron, Rough Sets And Boolean Reasoning, Inform. Sci. 177 (2007) 41–73.
5. Pawlak Z., A. Skowron, Rough Sets: Some Extensions, Inform. Sci. 177 (2007) 28–40.
6. Pawlak Z., Classification Of Objects By Means Of Attributes, Polish Acad. Sci., 429.
7. Pawlak Z.,Some Issues On Rough Sets, Trans. Rough Sets 21 (2004) 1–58.
8. Degang C. , Y. Wenxia , L. Fachao; Measures Of General Fuzzy Rough Sets On A Probabilistic Space. Information Sciences 178 (2008) 3177–3187, Doi:10.1016/J.Ins.2008.03.020.
9. Tuan-Fang Fan, Churn-Jung Liau, Duen-Ren Liu; A Relational Perspective Of Attribute Reduction In Rough Set-Based Data Analysis. European Journal Of Operational Research, In Press, Corrected Proof, Available Online 20 August 2010.
10. Hu Q., D. Yu, J. Liu, C. Wu; Neighborhood Rough Set Based Heterogeneous Feature Subset Selection. Information Sciences 178 (2008) 3577–3594, Doi:10.1016/J.Ins.2008.05.024.
11. Yee Leung, Wei-Zhi Wu, Wen-Xiu Zhang; Knowledge Acquisition In Incomplete Information Systems: A Rough Set Approach. European Journal Of Operational Research, Volume 168, Issue 1, 1 January 2006, Pages 164-180 .

12. Yao Y., Y. Zhao; Attribute Reduction In Decision-Theoretic Rough Set Models. Information Sciences 178 (2008) 3356–3373, Doi:10.1016/J.Ins.2008.05.010.
13. Pawlak Z., A. Skowron, Rudiments Of Rough Sets, Inform. Sci. 177 (2007) 3–27.
14. Pawlak Z., Rough Sets, Int. J. Comput. Inform. Sci. 11 (1981) 341–356.
15. Pawlak Z.,, Rough Sets – Theoretical Aspects Of Reasoning About Data, Kluwer Academic Publishers, Dordrecht, 1991.
16. Polkowski L., A. Skowron, Rough Mereology: A New Paradigm For Approximate Reasoning, Int. J. Approx. Reason. 15 (4) (1997) 333–365.
17. Polkowski L., Rough Sets. Mathematical Foundations, Springer-Verlag, Heidelberg, Germany, 2002.
18. Brtka V., E. Stokic, B. Srdic; Automated Extraction Of Decision Rules For Leptin Dynamics—A Rough Sets Approach. Journal Of Biomedical Informatics 41 (2008) 667–674, Doi:10.1016/J.Jbi.2008.01.005.
19. Davvaz B. ; A Short Note On Algebraic T-Rough Sets. Information Sciences 178 (2008) 3247–3252, Doi:10.1016/J.Ins.2008.03.014.
20. Zadeh L. A.: Fuzzy Sets And Information Granularity, In: Advances In Fuzzy Set Theory And Applications, N. Gupta, R. Ragade, And R. Yager (Eds.), North-Holland, Amsterdam, 1979, Pp. 3-18.
21. Muhammad Irfan Ali; A Note On Soft Sets, Rough Soft Sets And Fuzzy Soft Sets. Applied Soft Computing, Volume 11, Issue 4, June 2011, Pages 3329–3332,
22. Krzysztof Dembczyński, Salvatore Greco, Roman Słowiński; Rough Set Approach To Multiple Criteria Classification With Imprecise Evaluations And Assignments. European Journal Of Operational Research, Volume 198, Issue 2, 16 October 2009, Pages 626-636.
23. Asit Kumar Das, Jaya Sil; An Efficient Classifier Design Integrating Rough Set And Set Oriented Database Operations. Applied Soft Computing, Volume 11, Issue 2, March 2011, Pages 2279-2285.
24. Fotea V. L.; The Lower And Upper Approximations In A Hypergroup, Information Sciences 178 (2008) 3605–3615, Doi:10.1016/J.Ins.2008.05.009.
25. Masahiro Inuiguchi, Takuya Miyajima; Rough Set Based Rule Induction From Two Decision Tables. European Journal Of Operational Research, Volume 181, Issue 3, 16 September 2007, Pages 1540-1553.
26. Ken Kaneiwa; A Rough Set Approach To Multiple Dataset Analysis. Applied Soft Computing, Volume 11, Issue 2, March 2011, Pages 2538-2547.
27. Slowinski R., D. Vanderpooten, A Generalized Definition Of Rough Approximations Based On Similarity, IEEE Trans. Knowledge Data Eng. 12 (2000) 331–336.
28. Yang Y., R. I. John ; Generalizations Of Roughness Bounds In Rough Set Operations. International Journal Of Approximate Reasoning 48 (2008) 868–878, Doi:10.1016/J.Ijar.2008.02.002.
29. Puzio L., A. Walczak, Adaptive Edge Detection Method For Images, Opto-Elect. Rev. 16 (1) (2008) 60–67.
30. Randen T., J. Husoy, Filtering For Texture Classification: A Comparative Study, IEEE Trans. Pattern Anal. Mach. Intell. 21 (4) (1999) 291–310.
31. Wang X. , J. Zhai, S. Lu; Induction Of Multiple Fuzzy Decision Trees Based On Rough Set Technique. Information Sciences 178 (2008) 3188–3202, Doi:10.1016/J.Ins.2008.03.021.
32. Zhao S., E. C.C. Tsang ; On Fuzzy Approximation Operators In Attribute Reduction With Fuzzy Rough Sets, Information Sciences 178 (2008) 3163–3176, Doi:10.1016/J.Ins.2008.03.022.
33. Zhang S. Mining Class-Bridge Rules Based On Rough Sets, Expert Systems With Applications (2008), Doi: 10.1016/J.Eswa.2008.07.044, PII: S0957-4174(08)00511-3, Reference: ESWA 3038.
34. Yao Y.Y., Constructive And Algebraic Methods Of Theory Of Rough Sets, Inform. Sci. 109 (1998) 21–47.

3/5/2014