# Semi Supervised Classification of Web Content using Mixture Models

Roxana K. Aparicio Carrasco, Ph.D.
University of Puerto Rico
Río Piedras Campus

## ABSTRACT

Automatic classification of web documents has become an important matter due the proliferation of online documents, social media, microblogs, discussion forums and multimedia sharing sites. Applications of web classification are diverse: email filtering, online news filtering, web log classification, social media analytics, opinion extraction and semantic classification of product reviews, and more.

Many modern applications of automatic Web document classification require learning accurately with little training data. Addressing the need to reduce the manual labeling process, the semi-supervised classification technique has been proposed. This technique use labeled and unlabeled data for training. On the other hand, the emergence of web technologies has originated the collaborative development of ontologies. Ontologies are formal, explicit, detailed structures of concepts.

This paper investigates semi-supervised learning of mixture models using EM algorithm for Web content classification. We also explore the use of Ontologies in order to take advantage of domain knowledge to support the classification process.

# 1 INTRODUCTION

Automatic document classification has become an important subject due the proliferation of electronic text documents in the last years. This problem consists in learning to classify unseen documents into previously defined categories. The importance of making an automatic Web document classification is noticeable in many practical applications: email filtering (Gomez et al., 2012), online news filtering (Chan et al., 2001), web log classification (Yu et al., 2005), social media analytics (Melville et al., 2009), (Bandari et al., 2012), (Paltoglou et al., 2012), (Volkova et al., 2013), opinion extraction and semantic classification of product reviews (Dave et al., 2003) etc.

Supervised learning methods construct a classifier with a training set of documents. This classifier could be seen as a function or decision rule that is used for classifying future documents into previously defined categories. Supervised text classification algorithms have been successfully used in a wide variety of practical domains. The problem with supervised learning methods is that they require a large number of labeled training examples to learn accurately. Manual labeling is a costly and time-consuming process, since it requires human effort. On the other hand, there exist many unlabeled documents readily available, and it has been proved that in the document classification context, the use of unlabeled documents for training could benefit the classification process (Nigam, 1998). In particular, they represent the text as a mixture of multinomials and used Expectation-Maximization (EM) with Naïve Bayes to train the classifier.

Simultaneously, with the advances of web technologies, ontologies have increased on the World-Wide Web. Ontologies represent shared knowledge as a set of concepts within a domain, and the relationships between those concepts (Lacy, 2005). The ontologies on the Web range from large taxonomies categorizing Web sites to categorizations of products for sale and their features. They can be used to reason about the entities within that domain, and may be used to describe the domain. We propose the use of ontologies in order to assist the semi-supervised classification using EM with Naive Bayes.

# 2 WEB MINING

Web mining is the application of data mining techniques to discover patterns from the Web. Web mining presents more challenges than traditional text and data mining process (Munibalaji et al., 2012), (Bhatia, 2011):

- The amount of web documents available for learning is enormous.
- The coverage of web information is very wide and diverse.
- Much of web information is semi structured or semi-structured and heterogeneous (such as email, full-text documents, XML files and HTML files) (Fan, 2006).
- Information in the Web is linked.
- Much information in the Web is redundant.
- The web is noisy.
- The web is dynamic.

- The web is a virtual society.

## 2.1 Web Mining Categories

Web mining can be divided into three categories: Web content mining, Web usage mining and Web structure mining.

### *2.1.1 Web Content Mining*

Web content mining is extraction and integration of useful data, information and knowledge from Web page content. This category can also be divided into Web Text Mining and Web Multimedia Mining.

### *2.1.2 Web Usage Mining*

Web usage mining is the process of extracting information from server logs. Web usage mining can predict the behavior of users while they are interacting with the WWW. (Munibalaji et al., 2012).

### *2.1.3 Web Structure Mining*

Web structure mining is the process of analyze the connection structure of a web site. According to the type of web structural data, web structure mining can be divided into two kinds: (Bhatia, 2011), (Munibalaji et al., 2012).

1. Extracting patterns from hyperlinks in the web: a hyperlink is a structural component that connects the web page to a different location.

2. Mining the document structure: analysis of the tree-like structure of page structures to describe HTML or XML tag usage.

In this work we are interested in the Web content classification when the content is text.

## 2.2 Text Mining for Web Documents

Text mining, sometimes called Knowledge Discovery from Text (KDT), is the process of automatically analyzing text documents from different perspectives and providing useful information from them.

Starting with a collection of documents, a text-mining tool retrieves a particular document and preprocesses it. In order to run their knowledge discovery algorithms, text mining systems require to transform raw, unstructured, original-format content into structured data format (Konchady, 2006). Preprocessing is a major step in text mining compared to data mining since it involves significant processing steps for transforming a text into a structured format suitable for later analysis (Feldman et al., 2007)  as shown in Figure 2.1. Once we have structured data, we are ready to perform a text mining task to generate knowledge.

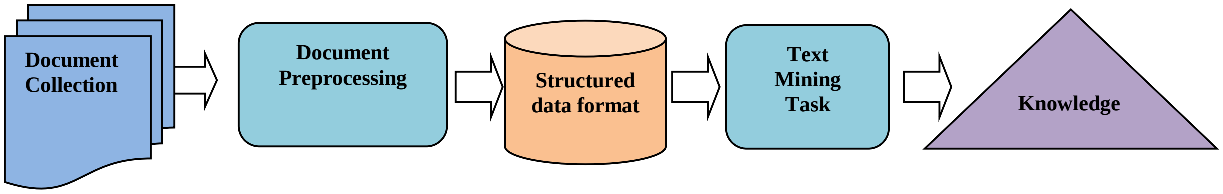

**Figure 2.1 Text mining Process**

In the following we will describe the preprocessing steps for the transformation of unstructured techniques for the transformation of unstructured text into structured formats.

### *2.2.1 Tokenization*

The first step in handling text is to break the stream of characters into words or, more precisely, tokens (Weiss, 2004). As defined in Konchady (Konchady, 2006), a token is a word, number, punctuation mark, or any other sequence of characters that should be treated as a single unit.

The importance of tokenization is sometimes overlooked since it appears to be a simple task. But the accurate extraction of tokens is important for precise results in higher-level applications. Vector representations of documents used in clustering and text categorization are made up of a sequence of tokens and weights. Documents can be correctly categorized only when the vector representatives accurately the contents of documents (Konchady, 2006).

### *2.2.2 Lemmatization or stemming*

Stemming consists of converting each word to its stem. In essence, to get the stem of a word it is necessary to eliminate its suffixes representing tag-of-speech and/or verbal/plural inflections. For instance, the words "taller" and "tallest" would both be converted to their stem "tall" (Larocca, 2000).

Whether or not this step is necessary is application-dependent. One effect of stemming is to reduce the number of distinct types in a text corpus and to increase the frequency of occurrence of some individual types. Stemming algorithms usually incorporate a great deal of linguistic knowledge, so that they are language-dependent.

### *2.2.3 Vectorization*

Vector based representations has been widely used in text mining process for their simplicity. They are also referred to as a ‘bag of words’, emphasizing that document vectors are invariant with respect to term permutations, since the original word order in the document is clearly lost. Though, many text retrieval and categorization tasks can be performed quite well in practice using the vector-space model.

The collective set of tokens or words is typically called a dictionary or vocabulary (*V*). They form the basis for creating the numeric vectors corresponding to the document collection.

More precisely, a text document $d$ can be represented as a sequence of terms, $d = (w_1, w_2, \dots, w_{|d|})$, where $|d|$ is the length of the document and $w_t \in V$. A vector-space representation of $d$ is then defined as a real vector $x \in R^{|V|}$, where each component $x_j$ is a statistic related to the occurrence of the $j^{th}$ vocabulary entry in the document.

Note that typically the total number of terms in a set of documents is much larger than the number of distinct terms in any single document, |V|>>|d|, so that vector-space representations tend to be very sparse. This property can be advantageously exploited for both memory storage and algorithm design. The common vector-based representations are described below:

#### *Term Frequency*

Term frequency consists of counting the actual number of occurrences of each term in the document. This value may be multiplied by the constant $\frac{1}{|d|}$ to obtain a vector of term frequencies (TF) within the document. (Weiss, 2004).

Let $D = \{d_1, d_2, \dots, d_n\}$ be a collection of documents. For each term $w_j \in V$, let $n_{ij}$ denote the number of occurrences of $w_j$ in document $d_i$.Then we define:

$$TF_{ij} = \frac{n_{ij}}{|d_i|}$$

### *Inverse document frequency*

While term frequencies are relative to each document, inverse document frequency (IDF) is an ‘absolute' measure of term importance. IDF decreases as the number of documents in which the term occurs increases in a given collection. So terms that are globally rare receive a higher weight.

Let $D = \{d_1, d_2, \dots, d_n\}$ be a collection of documents and $w_j \in V$. Let $n_j$ be the number of documents that contain $w_j$ at least once. Then we define:

$$IDF_j = log \frac{n}{n_j}$$

The logarithmic function is employed as a damping factor.

### *The TF-IDF weight*

An important family of weighting schemes combines term frequencies with inverse document frequency. Let $\boldsymbol{x} = (x_{ij})$ be the vector representation of the TF-IDF weight of $w_j \in V$ in $d_i$ can be computed as:

$$x_{ij} = TF_{ij}.IDF_j$$

Alternative versions of the basic TF-IDF exist in which the general motivation is the same (Weiss, 2004).

# 3 WEB CONTENT CLASSIFICATION

## 3.1 Supervised classification

Automatic document classification consists in learning to classify unseen documents into previously defined categories. Given a collection of text documents and a set of categories, the task is to learn to predict the category for an unseen document.

We can describe supervised document classification as an automatic process with two phases:

### *Learning Phase*

In the learning phase the system takes as its input a set of documents, which have been previously labeled, and learns a function $\boldsymbol{f}$ from them. This assignment function is called a **classifier**. The labels assigned to the training documents belongs to a predefined set of categories C. Formally, $f: D \times C \rightarrow \{0,1\}$ where D is the set of all possible documents and

C is the set of predefined categories. The value $f(d, c)$ is 1 if the document $d$ belongs to the category $c$ and 0 otherwise.

### *Prediction Phase*

In the prediction phase a new unlabeled document is presented to the system and it assigns a label according to the classifier it has learned.

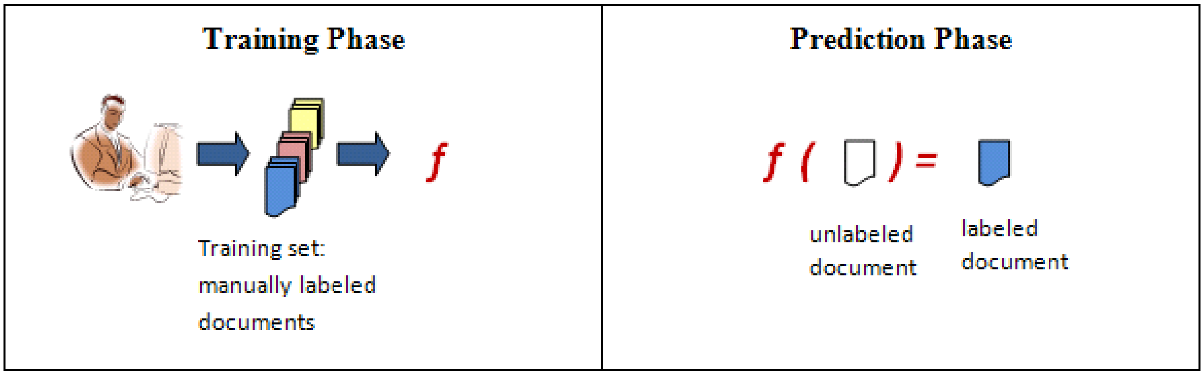

**Figure 3.1 Supervised classification process**

The practical applications of supervised text classification are extensive. They vary from automatic email sorting (or specifically filtering spam emails) (Sahami, 1998), sentiment detection of a text or opinion mining (Pang, 2002), classification of news articles (Chan, 2001), classification of the e-commerce customer logs/notes (Yu, 2005), detecting the document language (English, Turkish, etc.) (Feinerer, 2008), etc .

Supervised text classification algorithms have been successfully used in a wide variety of practical domains. In experiments conducted by Namburú et al. (Namburu, 2005), using high accuracy classifiers with the most widely used document datasets, they report up to 96% of accuracy with a binary classification in the Reuters dataset. However, they needed 2000 manually labeled documents to achieve this good result.

The problem with the supervised learning methods, is that they require a large number of labeled training examples to learn accurately. Manual labeling is a costly and time-consuming process, since it requires human effort. In some applications, this approach becomes impractical, since most users would not have time to spend in label thousands of documents (Nigam, 2001).

## 3.2 Unsupervised classification

Unsupervised document classification, also known as document clustering, is a process through which documents are classified into meaningful groups called clusters, without any prior information.

A clustering task may include a definition of proximity or similarity measure suitable to the domain. There are many possible similarity measures, however, the cosine similarity measure is the most common for the text clustering: Let *x* and *y* vector representations of two documents,

$$Sim(x,y) =< \bar{x}, \bar{y} >= \sum_{k} \bar{x}_k, \bar{y}_k$$

where $\bar{x}$ is the normalized vector $\bar{x} = \frac{x}{\|x\|}$ . (Feldman, 2007)

An unsupervised learning system takes as its input a collection of unlabeled documents. The system classifies documents according to a similarity measure and generates clusters of documents which are similar with certain probability. This description is depicted in Figure 3.2.

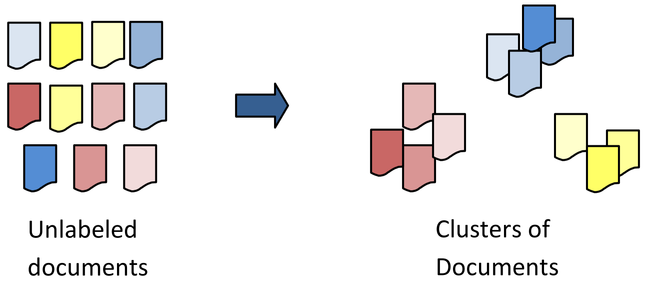

**Figure 3.2 Unsupervised classification process**

## 3.3 Semi-supervised Document Classification

The general idea of semi-supervised learning is to use a small number of labeled examples and a large number of unlabeled examples to achieve high-accuracy classification. The motivation for the use of unlabeled documents for text classification is that we have many electronic documents readily available. But, labeling the documents must typically be done by a person, which is a costly and time-consuming process. Nigam et al. showed that, in certain circumstances, it is possible to train a system using both unlabeled and labeled documents and explained why unlabeled data could benefit the classification task (Nigam, 2001).

### *3.3.1 The value of unlabeled data*

An intuitive idea given by Nigam can make us understand why unlabeled data can be helpful. "Suppose we have some web pages about academic courses, along with a large number of web pages that are unlabeled. By looking at just the labeled data we determine that pages containing the word homework tend to be about academic courses. If we use this fact to estimate the classification of the many unlabeled web pages, we might find that the word lecture occurs frequently in the unlabeled examples that are now believed to belong to the positive class. This co-occurrence of the words homework and lecture over the large set of unlabeled training data can provide useful information to construct a more accurate classifier that considers both homework and lecture as indicators of positive examples" (Nigam, 2001).

### *3.3.2 Semi-supervised Expectation Maximization with Naive Bayes*

The model considered in (Nigam, 2001) uses an algorithm for learning from labeled and unlabeled documents based on the combination of Expectation-Maximization (EM) and the naive Bayes classifier.

#### 3.3.2.1 The probabilistic model

In order to model the data, assume that documents are generated by a mixture of multinomial distributions model, where each mixture component corresponds to a class.

Suppose that C is the number of classes, the vocabulary is of size $|V|$, and each document $d_i$ has $|d_i|$ words in it.

The likelihood of seeing document *d<sub>i</sub>* is a sum of total probability over all mixture components. That is,

$$P(d_i|\theta) = \sum_{j=1}^{C} P(c_j|\theta)P\left(d_i\middle|c_j;\theta\right) \qquad (1)$$

Using the above along with standard Naive Bayes assumption: that the words of a document are conditionally independent among them, given the class label, we can expand the second term of equation 1, and express the probability of a document given a mixture component in terms of its constituent features: the document length and the words in the document.

$$P\left(d_i\middle|c_j;\theta\right) \approx P(|d_i|) \prod_{w_t \in V} P(w_t|c_j;\theta)^{N_{it}} \qquad (2)$$

Where $N_{it}$ refers to the number of times word $w_t$ occurs in document $d_i$.

The full generative model, given by combining equations 1 and 2, assigns probability $P(d_i|\Theta)$ to generating document $d_i$ as follows:

$$P(d_i|\theta) \approx P(|d_i|) \sum_{j=1}^{C} P(c_j|\theta) \prod_{w_t \in V} P(w_t|c_j;\theta)^{N_{it}} \qquad (3)$$

### *Dirichlet distribution*

Let $p = (p_1, \dots, p_k)$ a random vector such that $\sum_{i=1}^{k} p_i = 1,\ 0 < p_i < 1\,, i = 1, \dots, k.$

The Dirichlet distribution with parameters $\alpha_1, \alpha_2, \dots, \alpha_k$ is given by:

$$P(p|\alpha_1, \dots \alpha_k) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} p_k^{\alpha_k - 1} \qquad (3)$$

Where an $\alpha$ with large components correspond to strong prior knowledge about the distribution and $\alpha$ with small components correspond to ignorance.

Using **maximum a posteriori** (MAP) to estimate the parameters of a multinomial distribution with Dirichlet prior, yields:

$$\hat{\theta}_{w_t|c_j} \equiv P(w_t|c_j;\theta) = \frac{1+\sum_{i=1}^{|D|}\delta_{ij}N_{it}}{|V|+\sum_{s=1}^{|V|}\sum_{i=1}^{|D|}\delta_{ij}N_{is}} \tag{4}$$

$$\hat{\theta}_{c_j} \equiv P(c_j|\theta) = \frac{1+\sum_{i=1}^{|D|}\delta_{ij}}{C+|V|} \tag{5}$$

Given estimates of these parameters, it is possible to calculate the probability that a particular mixture component generated a given document to perform classification. By applying Bayes rule it follows that:

$$P\left(y_i = c_j \middle| d_i;\hat{\theta}\right) \approx \frac{P(c_j|\hat{\theta})\prod_{w_t\in V}P(w_t|c_j;\hat{\theta})^{N_{it}}}{\sum_{k=1}^{C}P(c_k|\hat{\theta})\prod_{w_t\in V}P(w_t|c_j;\hat{\theta})^{N_{it}}} \tag{6}$$

Then, to classify a test document into a single class, the class with the highest posterior probability is selected.

#### 3.3.2.2 EM Semi-supervised algorithm

When there exist unlabeled data, we would still like to find MAP parameter estimates, as in the supervised setting above. Using the Expectation-Maximization (EM) technique, we can find locally MAP parameter estimates for the generative model.

The probability of an individual unlabeled document is a sum of total probability over all the classes, as in Equation 1. Hence, the expected log probability of the data, containing $|D|$, is:

$$l(\theta|D,Y) = log(P(\theta)) + \sum_{i=1}^{|D|} log \sum_{j=1}^{C} P(c_j|\theta)P\left(d_i\middle|c_j;\theta\right) \tag{7}$$

The Expectation-Maximization (EM) is a two step process that provides an iterative approach to finding a local maxima of model probability in parameter space. The E-step of the algorithm estimates the expectations of the class given the latest iteration of the model parameters. The M-step maximizes the likelihood of the model parameters using the previously computed expectations of the missing values as if they were the true ones.

In practice, the E-step corresponds to performing classification of each unlabeled document using equation 6. The M-step corresponds to calculating a new maximum a posteriori estimate for the parameters, using equations 4 and 5 with the current estimates.

This algorithm is guaranteed to converge to some local maxima. The algorithm iterates until it converges to a point where the parameters do not change from one iteration to the next.

For the semi-supervised case, we consider using a limited number of labeled data in the initialization step. We first train a classifier using the labeled data, and then estimate the parameters. After that, the algorithm iterates trying to improve the log likelihood of the data.

The algorithm for the semi-supervised document classification is as follows:

```
EM_Semi-supervised_NaiveBayes
Inputs:
  $D_l$= Collection of labeled documents
  $D_u$= Collection of unlabeled documents

1.  Train a classifier with the labeled data and use maximum a posteriori parameter
```

estimation to find θ.

2. Loop while classifier parameters improve, as measured by the change in $l(\vartheta|D,Y)$.
   **(E-step)** Use the current classifier, θ , to estimate component membership of each document, $P(c_j|D_i;\vartheta)$.
   **(M-step)** Re-estimate the classifier, θ , given the estimated component membership of each document. Use MAP estimation to find $\vartheta=argmax_{\vartheta}P(D,Y|\vartheta)P(\vartheta)$.

## 3.4 Ontology

The term 'ontology' in the context of information management is defined as a formal, explicit specification of a shared conceptualization (Gruber, 1993). A conceptualization refers to an abstract model of some phenomenon in the world which identifies the relevant concepts, relations and constraints. These concepts, relations and constraints must be explicitly defined. Formal refers to the fact that the ontology should be machine-readable. And, finally an ontology represents shared knowledge, that is a common understanding of the domain between several parties.

In other words, an ontology specifies a domain theory. It is a formal description of concepts and their relations, together with constraints on those concepts and relations. (Alexiev, 2005).

Ontologies provide an unambiguous terminology that can be shared by all involved in a software development process (Green et al., 2000). Ontologies are widely used in different domains to give standard representation and semantics to concepts, predicates and actions of a particular domain and have been used recently in Web mining applications, among which we can list: ontology recommendation system in E-Commerce (Wang, 2012), ranking web sites using domain ontology concepts (Kayed et al., 2010), ontology-based shopping agent for e-marketing (Chatwin et al., 2010), etc.

## 3.5 Proposed Work

We propose a learning approach that exploits the use of ontologies, in order to assist the semi-supervised document classification task. Using this information we could guide the direction of the use of unlabeled data, respecting the particular method rules. We use the information provided by the ontologies when the learner needs to make a decision, and we give the most probable label when otherwise arbitrary decision is to be made.

Given the nature of web documents we explore the use of co-training to exploit the linkage between web documents, this is particularly important for social networks. We also plan to combine the mining of web content with web link information when the web content is short; this is particularly true for microblogs data.

We plan to use two different data sets: Web KB and Rovereto Twitter N-Gram Corpus (RTC).

The WebKB1 dataset contains Web pages that were collected from the computer science departments of universities. The pages are divided seven categories: student, faculty, staff, course, project, department and other.

RTC is an n-gram dataset based on almost 75 million short, personal, social media posts in English, along with aggregated information on the gender of the authors of the posts and the time of the posting. It was made available by (Herdagdelen et al., 2011). Twitter data is particularly interesting because tweets are available as they happen in almost real time, and represent many different segments of society. Also, as other social networks, Twitter provides valuable information through the links between people and things that they care about (Russell, 2013).

Our expected contributions are as follows:

1. Incorporate the use of ontologies to the semi-supervised learning approach. In particular, we plan to use EM with Naïve Bayes algorithm.
2. Provide an efficient implementation that works accurate and efficiently.
3. Present empirical evaluations.

# BIBLIOGRAPHY

**Alexiev V., Breu, M., de Bruijn, J., Fensel, D., Lara, R. and Lausen, H.** Information Integration with Ontologies: Experiences from an Industrial Showcase [Book]. - [s.l.] : Wiley, 2005.

**Bandari Roja, Asur Sitaram and Huberman Bernardo A** The Pulse of News in Social Media: Forecasting Popularity. [Conference] // ICWSM. - 2012.

**Bhatia Tamanna** Link Analysis Algorithms For Web Mining [Journal] // IJCST. - [s.l.] : Citeseer, 2011. - 2 : Vol. 2.

**Chan C., Sun A. and Lim E.** Automated Online News Classification with Personalization [Conference] // 4th International Conference of Asian Digital Library. - 2001.

**Chan C., Sun, A., Lim, E.** Automated Online News Classification with Personalization [Conference] // 4th International Conference of Asian Digital Library. - 2001.

**Chatwin CR and Meng Sam Kin** Ontology-based shopping agent for e-marketing [Journal] // International Journal of Intelligent Information Technologies. - [s.l.] : IGI Global, 2010. - 2 : Vol. 6. - pp. 21-43.

**Dave Kushal, Lawrence Steve and Pennock David M** Mining the peanut gallery: Opinion extraction and semantic classification of product reviews [Conference] // Proceedings of the 12th international conference on World Wide Web. - 2003. - pp. 519-528.

**Fan W., Wallace, L., Rich, S., Zhang, Z.** Tapping the power of text mining [Article] // Magazine Communications of the ACM - Privacy and security in highly dynamic systems. - September 2006. - 9 : Vol. 49. - pp. 77-82.

**Feinerer I.** A text mining framework in R and its applications [Report] : Doctoral thesis / University of Economics and Business. - Vienna : [s.n.], 2008.

**Feldman R. and Sanger J.** The Text Mining Handbook. Advances Approaches in Analyzing Unstructured Data [Book]. - [s.l.] : Cambridge, 2007.

**Gomez Juan Carlos, Boiy Erik and Moens Marie-Francine** Highly discriminative statistical features for email classification [Journal] // Knowledge and information systems. - [s.l.] : Springer, 2012. - 1 : Vol. 31. - pp. 23-53.

**Green Peter and Rosemann Michael** Integrated process modeling: an ontological evaluation [Journal] // Information systems. - [s.l.] : Elsevier, 2000. - 2 : Vol. 25. - pp. 73-87.

**Gruber T.** A translation approach to portable ontology specifications [Journal] // KNOWLEDGE ACQUISITION. - 1993. - Vol. 5. - pp. 199-220.

**Herdagdelen Amac and Baroni Marco** Stereotypical gender actions can be extracted from web text [Journal] // Journal of the American Society for Information Science and Technology. - [s.l.] : Wiley Online Library, 2011. - 9 : Vol. 62. - pp. 1741-1749.

**Kayed Ahmad, El-Qawasmeh Eyas and Qawaqneh Zakariya** Ranking web sites using domain ontology concepts [Journal] // Information \& management. - [s.l.] : Elsevier, 2010. - 7 : Vol. 47. - pp. 350-355.

**Konchady M.** Text Mining Application Programming [Book]. - 2006.

**Lacy L.** Owl: Representing Information Using the Web Ontology Language [Book]. - 2005.

**Larocca J., Santos, A., Kaestner ,C., Neto ,A., Kaestner ,A., Freitas ,A.** Document Clustering and Text Summarization. - 2000.

**Melville P., Sindhwani V. and Lawrence R.** Social media analytics: Channeling the power of the blogosphere for marketing insight. [Conference] // Workshop onInformation in Networks. - 2009.

**Munibalaji T and Balamurugan C** Analysis of link algorithms for web mining [Journal] // World Wide Web. - 2012. - 2 : Vol. 1.

**Namburu S., Haiying, T., Jianhui L., Pattipati, K.** Experiments on Supervised Learning Algorithms for Text Categorization [Conference] // Aerospace Conference, 2005 IEEE. - 2005.

**Nigam K.** Using Unlabeled Data to Improve Text Classification [Report] : Doctoral Dissertation / School of Computer Science ; Carnegie Mellon University. - 2001.

**Nigam K., McCallum, A., Thrun, S., and Mitchell, T.** Learning to classify text from labeled and unlabeled documents [Conference] // Tenth Conference on Artificial intelligence . - Madison, Wisconsin, United States : [s.n.], 1998.

**Paltoglou Georgios and Thelwall Mike** Twitter, MySpace, Digg: Unsupervised sentiment analysis in social media [Journal] // ACM Transactions on Intelligent Systems and Technology (TIST). - [s.l.] : ACM, 2012. - 4 : Vol. 3. - p. 66.

**Pang B., Lee, L. and Vaithyanathan, S.** Thumbs up? Sentiment Classification using Machine Learning Techniques [Conference] // Conference on Empirical Methods in Natural Language Processing (EMNLP). - 2002. - pp. 79-86.

**Russell Matthew** Mining the social web [Book]. - [s.l.] : O'Reilly, 2013.

**Sahami M. , Dumais, S., Heckerman, D., Horvitz, E.** A Bayesian Approach to Filtering Junk E-Mail. - 1998.

**Volkova Svitlana, Wilson Theresa and Yarowsky David** Exploring sentiment in social media: Bootstrapping subjectivity clues from multilingual twitter streams [Conference] // Association for Computational Linguistics (ACL). - 2013.

**Wang TingZhong** The ontology recommendation system in E-commerce based on data mining and web mining technology [Book Section] // Advances in Electronic Commerce, Web Application and Communication. - [s.l.] : Springer, 2012.

**Weiss S., Indurkhya, N., Zhang, T., and Damerau, F.** Text Mining: Predictive Methods for Analyzing Unstructured Information [Book]. - [s.l.] : Springer, 2004.

**Yu J. [et al.]** Identifying Interesting Customers through Web Log Classification [Journal] // IEEE Intelligent Systems. - Piscataway, NJ, USA : [s.n.], 2005. - 3 : Vol. 20. - pp. 55-59.

**Yu J., Ou, Y. Zhang, C., Zhang, S.** Identifying Interesting Customers through Web Log Classification [Journal] // IEEE Intelligent Systems. - Piscataway, NJ, USA : [s.n.], 2005. - 3 : Vol. 20. - pp. 55-59.