Global Journal of Computer Science and Technology: H
Information & Technology
Volume 17 Issue 2 Version 1.0 Year 2017
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Probability of Semantic Similarity and *N-Grams* Pattern Learning for Data Classification

By V. Vineeth Kumar & Dr. N. Satyanarayana
Jawaharlal Nehru Technological University

Abstract- Semantic learning is an important mechanism for the document classification, but most classification approaches are only considered the content and words distribution. Traditional classification algorithms cannot accurately represent the meaning of a document because it does not take into account semantic relations between words. In this paper, we present an approach for classification of documents by incorporating two similarity computing score method. First, a semantic similarity method which computes the probable similarity based on the Bayes' method and second, n-grams pairs based on the frequent terms probability similarity score. Since, both semantic and N-grams pairs can play important roles in a separated views for the classification of the document, we design a semantic similarity learning (SSL) algorithm to improves the performance of document classification for a huge quantity of unclassified documents.

Keywords: semantic similarity, classification, naive bayes, n-grams pattern.

GJCST-H Classification: G.3 I.5, I.5.2

PROBABILITYOFSEMANTICSIMILARITYANDNGRAMSPATTERNLEARNINGFORDATACLASSIFICATION

Strictly as per the compliance and regulations of:

Probability of Semantic Similarity and *N-Grams* Pattern Learning for Data Classification

V. Vineeth Kumar [α] & Dr. N. Satyanarayana [σ]

Abstract- Semantic learning is an important mechanism for the document classification, but most classification approaches are only considered the content and words distribution. Traditional classification algorithms cannot accurately represent the meaning of a document because it does not take into account semantic relations between words. In this paper, we present an approach for classification of documents by incorporating two similarity computing score method. First, a semantic similarity method which computes the probable similarity based on the Bayes' method and second, *n*-grams pairs based on the frequent terms probability similarity score. Since, both semantic and *N*-grams pairs can play important roles in a separated views for the classification of the document, we design a semantic similarity learning (SSL) algorithm to improves the performance of document classification for a huge quantity of unclassified documents. The experiment evaluation shows an improvisation in accuracy and effectiveness of the proposal for the unclassified documents.

Keywords: semantic similarity, classification, naive bayes, n-grams pattern.

I. Introduction

Web mining is facing an important problem in measuring the semantic similarity among the words in the process of information retrieval and language processing. The most semantic based application requires the accurate measuring of semantic similarity among the document concepts and words. In information search, one of the most important problems is to semantically to get a number of documents correlated to a user's request. Semantic similarity between the words such as "word sense disambiguation" (WSD) can be an efficient assessment for the text entailment and automatic document classification, it is also important for the variety of natural language processing tasks. Automatic classification of documents is an important part of the research in the vision, and an enormous prospective for numerous applications around the text, such as search and analysis. Its purpose is to allocate a document given to the group of default to which it is in the right places. So far, applications have different types of algorithms based on the study or automatic calculation in this process and showed how much work [2], [3], [5]. However, mainly of the work functional to this task used aneffortless word-collection representation where each attribute communicates to a particular word. That is, assume that words are independent and utilize only the distribution of content words.

Over the past few years, we've seen the Web evolve into a semantic Web. The amount of information posted with linked data has consistently increased. With this increase, annotation and classification systems have created new opportunities to reuse this data as a semantic knowledge base and can be interconnected and structured to increase the accuracy and recovery of annotation and classification mechanisms. The Semantic web aims to explain the meaning of the information posted on the Web in order to make it possible to search by understanding the meaning of the information accurately. In this regard, document text learning and classification is most common, by assigning text to one or more existing class. This development determines the class membership of a text document that has a separate set of classes with profiles and different features. Criteria for deciding appropriate features for classification are important and are determined by the priority of the classifier. Semantic classification occurs when the target document element or term of the classification represents the meaning of the document.

Measuring the semantic similarity among texts is a basic task and can be capable of being utilized for a variety of applications, together with "text clustering" [1] and "text classification" [2]. The challenge in evaluation similarities among texts is infrequent, that is, there will be no coincidence of terms between the two texts. For example, two texts "Apple's New Product" and "iPhone-6" refer to related topics, even though they do not use similar terms.

To overcome scarcity, we need to use external data or knowledge to enrich the semantic representation of text. The semantically associated words of a particular word are listed in amanually created universal dictionary vocabulary ontology such as "Word Net". In this, a synset includes a set of synonyms for a specific word sense. However, semantic similarities among individual transform more than time and across domains. For example, apples are often associated with computers on the web. However, this apple sensation is not listed in most universal thesauri or dictionaries. Users searching for apples on the web may be concerned in the meaning of "apple" and "not apple" as a fruit. Innovative words are stably generated and new

Authors α: Research Scholor, JNTU Hyderabad.
e-mail: vineethvyas@gmail.com
Author σ: Principal, NGITS, Hyderabad.

Global Journal of Computer Science and Technology (H) Volume XVII Issue II Version I Year 2017 1

Global Journal of Computer Science and Technology (H) Volume XVII Issue II Version I Year 2017

senses are dispensed to existing words. Preserving ontology manually to confined these innovative words and senses is costly, if not impracticable.

In this paper, we contribute an automated semantic similarity learning (SSL) move towards to compute the probability of semantic similarity among terms or entities of documents with the class knowledge set entities. Here, we define two probabilistic scores, a semantic similarity (SS) score and *N-grams* pair similarity (GS) score enhancing Naive Bayes probabilistic method to aggregate the relation between document and class entities. Semantic Similarity method relates the trained class entities terms with the extracted document key terms to compute the document probable SS score against each class entities, and *N-grams* pair similarity method relate a document with each trained class entity with the constructed *N-grams* pairs, which is constructed using most frequent terms extracted from the document and the probable GS score is the summation of all individual *N-grams* pairs, i.e., $sum(GS_1, GS_2, .. , GS_n)$.We perform an experiment evaluation on Reuters-21578 Datasets to demonstrate the effectiveness of the proposal.

This papers organized in 6 sections. Section-1 above describes the introduction, section-2 discuss the background works, section-3 presents the proposed works outline, probabilistic semantic and N-gram pairs pattern learning, section-4 discuss the semantic similarity classification approach, section-5 present experiment methodology and results and finally section-6 presents the conclusion of the work.

II. Background Study

Semantic similarity plays an significant responsibility in "natural language processing", "information retrieval", "text summarization", "text classification", and "text clustering". Particularly, "Explicit Semantic Analysis" (ESA) [6] is extensively utilized because of its accessibility and diversity. ESA was build up to calculate word relationship as well as text comparison in natural language. ESA creates a "weighted index" that maps each phrase to the listing of articles that appears and calculates the similarity among the two words or a vector of text.

Naive Bayes [1] classification performance using semantic similarity has made various efforts. An approach that is often used to mitigate naive independent assumptions is to express attribute addiction in a graph-based model called a "Bayesian network", where nodes correspond to attributes. Oriented arch is weighted by the circumstances probability for each node specified a close relation. Because "Bayesian network learning" is NP-hard [6], numerous approaches recommend imposing model constraints to formulate it easier to deal with learning problems.

Subsequent approaches in [8], [9], [17], [18] have brought considerable improvements. For example, in [21], an ensemble of Tree Augmented Naive-Bayes (TANs) was be trained, each rooted in a dissimilar attribute. It then compiles the classifications of all eligible TANs to predict class labels. In [8], we assume that the entire Bayesian network structure is learned first and all attributes are dependent. Unlike [18], the "Markov network model" is utilized to express characteristic dependencies that are estimated similar to [39] by taking advantage of the conditional log probability intention purpose. However, performing andtake advantage operation can be computationally demanding. Many methods are used to inherit the structural simplicity of Naive Bayes classifiers to keep away from the complication of the construction learning process [9],[10],[12],[13]. While the "Naive Bayes classification" is functional at the "decision tree leaves level" and is act upon on a subset of the training data, the data set properties are divided into two collections as in [11],where one group is assigned a class probability based on "Naive Bayes", and the other is supported on a "decision table".

Despite its effortlessness, the previously point out the classifier still shows a few constraints in handling very much related data. In [12], [13], the features are weighted dissimilarly depending on the involvement to the classification. A comparable approach was applied to the most effective "Bayesian Network classifier" and " Hidden Naive Bayes" [9]. In [9], the authors recommended generating a hidden close qualified that correspond to the effect of everything else on each property. The effect is computed as a linear arrangement of circumstance common information among attribute pairs, similar to [8]. Therefore, the parent correlation is ignored.

Dissimilarity like [10], [11], [19], "En Bay" [2] implements a new, uncomplicated, and useful approach that unites the generation of conditionally independent decision models and the reliable probability approximation by class. En Bay is a pattern-based Bayesian classifier that frequently uses a set of items frequently to estimate Bayesian probabilities. En Bay uses new and effective probabilistic approximation estimates that adhere to the conditional independence model. The set of extended, normal and separate items to be comprised in a class-based approximation is chosen by entropy-based heuristics, and the set of properties is conditionally mutually independent, depending on the class being evaluated. We extend En Bay probability computation methods to computes the semantic similarity probability score based on the terms dependency over the trained class terms entities, as discussed in section 3.2 below.

The "Large Bayes classifier" [11] performed the primary challenge to mitigate well-built independent assumptions using a lengthy and frequent set of items to estimate the probability through product form approximation [12]. However, all preceding pattern-based Bayesian advance create inimitable product approximations for all test cases. Thus, estimations are only tied to the considered grade. Moreover, since it is necessary to extract animmense number of long and redundant repeated item sets, the superiority of the approximation is sensitive to changes in the "support threshold", and the classification algorithm cannot cope with a large data set. We extend this constructing *N*-grams pairs using frequent items to estimates the *N*-grams similarity probability score, as discussed in section 3.3 below.

III. Proposed Approach

a) Outline

The Semantic Similarity Learning (SSL) method, which uses probabilistic performances to describe probabilistic scores and put together scores supported on Bayes' method for accurate document classification to measure the robust discovery and semantic similarity of related entities to document.

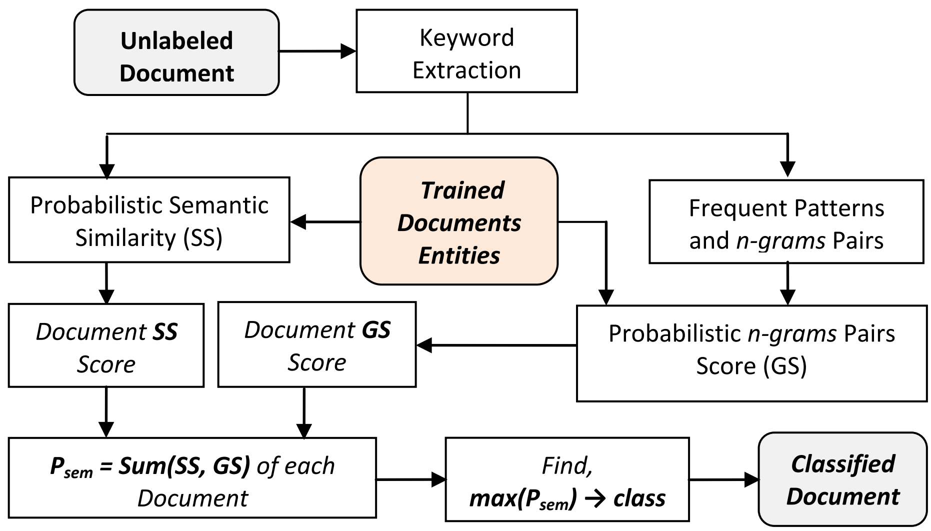

Figure 1: Outline of the proposed Approach

Fig.1. outlines our the proposed approach method. Our method obtains the main points of the probabilistic analysis of associations and related documents on the basis of trained document entities. The approach performs two probabilistic score computation method. First, Semantic Similarity Method which measures the similarity of their associated entity of a document with the list of trained class entity terms by $SS_k = P(d_k(t) \mid C_m)$, *i.e.,* probability of a document $d_k(t)$terms associated with a set of class C_m terms by means of cosine similarity.

The second method extracts the most frequent terms *F* from the extracted document terms using term frequency *(tf)* and using *F* we construct *N-grams* pairs. In general, an *N-gram* method slice a longer text into *n-characters*, but we customized this to slice a pattern into number words pairs (*V-Pair*) based on *n* which we term as *N-gram* pattern, an illustration is shown in Fig. 2.Using the constructed pairs we compute, $GS_k = \sum_{i=1}^{n} W_i$, where *n* is the number of pairs and $W_i = P(\ V\text{-}Pair_n \mid C_m\ )$ *i.e.,* probability of N-gram pair terms related to the set of class C_m terms using cosine similarity.

Now, we compute the final probability of semantic similarity $P_{sem} = sum\ (SS_k, GS_k\ )$ for each document against each trained class. To classify the document we find the max P_{sem} among the computed probability of semantic similarity of each class. The class which has the max P_{sem} will be considered as the document class. We describe each method mechanism in the aspect in the following sections.

Global Journal of Computer Science and Technology (H) Volume XVII Issue II Version I Year 2017

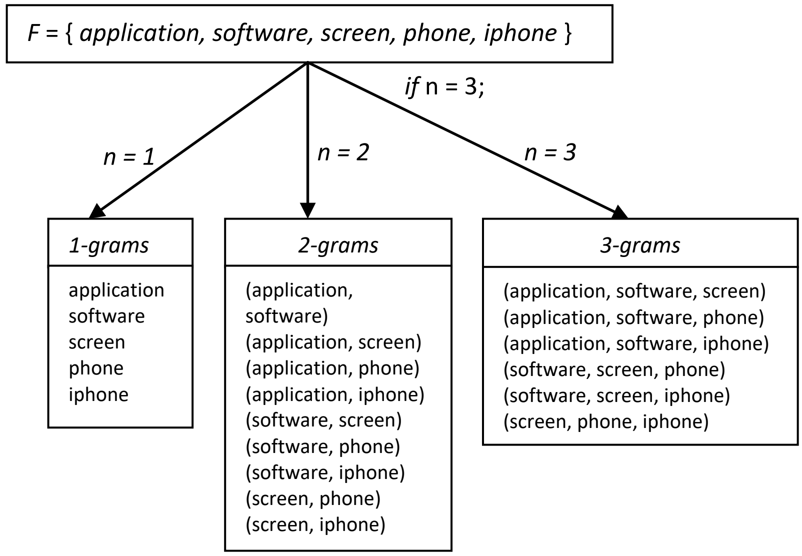

Figure 2: An illustration of *N-gram* pairs

b) Probabilistic Semantic Similarity

Classification of classifiers named a set of classes that train classifiers built from sets of the abstract model defined aims. Then use the categorizer to properly classify new data whose class labels are unknown. Various approaches have been proposed to make accurate classifiers such as, "Bayesian classifiers" [1], "Decision Trees" [2], "SVMs"[3], "Rule-based" [4], and "Associative classifiers" [5].

Bayesian classification methods recognized a classification supported on the of "Bayes theorem" [1].It predicted that a class based on test documents previously un seen $T = \{ a_1, a_2, . . . , a_n \}$by opting the class c_i that make the most of the subsequent formula:

$$P(c_i \mid T) = \frac{P(T, c_i)}{P(T)} = \frac{P(c_i) \cdot P(T \mid c_i)}{P(T)}, \quad (1)$$

Where $P\ (T|\ c_i)$indicates the provisional possibility of the test document T of a given class c_i. Probability is approximate commencing from the training set. Since classification focuses on choosing the class that takes advantage based on the equation (1), relatively than assigning an unambiguous probability to each class, denominator $P(T)$ in (1) can be misplaced because it does not influence the comparative class instruct.

Despite the simplicity, the Bayesian approach is calculation intractable without compelling a powerful model simplification [1], [6], [7]. The most important instance of simplification is the "Naive Bayes classifier" [1], which solves the problem by assuming that all attributes are conditionally self-determined and given as the class c_i. Therefore, the join probability of (1), is based on the generated Naive Bayes model, which can be approximated as,

$$\begin{aligned} P(T, c_i) &= P(a_1, a_2, \ldots, a_n, c_i) \\ &\simeq P(c_i)P(a_1 \mid c_i)P(a_2 \mid c_i) \cdots P(a_n \mid c_i) \\ &= P(c_i) \prod_{j=1}^{n} P(a_j \mid c_i). \end{aligned} \quad (2)$$

Based on the approximation we combine the probabilistic semantic similarity (SS) scores extracted from the training data to find the appropriate entities for the document. Let's assume that multiple key terms are entered as input. That is, we compute $P\ (c\,|\,T\,')$for the set of core key terms $T' = \{t_1, t_2, \ldots, t_k\}$, where T' is a key term, which are derived using traditional Naive Bayes for any related class c_i.

One possible approach to this task is a two-step method of determining the key terms first and then applying the existing Naive Bayes. However, this approach raises the question of how key terms are established. We have developed a probabilistic similarity method for finding related entities. It can be functional to a set with probability determined members.

For a particular, a set of key terms *T*, *P (c|T′)* is calculated for all probable states *T′*.Fig.3,summarizes an illustration of the probabilistic semantic similarity method for a set of key terms of a document $d_k$as $t_1, . . . , t_k$. SS method is utilized to calculate $P(t_K|C_m)$, which is the probability score SS_k of the set of key terms, *T* for the class C_m.

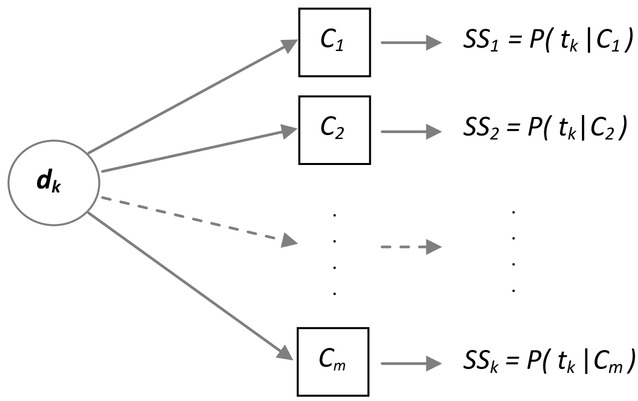

Figure 3: Probabilistic Semantic Similarity Method

Unfortunately, the circumstances of self-determination supposition prepared by Naive Bayes may not true always, to obelieves high-order associations for the period of the probability estimate, a parallel proposal is made based frequent pattern learning in *T,* and constructing a *N-grams* pairs to support accurate classification.

c) N-grams Pattern Learning

The *N*-gram is defined as a sequence of terms, the length is *n*, and the words taken are called terms. In the literature, we can see the definition of an *N*-gram as a concurrent set of terms, but only consecutive term sequences were used in this study. One word in the document is represented by a set of overlapping *N*-grams as shown in Fig. 2. The *N*-gram model can be fictional by introduction a small window over a sentence or text, where only *n* words can be seen at the same time. So the effort less *N*-gram model is the so-called "unigram model". This is a one-word model at a time. For example, the "Latest application and iPhone released." sentence contains five unigrams as, "Latest", "application", "and", "iPhone" and "released" Of course, this is not very beneficial information. It is just a word that makes up the sentence. In fact, *N*-grams are interesting when *n* is greater than 2 (bigram) or more.

Each word happens in a document with a dissimilar frequency. The main thought of categorization utilized by Trenkle and Cavnar [5] is that they should have similar *N*-gram frequency distributions when comparing documents of the same category. We performanN-gram pattern learning through creating *n* pairs using frequent document terms.

For a given document *d* having a *T* terms. Let's assume the frequent terms represent as *F*. Using the *F* terms we construct *N-grams* pairs as *V-Pair.* To learn the probability of *V-pair* pattern associationW_n of a document with a class c_m we calculate $P(V\text{-}Pair_n|c_m)$as shown in Fig .3. Here, the class must contain all the pair terms to match the association. To compute the N-grams probable similarity *GS,* we done the summation of all W_n as,

$$GS_1 = \sum_{i=1}^{n} W_i$$

Global Journal of Computer Science and Technology (H) Volume XVII Issue II Version I 5 Year 2017

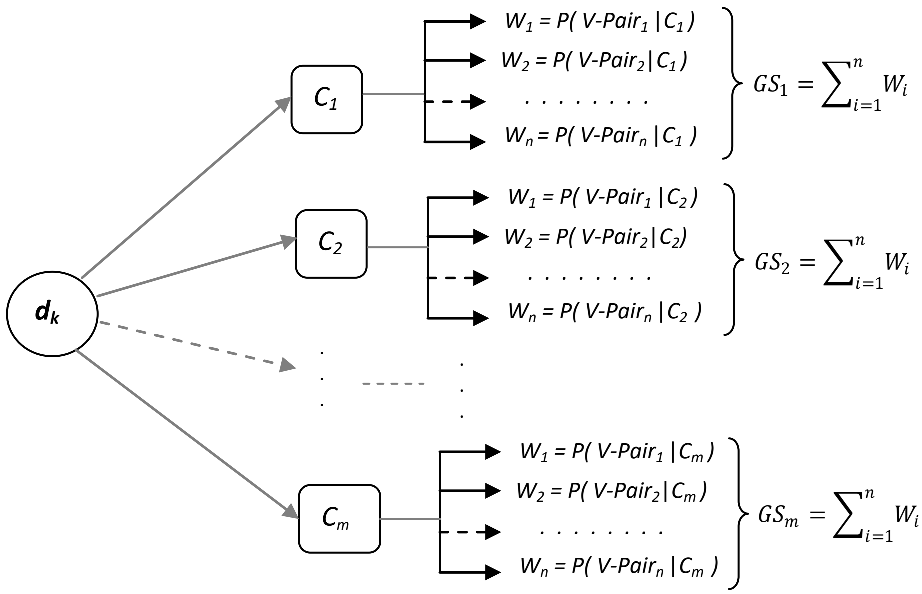

Figure 4: Probabilistic *N-grams* Similarity Method

IV. SSL BASED CLASSIFICATION

a) *Training*

In order to efficiently search for class-specific patterns, in the SSL training phase, an FP-growth data pattern representation [18] is separately created to store training data belonging to each class in a compressed form. The FP-growth pattern is a single-tree data structure for the minimum support (*min_sup*) frequent item set used in the class pattern learning context. Algorithm-1 shows the pseudo-code in the SSL training phase. Minimum supported thresholds were applied to remove infrequently used items. In this case, items that do not meet the necessitated minimum support threshold are not consisted of in the FP-growth pattern.

Algorithm 1. SSL Training Phase (*D, min_sup*)

Input: The training set *D* and the minimum support threshold *min_sup*
Output: FP-G = $\{ T_i \}$ $\forall c_i \in C$, a FP-Tree for each class belonging to the training class set *C*

for all c_i in *C* **do**
    ac_i = set of all items belonging to class c_i
    $FT_i = ExtractPattern(ac_i, min\_sup)$
    $FP\text{-}G = FP\text{-}T \cup \{ FT_i \}$
end for
return *FP-G.*

The obtained *FP-G* of each individual class c_i , will be used as a trained knowledge for the SSL classifier.

b) *SSL Classification*

The SSL classification approach is one of the accomplished algorithms managing unlabeled documents. It applies two probability computation as semantic similarity SS and N-gram Similarity G Son the dataset to perform the classification using the trained FP-growth pattern knowledge as shown in Fig.1. The SSL classifier initialized with anonly some trained class item sets. At each iteration, it chose an unlabeled document and perform the computation to compute the SS and GS score. It learns separate similarity score over each class pattern learning, and support a set of class labels for the unlabeled documents. For each class c_i belonging to the training data set is corresponding to FP-growth is visited to construct the class-centric product estimation and calculated the probability $P(T,c_i)$.

Its ultimate prediction is through by coming together both SS and GS score, $P_{sem} = sum(SS_k, GS_k)$ which decline classification error spredictions. The massive is the P_{sem} of the class will be predicated as the closer association. This predictionim provises the performance of the algorithm classification accuracy. Since SSL classification uses two probability computation score with the FP-growth pattern, its presentation is better than any particular classifier.Algorithm-2briefly summarizes the SSL classification algorithm.

Algorithm 2.SSL Classification

Input: Document Terms *D* and the class set *FP-G* = { FT_i}

*Output:*Classified class of *D.*

for all*d*in *D*do
//-- For all document in D --
// -- Probability of Semantic Similarity(SS) Score
$d_k = \{T_k\} \ \forall D_k$;
//-- For all class in FP-G vector --
for all c_i *in FP-G*do
$c_i = \{FT_i\} \ \forall FP\text{-}G_i$;
$SS_i = P\ (d_k \epsilon\ c_i)$;
$V\_SS[i] = SS_i$;
end for
$VD_s[k] = V\_SS$;
// -- Probability of N-grams Similarity(GS) Score
//-- Most frequent terms--
F = frequent_Terms (d_k, *min_sup);*
//-- Builds N-grams Patterns --
NP = BuildPattern(F, n);
//-- For all n-grams patterns --
for all ng_pin*NP* do
ng_terms = ng_p;
//-- For all class in FP-G vector --
for all c_i *in FP-G*do
$c_i = \{FT_i\} \ \forall FP\text{-}G_i$;
$GS_i = P\ (ng\_terms\ \epsilon\ c_i)$;
$V\_GS[i] = GS_i$;
end for
end for
$VD_g[k] = V\_GS$;
// -- Summation Probability --
for all c_i *in FP-G*do
$SS\_P_{sem} = VD_s[i]$;
$GS\_P_{sem} = VD_g[i]$;
$P_{sem} = sum(SS\_P_{sem}, GS\_P_{sem})$;
$VP_{sem}[i] = P_{sem}$;
end for*pmax = findMax* (VP_{sem});
d_k_class = *getClass (pmax, C);*
end for

The obtained d_k_class class from *C*is determined by the summation of two probabilities scores. It classifies the most excellent class *c*of document *d* is set to the individual with the maximum probability:

V. Experiment

a) Datasets

The "Reuters-21578 corpus" is the mainly common utilized benchmark corpus in text classification. It consists of over 20,000 Reuters news stories from 1987 to 1991, and 135 subject classes are used in the experiment. This version contains "9603 training documents", "3299 test documents", and "27,863 inimitable words" after stopping stemming and word removal. We consider only 10 topics as classes of Reuters-21578 data for experimental evaluation measurements.

b) Performance Measure

To estimate the classification performance of the proposed method, we utilize the precision, recall, and accuracy. Let considered *P* is all relevant documents and *N* is all negative document. PC_+ as a positively classified, NC_+ as negatively classified documents. PC_- as a positively classified for an incorrect document, NC_- as negatively classified for correct documents. By constructing a confusion matrix for the above evaluation measure we compute the classifier performance.

To measure the classifier precision rate *CP*, the classifier recall rate *CR* and the classifier accuracy rate *CA* the following equation are used.

$$CP = \frac{PC_+}{PC_+ + NC_+} \quad (4)$$

$$CR = \frac{PC_+}{PC_+ + NC_-} \quad (5)$$

$$CA = \frac{PC_+ + NC_+}{P + N} \quad (6)$$

c) Evaluation Results

In the Reuters-21578 datasets we do consider both labeled and unlabeled documents, the effect of using two probabilistic semantic similarity learning is given in Table 1. We initially evaluate with Semantic Similarity Score (SS), then with N-grams patterns pairs Score (GS) and finally with both. The classification performance using both the Semantic Similarity and the N-gram pattern pairs learning outperforms over the one using any single learning for most classes.

Global Journal of Computer Science and Technology (H) Volume XVII Issue II Version I Year 2017

Table 1: The accuracy enhancement by using semantic similarity learning on "Reuters-21578 corpus".

Class	Relevant documents	Semantic Similarity	N-gram pattern pairs	Both
Acq	1650	1591	861	1629
Corn	181	93	92	168
Crude	389	328	146	354
Earn	2895	2765	1621	2825
Grain	433	396	208	426
Interest	347	284	159	341
Money-fx	538	327	179	493
Ship	197	106	68	188
Trade	369	235	97	355
Wheat	212	184	102	206

Global Journal of Computer Science and Technology (H) Volume XVII Issue II Version I Year 2017

We found that the greater the number of related documents in the training set, the higher the accuracy of using *N*-gram pattern pair learning. This is because Naive Bayes has a low error rate and high accuracy when there are many documents in the class. The classification comparison result is shown in Fig.4.

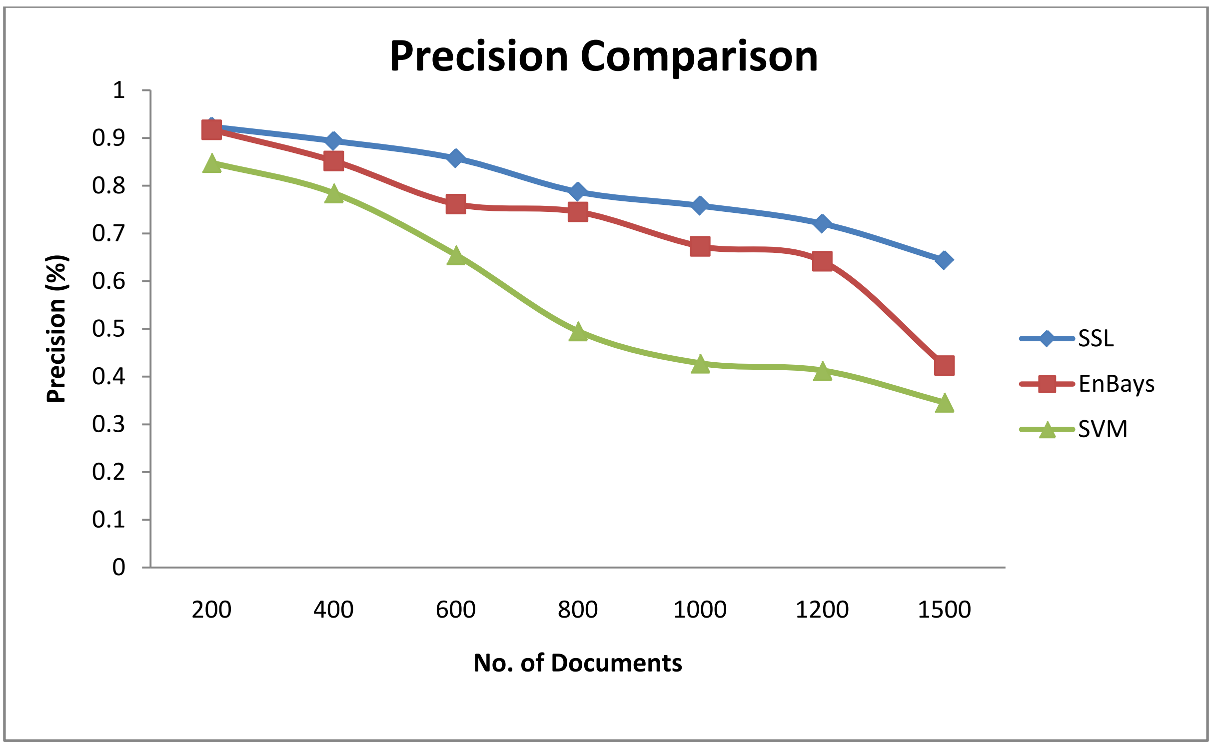

Figure 4: Classification Comparison

Accuracy measures the capability of a taxonomy to correctly classify unlabeled data. The ratio of the number of correctly categorized data to the number of given data, including accurate and incorrect classification. Experimental results show that average SSL outperforms other classified segments. The statistical significance of improving SSL accuracy is discussed below.

At first, we performed comparisons with state-of-the-art Bayesian classifiers. And because our approach is pattern-based, we compare it with the well-known associative classifiers SVM and the new improved Bayesian approach known as En Bays [2]. Finally, we performed a comparative assessment of precision, recall, and accuracy rates as a classifier for classifiers.

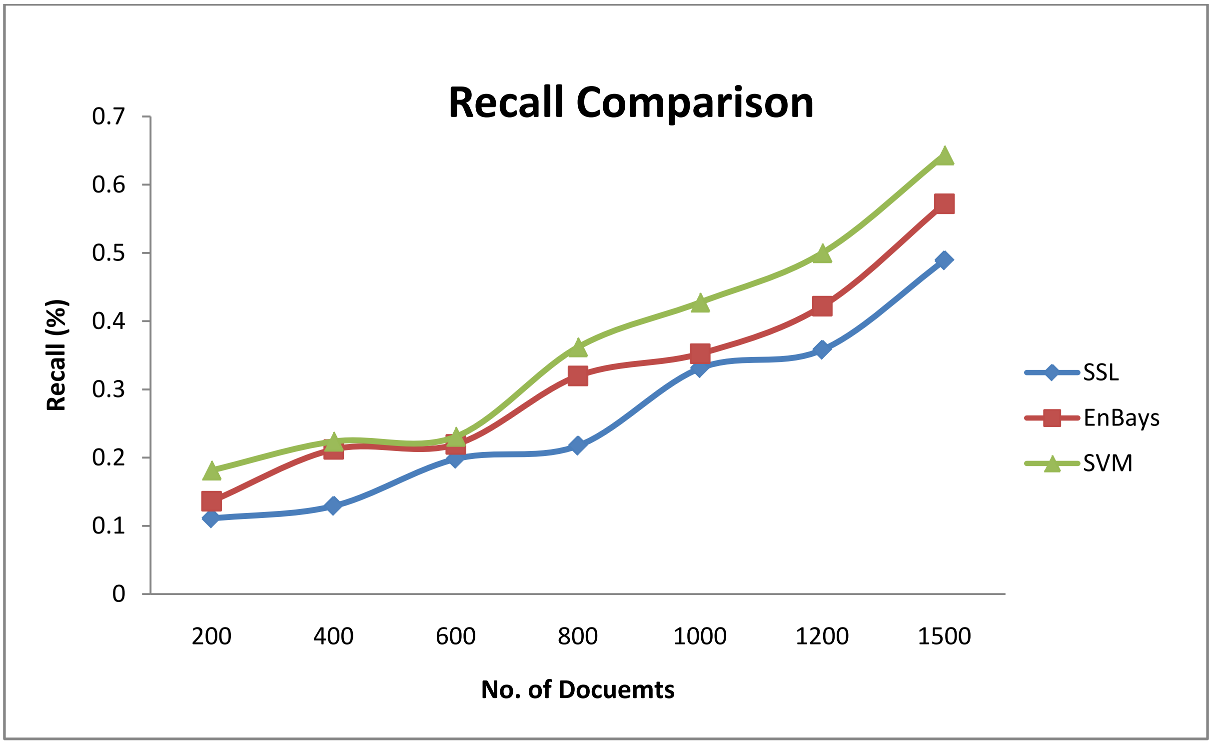

Figure 5: Precision Comparison

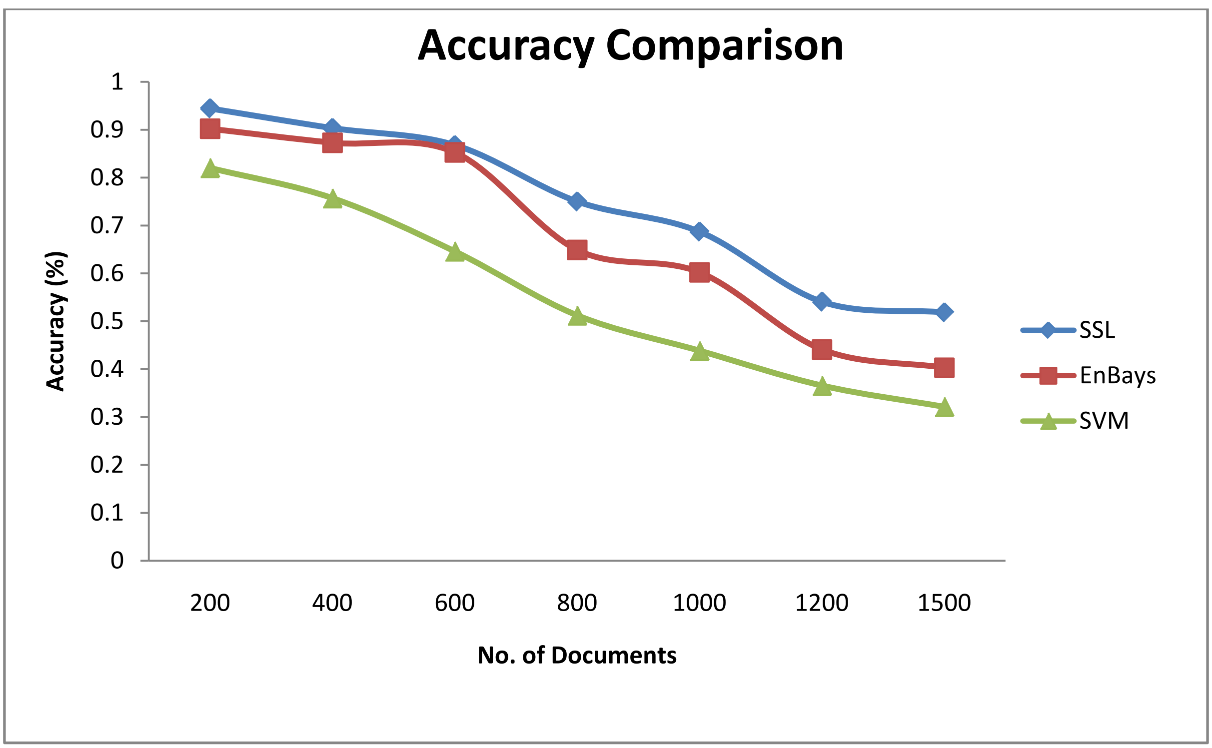

Figure 6: Recall Comparison

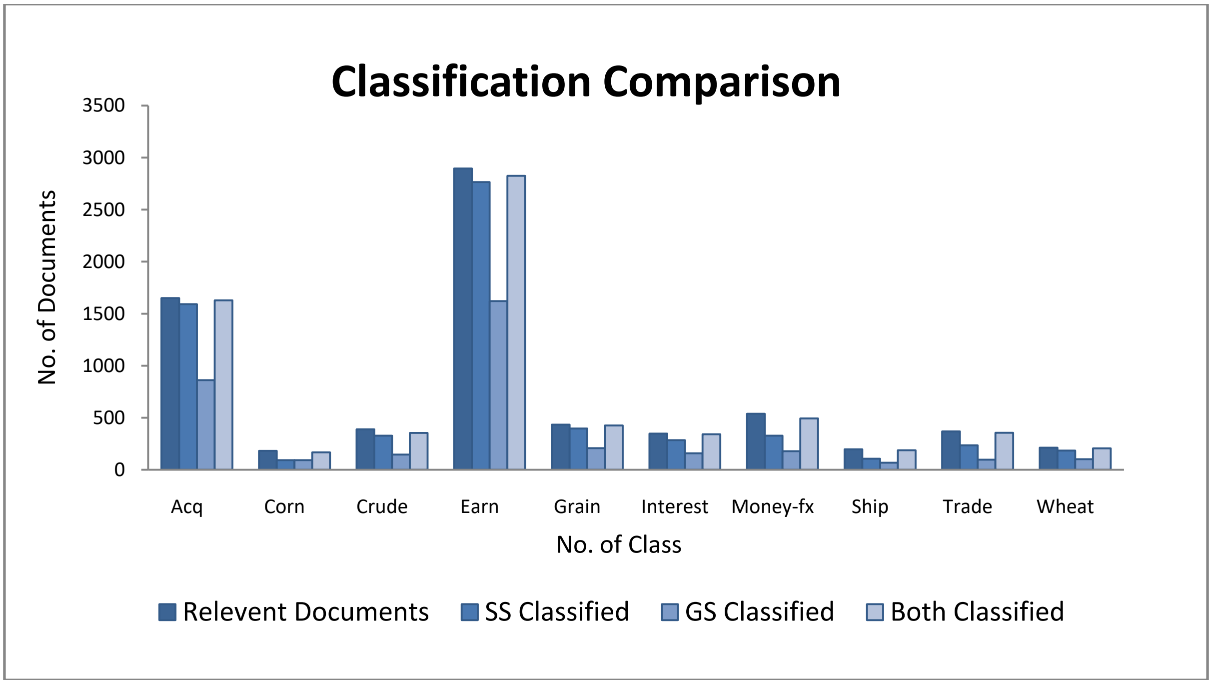

Figure 7: Accuracy Comparison

The rate of precision and recall in Fig. 5 and 6 shows an improvisation in compared to SVM and En Bays method. The effects of both SS and GS score in probability similarity measure shows SSL precision improvisation. Fig. 7 shows the classifier accuracy measures comparison. It also shows an improvisation of SSL approach in compare to others. The falling of accuracy with increasing of the document due to the limitation of trained class knowledge. As both the method has a dependency on the trained data knowledge for performing probability similarity computation cause the falling of the rate.

VI. Conclusion

In this paper, we propose a semantic similarity and *N*-gram pattern learning method based on the Bayesian classifier, which approximates Bayesian probability using frequent itemsets. It utilized new and more efficient probability approximations that adhere to the conditional independence model. A long, frequent, and separate set of items to be included in a class-based approximation is selected. It is based on the Baye's theorem and semantic similarity computation approach. Our method is a sort of probabilistic semantic similarity learning (SSL) that uses vectors to generate vectors of related entities as semantic representations of specific text and to measure semantic similarities. SSL combines vectors using expanded Naive Bayes, while SSL simply adds up the vectors for each term occurring in the text based on the majority of rules. This method uses both Semantic Similarity Learning for SSL algorithms and *N*-gram pattern learning and applies algorithms to unstructured document classification. Experiments on the Reuters-21578 document show that the SSL approach improves classification performance, and unlabeled documents are a good resource to overcome documents with a limited number of labels.

Future developments in this work will address the integration of generalized item aggregation mining algorithms to further improve classification and accuracy in noise-prone areas of data where there liability of probability estimation is particularly important.

References Références Referencias

1. Masumi Shirakawa, Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio, "Wikipedia-Based Semantic Similarity Measurements for Noisy Short Texts Using Extended Naive Bayes", IEEE Transactions On Emerging Topics In Computing, Volume 3, No. 2, June 2015.
2. Elena Baralis, Luca Cagliero, and Paolo Garza, "En Bay: A Novel Pattern-Based Bayesian Classifier", IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 12, December 2013.
3. D. Grossman and P. Domingos, "Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood," Proc. 21st Int'l Conf. Machine Learning (ICML '04), http://doi.acm.org/10.1145/1015330.1015339, p. 46, 2004.
4. R. Kohavi, "Scaling up the Accuracy of Naive Bayes Classifiers: A Decision-Tree Hybrid," Proc. Second Int'l Conf. Knowledge Discovery Data Mining (KDD '96), pp. 740-743, 1996.
5. M. Hall and E. Frank, "Combining Naive Bayes and Decision Tables," Proc. 21st Int'l Florida Artificial

Intelligence Research Soc. Conf., pp. 318-319, 2008.

6. J.T.A.S. Ferreira, D.G.T. Denison, and D.J. Hand, “Weighted Naive Bayes Modelling for Data Mining,” 2001.
7. M. Hall, “A Decision Tree-Based Attribute Weighting Filter for Naive Bayes,” Research and Development in Intelligent Systems XXIII, M. Bramer, F. Coenen, and A. Tuson, eds., pp. 59-70, Springer, 2007.
8. R. Agrawal, T. Imielinski, and A. Swami, “Mining Association Rules between Sets of Items in Large Databases,” ACM SIGMOD Record, vol. 22, pp. 207-216, 1993.
9. R. Johnson and J. Shore, “Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross Entropy,” IEEE Trans. Information Theory, vol. IT-26, no. 1, pp. 26-37, Jan. 1980.
10. E. Baralis, L. Cagliero, T. Cerquitelli, V. D’ Elia, and P. Garza, “Support Driven Opportunistic Aggregation for Generalized Itemset Extraction,” Proc. Fifth Int’l Conf. Intelligent Systems, 2010.
11. [1]S. Banerjee, K. Ramanathan, and A. Gupta, "Clustering short texts using Wikipedia", in Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), Jul. 2007, pp. 787-788.
12. X. Sun, H. Wang, and Y. Yu, "Towards effective short text deep classification", in Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), Jul. 2011, pp. 1143 - 1144.
13. Preethi, Ms. N., and Devi, Dr. T., Case and Relation (CARE) based Page Rank Algorithm for Semantic Web Search Engines. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1, May 2012.
14. Lee, T. B., Hendler, J., and Lassila, O., "The semantic web". Scientific American, vol. 284(5), May 2001.
15. Ch.-Qin Huan, Ru-Lin Duan, Y. Tang, Zhi-Ting Zhu, Y.-Jian Yan, and Yu-Qing Guo, ."EIIS: an educational information intelligent search engine supported by semantic services". international Journal of Distance Education Technologies, January 1, 2011.
16. Robin Sharma, Ankita Kandpa, and Priyanka Bhakuni, Rashmi Chauhan, R.H. Goudar and Asit Tyagi." Web Page Indexing through Page Ranking for Effective Semantic Search". Proceedings of7'h International Conference on Intelligent Systems and Control (ISCO 2013).
17. Yuan LIN, Hongfei LIN, and Li HE." A Cluster-based Resource Correlative Query Expansion in Distributed Information Retrieval ".Journal of Computational Information Systems 8: 1, 2012, 31–38.
18. W. W. Chu, Z. Liu and W. Mao."Textual document indexing and retrieval via knowledge sources and data mining". Commun. Inst. Inf. Comput. Mach. (CIICM), Taiwan, 2002, 5, (2), pp. 135–160
19. A. Vizcaíno, F. García, I. Caballero, J.C. Villar, M. Piattini."Towards an on tology for global software development". IET Softw., 2012, 6, (3), pp. 214–225
20. N. Tyagi and S. Sharma."Weighted Page Rank Algorithm Based on Number of Visits of Links of Web Page". In International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-3, July 2012.
21. N. Duhan, A. K. Sharma and K. K. Bhatia."Page Ranking Algorithms: A Survey". In proceedings of the IEEE International Advanced Computing Conference (IACC), 2009.
22. Vishal Jain, Dr. Mayank Singh."Ontology Based Information Retrieval in Semantic Web: A Survey ", I.J. Information Technology and Computer Science, 2013, 10, 62-69.

This page is intentionally left blank