POLITECNICO DI TORINO

PhD in Information and System Engineering
XXVI cycle

III Facoltà di Ingegneria
Settore scientifico ING-INF/05

PhD Thesis

Document analysis by means of data mining techniques

Author:
Saima Jabeen
Matr. 168999

Supervisor:
Prof. Elena Baralis

March 2014
A.A. 2013/2014

In loving memory of my father

Acknowledgments

Thanks to Allah Almighty for His countless blessings and for making my directions enlighten.

First and foremost, I would like to thanks my supervisor Professor Elena Baralis for her kind supervision with valuable suggestions and comments. Her advices were precious and fundamental for me to accomplish my tasks on time.

Next, I would like to show my deepest gratitude to Dr. Alessandro Fiori, a respectable, responsible and open minded person and Dr. Luca Cagliero who is very kind, always energetic and enthusiastic scholar. Both of them have provided me valuable guidance by their very active role in my PhD activities and in writing of this thesis. I could not have completed my thesis without their enlightening instructions, impressive kindness and patience.

I also appreciate my Lab fellow Dr. Alberto Grand, a very intelligent and so kind person who always helped me for my lab matters.

I shall extend my thanks to Politecnico di Torino for providing me a good platform to obtain deeper and broader training in Computer Engineering. My sincere appreciation also goes to Higher Education Commission of Pakistan (HEC), without which, I would not have the opportunity to study abroad and get sponsored.

Last but not least, I'd like to thank all my families and friends especially my loving father (Late) for his guidance and belief in me that brought me to this stage. I am very pleased to fulfill his wish to be a doctorate but at the same time he is highly missed on this stage. I am also very grateful to my beloved husband for his love, care and help in my hard times and I find myself in short of words to express my gratitude to him for his presence which gives me strength in every walk of life.

Contents

Chapter 1

Introduction

The huge amount of textual data produced everyday by scientists, journalists and Web users, allows investigating many different aspects of information stored in the published documents. Data mining and information retrieval techniques are exploited to manage and extract information from huge amount of unstructured textual data. Text mining also known as text data mining is the processing of extracting high quality information (focusing relevance, novelty and interestingness) from text by identifying patterns etc. Text mining typically involves the process of structuring input text by means of parsing and other linguistic features or sometimes by removing extra data and then finding patterns from structured data. Patterns are then evaluated at last and interpretation of output is performed to accomplish the desired task. Recently, text mining has got attention in several fields such as in security (involves analysis of Internet news), for commercial (for search and indexing purposes) and in academic departments (such as answering query). Beyond searching the documents consisting the words given in a user query, text mining may provide direct answer to user by semantic web for content based (content meaning and its context). It can also act as intelligence analyst and can also be used in some email spam filters for filtering out unwanted material [30]. Text mining usually includes tasks such as clustering, categorization, sentiment analysis, entity recognition, entity relation modeling and document summarization.

In particular, summarization approaches are suitable for identifying relevant sentences that describe the main concepts presented in a document dataset. Furthermore, the knowledge existed in the most informative sentences can be employed to improve the understanding of user and/or community interests.

Different approaches have been proposed to extract summaries from unstructured text documents. Some of them are based on the statistical analysis of linguistic features by means of supervised machine learning or data mining methods, such as Hidden Markov models, neural networks and Naive Bayes methods. An appealing research field is the extraction of summaries tailored to the major user interests. In this context, the problem of extracting useful information according to domain knowledge related to the user interests is a challenging task.

The main topics have been to study and design of novel data representations and data mining algorithms useful for managing and extracting knowledge from unstructured documents. This thesis describes an effort to investigate the application of data mining approaches, firmly established in the subject of transactional data (e.g., frequent itemset mining), to textual documents. Frequent itemset mining [7] is a widely exploratory technique to discover hidden correlations that frequently occur in the source data. Although its application to transactional data is well-established, the usage of frequent itemsets in textual document summarization has never been investigated so far. A work is carried on exploiting frequent itemsets for the purpose of multi-document summarization so a novel multi-document summarizer, namely ItemSum (Itemset-based Summarizer) is presented, that is based on an itemset-based model, i.e., a framework comprise of frequent itemsets, taken out from the document collection. Highly representative and not redundant sentences are selected for generating summary by considering both sentence coverage, with respect to a sentence relevance score, based on tf-idf statistics, and a concise and highly informative itemset-based model. To evaluate the ItemSum performance a suite of experiments on a collection of news articles has been performed. Obtained results show that ItemSum significantly outperforms mostly used previous summarizers in terms of precision, recall, and F-measure. We also validated our approach against a large number of approaches on the DUC'04 [50] document collection. Performance comparisons, in terms of precision, recall, and F-measure, have been performed by means of the ROUGE [83] toolkit. In most cases, ItemSum significantly outperforms the considered competitors. Furthermore, the impact of both the main algorithm parameters and the adopted model coverage strategy on the summarization performance are investigated as well.

In some cases, the soundness and readability of the generated summaries are unsatisfactory, because the summaries do not cover in an effective way all the semantically relevant data facets. A step beyond towards the generation of more accurate summaries has been made by semantics-based summarizers (e.g., [43, 44]). Such approaches combine the use of general-purpose summarization strategies with ad-hoc linguistic analysis. The key idea is to also consider the semantics behind the document content to overcome the limitations of general-purpose strategies in differentiating between sentences based on their actual meaning and context. Most of the previously proposed approaches perform the semantics-based analysis as a preprocessing step that precedes the main summarization process. Therefore, the generated summaries could not entirely reflect the actual meaning and context of the key document sentences. In contrast, we aim at tightly integrating the ontology-based document analysis into the summarization process in order to take the semantic meaning of the document content into account during the sentence evaluation and selection processes. With this in mind, we propose a new multi-document summarizer, namely Yago-based Summarizer, that integrates an established ontology-based entity recognition and disambiguation step. Named Entity Recognition from Yago ontology is being used for the task of text summarization. The Named Entity Recognition (NER) task is concerned with marking occurrences of a specific object being mentioned. These mentions are then classified into a set of predefined categories. Standard categories include "person", "location", "geo-political organization", "facility", "organization", and "time". The use of NER in text summarization improved the summarization process by increasing the rank of informative sentences. To demonstrate the effectiveness of the proposed approach, we compared its performance on the DUC'04 benchmark document collections with that of a large number of state-of-the-art summarizers. Furthermore, we also performed a qualitative evaluation of the soundness and readability of the generated summaries and a comparison with the results that were produced by the most effective summarizers.

A parallel effort has been devoted to integrating semantics-based models and the knowledge acquired from social networks into a document summarization model named as SociONewSum. The effort addresses the sentence-based generic multi-document summarization problem, which can be formulated as follows: given a collection of news articles ranging over the same topic, the goal is to extract a concise yet informative summary, which consists of most salient document sentences. An established ontological model has been used to improve summarization performance by integrating a textual entity recognition and disambiguation step. Furthermore, the analysis of the user-generated content coming from Twitter has been exploited to discover current social trends and improve the appealing of the generated summaries. An experimental evaluation of the SociONewSum performance was conducted on real English-written news article collections and Twitter posts. The achieved results demonstrate the effectiveness of the proposed summarizer, in terms of different ROUGE scores [83], compared to state-of-the-art open source summarizers as well as to a baseline version of the SociONewSum summarizer that does not perform any UGC analysis. Furthermore, the readability of the generated summaries has also been analyzed.

This thesis is organized as follows. Chapter 2 overviews document collection analysis in terms of text mining algorithms. Chapter 3 presents a study on knowledge representation and extraction schemes such as frequent itemset extraction and ontologies etc. Chapter 4 presents the ItemSum (Itemset-based Summarizer) multi-document summarizer. Chapter 5 addresses the use of semantics in summarizing multiple documents. Chapter 6 presents SociONewSum, a summarizer based on integration of semantics and social knowledge to improve the performance of multi-document summarization model on real news document collection. Finally, Chapter 7 draws conclusions and presents some future developments for the discussed approaches.

Chapter 2

Document collection analysis

This chapter presents a suite of analysis tools used for mining text of document collections. Introduction of text mining is presented in section 2.1. An overview of some established and extensively used text mining approaches is presented in section 2.2 along with their merits and demerits. Section 2.3 and 2.4 briefly discuss some categories of summarization models and some types of document collections respectively.

2.1 Introduction

Data mining is the process to discover meaningful knowledge from huge data sources, databases and data warehouses whereas Text mining (TM) or text data mining mentioned for the first time in [54] is the automated or partially automated processing of textual data. For the enormous increase in volume of electronic documents with the time in World Wide Web (WWW), it is becoming mountain to climb to analyze these document collections. Text mining as a set of techniques pitched for discovering information from large collections of texts. There are text mining approaches adopted to process documents for retrieving information needed. Labor-intensive manual methods of text mining first appeared in the mid-1980s, but they rapidly became advance because of technological progress in the past decade.

According to an estimation, currently 80% information is saved as text so there is a vast scope of text mining to explore these text collections. Text mining includes application of approaches from several fields e.g., information retrieval as well as information extraction and links with KDD, data

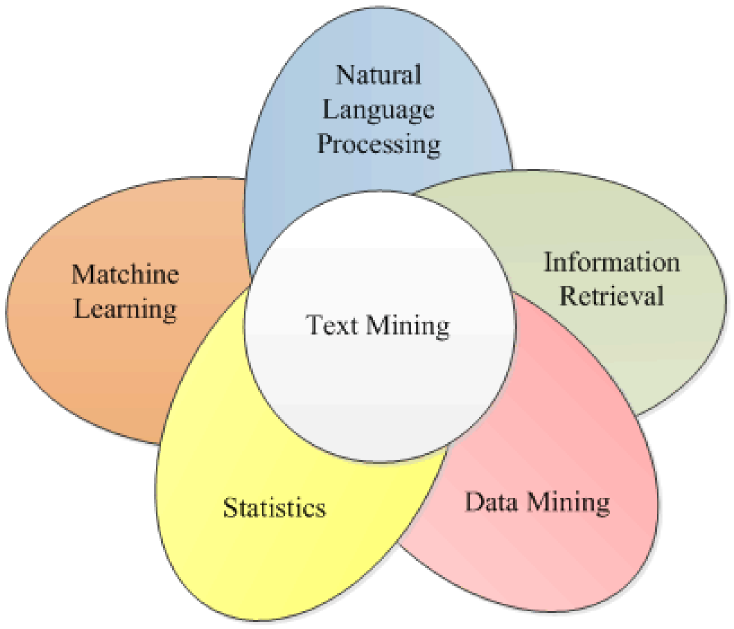


Figure 2.1: Text Mining as a multi-disciplinary field

mining, statistics, machine learning and natural language processing or computational linguistics as depicted in figure-2.1.

With the aim to attain high quality information from textual data, text mining involves the processing of unstructured input text (by means of some linguistic features, removal of noisy fragments, parsing and placing it in a database), then patterns are obtained from structured data which are finally evaluated and presented as a result. Relevance, interestingness and novelty all together measure the output of the text mining process [30]. Following are the brief topics closely related to the text mining process:

- **Text Normalization** is the initial task of text mining process where text is transformed into a single canonical form which guarantees consistency before actual processing. Since, there is no all-purpose normalization method so it should be known that what type of text is being normalized and where it will be used [121]. Text normalization is extensively performed when text to speech conversion is made or for storing and searching text in a database. It deals with acronyms, abbreviations, disambiguated words, dates and numbers [122]. Sometimes, text is normalized by removing diacritical signs and bringing the text to a single case and/or stemming and removal of stop-words may also be required.

- **Information Retrieval (IR)** systems assist text mining process with the aim to retrieve documents from a collection which satisfy or match a user's query [93]. These systems facilitate users by concentrating the documents dataset related to a specific problem. Search engines are popular IR systems e.g., Google aims to find those documents on the Internet that are pertinent to the provided query words. These systems has also exclusive application in libraries where data about documents is stored as digital records not the books themselves. IR system makes the task faster for text mining which involves computationally intensive algorithms by minimizing the quantity of documents for the purpose of analysis.

- **Natural Language Processing (NLP)** is the analysis of human language in its written electronic form so that computers could understand like humans [93]. This goal is not completely achieved yet but NLP performs some kinds of analysis with appreciable accuracy such as shallow as well as deep parsing. Shallow parsers recognize only noun and verb phrases which are major grammatical components of a sentence whereas deep parsers present a complete picture of sentence in terms of its grammatical structure. Text mining process takes the benefit of NLP in terms of its provided linguistic data to the systems in the information extraction stage in order to perform their task(s). More specifically, information extraction tools get benefit by annotated documents with sentence boundaries, part-of-speech (POS) tags and parsing outcomes etc.

- **Text Data Mining (TDM)** can be defined closely to data mining consisting a set of partial steps. These steps are the use of data mining, information extraction or statistical approaches. It can also be defined as the application of algorithms and methods from the machine learning and statistics with the aim to unveil unknown and meaningful knowledge by identifying patterns in huge document collections. To accomplish its task(s), there is need of preprocessing the text accordingly. Usually, some easy pre-processing steps, information extraction approaches or natural language processing are used by text mining techniques for preprocessing task. Afterwards, data mining algorithms can be exploited on the extracted data. Outcomes of mining process are stored in a separate database so that user could input his query to this database through an appropriate graphical interface then visual view of query results can also be made [93].

- **Information Extraction (IE)** is the automated task for bringing unstructured textual data into its structured form by exclusively using the data produced by NLP systems [93]. TM/TDM essentially corresponds to information extraction (extracting facts from text). In this process, it is required to define a general state of information which can be interestingly different for different users guiding the extraction process. Several processing steps can be performed such as tokenization, POS (Part-of-speech) tagging, sentence segmentation and named entity recognition i.e., name of person, location, organization etc. under the umbrella of IE. Phrases and sentences are parsed at a higher level and then semantically interpreted and combined afterwards to bring text in structured form which is the organization of entities and relations in regular and accessible form making the tasks easy for text mining process.

2.2 Text mining algorithms

Text mining and data mining techniques are often considered similar that is why they are often described together [22, 55, 70, 139] but there is a fine line that separates them i.e., nature of the input provided to them. Data mining techniques works on the format of highly structured data governed by extensive data preparation which expects to receive either highly structured data or first transforms original data into structured format while text mining usually operates on human readable textual document collections. Having this difference, still they share a lot of concepts and techniques so usually similar approaches are selected for mining data or text analysis. Main applications such as feature extraction, clustering and classification truly belong to text mining while regression and anomaly detection functions are suitable for diverse type of data like for both unstructured and structured data. The unstructured form of textual documents makes their access difficult for the users so data mining approaches are applied to structure these document collections for providing users easy access to a document collection. Book indexes/library catalogs are very familiar access structures but its manual indexing is tedious and time taking process in order to maintain the records so it is not applicable for rapidly changing information channels such as World Wide Web. To structure document collections, existing methods classify or categorize the documents by assigning keywords from a provided set of keywords or they aim to automatic grouping of similar documents based on clustering approaches. Therefore, text mining focuses data classification and finding associations among data and tries to overcome data mining problems by the following algorithms:

2.2.1 Classification

In classification, a decision class label is assigned to a set of unclassified objects illustrated by a determined set of features [93]. In other words, classification is predicting an instance class on the basis of its description. Classification problem in its *hard version* assigns a particular label to the instance whereas a probability value is assigned to the test instance in its *soft version*. Classification in its other variations lets for a test instance to rank various class choices or multiple labels [62] are allowed to assign to a test instance. Although continuous values are also possible to use as labels in classification problem but usually categorical values are assumed for the labels. The former is known as regression modeling problem.

Text classification problem specifically concerns to classify records using set-valued features [39]; but it considers information regarding the presence or absence of words, which is used in a document. Infact, frequency of words also has a significant role in classification process and conventional domain-size of textual content is much bigger than a common set-valued task of classification [6].

Classification task is widely performed in data mining, database, information retrieval, knowledge discovery and machine learning communities. It has applications in various domains such as news filtering and organization (also known as text filtering), document organization and retrieval, medical diagnosis and target marketing [6]. There is a large volume of electronic news articles produced by news services on daily basis over the Internet. Since, it is very difficult to manually organize this news collection so there is need of automatic approaches to categorize the news in a variety of web portals [79]. Beyond news filtering and organization, text filtering is useful for many other applications. In the context of document organization and retrieval, supervised methods play their role to organize the documents in various domains such as web repositories, digital libraries, research articles and social feeds.

The organization of document collections in hierarchical form brings ease in browsing and retrieving information from document collections [34]. In the application of opinion mining, customer's short text messages as reviews or opinions are mined in order to get useful information. In email classification and spam filtering (also known as email filtering) application, emails are classified [33, 38, 81] to find subject or junk email [113].

There are many approaches developed for text classification while some broad categories of these approaches and their usefulness for classification problem will be discussed in this section. These categories also work for other contexts like categorical and quantitative data. As, words frequency measures can model the text as quantitative data therefore, many quantitative methods can also be used on text but text is a special kind of data where word attributes are sparsed and for majority of words in high dimensional text can have low frequency count that makes the design of classification methods difficult. A survey of some major categories of text classification algorithms is presented below:

Decision Trees

Decision trees use different text features to design hierarchical ordering of the underlying training data that creates class divisions more skewed regarding their class distribution [6]. A most likely partition is determined for a given text instance to belong to and is used for classification purposes. In other words, training data is decomposed in hierarchical order where hierarchical division of data space (training data) is made using a condition or predicate on the attribute value as depicted in figure-2.2. These predicates are the conditions on presence or absence of words in the text of document collections.

Such predicates are typically conditions on the presence or absence of one or more words in the document of textual collection. In decision tree, a recursive process divides the data space till the leaf nodes meets a threshold of minimum number of records. For classification purposes, the majority class label in the leaf node is used. A sequence of predicates at the nodes is applied for a given test instance to traverse a tree-path in top-down manner then related leaf node is determined this way. The hold out data portion is used in order to take decision of pruning or not the constructed leaf node. It is determined to prune the node when it overfits the data space by observing that the class distribution in data space (for constructing DT) does not match with the class distribution in the training which is used for pruning.

The terms in the textual dataset specify the predicates for the DT nodes e.g., the information about existence or absence of a specific term in the text is used to partition a node into its children nodes. Different terms may be used to partition different nodes of decision tree at the same level. There can be many other types of predicates for example, similarity measure of documents to correlated sets of terms can be used for partitioning as well

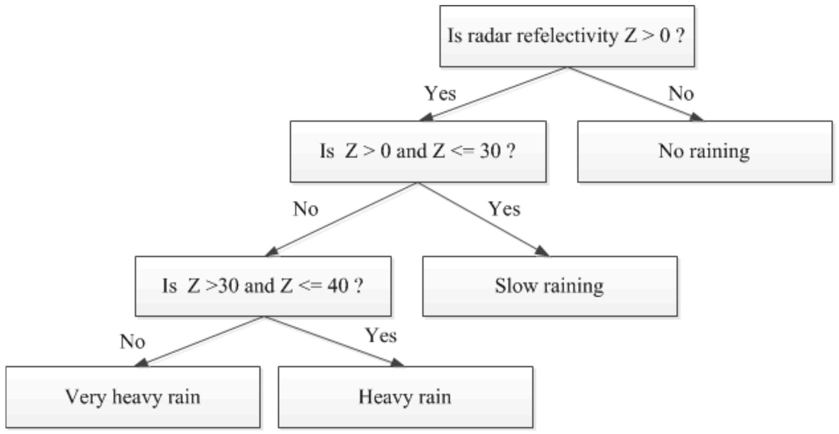


Figure 2.2: Example of a Decision Tree

and further document partitioning may also take place using these similarity measures. There are different splits such as *Single Attributes Splits*, *Similarity-based multi-attribute split* and *Discriminant-based multi-attribute split* to partition the dataset. In *Single Attribute Splits*, the split is performed using the information of presence and absence of words or phrases at a specific tree node and where the word providing the maximum discrimination between different classes is picked at any given level. Gini-index or information gain measures can be used to determine the entropy level. In *Similarity-based multi-attribute split*, split is performed by the the similarity of the documents to words clusters (such as frequent word clusters). For the selected word cluster, the documents are further divided into groups by rank-ordering the documents by similarity value and splitting at a particular threshold. That word cluster is selected for which rank-ordering by similarity provides best separation between the different classes. For multi-attribute case, the best choice for performing the split is *Discriminant-based multi-attribute split* such as Fisher discriminant. The documents are projected on discriminant vector for rank-ordering and split is performed at particular coordinate. The choice of split is selected to maximize the discrimination among different classes.

- **Advantages:** Decision trees can be visualized and simple to understand and interpret. DTs need little data preparation unlike other data mining techniques which require data normalization, creation of temporary variables and removal of empty values. However, it does not support missing values. It is able to deal multi-output problems as well as works for both categorical and numerical data unlike other techniques such as relation rules only deal with nominal variables and neural networks deal with numerical values. Also being a white box model, it can be explained easily. Statistical tests for its validation are possible to judge the reliability of this model. It is robust and also shows good performance when deals large datasets in reasonable time.

- **Disadvantages:** The possibility of overfitting can be there in decision trees if over-complex trees are created by decision-tree learners which can also effect the data generalization. DTs stability can be an issue when they become unstable because of small changes in data which can come up with a completely different tree. An optimal decision tree learning in NP-complete problem under optimality and even for simple concepts. Since DTs do not express them easily so it is hard for them to learn some concepts such as parity, XOR or multiplexer problems. They also create biased trees in some cases where classes dominate. There are also some strategies adopted to mitigate these drawbacks of DTs which can be studied from [6].

Rule-based Classifiers

A classification rule is a procedure where the separable members of population set are assigned to different classes [140]. If every element is assigned to the class it really belongs then it is known as perfect test while if some errors appear then it is imperfect test so statistical analysis is performed for analyzing the classification. *Binary classifications* is a special type of classification where the false positives and false negatives are the elements that are not correctly classified [6].

In general, rule-based classifiers are related to DTs. In the context of textual data, most likely word patterns related to different classes are determined then set of rules are modeled with the data space in rule-based classifiers. Word pattern or a condition on the considered feature-set is on the left-hand side of a rule while a class label on its right-hand side. These rules are used for classification purposes. The rule set models are produced from the training data. For a given test instance, the set of rules are determined for which test instance meets the condition on the left-hand side of the rule then a class label is predicted as a function of the class labels of the rules which are satisfied by the test instance. In other words, classification rule is a function that assigns a class to each member of a dataset consisting x and y in couples where x is the element of the population and y is the class it belongs to. In binary classification, label y can take only two values. Lets say a classification rule or classifier is a function h that can be evaluated for any possible value of x, specifically, provided the data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, h(x) will result a similar classification $\hat{y} = h(x)$, as close as possible to the true group label y. The true labels y_i can be known but will not necessarily match their approximations $\hat{y_i} = h(x_i)$ [6].

Usually, the left-hand side of the rule is a boolean condition expressed in Disjunctive Normal Form (DNF). However, mostly the condition on the left-hand side is much simpler representing a set of terms which all should must present in the document in order to satisfy the condition. It is very rare to use absence of terms in rules because these rules are not so informative for sparse nature of textual data. Furthermore, in most of the cases, the set intersection of conditions on term presence is used while it is very rare to use the union of such conditions in a single rule. It is due to the fact that these rules can be divided into two separate rules where each rule is itself informative by its own. For example, the rule $Mammal \cup Reptile => Animal$ can be replaced by two separate rules i.e., $Mammal => Animal$ and $Reptile => Animal$ with no loss of information. It is more useful to determine confidence of each of these two rules calculated separately. On the other side, the rule $Mammal \cap Reptile => Animal$ is much more informative than its separate rules. Therefore, expressing rules as a simple conjunction of conditions on terms presence is more practical in sparse datasets such as textual collections.

Decision tree framework works in a hierarchical way of partitioning the data space while overlapping in the decision space is allowed in rule-based classifiers so that such a rule set is created that could cover all the points in the decision space by using at least one rule. It is usually accomplished by generating a set of targeted rules related to different classes then one default catch-all rule with the aim to cover all remaining instances.

Support and *Confidence* are two prominent conditions among a number of criteria to generate the rules from training data. These are common conditions to all rule-based pattern classifiers where *Support* quantities the absolute number of instances in the training data set relating to the rule. For example, a corpus consisting 100,000 documents, a rule is more important if its both sides (left and right) are satisfied by 50,000 documents than a rule which is satisfied by 50 documents. This quantifies the statistical volume associated with the rule however, it does not contribute in strength of the rule. The second condition i.e., *Confidence* quantifies the conditional probability that both sides of the rule should be satisfied and this measure highly contributes to show the strength of a rule. There are also other possible measures but *Support* and *Confidence* are widely used in data mining and machine learning areas for textual and non-textual content.

A rule-based classifier for textual data is proposed in [13] based on the technique using iterative methodology first proposed in [138] for generating rules. In the adopted method, the single best rule defined in terms of the confidence, is determined to any particular class in the training data. The best rule and its corresponding instances are eliminated from the training dataset and this process continues until no more robust rules exist in the training data and predictive value is obtained fully.

- **Advantages:** Rule-based classifiers are very expressive as DTs, easy to generate and interpret. They can quickly classify new instances and their performance is comparable to decision trees. They are able to easily handle missing values and numeric attributes. Rules are very accurate if written by experts. Also, classification criteria can be easily managed with the small number of rules. Rule-based classifiers have an advantage over DT that they are not limited to hierarchical partitioning of the feature space but also permit overlapping and inconsistencies among different rules. So, it is relatively easy to modify the rule set for new training examples related to a new class or new section of the feature space. Therefore, rule-based techniques are more frequently used due to their ease of maintenance and interpretability in many practical scenarios.
- **Disadvantages:** In some cases, rules conflict each other and become slower in the presence of large and noisy data collections. It becomes difficult to maintain the rules when they increase in number. With the change of target domain, rules need to be reconstructed. They provide low coverage due to wide variety of expressions.

Genetic algorithms

Genetic algorithms (GAs) are heuristic optimization methods whose mechanisms are similar to biological evolution [93]. Solutions of GA are known as individuals or chromosomes where initial population is generated randomly then a loop is executed for selection and variation function and it ends until a specified terminating criteria reaches. Each execution of loop is called generation. Average quality of the population is enhanced by selection operator that provides high quality individuals with high probability to be copied to next generation. Fitness function determines the quality of an individual. *Genetic Operators* are used by GA as crossover and mutation operators to produce the offspring of the existing population. Parents have been selected for evolution to the next generation before applying genetic operators. Then next generation is produced by the crossover and mutation algorithm. A user can change the probability of applying crossover and mutation operators. GA requires an *end condition* for ending the generation procedure. It stops if there is not enough improvement in two or more consecutive generations. Also, in some cases, time limit is used to end the process.

GAs utilized the mechanisms of genetic evolution and survival of the fittest in natural selection in opinion mining, web mining, feature extraction, knowledge discovery, classification and different text mining techniques in order to optimize the solution. The use of GA has been proven to be beneficial in text mining as it is used in text summarization [23], in extracting details from text resumes [26] and in E-mail classification [73]. There is also a vast scope of research in bioinformatics to dig out the text from large data collections where manual processing of text is not feasible.

- **Advantages:** GA is a very easy to understand method and it does not require mathematical knowledge. Multiple solutions are used by a GA to solve problems. Problems such as multi-dimensional, non-differential, non-continuous and even non-parametric can be solved by the genetic algorithm as it is not dependent upon the error surface. Solution structure and solution parameter problems at a time can be solved by structural genetic algorithm by means of GA. GAs are able to solve every optimization problem which can be described with the chromosome encoding. They can be easily shifted to existing simulations and models. The prominent advantage of using genetic algorithm in text mining is its capability to perform global search and its less time complexity as it rely on greedy approach [123].

- **Disadvantages:** GAs can not solve some optimization problems also known as variant problems because of poorly known fitness functions that produce bad chromosome blocks in spite of the fact that cross-over is blocked by only high quality chromosome. When populations have a lot of subjects then GA does not assure to find a global optimum. It also does not assure constant optimization response time which unfortunately limits the use of GAs in real time applications. GA real time applications in controls are limited due to random solutions and convergence. Improvement of entire population does not vote the same for an individual of this population. So, using GAs for on-line controls in real systems is not sensible without testing them first on simulation model.

Probabilistic and Bayesian classifiers:

Probabilistic classifiers are also known as generative classifiers where an implicit mixture model is used for generation of the underlying documents. Each class is as a component of the mixture and each component is a generative model providing the probability of sampling a particular term for that component or class [6]. The naive Bayes is the simplest and most commonly used generative classifiers. It uses a probabilistic model with independence assumptions about the distribution of different terms to model the distribution of the documents in each class. There are two commonly used models for naive Bayes classification which compute the posterior probability of a class based on the distribution of the document words by considering "bag of words" assumption. Words position in the document is not considered in these models. The major difference between these two models is the matter of considering (or not considering) word frequencies and the corresponding approach for sampling probability space. Here is a brief description of both models (detail study can be found in [6]):

- Multivariate Bernoulli Model: Words frequencies are not considered in this model and the document is modeled by considering the features such as the presence or absence of words in a text document. Two possible values i.e., presence or absence of a term in the text show the binary nature of the word features which are modeled and therefore, multivariate Bernoulli model is a model of documents in each class.
- Multinomial Model: In this model, a document is represented with bag of words (BOW) scheme and frequencies of document terms are calculated. Then, documents in each class are modeled as samples drawn from a multinomial word distribution. Consequently, the conditional probability of a document given a class is simply the product of the probability of each observed word in the respective class.

The posterior probability of the class for an underlying document is computed by the component class models (i.e., generative models for documents in each class) in conjunction with the Bayes rule then the class having highest posterior probability is assigned to the document.

- **Advantages:** Bayesian Classifiers are easy to implement. They also produce good results in most of the cases.

- **Disadvantages:** One limitation is that usually assumption of class conditional independence does not hold in Bayesian Classifiers. Also, Naive Bayesian Classifiers can not model the dependencies.

Linear Classifiers

In Linear classifiers, the output of the linear predictor is defined to be $p = \bar{A}.\bar{X} + b$, where $\bar{X} = (x_1 \dots x_n)$ is the normalized document word frequency vector, $\bar{A} = (a_1 \dots a_n)$ is a vector of linear coefficients with the same dimensionality as the feature space and b is a scalar. In categorical class labels, the predictor p acts as a separating hyperplane between different classes.

The **generalized linear model** (GLM) is a statistical linear model. It is flexible generalization of a common linear regression for relating responses to those variables that have error distribution models as special cases. It generalizes linear regression by allowing the linear model to be related to the response variable through a link function while by allowing the magnitude of the variance of each measurement to be a function of its predicted value [6]. In addition to linear regression models for continuous dependent variables, models for counts, rates and proportions, binary, ordinal and multinomial variables can be considered as GLMs. The GLM approach is appealing as it presents a general theoretical framework for several commonly encountered statistical models. It also simplify the implementation of these different models in statistical software, since essentially the same algorithm can be used for inference, estimation and determining model acceptability for all GLMs [71].

Among linear classifiers, **Support Vector Machines** (SVM) are such classifiers which aim to classify documents into user specified categories based on the content. They find 'good' linear separators among different classes. In other words, SVMs aim to determine separators in the search space which can best separate the different classes [6]. They were first proposed for numerical data. Since, SVM method examines the appropriate combination of features and tries to find the optimum direction of discrimination in the feature space so it is quite robust to high dimensionality. For this same reason, it is very suitable to use SVM method for text classification due to the sparse and high-dimensional nature of text where few features are irrelevant but they have tendency to correlate with one another and are arranged into linearly separable categories. In the SVM algorithm, text documents are represented by vectors where the number of distinct keywords are the dimension. With the increasing size of document, dimension which is utilized by text classification enormously increases resulting the high computational cost. In order to reduce dimensionality, there are strategies such as feature extraction algorithms and computational complexity can be substantially reduced by building SVM model from features extracted from the training data set. There is not only to use linear function for SVM classifier, rather nonlinear decision surface in the feature space can also be constructed by SVM but linear SVM are typically used more often due to their simplicity and ease of interpretability. SVM classifiers are used in textual context by various approaches in literature. For example, [49] used this method to email data to classify it as spam or non-spam data. It was described that the SVM method shows much more robust performance than many other approaches such as boosting DTs, the rule-based RIPPER method and the Rocchio approach. Being flexible, SVM approach can be easily combined with interactive user-feedback approaches [107] and also has been successfully used [52] in the scenario of a hierarchical organization of the classes as often happens in web data where different classifier is built at different positions of the hierarchy. The SVM classifier also worked well in large scale contexts where a small amount of labeled data and a large volume of unlabeled data is available [119]. The adopted approach is a semi-supervised approach as it uses unlabeled data in the classification process. Furthermore, SVM have shown good performance to work on real world applications such as text classification, recognizing hand-written characters, image classification, bioinformatics and analyzing bio sequences.

Linear classifiers are closely related to many feature transformation methods (e.g., Fisher discriminant) that try to use these directions to transform the feature space and then other classifiers are used on this transformed feature space [6]. So, linear classifiers are intimately related to linear feature transformation methods as well. A more direct and traditional statistical approach for text classification is Regression modeling (e.g., the least square method).

Regression-based Classifiers are used in those cases where a numerical variable is generated as a target variable instead of categorical one. The method of regression modeling is commonly used to learn the relationships among real-valued attributes. These methods are typically designed for real-valued attributes as opposed to binary attributes. However, it is not an obstacle to use it in classification as the binary value may be dealt as a special case of a real value. Some regression models (e.g., logistic regression) can also naturally model discrete response variables. Linear Least Squares Fit (LLSF) method [144] is an early application of regression to text classification and has been shown very robust by comparing with variety of other methods in the literature. *Logistic regression classifier* is more natural way of modeling the classification problem with regression as it differs from the LLSF method in that the objective function to be optimized is the likelihood function. Since the decision is determined by a linear function of the features therefore, logistic regression is also a linear classifier. For classification problem, some sets of words can be more prominent than others in a domain knowledge for some cases such as for classification of a target category, some domain-words or Knowledge Words may be more important. So, such domain knowledge can be encoded into logistic regression model in the form of prior on the model parameters and Bayesian estimation of model parameters is then used. Regression classifiers much resemble with SVM model for classification. As, being linear classifiers, all LLSF, Logistic Regression and SVM work similarly at conceptual level while the difference among them exists in optimization formulation and implementation details. Like SVM classifiers, a costly optimization process is required also for regression classifier such as an expensive matrix computations in the form of a singular value decomposition process is required by fitting LLSF [6].

Neural networks classifier is another form of linear classifiers where the function computed by a set of neurons is essentially linear. In supervised learning, a set of example pairs (x,y) are provided where xϵX, yϵY with the goal to get a function f in the allowed class of functions that matches the examples [93]. In other words, one has to infer the mapping implied by the data. The cost functions is related to the mismatch between inferred mapping and the data and it contains implicitly prior knowledge about the problem domain. Supervised learning performs the tasks of classification/-pattern recognition and regression also known as function approximation and it is also applicable to sequential data. Neuron is the basic unit in a neural network and each unit is responsible to receive a set of inputs denoted by the vector $\bar{X}_i$ representing the correspondence to the term frequencies in the i^{th} document. A neuron is also associated with a set of weights used to compute a function of its inputs. If a vector $\bar{X}_i$ is made-up from a lexicon of d words then the weight vector also contains d elements. For a binary classification, all labels are drawn from $\{+1, -1\}$. If Y_i is assumed to be the label of $\bar{X}_i$ then the sign of the predicted function p_i plays its role to produce the class label. This is specifically feasible for domains such as text where values of all features are small non-negative values of relatively similar magnitude. Various studies describe many implementations of neural network approaches for text data [45, 95, 115, 141]. This description leads to two assumption where one is a network of perceptions i.e., Multilayer perception (MLP) which is a simple feed forward neural network. Here, neurons are placed in layers with outputs always flowing toward the output layer. If only one layer exists then it is called perception but if there exist multiple layers then it is known as MLP. The second assumption is back propagation algorithm which is a learning technique that adjust weights in neural network by propagating weight changes backward from the sink to the source nodes. Thus, perceptron (or single layer network) which is the simplest form of neural network are specifically designed for linear separation and it works well for the text. However, for such a case where question arises that how to use neural network, as a linear separator is not seemed to be able to finely separate classes from one another then there can be use of multiple layers of neurons for inducing such non-linear classification boundaries [6]. In other words, it is also possible to generalize the method by using multiple layers of neurons for non-linear separation which induce multiple piece-wise linear boundaries that approximate particular class by enclosing relevant regions. In this scenario, the outcome of neurons from earlier layers is provided to the neurons of later layers. Errors need to be back-propagated over various layers that makes the training process complex in such networks. It is observed in textual context [115, 141] that linear classifiers shows comparable results to non-linear data and there is relatively small improvements of non-linear classification approaches. This finding suggests that significantly better classification is not achieved by additional complexity of designing more non-linear models.

Artificial neural networks are applied in many tasks e.g., in regression analysis, function approximation, time series prediction and modeling, classification, recognition, novelty detection and sequential decision making. They are also applied for the tasks of data processing, filtering, clustering, blind source separation and compression.

- **Advantages:** Neural networks are not biased in analysis as they do not make assumption about data distribution. They use the data with at least one middle layer to create an internal representation of relations among variables. They are best for discovering nonlinear relationships for example to deal with time series data which is dynamic in nature. They also show good performance to deal with missing and incomplete data and when new input data arrives then they adapt their weights showing their adaptive nature.

- **Disadvantages:** Artificial neural networks are not able to calculate estimation or prediction errors. Being ‘black boxes’, they are not able to figure out the estimation of relations in hidden layers.

Proximity-based Classifiers

These classifiers use distance-based measures for classification purpose [6] where dot product or the cosine metric similarity measures determine the closeness of the documents belonging to the same class. For a given test instance, there are two methods for classification:

- In the training data, K-nearest neighbor to the test instance is determined. Class label is the most abundant class from these k-neighbors. Based on the underlying corpus, researchers ususally set the value of k ranges between 20 and 40.
- In this method, training data aggregation is performed during preprocessing step in order to generate document clusters/groups related to the same class. From each group, a representative meta-document is generated against which the k-nearest neighbor approach is applied.

WHIRL approach [40] utilized nearest neighbor classification in textual data where it performs soft similarity joins based on the attributes of text. In soft similarity, two records may not be exactly the same on joint attribute but the similarity notion is used. It is found that by using relevant text documents as the joined attributes, any method of similarity join can also be assumed as a nearest neighbor classifier [40]. Detailed study can be found in [6]. For each document, a ranked list of categories can be generated using the nearest neighbour classifier. If multiple categories seem fit for a document then these categories are revealed for the document until a threshold method becomes available.

Linked and Web data Classification

A massive information is added over the Internet on daily basis with increasing use of web and social media in recent years that leads to a huge volume of document data expressed in terms of linked networks. Web is its simplest form where documents are linked using hyper-links. The data produced by social network channels is noisy due to the tags and different links to address users for connection. Classification task can be beneficial from this linkage information as similar topics are normally connected together [6]. This phenomenon is used in collective classification literature [25] where a subset of network nodes are labeled and the information about linkages among the nodes is used for classifying the rest of the nodes. A content-based network is generally denoted by $G = (N, A, C)$, where N represents the set of nodes, A shows links/edges between the nodes while C represents a set of text documents. The set N contains nodes as its members representing text documents in C. If a node does not contain any content then it shows that the corresponding document is empty. In training data, the subset of nodes in N are labeled. Both the content and structure play important role in classification process. A study on several approaches for classification of web content is presented in [6] where some techniques make use of Bayesian method, random-walk method for dynamic classification and the use of bridges for improving accuracy of classification.

Meta-algorithms for Text classification

Meta-algorithms combine or make a general change in different algorithms with the aim to enhance the performance and accuracy of already existing classification algorithms [6]. *Bagging*, *stacking* and *boosting* are the typical examples of meta-algorithms [51] where some of them combine other classifiers by changing the given distribution of the training data and sometimes algorithms are changed in order to satisfy a particular classification criteria. Here is the brief description of different classes of meta-algorithms:

- **Stacking:** This method has a voting mechanism over a combination of classifiers to perform the classification task with the aim to produce robust results by the resultant combination classifier that tries to overcome errors of different classifiers. The method is also known as classifier ensemble learning or classifier committee construction that focus on combining different classifiers. This method has been frequently used for text categorization by simply using classifier output of weighted combinations represented as scores or ranks for generating the output of ultimate classification.

- **Boosting and Bagging:** Boosting and bagging generally train a classifier by working on different sections of training data in order to generate different models. Also, training models in a *boosting* method are constructed sequentially not independently. Results of these different models are combined together and reported for a given test instance. *Bagging* methods aim to minimize the model overfitting error which arises in the learning process. In this method, samples with replacement are selected from a given collection and classifiers are trained in these samples. Final result is computed by combining the classification results produced by these different samples.

- **Cost-sensitive approaches** is another kind of meta-algorithms for text classification where it is aimed to optimize specific measures of accuracy of the classifiers by considering the fact that the absolute classification accuracy is not the only measure related to classification algorithms. For example, it is more expensive to misclassify examples of one class than another in some scenarios such as in fraud detection and spam filtering where in spam filtering, it can be tolerable to inbox some of spam email but highly undesirable of sending legitimate email to spam incorrectly. In cases of rare class for the interest of capturing rare examples also lead towards the problem of cost-senstive classification. In such scenarios, the ***cost-weighted accuracy*** of classification process is desired to optimize. Details about incorporating cost-sensitivity in classification algorithms can be found in [6]. The use of a cost-sensitive approach is a modification of the objective function of the classification as an optimization problem where standard classification problem tries to optimize accuracy, the cost-sensitive approach aims to optimize a cost-weighted objective function.

Further observations

In case of linear classifiers, the basic conceptual level of all linear classifiers is surprisingly similar although they were developed independently. The difference among them is regarding details of the optimized objective function and the iterative approach adopted for finding the optimum direction of separation. Another observation of linear classifiers is that they all can be accomplished with non-linear variants of their classifiers as well. However, there is consensus that linear versions work very well and extra complication of non-linear classification is not so fruitful except for some special datasets. It may be due to high-dimensional nature of textual domain with highly correlated features and sparse features with small non-negative values. Also, the classifiers with the capability to exploit redundancies and relationships among various features are suitable for high-dimensional nature of correlated text dimensions to enclose the text into different classes. Linear classifiers have shown unprecedented success due to the reason that common applications produced linearly separable over high dimensional domain of text in the form of class structures [6].

In *meta-algorithms* for classification purposes, ensemble learning has to battle with the challenge of appropriate combination of classifiers for a specific scenario which can be significantly vary with the dataset and the scenario. Its counterpart, *boosting* is also a kind of ensemble learning methodology except that here same model is trained on different subsets of the data to create ensemble. The main criticism on *boosting* is the presence of some noisy training records in many datasets and classification model should be resistant to over-training on the data. The possibility of using weighted error-prone examples in successive rounds can lead the classification process to be more vulnerable to overfitting especially in the case of noisy datasets. Here comes the *bagging* methods to overcome the issue of model overfitting error. These methods can be used with any kind of classifier but generally they work in conjunction with decision trees. *Bagging* method faces major criticism of leading to a reduction in accuracy due to the smaller size of each individual training sample. Since, *bagging* method aims to minimize overfitting error so it is applicable only in case of model instability to minor details of the training algorithm such as decision tree model where high sensitivity exist for construction of the higher levels of the tree in a high dimensional feature space e.g., text. In general, there is no frequent use of *bagging* methods in text classification.

Since, various kinds of experimental design such as leave-one-out cross validation (LOOCV), k-fold cross-validation, bootstrap and re-substitution are used to compute classification accuracy so it becomes difficult to compare performance and results of Classifiers in different works. A comparison of some previously described classification techniques is presented in [102]. According to the study, operational profiles of DTs and rule classifiers are similar whereas Decision trees (DTs) and Bayesian Network (BN) generally have different operational profiles. When DTs are very accurate the other is not and when BN performs good then the performance of DTs disappoint. A number of approaches have been proposed for creating ensemble of classifiers, perhaps due to which picture for best approach is not yet clear.

Apart from afore-mentioned classification algorithms, there are also some other important data Mining algorithms briefly described below that can be used for text mining as well:

2.2.2 Clustering

Clustering hails from mathematics, statistics and numerical analysis. In clustering algorithm, data is divided into groups of similar objects where each group is called cluster. Each cluster has homogeneous elements while differing with the elements of other cluster. If data is grouped into fewer clusters then it shows simplification but leads to lossy compression where certain details are lost. It shows many data objects by few clusters and data is modeled by its clusters [93]. Clustering algorithms are classified into two models i.e., *Hierarchical clustering* and *Partitional clustering.* Brief description of both models are as follows:

Hierarchical clustering

It constructs a cluster hierarchy or a clusters tree called dendogram. It provides to explore data on different levels of granularity. *Agglomerative* (bottom-up) and *divisive* (top-down) are two categories of hierarchical clustering where the former one starts with one-point (singleton) clusters and combines two or more top appropriate clusters recursively. *Divisive* clustering makes partitions of a given data set by classifying it into k clusters. Being one of the simplest unsupervised algorithm, K-means works in an easy way. It randomly places initial group centroids in 2d space then each object is assigned to the group that has the nearest centroid. Then, the positions of the centroids are recalculated. If positions of centroids get change then the algorithm works again from assigning each object to group that has the nearest centroid but if positions of centroids did not change then k-means procedure ends. The procedure continues until a specific criterion (k number of clusters) is achieved. There are both merits and de-merits of hierarchical clustering. Its advantages are its embedded flexibility regarding the granularity level and easy handling of any forms of similarity or distance. Consequently, it is applicable to any attribute type. Disadvantages are its ill-defined termination criteria and the fact that most of the hierarchical algorithms do not revisit once constructed (intermediate) clusters in order to improve them.

Partitional clustering

In Partitional clustering, data is divided into several subsets and then different relocation schemes are adopted to reassign points among k clusters by using some greedy heuristics. Relocation algorithms make possible to revisit the clusters after their construction which leads improvement in clusters and high quality clusters can be formed with the availability of appropriate data. In order to partition data, a conceptual point of view is taken that finds the cluster with a specific model with unknown parameters which have to be identified. This type of clustering has advantage of representing concise and interpretable clusters that exhibits inexpensive computation of intra-clusters measure which leads to a global objective function [93].

2.2.3 Non-Negative Matrix Factorization

A group of algorithms is called Non-Negative Matrix Factorization (NMF) in linear algebra and multivariate analysis where a matrix is factorized into two matrices with the characteristic of non-negative elements in all three matrices. The non-negativity makes easy inspection of resulting matrices [6]. NMF can be used in various text mining applications where a document-term matrix is constructed with the weights (such as word frequency) of different terms from a document collection. This matrix is factored into a term-feature and a feature-document matrix where the features are extracted from documents content and feature-document matrix presents data clusters of related documents. Hierarchical NMF is used in an application on small subset of scientific abstracts from PubMed [97]. In another application [24], Enrol email dataset with 65,033 messages and 91,133 terms is grouped into 50 clusters. NMF is also utilized in citations data where Wikipedia articles and scientific journals are clustered based on outbound scientific citations in Wikipedia [96]. It also used to learn topic models by polynomial-time algorithms where the algorithm considers that the topic matrix satisfies a separability condition which is usually found to hold in these settings [14]. NMF also has applications in various other fields like in document clustering, recommender systems and computer vision.

2.2.4 Association rule mining

The task of association rules in text data mining is to discover associations between concepts and to express them as rules in the form $A \implies B$ [support, confidence], where A and B may be a set of concepts or a unique concept [86]. The rule implies that *if A exists in the text then B exists with a certain support and a certain confidence*. According to [66], *confidence* is texts proportion that have $A AND B$ in relation to the number of texts that have only A while *support* is the proportion of texts that have $A\ AND\ B$ in relation to number of all texts in the collection. Discovered rules may have combination of concepts and/or words such as $WORD_1\ AND\ WORD_2\ AND\ CONCEPT_1\ AND\ CONCEPT_2 \implies CONCEPT_3$. Sub-collections of text documents where some words are present can be selected by these kind of rules [100].

Apriori algorithm is a historically significant algorithm and most often used approach for learning association rules. Its name is based on the fact that it uses prior knowledge of frequent itemset properties in the following way. From a given set of itemsets, Apriori algorithm aims to find subsets common to atleast a minimum number N of the itemsets. It is a ‘bottom up’ approach where frequent subsets are extended one item at a time (candidate generation) and then candidates groups are tested against the data. Algorithm ends on unsuccessful extension [6]. In text mining scenario, its usual working in two steps is as it first tries to get all frequent itemsets such as tuples of concepts or terms as itemsets that meet a specified minimum support count which guarantees the occurrence of selected itemsets as frequently as it defines. Secondly, it produces strong association rules from the frequent itemsets which have to satisfy specified minimum support and minimum confidence.

The need to mine frequent patterns exist in many areas such as market basket analysis where frequently bought items together at a supermarket are analyzed from customer shopping records. Frequent itemset mining leads to extract association rules among the itemsets. It makes one able to make statements about how likely are two itemsets to co-occur. In market basket analysis, one can find rules such as ‘customers who buy flour and sugar also would like to buy Eggs’. This rule may prompt a grocery store to place these items close in a shelf.

Apriori is a classic algorithm but it has some inefficiencies or trade-offs which emerged the need of other algorithms. It tries to load up as many candidate set as possible before each scan in candidate generation step which results large numbers of subsets. Also, bottom-up subset exploration finds any maximal subset S only after all $2^{|S|}-1$ of its proper subsets. It also does not scale well for large databases or datasets so with the case of document collections as well. Frequent-pattern growth or FP-growth is another interesting approach for generating the set of frequent itemsets without candidate generation based on divide-and-conquer strategy [66].

2.3 Summarization Models

There is massive information overload on the Internet in the form of news articles, research papers, web pages, blogs and users comments. It is because of day to day writings from journalists, scientists, researchers and web users. These documents generally consist unstructured form of textual content while different users require different aspects of information from these documents. So, there is need of data mining and information retrieval techniques to get required information from these huge data collections and to present it in short and comprehensive form known as *summary*. A summary is same as the abstract of a research paper or the summaries which we were used to write at school level manually, however, if computer automatically performs this task to generated summary then the produced summary is known as *automatic summary*. In other words, Text summarization models aim to select most relevant parts of the text and present a concise and compact version of the text to the user. They reduce and organize unstructured textual document collections. Summarization models for generating automatic summaries fall into various categories such as:

- **Single-document Vs. Multi-document summarization:** Those models are known as single-document summarizers that work over a single document to produce its summary while if a summarization model is developed to generate summary of multiple documents of a collection then it is known as multi-document summarizer. Our focus has been to work on multi-document summarization discussed in this thesis.

- **Keywords Vs. Sentence based text summarization:** Some summarization models capture the key terms across a document or a set of documents and present them to the user as a summary. In this way, *Keywords based summarizer* generates a summary consisting main keywords of the document(s). In *sentence-based summarizer*, most informative sentences are extracted from a document (single-based) or a document collection (multi-document) are included in the resultant summary. Our interest has been to get informative sentences from multiple documents of a collection to generate a summary so our proposed summarization models in this thesis are sentence-based text summarizers.

- **Domain-specific Vs. Generic Text summarization:** *Domain-specific summarizers* are developed for a specific domain such as natural disaster or business related domain. Such systems need a domain vocabulary or dictionary consisting domain's important concepts to rank the information existing in the underlying text. On the other hand, *generic text summarization* presents the broader view that works on any domain and tries to cover most of the topics of document(s). The work discussed in this thesis focuses general-purpose text summarization.

- **Abstractive Vs. Extractive Text summarization:** *Abstractive* text summarizers modify the generated summaries to make them more coherent and cohesive so that they could be more understandable for end users while *extractive* summaries consist the sentences in the same form as they exist in the source documents. The focus of work presented in this thesis has been on extractive summarization models.

2.4 Types of document collections

Since, a mass of data has been collected over the time ranging from simple numerical measurements and textual documents to more complex information e.g., multimedia content, spatial data and hypertext documents. Some types of digital information repositories discussed in [148] are presented as follows:

- **Business records:** Each transaction is recorded in business industry that can be inter-business agreements like stock, purchases, banking or exchanges etc. or intra-business deals like assets and in-house wares. In this way, terabytes of data is stored by using bar codes that stores day-to-day millions of transactions in large departmental stores. With low price of hard disks, storage of massive data has not be an issue but there is need for using the data in effective way so that it could be use in real time or in decision-making purposes.

- **Scientific data:** The data collected in research or scientific organizations has huge volume whether it is from any nuclear laboratory, readings of forest study, research outcomes in a university or weather forecasting measures, it needs to be analyzed. The problem is to deal with old data already collected rather than analyzing newer data.

- **Video and pictures:** Low priced cameras made their use so frequent to make a massive volume of tapes and images. Surveillance cameras are usually recycled so their content is lost therefore, there emerges a need of storing their tapes and even digitize them for future analysis and use.

- **Medical and personal data:** A huge collections of data related to individuals as well as groups are continuously being collected by governments, companies and organizations like hospitals. This data helps them to well manage human resources, better client assistance or understanding market behavior.

- **Satellite findings:** A non-stop steam of data is continuously sent to the surface by countless satellites around the globe. Many satellite pictures and data are stored and sent to other researchers for analysis.

- **Text reports and memos:** The communication among individuals, companies or organizations are massively done through digital form such as email messages. Email messages may extensively exchange text reports and memos that are in textual forms. This textual content is stored digitally for future use and reference.

- **World Wide Web repositories:** The largest repository of data has been built with the emergence of World Wide Web (WWW) that has documents of any format, content and description and hyperlinks connect data to other data making a chain of information. WWW is the most important data repository although it has heterogeneous characteristic, redundancy and inconsistency but it is used regularly for various purposes. News papers publish their news stories and articles on Web which pulls the attention of researchers to mine the news categories coming from different sources. Also, social media content from social networking sites has become an active research area in these days.

Chapter 3

Knowledge representation and extraction

This chapter presents a study on knowledge representation and extraction techniques in data mining. A brief discussion is given in section 3.1. Section 3.2 describes some methods of knowledge representation existing in literature. Section 3.3 presents a study on knowledge extraction while Section 3.4 discusses some well-known knowledge extraction tools.

3.1 Knowledge Representation

Knowledge representation is the branch of Artificial intelligence which is a multidisciplinary field that utilizes theories and techniques from mainly three other areas [120] i.e., Logic, Ontology and Computation. A Logic is a formal language, with precisely defined syntax and semantics, which provides inference. Ontology is the set of concepts (entities) with their relationships in the application domain whereas Computational model differentiate knowledge representation from pure philosophy. Knowledge is ambiguous without logic while in the absence of ontology, entities and symbols could not be defined properly. Computation is also important because in its absence, both logic and ontology can not be implemented in computer programs. Thus, knowledge representation is the application of logic and ontology for generating computable models for a domain. Application domain or domain of interest covers partially or fully any real world or hypothetical system about which the knowledge is represented for computational purposes. A distinction can be made between knowledge level and symbol level in knowledge

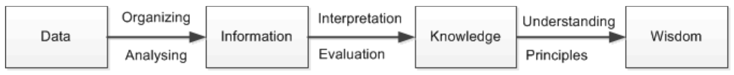


Figure 3.1: Knowledge Progression

representation. Knowledge level is the content of knowledge while symbol level is a formal language representation. Some of the content of knowledge may be lost while translating the knowledge from knowledge level to symbol level. A knowledge-base is required to store the symbol level of knowledge. A knowledge-based system has computational model with domain knowledge represented in the form of symbols.

3.2 Knowledge Representation Methods

There are five major knowledge representation methods i.e., Logic, Semantic Networks (Ontologies), Frames, Production Rules (Association rules) and Classes in Object Oriented paradigm. Out of these methods, Production Rules (Association rules) and Semantic Networks(Ontologies) are focused more intensively while a brief study of other methods is presented in 3.2.3.

3.2.1 Production/Association Rules

Production rules are simply IF-THEN statements while association rules are constrained production rules. The best-known constraints on association rules are minimum thresholds on support and confidence. Support plays role in finding frequent item sets, while confidence is used for rules generation. Following are few examples of the traditional production rules (IF-THEN statements):

- IF the sky is blue THEN It is not raining. $\Rightarrow$ blue(sky) $\rightarrow \neg$(raining)
- IF X is a student and X has publications THEN X will get admission. $\Rightarrow$ student(X) $\wedge$ publications(X) $\rightarrow$ getAdmission(X)

Association rules analysis is a key tool for extraction or discovery of interesting relationships implicitly present in huge data sets or databases. It is an implication expression represented as $X \rightarrow Y$, where $X \cap Y = \phi$. X is the antecedent while Y is known as the consequent. In the well known Market Basket scenario, the objective of Association rules analysis is to study the behavior of customers, in order to improve business related applications such as marketing, customer relationship management [99]. It is also suitable for other application domains e.g., bioinformatics, medical diagnosis, Web mining and text summarization.

TID	**Items**
1	Milk, Diaper, Tomatoes
2	Bread, Milk
3	Bread, Coke, Onions, Tomatoes
4	Bread, Diaper, Tomatoes, Onions
5	Coke, Potatoes, Tomatoes

Table 3.1: transactions at point-of-sale

The table 3.1 is an example of a transactional database where one can be interested in finding if a customer buys *tomatoes and onions* then he/she also buys *bread*. It is formulated as:

$$\{Tomatoes, Onions\} \Rightarrow \{Bread\}$$

Frequent itemset i.e., *Support* tells that how many times it happened when a customer buys *Tomatoes and Onions*, he/she also buys *Bread*? To generate this rule, $Confidence$ indicates that how much one is confident that when a customer will buy *Tomatoes and Onions*, he/she will also buy *Bread*? *Support* and $Confidence$ are formulated as:

$$Support, s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \tag{3.1}$$

$$Confidence, c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \tag{3.2}$$

In the table 3.1, there are Number of transaction (N) = 5 while customer buys 2 times $\{Tomatoes, Onions, Bread\}$. Therefore, the support of the rule is 2/5=0.4. The rule's confidence is obtained by dividing the support for $\{Tomatoes, Onions, Bread\}$ which is 2 by the support count for $\{Tomatoes, Onions\}$ which is also 2. Therefore, the obtained confidence is 2/2=1.

Frequent itemset mining [7] is a widely exploratory data mining technique that focuses on discovering correlations, i.e., itemsets, that frequently occur in the source data. An itemset I of length k, i.e., a k-itemset, is a set of k distinct items. Let T be the document collection in the transactional data format. $\mathcal{D}(I)$ is denoted as the set of transactions supported by I, i.e., $\mathcal{D}(I) = \{tr_{jk} \in T \mid I \subseteq tr_{jk}\}$. The support of an itemset I is the observed frequency of occurrence of I in D, i.e., $sup(I)=\frac{\mathcal{D}(I)}{|T|}$. Since the problem of discovering all itemsets from a transactional dataset is computationally intractable [7], itemset mining is commonly driven by a minimum support threshold. mining process and select the most informative yet not redundant itemsets without the need of a post-pruning step. Frequent itemset mining technique can be used for selecting the most informative yet not redundant itemsets. Its efficiency and effectiveness in discovering succinct transactional data summaries makes it particularly suitable for the application to text summarization.

3.2.2 Ontology analysis

"Ontologies are formal representations of the most peculiar concepts that are related to a specific knowledge domain and their corresponding relationships" [17]. They are made up of individual entities that exist in a domain, the relationship between that entity with other entities and the behavior of the entity. An ontology should have several characteristics such as formality, conceptuality, explicitness and being shared. In order to provide a formal semantics, an ontology is expressed in a knowledge representation language that ensures well-defined way of interpretation and machine-processability. Explicitness provided by an ontology makes it accessible for machines. Ontology construction is associated with a social process of reaching consensus in a community so its being shared with a limited group of people in that domain. Principally, ontology is comprised of concepts, relations and instances where concepts are generic nodes which represent an ontological category in the semantic networks and relations are the arcs while instances are the individual nodes which represent concrete objects.

In the figure-3.2, *Degree* and *Certificate* are the concepts while *PhD* and *DAE* are the instances and *is_a* and *takes* etc. with arrows are the relations.

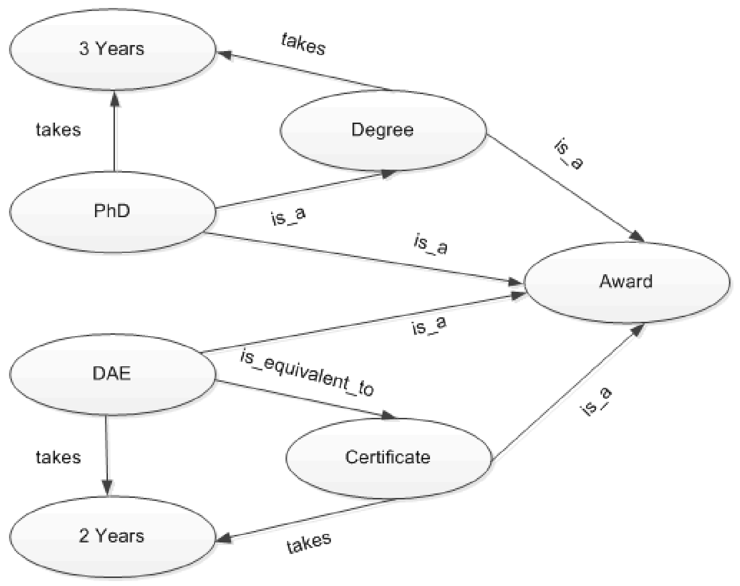
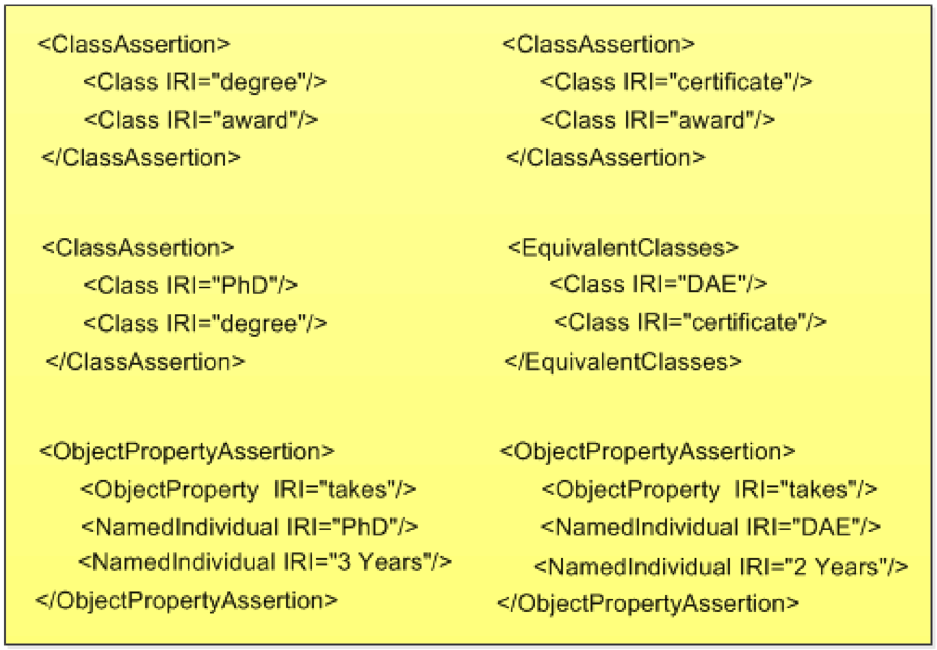


Figure 3.2: Example of an ontology

XML, RDF and OWL are prominent languages to represent ontologies where XML is a language for describing documents. RDF and RDFS are languages for describing the organization of Web resources. OWL (and Description logics) are knowledge representation languages useful, well-founded and efficient enough for being the basis of knowledge representation languages for representing ontologies. The use of the OWL Web Ontology is made in order to define and instantiate Web ontologies. OWL is a semantic Web language where things, groups of things and associations between things are represented as rich and complex knowledge. Consistency of the knowledge expressed in OWL is verified or implicit knowledge is made explicit by computer programs which uses computational logic-based language such as OWL. OWL is not a programming language, it is declarative which uses a logical way to describe a state of affairs. Figure-3.3 shows a simple example of OWL in XML syntax describing a state of affairs existing in Figure-3.2 in a logical way.

Ambiguous entities are matched with semantic entities of knowledge-base to get corresponding disambiguated entity. Wikipedia or DBpedia due to their wide coverage of entities have been used for named entity recognition and disambiguation in most of the recently proposed approaches in which some use BOW (Bag of Words) while some use semantic relatedness among concepts. Yago ontology is the state of the art knowledge-base where semantic entities can be retrieved as it consist of meaningful entities. It is a



```
<ClassAssertion>
    <Class IRI="degree"/>
    <Class IRI="award"/>
</ClassAssertion>

<ClassAssertion>
    <Class IRI="PhD"/>
    <Class IRI="degree"/>
</ClassAssertion>

<ObjectPropertyAssertion>
    <ObjectProperty  IRI="takes"/>
    <NamedIndividual IRI="PhD"/>
    <NamedIndividual IRI="3 Years"/>
</ObjectPropertyAssertion>

<ClassAssertion>
    <Class IRI="certificate"/>
    <Class IRI="award"/>
</ClassAssertion>

<EquivalentClasses>
    <Class IRI="DAE"/>
    <Class IRI="certificate"/>
</EquivalentClasses>

<ObjectPropertyAssertion>
    <ObjectProperty  IRI="takes"/>
    <NamedIndividual IRI="DAE"/>
    <NamedIndividual IRI="2 Years"/>
</ObjectPropertyAssertion>
```

Figure 3.3: Example of Web Ontology Language (OWL)

huge ontology of good quantity and quality metrics having millions of entities and millions of relations. AIDA is an online tool for named entity recognition and disambiguation based on YAGO knowledge-base where the subgraph of mentioned entities in the text builds in the method and then the best mention-entity mapping is approximated with a greedy algorithm. We exploited the knowledge of obtained disambiguated named entities in text summarization of multiple documents where it weighs the recognized named entities and ranks the document sentences accordingly then highly ranked and dissimilar sentences are selected to include in the resultant summary of underlying document collection.

3.2.3 Other knowledge representation methods

To represent conceptual models with sparse representation, there are different languages with different characteristics in terms of computational complexity, ease of use and their expressiveness. Object-oriented knowledge representation languages such as frame and UML, and vocabularies defined natural language are some of KR languages used for ontologies. The ontologies which are purely hand-crafted and simple tree-like structures fall under the category of *Vocabularies* but this category exhibits limitation as it provides single inheritance and also maintenance and ill-formed defined semantics [125].

A *frame-based system* is able to represent broader structure because it is based on classes which represent the concepts of ontology by representing collections of instances [125]. In knowledge representation world, the frame-based systems have been widely used specially in natural language processing applications. Although frames provide huge set of language constructs but also offer limited constraints like how they define classes or how they should be combined.

Description language is the alternative of frames, having limited number of constructs but it solves the problem of constructs to be defined and combined by its logic reasoning system. Classification taxonomies are automatically derived by the description logics which describe knowledge in terms of concepts and their relations [125]. The problem with description languages is their increasing computational complexity of reasoning with their increasing span of expressiveness. Frames and description logics are not so far; description logics are reformulation of frames. Reasoning services of a description language can be combined with the simplicity of frames resulting a unified language defined using RDF [69].

3.3 Knowledge Extraction

Knowledge extraction concerns identifying specific informative pieces of data in natural-language documents. It generates knowledge from structured (relational databases, XML) and unstructured textual (text, images, audio/video) sources. *Named entity recognition* is a type of knowledge extraction where it is aimed to identify specific kinds of objects e.g., names of persons, organizations and locations etc. With recognizing named entities, it is also important to extract particular kinds of relations between entities for some applications.

There are different ways to build systems for information extraction. In one way, patterns are encoded to manually develop information extraction rules using regular expression to recognize entities and their relationships among them but such systems are rarely robust and also if very difficult and tedious to manually developing these patterns. In order to develop robust information extraction systems, supervised machine learning approaches became popular which are trained on human annotated corpora.

Social media content is also a knowledge extraction platform where sentiments, opinion mining and analysis are involved. Semantic knowledge in terms of recognizing named entities existing in social media content can also be located and can be used in applications such as document analysis, text summarization of related news articles etc.

3.4 Knowledge Extraction Tools

Knowledge extraction (KR) has been focused from many decades by software community but with the emergence of linked data such as DBpedia, it has been exclusively associated with semantic web as well. Various tools have been used ranging from learning basic semantic structures such as recognizing named entities, entity relations, topics of the underlying documents to the complex analysis of knowledge extraction tasks such as event recognition, event dependency detection etc [57].

There are recently developed tools for deeper KE by combining useful smart heuristics as well as existing knowledge from Linked Open Data. [57] summarizes some of these tools in a comprehensive way as follows:

- **AIDA** is a framework and online tool for named entity recognition and disambiguation. It consults YAGO2 ontology with the aim to map mentions of source document text with disambiguated entities of YAGO2 knowledge-base [126]. It considers prior probability in terms of Wikipedia links, contextual similarity and coherence among the recognized entities. It is available as a online demo [92] and we used this service for obtaining recognized and disambiguated named entities to incorporate them for the task of multi-document text summarization (see chapter-5 and chapter-6).

- **DBpedia Spotlight** tool is developed for automatically annotating mentions of DBpedia resources in the text. It is available as a demo web application, as a REST service or downloadable [89].

- **AlchemyAPI** aims to extract named entities, their relationships, topics and sense tagging by analyzing web or text-based content using parsing and machine learning. It does not provide a direct RDF encoding. AlchemyAPI is available as online demo service and for mobile SDKs as well [1].

- **CiceroLite** also called Extractiv, recognizes named entities for English, Arabic, Chinese and some European-language texts. It also performs relation extraction, semantic role labeling and sense tagging. It is available as a online demo and as a REST application [3].

- **NERD** is merger of knowledge extraction tools (AlchemyAPI, DBpedia Spotlight, Extractiv, Lupedia, OpenCalais, Saplo, SemiTags, Wikimeta, Yahoo! Content Analysis, and Zemanta) focusing sense tagging and named entity recognition and resolution. It is available as a demo web application and as a web service [111].

- **Wikimeta** performs sense tagging and recognizes named entities and disambiguate them. Text data is linked to concepts of the Linked Open Data network through different sources like Geonames, DBpedia, Wikipedia or CIA World Factbook or the web on unavailability of any resource. It is available as a demo web service and as a REST application [53].

- **FOX** focuses sense tagging, named entity recognition and resolution, term and relation extraction. It is available as a demo web service [16].

- **FRED** automatically produces RDF/OWL ontologies and linked data from text based on deep semantic parsing, discourse representation theory, linguistic frames and ontology design patterns. Semiosearch Wikifier produces NER results. It is available as a demo web service, as a REST service or downloadable [104].

- **Zemanta** has interaction capabilities where it matches text with publicly available content and produces it in the creation tool as it is being written. It also performs content linking and recognizes named entities and disambiguate them. It is available as a demo web service and provides an API for content management systems [5].

- **PoolParty Knowledge Discoverer** performs text mining and recognizes named entities based on knowledge models, thesauri and linked data. It depends upon a reference knowledge-base derived from some controlled vocabularies such as thesaurus. Images, tags, content and categories are recommended automatically when controlled vocabularies are used as a base knowledge model. It is available as a demo web application [4].

- **Open Calais** is a knowledge extraction tool used for named entity recognition with sense tags, facts and events. It is available as a web demo application and as a web service [110]. Through web application, it is used for homogeneity with the other tools.

- **ReVerb** automatically identifies and extracts binary relationships from English sentences. It runs on a model trained out of the big dataset of Open Information Extraction web triples. It takes raw text as input and produces triples (argument1, relation phrase, argument2) as output. It does not work on bulk text. It is available as demo web application and also can be downloaded [12].

- **Apache Stanbol** semi-automatically enhances unstructured text with semantic annotations to be able to link documents with related entities and topics. Current enhancers include RDF encoding of results from multilingual named entity recognition and resolution, sense tagging with reference to DBpedia and GeoNames and related images etc. It is available as demo web service, as a REST service or downloadable [2].

- **Semiosearch Wikifier** integrates various components i.e., a named entity recognizer (currently Alchemy), a semiotically informed index of Wikipedia pages, as well as matching and heuristic strategies for resolving arbitrary named entities or terms on DBpedia entities. It is available as a demo web service [41].

Chapter 4

ItemSum

This chapter presents the ItemSum (Itemset-based Summarizer) multi-document summarizer. Introduction is given in section 4.1. Section 4.2 describes the main steps of the ItemSum method. Section 4.3 presents the experiments performed to validate the proposed approach.

4.1 Introduction

In last years, the increasing availability of textual documents in the electronic form has prompted the need of efficient and effective data mining approaches suitable for textual data analysis. It has been aimed to analysis these huge data collections to get desired information and to present it in a concise and short form known as summary. A summary is a succinct and informative description of a data collection. In the context of multi-document summarization, the selection of the most relevant and not redundant sentences belonging to a collection of textual documents is definitely a challenging task.

A number of different approaches have been proposed to generate summary by selecting the most relevant sentences (e.g., [105, 128, 136]). They commonly evaluate sentences according to cluster-based or graph-based models. For instance, the approach recently proposed in [136] exploits an incremental hierarchical clustering algorithm with the two-fold aim at identifying groups of sentences that share the same content and updating summaries over time. Differently, [105] proposed to represent correlations among sentences by means of a graph-based model. Most relevant sentences are selected according to the eigenvector centrality computed by means of the well-known PageRank algorithm [29]. A parallel research effort has been devoted to formalizing the summarization task as a maximum coverage problem with Knapsack constraints based on sentence relevance within each document [128]. However, previous approaches typically focus on single word significance while do not effectively capture correlations among multiple words at the same time.

Frequent itemset mining is a well-established data mining technique to discover correlations among data. Although it has been widely used in transactional data analysis, to the best of our knowledge, its exploitation in document summarization has never been investigated so far. In recent years, a number of approaches addressed the discovery and selection of the most informative yet non-redundant set of frequent itemsets mined from transactional data (e.g., [72, 130]). Some of them compared the observed frequency (i.e., the support) of each itemset against some null hypotheses (e.g., its expected frequency) to evaluate its interestingness. Others considered previously selected patterns in itemset evaluation to reduce model redundancy. Among them, [130] effectively exploited an heuristics to solve the maximum entropy model that allows evaluating on-the-fly the significance of an itemset during the itemset mining process.

The presented ItemSum multi-document summarizer in this chapter provides two main contributions: (i) the usage of an itemset-based model to represent the most relevant and not redundant correlations among document terms, and (ii) the selection of the minimal set of representative sentences that best covers the itemset-based model. From a transactional representation of the document collection, an highly informative and not redundant itemset-based model is extracted to represent significant higher order correlations among document terms. To address this issue, the algorithm first proposed in [87] in the context of transactional data is adopted. Since it pushes the itemset selection into the mining process, it is particularly suitable for being applied in text summarization. To better discriminate among single word occurrences within each document, ItemSum combines the usage of the itemset-based model with a sentence relevance score, computed from the bag-of-word sentence representation and based on the well-founded tf-idf statistics [74]. The problem of selecting the minimal set of sentences that best covers the itemset-based model is formalized as a set covering problem. To solve the problem efficiently and effectively, a greedy approach is adopted.

Experiments are conducted on a collection of real news articles as well as on the DUC'04 document collection by means of ROUGE toolkit which shows the effectiveness of the proposed approach. Experimental results performed on the DUC'04 document collection show that the proposed approach

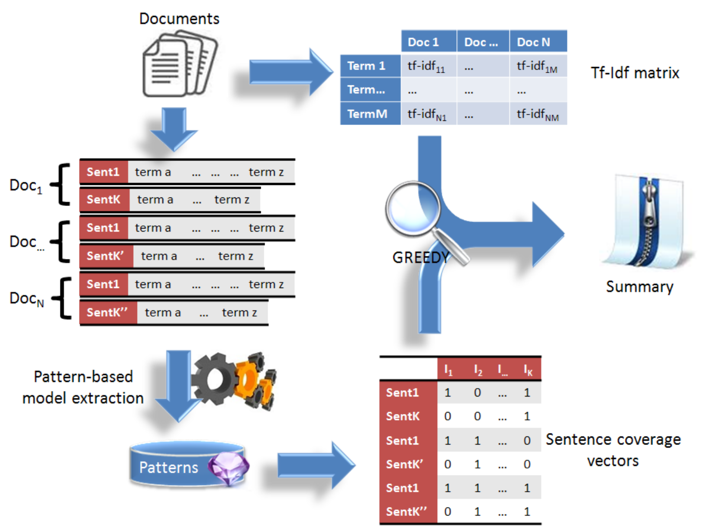


Figure 4.1: The ItemSum method

achieves better performance than a large set of competitors.

4.2 Method

ItemSum is a novel summarizer that selects the most representative sentences based on both their coverage of an itemset-based model and their single-word statistical relevance. To this aim, it relies on a two-way data representation, which is formally stated in the following section whereas Figure 4.1 shows the main steps behind the proposed approach, which will be thoroughly described in the following.

4.2.1 Document representation

Two different document/sentence representations are exploited by ItemSum: (i) the traditional bag-of-word (BOW) sentence representation to evaluate the sentence relevance score and (ii) the transactional data format to compute the itemset-based model. The raw document content is first preprocessed to make it suitable for the data mining and knowledge discovery process. To avoid noisy information stopwords, numbers, and website URLs are removed, while the Wordnet stemming algorithm [27] is applied to reduce document words to their base or root form (i.e., the stem). Let D=$\{d_1, \ldots, d_n\}$ be a document collection, where each document d_k is composed of a set sentences S_k=$\{s_{1k}, \ldots, s_{zk}\}$. Documents are composed of a sequence of sentences, each one composed of a set of words. The BOW representation of the j-th sentence s_{jk} belonging to the k-th document d_k of the collection D is the set of all word stems (i.e., terms) occurring in s_{jk}.

Consider now the set tr_{jk}=$\{w_1, \ldots, w_l\}$ where $tr_{jk} \subseteq s_{jk}$ and $w_q \neq w_r$ $\forall\ q \neq r$. It includes the subset of distinct terms occurring in the sentence s_{jk}. To tailor document sentences to the transactional data format, we consider each document sentence as a transaction whose items are distinct terms taken from its BOW representation, i.e., tr_{jk} is the transaction that corresponds to the document sentence s_{jk}. A transactional representation T of the document collection D is the union of all transactions tr_{jk} corresponding to each sentence s_{jk} belonging to any document $d_k \in D$.

The document collection is associated with the statistical measure of the term frequency-inverse document frequency (tf-idf) that evaluates the relevance of a single word in the whole collection. A more detailed description of the tf-idf statistic follows. The whole document content could be represented in a matrix form TC, in which each row represents a distinct term of the document collection while each column corresponds to a document. Each element tc_{ik} of the matrix TC is the tf-idf value associated with a term w_i in the document d_k belonging to the whole collection D. It is computed as follows:

$$tc_{ik} = \frac{n_{ik}}{\sum_{r \in \{q\ :\ w_q \in d_k\}} n_{rk}} \cdot \log \frac{|D|}{|\{d_k \in D\ :\ w_i \in d_k\}|} \tag{4.1}$$

where n_{ik} is the number of occurrences of i-th term w_i in the k-th document d_k, D is the collection of documents, $\sum_{r \in \{q\ :\ w_q \in d_k\}} n_{rk}$ is the sum of the number of occurrences of all terms in the k-th document d_k, and $\log \frac{|D|}{|\{d_k \in D\ :\ w_i \in d_k\}|}$ represents the inverse document frequency of term w_i.

4.2.2 Itemset-based model generation

Frequent itemset mining [7] is a widely exploratory data mining technique that focuses on discovering correlations, i.e., itemsets, that frequently occur in the source data. An itemset I of length k, i.e., a k-itemset, is a set of k distinct items. Let T be the document collection in the transactional data format. We denote as $\mathcal{D}(I)$ the set of transactions supported by I, i.e., $\mathcal{D}(I) = \{tr_{jk} \in T \mid I \subseteq tr_{jk}\}$. The support of an itemset I is the observed frequency of occurrence of I in D, i.e., $sup(I)=\frac{\mathcal{D}(I)}{|T|}$. Since the problem of discovering all itemsets from a transactional dataset is computationally intractable [7], itemset mining is commonly driven by a minimum support threshold.

Given a minimum support threshold min_sup and a model size ms, ItemSum generates an itemset-based model that includes the most informative yet non-redundant set of ms frequent itemsets discovered from the document collection T. Among the large set of previously proposed approaches focused on succinctly representing transactional data by means of itemsets [130], we adopt an algorithm recently proposed in [87]. Unlike previous approaches, it exploits an entropy-based heuristics to drive the mining process and select the most informative yet not redundant itemsets without the need of a postpruning step. Its efficiency and effectiveness in discovering succinct transactional data summaries makes it particularly suitable for the application to text summarization.

4.2.3 Sentence evaluation and selection

ItemSum exploits the itemset-based model to evaluate and select most relevant sentences to include in the summary. Sentence evaluation and selection steps consider (i) a sentence relevance score that combines the tf-idf statistics [74] associated with each sentence term, and (ii) the sentence coverage of the generated itemset-based model (see Section 4.2.2). In the following we formalize both sentence relevance score and sentence model coverage.

4.2.3.1 Sentence relevance score

The relevance score of a sentence is computed from the BOW sentence representation. It is defined as the sum of the tf-idf values [74] of each distinct term belonging to a generic sentence s_{jk} in the document collection.

$$SR(s_{jk}) = \frac{\sum_{i \ | \ w_i \in t_{jk}} tc_{ik}}{|t_{jk}|} \tag{4.2}$$

where t_{jk} is the set of distinct terms occurring in s_{jk}, and $\sum_{i \ | \ w_i \in s_{jk}} tc_{ik}$ is the sum of the tf-idf values associated with terms (i.e., word stems) in s_{jk}.

4.2.3.2 Sentence model coverage

The sentence coverage measures the pertinence of each sentence to the generated itemset-based model. To this aim, it considers document sentences tailored to the transactional data format. We first associate with each sentence $s_{jk} \in D$ a binary vector, denoted in the following as *sentence coverage vector*, SC_{jk}=$\{sc_1, \ldots, sc_{ms}\}$ where ms is the number of itemsets belonging to the model and $sc_i = \mathbf{1}_{tr_{jk}}(I_i)$ indicates whether itemset I_i supports or not tr_{jk} (see Section 4.2.2). More formally, $\mathbf{1}_{tr_{jk}}$ is an indicator function defined as follows:

$$\mathbf{1}_{tr_{jk}}(I_i) = \begin{cases} 1 & \text{if } I_i \subseteq tr_{jk}, \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

The coverage of a sentence s_{jk} with respect to the pattern-based model is defined as the number of ones that occur in the corresponding coverage vector SC_{jk}.

We formalize the problem of selecting the most informative and not redundant sentences according to the model as a set covering problem.

4.2.3.3 The set covering problem

The set covering optimization problem focuses on selecting the minimal set of sentences, of arbitrary size l and maximal score, whose logic OR of the corresponding coverage vectors, i.e., SC^*=$SC_1 \vee \ldots \vee SC_l$, generates a binary vector with the maximum number of 1's. This implies that each itemset belonging to the model covers at least one sentence. The SC^* vector will be denoted as the *summary coverage vector* in the rest of this section. ItemSum addresses the set covering problem to select sentences with the best model coverage and relevance score.

The set covering optimization problem is known to be NP-hard. To solve the problem, we adopt a greedy strategy similar to that successfully applied in [18] in the context of biological data feature selection. The greedy sentence selection strategy considers sentence model coverage as the most discriminative feature, i.e., sentences that cover the maximum number of itemsets belonging to the model are selected first. At equal terms, the sentence with maximal coverage that is characterized by the highest relevance score SR is preferred. A more detailed description of the adopted greedy strategy follows.

Algorithm 1 Greedy sentence selection

```
Input:    set of sentence relevance scores $SR$, set of sentence coverage vectors $SC$, tf-idf matrix $TC$
Output:   summary $\mathcal{S}$
1: $\mathcal{S} = \emptyset$
2: $ESC = \emptyset$ /*set of eligible sentence coverage vectors*/
3: $SC^* =$ all_zeros() /*summary coverage vector with only 0s*/
4: /*Cycle until either $SC^*$ contains only 1s or all the $SC$ vectors contain only zeros*/
5: while not (all_ones($SC^*$) or only_zeros($SC^*$)) do
6:     /*Determine the sentences with the highest number of ones*/
7:     $ESC =$ max_ones_sentences()
8:     if $ESC$ != $\emptyset$ then
9:         /*Select the sentence with maximum relevance score*/
10:        $SC_{best} = ESC[best]$ with $best = \arg_i \max SR_i$
11:        /*Update sets and summary_coverage_vector*/
12:        $\mathcal{S} = \mathcal{S} \cup SC_{best}$
13:        $SC^* = SC^*$ OR $SC_{best}$
14:        $ESC = ESC \setminus SC_{best}$
15:        /*Update the sentence coverage vectors belonging to $\mathcal{V}^*$*/
16:        for all $SC_i$ in $SC$ do
17:            $SC_i = SC_i$ AND $\overline{SC^*}$
18:        end for
19:    else
20:        break
21:    end if
22: end while
23: return $\mathcal{S}$
```

4.2.3.4 The greedy strategy

The adopted algorithm identifies, at each step, the sentence s_{jk} with the best complementary vector SC_{jk} with respect to the current summary coverage vector SC^*. The pseudo-code of the greedy approach is reported in Algorithm 1. It takes in input the set of sentence relevance scores SR, the set of sentence coverage vectors SC, and the tf-idf matrix TC. It produces the summary $\mathcal{S}$, i.e., the minimal subset of the most representative sentences. The first step is the variable initialization and the sentence coverage vector computation (lines 1-3). Next, the sentence with maximum coverage, i.e., the one whose coverage vector contains the maximum number of ones, is iteratively selected (line 6). At equal terms, the sentence with maximum relevance score (Cf. Formula 4.2) is preferred (line 10). Finally, the selected sentence is included in the summary $\mathcal{S}$ while the summary and sentence coverage vectors are updated (lines 12-18). The procedure iterates until either the summary coverage vector contains only ones, i.e., the model is fully covered by the summary, or the remaining sentences are not covered by any itemset, i.e., the remaining sentences are not pertinent to the model (line 5).

Experimental results, reported in Section 4.3.2.2.3, show that the proposed sentence selection algorithm is more effective and efficient than a branch-and-bound algorithm for text summarization purposes.

4.3 Experimental results

To compare ItemSum with the other approaches, we used the ROUGE [83] toolkit, which has been adopted as official DUC'04 tool for performance evaluation[1]. It measures the quality of a summary by counting the unit overlaps between the candidate summary and a set of reference summaries. Intuitively, the summarizer that achieves the highest ROUGE scores could be considered as the most effective one.

Results by ItemSum on a collection of real-life news articles are compared with OTS [112] and TexLexAn [109] in section 4.3.1. Several automatic evaluation scores are implemented in ROUGE. For the sake of brevity, we reported only ROUGE-2 and ROUGE-4 as representative scores. Analogous results have been obtained for the other scores.

We also evaluated our method on task 2 of DUC'04 [50], which is the latest DUC dataset on generic text summarization [136]. The provided collection is composed of 50 document groups, each of them including 10 documents. For each group, a golden summary is given. We compared our approach in section 4.3.2 with 30 methods submitted to the DUC conference and two other widely used text summarizers, i.e., the Open Text Summarizer (OTS) [112] and TexLexAn [109].

4.3.1 Results on a collection of real-life news articles

We conducted a set of experiments to address the following issues: (i) the effectiveness of the proposed summarization approach against two widely used summarizers, i.e., the Open Text Summarizer (OTS) [112] and TexLexAn [109] (Section 4.3.1.1), and (ii) the impact of the pattern-based model size and the support threshold on the performance of ItemSum (Section 4.3.1.2).

We evaluated all the summarization approaches on a collection of real-life news articles. To this aim, the 10 top-ranked news documents, provided by the Google web search engine (http://www.google.com), that concern the following recent news topics have been selected:

- **Natural Disaster**: Earthquake in Spain 2011
- **Royal Wedding**: Prince William and Kate Middleton wedding

[1]The provided command is: ROUGE-1.5.5.pl -e data -x -m -2 4 -u -c 95 -r 1000 -n 4 -f A -p 0.5 -t 0 -d -a

- **Technology**: Microsoft purchased Skype
- **Education**: Wealthy parents could buy their children places at elite universities
- **Sport**:Australia defeat Pakistan in Azlan shah Hockey

The datasets relative to the above news categories are made available for research purposes, upon request to the authors.

Since a "golden summary" (i.e., the optimal document collection summary) is not available for web news document, we performed a leave-one-out cross validation. More specifically, for each category we summarized nine out of ten news documents and we compared the resulting summary with the remaining (not yet considered) document, which has been selected as golden summary at this stage. Next, we tested all other possible combinations by varying the golden summary and we computed the average performance results, in terms of precision, recall, and F-measure, achieved by each summarizer for both ROUGE-2 and ROUGE-4.

4.3.1.1 Performance comparison and validation

We evaluated the performance, in terms of ROUGE-2 and ROUGE-4 precision (Pr), recall (R), and F-measure (F), of ItemSum against OTS and TexLexAn. For both OTS and TexLexAn we adopted the configuration suggested by the respective authors. For ItemSum we enforced a minimum support threshold min_sup=1.5% and we tuned the value of the pattern-based model size p to its best value for each considered dataset. A more detailed discussion on the impact of both min_sup and p on the performance of ItemSum is reported in Section 4.3.1.2.

ItemSum performs better than the other considered summarizers on all tested datasets. To validate the statistical significance of ItemSum performance improvement against OTS and TexLexAn, we used the paired t-test [46] at significance level $p - value = 0.05$ for all evaluated datasets and measures. For ROUGE-2, ItemSum provides significantly better results than OTS, whose summarization approach is mainly based on tf-idf measure, and TexLexAn in terms of precision and/or recall on 3 out of 5 datasets (i.e., Natural disaster, Technology and Sports). Moreover, ItemSum significantly outperforms TexLexAn and OTS in terms of F-measure (i.e., the harmonic average of precision and recall [129]) on, respectively, 2 and 3 of them (i.e.,

dataset	**ItemSum**				**OTS**			**TexLexAn**		
	p	R	Pr	F	R	Pr	F	R	Pr	F
Natural Disaster	16	**0.116**	**0.288**	**0.141**	0.040	0.120	0.053	0.038	0.114	0.045
Royal Wedding	12	**0.036**	**0.215**	**0.058**	0.034	0.174	0.054	0.030	0.150	0.047
Technology	5	**0.141**	**0.465**	**0.210**	0.042	0.208	0.067	0.042	0.172	0.065
Sports	10	**0.145**	**0.297**	**0.189**	0.055	0.133	0.075	0.071	0.149	0.093
Education	8	**0.039**	**0.241**	**0.064**	0.036	0.170	0.054	0.034	0.150	0.051

Table 4.1: Performance comparison in terms of ROUGE-2 score.

dataset	**ItemSum**				**OTS**			**TexLexAn**		
	p	R	Pr	F	R	Pr	F	R	Pr	F
Natural-Disaster	16	**0.060**	**0.125**	**0.068**	0.005	0.012	0.006	0.005	0.011	0.006
Royal-wedding	12	**0.009**	**0.082**	**0.015**	0.003	0.018	0.005	0.003	0.018	0.005
Technology	5	**0.113**	**0.356**	**0.167**	0.009	0.065	0.016	0.003	0.011	0.005
Sports	10	**0.059**	**0.112**	**0.077**	0.004	0.010	0.006	0.022	0.036	0.027
Education	8	**0.017**	**0.141**	**0.030**	0.003	0.012	0.005	0.003	0.009	0.004

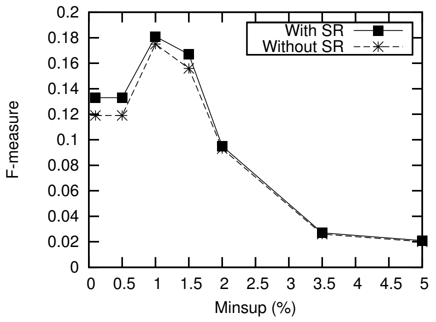
Table 4.2: Performance comparison in terms of ROUGE-4 score.

Natural disaster and Technology for both, and Sports for TexLexAn). Similar results were obtained for ROUGE-4.

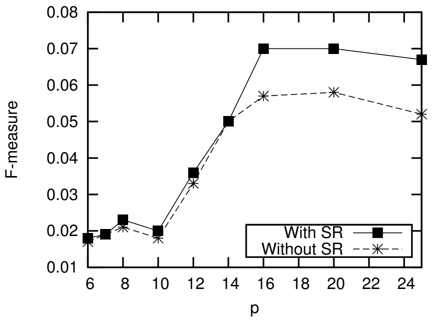
4.3.1.2 ItemSum parameter analysis

We analyzed the impact of the minimum support threshold and the pattern-based model size, i.e., the number of generated itemsets, on the performance of the ItemSum summarizer. To also test the impact of the tf-idf statistic on the performance of the pattern-based summarizer, we entail (i) neglecting the relevance score evaluation (i.e., by simply selecting the top-ranked maximal coverage sentence provided by the itemset miner [87]), and (ii) considering other statistical measures in place of the tf-idf score. Among all the evaluated scores, the tf-idf statistic turns out to be most effective measure in discriminating among sentences.

In Figures 4.2(a) and 4.2(b) we reported the F-measure achieved by ItemSum, by either considering or not the relevance score in the sentence evaluation, and by varying, respectively, the support threshold on Technology and the model size on the Natural Disaster document collection. For the



(a) Technology. p=5. Impact of the support threshold.



(b) Natural Disaster. min_sup=1.5%. Impact of the pattern-based model size.

Figure 4.2: ItemSum performance analysis by either considering or not of the relevance score (SR). Rouge-4 score. F-measure.

sake of brevity, we reported only the results obtained with the ROUGE-4 score. Analogous results have been obtained for the other ROUGE scores, for precision and recall measures, and for all other configurations.

The usage of the relevance score based on the tf-idf statistic always improves the performance of ItemSum in the range of those values of p and min_sup yielding the highest F-measure. This improvement is due to its ability to well discriminate sentence term occurrence among documents. When higher support thresholds (e.g., 5%) are enforced, many informative patterns are discarded, thus the model becomes too general to yield high summarization performance. Oppositely, when very low support thresholds (e.g., 0.1%) are enforced, data overfitting occurs, i.e., the model is too much specialized to effectively and concisely summarize the whole document collection content. At medium support thresholds (e.g., 1.5%) the best balancing between model specialization and generalization is achieved, thus, ItemSum produces very concise yet informative summaries.

The model size may also significantly affect the summarization performance. When a limited number of itemsets (e.g., $p = 6$) is selected, the relevant knowledge hidden in the news category Natural Disaster is not yet fully covered by the extracted patterns (see Figure 4.2(b)), thus the generated summaries are not highly informative. When $p = 16$ the pattern-based model provides the most informative and non-redundant knowledge. Consequently, the multi-document pattern-based summarization becomes very effective. When a higher number of itemsets is included in the model, the quality of the generated summaries worsens as the model is still informative but redundant. The best values of model size and support threshold achieved by each news category depend on the analyzed document term distribution.

4.3.2 Results on the DUC'04 document collection

We performed a set of experiments to address the following issues: (i) the performance comparison between ItemSum and other text summarizers (Section 4.3.2.1), and (ii) the impact of model size, support threshold, and set covering algorithm on the performance of ItemSum (Section 4.3.2.2)

To perform a fair comparison the golden and generated summaries have been preliminary normalized before using the ROUGE tool.

4.3.2.1 Performance comparison and validation

We evaluated the performance of ItemSum in terms of ROUGE-2, ROUGE-3, and ROUGE-4 precision (Pr), recall (R), and F-measure (F). For both OTS and TexLexAn we adopted the configuration suggested by the respective authors. For the DUC competitors the results provided by the DUC'04 system [50] are considered. For ItemSum we set, as standard configuration, the minimum support threshold *min_sup*=3% and the model size *ms*=12. A more detailed discussion on the impact of both *min_sup* and *ms* on the performance of ItemSum is reported in Sections 4.3.2.2.1 and 4.3.2.2.2. Table 4.3 summarizes the achieved results. For the lack of space, among the DUC'04 competitors, we reported just three representative ones, i.e., the two that achieve the best performance in terms of ROUGE-4 F-measure and one that achieves medium performance. ItemSum performs better than OTS, TexLexAn, and all the other considered summarizers for any tested measure, except for ROUGE-2 Precision. To validate the statistical significance of the performance improvements, we used the paired t-test [46] at 95% significance level for each evaluated dataset and measure. ItemSum significantly outperforms all the considered approaches in terms of ROUGE-4 F-measure (25 significant improvements out of 30 DUC'04 competitors) and 4 out 5 in terms of ROUGE-3 F-measure (24 out of 30). Although one of the DUC'04 competitors slightly outperforms ItemSum in terms of ROUGE-2 Precision, the performance worsening is not significant. Furthermore, ItemSum significantly outperforms OTS, TexLexAn, and 24 out of 30 DUC'04 competitors in terms of ROUGE-2 F-measure.

Summarizer		**ROUGE-2**			**ROUGE-3**			**ROUGE-4**		
		R	Pr	F	R	Pr	F	R	Pr	F
OTS		0.0746*	0.0740*	0.0743*	0.0236*	0.0234*	0.0235*	0.0087*	0.0086*	0.0087*
TexLexAn		0.0655*	0.0643*	0.0649*	0.0197*	0.0193*	0.0195*	0.0071*	0.0069*	0.0070*
DUC'04 competitors	peer102	0.0840	0.0846	0.0843	0.0264	0.0267	0.0265	0.0103*	0.0104*	0.0104*
	peer103	0.0774*	**0.0896**	0.0826	0.0243*	0.0284	0.0260*	0.0092*	0.0108	0.0099*
	peer140	0.0685*	0.0692*	0.0688	0.0218*	0.0220*	0.0219*	0.0093*	0.0094*	0.0093*
ItemSum		**0.0864**	0.0869	**0.0866**	**0.0307**	**0.0309**	**0.0308**	**0.0135**	**0.0136**	**0.0135**

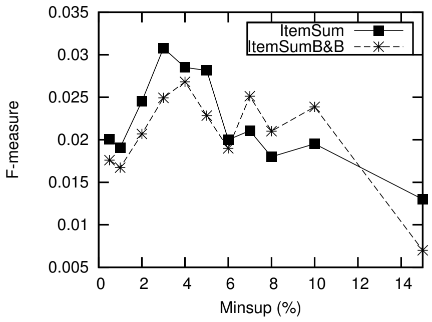
Table 4.3: Performance comparison in terms of ROUGE-2, ROUGE-3 and ROUGE-4 scores. Statistically relevant worsening in the comparisons between ItemSum and the other approaches are starred.

4.3.2.2 Parameter analysis

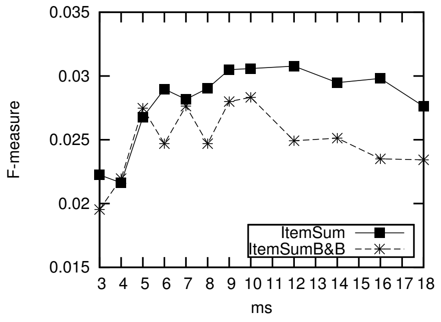
We analyzed the impact of the minimum support threshold and the pattern-based model size, i.e., the number of selected itemsets, on the performance of ItemSum. To also evaluate the impact of the greedy algorithm on the summarization performance, we tested a slightly modified version of ItemSum, which adopts a branch-and-bound algorithm [108] to solve the set covering problem, as well. For the lack of space, we only reported, in Figures 4.2(a) and 4.2(b), the achieved ROUGE-3 F-measure values. Analogous results have been obtained for the other ROUGE scores, for precision and recall measures, and for all other configurations. In the following, we discuss the impact of each considered factor separately.

4.3.2.2.1 Impact of the support threshold When higher support thresholds (e.g., 10%) are enforced, many informative itemsets are discarded, thus the itemset-based model becomes too general to yield high summarization performance. Oppositely, when very low support thresholds (e.g., 0.1%) are enforced, data overfitting occurs, i.e., the model is too much specialized to effectively and concisely summarize the whole document collection content. At medium support thresholds (e.g., 3%) the best balancing between model specialization and generalization is achieved, thus, ItemSum produces succinct yet informative summaries.

4.3.2.2.2 Impact of the model size The model size may significantly affect the summarization performance. When a limited number of itemsets (e.g., ms=5) is selected, the relevant knowledge hidden in the document collection is not yet fully covered by the extracted patterns (see Figure



(a) DUC'04 documents. *ms*=12. Impact of the support threshold.



(b) DUC'04 documents. *min_sup*=3%. Impact of the pattern-based model size.

Figure 4.3: Comparison between ItemSum and ItemSum$_{B\&B}$ on Rouge-3 with different parameter settings.

4.2(b)), thus the generated summaries are not highly informative. The multi-document itemset-based summarizer becomes very effective when the model size is around 12 as selected patterns are representative of the most informative and non-redundant knowledge. Differently, when the model size further increases the quality of the generated summaries worsens as the model is still informative but redundant. The best value of model size depends on the analyzed document term distribution.

4.3.2.2.3 Impact of the set covering algorithm ItemSum exploits a greedy algorithm to solve the set covering optimization problem. Generally speaking, it produces an approximated solution to the problem of selecting the set of sentences that best covers the itemset-based model. Since the set covering problem is a min-max problem, it may be converted to a linear programming problem and addressed by using combinatorial optimization strategies. Thus, we compared the performance achieved by ItemSum with that achieved by a slightly modified version, namely ItemSum$_{B\&B}$, which exploits a branch-and-bound implementation [108] to address the set covering task. In most cases ItemSum outperforms ItemSum$_{B\&B}$ and shows a more stable trend in terms of ROUGE scores. In particular, when high-quality models are generated (e.g., when $p = 12$ and $min\_sup = 3\%$) the best results are achieved by the greedy approach. ItemSum$_{B\&B}$ yields better results when the itemset-based model is low-quality or redundant (e.g., when $p < 6$ or $min\_sup > 5\%$). However, it does not provide any significant performance improvement with respect to ItemSum, while it requires a higher execution time (e.g., around 5% more when $p = 12$ and $min\_sup = 3\%$).

Chapter 5

YagoSum

This chapter describes Yago-based Summarizer, that relies on an ontology-based evaluation and selection of the document sentences. Section 5.1 presents introduction while some related work is discussed in section 5.2. The main steps of the framework of Yago-based Summarizer, are presented in section 5.3 while 6.4 discusses the evaluated results.

5.1 Introduction

Discovering the most salient information hidden in textual Web documents is often a challenging task. In fact, the huge volume of electronic documents that users could retrieve from the Web is commonly very difficult to explore without the help of automatic or semi-automatic tools. To tackle this issue, a particular attention has been paid to the development of text summarization tools. Summarizers focus on generating a succinct representation of a textual document collection. Specifically, sentence-based multi-document summarizers generate concise yet informative summaries of potentially large document collections, which consist of the most representative document sentences.

A significant research effort has been devoted to tackling the summarization problem by means of general-purpose information retrieval or data mining techniques. For example, clustering-based approaches (e.g., [136, 137]) adopt clustering algorithms to group document sentences into homogeneous clusters and then select the most authoritative representatives within each group. In contrast, graph-based approaches (e.g., [105, 145, 150]) first generate a graph-based model in which the similarity relationships between pairs of sentences are represented. Next, they exploit popular indexing strategies (e.g., PageRank [29]) to identify the most salient sentences (i.e., the most authoritative graph nodes).

In recent years, the increasing availability of semantics-based models (e.g., ontologies and taxonomies) has prompted researchers to investigate their usefulness for improving summarizer performance. Ontologies are formal representations of the most peculiar concepts that are related to a specific knowledge domain and their corresponding relationships [17]. Ontologies find application in several research contexts, among which user-generated content analysis [65], e-learning platform development [80], and video and image analysis [134]. The attention of the research community has been focused on both learning meaningful ontologies that contain salient document keywords [21] and improving the performance of the document summarization process by integrating ontological knowledge [67, 77, 36, 103]. For example, ontologies have been used to identify the document concepts that are strongly correlated with a user-specified query [77, 36] or to map the document content to non-ambiguous ontological concepts [67, 103].

However, semantics-based document analysis is often applied as a preprocessing step, rather than integrating the discovered knowledge into the summarization process. In contrast, In this work, it is aimed to tightly integrate the ontology-based document analysis into the summarization process in order to take the semantic meaning of the document content into account during the sentence evaluation and selection processes. With this in mind, a new multi-document summarizer, namely Yago-based Summarizer, is proposed that integrates an established ontology-based entity recognition and disambiguation step. Specifically, a popular ontological knowledge base, i.e., Yago [126], is used to identify the key document concepts. The same concepts are also evaluated in terms of their significance with respect to the actual document context. The result of the evaluation process is then used to select the most representative document sentences. In such a way, the knowledge that is inferred from Yago is tightly integrated into the sentence evaluation process. Finally, a variant of the Maximal Marginal Relevance (MMR) evaluation strategy [31] is adopted to iteratively choose the best subset of informative yet non-redundant sentences according to the previously assigned sentence ranks.

The Yago-based Summarizer, described in this chapter relies on an ontology-based evaluation and selection of the document sentences. To capture the actual meaning and context of the document sentences and generate sound document summaries, an established entity recognition and disambiguation step based on the Yago ontology is integrated into the summarization process.

The experimental results, which were achieved on the DUC'04 benchmark collections, demonstrate the effectiveness of the proposed approach compared to a large number of competitors as well as the qualitative soundness of the generated summaries.

5.2 Related work

A significant research effort has been devoted to summarizing document collections by exploiting information retrieval or data mining techniques. Two main summarization strategies have been proposed in literature. *Sentence-based* summarization focuses on partitioning documents in sentences and generating a summary that consists of the subset of most informative sentences (e.g., [32, 59, 136]). In contrast, *keyword-based* approaches focus on detecting salient document keywords using, for instance, graph-based indexing [84, 145, 150] or latent semantic analysis [48]. Since sentence-based approaches commonly generate humanly readable summaries without the need for advanced postprocessing steps, our summarizer relies on a sentence-based approach. Summarizers can be further classified as constraint-driven if they entail generating a summary that satisfy a set of (user-specified) constraints [10]. For example, sentences that are pertinent to a user-specified query can be selected [91, 101]. Unlike [10, 91, 101] our summarizer relies on a constraint-less approach.

Most of the recently proposed (constraint-less) sentence-based summarizers exploit one of the following general-purpose techniques: (i) clustering, (ii) graph mining, (iii) linear programming, and (iv) itemset mining. Clustering-based approaches (e.g., [136, 137]) group document sentences into homogeneous clusters and then select the best representatives (e.g., the centroids or the medoids [129]) within each cluster. While the authors in [136] propose a static summarization framework, the work that was first presented in [137] addresses the problem of incremental summary update: whenever a set of documents is added/removed from the initial collection, the previously generated summary is updated without the need for recomputing the whole clustering model. In parallel, some attempts to cluster documents rather than sentences have also been made [106, 20]. For example, MEAD [106] analyzes the cluster centroids and generates a pseudo-document that includes the sentences with the highest tf-idf term values [83]. Then, the sentence selection process is driven by a score that considers (i) the sentence similarity with the centroids, (ii) the sentence position within the document, and (iii) the sentence length. A similar approach has also been adopted to summarize articles coming from the biological domain [20]. To tailor the generated summaries to the most relevant biological knowledge biologists are asked to provide a dictionary that is used to drive the sentence selection process. Similarly, this work also considers the document context to improve the summarization performance. Unlike [20], it exploits an ontological knowledge base, rather than a plain-text dictionary, to drive the sentence evaluation and selection process.

Graph-based approaches to sentence-based summarization (e.g., [105, 133, 135, 145, 150]) generate a graph in which the nodes represent the document sentences, whereas the edges are weighted by a similarity measure that is evaluated on each node pair. Popular indexing strategies (e.g., PageRank [29], HITS [76]) are exploited to rank the sentences based on their relative authoritativeness in the generated graph. In parallel, other approaches formalize the sentence selection task as a min-max optimization problem and tackle it by means of linear programming techniques [9, 11, 56, 128]. Still others analyze the underlying correlations among document terms by exploiting (i) frequent itemset mining techniques [19], (ii) probabilistic approaches [43, 44], or (iii) the Singular Value Decomposition (SVD) [124].

Ontologies have already been exploited to improve the document summarization performance. Specifically, they have been used to (i) identify the concepts that are either most pertinent to a user-specified query [77, 36] or most suitable for performing query expansion [94], (ii) model the context in which summaries are generated in different application domains (e.g., the context-aware mobile domain [58], the business domain [143], the disaster management domain [82]), and (iii) enrich existent ontological models with textual content [21]. Some attempts to consider the text argumentative structure into account during the summarization process have also been made. For example, in [103] the authors propose to identify and exploit salient lexical chains to generate accurate document summaries. The summarizer proposed in [67] exploits Support Vector Machines (SVMs) to maps each sentence to a subset of taxonomy nodes. Similarly, in [15] a rhetorical role is assigned to each sentence by a stochastic CRF classifier [114], which is trained from a collection of annotated sentences. Unlike [15, 67] our approach does not rely on classification models. Furthermore, since our summarizer evaluates the sentence relevance regardless of the underlying document structure, our approach is, to some extent, complementary to the ones that have previously been proposed in [15, 103].

5.3 Yago-based Summarizer

Yago-based Summarizer is a novel multiple-document summarizer that exploits the Yago ontological knowledge base [126] to generate accurate document summaries.

Consider a collection of textual documents $D=\{d_1, \ldots, d_N\}$, where each document $d_i \in D$ is composed of a set of sentences s_1^i, …, s_M^i. The summarizer generates a summary $S=\{s_j^i\}$ $1 \leq i \leq N$, $1 \leq j \leq M$. The summary includes a worthwhile subset of sentences that are representative of the whole collection D.

Figure 5.1 outlines the main Yago-based Summarizer steps, which are briefly summarized below.

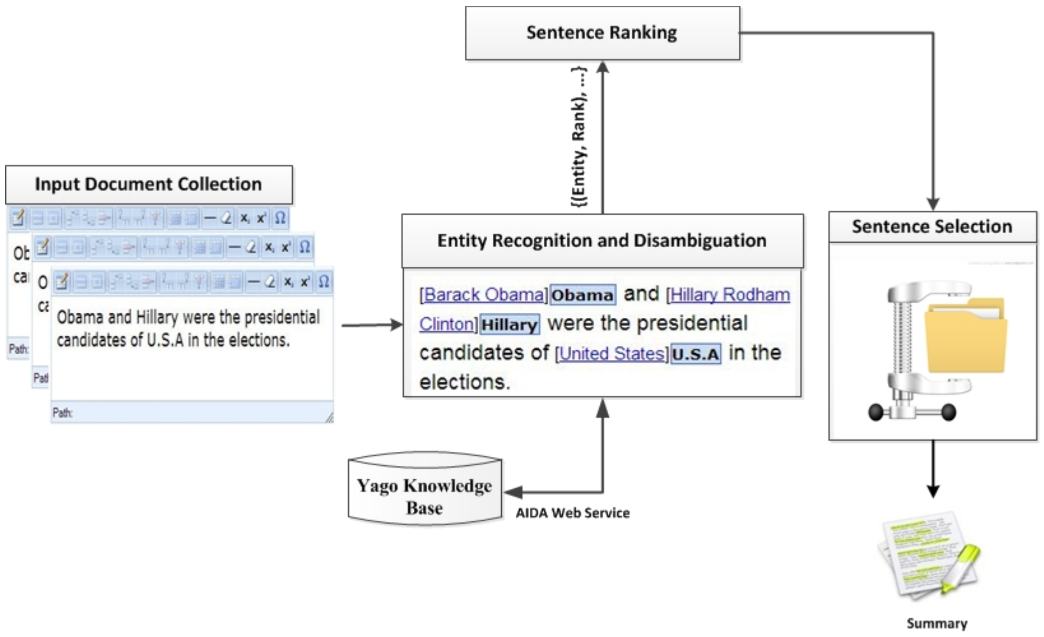

Sentence Ranking
{(Entity, Rank), ...}
Input Document Collection
Obama and Hillary were the presidential candidates of U.S.A in the elections.
Entity Recognition and Disambiguation
[Barack Obama] Obama and [Hillary Rodham Clinton] Hillary were the presidential candidates of [United States] U.S.A in the elections.
Sentence Selection
Yago Knowledge Base
AIDA Web Service
Summary

Figure 5.1: The Yago-based Summarizer.

- **Entity recognition and disambiguation.** This step analyzes the input document collection with the goal of identifying the most relevant concepts and their corresponding context of use. To this aim, the Yago knowledge base is used to map the words that occur in the document sentences to non-ambiguous ontological concepts, called *entities*. To discriminate between multiple candidate entities for the same word combination, it adopts an entity relevance score that considers both the popularity and the contextual pertinence of each candidate entity in the analyzed document collection.

- **Sentence ranking.** To include in the summary only the most pertinent and semantically meaningful document content, sentences are evaluated and ranked according to the previously assigned entity scores.

- **Sentence selection.** To generate a summary of the document collection an iterative procedure is applied to select the top-ranked sentences that are least similar to the previously selected ones.

5.3.1 Entity recognition and disambiguation

Entity recognition and disambiguation are established document analysis tasks that aim at mapping the natural text to a set of non-ambiguous ontological concepts [129]. Yago-based Summarizer exploits the Yago ontological knowledge base [126], which relies on the Wikipedia free encyclopedia [142], to support the entity recognition and disambiguation process. The Yago analytical procedures have been called through the AIDA Web Service [68].

Consider a sentence s_j^i that is composed of a collection of (possibly repeated) words w_1, w_2, $\ldots$, w_Z. The goal is to map words w_k, $1 \leq k \leq Z$ to Yago ontological concepts, i.e., the entities. Note that an entity may be associated either with a single word or with a combination of words. The entity recognition step recognizes the entities that are associated with noun, dates, times, or numbers. As a clarifying example, consider the sentence reported in the left-hand side of Figure 5.2.

Each of the three underlined word combinations *Mercury*, *Solar System*, and *Sun* is associated with at least one candidate entity in Yago. Note that not all the sentence words match at least one Yago entity. For example, the word *Innermost* has no matching entity. Furthermore, some words have many candidate entities, meaning that a word could have different meanings in different contexts. For example, *Mercury* could be associated with the candidate entities *Mercury(Element)* and *Mercury(Planet)*, which correspond to the well-known chemical element and planet, respectively. Note also that for each entity Yago provides (i) a popularity score, which reflects its frequency of usage (e.g., 241 for *Mercury(Element)*), (ii) a list of related keywords (e.g., *Chemistry*, *Liquid*) for its corresponding context of use, and (iii) the number of incoming and outcoming Wikipedia links [142].

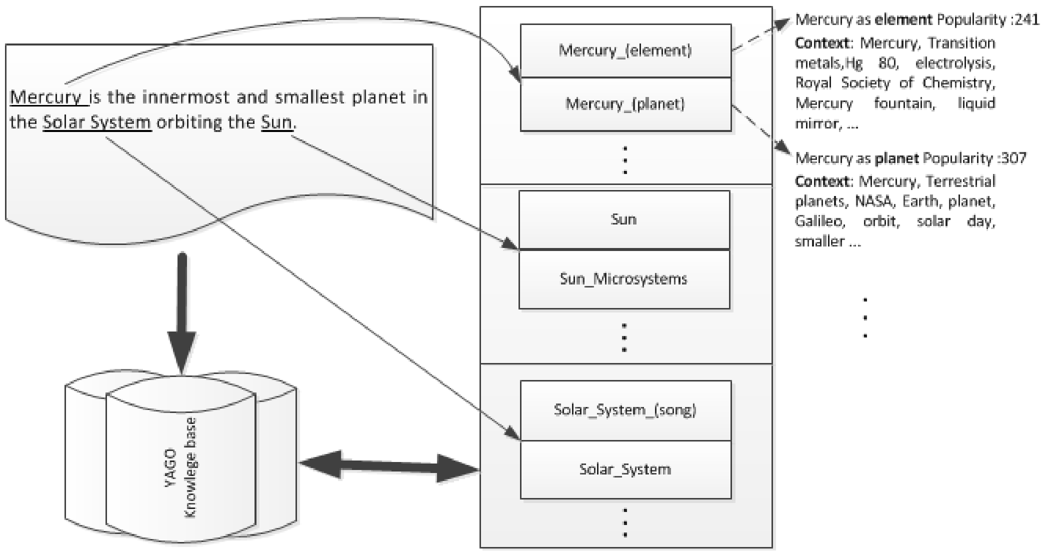

Mercury is the innermost and smallest planet in the Solar System orbiting the Sun.
YAGO Knowlege base
Mercury_(element)
Mercury_(planet)
Sun
Sun_Microsystems
Solar_System_(song)
Solar_System
Mercury as element Popularity :241
Context: Mercury, Transition metals,Hg 80, electrolysis, Royal Society of Chemistry, Mercury fountain, liquid mirror, ...
Mercury as planet Popularity :307
Context: Mercury, Terrestrial planets, NASA, Earth, planet, Galileo, orbit, solar day, smaller ...

Figure 5.2: Entity Recognition and Disambiguation example.

Entity recognition for times, date, and numbers is based on regular expressions and returns a single entity. For example, the expression *March, 1st 2012* corresponds to the date 01/03/2012. Conversely, the entity recognition procedure for nouns could return many candidate entities. Hence, in the latter case a disambiguation step is applied in order to select, among the candidate entities, the most appropriate one. To tackle this issue, each candidate entity is weighted by a relevance score, which considers both its popularity and pertinence to the analyzed document context. Specifically, the rank $entityRank(e_q)$ of an entity e_q with respect to a word w_k that occurs in the document $d_i \in D$, is defined as follows:

$$\begin{aligned} entityRank(e_q) &= \theta \cdot popularity(e_q) \\ &+ \phi \cdot sim(cxt(e_q), cxt(d_i)) \\ &+ (1-\theta-\phi) \cdot coh(e_q, D) \end{aligned} \tag{5.1}$$

where $\theta, \phi \in [0,1]$ are user-specified parameters that weigh the importance of each summation term, *popularity(e_q)* is the Yago popularity score that is associated with the candidate entity e_q, *sim(cxt(e_q),cxt(d_i))* is the similarity between the context of use of the candidate entity and the document d_i, and *coh(e_q, D)* is the coherence of e_q with respect to the whole document collection. By following the indications reported in [68], we set the values of θ and ϕ to 0.34 and 0.47, respectively. A thorough assessment of the entity recognition system performance on real data is also reported in [64]. The first summation term is a popularity score, which indicates the global frequency of occurrence of the concept in the knowledge base. For instance, in Yago *Mercury-Planet* has, on average, a higher popularity score than *Mercury-Element* (307 against 241). However, the relevance of an entity within a document also depends on its context of use. Hence, the second summation term indicates the pertinence of the entity to the document. Specifically, it measures the cosine distance [129] between the context of the candidate entity e_q, i.e., the list of contextual keywords that are provided by Yago (e.g., *Chemistry*, *Liquid* for the candidate entity *Mercury-Element* in Figure 5.2), and the context of the word w_k at the document level (i.e., the list of words that co-occur with w_k in d_i). Roughly speaking, the more contextual keywords match the document content the higher the pertinence of the candidate entity is. Finally, since the recognized entities are likely to be correlated each other, the last summation term measures the coherence of the candidate entity with respect to all of the other recognized candidate entities that correspond to any word in D. Since coherent entities are likely to share many Wikipedia links, similar to [68], we evaluate the entity coherence within the document collection D as the number of incoming Wikipedia links that are shared by e_q and all of the other candidate entities that have been recognized in D.

The entity scores will be used to drive the summary generation process, as discussed in the following sections.

5.3.2 Sentence ranking

Yago-based Summarizer exploits the semantic knowledge that has been inferred at the previous step to evaluate and rank the document sentences according to their significance in the document collection. To this aim, a rank is associated with each document sentence. The sentence rank reflects the relevance of the entities associated with its corresponding sentence words.

Let s_j^i be an arbitrary sentence and $E(s_j^i)$ the set of entities (nouns, date, times, or numbers) that are associated with any word $w_k \in s_j^i$. The s_j^i's rank is computed as follows:

$$SR(s_j^i) = \frac{\sum_{e_q \in E(s_j^i)} EntityScore(e_q)}{|E(s_j^i)|} \tag{5.2}$$

where $EntityScore(e_q)$ is defined by

$$EntityScore(e_q) = \begin{cases} \gamma & \text{if } e_q \text{ is a date, time, or number entity,} \\ \gamma + EntityRank(e_q) & \text{if } e_q \text{ is a named entity} \end{cases} \tag{5.3}$$

γ is a user-specified parameter that is used to privilege the sentences that contain many recognized entities. $\sum_{e_q \in E(s_j^i)} EntityScore(e_q)$ is the summation of the entity ranks of all of the entities that are mapped to any word in s_j^i (see Definition 5.1). Note that the sentences that do not contain any recognized Yago entity have minimal sentence rank (i.e., 0), because they are not likely to contain any semantically relevant concept. In contrast, because of the γ correction, the sentences that contain only dates, times, or numbers are considered to be, on average, more relevant than the former ones, but less relevant than those that also contain named entities. The impact of the user-specified parameter γ on the summarization performance is discussed in Section 6.4.

5.3.3 Sentence selection

Given a sentence ranking, the selection step focuses on generating the output summary of the document collection by including only the most representative sentences. To achieve this goal, Yago-based Summarizer adopts a variant of an established iterative re-ranking strategy, called Maximal Marginal Relevance (MMR) [31]. MRR has first been introduced in the context of query-based summary generation. At each algorithm iteration, it picks out the candidate sentence that is characterized by (i) maximal relevance with respect to the given query and (ii) minimal similarity with respect to the previously selected sentences. Since our approach is not query-based, we adapt the former selection strategy to the problem under analysis. Specifically, Yago-based Summarizer selects, at each iteration, the top-ranked sentence with minimal redundancy with respect to the already selected sentences. At each iteration the former optimization problem can be formulated as follows:

$$\begin{aligned} &\underset{\{s_j^i\}}{\text{maximize}} \quad \alpha \cdot SR(s_j^i) - (1-\alpha) \cdot sim(s_j^i, \bar{s}_t^r) \\ &\text{subject to} \\ &\qquad\qquad \alpha \in [0,1] \\ &\qquad\qquad s_j^i \notin S \\ &\qquad\qquad \bar{s}_t^r \in S \end{aligned} \tag{5.4}$$

where S is the output summary that possibly includes some of the document sentences $\bar{s}_t^r$, α is a user-specified parameter, and $\{s_j^i\}$ is the set of candidate sentences not yet included in the summary. The sentence ranking is evaluated using the expression reported in Formula 5.2. Furthermore, the similarity $sim(s_j^i, \bar{s}_t^r)$ between the pair of sentences s_j^i and $\bar{s}_t^r$ is evaluated using the cosine similarity [129] and takes value zero when the summary is empty (i.e., at the first algorithm iteration). The impact of the entity relevance and the similarity score is weighted by the α parameter. Specifically, the higher the value of α is, the more important the entity relevance score is with respect to the similarity score. Therefore, setting relatively high α values could yield informative but partially redundant summaries. Conversely, for lower α values the sentence relevance is partially neglected in behalf of a lower summary redundancy.

5.4 Experimental results

We performed a variety of experiments to address the following issues: (i) a performance comparison between Yago-based Summarizer and many state-of-the-art summarizers on document benchmark collections (see Section 5.4.2), (ii) a qualitative comparison between the summaries generated by our approach and those produced by two representative competitors (see Section 5.4.3), and (iii) an analysis of the impact of the main system parameters on the Yago-based Summarizer performance (see Section 5.4.4).

All the experiments were performed on a 3.0 GHz 64 bit Intel Xeon PC with 4 GB main memory running Ubuntu 10.04 LTS (kernel 2.6.32-31). The source code for Yago-based Summarizer is available, for research purposes, upon request to the authors. A detailed description of the experimental evaluation context is given below.

5.4.1 Evaluation context

We evaluated the Yago-based Summarizer performance on the task 2 of the Document Understanding Conference (DUC) 2004, which is the latest benchmark contest that were designed for generic English-written multi-document summarization [136]. The analyzed DUC'04 collections have been provided by the contest organizers [50]. They consist of a large variety of English-written articles which range over different subjects. According to their subject, articles were preliminary clustered in 50 document groups. Each homogeneous collection contains approximately 10 documents. Furthermore, for each collection at least one *golden* summary is given by the DUC'04 organizers. Participants to the DUC'04 contest had to submit their own summaries and compare them with the reference (golden) ones. The more similar the generated summaries are to the reference models, the more accurate the summarization process is.

To perform an analytical comparison between the summarizers' performance on the task 2 of DUC'04 we used the ROUGE toolkit [83], which has been adopted as official DUC'04 tool for performance evaluation[1]. ROUGE measures the quality of a summary by counting the unit overlaps between the candidate summary and a set of reference summaries (i.e., the golden summaries). The summary that achieves the highest ROUGE score could be considered to be the most similar to the golden summary. To perform a fair comparison, before using the ROUGE toolkit we normalized the generated summaries by truncating each of them at 665 bytes (we round the number down in case of straddled words). Several automatic evaluation scores are implemented in ROUGE. As previously done in [19, 136], we will report only the ROUGE-2 and ROUGE-4 representative scores [83]. Similar results were achieved for the other ROUGE scores.

5.4.2 Performance comparison on the DUC'04 collections

We compared the Yago-based Summarizer performance on the DUC'04 benchmark collections with that of: (i) the 35 summarizers submitted to the DUC'04 conference, (ii) the 8 summaries generated by humans and provided by the DUC'04 system (beyond the golden summaries), (iii) two widely used open source text summarizers, i.e., the Open Text Summarizer (OTS) [112] and TexLexAn [109], (iv) a recently proposed itemset-based summarizer [19], named ItemSum (Itemset-based Summarizer), and (v) a baseline version of Yago-based Summarizer, namely Baseline, which adopts an established term relevance evaluator, i.e., the tf-idf score [83], rather than the ontology-based entity rank evaluator (see Definition 5.1).

[1]The provided command is: ROUGE-1.5.5.pl -e data -x -m -2 4 -u -c 95 -r 1000 -n 4 -f A -p 0.5 -t 0 -d -a

For the DUC'04 competitors we considered the results that were provided by the DUC'04 system [50]. Specifically, for the top-ranked DUC'04 summarizer, i.e., CLASSY [44], we considered its most effective version (i.e., peer65). Similarly, for the other competitors we tuned the algorithm parameters to their average best value by following the indications that were given by the respective authors. For Yago-based Summarizer we set, as *standard* configuration, γ to 0.3 and α to 0.9. In Section 5.4.4 we analyze more in detail the impact of both parameters on the Yago-based Summarizer performance.

Table 5.1 summarizes the results that were achieved by Yago-based Summarizer, Baseline-tf-idf, ItemSum, OTS, TexLexAn, the 8 humanly generated summaries, and the 10 most effective summarizers presented in the DUC'04 contest. To validate the statistical significance of the Yago-based Summarizer performance improvement against its competitors we performed the paired t-test [46] at 95% significance level for all of the evaluated measures. Every statistically relevant worsening in the comparison between Yago-based Summarizer and the other approaches is starred in Table 5.1.

Table 5.1: DUC'04 Collections. Comparisons between Yago-based Summarizer and the other approaches. Statistically relevant differences in the comparisons between Yago-based Summarizer (standard configuration) and the other approaches are starred.

Summarizer		ROUGE-2			ROUGE-4		
		R	Pr	F	R	Pr	F
TOP RANKED DUC'04 PEERS	peer120	0.076*	**0.103**	0.086*	0.014*	**0.019**	0.016
	peer65	0.091*	0.090*	0.091*	0.015*	0.015	0.015*
	peer19	0.080*	0.080*	0.080*	0.010*	0.010*	0.010*
	peer121	0.071*	0.085*	0.077*	0.012*	0.014*	0.013*
	peer11	0.070*	0.087*	0.077*	0.012*	0.015*	0.012*
	peer44	0.075*	0.080*	0.078*	0.012*	0.013*	0.012*
	peer81	0.077*	0.080*	0.078*	0.012*	0.012*	0.012*
	peer104	0.086*	0.084*	0.085*	0.011*	0.010*	0.010*
	peer124	0.083*	0.081*	0.082*	0.012*	0.012*	0.012*
	peer35	0.083*	0.084	0.083*	0.010*	0.011*	0.011*
DUC'04 HUMANS	A	0.088*	0.092*	0.090*	0.009*	0.010*	0.010*
	B	0.091*	0.096	0.092	0.013*	0.013*	0.013*
	C	0.094	0.102	0.098	0.011*	0.012*	0.012*
	D	0.100	**0.106**	0.102	0.010*	0.010*	0.010*
	E	0.094	0.099	0.097	0.011*	0.012*	0.012*
	F	0.086*	0.090*	0.088*	0.008*	0.009*	0.009*
	G	0.082*	0.087*	0.084*	0.008*	0.008*	0.007*
	H	**0.101**	0.105	**0.103**	0.012*	0.013*	0.012*
OTS		0.075*	0.074*	0.074*	0.009*	0.009*	0.009*
texLexAn		0.067*	0.067*	0.067*	0.007*	0.007*	0.007*
ItemSum		0.083*	0.085*	0.084*	0.012*	0.014*	0.014*
Baseline		0.092*	0.091*	0.092*	0.014*	0.014*	0.014*
Yago-based Summarizer		**0.095**	0.094	**0.095**	**0.017**	0.017	**0.017**

Yago-based Summarizer performs significantly better than ItemSum, OTS, TexLexAn, and Baseline for all of the analyzed measures. Hence, the ontology-based sentence ranking and selection strategies appear to be more effective than traditional information retrieval techniques (e.g., the tf-idf-based sentence evaluation [83]) for summarization purposes. Although, in some cases, peer120 performs best in terms of ROUGE-2 and ROUGE-4 precision, Yago-based Summarizer performs significantly better than all the 35 DUC'04 competitors in terms of ROUGE-2 and ROUGE-4 F1-measure (i.e., the harmonic average between precision and recall). Hence, the summaries that were generated by Yago-based Summarizer are, on average, the most accurate and not redundant ones.

Compared to the 8 humanly generated summaries, Yago-based Summarizer significantly outperforms 3 out of 8 and 8 out of 8 competitors in terms of ROUGE-2 and ROUGE-4 F1-measure, respectively. In contrast, CLASSY (peer65) performs significantly better than 2 out of 8 and 6 out of 8 humans in terms of ROUGE-2 and ROUGE-4 F1-measure, respectively. Similarly, peer120 performs worser than all the humans in terms of ROUGE-2 and outperforms 7 out of 8 humans in terms of ROUGE-4. Hence, Yago-based Summarizer placed, on average, better than CLASSY and peer120 with respect to the humans.

5.4.3 Summary comparison

We conducted a qualitative evaluation of the soundness and readability of the summaries that were generated by Yago-based Summarizer and the other approaches. Tables 5.2 reports the summaries that were produced by Yago-based Summarizer, the top-ranked DUC'04 summarizer, i.e., CLASSY [44] (Peer-65), and a commonly used open source summarizer OTS [112] on a representative DUC'04 collection, which relates the activities and the main achievements of the Yugoslav war crime tribunal.

The summary that was generated by Yago-based Summarizer appears to be the most focused one, because it covers all the main document topics, i.e., (1) the role of the Yugoslav war crime tribunal, (2) the acquittal of the Muslim military commander, and (3) the arrest of the Bosnian Serb general. In contrast, OTS and CLASSY cover, to some extent, only the topic (2). On the other hand, both OTS and CLASSY select other contextual sentences about the Kosovo war, which are very general and not representative of the key document message. Hence, the corresponding summaries are deemed to be partially redundant.

Table 5.2: Summary examples.

Method	Summary
Yago-based Summarizer	Yugoslavia must cooperate with the U.N. war crimes tribunal investigating alleged atrocities during the wars in Croatia and Bosnia, international legal experts meeting in Belgrade said Sunday. The Yugoslav war crimes tribunal Monday acquitted a Muslim military commander of war crimes against Bosnian Serb prisoners in 1992, but convicted three underlings in the first U.N. case dealing with anti-Serb atrocities. American and allied forces in Bosnia on Wednesday arrested a Bosnian Serb general who was charged with genocide by the international war crimes tribunal in a recent secret indictment.
CLASSY	Some of the closest combat in the half year of the Kosovo conflict, to the point of fighting room to room and floor to floor, occurred near this village six weeks ago, in the days before 21 women, children and elderly members of the Delijaj clan were massacred by Serbian forces, their mutilated bodies left strewn on the forest floor. In its first case to deal with atrocities against Serbs during Bosnia's civil war, a U.N. war crimes tribunal on Monday convicted three prison officials and guards, but acquitted a top military commander who oversaw the facility. Hundreds of people gathered at Sarajevo airport on Saturday to welcome Zejnil Delalic, who was cleared of war crimes charges earlier this week after spending 980 days in jail of the international war crimes tribunal in The Hague.
OTS	The Yugoslav war crimes tribunal Monday acquitted a Muslim military commander of war crimes against Bosnian Serb prisoners in 1992, but convicted three underlings in the first U.N. case dealing with anti-Serb atrocities. The Yugoslav war crimes tribunal cleared Zejnil Delalic, a Muslim, of responsibility for war crimes committed against Serb captives at a Bosnian government-run prison camp under his command. court convicted camp commander Zdravko Mucic, a Croat, of 11 war crimes and grave breaches of the Geneva Conventions because he oversaw guards who murdered nine Serbs and tortured six. Indeed, the conflict between Serbian forces bent on keeping Kosovo in Serbia and guerrillas fighting for the independence of its heavily ethnic Albanian population first drew international attention with the massacre of the Jasari clan in early March by Serbian units at Prekaz, in central Kosovo.

5.4.4 Parameter setting

The setting of the user-specified α and γ parameters could affect the Yago-based Summarizer performance significantly. Hence, we thoroughly analyzed their impact on the Yago-based Summarizer ROUGE scores.

Figures 5.3 and 5.4 plot the ROUGE-2 and ROUGE-4 F1-measure that were achieved by Yago-based Summarizer on the DUC'04 collection by varying the value of γ in the range [0,1] and by setting α to its best value (0.9), respectively. In contrast, Figures 5.5 and 5.6 plot the ROUGE-2 and ROUGE-4 F1-measure scores by setting the best γ value (0.3) and by varying α in the range [0,1].

When increasing the value of the γ parameter, the sentences that include many unrecognized words are on average penalized (see Formula 5.1). Based on the results that are reported in Figures 5.3 and 5.4, the best performance

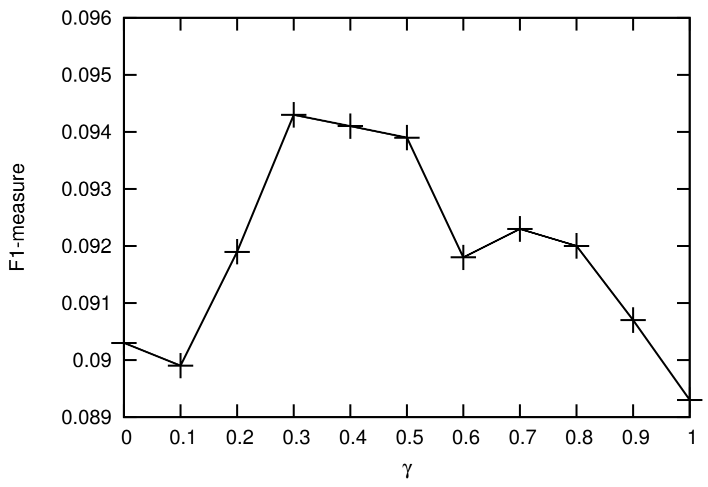


Figure 5.3: ROUGE-2 F1-measure: Impact of γ on the Yago-based Summarizer performance. α=0.9. DUC'04 collections.

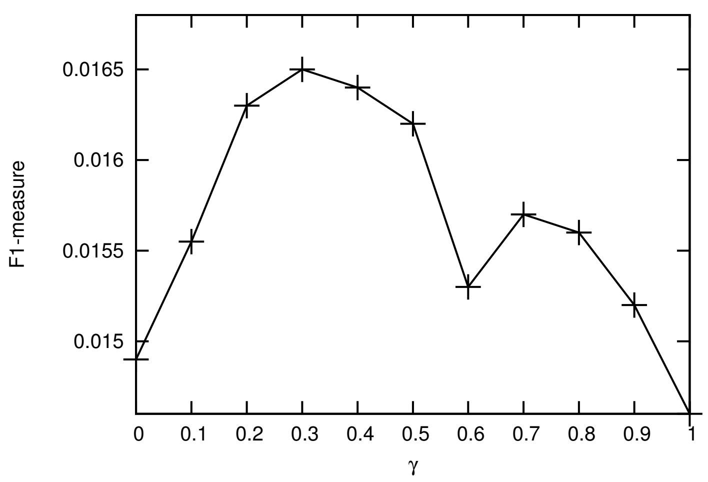


Figure 5.4: ROUGE-4 F1-measure: Impact of γ on the Yago-based Summarizer performance. α=0.9. DUC'04 collections.

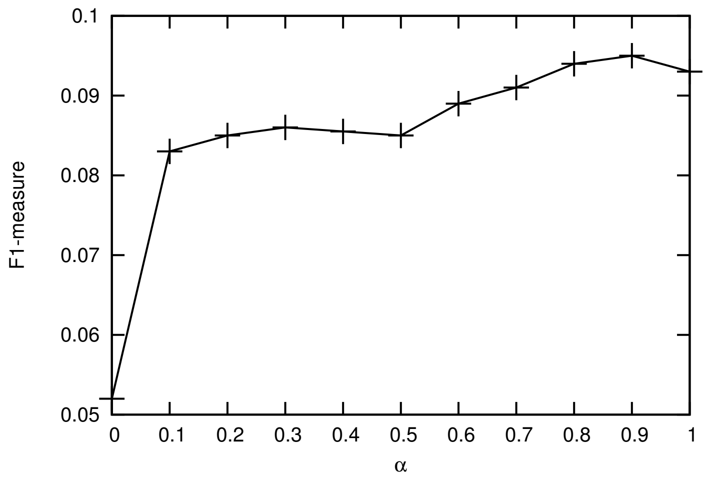


Figure 5.5: ROUGE-2 F1-measure: Impact of α on the Yago-based Summarizer performance. γ=0.3. DUC'04 collections.

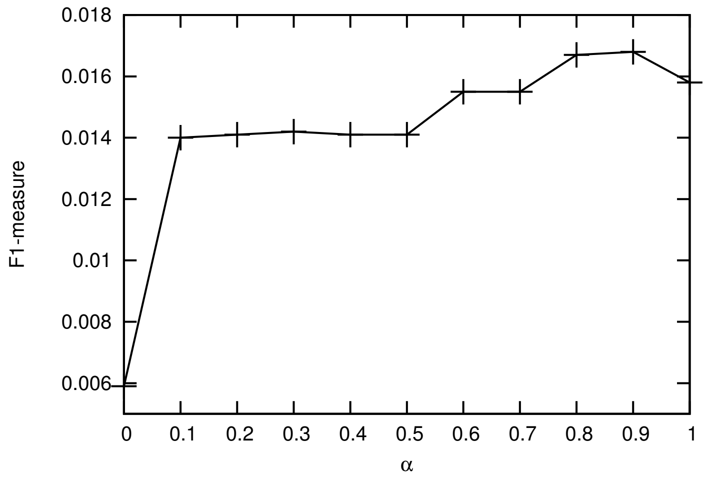


Figure 5.6: ROUGE-4 F1-measure: Impact of α on the Yago-based Summarizer performance. γ=0.3. DUC'04 collections.

results were achieved by setting γ=0.3. Furthermore, the results remain relatively stable when γ ranges between 0.2 and 0.5. Since the EntityRank score of many of the recognized entities fall in the same value range, it means that redoubling the score of the recognized named entities with respect to the time/date/number entities yields good summarization performance. In contrast, setting γ out of the above value range implies giving an under- or over-emphasis to the least interesting entities.

The α parameter allows the user to decide to which extent the similarity between the already selected sentence is relevant compared to the entity-based sentence rank for sentence selection. The higher the value of α is, the more important the ontology-based sentence rank becomes with respect to the similarity with the previously selected sentences (see Formula 5.4). Since the analyzed documents contain a limited amount of redundancy, Yago-based Summarizer achieves averagely high ROUGE scores by setting high α values (i.e., $\alpha > 0.7$). With the DUC'04 collections, the best performance results were achieved by setting α=0.9. Note that, with such configuration setting, Yago-based Summarizer disregards, to a large extent, the impact of the similarity score with respect to the ontology-based sentence ranking. However, when coping with document collections that contain a larger number of repetitions, the user should set lower α values in order to achieve a good trade-off between summary relevance and redundancy.

Chapter 6

SocioNewSum

This chapter presents a multi-document summarization model i.e., SociONewSum. Introduction is given in section 6.1. Section 6.2 discusses some related work. The method is presented in section 6.3 while 6.4 discusses the evaluated results.

6.1 Introduction

Since large document collections (e.g., news articles, scientific papers, blogs) are nowadays easily accessible through Web search engines, digital libraries, and online communities, Web users are commonly interested in exploring easy-to-read text summaries rather than perusing tens of potentially large documents. Multi-document summarization focuses on automatically generating concise summaries of large document collections. Text summarizers can be classified as sentence- or keyword-based. Specifically, sentence-based approaches entail partitioning document(s) into sentences and selecting the most informative ones to include in the summary [32, 59, 136, 137], whereas keyword-based approaches focus on detecting salient keywords to summarize documents using either co-occurrence measures [83] or Latent Semantic Analysis [48]. Summarizers can be further classified as query-based or generic. While query-based summaries are targeted at a specific user query, the generic summarization task entails producing a general-purpose summary that consists of a selection of most informative document sentences or keywords.

To effectively address document summarization, different data mining and information retrieval techniques have been adopted. For example, clustering techniques [82, 137] have been applied to first group document sentences into homogeneous clusters and then pick out the most representative one within each cluster, whereas graph-based approaches [105, 133] generate graphs that represent the underlying correlations between keywords or sentences. These models are then indexed by means of established graph ranking algorithms [29] to identify the most salient document content. Unfortunately, general-purpose summarization strategies hardly differentiate between relevant concepts and not within a specific knowledge domain. Hence, the generated summaries may not meet reader's expectations and interests. To address this issue, two promising research directions have recently been investigated: (i) the exploitation of advanced semantics-based models (e.g., ontologies, taxonomies) to drive the summarization process [44, 67, 143] and (ii) the integration of social data analysis steps to identify the current user interest's [145, 150]. Semantics-based approaches evaluate the document content according to established semantics-based models, such as ontologies or controlled vocabularies. Integrating ontologies into document summarizers allows us to automatically and effectively differentiate between terms having different meanings in different contexts as well as map term occurrences to their actual (non-ambiguous) concepts. The parallel analysis of the User-Generated Content (UGC) acquired from social networks and online communities can significantly improve summarizer performance [42, 114, 118, 145, 150]. For example, highlighting the current social trends [88], the subjects that are currently matter on contention on the Web [60, 90], or the context in which Web documents were published [146] can be useful for generating appealing text summaries.

Our goal is to combine the knowledge coming from semantic models and social data to improve document summarization performance. The contribution of this Chapter is twofold. Firstly, it overviews most recent research advances in document summarization and classifies the related approaches according to the adopted data mining or information retrieval strategy. Secondly, it presents SociONewSum (Social and Ontology-based News Summarizer), a novel summarizer that selects a worthwhile subset of sentences from a collection of news ranging over the same topic. Sentences are evaluated and ranked according to their relevance with respect to an ontological knowledge base, i.e., Yago [126]. To this purpose, an established Entity Recognition and Disambiguation step [68] is preliminary applied to identify most pertinent ontological concepts. Furthermore, the same concepts are also evaluated according to their importance in the textual messages posted on Twitter. The goal is to identify ontological concepts that are significant both in the news article collection and in the social network messages. Such information is then used to drive the generation of the summary of the document collection.

This Chapter describes news articles summarization by combining semantics and social knowledge. With the diffusion of online newspapers and social media, users are become capable of retrieving tens of news articles ranging over the same topic in a short time. News article summarization is the task of automatically selecting a worthwhile subset of news sentences that users could easily explore. Promising research directions in this field are the use of semantics-based models (e.g., ontologies and taxonomies) to identify key document topics and the integration of social data analysis to also consider the current user's interests during summary generation.

The chapter overviews most recent research advances in document summarization and presents a novel strategy to combine ontology-based and social knowledge for addressing the problem of generic (not query-based) multi-document summarization of news articles. To identify most salient news article's sentences, an ontology-based text analysis is performed during the summarization process. Furthermore, the social content acquired from real Twitter messages is separately analyzed to also consider the current interests of social network users for sentence evaluation.

The combination of ontological and social knowledge allows the generation of accurate and easy-to-read news summaries. Moreover, the proposed summarizer performs better than the evaluated competitors on real news articles and Twitter messages.

6.2 Related works

This section overviews the most recent summarization approaches and classifies them according to the mainly adopted data mining or information retrieval technique.

6.2.1 Clustering-based summarization

Clustering

Clustering is a well-established unsupervised data mining technique that has already been used to perform document summarization (e.g., [105, 136, 137].

The goal is to group document sentences or keywords into homogeneous clusters and then select the most authoritative representative within each group. For example, in [137] clusters represent groups of sentences from which the best representatives (e.g., the centroids or the medoids) are selected. In contrast, [106] propose a text summarizer, namely MEAD, which clusters documents instead of sentences. For each cluster, it generates a pseudo-document which consists of document terms having a high tf-idf value [129]. Then, pseudo-document sentences are ranked based on (i) the similarity to the centroids, (ii) the sentence position in the document, and (iii) the sentence length.

A parallel issue addressed by clustering-based approaches is incremental summary update. Once a document is added or removed from the collection, the task is to update the summary without regenerating the whole clustering model. The authors in [136] addressed the problem by integrating into the summarization process an incremental hierarchical clustering algorithm similar to the one previously proposed in [63]. Unlike [136], this chapter does not address the issue of summary updating. Unfortunately, clustering-based algorithms suffer from the presence of noise or outliers in the analyzed data. To make clustering algorithms robust to complex data distributions, the complexity of the mining process could relevantly increases.

6.2.2 Graph-based summarization

Graph-based strategies focus on modeling the correlations between document terms or sentences as graphs, which are then used to drive the sentence/keyword selection process [105, 135, 32, 133]. Graph indexing algorithms are commonly used to identify authoritative graph nodes. For example, the summarizer proposed in [105] ranks sentences according to the eigenvector centrality computed on the sentence linkage matrix by means of the well-known PageRank algorithm [29]. To reduce the computational complexity, many approaches perform edge pruning before executing the graph indexing algorithm. For example, [135] consider both sentence novelty and information richness to reduce the model complexity and select the most representative sentences.

A parallel effort has been devoted to integrating social knowledge into the document summarization process [150, 145]. Specifically, the authors in [150] propose to enrich the document content with social tag annotations. Furthermore, a new tag ranking algorithm is proposed and adopted to reduce noise in tags. In contrast, the summarizer presented in [145] combines the original documents with a graph-based social context description that was generated from a previous Twitter data analysis. Similar to [145], the summarizer proposed in this chapter also integrates social data coming from Twitter. [116, 117] address microblog data summarization using graph-based strategies. For example, in [116] a phrase reinforcement graph is generated to automatically analyze the structure of the sentence without the need for exploiting ontology-based models. Unlike the previously proposed approach, our approach integrates ontology-based and social models into a unified framework.

6.2.3 Summarization based on supervised techniques

Supervised data mining techniques (e.g., classification, regression) focus on predicting the values of one or more features of a given data instance based on a set of previously labeled instances. Many previous approaches adopted classification techniques to effectively address the document summarization problem [35, 75, 149, 15]. For example, in [35] a Support Vector Machine (SVM) model is used to classify document sentences as eligible or not for being included in the text summary. The work presented in [75] compares the performance of summarization techniques based on SVMs and Neural Networks. More recently, the summarizer presented in [15] assigns a rhetorical role to each sentence by means of a stochastic CRF classifier, which is trained from a collection of annotated sentences.

To summarize the textual messages posted in Twitter, the authors in [149] adopt speech act recognition, which is an established multi-class classification problem. The problem is solved by using word-based and symbol-based features that capture both the linguistic features of speech acts and the particularities of Twitter text. The recognized speech acts in tweets are then used to direct the extraction of key words and phrases to fill in templates designed for speech acts. Similarly, [85] propose to explore a variety of text sources for summarizing the Twitter topics and also perform content-based optimization for Twitter topic summarization.

6.2.4 Itemset-based summarization

Frequent itemset mining is a widely exploratory unsupervised data mining technique to discover valuable correlations among data. This technique was first introduced in [7] in the context of market basket analysis.

An appealing research issue is summarizing data by means of itemsets. Many previous approaches focus on discovering and selecting a worthwhile subset of frequent itemsets from transactional data. For example, in [28] and [72] the authors compare the observed frequency of each itemset against some null hypothesis, i.e., they measure how the itemset support diverges from its expected value. However, since this approach is static, the discovered summaries are often redundant. More recent approaches (e.g., [78, 131, 132] adopt entropy-based models to dynamically evaluate itemset interestingness. Specifically, they perform frequent itemset mining followed by post-pruning to select only a worthwhile itemset subset. A more effective and parameter-free itemset-based method for succinctly summarizing transactional data has been proposed in [87]). It exploits a novel entropy-based heuristics to solve the maximum entropy problem. Furthermore, it pushes itemset evaluation deep into the itemset mining process.

Under this category of Itemset-based summarization, our work on Item-Sum is described in chapter-4 that integrates the entropy-based itemset selection strategy proposed in [87] into the sentence-based summarization process. The problem of summarizing English-written documents is addressed by exploiting frequent itemsets. More specifically, we first exploited the strategy presented in [87]) to generate succinct yet informative itemset sets. Then, document sentences are evaluated and selected according to (i) the previously generated itemset-based model and (ii) the commonly used term frequency-inverse document frequency (tf-idf) statistic [83]. Unlike the summarizer presented in this chapter, the approach presented in chapter-4 does not rely neither on semantic-based nor on social models.

6.2.5 Summarization based on optimization strategies and Latent Semantic Analysis

Linear programming strategies have also been used to address the text summarization problem [127, 128]. The key idea is to formalize the summarization task as a min-max optimization problem. The optimal problem solution is, in our context, the subset of document sentences that maximizes a certain cost function. For instance, [56] represent sentences as sets of words and formalize the summarization task as a maximum coverage problem with Knapsack constraints. Similarly, the approaches proposed by [127, 128] search for linear programming solutions by also considering the relevance of each sentence within documents. The most significant limitation of the summarizers based on optimization strategies is the complexity of the system parameter setting, which often requires the intervention of a domain expert.

[61] first propose to adopt Latent Semantic Analysis in document summarization. Inspired by the latent semantic indexing, they applied the Singular Value Decomposition (SVD) to generic text summarization. More recently, the author in [98] and [147] the authors specialize the LSA-based analysis to Turkish and Amharic texts, respectively. A method of text summarization based on latent semantic indexing (LSI) is also proposed in [8]. It uses semantic indexing to calculate the sentence similarity and improve the accuracy of sentence similarity calculations and subject delineation.

6.2.6 Ontology-based summarization

Ontologies have already been used to improve summarization performance. For example, they have been used to (i) identify the concepts that are most pertinent to a user-specified query [77, 36], (ii) model the context in which summaries are generated in different application domains (e.g., the disaster management domain [82, 143]), and (iii) enrich existent ontological models with textual content [21].

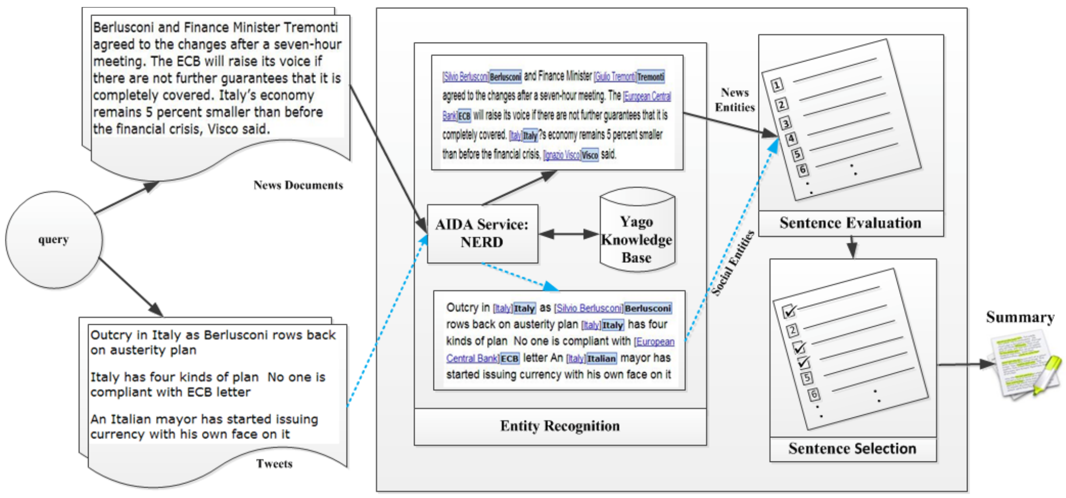
Some attempts to exploit the Wikipedia knowledge base to improve summarization performance have been made [94, 90]. In parallel, many ontology-based strategies focus on considering the text argumentative structure during the summarization process. For example, the summarizer proposed in [67] exploits a Support Vector Machine classifier to map each sentence to a subset of taxonomy nodes. Unlike all of the above-mentioned approaches, the summarizer presented in this chapter does not rely on a supervised approach.

6.3 The SociONewSum SYSTEM

SociONewSum is a novel news document summarizer that selects a worthwhile subset of news sentences by analyzing (i) the semantics behind the text, to highlight the key news concepts, and (ii) the on-topic textual messages that were published on the microblogging website Twitter, to discover the current user's interests. Figure 6.1 depicts the main architectural blocks of the proposed framework.

The summarizer takes in input a collection $D = \{d_1, ..., d_n\}$ of news documents that range over the same topic. Each document d_i consists of a set of sentences $S_i = \{S_1 i, ..., S_{ki}\}$. Beyond the original news documents, the system also analyzes the on-topic messages that were published on Twitter to discover and exploit the current interests of social network users. The Twitter content is modeled a single textual document P, which contains a set $P = \{p_1, ..., p_u\}$ of sentences. For sake of brevity, P is denoted for *social content* throughout the chapter.

The analyzed news documents were crawled by submitting an (analyst-provided) query to the Google News search engine. In parallel, the Twitter Application Programming Interfaces (APIs) were used to retrieve the corresponding set of on-topic Twitter messages (i.e., the tweets). To produce the summaries, both news and social content are processed off-line by the SociONewSum system.


Berlusconi and Finance Minister Tremonti agreed to the changes after a seven-hour meeting. The ECB will raise its voice if there are not further guarantees that it is completely covered. Italy's economy remains 5 percent smaller than before the financial crisis, Visco said.

Outcry in Italy as Berlusconi rows back on austerity plan

Italy has four kinds of plan No one is compliant with ECB letter

An Italian mayor has started issuing currency with his own face on it

[Silvio Berlusconi]Berlusconi and Finance Minister [Giulio Tremonti]Tremonti agreed to the changes after a seven-hour meeting. The [European Central Bank]ECB will raise its voice if there are not further guarantees that it is completely covered. [Italy]Italy?s economy remains 5 percent smaller than before the financial crisis, [Ignazio Visco]Visco said.

Outcry in [Italy]Italy as [Silvio Berlusconi]Berlusconi rows back on austerity plan [Italy]Italy has four kinds of plan No one is compliant with [European Central Bank]ECB letter An [Italy]Italian mayor has started issuing currency with his own face on it

Figure 6.1: The SociONewSum architecture.

The summarizer performs a three-step process. A short description of each step is given below.

- **Entity Recognition.** News documents are processed by means of an established Entity Recognition and Disambiguation step, which maps the most significant combinations of news words to non-ambiguous ontological concepts, namely the *news entities*. Furthermore, to consider the social trends about the news, the Twitter UGC is also analyzed to discover the *social entities* of topical interest.

- **Sentence evaluation.** To evaluate news sentence relevance, each sentence is mapped to a set of news and social entities. Furthermore, a sentence relevance score is assigned to each sentence according to the importance of the corresponding news and social entities.

- **Sentence selection.** To generate highly informative news summaries, sentences are evaluated and ranked according to the significance of the contained news and social entities. To select a worthwhile subset of document sentences, a Maximal Marginal Relevance strategy is adopted to pick out the subset of top-ranked news sentences with minimal content overlapping.

In the following the main characteristics of each step are thoroughly described.

6.3.1 Entity Recognition

This step focuses on discovering most relevant ontological concepts, called *entities*, which are contained either in the analyzed news collection or in the social content. This information is deemed to be worthy for summarization purposes, because an accurate news summary is expected to cover (i) all the relevant news facets and (ii) the most significant social content information. Note that, since news articles and Twitter posts range over the same topic, the two data sources are expected to share most relevant ontological concepts.

SociONewSum performs an Entity Recognition and Disambiguation step based on the Yago ontological knowledge base [126] from news articles and Twitter posts. Yago is a recently proposed ontological model, whose information was extracted from the Wikipedia free encyclopedia (www.wikipedia.org). Yago covers a broad range of shared concepts (e.g., more than 10 million entities and more than 120 million facts about these entities). To interface with Yago, SociONewSum exploits the APIs provided by the AIDA Web Service [68].

Consider a sentence composed of possibly repeated words $w_1, w_2, ..., w_z$ that belongs either to news documents or to the social content. The Entity Recognition and Disambiguation step maps sentence words to ontological entities. Specifically, an entity could be associated with either a single word or with a combination of words. In our context, we consider names, numbers, times, and dates that occur in each sentence and we associate with each of them the corresponding entity (if any). Entity recognition for times, dates, and numbers is performed matching the word combinations with a list of regular expressions. Such procedure returns at most one single entity per word combination. For example, the expression "10 p.m." corresponds to "22:00:00", according to the standard timing notation. Conversely, named entity recognition searches for Yago entities that fit the considered words. Unfortunately, a name is frequently mapped to more than one candidate entities, because natural language expressions commonly have different meanings in different contexts. For this reason, the summarizer comprises a disambiguation step which selects, among the candidate named entities for a given word combination, the most appropriate one. For example, according to the context of use, the word "Mercury" could be mapped to the planet of the solar system, i.e., the entity Mercury(Planet), or to the chemical element, i.e., Mercury(Element).

The disambiguation process considers the following properties of a Yago entity:

- the popularity score, which indicates the frequency of usage of the entity in the Wikipedia encyclopedia,
- the set of related keywords, which describe the context of use for the given entity (e.g., {*"Chemistry″*, *"Thermometer″*} for the entity Mercury(Element)), and
- the numbers of incoming and outcoming Wikipedia links, which represent the entity relations with the other Wikipedia resources and, thus, measures the authority of the entity in the knowledge base.

Let $c_i \epsilon C$ be an arbitrary document collection (either a news document $d_i \epsilon D$ or a Twitter post P). According to the above properties, we assign a rank to an arbitrary entity ne_q in c_i, which indicates the relevance of the entity with respect to the analyzed document. Its expression is given as follows:

$$ER(ne_q, c_i) = \begin{cases} \theta \cdot popularity(ne_q) + \phi \cdot sim(cxt(ne_q), cxt(c_i)) + (1 - \theta - \phi) \end{cases}$$
$$\cdot coh(ne_q, C) \text{if } ne_q \text{ named entity } \delta \text{ otherwise}$$

where θ ,ϕ and δ are user-specified parameters that take value between 0 and 1. $popularity(ne_q)$ is the popularity score of the entity and measures the global relevance of the entity in the knowledge base. $sim(cxt(ne_q), cxt(c_i))$ is the similarity between the context of use of the entity ne_q and the document c_i; it measures the contextual pertinence of the entity to the news document d_i, and it is computed by the cosine similarity between the set of related keywords of the entity and the document [129]. Finally, $coh(ne_q, C)$ is the coherence of the entity with respect to all the other recognized entities for any word in the collection. When coping with a single document c_i rather than a collection of documents C (e.g.., for the Twitter posts), both similarity and coherence measures are evaluated on c_i. By following the indications reported in [68], we set the values of θ and ϕ to 0.34 and 0.47, respectively. A thorough assessment of the entity recognition system performance on real data is also reported in [64]. Non-named entities (i.e., numbers, times, and dates) take a fixed relevance value δ, which indicates their relative importance in the collection.

The disambiguation process for names selects the top-ranked recognized named entity according to the previously introduced entity ranking. Although the relevance scores for non-named entities are not useful for disambiguation purposes, they will be used in the following steps to evaluate news sentence relevance, as discussed in the following section.

6.3.2 Sentence evaluation

To generate the summary, each sentence of the news collection is first evaluated and ranked according to the relevance of its contained entities.

To achieve this goal, each news document sentence $S_{ji} \epsilon d_i \epsilon D$ is modeled as the corresponding set E_{ji} of assigned entities. Specifically, E_{ji} contains all the entities that are mapped to any word combination in S_{ji}, which hereafter will be denoted as *news entities*. Furthermore, to evaluate sentence relevance with respect to the current social network user's interest, the Twitter UGC is modeled as the set E_P of entities, hereafter denoted as *social entities*, which are mapped to any sentence of the social content P.

Sentences are evaluated based on the relative importance (rank) of their contained news and social entities. A sentence rank $SR(S_{ji})$ is associated with each sentence $S_{ji} \epsilon d_i \epsilon D$. Its expression is given by:

$$SR(S_{ji}) = \frac{\sum_{NE_q \epsilon E_i | S_{ji} \epsilon d_i \epsilon D} ER(NE_q, d_i))}{\mid \{NE_q \epsilon E_i | S_{ji} \epsilon d_i \epsilon D\} \mid}$$

where the first multiplication term measures the relevance of the sentence in the news collection and is expressed by the average entity rank for the news entities associated with any word combination in S_{ji}, whereas the second multiplication term indicates the relevance of the sentence with respect to the Twitter UGC. The latter term is expressed by the cosine similarity between the subset of social entities that are associated with any word combination in S_{ji} and the subset of top-K social entities in P (where K is a user-specified parameter). Note that sentences that contain many recognized news and social entities on average ranked first, whereas sentences that do not cover any significant news document topics or any potentially relevant social content facet are penalized.

6.3.3 Sentence selection

This step selects a worthwhile subset of top-ranked news sentences with minimal content overlapping. Specifically, SociONewSum adopts a variant of the established iterative re-ranking strategy, called Maximal Marginal Relevance (MMR) [31], which has first been introduced in the context of query-based summary generation.

At each algorithm iteration, the MMR-based strategy picks out the candidate sentence that is characterized by (i) maximal relevance score and (ii) minimal similarity with respect to the previously selected sentences. The maximization function can be formulated as follows:

$$maximize_{\{S_{ji}\}} \alpha \cdot SR(S_{ji}) - (1-\alpha) \cdot sim(S_{ji}, S_{wo})$$

where $\alpha \epsilon [0,1]$ is a user-specified parameter that weighs the impact of each summation term, S_{ji} is an arbitrary sentence not yet included in the summary, and S_{wo} is an arbitrary sentence already contained in the summary. A thorough analysis of the impact of the parameters K, δ, and α on the summarizer performance is given in Section "Parameter analysis".

6.4 EXPERIMENTS

We performed a large number of experiments to evaluate: (i) the performance of SociONewSum on real-life news articles using the ROUGE toolkit [83], (ii) the readability of the generated summaries, and (iii) the impact of the input parameters on the summarizer performance. We analyzed six different real-life news article collections. Each collection ranges over a different topic. The list of analyzed topics is given below:

- **Irene:** The Irene hurricane beats down on the U.S. East Coast

- **Apple:** Steve Jobs resignation announcement from his role as Apple's CEO
- **UK_riots:** The U.K. suffered widespread rioting, looting and arson in August 2011
- **US_Open:** The U.S. Open tennis tournament held in New York City (USA) in August 2011
- **Debt_crisis:** The ongoing crisis of the European sovereign debt
- **Terrorism:** The fight against religious terrorism conducted by the U.S.A.

For each considered topic we collected the top-ranked news that were retrieved in August 2011 by the Google News Web search engine (http://news.google.com). Among them, we selected the top-10 news that (i) were retrieved from the most authoritative newspapers (e.g., The Guardian, BBC, Reuters, New York Post, etc.) and (ii) have length between 600 and 2,000 words. We also retrieved and analyzed the on-topic messages that were posted on Twitter (www.twitter.com) during the same time period in which news articles were published. For Twitter data retrieval, we used the Twitter Search Application Programming Interface (API) by specifying the same news queries. The considered tweet collections consist of approximately 800 tweets each. Both the news articles and the on-topic Twitter posts are available for research purposes, upon request to the authors.

6.4.1 Performance evaluation

We compared the performance of our summarizer with that of: (i) two widely used open source summarizers, i.e., the Open Text Summarizer (OTS) [112] and TexLexAn [109], and (ii) a baseline version of SociONewSum, namely Baseline, which does not consider the social content for sentence evaluation, i.e., the sentence rank depends solely on the average news entity rank. To compare SociONewSum with other approaches, we used the ROUGE [83] toolkit (version 1.5.5). ROUGE was the reference performance evaluation system[1] for the Document Understanding Conference (DUC) and the Text Analysis Conference (TAC). ROUGE measures the quality of a candidate summary by counting the unit overlaps between the summary and a set of

[1]The provided command is: ROUGE-1.5.5.pl -e data -x -m -2 4 -u -c 95 -r 1000 -n 4 -f A -p 0.5 -t 0 -d -a.

reference summaries, which were provided by the contest organizers. Intuitively, the summarizer that achieves the highest ROUGE scores can be considered to be the most effective one. ROUGE implements several evaluation scores (e.g., ROUGE-1, ROUGE-2, ROUGE-SU4) and measures (e.g., precision, recall, F1-measure). For the sake of brevity, we only reported, as representative scores, ROUGE-1 and ROUGE-SU4. Analogous results were achieved for the other scores. Since reference summaries (i.e., the optimal news document summaries) are not available for the news document collections, we performed, similar to [37], a leave-one-out cross validation. More specifically, for each news collection we summarized nine out of ten news documents and we compared the produced summary with the remaining (not yet considered) one, which is selected as reference summary. Next, we tested all the other possible combinations by varying the reference summary and, for each summarizer, we computed the average performance results, in terms of precision (P), Recall (R), and F1-measure (F1), for both the ROUGE-1 and ROUGE-SU4 evaluation scores. Since we specifically cope with news article ranging over the same topic, the assumption that a new document is a representative summary of all the other documents in the news collection is acceptable. Tables 1 and 2 summarize the results achieved by SociONewSum and its evaluated competitors in terms of ROUGE-1 and ROUGE-SU4 precision (Pr), recall (R), and F1-measure (F1). For OTS, TexLexAn, Baseline we reported the results that were achieved by setting the averagely best configuration setting (the same for all news collections). For SociONewSum we reported the results achieved by both a standard configuration, for which parameter values are fixed for all news collections, and a tuned configuration. For the standard configuration we set K to 10, δ to 0.2 and α to 0.7. For each collection the tuned configuration settings are reported in Tables 1 and 2. A detailed analysis of the impact of each parameter on the summarizer performance is given in Section "Parameter analysis".

Dataset (Tuned setting)	**TexLexAn**			**OTS**			**Baseline**			**SocioNewSum standard result (Tuned result)**		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1
Debt Crisis (K=50,δ=0.1,α=0.6)	0.068	0.541	0.121	0.077	**0.581**	0.135	0.070	0.518	0.121	**0.081*** **(0.082*)**	0.569 **(0.582)**	**0.141*** **(0.143*)**
Irene (K=30,δ=0.1,α=0.2)	0.062	0.577	0.110	0.064	0.602	0.116	0.055	0.532	0.099	**0.066*** **(0.070*)**	**0.612*** **(0.646*)**	**0.119*** **(0.127*)**
Steve Jobs (K=40,δ=0.2,α=0.8)	0.057	0.533	0.102	0.060	0.573	0.109	0.068	0.615	0.122	**0.072*** **(0.072*)**	**0.625*** **(0.645*)**	**0.126*** **(0.128*)**
Terrorism (K=70,δ=0.1,α=0.7)	0.047	0.447	0.086	0.052	0.479	0.093	0.045	0.444	0.082	**0.055*** **(0.058*)**	**0.517*** **(0.544*)**	**0.099*** **(0.104*)**
UK riots (K=10,δ=0.1,α=0.2)	0.060	0.522	0.107	**0.065**	**0.586**	**0.117**	0.052	0.456	0.092	**0.065** **(0.067)**	0.578 (0.578)	**0.117** **(0.120*)**
US Open (K=40,δ=0.1,α=0.5)	0.076	0.651	0.136	0.065	0.545	0.116	0.065	0.492	0.114	**0.080*** **(0.082*)**	**0.654** **(0.663*)**	**0.142*** **(0.145*)**

Table 6.1: Performance comparison in terms of ROUGE-1.

Dataset (Tuned setting)	**TexLexAn**			**OTS**			**Baseline**			**SocioNewSum standard result (Tuned result)**		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1
Debt Crisis (K=50,δ=0.1,α=0.6)	0.021	0.175	0.038	0.024	**0.188**	0.042	0.019	0.147	0.033	**0.026*** **(0.027*)**	0.187 **(0.199*)**	**0.045*** **(0.048*)**
Irene (K=30,δ=0.1,α=0.2)	0.021	0.210	0.039	0.022	0.209	0.039	0.017	0.173	0.031	**0.023** **(0.024*)**	**0.221*** **(0.230*)**	**0.041** **(0.043*)**
Steve Jobs (K=40,δ=0.2,α=0.8)	0.018	0.173	0.032	0.021	0.203	0.037	0.024	0.225	0.043	**0.025** **(0.026)**	**0.232*** **(0.244*)**	**0.045*** **(0.047*)**
Terrorism (K=70,δ=0.1,α=0.7)	0.015	0.147	0.028	**0.017**	**0.161**	**0.031**	0.014	0.142	0.025	0.014 **(0.019*)**	0.142 **(0.188*)**	0.025 **(0.035*)**
UK riots (K=10,δ=0.1,α=0.2)	0.022	0.195	0.039	0.022	0.203	0.040	0.015	0.139	0.027	**0.023** **(0.024)**	**0.211*** **(0.211*)**	**0.043*** **(0.043*)**
US Open (K=40,δ=0.1,α=0.5)	0.025	0.221	0.045	0.021	0.182	0.038	0.025	0.198	0.044	**0.032*** **(0.034*)**	**0.264*** **(0.278*)**	**0.056*** **(0.059*)**

Table 6.2: Performance comparison in terms of ROUGE-SU4.

Consider, as representative measure, the F1-measure, which is expressed by the harmonic average between precision and recall. SociONewSum Standard performs best on 6 out of 6 and 5 out of 6 collections in terms of ROUGE-1 and ROUGE-SU4, respectively. Furthermore, with the tuned configuration SociONewSum outperforms all the other tested summarizers in terms of ROUGE-1 and ROUGE-SU4 F1-measure for every tested collection. Similar results were achieved for the other measures (i.e., precision and recall).

To validate the statistical significance of the achieved performance improvements, we performed the paired t-test of statistical significance [47] by setting the significance level to 95% for all the evaluated datasets and measures. Significant improvements between SociONewSum and all the other approaches for every news collection are starred in Tables 1 and 2. The results confirm the significance of the achieved performance improvements for most evaluated datasets and measures (e.g., with the standard configuration, for ROUGE-1 F1-measure 6 out of 6 against Baseline and TexLexAn, 5 out of 6 against OTS, for ROUGE-SU4 F1-measure 5 out of 6 against Baseline, 4 out 6 against OTS and TexLexAn).

In light of the above-mentioned results, the proposed approach appears to be, on average, more accurate than state-of-the-art approaches. Furthermore, based on the results of the comparison with Baseline, the integration of social data analysis into the summarization process is shown to relevantly improve the effectiveness of the proposed summarizer.

6.4.2 Summary comparison

To validate the readability and soundness of the news summaries, we compared the results that were achieved by SociONewSum with those produced by OTS TexLexAn, and Baseline. Table 3 reports the top-3 summary sentences selected by SociONewSum and the other competitors for a representative news collection, i.e., Irene Hurricane.

The summary produced by SociONewSum appears to be sound and properly focused on the news topic. It relates the destructive tropical hurricane Irene, including the reaction of the U.S. government to the disaster. In contrast, Baseline's, OTS's and TexLexAn's summaries contain less interesting or too much detailed information (e.g., the situation in Vermont).

Since the tweets related to the Irene Hurricane collection report the announcements made by the White House about the disaster, SociONewSum deemed the message sent by Barack Obama to be worthy for summarization purposes. Conversely, Baseline ignores the Twitter user's interests and, thus, completely disregards the official U.S. government announcements.

6.4.3 Parameter analysis

We also analyzed the impact of the main input parameters on the summarizer performance. Specifically, we considered the SociONewSum standard configuration (K=10, δ=0.2, α=0.7) and we varied the value of each of the input parameter separately while keeping the value of all the other ones constant.

In Figure 6.2 we plotted the variation of the representative ROUGE-1 F1-measure by varying the value of α in the range [0,1] for a representative news collection (i.e., U.S. Open). The parameter α weighs the importance of the sentence rank with respect to the similarity score evaluated between the candidate sentence and the already selected ones (see Section "Sentence evaluation and selection"). The higher the value of α is, the more significant the selected sentence rank is. However, since the impact of the similarity score decreases, potentially redundant information could be selected as well. According to the achieved results, the best performance results were achieved by setting medium values of α, i.e.,. Hence, a good trade-off between sentence relevance and novelty is required.

The parameter δ appears to only slightly affect the summarizer performance (see Figure 6.3). In most cases, setting a relatively low values of δ (e.g.,

Summarizer	Result
SociONewSum	It's one of several towns in states such as New Jersey, Connecticut, New York, Vermont and Massachusetts dealing with the damage of torrential rain and flooding spawned by Hurricane Irene (Cleveland.com). Barack Obama, US president, has called Irene a "historic hurricane" and declared a state of emergency in New York, ordering federal aid to supplement state and local response efforts starting on Friday. Beyond deadly flooding that caused havoc in upstate New York and Vermont, the hurricane flooded cotton and tobacco crops in North Carolina, temporarily halted shellfish harvesting in Chesapeake Bay, sapped power and kept commuters from their jobs.
OTS	As emergency airlift operations brought ready-to-eat meals and water to Vermont residents left isolated and desperate, states along the Eastern Seaboard continued to be battered Tuesday by the after effects of Irene, the destructive hurricane turned tropical storm. Dangerously-damaged infrastructure, 2.5 million people without power and thousands of water-logged homes and businesses continued to overshadow the lives of residents and officials from North Carolina through New England, where the storm has been blamed for at least 44 deaths in 13 states. But new dangers developed in New Jersey and Connecticut, where once benign rivers rose menacingly high.
TexLexAn	As emergency airlift operations brought ready-to-eat meals and water to Vermont residents left isolated and desperate, states along the Eastern Seaboard continued to be battered Tuesday by the after effects of Irene, the destructive hurricane turned tropical storm. Search-and-rescue teams in Paterson have pulled nearly 600 people from flooded homes in the town after the Passaic River rose more than 13 feet above flood stage, the highest level since 1903. It's one of several towns in states such as New Jersey, Connecticut, New York, Vermont and Massachusetts dealing with the damage of torrential rain and flooding spawned by Hurricane Irene (Cleveland hurricane Katrina was a Category 5 hurricane and Hurricane Irene was a Category 1 hurricane).
Baseline	Stransky pointed out that, "because Irene tracked west of several islands including Abaco Island, which contains the third highest insured property value after New Providence and Grand Bahama it is likely that losses in the Bahamas from Hurricane Irene will be higher than those from Hurricane Floyd". US Geological Survey scientists are sampling water from six major rivers in the Northeast from the Susquehanna, which flows out of New York into the Chesapeake Bay, to Boston's Charles River for sewage contamination. In Vermont, Craig Fugate, head of the Federal Emergency Management Agency, joined the state's governor, Peter Shumlin (D).

Table 6.3: Summary examples. Irene hurricane news collection.

0.2) yields good summarization performance. With such parameter setting, sentences that do not contain any recognized named entity are mildly penalized, whereas sentences that contain several named entities usually achieve fairly high ranks. From a practical point of view, this is reasonable because names are most likely to contain meaningful information for summarization purposes. In contrast, higher δ values on average worsens the quality of the result, because the scores assigned to different entity types become unevenly distributed. Since for all the analyzed collections tuning the value of δ does not produce significant performance improvements, the use of the standard parameter value (i.e., δ=0.2) is recommended.

In Figure 6.4 we separately analyzed the impact of the parameter K on the summarizer performance. During sentence selection, K represents the number of considered top-ranked news entities (see Section "Sentence selec-

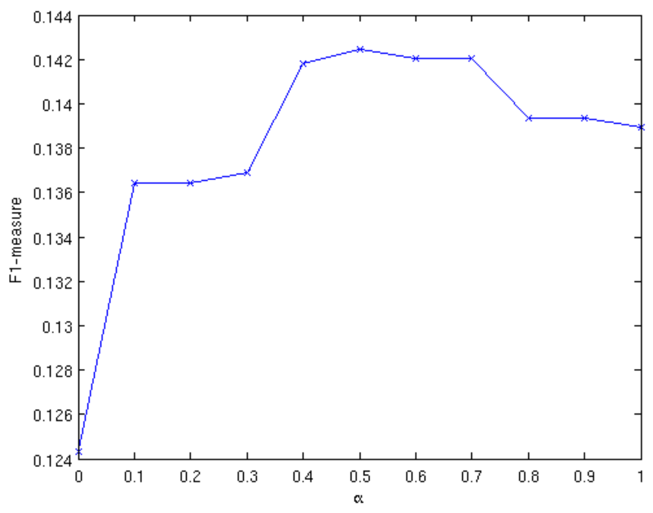


Figure 6.2: Impact of α on the Rouge-1 F1-measure. U.S. Open.

tion"). The higher the value of K is, the more likely the occurrence of a news entity in a candidate sentence becomes. However, the more news entities are considered, the lower, on average, the similarity scores becomes, because a higher number of unmatched social entities is averagely considered. On the other hand, some of the low-ranked social entities may represent noisy information and, thus, should be discarded. Hence, to achieve good summarization performance medium values of K (i.e., between 10 and 50) should be enforced. The best configuration setting actually depends on the analyzed data distribution.

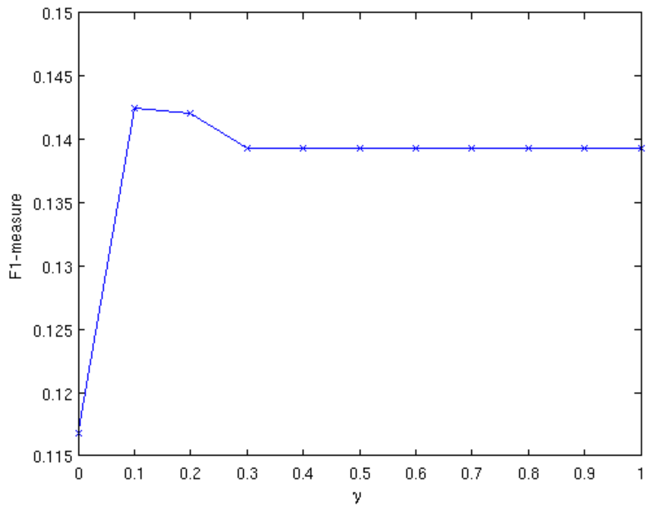


Figure 6.3: Impact of δ on the Rouge-1 F1-measure. U.S. Open.

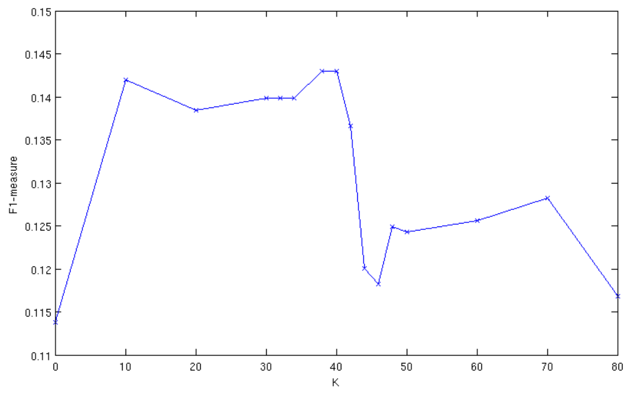


Figure 6.4: Impact of K on the Rouge-1 F1-measure. U.S. Open.

Chapter 7

Conclusions

Data mining aims to detect patterns in numeric data in electronic documents but important information is very often exist in the form of text. Unlike numeric data, it is difficult to deal with text as it is often amorphous. Text mining is a suite of analysis tools for multiple textual documents that involves extraction of key phrases, concepts etc. and text is processed to prepare it in such a form that becomes suitable for analysis by numeric data mining techniques. Data mining goals are to extract features by identifying frequently used terms and concepts from input document collections and to discover any associations among features e.g., associations existing between symptoms of patients etc. Thus, "coding" the information in the input text is the primary step to text mining and then relations among features are determined by association rules algorithms. Working in the same direction, a proposed multi-document summarizer combines the knowledge provided by an itemset-based model with a statistical evaluator, based on tf-idf statistics, to select the most representative and not redundant sentences. This approach is the first attempt to exploit frequent itemsets in text summarization. Experiments, conducted on DUC'04 document collection show the effectiveness of the proposed approach. Future works of this methodology will address: (i) the extension of the proposed approach to address the problem of incremental summary updating where It will be focused to generate updated summaries in real time with the arrival of new documents (news etc.), and (ii) the application of text disambiguation techniques to improve the summarization performance. Any proficient knowledgebase can be used such as Yago ontology for identifying and disambiguating the important concepts (entities) of the text.

In recent years, semantics-based document analysis has shown to improve the performance of document summarization systems significantly. However, since most of the related approaches perform semantics-based analysis as a preprocessing step rather than integrating the ontological knowledge into the summarization process, the quality of the generated summaries remains, in some cases, unsatisfactory. YagoSum aims to improve the performance of state-of-the-art summarizers by integrating an ontology-based sentence evaluation and selection step into the summarization process. Specifically, an established entity recognition and disambiguation step based on the Yago ontology is used to identify the key document concepts and evaluate their significance with respect to the document context. The same results are then exploited to select the most representative document sentences. The experimental results show that Yago-based Summarizer performs better than many state-of-the-art summarizers on benchmark collections. Furthermore, a qualitative comparison between the summaries generated by Yago-based Summarizer and the state-of-the-art summarizers demonstrate the usefulness and applicability of the proposed approach. Future developments on this work will address the application of the proposed summarization approach to multilingual document collections requiring multi-lingual knowledgebases and the use of entropy-based sentence selection strategies to further improve the compactness of the generated summaries. This semantic approach of text summarization can also be extended to the real time updating text summaries in incremental manner. However due to its computational cost, it will be interesting to work on the feasibility of this approach.

A novel text summarizer *SocioNewSum* focuses on extracting accurate and appealing summaries of collections of news articles ranging over the same topic. The proposed summarizer combines the use of an established semantics-based model with the analysis of the on-topic textual message published on Twitter. To identify the key news topics and also consider the current interests of social network users, news sentences are evaluated and selected according to the importance of their contained ontological concepts in the news collection and in the Twitter UGC. The experiments, conducted on real-life news article collections and driven by on-topic Twitter posts, demonstrate the effectiveness of the proposed approach compared to other open source summarizers and the usefulness of social content for improving the summarization performance. Furthermore, the readability of the generated summaries has also been evaluated on real-life data. Research on semantics-based news document summarization can be extended in different directions. For example, itemset-based approaches (e.g., [19]) have recently shown fairly good performance on generic documents. Hence, the integration of ontologies with itemset-based models is a promising research direction to improve news summarization performance. A parallel issue concerns the analysis of the evolution of news articles over time to address incremental news summary updating. For example, once a set of news articles is added to the original collection, the text summary should be updated without the need for calculating the whole data mining model. Furthermore, since SociONewSum also analyzes the Twitter UGC to generate accurate and up-to-date news summaries, a parallel analysis of the temporal evolution of the UGC coming from social networks could further improve the performance of the proposed summarizer. Finally, we aim to integrate the proposed approach in a real-world content curation platform, in which users could select a subset of news topics of interest, create and personalize their virtual newspaper, and access to the news summaries instead of the whole article content.

List of Figures

List of Tables

Bibliography

[1] alchemyapi. http://www.alchemyapi.com/.

[2] Apache stanbol enhancer. http://dev.iks-project.eu:8081/enhancer/.

[3] cicerolite. http://demo.languagecomputer.com/cicerolite/.

[4] Poolparty extractor. www.poolparty-software.com.

[5] Zemanta. http://www.zemanta.com/blog/demo/.

[6] Charu C. Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 163–222. Springer, 2012.

[7] R. Agrawal, T. Imielinski, and Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD 1993*, pages 207–216, 1993.

[8] Dongmei Ai, Yuchao Zheng, and Dezheng Zhang. Automatic text summarization based on latent semantic indexing. *Artif. Life Robot.*, 15(1):25–29, August 2010.

[9] Rasim M. Alguliev, Ramiz M. Aliguliyev, and Makrufa S. Hajirahimova. Gendocsum + mclr: Generic document summarization based on maximum coverage and less redundancy. *Expert Syst. Appl.*, 39(16):12460–12473, 2012.

[10] Rasim M. Alguliev, Ramiz M. Aliguliyev, and Nijat R. Isazade. Cdds: Constraint-driven document summarization models. *Expert Syst. Appl.*, 40(2):458–465, 2013.

[11] Rasim M. Alguliev, Ramiz M. Aliguliyev, and Nijat R. Isazade. Multiple documents summarization based on evolutionary optimization algorithm. *Expert Syst. Appl.*, 40(5):1675–1689, April 2013.

[12] Robert Bart Stephen Soderland Anthony Fader, Michael Schmitz and Oren Etzioni. Reverb open information extraction software. http://reverb.cs.washington.edu.

[13] Chidanand Apté, Fred Damerau, and Sholom M. Weiss. Automated learning of decision rules for text categorization. *ACM Trans. Inf. Syst.*, 12(3):233–251, July 1994.

[14] Sanjeev Arora, Rong Ge, and Ankur Moitra. Learning topic models - going beyond svd. In *FOCS*, pages 1–10. IEEE Computer Society, 2012.

[15] R. Atkinson, J. Munoz. Rhetorics-based multi-document summarization. *Expert Systems with Applications*, 2013.

[16] Rene Speck Axel-Cyrille Ngonga Ngomo, Norman Heino and Stanley Hillner. Federated knowledge extraction for semantic web applications.

[17] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The description logic handbook: theory, implementation, and applications.* Cambridge University Press, New York, NY, USA, 2003.

[18] E. Baralis, G. Bruno, and A. Fiori. Minimum number of genes for microarray feature selection. *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC-08)*, pages 5692–5695, 2008.

[19] Elena Baralis, Luca Cagliero, Alessandro Fiori, and Saima Jabeen. Multi-document summarization exploiting frequent itemsets. In *ACM Symposium on Applied Computing (SAC 2012)*, 2012.

[20] Elena Baralis and Alessandro Fiori. Summarizing biological literature with biosumm. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1961–1962, New York, NY, USA, 2010. ACM.

[21] Klimt B. Grobelnik M. Schneider D. Witbrock M. Mladenic D. Baxter, D. Capturing document semantics for ontology generation and document summarization. *Semantic Knowledge Management: Integrating Ontology Management, Knowledge Discovery, and Human Language Technologies*, pages 141–154, 2009.

[22] Moty Ben-Dov and Ronen Feldman. Text mining and information extraction. In Oded Maimon and Lior Rokach, editors, *The Data Mining and Knowledge Discovery Handbook*, pages 801–831. Springer, 2005.

[23] Mine Berker and Tunga Gungor. Using genetic algorithms with lexical chains for automatic text summarization. In Joaquim Filipe and Ana L. N. Fred, editors, *ICAART (1)*, pages 595–600. SciTePress, 2012.

[24] Michael W. Berry and Murray Browne. Email surveillance using non-negative matrix factorization. *Comput. Math. Organ. Theory*, 11(3):249–264, 2005.

[25] Smriti Bhagat, Graham Cormode, and S. Muthukrishnan. Node classification in social networks. In Charu C. Aggarwal, editor, *Social Network Data Analytics*, pages 115–148. Springer, 2011.

[26] Deepankar Bharadwaj and Suneet Shukla. Article: Text mining technique using genetic algorithm. *IJCA Proceedings on International Conference on Advances in Computer Application 2013*, ICACA 2013:7–10, February 2013.

[27] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. O'Reilly Media, 2009.

[28] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 265–276, 1997.

[29] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.

[30] B. Sushma B.V.Rama Krishna. Novel approach to museums development & emergence of text mining. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, 2.

[31] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM.

[32] Giuseppe Carenini, Raymond T. Ng, and Xiaodong Zhou. Summarizing email conversations with clue words. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 91–100, New York, NY, USA, 2007. ACM.

[33] Vitor R. Carvalho and William W. Cohen. On the collective classification of email "speech acts". In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in*

information retrieval, SIGIR '05, pages 345–352, New York, NY, USA, 2005. ACM.

[34] Soumen Chakrabarti, Byron E. Dom, Rakesh Agrawal, and Prabhakar Raghavan. Using taxonomy, discriminants, and signatures for navigating in text databases. In Matthias Jarke, Michael J. Carey, Klaus R. Dittrich, Frederick H. Lochovsky, Pericles Loucopoulos, and Manfred A. Jeusfeld, editors, *Proceedings of VLDB-97, 23rd International Conference on Very Large Data Bases*, pages 446–455, Athens, GR, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[35] Yllias Chali, Sadid A. Hasan, and Shafiq R. Joty. A svm-based ensemble approach to multi-document summarization. In *Proceedings of the 22nd Canadian Conference on Artificial Intelligence: Advances in Artificial Intelligence*, Canadian AI '09, pages 199–202, Berlin, Heidelberg, 2009. Springer-Verlag.

[36] Ping Chen and Rakesh M. Verma. A query-based medical information summarization system using ontology knowledge. In *CBMS*, pages 37–42. IEEE Computer Society, 2006.

[37] Li-Yeh Chuang, Cheng-Huei Yang, Jung-Chike Li, and Cheng-Hong Yang. A hybrid bpso-cga approach for gene selection and classification of microarray data. *Journal of Computational Biology*, 19(1):68–82, 2012.

[38] William Cohen. Learning rules that classify e-mail. In *In Papers from the AAAI Spring Symposium on Machine Learning in Information Access*, pages 18–25. AAAI Press, 1996.

[39] William W. Cohen. Learning trees and rules with set-valued features. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, AAAI'96, pages 709–716. AAAI Press, 1996.

[40] William W. Cohen and Haym Hirsh. Joins that generalize: Text classification using whirl. In *Fourth International Conference on Knowledge Discovery and Data Mining*, 1998.

[41] Alfio Gliozzo Concetto Elvio Bonafede, Aldo Gangemi. Semiosearch wikifier. http://stlab.istc.cnr.it/stlab/STLabWikifier.

[42] Jack G. Conrad, Jochen L. Leidner, Frank Schilder, and Ravi Kondadadi. Query-based opinion summarization for legal blog entries. In

Proceedings of the 12th International Conference on Artificial Intelligence and Law, ICAIL '09, pages 167–176, New York, NY, USA, 2009. ACM.

[43] J. Kubina J. Rankel P. O'Leary D. Conroy, J. Schlesinger. Classy 2011 at tac: Guided and multi-lingual summaries and evaluation metrics. In *TAC'11: Proceedings of the The 2011 Text Analysis Conference*, 2011.

[44] J.D. Goldstein J. O'Leary D.P. Conroy, J.M. Schlesinger. Left-brain/right-brain multi-document summarization. In *DUC 2004 Conference Proceedings*, 2004.

[45] Ido Dagan, Yael Karov, and Dan Roth. Mistake-driven learning in text categorization. In *In EMNLP-97, The Second Conference on Empirical Methods in Natural Language Processing*, pages 55–63, 1997.

[46] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.*, 10(7):1895–1923, October 1998.

[47] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.*, 10(7):1895–1923, October 1998.

[48] Mark Dredze, Hanna M. Wallach, Danny Puller, and Fernando Pereira. Generating summary keywords for emails using topics. In *Proceedings of the 13th international conference on Intelligent user interfaces*, IUI '08, pages 199–206, New York, NY, USA, 2008. ACM.

[49] H. Drucker, D. Wu, and V. Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5), 1999.

[50] DUC-HTL/NAACL. Document understanding conference. htl/naacl workshop on text summarization, 2004.

[51] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[52] Susan Dumais and Hao Chen. Hierarchical classification of web content. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 256–263, New York, NY, USA, 2000. ACM.

[53] Michel Gagnon Eric Charton. Wikimeta. http://www.wikimeta.com/wapi/semtag.pl.

[54] Ronen Feldman and Ido Dagan. Knowledge discovery in textual databases (kdt). In *In Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95*, pages 112–117. AAAI Press, 1995.

[55] Ronen Feldman and James Sanger. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, December 2006.

[56] Elena Filatova and Vasileios Hatzivassiloglou. A formal model for information selection in multi-sentence text extraction. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[57] Aldo Gangemi. A comparison of knowledge extraction tools for the semantic web. In Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *ESWC*, volume 7882 of *Lecture Notes in Computer Science*, pages 351–366. Springer, 2013.

[58] LuÃs Fernando Fortes Garcia, JosÃ© Valdeni de Lima, Stanley Loh, and JosÃ© Palazzo Moreira de Oliveira. Using ontological modeling in a context-aware summarization system to adapt text for mobile devices. In Peter P. Chen and Leah Y. Wong, editors, *Active Conceptual Modeling of Learning*, volume 4512 of *Lecture Notes in Computer Science*, pages 144–154. Springer, 2006.

[59] Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization - Volume 4*, NAACL-ANLP-AutoSum '00, pages 40–48, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[60] Qu Y. & Tian S. Gong, S. Summarization using wikipedia. In *Proceedings of Text Analysis Conference*, 2010.

[61] Yihong Gong and Xin Liu. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 19–25, New York, NY, USA, 2001. ACM.

[62] Siddharth Gopal and Yiming Yang. Multilabel classification with meta-level features. In *Proceedings of the 33rd International ACM SIGIR*

Conference on Research and Development in Information Retrieval, SIGIR '10, pages 315–322, New York, NY, USA, 2010. ACM.

[63] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams: Theory and practice. *IEEE Trans. on Knowl. and Data Eng.*, 15(3):515–528, March 2003.

[64] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. Evaluating entity linking with wikipedia. *Artif. Intell.*, 194:130–150, January 2013.

[65] Y. Nishimura T. & Takeda H. Hamasaki, M. Matsuo. Ontology extraction by collaborative tagging. In *WWW 2009*. ACM press.

[66] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[67] Leonhard Hennig, Winfried Umbrath, and Robert Wetzker. An ontology-based approach to text summarization. In *Web Intelligence/IAT Workshops*, pages 291–294. IEEE, 2008.

[68] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Furstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *EMNLP*, pages 782–792. ACL, 2011.

[69] I. Horrocks et.al. The ontology interchange language oil the grease between ontologies. Technical report, Dep. of Computer Science, Univ. of Manchester, UK/ Vrije Universiteit Amsterdam, NL/ AIdministrator, Nederland B.V./ AIFB, Univ. of Karlsruhe, DE, 2000. http://www.cs.vu.nl/˜dieter/oil/.

[70] Andreas Hotho, Andreas Nurnberger, and Gerhard Paab. A brief survey of text mining. *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, 20(1):19–62, May 2005.

[71] Simon Jackman. Generalized linear models.

[72] Szymon Jaroszewicz and Dan A. Simovici. Interestingness of frequent itemsets using bayesian networks as background knowledge. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 178–186, New York, NY, USA, 2004. ACM.

[73] Maringanti Hima Bindu Jitendra Nath Shrivastava. E-mail classification using genetic algorithm with heuristic fitness function. *International Journal of Computer Trends and Technology (IJCTT)*, 4, 2013.

[74] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[75] Keivan Kianmehr, Shang Gao, Jawad Attari, M. Mushfiqur Rahman, Kofi Akomeah, Reda Alhajj, Jon Rokne, and Ken Barker. Text summarization techniques: Svm versus neural networks. In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '09, pages 487–491, New York, NY, USA, 2009. ACM.

[76] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September 1999.

[77] Balasubramanie B. Kogilavani, A. Ontology enhanced clustering based summarization of medical documents. *International Journal of Recent Trends in Engineering*, 1(1), 2009.

[78] Kleanthis-Nikolaos Kontonasios and Tijl De Bie. An information-theoretic approach to finding informative noisy tiles in binary databases. In *SDM*, pages 153–164. SIAM, 2010.

[79] Ken Lang. Newsweeder: Learning to filter netnews. In *in Proceedings of the 12th International Machine Learning Conference (ML95*, 1995.

[80] R.Y.K. Lau, D. Song, Y. Li, T.C.H. Cheung, and J.X. Hao. Toward a fuzzy domain ontology extraction method for adaptive e-learning. *Knowledge and Data Engineering, IEEE Transactions on*, 21(6):800–813, 2009.

[81] David E. Lewis, Kimberly A. Knowles, Bob Smith, and Writes (unquoted. Threading electronic mail: A preliminary study, 1997.

[82] Lei Li, Dingding Wang, Chao Shen, and Tao Li. Ontology-enriched multi-document summarization in disaster management. In Fabio Crestani, StÃ©phane Marchand-Maillet, Hsin-Hsi Chen, Efthimis N. Efthimiadis, and Jacques Savoy, editors, *SIGIR*, pages 819–820. ACM, 2010.

[83] C. Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Language*

Technology Conference (HLT-NAACL-2003), Edmonton, Canada, May 27 - June 1 2003.

[84] Marina Litvak and Mark Last. Graph-based keyword extraction for single-document summarization. In *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*, MMIES '08, pages 17–24, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[85] Fei Liu, Yang Liu, and Fuliang Weng. Why is "sxsw" trending?: exploring multiple text sources for twitter topic summarization. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 66–75, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[86] Stanley Loh, Leandro Krug Wives, and José Palazzo M. de Oliveira. Concept-based knowledge discovery in texts extracted from the web. *SIGKDD Explor. Newsl.*, 2(1):29–39, June 2000.

[87] Michael Mampaey, Nikolaj Tatti, and Jilles Vreeken. Tell me what i need to know: succinctly summarizing data with itemsets. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 573–581, New York, NY, USA, 2011. ACM.

[88] Michael Mathioudakis and Nick Koudas. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 1155–1158, New York, NY, USA, 2010. ACM.

[89] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 1–8, New York, NY, USA, 2011. ACM.

[90] & Li C. Miao, Y. Wikisummarizer - a wikipedia-based summarization system. In *Proceedings of Text Analysis Conference*, 2010.

[91] S. Mohamed, A. Rajasekaran. Improving query-based summarization using document graphs. In *Signal Processing and Information Technology, 2006 IEEE International Symposium on*, pages 408–410.

[92] Ilaria Bordino Marc Spaniol Mohamed Amir Yosef, Johannes Hoffart and Gerhard Weikum. Aida: An online tool for accurate disambiguation of named entities in text and tables. In *In Proceedings of the 37th International Conference on Very Large Databases, VLDB, Seattle, WA, US*, 2011.

[93] Prof. Samir K. Bandyopadhyay Mrs. Sayantani Ghosh, Mr. Sudipta Roy. A tutorial review on text mining algorithms. *IJRET: International Journal of Research in Engineering and Technology*, 02.

[94] Vivi Nastase. Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In *EMNLP*, pages 763–772. ACL, 2008.

[95] Hwee Tou Ng, Wei Boon Goh, and Kok Leong Low. Feature selection, perceptron learning, and a usability case study for text categorization. *SIGIR Forum*, 31(SI):67–73, July 1997.

[96] Finn Arup Nielsen. Clustering of scientific citations in wikipedia. *CoRR*, abs/0805.1154, 2008.

[97] Finn Arup Nielsen, Daniela Balslev, and Lars Kai Hansen. Mining the posterior cingulate: segregation between memory and pain components. *NeuroImage*, 27(3):520–532, 2005.

[98] Makbule Gulcin Ozsoy, Ilyas Cicekli, and Ferda Nur Alpaslan. Text summarization of turkish texts using latent semantic analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 869–876, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[99] V. Kumar P. N. Tan, M. Steinbach. *Association Analysis: Basic Concepts and Algorithms*. 2006.

[100] Jan Paralic and Peter Bednar. Text mining for documents annotation and ontology support, 2003.

[101] Sun Park and ByungRae Cha. Query-based multi-document summarization using non-negative semantic feature and nmf clustering. In Jinhwa Kim, Dursun Delen, Jinsoo Park, Franz Ko, and Yun Ji Na, editors, *NCM (2)*, pages 609–614. IEEE Computer Society, 2008.

[102] Thair Nu Phyu. Survey of classification techniques in data mining, 2009.

[103] Mohsen Pourvali and Mohammad Saniee Abadeh. Automated text summarization base on lexicales chain and graph using of wordnet and wikipedia knowledge base. *CoRR*, abs/1203.3586, 2012.

[104] Valentina Presutti, Francesco Draicchio, and Aldo Gangemi. Knowledge extraction based on discourse representation theory and linguistic frames. In *Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management*, EKAW'12, pages 114–129, Berlin, Heidelberg, 2012. Springer-Verlag.

[105] Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:2004, 2004.

[106] Dragomir R. Radev, Hongyan Jing, Malgorzata Stys, and Daniel Tam. Centroid-based summarization of multiple documents. *Inf. Process. Manage.*, 40(6):919–938, November 2004.

[107] Hema Raghavan and James Allan. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 79–86, New York, NY, USA, 2007. ACM.

[108] T. Ralphs and M. Guzelsoy. The SYMPHONY callable library for mixed integer programming. *The Next Wave in Computing, Optimization, and Decision Technologies*, 29:61–76, 2006. Software available at `http://www.coin-or.org/SYMPHONY`.

[109] Jean-Pierre Redonnet. Texlexan: An open-source text summarizer, 2011.

[110] Thomson Reuters. Open calais, January 2008. http://viewer.opencalais.com/.

[111] Giuseppe Rizzo and Raphaël Troncy. NERD: A framework for evaluating named entity recognition tools in the Web of data. In *ISWC 2011, 10th International Semantic Web Conference, October 23-27, 2011, Bonn, Germany*, Bonn, GERMANY, 10 2011.

[112] Nadav Rotem. Open source text summarizer, 2006.

[113] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk e-mail, 1998.

[114] M. Saravanan and B. Ravindran. Identification of rhetorical roles for segmentation and summarization of a legal judgment. *Artif. Intell. Law*, 18(1):45–76, March 2010.

[115] Hinrich Schütze, David A. Hull, and Jan O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 229–237, New York, NY, USA, 1995. ACM.

[116] Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. Summarizing microblogs automatically. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 685–688, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[117] Beaux Sharifi, Mark-Anthony Hutton, and Jugal K. Kalita. Experiments in microblog summarization. In Ahmed K. Elmagarmid and Divyakant Agrawal, editors, *SocialCom/PASSAT*, pages 49–56. IEEE Computer Society, 2010.

[118] Hutton M.A. & Kalita J. Sharifi, B. Automatic summarization of twitter topics. In *National Workshop on Design and Analysis of Algorithms*, 2010.

[119] Vikas Sindhwani and S. Sathiya Keerthi. Large scale semi-supervised linear svms. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 477–484, New York, NY, USA, 2006. ACM.

[120] John Sowa. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks/Cole, Pacific Grove, CA, 2000.

[121] Richard Sproat and Steven Bedrick. Cs506/606: Txt nrmlzt, 2011.

[122] Richard Sproat, Alan W. Black, Stanley F. Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. Normalization of non-standard words. *Computer Speech & Language*, 15(3):287–333, 2001.

[123] M. Srinivas and Lalit M. Patnaik. Genetic algorithms: A survey. *IEEE Computer*, 27(6):17–26, 1994.

[124] M. Steinberger R. Tanev H. Turchi M. Zavarella V. Steinberger, J. Kabadjov. Jrc's participation at tac 2011: Guided and multilingual summarization tasks. In *TAC'11: Proceedings of the The 2011 Text Analysis Conference*, 2011.

[125] R. Stevens, C. Goble, and S. Bechhofer. Ontology-based Knowledge Representation for Bioinformatics. *Briefings in Bioinformatics.*, 1.4:398–414, 2000.

[126] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM.

[127] Hiroya Takamura and Manabu Okumura. Text summarization model based on maximum coverage problem and its variant. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 781–789, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[128] Hiroya Takamura and Manabu Okumura. Text summarization model based on the budgeted median problem. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1589–1592, New York, NY, USA, 2009. ACM.

[129] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

[130] Nikolaj Tatti. Probably the best itemsets. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 293–302, 2010.

[131] Nikolaj Tatti and Hannes Heikinheimo. Decomposable families of itemsets. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *ECML/PKDD (2)*, volume 5212 of *Lecture Notes in Computer Science*, pages 472–487. Springer, 2008.

[132] Nikolaj Tatti and Michael Mampaey. Using background knowledge to rank itemsets. *Data Min. Knowl. Discov.*, 21(2):293–309, September 2010.

[133] Khushboo S. Thakkar, R.V. Dharaskar, and M.B. Chandak. Graph-based algorithms for text summarization. *Emerging Trends in Engineering & Technology, International Conference on*, 0:516–519, 2010.

[134] Christopher Town. Ontological inference for image and video analysis. *Machine Vision and Applications*, 17:94–115, 2006. 10.1007/s00138-006-0017-3.

[135] Xiaojun Wan and Jianwu Yang. Improved affinity graph based multi-document summarization. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 181–184, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[136] Dingding Wang and Tao Li. Document update summarization using incremental hierarchical clustering. In Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An, editors, *CIKM*, pages 279–288. ACM, 2010.

[137] Dingding Wang, Shenghuo Zhu, Tao Li, Yun Chi, and Yihong Gong. Integrating document clustering and multidocument summarization. *ACM Trans. Knowl. Discov. Data*, 5(3):14:1–14:26, August 2011.

[138] Sholom M. Weiss and Nitin Indurkhya. Optimized rule induction. *IEEE Expert*, 8(6):61–69, 1993.

[139] Sholom M. Weiss, Nitin Indurkhya, and T. Zhang. *Text Mining. Predictive Methods for Analyzing Unstructured Information.* Springer, Berlin, 1 edition, 2004.

[140] Eric W. Weisstein. Statistical test.

[141] Erik D. Wiener, Jan O. Pedersen, and Andreas S. Weigend. A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 317–332, Las Vegas, US, 1995.

[142] Wikipedia. Wikipedia website. last access: 01/03/2013, 2013.

[143] Chia-Wei Wu and Chao-Lin Liu. Ontology-based text summarization for business news articles. In Narayan C. Debnath, editor, *Computers and Their Applications*, pages 389–392. ISCA, 2003.

[144] Yiming Yang and Christopher G. Chute. An example-based mapping method for text categorization and retrieval. *ACM Trans. Inf. Syst.*, 12(3):252–277, July 1994.

[145] Zi Yang, Keke Cai, Jie Tang, Li Zhang, Zhong Su, and Juanzi Li. Social context summarization. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 255–264, New York, NY, USA, 2011. ACM.

[146] Z. Yin, R. Li, Q. Mei, and J. Han. Exploring social tagging graph for web object classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 957–966. ACM, 2009.

[147] Eyob Delele Yirdaw and Dejene Ejigu. Topic-based amharic text summarization with probabilistic latent semantic analysis. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, MEDES '12, pages 8–15, New York, NY, USA, 2012. ACM.

[148] Osmar R. Zaiane. Introduction to data mining, 1999.

[149] Renxian Zhang, Wenjie Li, Dehong Gao, and Ouyang You. Automatic twitter topic summarization with speech acts. *IEEE Transactions on Audio, Speech & Language Processing*, 21(3):649–658, 2013.

[150] Junyan Zhu, Can Wang, Xiaofei He, Jiajun Bu, Chun Chen, Shujie Shang, Mingcheng Qu, and Gang Lu. Tag-oriented document summarization. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 1195–1196, New York, NY, USA, 2009. ACM.