Mathematical and Software Engineering, Vol. 3, No. 1 (2017), 78-86.
Varεpsilon Ltd, http://varepsilon.com

Comparative analysis of Mechanisms for Categorization and Moderation of User Generated Text Contents on a Social E-Governance Forum

Imeobong Frank Inyang, Simeon Ozuomba*, and Chinedu Pascal Ezenkwu

*Corresponding Author: simeonoz@yahoo.com
Electrical/Electronic and Computer Engineering Dept., University of Uyo, Uyo, Nigeria

Abstract

This paper presents a comparative analysis of two mechanisms for an automated categorization and moderation of User Generated Text Contents (UGTCs) on a social e-governance forum. Posts on the forum are categorized into "relevant", "irrelevant but interesting" and "must be removed". Relevant posts are those posts that are capable of supporting government decisions; irrelevant but interesting category consists of posts that are not relevant but can entertain or enlighten other users; must be removed posts consists of abusive or obscene posts. Two classifiers, Support Vector Machine (SVM) with One-Vs-The-Rest technique and Multinomial Naive Bayes were trained, evaluated and compared using Scikit-learn. The results show that SVM with an accuracy score of 96% on test set performs better than Naive Bayes with 88.6% accuracy score on the same test set.

Keywords: Moderation; Ranking; UGC; UGTC; web 2.0; Sentiment analysis; Social e-governance.

1. Introduction

Growing computerization and increasing Internet connectivity have encouraged the use of Information and Communication Technology (ICT) in the coordination and facilitation of several businesses. Moreover, the application of social network technologies for the purpose of improving governance has generated interest recently due to the emergence of web 2.0. In this paper, this has been referred to as social e-governance. The essence of social e-governance forums is that they encourage candid opinions from the citizens thereby promoting people-oriented decisions by the government. In view of this, there is a need for mechanisms that can be used to categorize and moderate users' posts to ensure that only relevant or interesting posts are allowed on the platform. Moderation is the process of reviewing a UGC and taking decision on whether to delete it or allow it to be accessed by other users. Moderation can be an automated moderation, using computer applications and algorithms; community moderation, which leverages the online community to self-moderate contents and human moderation, in which there is a dedicated staff acting as a moderator. Three moderation approaches include – pre-moderation, reactive moderation

and post-moderation. Unlike in post-moderation where posts are allowed to appear online before moderation, in pre-moderation, all posts are moderated before they appear online. This moderation approach requires more prompt response; as such the best method for pre-moderation is the automated moderation. Human moderation cannot provide 24 hours 7 weeks moderation because posts submitted overnight or in the weekend may not be moderated until the next working days. Moreover, community moderation warrants other users to access posts and react accordingly. This, in other words, is a reactive moderation. Reactive moderation is a variant of post-moderation whereby the online community, instead of a dedicated individual, carryout the function of moderation. The danger with post-moderation is that the post might already have a negative impact on the online community before it is deleted; as such, post-moderation is not encouraged where the risk associated with publishing inappropriate contents is high. Furthermore, community moderation is prone to Sybil or one-man-crowd attack, whereby a user creates multiple accounts or sockpuppets in order to influence votes on posts in an online community. In view of this, automated moderation is indispensable, since it does not give room to Sybil attack, being human independent. There are several sentiment analytic techniques employed to automate the process of UGC moderation on online communities. Some popular machine learning classifiers used in sentiment analysis include Naive Bayes classifier, SVM, decision tree, random forest and so on.

In this paper, mechanism for automated moderation of an e-governance forum is presented. The paper considered the performances of two classifiers, which are SVM and Naive Bayes classifiers. The classifiers are trained and evaluated using text corpus generated by a group of three hundred (300) students on a locally hosted e-governance forum. Each student was encouraged to generate at least eight different texts. The texts are to belong to “relevant”, “irrelevant but interesting” and “must be removed” categories. Summarily, the text corpus used for the training and evaluation of the classifiers contains a total of two thousand and twenty (2020) texts. 730 of the texts belong in the relevant category; 653 belong in the irrelevant but interesting category while 637 belong in the must be removed category. Using this text corpus, Support Vector Machine (SVM) with One-Vs-The-Rest technique and Multinomial Naive Bayes were trained using Scikit-learn. SVM proved better than Naive Bayes for the e-governance system. Subsequent sections of the paper include literature review, methodology, results and conclusion.

2. Review of Relevant Literatures

2.1. E-governance

According to Keohane and Nye [1], “Governance implies the processes and institutions, both formal and informal, that guide and restrain the collective activities of a group. Government is the subset that acts with authority and creates formal obligations. Governance need not necessarily be conducted exclusively by governments. Private firms, associations of firms, nongovernmental organizations (NGOs), and associations of NGOs all engage in it, often in association with governmental bodies, to create governance; sometimes without governmental authority.” In Kettl [2] view, "Governance is a way of describing the links between government and its broader environment - political, social, and administrative." With the revolutionary changes that ICTs are bringing to our global society, governments worldwide continue to develop more sophisticated ways to digitize its routines and practices so that they can offer the public access to government services in more effective and efficient ways. The delivery

of government services and information to the public using ICT is referred to as e-governance [3]. The UNESCO define e-governance as "the public sector's use of information and communication technologies with the aim of improving information and service delivery, encouraging citizen participation in the decision-making process and making government more accountable, transparent and effective. E-governance involves new styles of leadership, new ways of debating and deciding policy and investment, new ways of accessing education, new ways of listening to citizens and new ways of organizing and delivering information and services. E-governance is generally considered as a wider concept than e-government, since it can bring about a change in the way citizens relate to governments and to each other. E-governance can bring forth new concepts of citizenship, both in terms of citizen needs and responsibilities. Its objective is to engage, enable and empower the citizen" [4]. Social networks provide the technological platform for individuals to connect, produce and share content online [5]. Web 2.0 has changed the one-way notion of traditional e-governance, whereby information only flows from government to the citizens. Nowadays, there is a need for government to access firsthand information from the citizens, so as to encourage grassroots development and targeted governance. The use of social networks as a tool to facilitate e-governance has been referred to this paper as social e-governance.

2.2. Moderation in Social Networks

According to Ochoa and Duval [6] "UGC is becoming the most popular and valuable information available on the WWW". The explosive growth of UGC has stimulated interests in moderation on social networks. Khadilkar, Pai, and Ghadiali [7] observed that "4.1 million minutes of video are uploaded to YouTube everyday … six billion images per month are uploaded to Facebook … 40% of images and 80% of videos [created]are inappropriate for business. UGC comes in different forms, including short-text content family such as tweets and forum comments; long-text posts on blogs and profiles; and multimedia material such as images, audio, video and applications". Moderation is the review of user generated content and the decision to publish, edit or delete the content or at times to engage with the online community [8]. Interactive advertising bureau Australia [9] opined that all stakeholders have a role in managing user comments on the web, as follows – " Users should think about the appropriateness of their content before they post it and take responsibility for their comments; Platforms should remove comments reported to them which are illegal or violate their terms and conditions and empower organizations using their platforms with tools to assist them in moderating their properties; The community should report comments that violate applicable rules; and Organizations should engage in responsible moderation of user comments posted to their social media channels". Maintaining a content is a foundation of a healthy and flourishing community platform. In order to maintain this quality, the community platform needs governance. Governance of a web community can be understood as steering and coordinating the activities of community members. Moderation is extremely important in social networking systems, sorting good from bad content and helping readers to find useful information. Khadilkar et al [7] stated that moderation can be automated moderation; community moderation and human moderation. Automated content moderation has grown into a discipline that requires expertise in pattern detection and labelling, the less downstream volume and analysis [7]. These automated moderation techniques are embodied under the subject of sentiment analysis. According to Liu [10] "sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, evaluations,

appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes". Most machine learning algorithms are often used for sentiment analysis. The following section reviews Naive Bayes algorithms and SVM.

2.3. Naive Bayes Algorithm

Naive Bayes is a family of probabilistic classifiers that leverages the Bayes' theorem with strong independence assumptions among the features. Naive Bayes has been well-applied in text categorization. An important advantage of naive bayes is that a small number of training data is sufficient to estimate the parameters necessary for out-of-sample classifications [11]. Given a class variable y and a dependent feature vector x throughx_n, Bayes' theorem states the following relationship:

$$P(y|x_1, \dots\dots, x_n) = \frac{P(y)P(x_1,\dots\dots,x_n|y)}{P(x_1,\dots\dots,x_n)} \quad (1)$$

Introducing the Naive Bayes independence assumption that
$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, .., x_n) = P(x_i|y)$ for all i, the equation (1) is simplified to equation (2);

$$P(y|x_1, \dots\dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1,\dots\dots,x_n)} \quad (2)$$

$P(x_1, \dots\dots, x_n)$is a normaliser and it is constant given the input. Naive bayes uses Maximum a posterior (MAP) decision rule in choosing the hypothesis that is most probable. Naive Bayes classifier uses the classification rule;

$$\hat{y} = \underset{y}{argmax}\, P(y)\prod_{i=1}^{n} P(x_i|y) \quad (3)$$

Based on the distributions of features, Naive Bayes classifier can be Gaussian, Bernoulli or multinomial. Gaussian Naive Bayes is used when dealing with continuous data with the assumption that the features are distributed according to Gaussian distribution.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}}\exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right) \quad (4)$$

The parameters $\sigma_y$and $\mu_y$are estimated using maximum likelihood.
Bernoulli Naive Bayes is for data that is distributed according to Bernoulli distributions. In the case of text classification using multivariate event model, word occurrence vectors, rather than word count vectors, are often used to train and use the classifier. Multinomial Naive Bayes is used for multinomially distributed data. It is uses word count vectors instead of word training vectors in training and using the classifier. The distribution is parameterized by vectors $\theta_y = (\theta_{y1}, \dots, \theta_{yn})$ for each class y, where n is the number of features (in text classification, the size of the vocabulary) and θ_{yi} is the probability $P(x_i|y)$ of feature i appearing in a sample belonging to class y.
The parameter θ_y is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

$$\hat{\theta}_y = \frac{N_{yi}+\alpha}{N_y+\alpha n} \quad (5)$$

Where, N_{yi}=$\sum_{x\in T} x_i$ is the number of times feature i appears in a sample of class y in the training set T, and N_y= $\sum_{i=1}^{|T|} N_{yi}$ is the total count of all features for class y.
The smoothing priors $\propto \geq 0$ accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting $\alpha = 1$ is called Laplace smoothing, while $\alpha < 1$ is called Lidstone smoothing.

2.4. Support Vector Machine (SVM)

SVM constructs a hyper-plane or a set of hyper-planes in a high dimensional space for the purpose of classification, regression or outline detection. It chooses the hyper-plane that has the largest distance to the nearest data points of any class so as to lower the generalization error of the classifier.

Given training vectors $x_i \epsilon \mathbb{R}^p, i = 1, \dots, n$,in two classes and a vector $y \epsilon \{1, -1\}^n$, SVM solves the following primal problem:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \quad \text{Subject to } y_i(w^T \phi(x_i) + b) \geq 1\text{-}\zeta \qquad \zeta \geq 0, i, \dots \dots n \tag{6}$$

Its dual is

$$\min_{\propto} \frac{1}{2} \propto^T Q \propto - e^T \propto \qquad \text{Subject to } y^T \propto = 0 \qquad 0 \leq \propto \leq C, i = 1, \dots, n \tag{7}$$

Where e is a vector of all ones, C>0 is the upper bound, Q is an $nbyn$ positive semi-definite matrix. $Q_{ij} = y_i y_j K(x_i, x_j)$; where, $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$is the kernel. The function ϕ implicitly maps the training vectors to higher dimensional space. The decision function is given as $(\sum_{i=1}^{n} y_i \propto_i K(x_i, x_j) + b)$.

3. Methodology

Figure 1 presents the research process flow. The text corpus comprises 2020 texts generated by 300 university students on a locally hosted e-governance forum. Summarily, the text corpus used for the training and evaluation of the classifiers contains a total of two thousand and twenty (2020) texts. 730 of the texts belong in the relevant category; 653 belong in the irrelevant but interesting category while 637 belong in the must be removed category. The texts were labeled accordingly for supervised learning.

Feature Extraction: This is the process of converting the texts in the corpus into numerical features compatible with machine learning techniques. The processes of feature extraction include lower casing; removal of stop words from each text in the text corpus; removal of non-word and word stemming;

Lower casing – The entire texts in the corpus are converted to lower case so as to ignore capitalization.

Removal of stop words- In python, NLTK library can be used to import stop words in different languages. Using this library, stop words in English language were imported and removed from each text in the text corpus.

Removal of words that occur too rarely in the corpus – To avoid over-fitting of the training set, words which occur less than 100 times in the corpus are removed.

Removal of non-words - All non-words including punctuations are removed. White spaces such due to tabs, spaces, newlines, etc. are trimmed to single space character.

Word Stemming – Words are reduced to their stem forms. For examples, words like discounted and discounting are replaced with discount. Words like include, includes, included and including are reduced to includ. This is achieved in Python using a stemmer function present in NLTK library.

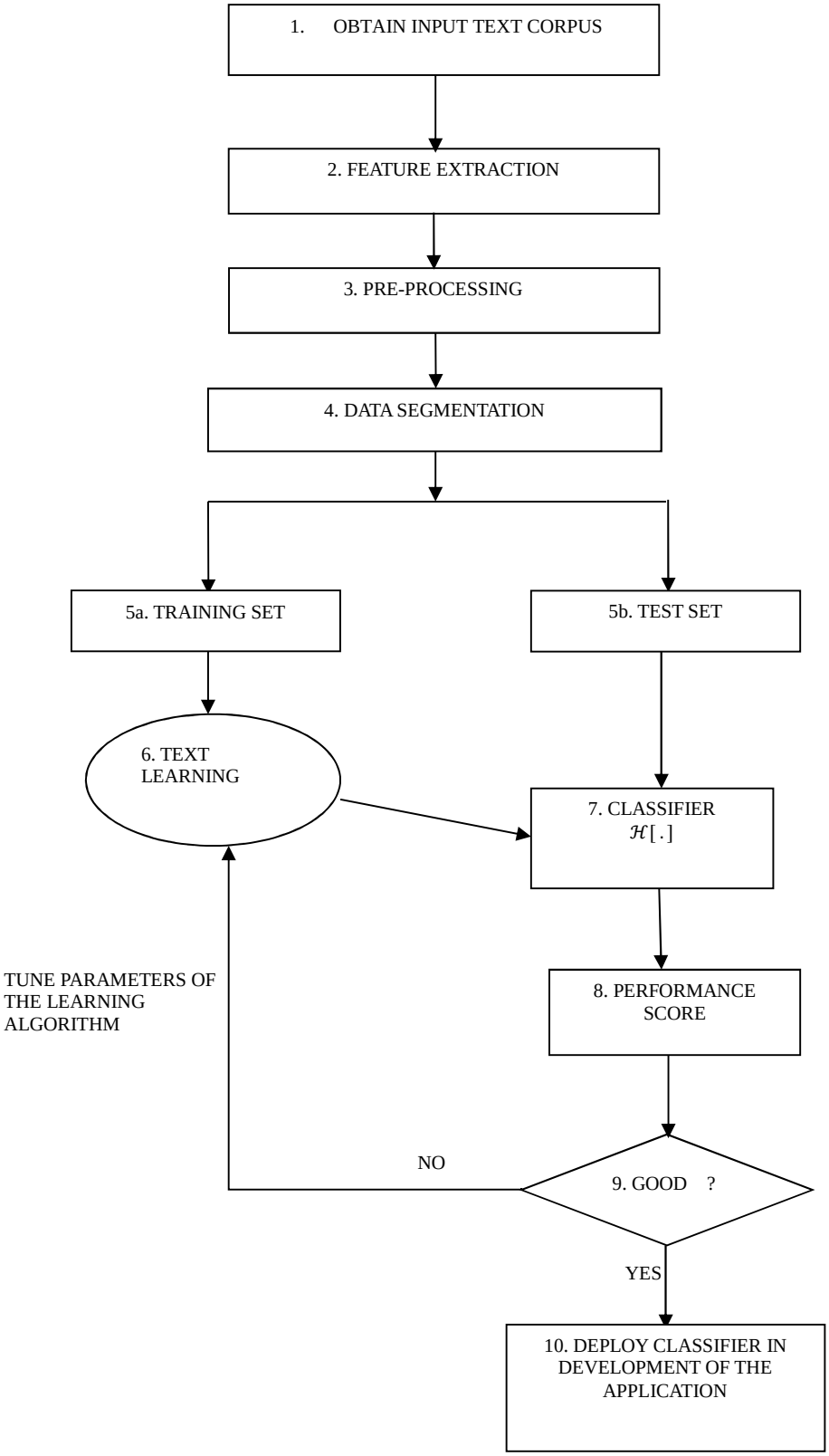

Figure 1: Flow Diagram for the Research process

Bag-of-words representation - A bag-of-word representation is the representation of a corpus of text documents in a matrix with one row per document and one column per token occurring in the corpus. The texts in the corpus are represented as numerical feature vectors with a fixed size rather than the raw text documents with variable lengths. Scikit-learn has functionalities for building the bag of words. The strings are

tokenized using white spaces as separators. Integer indexes are given to each possible token. The occurrences of tokens in each text document are counted. Each individual token occurrence frequency is treated as a feature. The vector of all the token frequencies for a given document is considered a multivariate sample. In Scikit-learn, the CountVectorizer function is designed for this purpose.

Pre-processing: The features were scaled to lie between 0 and 1. This was achieved in Scikit-learn using MinMaxScaler function present in the preprocessing library of Scikit-learn.

Data Segmentation: The data is randomly split such that 80% were used for training while 20% were used for training. The essence of this is to ensure that each classifier is validated with out-of-sample inputs, as such, this is a better proof of the system's generalization performance.

4. System Implementation

4.1. Training of Classifiers

The two classifiers, Naive Bayes and SVM, considered in this paper, were trained on the text corpus. The classifiers were implemented using Scikit-learn. Scikit-learn involves 4-step modelling pattern. In step one the relevant classes are imported. Step two involves the instantiation of the estimator, in which the hyper-parameters can as well be specified or left as defaults. In step three the model is fitted with data and step four is to apply the fitted model on the test set. For example, the 4-step modelling pattern of Scikit –learn for the Multinomial Naive Bayes is shown in the appendix. The SVM classifier was also implemented using the same 4-step modelling pattern in Scikit-learn. The SVC class was used due to its ability to implement multiclass classification on a dataset. The default hyper-parameters were used without tuning.

4.2. Performance Scores of Classifiers

Each of the classifiers were evaluated using the accuracy_score function in the accuracy library in Scikit-learn. The result shows that SVM classifier had a 96% out-of-sample performance while Naive Bayes had an 88.6% out-of-sample performance. Figure 2 and 3 show the implementations of the Naïve Bayes and SVM classifiers in Scikit-learn.

```
In [5]: import pandas as pd
        from token import *
        from labels import *
        from sklearn.naive_bayes import MultinomialNB
        from sklearn.cross_valiation import train_test_split
        from sklearn.metrics import accuracy_score
        data = pd.read_csv('C:\Users\Chinedu\Desktop\OfData.csv')
        nb = MultinomialNB()
        input_X = data[feature_attr]
        output_y = data[label]
        X_train,X_test,y_train,y_test =train_test_split(input_X,output_y,random_stat
        nb.fit(X_train,y_train)
        pred = nb.predict(X_test)
        print accuracy_score(pred,y_test)

0.885704323402
```

Figure 2: Multinomial Naïve Bayes implementation in Scikit-learn

```
In [17]: import pandas as pd
         from token import *
         from labels import *
         from sklearn import svm
         from sklearn.metrics import accuracy_score
         from sklearn.cross_valiation import train_test_split
         data = pd.read_csv('C:\Users\Chinedu\Desktop\OfData.csv')
         clf = svm.SVC(decision_function_shape='ovo')
         input_X = data[feature_attr]
         output_y = data[label]
         X_train,X_test,y_train,y_test =train_test_split(input_X,output_y,random_stat
         clf.fit(X_train,y_train)
         pred = clf.predict(X_test)
         print accuracy_score(pred,y_test)

         0.962802289601
```

Figure 3: One-Vs-all SVM implementation in Scikit-learn

4.3. Development of the Application

The social e-governance application was developed following an evolutionary software development process model. The process involves the system analysis, design, implementation, testing and deployment. The system was implemented with Python as the scripting language and deployed locally on Google App Engine for demonstration and testing. SVM classifier was used for the users' posts moderation and categorization. Figure 4 shows the screenshot of the system.

Figure 4: Screenshot of the system

5. Conclusion

In this paper, two classifiers, Naive Bayes and SVM were compared for UGTCs moderation and categorization using Scikit-learn. The result shows that SVM classifier had a 96% out-of-sample performance while Naive Bayes had an 88.6% out-of-sample performance. The social e-governance application was developed using python as scripting language. The SVM classifier was employed for the users' posts moderation

and categorization. Furthermore, the application was deployed locally on Google App Engine for demonstration and testing.

References

[1] Keohane, R.O., & Nye, J.S.Jr. (2002). Governance in a globalization world. Power and governance in a partially globalized world, 193-218.

[2] Kettl, D.F. (2015). The transformation of governance: Public administration for the twenty-first century. JHU Press.

[3] OJO, J. S. (2014). E-governance: An imperative for sustainable grass root development in Nigeria. Journal of Public Administration and Policy Research, 6(4), 77-89.

[4] Palvia, S.C.J., & Sharma, S.S. (2007). E-Government and E-Governance: Definitions/Domain Framework and Status around the World. In International Conference on E-governance., 5 International Conference on EGovernance, Foundations of E-Government, 1-12.

[5] Cvijikj, I.P. and Michahelles, F. (2012) Understanding the user generated content and interactions on a Facebook brand page, Int. J. Social and Humanistic Computing, Vol. 2, No. 1-2, 118–140.

[6] Ochoa, X., Duval, E. (2008). Quantitative analysis of user-generated content on the web. Proceedings of WebEvolve2008: web science workshop at WWW2008, 1-8.

[7] Khadilkar, A., Pai, T., Ghadiali, S. (2012). How to De-Risk the Creation and Moderation of User-Generated Content, Available at : http://www.cognizant.ch/InsightsWhitepapers/ How-to-De-Risk-the-Creation-and-Moderation-of-User-Generated-Content.pdf. Accessed on: 10th October 2016.

[8] ABC Managing Director (2011). Moderating User Generated Content, 9, Available at: http://about.abc.net.au/wp-content/uploads/2012/06/GNModerationINS.pdf. Accessed on: 10th October 2016.

[9] Interactive advertising bureau Australia (2013) Best Practice for User CommentModeration: Including commentary for organisations using social media platforms. Available at: https://www.iabaustralia.com.au/uploads/uploads/2013-09/ 1380477600_b054b0ef30db4de990bd1527ed6758e4.pdf, Accessed on: 10th October 2016.

[10] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1), 1-167.

[11] Kumar, S. A., &Vijayalakshmi, M. N. (2012). Inference of Naïve Baye's Technique on Student Assessment Data. In Global Trends in Information Systems and Software Applications, Volume 270 of the series Communications in Computer and Information Science, 186-191.