A High Performance Semi-Supervised Learning technique for Non-Standard Word Classification in Bengali News Corpus

Chandan Kundu[1], Rajib Kumar Das[2], Kalyan Sengupta[3]
[1] Research Scholar
West Bengal State University
Barasat, Kolkata, India
[2] Dept. of Statistics, EIILM, Kolkata, India
[3] Dept. of System, IISWBM, Kolkata, India
chandankundu2008@gmail.com, rkdas70@gmail.com, kalyansen@iiswbm.edu

ABSTRACT: *The key disadvantage of the supervised learning technique is that it requires many hand-labeled test data to learn the classifier accurately. However, in this dynamic world, neither it is possible always to create database of labeled data, nor it is readily available in hand. Therefore, most of the users of a practical system would prefer algorithms that take few numbers of labeled data. This research paper demonstrates that semi-supervised naïve Bayes classifier using Expectation Maximization algorithm with few labeled data and huge number of inexpensive unlabeled data can create a high-accuracy non-standard word (NSW) classifier. It has been found that low information features contribute little to the accuracy of the naïve Bayes classifier. Therefore, we have eliminated these low information features during the estimation process and applied in the semi-supervised technique, thus provides a high performance model. The performance of the naïve Bayes classifier is good enough when there is huge number of labeled data. However, the EM method dramatically improves the accuracy of a NSW classifier, especially when there are only a few labeled data. We have carried out experiment on Bengali and English news corpus, but this is a general approach that can be applied to any language.*

Keywords: Naïve Bayes Theorem, Expectation Maximization Algorithm, Labeled Data, Unlabeled Data, Non-Standard Word

Received: 10 May 2015, Revised 14 June 2015, Accepted 23 June 2015

1. Introduction

Information retrieval is the process of extraction of information from structured and unstructured data. In this process, the information that is needed to be extracted is presented in the form of a query and this query tries to match against the information contained in the database. Much of the research and development in information retrieval is aimed at improving retrieval efficiency [Göker and Davies 2009, Feldman and Sanger 2007].

Non-standard word identification and interpretation [Cavnar and Trenkle 1994, Yarowsky 1996] is a type of information retrieval technique and plays important roles while we are conducting research works mainly on natural language processing (NLP). Text normalization could be considered as a prerequisite for different speech and language processing tasks. The processing of text is required in language and speech technology applications such as text-to-speech (TTS) and automatic speech recognition (ASR)

systems. Non-standard representations in the text must typically be normalized to be processed in those applications. Text normalization is a process by which text is transformed in some ways to make it consistent in a way which it might not have been before. Text normalization takes account of classification of non-standard forms (dates, URL, numbers, currency, time, amounts, etc.), as well as converting theses forms into its corresponding word formats (e.g. 2.30 a.m. —> two thirty a.m.). Normalization process can be applied in different ways and in different intense to the various speech and language processing tasks. The approaches that have been followed in different normalization techniques may work well on one textual domain but may not work on another. This research paper addresses the design and implementation of supervised and semi-supervised techniques that are general in nature for the identification and interpretation of NSWs, formally known as text normalization system [Sproat et al. 2001]. The experiments have been carried on a set of Bengali (Bangla) and English news corpus.

A naïve Bayes classifier utilizes Bayes theorem and it believes in "independent feature model". Naïve means 'independence'. A naïve Bayes classifier assumes that presence or absence of one feature in a class is independent to the presence or absence of other features in that class [Carlin and Louis 1996]. For example, an animal may be considered as a dog if it is hairy, four-footed, about 2ft in length and white. Even though these features depend on each other or they are related, a naïve Bayes assumes all these features independently contribute to the probability that it is a dog.

A classifier learns itself either from knowledge base or from training data. Training data is of two types; labeled data have been designated with specific class labels and unlabeled data have no class labels. In real life applications, it is very difficult to get appropriate labeled data at right time. Preparation of labeled data is time consuming, expensive, error prone and tedious. However, collection of unlabeled data is relatively easier since they are not required to be labeled with appropriate class labels, thus resulting in savings in both the time and cost required for training classifier. Learning the classifier with labeled and unlabeled data is known as semi-supervised learning. [Merz et al. 1992] first coined the term "semi-supervised" for classification with both labeled and unlabeled data.

[Blum and Mitchell 1998] proposed the co-training technique where the training set is split into two individual training sets. During the training process, each classifier uses the labeled training data to assign class labels to the unlabeled data. [Ghani 2001] extended the capability of co-training algorithm by incorporating error correcting output codes. It has been shown that this technique provides better results in semi-supervised domain. [Szummer 2001] developed an apparently new technology in semi-supervised learning technique where kernel structure was used on labeled and unlabeled data. [Celeux and Govaert 1992] initiated the Classification Maximum Likelihood (CML) and Classification EM (CEM) approaches that are applicable both for discriminative and generative models. Later, CML approach had been introduced in generative modeling by [McLachlan and Krishnan 1997]. Researchers now understand the importance of semi-supervised techniques and are trying to explore the power of it in different fields of NLP.

In this paper, we explore the use of unlabeled data to train an NSW classifier in a semi-supervised manner that are based on the Expectation-Maximization (EM) algorithm [Dempster et al. 1977] employed in NSW classification. [Miller and Uyar 1997], for the first time, proposed the EM algorithm for semi-supervised learning. [Nigam et al. 2000] have used the semi-supervised EM approach to text classification problems. Another approach described by [*Pakhomov et al. 2002*] utilized the concept of Maximum Entropy (ME) technique in semi-supervised modeling. They proposed a method of automatically generating training data for Maximum Entropy technique of abbreviations and acronyms and shown that ME is a powerful technique for abbreviation and acronym normalization.

In our work, each NSW class is modeled with a Gaussian mixture model (GMM). The parameters are estimated by iterative EM algorithm. The algorithm is divided into two stages. In the first step i.e. Expectation step (commonly known as E-step), we estimate the posterior distribution with the available information and in the Maximization step (commonly known as M-step), we re-estimate values of different parameters that have been used in E-step of the next iteration. Experiments reveal that unlabeled data along with a few numbers of labeled data can reduce the error rate to a greater extent.

Semi-supervised NSW classifier is constructed considering naïve Bayes theorem. Usually the classification model has hundreds or thousands of features, as in the case of NSW categorization. It has been found that some features are common across all classes and therefore contribute little information to the classification process. These features are commonly known as low information features. Individually they are harmless, but in aggregate**, low information features can decrease performance.** In this paper we have avoided the effect of low information features considering (i) context window (ii) elimination of low information

features.

Elimination of low information features gives our model simplicity by removing noisy data. It helps to avoid over-fitting problem and the curse of dimensionality. Use of only higher information features, increases the performance of the model and at the same time also decreasing the size of the model that provides less memory usage along with faster training and classification.

2. Naïve Bayes NSW Classification

This section describes the basic naïve Bayes NSW classifier. The naïve Bayes classifier is trained with number of labeled training data. It estimates the probable parameters that best fit to the proposed model given the observed labeled data.

2.1. Training a Naïve Bayes Classifier with Labeled Data

We can calculate the likelihood of a segment (containing NSW) [see section 2.4] *di* ($D = \{d_1,\ldots, d_{|D|}\}$) with a sum of total probability over all components

$$P(d_i|\theta) = \sum_{j=1}^{|C|} P(c_j|\theta)\, P(d_i|c_j;\theta) \tag{2.1}$$

where a segment, d_i, is generated according to the mixture weights (or class probabilities), $P(c_j|\theta)$, with distribution $P(d_i|c_j;\theta)$. Now naïve Bayes expression for the probability of a segment (containing NSW) given its class:

$$P(d_i|c_j;\theta) = P(|d_i|) \prod_{k=1}^{|d_i|} P(\omega_{d_{i,k}}|c_j;\theta) \tag{2.2}$$

where $\omega_{d_{i,k}}$ indicates the word at position k in segment i.

Naïve Bayes classifier is needed to learn using labeled data, $D = \{d_1,\ldots, d_{|D|}\}$, for estimation of the parameters. The estimation of the parameters θ, i.e. $\hat{\boldsymbol{\theta}}$ is achieved using maximum likelihood (ML) technique, hence finding $argmax_\theta P(\theta \mid D)$ given the evidence of the training data and a prior.

The probability of a word given its class, $\hat{\boldsymbol{\theta}}_{\omega_t|c_j}$, is the ratio of number of times word ω_t occurs in the training data for class c_j and total number of word counts in the training data for that particular class. The word probability $\hat{\boldsymbol{\theta}}_{\omega_t|c_j}$ is given by

$$\hat{\theta}_{\omega_t|c_j} \equiv \frac{1 + number\ of\ occurrence\ of\ \omega_t\ in\ class\ j}{number\ of\ words\ in\ class\ j}$$

$$\hat{\theta}_{\omega_t|c_j} \equiv P(w_t|c_j;\hat{\theta}) \equiv \frac{1 + \sum_{i=1}^{|D|} N(w_t,d_i) P(y_i = c_j|d_i)}{\sum_{s=1}^{|V|}\sum_{i=1}^{|D|} N(w_s|d_i)\, P(y_i = c_j|d_i)} \tag{2.3}$$

where $N(\omega_t|d_i)$ is the number of times word ω_t occurs in d_i and $P(d_i = c_j|d_i) = 1$ if segment i is in class j, or 0 otherwise.

The class probabilities can be calculated as

$$\hat{\theta}_{c_j} = \frac{1 + number\ of\ segment\ in\ class\ j}{|C| + |D|}$$

$$\hat{\theta}_{c_j} \equiv P(c_j|\hat{\theta}) \equiv \frac{1 + \sum_{i=1}^{|D|} P(y_i = c_j|d_i)}{|C| + |D|} \tag{2.4}$$

For the simplification of calculation we have augmented "*pseudo-counts*" (one for each word) both for numerator and denominator. The "*pseudo-counts*" comes from the prior distribution over θ. The technique that helps to use this type of prior is sometimes referred to as *Laplace smoothing*. This smoothing technique is required to avoid zero probabilities for infrequently occurring words.

2.2. Classifying New NSW with Naïve Bayes

We assume that there is a one to one correspondence between the target NSW and class label. During training, to get the value of posterior distribution, different entities of Bayes theorem are estimated from the available information according to Eq. 2.3 and 2.4. Given estimates of these parameters calculated from the training segments, we can classify a new NSW.

$$P\left(y_i = c_j \middle| d_i; \hat{\theta}\right) = \frac{P\left(c_j \middle| \hat{\theta}\right) P(d_i | c_j; \hat{\theta})}{P(d_i | \hat{\theta})} \qquad (2.5)$$

Now if the task is to classify a test segment d_i containing a NSW into a single class, then the class with highest posterior probability, $argmax_j\, P\,(y_i = c_j | d_i; \hat{\boldsymbol{\theta}})$, is being selected.

2.3. Learning a Naïve Bayes Model from Labeled and Unlabeled Data

In naïve Bayes model, we have shown the ML estimation given a set of labeled data. Now with labeled and unlabeled data we are going to estimate different parameters. Here, we are using mixture of labeled and unlabeled data [Ratsaby and Venkatesh 1995]. Since the labels of the unlabeled data are not known, closed-form equation cannot be evaluated here. However, we can overcome the problems using iterative Expectation Maximization (EM) algorithm (section 2.3.1) to calculate the ML parameters locally. EM algorithm is a numerically stable algorithm where each iteration increases likelihood. Under fairly general situation, it has consistent global convergence. The cost of each iteration is generally low, therefore can accommodate large number of iterations also. EM is a special type of algorithm that can be used to provide estimate of missing data. The algorithm is shown in Figure 1.

In semi-supervised naïve Bayes model, at first a naïve Bayes classifier is built from the limited number of labeled training data. Then, with the help of constructed model, we classify the unlabeled data. It is interesting to note that instead of mentioning the class labels, we are considering the probability distribution of each unlabeled data. In the next step, we rebuild the naïve Bayes classifier with all the class labels- given and estimated. We iterate this process until it converges to a stable state. At that instance, assign class labels to the unlabeled data. The pictorial representation of the above process is given below:

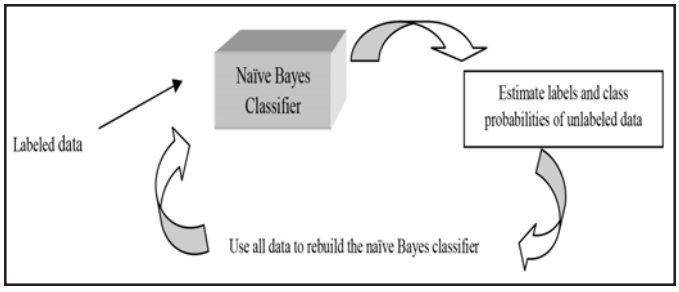

Figure 1. Schematic diagram of NSW classifier form labeled and unlabeled data

The whole task is outlined in sequential steps as follows:

1. A naïve Bayes classifier is built with labeled data following supervised learning technique.

2. Classify the unlabeled data using naïve Bayes classifier learned in step 1. Now the unlabeled data have been classified into most likely classes along with class probabilities associated with each class.

3. Rebuild a naïve Bayes classifier with all data – labeled and unlabeled. Class labels and class probabilities are assigned to the unlabeled data as calculated in step 2.

4. Iterate steps 2 and 3 of classifying the unlabeled data and rebuild the naïve Bayes model until it converges to a stable classifier having a set of labels for the data.

2.3.1 Expectation-Maximization

Expectation Maximization (EM) algorithm [Dempster 1977] is an iterative technique that takes labeled and unlabeled data [Nigam etal. 1998] as an input and iteratively rebuilds the classifier to get the maximum estimate of $\hat{\theta}$. Training data, D, is divided into two disjoint subsets, D^l and D^u. D^l are labeled data that have class labels $c_j \varepsilon C$, whereas D^u, unlabeled data, do not have any class labels. Therefore we can write $D = D^l \cup D^u$. The EM algorithm[1] is given below:

- **Input:** Collection of labeled, D^l, and unlabeled, D^u, segments where $D = D^l \cup D^u$.
- Initially build a naïve Bayes classifier, $\hat{\theta}$, from the labeled segments, D^l, only. Calculate the value of $\hat{\theta} = argmax_\theta P(\theta \mid D)$ using ML estimation.
- Loop while classifier parameters improve.
- **E-step**: Use the current classifier $\hat{\theta}$ to estimate class membership of each unlabeled segment, i.e., the probability that generated each segment, $P(c_j \mid d_i; \hat{\theta})$ [see Eq. 2.5]
- **M-step**: Re-estimate the classifier $\hat{\theta}$, given the estimated class membership of each segment. Use ML parameter estimation to find $\hat{\theta} = argmax_\theta P(\theta \mid D)$ [see Eq. 2.3 and 2.4]
- **Output**: A classifier, $\hat{\theta}$, that takes an unlabeled segment as input and predicts a class label.

Figure 2. The basic EM algorithm

The maximization of log likelihood is achieved by:

$$l(\theta|D) = \sum_{d_i \in D^u} log \sum_{j=1}^{|C|} P(c_j|\theta) P(d_i|c_j;\theta) + \sum_{d_i \in D^l} \log \left(P(y_i = c_j|\theta) P(d_i|y_i = c_j;\theta)\right) \quad (2.6)$$

This is an incomplete log probability because the labels are not given for the unlabeled data. This equation contains a log of sums for the unlabeled data. Therefore, it becomes computationally intractable when we try to maximize it by applying partial derivatives. However, if we have right to use class labels of all the segments (considering a binary indicator z_{ij}), then we can convert the incomplete log probability into complete log probability of the parameters, $log\ P(\theta \mid D; z)$, without considering a log of sums, since only one term within the sum would be non-zero. The complete log-likelihood becomes:

$$l(\theta|D;z) = \sum_{d_i \in D} \sum_{j=1}^{|C|} z_{ij} \log \left(P(c_j|\theta)\ P(d_i|c_j;\theta)\right) \quad (2.7)$$

3. Experiments and Results

The experiment is carried out into different steps, namely, primary classification, feature vector generation, final classification using naïve Bayes classifier and number to word conversion. In the primary classification [Kundu et al. 2013], we are extracting examples (sentences) containing NSW only from the initial databases created from Bengali news corpus (Anandabazar Patrika[2], Bartaman Patrika[3] and Aajkaal Patrika[4]) and English news corpus (The Times of India[5]) , thus avoiding manual separation from sentences not containing NSW. Hence we avoid from the unwanted sentences in the database and we are trying to minimize the overall complexities of the systems. Subsequently, based on NSW in each sentence, we target the *context window* [Kundu et al. 2013] both on left and right of a given NSW and ultimately produce segments. The objective of employing *context window* is to

[1] Adopted from [Nigam et al. 2000]

[2] http://www.anandabazar.com/

[3] www.bartamanpatrika.com

[4] www.aajkaal.net

[5] http://epaperbeta.timesofindia.com/

eliminate words that have either negligible or no contextual importance for any class of non-standard words. Moreover these words (proper nouns, verbs, articles, prepositions etc.) are common across all classes, and therefore do not have any contribution in the classification process. From the generated data base, *D*, we create word features which is a list of every distinct word presents in *D* [Ravikiran 2012, Luce 2012]. To train a classifier we require identifying what features are relevant [Ravikiran 2012, Luce 2012, Kundu 2014]. For that, we have generated a feature vector indicating what words are contained in the input *D* passed.

features column *j*
...
examples *i* 0 1 0 1 0 0 0 0 1....
1 0 0 0 1 1 0 0 0....
...

Each row represents one example (or, one sentence containing NSW), and each column represents one feature, where '1' denotes the existence of the feature in this context, and 0 denotes the nonexistence.

In the final classification step, the feature vector containing word features of labeled as well as unlabeled data becomes the input. In case of supervised learning, the classifier is learned with database, D (5372 labeled) and in case of semi-supervised learning, the classifier is trained with labeled (5372) as well as unlabeled (5372) texts. We have carried out experiment on Bengali and as well as English new corpus. During the training phase, we have selected the words having higher information gain. To find the highest information features, we require calculating information gain for every individual word. Information gain as described by Shannon (1948) for classification is a measure of how common a feature is in a particular class compared to how common it is in all other classes. A word that occurs primarily in one NSW class (e.g. 'Time') and rarely in another class (e.g. 'Quantity') is high information. For example, the presence of the word "a.m. / p.m." in a text is a strong indicator that the text most probably contains 'Time' NSW. That makes "a.m. / p.m." a high information word. It is interesting to note that the most informative features never change. That makes sense because the point is to use only the most informative features and ignore the rest.

Semiotic class	Accuracy of supervised NBN[7] on Bengali News corpus (in %)	Accuracy of supervised NBHIF[8] on Bengali News corpus (in %)	Accuracy of semi-supervised NBHIF on Bengali News corpus (in %)	Accuracy of semi-supervised NBHIF on English News corpus (in %)
Date and month	97.2	97.6	97.6	98.0
Money	99.8	100	100	100
Telephone no.	100	100	100	100
Year	97.2	97.8	98.0	98.8
Time	96.8	97.0	97.0	100
URL	100	100	100	100
Percentage	100	100	100	100
Quantity	96.6	97.0	97.0	98.0
Float	100	100	100	100

Table 1. Accuracy values for different semiotic classes

One of the best metrics for calculation of information gain is 'chi-square' method. Python[1] NLTK (natural language toolkit) contains this in the *BigramAssocMeasures class* in the metrics package. First we need to calculate frequency for each word: its overall frequency and its frequency for individual class. This is accomplished by a '*FreqDist*' function for overall frequency of

[6]Used Python 3 programming language

words, and a '*ConditionalFreqDist*' function where the conditions are the class labels. Once we have calculated these values, we can measure the score of words with the '*BigramAssocMeasures.chi_sq*' function, and then sort these words by score and consider the top 10000 words. We then place these words into a set, and use a 'set membership' test in '*feature selection*' function to choose only those words that appear in the set. Now each NSW is classified in presence of these high information words.

After proper learning process, test examples (500) containing a specific NSW class are created manually from the news papers and are submitted to the system. Then, NSWs have been classified into their proper classes considering principles discussed before. The 'number to word' conversion phase has been performed using different python modules where digit-form of a specific NSW class is transformed into its corresponding word format (e.g. in case of 'year', '1972' will be translated as 'nineteen seventy two' whereas in case of 'money', it will be 'one thousand nine hundred seventy two').

The following table shows the accuracy levels (in percentage) of different semiotic classes in Bengali news corpus and English news corpus:

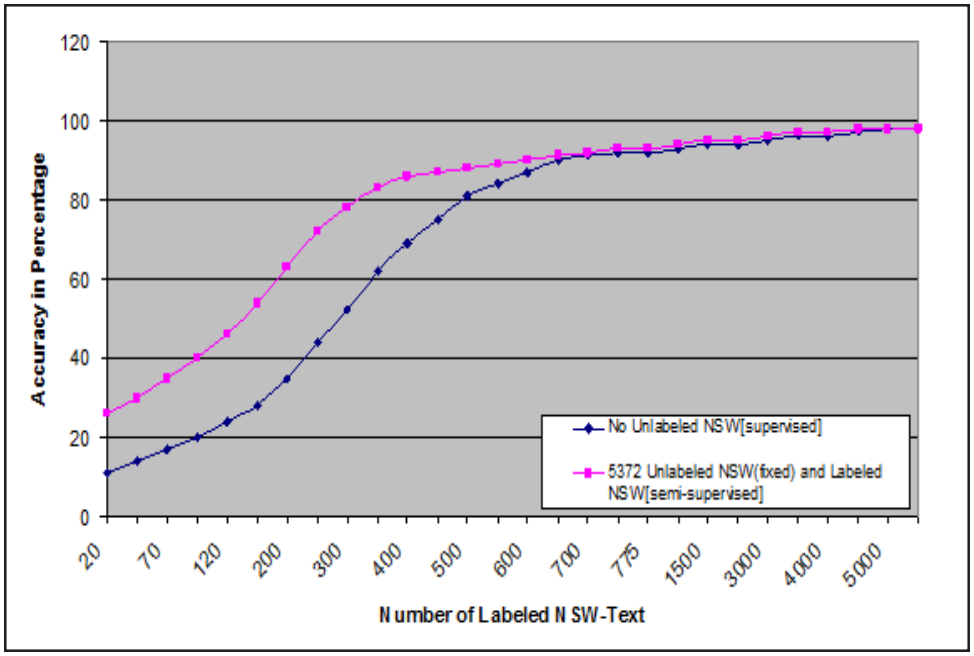

Figure 3. Accuracy values with and without unlabeled data

4. Conclusions

In general, EM estimates the value of θ, i.e. $\hat{\theta}$, that maximizes the posterior probability using labeled and unlabeled data. This technique offers the parameter estimation with limited number of labeled data and thereby helps to improve the overall accuracy. Naïve Bayes performs well when there is huge number of labeled data. In reality, EM algorithm significantly improves the accuracy of a NSW classifier, particularly when there are only a few labeled data. Figure 3 shows the effect of unlabeled data on NSW classifier. The experiment is carried out individually, one with 5372 unlabeled (fixed) and 5372 labeled data and another with only 5372 labeled data. EM performs significantly well than traditional naïve Bayes. For example, with 250 labeled data, naïve Bayes gives 44% accuracy whereas EM achieves 72%. It is required to mention that even with small number of labeled data i.e., 20, EM gives 26% while naïve Bayes gives only 11% accuracy. From the figure 3 we can see that with huge number of labeled data, curve converges to the curve with unlabeled data. So, we can conclude that with few labeled data, unlabeled data influences the accuracy levels but it does not make any significant difference with huge labeled data.

[7] NBG-naïve Bayes normal

[8] NBHIF-naïve Bayes with High Informative Features

The key lesson of this paper is that improved feature selection will improve the effectiveness of a classifier. Dimensionality reduction is one of the single best things that we can do to improve the performance of a classifier. It's reasonable to throw away data that is adding not enough value to the performance of a classifier and is making our model worse.

References

[1] Bennett, K. P., Demiriz, A., Maclin, R. (2002). Exploiting unlabeled data in ensemble methods. *In:* Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (KDD).

[2] Blum, A., Mitchell, T. (1998). Combining labeled and unlabeled data with Co-training. *In:* Proceedings of the Workshop on Computational Learning Theory, Madison, Wisconsin, USA 92—100.

[3] Carlin, B., Louis, T. (1996). Bayes and empirical Bayes methods for data analysis. *Chapman and Hall*.

[4] Cavnar, W. B., Trenkle, J. M. (1994). N-Gram-Based Text Categorization. *In proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval. Las Vegas, UNLV Publications/Reprographics* 161—175.

[5] Celeux, G., Govaert, G. (1992). A Classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis* 14 (3) 315—332.

[6] Dempster, A., Laird, N., Rubin, D. (1977) Maximum Likelihood from Incomplete Data Using the EM Algorithm. *Journal of the Royal Society of Statistics* 39 (1) 1—38.

[7] Feldman, R., Sanger, J. (2007). The Text Mining Handbook. Cambridge University Press.

[8] Ghani, R. (2001) Combining labeled and unlabeled data for text classification with a large number of categories. *In:* Proceedings of the IEEE International Conference on Data Mining.

[9] Göker, A., Davies, J. (2009). Information Retrieval: Searching in the 21st Century. *John Wiley & Sons Ltd*.

[10] Kundu, C., Das, R. K., Sengupta, K. (2013). Implementation of Context Window and Context Identification Array for Identification and Interpretation of Non Standard Word in Bengali News Corpus. *International Journal of Computational Linguistics Research* 4 (4): 159—171.

[11] Kundu, C. (2014). Identification and Interpretation of NSWs Using Variational Bayesian Inference in Bengali News Corpus. *International Journal of Computational Linguistics Research* 5 (4) 109—118.

[12] Luce, L. (2012). Twitter sentiment analysis using python and NLTK. http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk.

[13] McLachlan, G. and Krishnan, T. (1997). The EM Algorithm and Extensions. *New York: Wiley*.

[14] Merz, C. J., Clair, D. C. S., Bond, W. F. (1992). Semi-supervised adaptive resonance theory (smart2). *In:* International Joint Conference on Neural Networks 3: 851—856.

[15] Miller, D., Uyar, H. (1997). A mixture of experts classifier with learning based on both labeled and unlabelled data. *Advances in Neural Information Processing Systems 9, Cambridge, MA, MIT Press* 571—577.

[16] Neal, R. M. and Hinton, G. E. (1998) A view of the EM algorithm that justifies incremental, sparse and other variants in Learning in Graphical Models. *M.I.Jordan, Ed. Cambridge, MA: MIT Press* 355—368.

[17] Nigam, K., McCallum, A., Thrun, S., Mitchell, T. (1998) Learning to classify text from labeled and unlabeled documents. *Proceedings of the Fifteenth National Conference on Artificial Intelligence* 792—799.

[18] Nigam, K., McCallum, A. K., Thrun, S., Mitchell, T. M. (2000) Text classification from labeled and unlabeled documents using EM. *Machine Learning* 39 (3) 127—163.

[19] *Pakhomov, S.*, Buntrock, J., Duffy, P. (2002) Semi-Supervised *Maximum Entropy* Based Approach to Acronym and Abbreviation Normalization in Medical Texts, Association for Computational Linguistics (ACL), Philadelphia, 160-167.

[20] Ravikiran. (2012). How to build a twitter sentiment analyzer?. http://ravikiranj.net/drupal/ 201205/code/machine-learning/how-build-twitter-sentiment-analyzer.

[21] Ratsaby, J., Venkatesh, S. S. (1995). Learning from a mixture of labeled and unlabeled examples with parametric side information. *In:* Proceedings of the Eighth Annual Conference on Computational Learning Theory 412—417.

[22] Sproat, R., Black, A. W., Chen, S., Kumar, S., Osetendorfk, M., Richards, R. (2001) Normalization of non-standard words. *Computer Speech and Language*, 287—333.

[23] Szummer, M., Jaakkola, T. (2001) Kernel expansions with unlabeled examples. *Advances in Neural Information Processing Systems, Vancouver, British Columbia, Canada* 626—632.

[24] Yarowsky, D. (1996) Homograph Disambiguation in Text-to-Speech Synthesis. *Progress in Speech Synthesis, Springer-Verlag* 158—172.