A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques

Mehdi Allahyari
Computer Science Department
University of Georgia
Athens, GA
mehdi@uga.edu

Seyedamin Pouriyeh
Computer Science Department
University of Georgia
Athens, GA
pouriyeh@uga.edu

Mehdi Assefi
Computer Science Department
University of Georgia
Athens, GA
asf@uga.edu

Saied Safaei
Computer Science Department
University of Georgia
Athens, GA
ssa@uga.edu

Elizabeth D. Trippe
Institute of Bioinformatics
University of Georgia
Athens, GA
edt37727@uga.edu

Juan B. Gutierrez
Department of Mathematics
Institute of Bioinformatics
University of Georgia
Athens, GA
jgutierr@uga.edu

Krys Kochut
Computer Science Department
University of Georgia
Athens, GA
kochut@cs.uga.edu

ABSTRACT

The amount of text that is generated every day is increasing dramatically. This tremendous volume of mostly unstructured text cannot be simply processed and perceived by computers. Therefore, efficient and effective techniques and algorithms are required to discover useful patterns. Text mining is the task of extracting meaningful information from text, which has gained significant attentions in recent years. In this paper, we describe several of the most fundamental text mining tasks and techniques including text pre-processing, classification and clustering. Additionally, we briefly explain text mining in biomedical and health care domains.

CCS CONCEPTS

• **Information systems → Document topic models**; **Information extraction**; **Clustering and classification**;

KEYWORDS

Text mining, classification, clustering, information retrieval, information extraction

ACM Reference format:
Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. 2017. A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques . In *Proceedings of KDD Bigdas, Halifax, Canada, August 2017,* 13 pages.

1 INTRODUCTION

Text Mining (TM) field has gained a great deal of attention in recent years due the tremendous amount of text data, which are created in a variety of forms such as social networks, patient records, health care insurance data, news outlets, etc. IDC, in a report sponsored by EMC, predicts that the data volume will grow to 40 zettabytes[1] by 2020, leading to a 50-time growth from the beginning of 2010 [52].

Text data is a good example of unstructured information, which is one of the simplest forms of data that can be generated in most scenarios. Unstructured text is easily processed and perceived by humans, but is significantly harder for machines to understand. Needless to say, this volume of text is an invaluable source of information and knowledge. As a result, there is a desperate need to design methods and algorithms in order to effectively process this avalanche of text in a wide variety of applications.

Text mining approaches are related to traditional data mining, and knowledge discovery methods, with some specificities, as described below.

1.1 Knowledge Discovery vs. Data Mining

There are various definitions for ***knowledge discovery*** or ***knowledge discovery in databases (KDD)*** and ***data mining*** in the literature. We define it as follows:
Knowledge Discovery in Databases is extracting implicit valid, new and potentially useful information from data, which is non-trivial [45, 48]. **Data Mining** is a the application of particular algorithms for extracting patterns from data. KDD aims at discovering hidden patterns and connections in the data. Based on the above

[1] `1 ZB = $10^{21}$ bytes = 1 billion terabytes.`

definitions KDD refers to the overall *process* of discovering useful knowledge from data while *data mining* refers to a specific *step* in this process. Data can be structured like databases, but also unstructured like data in a simple text file.

Knowledge discovery in databases is a process that involves several steps to be applied to the data set of interest in order to excerpt useful patterns. These steps are iterative and interactive they may need decisions being made by the user. CRoss Industry Standard Process for Data Mining (Crisp DM[2]) model defines these primary steps as follows: *1*) understanding of the application and data and identifying the goal of the KDD process, *2*) data preparation and preprocessing, *3*) modeling, *4*) evaluation, *5*) deployment. Data cleaning and preprocessing is one of the most tedious steps, because it needs special methods to convert textual data to an appropriate format for data mining algorithms to use.

等价和包含

Data mining and knowledge discovery terms are often used interchangeably. Some would consider data mining as *synonym* for knowledge discovery, i.e. data mining consists of all aspects of KDD process. The second definition considers data mining as *part of* the KDD process (see [45]) and explicate the modeling step, i.e. selecting methods and algorithms to be used for searching for patterns in the data. We consider data mining as a modeling phase of KDD process.

Research in knowledge discovery and data mining has seen rapid advances in recent years, because of the vast progresses in hardware and software technology. Data mining continues to evolve from the intersection of diverse fields such as machine learning, databases, statistics and artificial intelligence, to name a few, which shows the underlying interdisciplinary nature of this field. We briefly describe the relations to the three of aforementioned research areas.

Databases are essential to efficiently analyze large amounts of data. Data mining algorithms on the other hand can significantly boost the ability to analyze the data. Therefore for the data integrit y and management considerations, data analysis requires to be int e-grated with databases [105]. An overview for the data mining fro m the database perspective can be found in [28].

Machine Learning (ML) is a branch of artificial intelligence that tries to define set of approaches to find patterns in data to be able to predict the patterns of future data. Machine learning involves study of methods and algorithms that can extract information automatically. There are a great deal of machine learning algorithms used in data mining. For more information please refer to [101, 126].

Statistics is a mathematical science that deals with collection, analysis, interpretation or explanation, and presentation of data[3]. Today lots of data mining algorithms are based on statistics and probability methods. There has been a tremendous quantities of research for data mining and statistical learning [1, 62, 78, 137].

1.2 Text Mining Approaches

Text Mining or knowledge discovery from text (KDT) – first introduced by Fledman et al. [46] – refers to the process of extracting high quality of information from text (i.e. structured such as RDBMS data [28, 43], semi-structured such as XML and JSON [39, 111, 112], and unstructured text resources such as word documents, videos, and images). It widely covers a large set of related topics and algorithms for analyzing text, spanning various communities, including information retrieval, natural language processing, data mining, machine learning many application domains web and biomedical sciences.

Information Retrieval (IR): Information Retrieval is the activity of finding information resources (usually documents) from a collection of unstructured data sets that satisfies the information need [44, 93]. Therefore information retrieval mostly focused on facilitating information access rather than analyzing information and finding hidden patterns, which is the main purpose of text mining. Information retrieval has less priority on processing or transformation of text whereas text mining can be considered as going beyond information access to further aid users to analyze and understand information and ease the decision making.

what we focus

Natural Language Processing (NLP): Natural Language Processing is sub-field of computer science, artificial intelligence and linguistics which aims at understanding of natural language using computers [90, 94]. Many of the text mining algorithms extensively make use of NLP techniques, such as part of speech tagging (POG), syntactic parsing and other types of linguistic analysis (see [80, 116] for more information).

Information Extraction from text (IE): Information Extraction is the task of automatically extracting information or facts from unstructured or semi-structured documents [35, 122]. It usually serves as a starting point for other text mining algorithms. For example extraction entities, Name Entity Recognition (NER), and their relations from text can give us useful semantic information.

Text Summarization: Many text mining applications need to summarize the text documents in order to get a concise overview of a large document or a collection of documents on a topic [67, 115]. There are two categories of summarization techniques in general: *extractive* summarization where a summary comprises information units extracted from the original text, and in contrary *abstractive* summarization where a summary may contain "synthesized" information that may not occur in the original document (see [6, 38] for an overview).

Unsupervised Learning Methods: Unsupervised learning methods are techniques trying to find hidden structure out of unlabeled data. They do not need any training phase, therefore can be applied to any text data without manual effort. *Clustering* and *topic modeling* are the two commonly used unsupervised learning algorithms used in the context of text data. Clustering is the task of segmenting a collection of documents into partitions where documents in the same group (cluster) are more similar to each other than those in

[2] http://www.crisp-dm.org/
[3] http://en.wikipedia.org/wiki/Statistics

other clusters. In topic modeling a probabilistic model is used to determine a *soft* clustering, in which every document has a probability distribution over all the clusters as opposed to hard clustering of documents. In topic models each *topic* can be represented as a probability distributions over words and each documents is expressed as probability distribution over topics. Thus, a topic is akin to a cluster and the membership of a document to a topic is probabilistic [1, 133].

Supervised Learning Methods: Supervised learning methods are machine learning techniques pertaining to infer a function or learn a classifier from the training data in order to perform predictions on unseen data. There is a broad range of supervised methods such as nearest neighbor classifiers, decision trees, rule-based classifiers and probabilistic classifiers [101, 126].

Probabilistic Methods for Text Mining: There are various probabilistic techniques including unsupervised topic models such as probabilistic Latent semantic analysis (pLSA) [66] and Latent Dirichlet Allocation (LDA) [16], and supervised learning methods such as conditional random fields [85] that can be used regularly in the context of text mining.

Text Streams and Social Media Mining: There are many different applications on the web which generate tremendous amount of streams of text data. news stream applications and aggregators such as Reuters and Google news generate huge amount of text streams which provides an invaluable source of information to mine. Social networks, particularly Facebook and Twitter create large volumes of text data continuously. They provide a platform that allows users to freely express themselves in a wide range of topics. The dynamic nature of social networks makes the process of text mining difficult which needs special ability to handle poor and non-standard language [58, 146].

Opinion Mining and Sentiment Analysis: With the advent of e-commerce and online shopping, a huge amount of text is created and continues to grow about different product reviews or users opinions. By mining such data we find important information and opinion about a topic which is significantly fundamental in advertising and online marketing (see [109] for an overview).

Biomedical Text Mining: Biomedical text mining refers to the task of text mining on text of biomedical sciences domains. The role of text mining in biomedical domain is two fold, it enables the biomedical researchers to efficiently and effectively access and extract the knowledge out of the massive volumes of data and also facilitates and boosts up biomedical discovery by augmenting the mining of other biomedical data such as genome sequences and protein structures [60].

2 TEXT REPRESENTATION AND ENCODING

Text mining on a large collection of documents is usually a complex process, thus it is critical to have a data structure for the text which facilitates further analysis of the documents [67]. The most common way to represent the documents is as a*bag of words* (BOW), which considers the number of occurrences of each term (word/phrase) but ignores the order. This representation leads to a vector representation that can be analyzed with dimension reduction algorithms from machine learning and statistics. Three of the main dimension reduction techniques used in text mining are *Latent Semantic Indexing* (LSI) [42], *Probabilistic Latent Semantic Indexing* (PLSA) [66] and *topic models* [16].

In many text mining applications, particularly information retrieval (IR), documents needs to be ranked for more effective retrieval over large collections [131]. In order to be able to define the importance of a word in a document, documents are represented as vectors and a numerical *importance* is assigned to each word. The three most used model based on this idea are vector space model (VSM) (see section 2.2), probabilistic models [93] and inference network model [93, 138].

2.1 Text Preprocessing

Preprocessing is one of the key components in many text mining algorithms. For example a traditional text categorization framework comprises preprocessing, feature extraction, feature selection and classification steps. Although it is confirmed that feature extraction [57], feature selection [47] and classification algorithm [135] have significant impact on the classification process, the preprocessing stage may have noticeable influence on this success. Uysal et al. [140] have investigated the impact of preprocessing tasks particularly in the area of text classification. The preprocessing step usually consists of the tasks such as tokenization, filtering, lemmatization and stemming. In the following we briefly describe them.

Tokenization: Tokenization is the task of breaking a character sequence up into pieces (words/phrases) called tokens, and perhaps at the same time throw away certain characters such as punctuation marks. The list of tokens then is used to further processing [143].

Filtering: Filtering is usually done on documents to remove some of the words. A common filtering is stop-words removal. Stop words are the words frequently appear in the text without having much content information (e.g. prepositions, conjunctions, etc). Similarly words occurring quite often in the text said to have little information to distinguish different documents and also words occurring very rarely are also possibly of no significant relevance and can be removed from the documents [119, 130].

Lemmatization: Lemmatization is the task that considers the morphological analysis of the words, i.e. grouping together the various inflected forms of a word so they can be analyzed as a single item. In other words lemmatization methods try to map verb forms to infinite tense and nouns to a single form. In order to lemmatize the documents we first must specify the POS of each word of the documents and because POS is tedious and error prone, in practice *stemming* methods are preferred.

Stemming: Stemming methods aim at obtaining stem (root) of derived words. Stemming algorithms are indeed language dependent. The first stemming algorithm introduced in [92], but the stemmer published in [110] is most widely stemming method used in English [68].

2.2 Vector Space Model

In order to allow for more formal descriptions of the algorithms, we first define some terms and variables that will be frequently used in the following: Given a collection of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_D\}$, let $\mathcal{V} = \{w_1, w_2, \ldots, w_v\}$ be the set of distinct words/terms in the collection. Then $\mathcal{V}$ is called the *vocabulary*. The *frequency* of the term $w \in \mathcal{V}$ in document $d \in \mathcal{D}$ is shown by $f_d(w)$ and the number of documents having the word w is represented by $f_{\mathcal{D}}(w)$. The term vector for document d is denoted by $\vec{t_d} = (f_d(w_1), f_d(w_2), \ldots,$ $f_d(w_v))$.

The most common way to represent documents is to convert them into numeric vectors. This representation is called "Vector Space Model" (VSM). Event though its structure is simple and originally introduced for indexing and information retrieval [121], VSM is broadly used in various text mining algorithms and IR systems and enables efficient analysis of large collection of documents [67].

In VSM each word is represented by a variable having a numeric value indicating the *weight* (importance) of the word in the document. There are two main term weight models: *1*) *Boolean model*: In this model a weight $\omega_{ij} > 0$ is assigned to each term $w_i \in d_j$. For any term that does not appear in $d_j, \omega_{ij} = 0$. *2*) *Term frequency-inverse document frequency* (TF-IDF): The most popular term weighting schemes is TF-IDF. Let q be this term weighting scheme, then the weight of each word $w \in d$ is computed as follows:

$$q(w) = f_d(w) * log \frac{|\mathcal{D}|}{f_{\mathcal{D}}(w)} \tag{1}$$

where $|\mathcal{D}|$ is the number of documents in the collection $\mathcal{D}$.

In TF-IDF the term frequency is normalized by *inverse document frequency*, IDF. This normalization decreases the weight of the terms occurring more frequently in the document collection, Making sure that the matching of documents be more effected by distinctive words which have relatively low frequencies in the collection.

Based on the term weighting scheme, each document is represented by a vector of term weights $\omega(d) = (\omega(d, w_1), \omega(d, w_2), \ldots,$ $\omega(d, w_v))$. We can compute the similarity between two documents d_1 and d_2. One of the most widely used similarity measures is cosine similarity and is computed as follows:

$$S(d_1, d_2) = cos(\theta) = \frac{d_1 \cdot d_2}{\sqrt{\sum_{i=1}^{v} w_{1i}^2} \cdot \sqrt{\sum_{i=1}^{v} w_{2i}^2}} \tag{2}$$

[120, 121] discussed term weighting schemes and vector space models in more details.

3 CLASSIFICATION

Text classification has been broadly studied in different communities such as data mining, database, machine learning and information retrieval, and used in vast number of applications in various domains such as image processing, medical diagnosis, document organization, etc. Text classification aims to assign predefined classes to text documents [101]. The problem of classification is defined as follows. We have a *training set* $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ of documents, such that each document d_i is labeled with a label ℓ_i from the set $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_k\}$. The task is to find a *classification model* (classifier) f where

$$f : \mathcal{D} \longrightarrow \mathcal{L} \qquad\qquad f(d) = \ell \tag{3}$$

which can assign the correct class label to new document d (test instance). The classification is called *hard*, if a label is explicitly assigned to the test instance and *soft*, if a probability value is assigned to the test instance. There are other types of classification which allow assignment of multiple labels [54] to a test instance. For an extensive overview on a number of classification methods see [41, 71]. Yang et al. evaluates various kinds of text classification algorithms [147]. Many of the classification algorithms have been implemented in different software systems and are publicly available such as BOW toolkit [98], Mallet [99] and WEKA[4].

To evaluate the performance of the classification model, we set a side a random fraction of the labeled documents (test set). After training the classifier with training set, we classify the test set and compare the estimated labels with the true labels and measure the performance. The portion of correctly classified documents to the total number of documents is called *accuracy* [67]. The common evaluation metrics for text classification are precision, recall and F-1 scores. Charu et al. [1] defines the metrics as follows: "*precision* is the fraction of the correct instances among the identified positive instances. *Recall* is the percentage of correct instances among all the positive instances. And *F-1 score* is the geometric mean of precision and recall".

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{4}$$

3.1 Naive Bayes Classifier

Probabilistic classifiers have gained a lot of popularity recently and have shown to perform remarkably well [24, 73, 84, 86, 118]. These probabilistic approaches make assumptions about how the data (words in documents) are generated and propose a probabilistic model based on these assumptions. Then use a set of training examples to estimate the parameters of the model. Bayes rule is used to classify new examples and select the class that is most likely has generated the example [96].

The *Naive Bayes* classifier is perhaps the simplest and the most widely used classifier. It models the distribution of documents in each class using a probabilistic model assuming that the distribution of different terms are *independent* from each other. Even though this so called "naive Bayes" assumption is clearly false in many real world applications, naive Bayes performs surprisingly well.

There are two main models commonly used for naive Bayes classifications [96]. Both models aim at finding the posterior probability

[4] http://www.cs.waikato.ac.nz/ml/weka/

of a class, based on the distribution of the words in the document. They difference between these two models is, one model takes into account the frequency of the words whereas the other one does not.

(1) **Multi-variate Bernoulli Model:** In this model a document is represented by a vector of binary features denoting the presence or absence of the words in the document. Thus, the frequency of words are ignored. The original work can be found in [88].
(2) **Multinomial Model:** We capture the frequencies of words (terms) in a document by representing the document as bag of words. Many different variations of multinomial model have been introduced in [76, 97, 101, 106]. McCallum et al. [96] have done an extensive comparison between Bernoulli and multinomial models and concluded that
   - If the size of the vocabulary is small, the Bernoulli model may outperform multinomial model.
   - The multinomial model always outperforms Bernoulli model for large vocabulary sizes, and almost always performs better than Bernoulli if the size of the vocabulary chosen optimally for both models.

Both of these models assume that the documents are generated by a mixture model parameterized by θ. We use the framework McCallum el at. [96] defined as follows:

The mixture model comprises mixture components $c_j \in C = \{c_1, c_2, \ldots, c_k\}$. Each document $d_i = \{w_1, w_2, \ldots, w_{n_i}\}$ is generated by first selecting a component according to priors, $P(c_j|\theta)$ and then use the component to create the document according to its own parameters, $P(d_i|c_j;\theta)$. Hence, we can compute the likelihood of a document using the sum of probabilities over all mixture components:

$$P(d_i|\theta) = \sum_{j=1}^{k} P(c_j|\theta) P(d_i|c_j;\theta) \tag{5}$$

We assume a one to one correspondence between classes $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_k\}$ and mixture components, and therefore c_j indicates both the jth mixture component and the jth class. Consequently, Given a set of labeled training examples, $\mathcal{D} = \{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$, we first learn (estimate) the parameters of the probabilistic classification model, $\hat{\theta}$, and then using the estimates of these parameters, we perform the classification of test documents by calculating the posterior probabilities of each class c_j, given the test document, and select the most likely class (class with the highest probability).

$$\begin{aligned} P(c_j|d_i;\hat{\theta}) &= \frac{P(c_j|\hat{\theta}) P(d_i|c_j;\hat{\theta}_j)}{P(d_i|\hat{\theta})} \\ &= \frac{P(c_j|\hat{\theta}) P(w_1, w_2, \ldots, w_{n_i}|c_j;\hat{\theta}_j)}{\sum_{c \in C} P(w_1, w_2, \ldots, w_{n_i}|c;\hat{\theta}_c) P(c|\hat{\theta})} \end{aligned} \tag{6}$$

where based on naive Bayes assumption, words in a document are independent of each other, thus:

$$P(w_1, w_2, \ldots, w_{n_i}|c_j;\hat{\theta}_j) = \prod_{i=1}^{n_i} P(w_i|c_j;\hat{\theta}_j) \tag{7}$$

3.2 Nearest Neighbor Classifier

Nearest neighbor classifier is a proximity-based classifier which use distance-based measures to perform the classification. The main idea is that documents which belong to the same class are more likely "similar" or close to each other based on the similarity measures such as cosine defined in (2.2). The classification of the test document is inferred from the class labels of the similar documents in the training set. If we consider the k-nearest neighbor in the training data set, the approach is called *k-nearest neighbor classification* and the most common class from these k neighbors is reported as the class label, see [61, 93, 117, 126] for more information and examples.

3.3 Decision Tree classifiers

Decision tree is basically a hierarchical tree of the training instances, in which a condition on the attribute value is used to divide the data hierarchically. In other words decision tree [50] recursively partitions the training data set into smaller subdivisions based on a set of tests defined at each node or branch. Each node of the tree is a test of some *attribute* of the traning instance, and each branch descending from the node corresponds to one the value of this attribute. An instance is classified by beginning at the root node, testing the attribute by this node and moving down the tree branch corresponding to the value of the attribute in the given instance. And this process is recursively repeated [101].

In case of text data, the conditions on the decision tree nodes are commonly defined in terms of terms in the text documents. For instance a node may be subdivided to its children relying on the presence or absence of a particular term in the document. For a detailed discussion of decision trees see [19, 41, 71, 113].

Decision trees have been used in combination with boosting techniques. [49, 125] discuss boosting techniques to improve the accuracy of the decision tree classification.

3.4 Support Vector Machines

Support Vector Machines (SVM) are supervised learning classification algorithms where have been extensively used in text classification problems. SVM are a form of *Linear Classifiers.* Linear classifiers in the context of text documents are models that making a classification decision is based on the value of the linear combinations of the documents features. Thus, the output of a linear predictor is defined to be $y = \vec{a} \cdot \vec{x} + b$, where $\vec{x} = (x_1, x_2, \ldots, x_n)$ is the normalized document word frequency vector, $\vec{a} = (a_1, a_2, \ldots, a_n)$ is vector of coefficients and b is a scalar. We can interpret the predictor $y = \vec{a} \cdot \vec{x} + b$ in the categorical class labels as a *separating hyperplane* between different classes.

The SVM initially introduced in [34, 141]. Support Vector Machines try to find a "good" linear separators between various classes [34, 142]. A single SVM can only separate two classes, a positive class and a negative class [67]. SVM algorithm attempts to find a hyperplane with the maximum distance ξ (also called *margin*) from the positive and negative examples. The documents with distance ξ from the hyperplane are called *support vectors* and specify the actual location of the hyperplane. If the document vectors of the two classes are not linearly separable, a hyperplane is determined such that the least number of document vectors are located in the wrong side.

One advantage of the SVM method is that, it is quite robust to high dimensionality, i.e. learning is almost independent of the dimensionality of the feature space. It rarely needs feature selection since it selects data points (support vectors) required for the classification [67]. Joachims et al. [74] has described that text data is an ideal choice for SVM classification due to sparse high dimensional nature of the text with few irrelevant features. SVM methods have been widely used in many application domains such as pattern recognition, face detection and spam filtering [21, 40, 108]. For a deeper theoretical study of SVM method see [75].

4 CLUSTERING

Clustering is one of the most popular data mining algorithms and have extensively studied in the context of text. It has a wide range of applications such as in classification [11, 12], visualization [22] and document organization [37]. The clustering is the task of finding groups of similar documents in a collection of documents. The similarity is computed by using a similarity function. Text clustering can be in different levels of granularities where clusters can be documents, paragraphs, sentences or terms. Clustering is one of the main techniques used for organizing documents to enhance retrieval and support browsing, for example Cutting et al. [36] have used clustering to produce a table of contents of a large collection of documents. [9] exploits clustering to construct a context-based retrieval systems. For a broad overview of clustering see [70, 82]. There are various software tools such as Lemur[5] and BOW [98] which have implementations of common clustering algorithms.

There are many clustering algorithms that can be used in the context of text data. Text document can be represented as a binary vector, i.e. considering the presence or absence of word in the document. Or we can use more refined representations which involves weighting methods such as TF-IDF (see section 2.2).

Nevertheless, such naive methods do not usually work well for text clustering, since text data has a number of distinct characteristics which demands the design of text-specific algorithms for the task. We describe some of these unique properties of text representation:

i. Text representation has a very large dimensionality, but the underlying data is sparse. In other words, the size of the vocabulary from which the documents are drawn is massive (e.g. order of 10^5), but a given document may have only a few hundred words. This problem becomes even more severe when we deal with short data such as tweets.
ii. Words of the vocabulary of a given collection of documents are commonly correlated with each other. i.e. the number of concepts in the data are much smaller that the feature space. Thus, we need to design algorithms which take the word correlation into consideration in the clustering task.
iii. Since documents differ from one another in terms of the number of words they contain, normalizing document representations during the clustering process is important.

The aforementioned text characteristics necessitates the design of specialized algorithms for representing text and broadly investigated in IR community and many algorithms have been proposed to optimize text representation [120].

Text clustering algorithms are split into many different types such as agglomerative clustering algorithms, partitioning algorithms and probabilistic clustering algorithms. Clustering algorithms have varied trade offs in terms of effectiveness and efficiency. For an experimental comparison of different clustering algorithms see [132, 151], and for a survey of clustering algorithms see [145]. In the following we describe some of most common text clustering algorithms.

4.1 Hierarchical Clustering algorithms

Hierarchical clustering algorithms received their name because they build a group of clusters that can be depicted as a hierarchy of clusters. The hierarchy can be constructed in top-down (called *divisive*) or bottom-up (called *agglomerative*) fashion. Hierarchical clustering algorithms are one of the *Distanced-based clustering algorithms*, i.e. using a similarity function to measure the closeness between text documents. An extensive overview of the hierarchical clustering algorithms for text data is found in [103, 104, 144].

In the top-down approach we begin with one cluster which includes all the documents. we recursively split this cluster into sub-clusters. In the agglomerative approach, each document is initially considered as an individual cluster. Then successively the most similar clusters are merged together until all documents are embraced in one cluster. There are three different merging methods for agglomerative algorithms: *1*) *Single Linkage Clustering:* In this technique, the similarity between two groups of documents is the highest similarity between any pair of documents from these groups. *2*) *Group-Average Linkage Clustering:* In group-average clustering, the similarity between two cluster is the *average* similarity between pairs of documents in these groups. *3*) *Complete Linkage Clustering:* In this method, the similarity between two clusters is the *worst case* similarity between any pair of documents in these groups. For more information about these merging techniques see [1].

4.2 *k*-means Clustering

k-means clustering is one the *partitioning* algorithms which is widely used in the data mining. The *k*-means clustering, partitions *n* documents in the context of text data into *k* clusters. representative around which the clusters are built. The basic form of *k*-means algorithm is:

Finding an optimal solution for *k*-means clustering is computationally difficult (NP-hard), however, there are efficient heuristics such as [18] that are employed in order to converge rapidly to a local optimum. The main disadvantage of *k*-means clustering is that it is indeed very sensitive to the initial choice of the number of *k*. Thus, there are some techniques used to determine the initial *k*, e.g. using another lightweight clustering algorithm such as agglomerative

[5] http://www.lemurproject.org/

ALGORITHM 1: k-means clustering algorithm

```
Input  : Document set $\mathcal{D}$, similarity measure $\mathcal{S}$, number $k$ of
         cluster
Output : Set of $k$ clusters
initialization
Select randomly $k$ data points as starting centroids.
while not converged do
    Assign documents to the centroids based on the closest
      similarity.
    Calculate the the cluster centroids for all the clusters.
end
return $k$ clusters
```

clustering algorithm. More efficient k-means clustering algorithms can be found in [7, 79].

4.3 Probabilistic Clustering and Topic Models

Topic modeling is one of the most popular the probabilistic clustering algorithms which has gained increasing attention recently. The main idea of topic modeling [16, 55, 66] is to create a *probabilistic generative model* for the corpus of text documents. In topic models, documents are mixture of topics, where a topic is a probability distribution over words.

The two main topic models are *Probabilistic Latent Semantic Analysis (pLSA)* [66] and *Latent Dirichlet Allocation (LDA)* [16]. Hofmann (1999) introduced pLSA for document modeling. pLSA model does not provide any probabilistic model at the document level which makes it difficult to generalize it to model new unseen documents. Blei et al. [16] extended this model by introducing a Dirichlet prior on mixture weights of topics per documents, and called the model Latent Dirichlet Allocation (LDA). In this section we describe the LDA method.

The latent Dirichlet allocation model is the state of the art unsupervised technique for extracting thematic information (topics) of a collection of documents. [16, 56]. The basic idea is that documents are represented as a random mixture of latent topics, where each topic is a probability distribution over words. The LDA graphical representation is shown is Fig. 1.

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$ is the corpus and $\mathcal{V} = \{w_1, w_2, \ldots, w_{|\mathcal{V}|}\}$ is the vocabulary of the corpus. A topic $z_j, 1 \leq j \leq K$ is represented as a multinomial probability distribution over the $|\mathcal{V}|$ words, $p(w_i|z_j)$, $\sum_i^{|\mathcal{V}|} p(w_i|z_j) = 1$. LDA generates the words in a two-stage process: words are generated from topics and topics are generated by documents. More formally, the distribution of words given the document is calculated as follows:

$$p(w_i|d) = \sum_{j=1}^{K} p(w_i|z_j)p(z_j|d) \tag{8}$$

The LDA assumes the following generative process for the corpus $\mathcal{D}$:

(1) For each topic $k \in \{1, 2, \ldots, K\}$, sample a word distribution $\varphi_k \sim \text{Dir}(\beta)$
(2) For each document $d \in \{1, 2, \ldots, D\}$,
   (a) Sample a topic distribution $\theta_d \sim \text{Dir}(\alpha)$
   (b) For each word w_n, where $n \in \{1, 2, \ldots, N\}$, in document d,
      i. Sample a topic $z_i \sim \text{Mult}(\theta_d)$
      ii. Sample a word $w_n \sim \text{Mult}(\varphi_{z_i})$

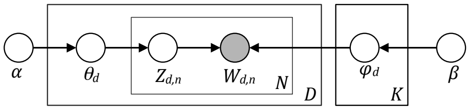

Figure 1: LDA Graphical Model

The joint distribution of the model (hidden and observed variables) is:

$$P(\phi_{1:K}, \theta_{1:\mathcal{D}}, z_{1:\mathcal{D}}, w_{1:\mathcal{D}}) = \prod_{j=1}^{K} P(\phi_j|\beta) \prod_{d=1}^{D} P(\theta_d|\alpha) \times \left(\prod_{n=1}^{N} P(z_{d,n}|\theta_d) P(w_{d,n}|\phi_{1:K}, z_{d,n}) \right) \tag{9}$$

4.3.1 Inference and Parameter Estimation for LDA. We now need to compute the *posterior* distribution of the hidden variables, given the observed documents. Thus, the posterior is:

$$P(\varphi_{1:K}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{P(\varphi_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{P(w_{1:D})} \tag{10}$$

This distribution is intractable to compute [16] due to the denominator (probability of seeing the observed corpus under any topic model).

While the posterior distribution (exact inference) is not tractable, a wide variety of approximate inference techniques can be used, including variational inference [16] and Gibbs sampling [56]. Gibbs sampling is a Markov Chain Monte Carlo [53] algorithm, trying to collect sample from the posterior to approximate it with an empirical distribution.

Gibbs sampling computes the posterior over topic assignments for every word as follows:

$$P(z_i = k|w_i = w, \mathbf{z}_{-i}, \mathbf{w}_{-i}, \alpha, \beta) = \frac{n_{k,-i}^{(d)} + \alpha}{\sum_{k'=1}^{K} n_{k',-i}^{(d)} + K\alpha} \times \frac{n_{w,-i}^{(k)} + \beta}{\sum_{w'=1}^{W} n_{w',-i}^{(k)} + W\beta} \tag{11}$$

where $z_i = k$ is the topic assignment of word i to topic k, z_{-i} refers to the topic assignments of all other words. $n_{w,-i}^{(k)}$ is the number of times word w assigned to topic k excluding the current assignment. Similarly, $n_{k,-i}^{(d)}$ is the number of times topic k is assigned to any words in document d excluding the current assignment. For a theoretical overview on Gibbs sampling see [23, 64].

LDA can be easily used as a module in more complicated models for more complex goals. Furthermore, LDA has been extensively used in a wide variety of domains. Chemudugunta et al. [27] combined LDA with concept hierarchy to model documents. [2, 5] developed ontology-based topic models based on LDA for automatic topic labeling and semantic tagging, respectively. [4] proposed a knowledge-based topic model for context-aware recommendations. [81, 127] defined more complex topic models based on LDA for entity disambiguation, [3] and [63] has proposed a entity-topic models for discovering coherence topics and entity linking, respectively. Additionally, many variations of LDA have been created such as supervised LDA (sLDA) [15], hierarchical LDA (hLDA) [14] and Hierarchical pachinko allocation model (HPAM) [100].

5 INFORMATION EXTRACTION

Information extraction (IE) is the task of automatically extracting structured information from unstructured or semi-structured text. In other words information extraction can be considered as a limited form of full natural language understanding, where the information we are looking for are known beforehand [67]. For example, consider the following sentence: "*Microsoft was founded by Bill Gates and Paul Allen on April 4, 1975.*"

We can identify following information:

`FounderOf`(*Bill Gates, Microsoft*)
`FounderOf`(*Paul Allen, Microsoft*)
`FoundedIn`(*Microsoft, April - 4 1975*)

IE is one of the critical task in text mining and widely studied in different research communities such as information retrieval, natural language processing and Web mining. Similarly, It has vast application in domains such as biomedical text mining and business intelligence. See [1] for some of the applications of information extraction.

Information extraction includes two fundamental tasks, namely, *name entity recognition* and *relation extraction.* The state of the art in both tasks are statistical learning methods. In the following we briefly explain two information extraction tasks.

5.1 Named Entity Recognition (NER)

A named entity is a sequence of words that identifies some real world entity, e.g. "Google Inc", "United States", "Barack Obama". The task of named entity recognition is to locate and classify named entities in free text into predefined categories such as *person*, *organization*, *location*, etc. NER can not be completely done simply by doing string matching against a dictionary, because *a*) dictionaries are usually incomplete and do not contain all forms of named entities of a given entity type. *b*) Named entities are frequently dependent on context, for example "big apple" can be the fruit, or the nickname of New York.

Named entity recognition is a preprocessing step in the relation extraction task and also has other applications such as in question answering [1, 89]. Most of the named entity recognition techniques are statistical learning methods such as hidden Markov models [13], maximum entropy models [29], support vector machines [69] and conditional random fields [128].

5.2 Hidden Markov Models

Standard probabilistic classification techniques usually do not consider the predicted labels of the neighboring words. The probabilistic models which take this into account are hidden Markov model (HMM). Hidden Markov model assumes a Markov process in which generation of a label or an observation depends on one or a few previous labels or observations. Therefore, given a sequence of labels $\mathbf{Y} = (y_1, y_2, \ldots, y_n)$ for an observation sequence $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, we have:

$$y_i \sim p(y_i|y_{i-1}) \qquad x_i \sim p(x_i|x_{i-1}) \tag{12}$$

Hidden Markov models have been successfully used in the named entity recognition task and speech recognition systems. For an overview on hidden Markov models see [114].

5.3 Conditional Random Fields

Conditional random fields (CRFs) are probabilistic models for sequence labeling. CRFs first introduced by Lafferty et al. [85]. We refer to the same definition of conditional random fields in [85] on observations (data sequences to be labeled) and **Y** (sequence of labels) as follows:

Definition. *Let $G = (V, E)$ be a graph such that $\mathbf{Y} = (\mathbf{Y}_v)_{v \in V}$, so that* **Y** *is indexed by vertices of* **G**. *Then* $(\mathbf{X}, \mathbf{Y})$ *is a conditional random field, when the random variables* $\mathbf{Y}_v$, *conditioned on* **X**, *obey Markov property with respect to graph, and*:

$$p(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, w \neq v) = p(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, w \sim v) \tag{13}$$

where $w \sim v$ means w and v are neighbors in G.

Conditional random fields are widely used in information extraction and part of speech tagging [85]

5.4 Relation Extraction

Relation extraction is another fundamental information extraction task and is the task of seeking and locating the semantic relations between entities in text documents. There are many different techniques proposed for relation extraction. The most common method is consider the task as a classification problem: Given a couple of entities co-occurring in a sentence, how to categorize the relation between two entities into one of the fixed relation types. There is a possibility that relation span across multiple sentences, but such cases are rare, thus, most of existing work have focused on the relation extraction within the sentence. Many studies using the classification approach for relation extraction have been done such as [25, 26, 59, 72, 77].

6 BIOMEDICAL ONTOLOGIES AND TEXT MINING FOR BIOMEDICINE AND HEALTHCARE

One of the domains where text mining is tremendously used is biomedical sciences. Biomedical literature is growing exponentially, Cohen and Hunter [31] show that the growth in PubMed/MEDLINE publications is phenomenal, which makes it quite difficult for biomedical scientists to assimilate new publications and keep up with relevant publications in their own research area.

In order to overcome this text information overload and transform the text into machine-understandable knowledge, automated text processing methods are required. Thus, text mining techniques along with statistical machine learning algorithms are widely used in biomedical domain. Text mining methods have been utilized in a variety of biomedical domains such as protein structure prediction, gene clustering, biomedical hypothesis and clinical diagnosis, to name a few. In this section, we briefly describe some of the relevant research in biomedical domain, including biomedical ontologies and then proceed to explain some of the text mining techniques in biomedical discipline applied for basic tasks of named entity recognition and relation extraction.

6.1 Biomedical Ontologies

We first define the concept of *ontology*. We use the definition presented in W3C's OWL Use Case and Requirements Documents[6] as follows:

> *An ontology formally defines a common set of terms that are used to describe and represent a domain. An ontology defines the terms used to describe and represent an area of knowledge.*

According to the definition above, we should mention a few points about ontology: *1*) Ontology is domain specific, i.e., it is used to describe and represent *an area of knowledge* such as area in education, medicine, etc [149]. *2*) Ontology consists of terms and relationships among these terms. Terms are often called *classes* or concepts and relationships are called *properties*.

There are a great deal of biomedical ontologies. For a comprehensive list of biomedical ontologies see Open Biomedical Ontologies (OBO)[7] and the National Center for Biomedical Ontology (NCBO)[8]. The NCBO ontologies are accessed and shared through BioPortal[9]. In the following, we briefly describe one the most extensively-used ontologies in biomedical domain:

Unified Medical Language System (UMLS): UMLS[10] [95] is the most comprehensive knowledge resource, unifying over 100 dictionaries, terminologies and ontologies in its Metathesaurus (large vocabulary whose data are collected from various biomedical thesauri) which is designed and maintained by National Library of Medicine (NLM). It provides a mechanism for integrating [17] all main biomedical vocabularies such as MeSH, Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT), Gene Ontology (GO), etc.

It also provides a semantic network that explains the relations between Metathesaurus entries, i.e., a dictionary that includes lexicographic information about biomedical terms and common English words and a set of lexical tools. Semantic network contains semantic types and semantic relations. Semantic types are categories of Metathesaurus entries (concepts) and semantic relations are relationships between semantic types. For more information about UMLS, see [17, 148].

Apart from the aforementioned ontologies and knowledge sources, there are various ontologies more specifically focused on biomedical sub-domains. For example, the Pharmacogenomics Knowledge Base [11], consists of clinical information including dosing guidelines and drug labels, potentially clinically actionable gene-drug associations and genotype-phenotype relationships.

The ontologies and knowledge bases described earlier are extensively used by different text mining techniques such as information extraction and clustering in the biomedical domain.

6.2 Information Extraction

As mentioned before (section 5), *information extraction* is the task of extracting structured information from unstructured text in a automatic fashion. In biomedical domain, unstructured text comprises mostly scientific articles in biomedical literature and clinical information found in clinical information systems. Information extraction is typically considered as a preprocessing step in other biomedical text mining applications such as question answering [10], knowledge extraction [124, 136], hypothesis generation[30, 91] and summarization [65].

6.2.1 Named Entity Recognition (NER).

Named Entity Recognition is the task of information extraction which is used to locate and classify biomedical entities into categories such as protein names, gene names or diseases [87]. Ontologies can be utilized to give semantic, unambiguous representation to extracted entities. NER are quite challenging in biomedical domain, because:

(1) There is a huge amount of semantically related entities in biomedical domain and is increasing quickly with the new discoveries done in this field. This non-stop growing of the volume of entities is problematic for the NER systems, since they depends on dictionaries of terms which can never be complete due to continues progress in scientific literature.
(2) In biomedical domain, the same concept may have many different names (synonyms). For example, "heart attack" and "myocardial infarction" point to the same concept and NER systems should be able to recognize the same concept regardless of being expressed differently.
(3) Using acronyms and abbreviations is very common in biomedical literature which makes it complicated to identify the concepts these terms express.

It is critical for NER systems to have high quality and perform well when analyzing vast amounts of text. Precision, recall and *F*-score are typical evaluation methods used in NER systems. Even though there are some challenges to obtain and compare evaluation methods reliably, e.g. how to define the boundaries of correctly identified entities, NER systems have demonstrated to achieve good results in general.

NER methods are usually grouped into several different approaches:

- **Dictionary-based approach**, is one of the main biomedical NER methods which uses a exhaustive dictionary of biomedical terms to locate entity mentions in text. It decides whether a word or phrase from the text matches with some biomedical entity in the dictionary. Dictinary-based methods are mostly used with more advanced NER systems.

[6] http://www.w3.org/TR/webont-req/
[7] http://www.obofoundry.org/
[8] http://www.bioontology.org/
[9] http://bioportal.bioontology.org/
[10] https://uts.nlm.nih.gov/home.html
[11] http://www.pharmgkb.org/

- **Rule-based approach**, defines rules that specifies patterns of biomedical entities. Gaizauskas et at. [51] have used context free grammars to recognize protein structure.
- **Statistical approaches**, basically utilize some machine learning methods typically supervised or semi-supervised algorithms [139] to identify biomedical entities. Statistical methods are often categorized into two different groups:
  (1) **Classification-based approaches**, convert the NER task into a classification problem, which is applicable to either words or phrases. Naive Bayes [107] and Support Vector Machines [83, 102, 134] are among the common classifiers used for biomedical NER task.
  (2) **Sequence-based methods**, use complete sequence of words instead of only single words or phrases. They try to predict the most likely tag for a sequence of words after being trained on a training set. Hidden Markov Model (HMM) [32, 129, 150], Maximum Entropy Markov Model [33] and Conditional Random Fields (CRF) [128] are the most common sequence-based approaches and CRFs have frequently demonstrated to be better statistical biomedical NER systems.
  (3) **Hybrid methods**, which rely on multiple approaches, such as combining dictionary- or rule-based techniques with statistical methods. [123] introduced a hybrid method in which they have use a dictionary-based method to locate known protein names along with a part-of-speech tagging (CRF-based method).

6.2.2 Relation Extraction. Relation extraction in Biomedical domain involves determining the relationships among biomedical entities. Given two entities, we aim at locating the occurrence of a specific relationship type between them. The associations between entities are usually binary, however, it can include more than two entities. In the genomic area, for example, the focus is mostly on extracting interactions between genes and proteins, such as protein-protein or gene-diseases relationships. Relation extraction comes across similar challenges as NER, such as creation of high quality annotated data for training and assessing the performance of relation extraction systems. For more information see [8].

There are many different approaches for biomedical relation extraction. The most straightforward technique is based on where the entities co-occur. If they mentioned together frequently, there is high chance that they might be related in some way. But we can not recognize the type and direction of the relations by using only the statistics. Co-occurrence approaches are usually give high recall and low precision.

Rule-based approaches are another set of methods used for biomedical relation extraction. Rules can be defined either manually by domain experts or automatically obtained by using machine learning techniques from an annotated corpus. Classification-based approaches are also very common methods for relation extractions in biomedical domain. There is a great work done using supervised machine learning algorithms that detects and discovers various types of relations, such as [20] where they identify and classify relations between diseases and treatments extracted from PubMed abstracts and between genes and diseases in human GeneRIF database.

6.3 Summarization

One of the common biomedical text mining task which largely utilizes information extraction tasks is *summarization.* Summarization is the task of identifying the significant aspects of one or more documents and represent them in a coherent fashion *automatically* . it has recently gained a great attention because of the huge growth of unstructured information in biomedical domain such as scientific articles and clinical information.

Biomedical summarization is often application oriented and may be applied for different purposes. Based on their purpose, a variety of document summaries can be created such as single-document summaries which targets the content of individual documents and multi-document summaries where information contents of multiple documents are considered.

The evaluation of summarization methods is really challenging in biomedical domain. Because deciding whether or not a summary is "good" is often subjective and also manual evaluations of summaries are laborious to carry out. There is a popular automatic evaluation technique for summaries that is called ROUGE (Recall-Oriented Understudy for Gisting Evaluation). ROUGE measures the quality of an automatically produced summary by comparing it with ideal summaries created by humans. The measure is calculated by counting the overlapping words between the computer-generated summary and the ideal human-produced summaries. For a comprehensive overview of various biomedical summarization techniques, see [1].

6.4 Question Answering

Question answering is another biomedical text mining task where significantly exploits information extraction methods. *Question answering* is defined as the process of producing accurate answers to questions posed by humans in a natural language. Question answering is very critical in biomedical domain due to data overload and constant growth of information in this field.

In order to generate precise responses, question answering systems make extensive use of natural language processing techniques. The primary steps of question answering system is as follows: *a*) The system receives a natural language text as input. *b*) Using linguistic analysis and question categorization algorithms, the system determines the type of the posed question and the answer it should produce. *c*) Then it generates a query and passes it to the document processing phase. *d*) In the document processing phase, system feeds the query, system sends the query to a search engine, gets back the retrieved documents and extracts the relevant snippets of text as candidate answers, and send them to answering processing stage. *e*) Answering processing stage, analyzes the candidate answers and ranks them according to the degree they match the expected answer type that was established in the question processing step. *f*) The top-ranked answer is selected as the output of the question answering system.

Question answering systems in biomedical discipline have recently begun to utilize and incorporate semantic knowledge throughout their processing steps to create more accurate responses. These biomedical *semantic knowledge-based* systems use various semantic components such as semantic meta-data represented in knowledge sources and ontologies and semantic relationships to produce answers. See [1, 10] for a complete overview of different biomedical question answering techniques.

7 DISCUSSION

In this article we attempted to give a brief introduction to the field of text mining. We provided an overview of some of the most fundamental algorithms and techniques which are extensively used in the text domain. This paper also overviewed some of important text mining approaches in the biomedical domain. Even though, it is impossible to describe all different methods and algorithms throughly regarding the limits of this article, it should give a rough overview of current progresses in the field of text mining.

Text mining is essential to scientific research given the very high volume of scientific literature being produced every year [60]. These large archives of online scientific articles are growing significantly as a great deal of new articles are added in a daily basis. While this growth has enabled researchers to easily access more scientific information, it has also made it quite difficult for them to identify articles more pertinent to their interests. Thus, processing and mining this massive amount of text is of great interest to researchers.

ACKNOWLEDGMENTS

This project was funded in part by Federal funds from the US National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services under contract #HHSN272201200031C, which supports the Malaria Host-Pathogen Interaction Center (MaHPIC).

8 CONFLICT OF INTEREST

The author(s) declare(s) that there is no conflict of interest regarding the publication of this article.

REFERENCES

[1] Charu C Aggarwal and ChengXiang Zhai. 2012. *Mining text data.* Springer.

[2] Mehdi Allahyari and Krys Kochut. 2015. Automatic topic labeling using ontology-based topic models. In *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on.* IEEE, 259–264.

[3] Mehdi Allahyari and Krys Kochut. 2016. Discovering Coherent Topics with Entity Topic Models. In *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on.* IEEE, 26–33.

[4] Mehdi Allahyari and Krys Kochut. 2016. Semantic Context-Aware Recommendation via Topic Models Leveraging Linked Open Data. In *International Conference on Web Information Systems Engineering.* Springer, 263–277.

[5] Mehdi Allahyari and Krys Kochut. 2016. Semantic Tagging Using Topic Models Exploiting Wikipedia Category Network. In *Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on.* IEEE, 63–70.

[6] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut. 2017. Text Summarization Techniques: A Brief Survey. *ArXiv e-prints* (2017). arXiv:1707.02268

[7] Khaled Alsabti, Sanjay Ranka, and Vineet Singh. 1997. An efficient k-means clustering algorithm. (1997).

[8] Sophia Ananiadou, Sampo Pyysalo, Jun'ichi Tsujii, and Douglas B Kell. 2010. Event extraction for systems biology by text mining the literature. *Trends in biotechnology* 28, 7 (2010), 381–390.

[9] Peter G Anick and Shivakumar Vaithyanathan. 1997. Exploiting clustering and phrases for context-based information retrieval. In *ACM SIGIR Forum*, Vol. 31. ACM, 314–323.

[10] Sofia J Athenikos and Hyoil Han. 2010. Biomedical question answering: A survey. *Computer methods and programs in biomedicine* 99, 1 (2010), 1–24.

[11] L Douglas Baker and Andrew Kachites McCallum. 1998. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 96–103.

[12] Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter. 2001. On feature distributional clustering for text categorization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 146–153.

[13] Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing.* Association for Computational Linguistics, 194–201.

[14] David M Blei, Thomas L Griffiths, Michael I Jordan, and Joshua B Tenenbaum. 2003. Hierarchical Topic Models and the Nested Chinese Restaurant Process.. In *NIPS*, Vol. 16.

[15] David M Blei and Jon D McAuliffe. 2007. Supervised Topic Models.. In *NIPS*, Vol. 7. 121–128.

[16] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3 (2003), 993–1022.

[17] Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research* 32, suppl 1 (2004), D267–D270.

[18] Paul S Bradley and Usama M Fayyad. 1998. Refining Initial Points for K-Means Clustering.. In *ICML*, Vol. 98. Citeseer, 91–99.

[19] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees.* CRC press.

[20] Markus Bundschus, Mathaeus Dejori, Martin Stetter, Volker Tresp, and Hans-Peter Kriegel. 2008. Extraction of semantic biomedical relations from text using conditional random fields. *BMC bioinformatics* 9, 1 (2008), 207.

[21] Christopher JC Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery* 2, 2 (1998), 121–167.

[22] Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. 2003. Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery* 7, 4 (2003), 399–424.

[23] Bob Carpenter. 2010. *Integrating out multinomial parameters in latent Dirichlet allocation and naive bayes for collapsed Gibbs sampling.* Technical Report. Technical report, LingPipe.

[24] Soumen Chakrabarti, Byron Dom, Rakesh Agrawal, and Prabhakar Raghavan. 1997. Using taxonomy, discriminants, and signatures for navigating in text databases. In *VLDB*, Vol. 97. 446–455.

[25] Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics.* Association for Computational Linguistics, 152–160.

[26] Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1.* Association for Computational Linguistics, 551–560.

[27] Chaitanya Chemudugunta, America Holloway, Padhraic Smyth, and Mark Steyvers. 2008. Modeling documents by combining semantic concepts with unsupervised statistical learning. In *The Semantic Web-ISWC 2008.* Springer, 229–244.

[28] Ming-Syan Chen, Jiawei Han, and Philip S. Yu. 1996. Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and data Engineering* 8, 6 (1996), 866–883.

[29] Hai Leong Chieu and Hwee Tou Ng. 2003. Named Entity Recognition with a Maximum Entropy Approach. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4 (CONLL '03).* Association for Computational Linguistics, Stroudsburg, PA, USA, 160–163. https://doi.org/10.3115/1119176.1119199

[30] Aaron M Cohen and William R Hersh. 2005. A survey of current work in biomedical text mining. *Briefings in bioinformatics* 6, 1 (2005), 57–71.

[31] K Bretonnel Cohen and Lawrence Hunter. 2008. Getting started in text mining. *PLoS computational biology* 4, 1 (2008), e20.

[32] Nigel Collier, Chikashi Nobata, and Jun-ichi Tsujii. 2000. Extracting the names of genes and gene products with a hidden Markov model. In *Proceedings of the 18th conference on Computational linguistics-Volume 1.* Association for Computational Linguistics, 201–207.

[33] Peter Corbett and Ann Copestake. 2008. Cascaded classifiers for confidence-based chemical named entity recognition. *BMC bioinformatics* 9, Suppl 11 (2008), S4.

[34] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[35] Jim Cowie and Wendy Lehnert. 1996. Information extraction. *Commun. ACM* 39, 1 (1996), 80–91.

[36] Douglass R Cutting, David R Karger, and Jan O Pedersen. 1993. Constant interaction-time scatter/gather browsing of very large document collections. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 126–134.

[37] Douglass R Cutting, David R Karger, Jan O Pedersen, and John W Tukey. 1992. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research*

and development in information retrieval. ACM, 318–329.

[38] Dipanjan Das and André FT Martins. 2007. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU* 4 (2007), 192–195.

[39] Mahmood Doroodchi, Azadeh Iranmehr, and Seyed Amin Pouriyeh. 2009. An investigation on integrating XML-based security into Web services. In *GCC Conference & Exhibition, 2009 5th IEEE*. IEEE, 1–5.

[40] Harris Drucker, S Wu, and Vladimir N Vapnik. 1999. Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on* 10, 5 (1999), 1048–1054.

[41] Richard O Duda, Peter E Hart, and David G Stork. 2012. *Pattern classification*. John Wiley & Sons.

[42] S Dumais, G Furnas, T Landauer, S Deerwester, S Deerwester, et al. 1995. Latent semantic indexing. In *Proceedings of the Text Retrieval Conference*.

[43] Sašo Džeroski. 2009. Relational data mining. In *Data Mining and Knowledge Discovery Handbook*. Springer, 887–911.

[44] Christos Faloutsos and Douglas W Oard. 1998. *A survey of information retrieval and filtering methods*. Technical Report.

[45] Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, et al. 1996. Knowledge Discovery and Data Mining: Towards a Unifying Framework.. In *KDD*, Vol. 96. 82–88.

[46] Ronen Feldman and Ido Dagan. 1995. Knowledge Discovery in Textual Databases (KDT).. In *KDD*, Vol. 95. 112–117.

[47] Guozhong Feng, Jianhua Guo, Bing-Yi Jing, and Lizhu Hao. 2012. A Bayesian feature selection paradigm for text classification. *Information Processing & Management* 48, 2 (2012), 283–302.

[48] William J Frawley, Gregory Piatetsky-Shapiro, and Christopher J Matheus. 1992. Knowledge discovery in databases: An overview. *AI magazine* 13, 3 (1992), 57.

[49] Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55, 1 (1997), 119–139.

[50] Mark A Friedl and Carla E Brodley. 1997. Decision tree classification of land cover from remotely sensed data. *Remote sensing of environment* 61, 3 (1997), 399–409.

[51] Robert Gaizauskas, George Demetriou, Peter J. Artymiuk, and Peter Willett. 2003. Protein structures and information extraction from biological texts: the PASTA system. *Bioinformatics* 19, 1 (2003), 135–143.

[52] John Gantz and David Reinsel. 2012. *THE DIGITAL UNIVERSE IN 2020: Big Data, Bigger Digital Shadows, and Biggest Grow th in the Far East*. Technical Report 1. IDC, 5 Speen Street, Framingham, MA 01701 USA. Accessed online on May, 2017. https://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf.

[53] Walter R Gilks, Sylvia Richardson, and David J Spiegelhalter. 1996. Introducing markov chain monte carlo. In *Markov chain Monte Carlo in practice*. Springer, 1–19.

[54] Siddharth Gopal and Yiming Yang. 2010. Multilabel classification with meta-level features. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 315–322.

[55] Thomas L Griffiths and Mark Steyvers. 2002. A probabilistic approach to semantic representation. In *Proceedings of the 24th annual conference of the cognitive science society*. Citeseer, 381–386.

[56] Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America* 101, Suppl 1 (2004), 5228–5235.

[57] Serkan Günal, Semih Ergin, M Bilginer Gülmezoğlu, and Ö Nezih Gerek. 2006. On feature extraction for spam e-mail detection. In *Multimedia Content Representation, Classification and Security*. Springer, 635–642.

[58] Pritam Gundecha and Huan Liu. 2012. Mining social media: a brief introduction. In *New Directions in Informatics, Optimization, Logistics, and Production*. Informs, 1–17.

[59] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 427–434.

[60] Juan B Gutierrez, Mary R Galinski, Stephen Cantrell, and Eberhard O Voit. 2015. From within host dynamics to the epidemiology of infectious disease: scientific overview and challenges. (2015).

[61] Eui-Hong Sam Han, George Karypis, and Vipin Kumar. 2001. *Text categorization using weight adjusted k-nearest neighbor classification*. Springer.

[62] Jiawei Han, Micheline Kamber, and Jian Pei. 2006. *Data mining: concepts and techniques*. Morgan kaufmann.

[63] Xianpei Han and Le Sun. 2012. An entity-topic model for entity linking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 105–115.

[64] Gregor Heinrich. 2005. *Parameter estimation for text analysis*. Technical Report. Technical report.

[65] William Hersh. 2008. *Information retrieval: a health and biomedical perspective*. Springer Science & Business Media.

[66] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 50–57.

[67] Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß. 2005. A Brief Survey of Text Mining.. In *Ldv Forum*, Vol. 20. 19–62.

[68] David A Hull et al. 1996. Stemming algorithms: A case study for detailed evaluation. *JASIS* 47, 1 (1996), 70–84.

[69] Hideki Isozaki and Hideto Kazawa. 2002. Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1–7.

[70] Anil K Jain and Richard C Dubes. 1988. *Algorithms for clustering data*. Prentice-Hall, Inc.

[71] Mike James. 1985. *Classification algorithms*. Wiley-Interscience.

[72] Jing Jiang and ChengXiang Zhai. 2007. A Systematic Exploration of the Feature Space for Relation Extraction.. In *HLT-NAACL*. 113–120.

[73] Thorsten Joachims. 1996. *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization*. Technical Report. DTIC Document.

[74] Thorsten Joachims. 1998. *Text categorization with support vector machines: Learning with many relevant features*. Springer.

[75] Thorsten Joachims. 2001. A statistical learning learning model of text classification for support vector machines. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 128–136.

[76] Tom Kalt and WB Croft. 1996. *A new probabilistic model of text classification and retrieval*. Technical Report. Citeseer.

[77] Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, 22.

[78] Mehmed Kantardzic. 2011. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons.

[79] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. 2002. An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24, 7 (2002), 881–892.

[80] Anne Kao and Stephen R Poteet. 2007. *Natural language processing and text mining*. Springer.

[81] Saurabh S Kataria, Krishnan S Kumar, Rajeev R Rastogi, Prithviraj Sen, and Srinivasan H Sengamedu. 2011. Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1037–1045.

[82] Leonard Kaufman and Peter J Rousseeuw. 2009. *Finding groups in data: an introduction to cluster analysis*. Vol. 344. John Wiley & Sons.

[83] Jun'ichi Kazama, Takaki Makino, Yoshihiro Ohta, and Jun'ichi Tsujii. 2002. Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain-Volume 3*. Association for Computational Linguistics, 1–8.

[84] Daphne Koller and Mehran Sahami. 1997. Hierarchically classifying documents using very few words. (1997).

[85] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).

[86] Leah S Larkey and W Bruce Croft. 1996. Combining classifiers in text categorization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 289–297.

[87] Ulf Leser and Jörg Hakenberg. 2005. What makes a gene name? Named entity recognition in the biomedical literature. *Briefings in Bioinformatics* 6, 4 (2005).

[88] David D Lewis. 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. In *Machine learning: ECML-98*. Springer, 4–15.

[89] Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1–7.

[90] Elizabeth D Liddy. 2001. Natural language processing. (2001).

[91] Anthony ML Liekens, Jeroen De Knijf, Walter Daelemans, Bart Goethals, Peter De Rijk, and Jurgen Del-Favero. 2011. BioGraph: unsupervised biomedical knowledge discovery via automated hypothesis generation. *Genome biology* 12, 6 (2011), R57.

[92] Julie B Lovins. 1968. *Development of a stemming algorithm*. MIT Information Processing Group, Electronic Systems Laboratory.

[93] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Vol. 1. Cambridge university press Cambridge.

[94] Christopher D Manning, Hinrich Schütze, et al. 1999. *Foundations of statistical natural language processing*. Vol. 999. MIT Press.

[95] AT Mc Cray. 1993. A. The Unified Medical Language System. *Meth Inf Med* 34 (1993), 281–291.

[96] Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, Vol. 752. Citeseer, 41–48.

[97] Andrew McCallum, Ronald Rosenfeld, Tom M Mitchell, and Andrew Y Ng. 1998. Improving Text Classification by Shrinkage in a Hierarchy of Classes.. In *ICML*, Vol. 98. 359–367.

[98] Andrew Kachites McCallum. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. (1996).

[99] Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. (2002).

[100] David Mimno, Wei Li, and Andrew McCallum. 2007. Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of the 24th international conference on Machine learning*. ACM, 633–640.

[101] Tom M Mitchell. 1997. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill* 45 (1997).

[102] Tomohiro Mitsumori, Sevrani Fation, Masaki Murata, Kouichi Doi, and Hirohumi Doi. 2005. Gene/protein name recognition based on support vector machine using dictionary as features. *BMC bioinformatics* 6, Suppl 1 (2005), S8.

[103] Fionn Murtagh. 1983. A survey of recent advances in hierarchical clustering algorithms. *Comput. J.* 26, 4 (1983), 354–359.

[104] Fionn Murtagh. 1984. Complexities of hierarchic clustering algorithms: State of the art. *Computational Statistics Quarterly* 1, 2 (1984), 101–113.

[105] Amir Netz, Surajit Chaudhuri, Jeff Bernhardt, and Usama Fayyad. 2000. Integration of data mining and relational databases. In *Proceedings of the 26th International Conference on Very Large Databases, Cairo, Egypt*. 285–296.

[106] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. 1998. Learning to classify text from labeled and unlabeled documents. *AAAI/IAAI* 792 (1998).

[107] Chikashi Nobata, Nigel Collier, and Jun-ichi Tsujii. 1999. Automatic term identification and classification in biology texts. In *Proc. of the 5th NLPRS*. Citeseer, 369–374.

[108] Edgar Osuna, Robert Freund, and Federico Girosi. 1997. Training support vector machines: an application to face detection. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE, 130–136.

[109] Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval* 2, 1-2 (2008), 1–135.

[110] Martin F Porter. 1980. An algorithm for suffix stripping. *Program: electronic library and information systems* 14, 3 (1980), 130–137.

[111] Seyed Amin Pouriyeh and Mahmood Doroodchi. 2009. Secure SMS Banking Based On Web Services. In *SWWS*. 79–83.

[112] Seyed Amin Pouriyeh, Mahmood Doroodchi, and MR Rezaeinejad. 2010. Secure Mobile Approaches Using Web Services.. In *SWWS*. 75–78.

[113] J. Ross Quinlan. 1986. Induction of decision trees. *Machine learning* 1, 1 (1986), 81–106.

[114] Lawrence Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286.

[115] Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the special issue on summarization. *Computational linguistics* 28, 4 (2002), 399–408.

[116] Martin Rajman and Romaric Besançon. 1998. Text mining: natural language techniques and text mining applications. In *Data Mining and Reverse Engineering*. Springer, 50–64.

[117] Payam Porkar Rezaeiye, Mojtaba Sedigh Fazli, et al. 2014. Use HMM and KNN for classifying corneal data. *arXiv preprint arXiv:1401.7486* (2014).

[118] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. 1998. A Bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 workshop*, Vol. 62. 98–105.

[119] Hassan Saif, Miriam Fernández, Yulan He, and Harith Alani. 2014. On stopwords, filtering and data sparsity for sentiment analysis of twitter. (2014).

[120] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.

[121] Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (1975), 613–620.

[122] Sunita Sarawagi et al. 2008. Information extraction. *Foundations and Trends® in Databases* 1, 3 (2008), 261–377.

[123] Yutaka Sasaki, Yoshimasa Tsuruoka, John McNaught, and Sophia Ananiadou. 2008. How to make the most of NE dictionaries in statistical NER. *BMC bioinformatics* 9, Suppl 11 (2008), S5.

[124] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association* 17, 5 (2010), 507–513.

[125] Robert E Schapire and Yoram Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine learning* 39, 2-3 (2000), 135–168.

[126] Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)* 34, 1 (2002), 1–47.

[127] Prithviraj Sen. 2012. Collective context-aware topic models for entity disambiguation. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 729–738.

[128] Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*. Association for Computational Linguistics, 104–107.

[129] Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2003. Effective adaptation of a hidden markov model-based named entity recognizer for biomedical domain. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine-Volume 13*. Association for Computational Linguistics, 49–56.

[130] Catarina Silva and Bernardete Ribeiro. 2003. The importance of stop word removal on recall values in text categorization. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, Vol. 3. IEEE, 1661–1666.

[131] Amit Singhal. 2001. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.* 24, 4 (2001), 35–43.

[132] Michael Steinbach, George Karypis, Vipin Kumar, et al. 2000. A comparison of document clustering techniques. In *KDD workshop on text mining*, Vol. 400. Boston, 525–526.

[133] Mark Steyvers and Tom Griffiths. 2007. Probabilistic topic models. *Handbook of latent semantic analysis* 427, 7 (2007), 424–440.

[134] Koichi Takeuchi and Nigel Collier. 2005. Bio-medical entity extraction using support vector machines. *Artificial Intelligence in Medicine* 33, 2 (2005), 125–137.

[135] Songbo Tan, Yuefen Wang, and Gaowei Wu. 2011. Adapting centroid classifier for document categorization. *Expert Systems with Applications* 38, 8 (2011), 10264–10273.

[136] E. D. Trippe, J. B. Aguilar, Y. H. Yan, M. V. Nural, J. A. Brady, M. Assefi, S. Safaei, M. Allahyari, S. Pouriyeh, M. R. Galinski, J. C. Kissinger, and J. B. Gutierrez. 2017. A Vision for Health Informatics: Introducing the SKED Framework An Extensible Architecture for Scientific Knowledge Extraction from Data. *ArXiv e-prints* (2017). arXiv:1706.07992

[137] Stéphane Tufféry. 2011. *Data mining and statistics for decision making*. John Wiley & Sons.

[138] Howard Turtle and W Bruce Croft. 1989. Inference networks for document retrieval. In *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1–24.

[139] Yu Usami, Han-Cheol Cho, Naoaki Okazaki, and Jun'ichi Tsujii. 2011. Automatic acquisition of huge training data for bio-medical named entity recognition. In *Proceedings of BioNLP 2011 Workshop*. Association for Computational Linguistics, 65–73.

[140] Alper Kursat Uysal and Serkan Gunal. 2014. The impact of preprocessing on text classification. *Information Processing & Management* 50, 1 (2014), 104–112.

[141] Vladimir Vapnik. 1982. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[142] Vladimir Vapnik. 2000. *The nature of statistical learning theory*. springer.

[143] Jonathan J Webster and Chunyu Kit. 1992. Tokenization as the initial phase in NLP. In *Proceedings of the 14th conference on Computational linguistics-Volume 4*. Association for Computational Linguistics, 1106–1110.

[144] Peter Willett. 1988. Recent trends in hierarchic document clustering: a critical review. *Information Processing & Management* 24, 5 (1988), 577–597.

[145] Rui Xu, Donald Wunsch, et al. 2005. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on* 16, 3 (2005), 645–678.

[146] Christopher C Yang, Haodong Yang, Ling Jiang, and Mi Zhang. 2012. Social media mining for drug safety signal detection. In *Proceedings of the 2012 international workshop on Smart health and wellbeing*. ACM, 33–40.

[147] Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 42–49.

[148] Illhoi Yoo and Min Song. 2008. Biomedical Ontologies and Text Mining for Biomedicine and Healthcare: A Survey. *JCSE* 2, 2 (2008), 109–136.

[149] Liyang Yu. 2011. *A developer's guide to the semantic Web*. Springer.

[150] Shaojun Zhao. 2004. Named entity recognition in biomedical texts using an HMM model. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*. Association for Computational Linguistics, 84–87.

[151] Ying Zhao and George Karypis. 2004. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning* 55, 3 (2004), 311–331.