Application of text mining techniques for classification of documents: a study of automation of complaints screening in a Brazilian Federal Agency

Patrícia Maia*, Rommel N. Carvalho*, Marcelo Ladeira†, Henrique Rocha* and Gilson Mendes*
* Department of Research and Strategic Information (DIE)
Brazilian Office of the Comptroller General (CGU)
Brasília, Brazil
Email: {patricia.andrade, rommel.carvalho, henrique.rocha, liborio}@cgu.gov.br
† Department of Computer Science (CIC)
University of Brasília (UnB)
Brasília, Brazil
Email: mladeira@unb.br

***Abstract*—The Brazil's Office of the Comptroller General (CGU) is the agency of the Federal Government in charge of assisting the President of the Republic in matters related to internal control activities, public audits, corrective and disciplinary measures, corruption prevention and combat, and coordinating ombudsman's activities. Through its website, citizens submit complaints related to corruption and misuse of public sources to CGU. These complaints must be screened and delivered to the corresponding department based on its content. Nowadays the complaints screening is done manually and they are delivered to one of the 82 (eighty two) different units from various departments at CGU. This paper presents a proof of concept model to automatically classify the complaints in order to increase the speed of the current screening process. The proposed model was built using text mining and achieved an F-measure of 0.84 using Random Forest.**

I. Introduction

The Brazil's Office of the Comptroller General (CGU), control and transparency agency, is constantly concerned in keeping the population motivated in social control activities. Therefore one of its goals is to increase citizens participation in the corruption prevention program. In order to achieve this goal, CGU has created an online form, available in its website[1], that allows Brazilian citizens to collaborate with the auditing of public resources by denouncing irregularities.

This communication channel aims to stimulate reports of actions related to the defense of public assets, the control over the usage of federal public resources, the corruption prevention, the ombudsman's activities, and the improvement of federal public management and transparency.

Besides that, the space reserved to the complaints is a text field without any kind of requirements in order to allow and encourage citizens to use it. Nevertheless, this simplification of having only a free text field to be filled hinders the extraction of important/relevant information to classify the complaint. Thus, the process of manually analyzing these complaints for screening purposes becomes complex and time consuming.

As a consequence, the increasing number of complaints received per day has far surpassed CGU's capacity of manually analyzing them. Therefore, some of these complaints have become overdue or outdated.

In this scenario, not being able to analyze a complaint or even delaying its analysis can generate irreversible damages to the Government image, which is very difficult to restore. A good example of this fact can be a complaint about a supposed fraudulent hiring. The timely analyses of this complaint could avoid the incorrect use of public resources, avoiding unnecessary losses. Moreover, those citizens that contributed by submitting complaints related to this supposed fraudulent hiring will probably have an increased feeling of impunity.

In order to overcome this problem of small throughput in complaint analysis, we propose the use of text mining and machine learning in order to learn a model to automatically classify the complaints. This model will eventually replace or at least speed up the current manual screening process done by CGU's civil servants. Besides improving the quality of the service provided to society, this automation will also make it cheaper, since it will significantly decrease the total number of hours needed to analyze each complaint.

This paper presents a proof of concept and feasibility study of the development of an automatic process for screening complaints made via CGU's webpage. The process hereby described was built using text mining [1] and machine learning [2] following the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology [3].

This paper is organized as follows. Section II gives an overview of related work in text mining classification. Section III describes our approach to classify the complaints

[1]http://www.cgu.gov.br/denuncias/

received. Section IV presents the results achieved. Finally, Section V outlines some conclusions and future work directions.

II. Related Work

We present three different types of text mining use for classifying documents, which are similar to the work presented in this paper. The first, from Youn and McLeod [4], classified e-mails as spam or not spam using four different classifiers: Neural Network, Support Vector Machine (SVM), Naive Bayesian, and Decision Tree - J48. The second, from Navadiay [5], used Naive Bayes and SVM to classify web pages. Finally, Adeva *et al.* [6] realized an experiment to screening medicine papers using Naive Bayes, SVM, K-Nearest Neighbours (KNN), and Rocchio classifiers. These are not the only studies on e-mail, web page, and paper classification. Nevertheless, they are used to illustrate these related areas of research.

Youn and McLeod [4] classified e-mails using four different classifiers: Neural Network, SVM, Naive Bayes, and Decision Tree. All e-mails were classified as spam or not spam. The Naive Bayes and Neural Network models did not obtain a good result. The J48 model showed a better result than the other models with a 95% accuracy. One of the main difference between this work and ours is that we deal with four classes, which makes the classification task harder. Furthermore, spam text usually presents specific characteristics that do not show up in most of regular e-mails, such as money offers and pornography, which makes it easier to differentiate between the two.

Navadiay [5] used Naive Bayes and SVM to classify web pages based on both content and some structure data. He classifies web pages in four categories: conference, course, department, and student. The Naive Bayes model, with a 92% accuracy, presented a slightly better result than the SVM model, with 91% accuracy. An interesting difference from our work is that the Naive Bayes and SVM models presented the worst results in our classification problem. Furthermore, we only rely on free text for our classification problem.

Adeva *et al.* [6] realized an experiment to screening medicine papers using Naive Bayes, SVM, K-Nearest Neighbours (KNN) and Rocchio [7]. Rocchio is a profile-based classification algorithm adapted from the classical Vector Space Model with Term Frequency - Document Frequency (TF/DF) weighting and relevance feedback to the classification process. In this work, they apply multiple machine learning algorithms combined with several feature selection techniques to different parts of the paper (title, abstract, or both). They conclude that the selected method for feature selection can make a significant difference. The results obtained using only the title alone was as good as when considering the abstract as well. SVM and Naive Bayes presented positive results in terms of overall precision and recall measurements, reaching values of up to 84%.

To the best of our knowledge, there is no other work on using text mining to classify complaints received from third parties. The work we present is the first attempt to automatically classify governmental complaints related to corruption and misuse of public resources.

III. Experiment

Presently there are more then 6,000 complaints in CGU's database servers waiting to be screened and analyzed. On a daily basis, Brazilian citizens submit 32 new complaints, while CGU is capable of analyzing around 20 per day. At this rate, in a year from now CGU will have about 10,400 complaints still waiting to be analyzed, an increase of 73%. Thus, it becomes obvious that without a major change in its screening process CGU will never be able to analyze all complaints submitted by the population. As previously discussed, this would bring about not only an irreversible damage to the Brazilian Government image but also possible financial losses.

One of the main activities in the screening process is analyzing what are the main topics related to the complaint and if they are relevant (*e.g.*, if it is not just junk message, spam, etc). Then, based on these topics, the complaints are distributed to different internal departments at CGU. Finally, each department will prioritize which complaints should be analyzed and investigated. All others are archived.

CGU has four main areas: Audit and Inspection, Preventing Corruption, Disciplinary Action, and Ombudsman's Office. These areas are divided in different departments.

The Audit and Inspection area is responsible for carrying out audits and inspection activities to check how public resources are being used. This task is carried out by CGU through the Federal Internal Control Secretariat (SFC, in Portuguese), which is the unit in charge of evaluating the execution of Federal Government budgets, inspecting the implementation of governmental programs, and auditing the management of federal public funds under the responsibility of public and private agencies and organizations, among other functions [8].

The Preventing Corruption is in charge for developing mechanisms to prevent corruption. The idea is that, besides detecting cases of corruption, CGU has the role of acting proactively by developing means to prevent their occurrence. CGU carries out this activity through its Transparency and Corruption Prevention Secretariat (STPC, in Portuguese) [9].

The Disciplinary Action area comprises activities related to the investigation of possible irregularities committed by civil servants and the enforcement of the applicable penalties. The CGU unit responsible for disciplinary activities is the National Disciplinary Board (CRG, in Portuguese) [10].

The Ombudsman's Office is responsible for receiving, examining, and forwarding complaints, praise, and suggestions referring to procedures and actions of Federal Executive agents, units, and entities. The National Ombudsman's Office is also in charge of the technical coordination of the segment of Ombudsman's Offices of the Federal Executive Branch, as well as the organization and interpretation of the complaints, praise, and suggestions received, and produces quantified surveys on the level of satisfaction of users of public services provided in the scope of the Federal Executive Branch [11].

The Citizen Treatment Department (CGCID, in Portuguese), from Ombudsman's Office, is the unit responsible for realizing the screening process. At the end of the process, it will forward the complaint to the unit responsible for the corresponding subject.

Nowadays, CGU does not have any automatic process to realize the complaints screening. Thus, it is done manually and individually, consuming an excessive amount of time and resources.

The complaints are stored in a SQL Server database. There are more than 60,000 complaints that have already been distributed to 82 different units at CGU.

Since we wanted to validate the results with each unit in order to understand the problems with the misclassification and how we could improve it, we decided to work with only a small subset of units. As a proof of concept, we decided to classify the complaints of only 4 out of the 82 available. The units wiling to help with this validation, thus the ones we used in our work, were all from the Audit and Inspection area: health care auditing unit (DSSAU, in Portuguese), cities auditing unit (DIURB, in Portuguese), agrarian development auditing unit (DRDAG, in Portuguese), and tourism and sports auditing unit (DRTES, in Portuguese).

DRSSAU is responsible for the national health system program (SUS, in Portuguese). DIURB usually investigates the construction of popular houses, urban paving, and storm water drainage. DRDAG takes care of the complaints involving the strengthening small family farming national program (PRONAF, in Portuguese) and the technical, social, and environmental assistance in agrarian reform program (ATES, in Portuguese). DRTES is responsible for programs related to the Tourism Ministry and the Sport Ministry.

Although there are other fields besides the free text field used to describe the complaint, they were not used in our work, since they are not used today during the screening process. Nevertheless, these fields can and are used during the analysis done by each unit. Besides that, citizens can also upload documents, photos, or any other type of file in order to provide evidence that the complaint being made is real. However, these are also ignored during the screening process.

The data used in this work were extracted from the SQL Server where the complaints are stored and imported into RStudio[2], the tool we used for text mining and machine learning. The only data used for classification was the description of the complaint. Besides that, we retrieved the ID of the complaint and the unit where it was sent to, *i.e.*, the class. Since the description is unstructured and freely typed by the citizen that submitted the complaint, it is subject to language errors. This kind of error can hinder the classification model. Therefore, it is necessary to treat it before using the text for learning.

In order to improve the performance of the generated models, the classifiers, we processed the data of the description text field. In this stage we applied techniques of stemming, removing stop words and low frequency words, changing upper to lower case letters, and removing punctuation, accentuation, numbers, and white spaces. These techniques are common during the text mining process and are discussed in more detail in [1].

For implementing the techniques of text mining, we used the text mining framework provided by the TM package in RStudio [3]. This package makes it possible to process, organize, transform, and analyze textual data[12].

We used the function `removeSparseTerms` [4], present in TM package. This function removes words that occur only in a few documents, without losing significant terms. This function was tested with the values 0.99 and 0.96 for the parameter sparse. In the first text, the result generated about 6,000 terms, while the second generated 1,200 terms. We applied tests in both sets, but there was no significant difference in the results. Considering the memory consumption for automated text classification, we decided to work with the the second parameter because it generated fewer words, thus generating less memory consumption.

Then, the Term-Document Matrix (TDM) was constructed with Term Frequency - Inverse Document Frequency (TF-IDF) parameter.TDM is a matrix where the rows represent the words (terms) and the columns represent the documents. For more details about TDM and TF-IDF, see [13]. Beyond the terms and documents, the matrix has another column, unit, which represents the class.

The term-document matrix data were divided in three parts: training, validation, and test. The first received 60% of the data and the others 20% each. The training data is used for learning the classification models. The validation data is used to adjust parameters used in each learning algorithm and to select the best model. Finally, the test model is used to evaluate the selected classification model performance, in order to verify if it can be generalized to unseen data.

These sets were trained and validated to generate the classification models with four machine learning algorithms: Decision Tree, Naive Bayes, Random Forest, and SVM. SVM [14] is a machine learning algorithm used for classification and regression analyses. This algorithm was chosen because it has been used in related works. Besides, Adeva *et al.* [6], for instance, achieved an F-measure of 0.877 with SVM in the corresponding classification problem. The R package used to generate the model with SVM was `e1017` [5]. Random Forest is an algorithm that uses a set of decision trees simultaneously constructed considering all the variables selected for analyses[15]. It is widely applied to text classification [16], [17], [18]. In R, the package used was `randomForest`[6]. Decision Tree creates a tree that maps the observations of each class according to the values in their attributes [19]. Youn and McLeod [4] achieved good results with it when classifying spam e-mails. In R, we used the `tree` package[7].

[2]RStudio IDE is a powerful and productive user interface for R. It is available at https://www.rstudio.com/.

[3]http://cran.r-project.org/web/packages/tm/index.html

[4]http://cran.r-project.org/web/packages/tm/index.html

[5]http://cran.r-project.org/web/packages/e1071/index.html

[6]http://cran.r-project.org/web/packages/randomForest/index.html

[7]http://cran.r-project.org/web/packages/tree/index.html

Naive Bayes is a probabilistic classification algorithm in which graphs represent the probabilities of a set of random variables and their interdependencies [20]. It is a simple, yet effective algorithm. Adeva *et al.* [6], for instance, achieved good results applying this algorithm in text classification. The package in R used for Naive Bayes was `KlaR`[8].

IV. Results

As explained in Section III, we learned the model using the training data, selected the best model by comparing its performance based on the validation data, and finally tested the performance of the selected model with the test data. This section presents these analysis and results.

According to [21], experimental evaluation of a classifier usually measures its effectiveness, that is the ability to make correct classification decisions. In text mining and machine learning, the results of evaluation are done using performance metrics like accuracy, precision, recall, F-measure, kappa, among others. In this section, our classification models were evaluated using the following measures: sensibility, specificity, Kappa, F-measure, and the confusion matrix itself.

Confusion matrix is usually used to evaluate the performance of an algorithm. This matrix shows the number per class of correctly classified and mislabeled instances [22].

Sensitivity is the proportion of positives that are correctly recognized as such. It is the percentage of correct items that are selected. Specificity is the proportion of negatives that are correctly recognized as such. It is the percentage of selected items that are correct [23]. Figure 1 presents a visual example of sensitivity and specificity for the DSSAU unit.

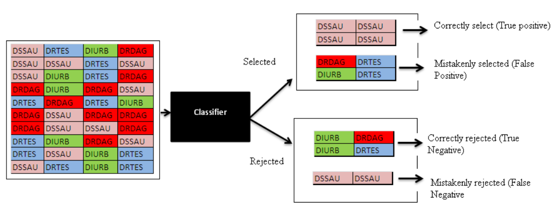

Fig. 1. Sensibility and Specificity

Therefore, specificity is defined in Equation (1) and sensitivity is defined in Equation (2).

$$Specificity = \frac{TruePositive}{TruePositive + FalseNegative} \quad (1)$$

$$Sensitivity = \frac{TrueNegative}{TrueNegative + FalsePositive} \quad (2)$$

One of the problems with analyzing just one of these measures is that in an unbalanced scenario, where one class dominates the other, the specificity might be really high, while the sensitivity might be really low. F-measure combines these measures to avoid this problem [24]. Kappa, on the other hand, is a statistic coefficient that measures a percentage of agreement beyond what is expected by chance. It considers all the confusion matrix in its calculation, including the elements outside the main diagonal, which account for the discrepancies in the classification [25]. Thus, by combining several metrics it is possible to resolve some of these questions and achieve a more consistent result.

The results were evaluated for all four algorithms using the validation set with the metrics described. Table I shows the Kappa and F-measure results for each classifier. The best result was obtained using Random Forest with a 0.84 F-measure and 0.77 Kappa. The second best result was obtained using Decision Tree with a 0.72 F-measure and 0.62 Kappa. The Naive Bayes model had a 0.70 F-measure and a Kappa of 0.60. Finally, the SVM model was significantly worse than the others with 0.33 F-measure and a Kappa of only 0.24.

TABLE I
Kappa and F-measure for each algorithm

Algorithms	Kappa	F-measure
Decision Tree	0.619	0.7208
Naive Bayes	0.6078	0.704
Random Forest	0.7679	0.8359
SVM	0.2445	0.332

Tables II, III, IV, and V show the confusion matrix for each algorithm using the validation data set. As it can be seen, the Random Forest model presents the best results and SVM the worst. In SVM confusion matrix, for DRTES class, the cell of the main diagonal is zero, meaning that there was no hit.

TABLE II
Confusion Matrix for Decision Tree

Decision Tree	DSSAU	DRTES	DIURB	DRDAG
DSSAU	133	10	13	11
DRTES	43	130	43	42
DIUNR	3	11	103	8
DRDAG	0	2	0	98

TABLE III
Confusion Matrix for SVM

SVM	DSSAU	DRTES	DIURB	DRDAG
DSSAU	125	27	16	25
DRTES	0	0	0	0
DIUNR	54	126	143	118
DRDAG	0	0	0	16

Tables VI, VII, VIII, and IX represent the sensitivity and specificity for all areas in each classifier. As we can see, SVM presented the worse result for sensitivity and specificity. The other algorithms have similar results.

[8] http://cran.r-project.org/web/packages/klaR/index.html

TABLE IV
CONFUSION MATRIX FOR NAIVE BAYES

Naive Bayes	DSSAU	DRTES	DIURB	DRDAG
DSSAU	129	15	14	13
DRTES	10	101	15	6
DIUNR	13	15	98	9
DRDAG	27	22	32	131

TABLE V
CONFUSION MATRIX FOR RANDOM FOREST

Naive Bayes	DSSAU	DRTES	DIURB	DRDAG
DSSAU	163	8	5	7
DRTES	6	118	19	11
DIUNR	8	21	130	15
DRDAG	2	6	5	126

TABLE VI
SENSITIVITY AND SPECIFICITY FOR DECISION TREE

Decision Tree	DSSAU	DRTES	DIURB	DRDAG
Sensitivity	0.7430	0.8497	0.6478	0.6164
Specificity	0.9278	0.7425	0.9552	0.9959

TABLE VII
SENSITIVITY AND SPECIFICITY FOR NAIVE BAYES

Naive Bayes	DSSAU	DRTES	DIURB	DRDAG
Sensitivity	0.7207	0.6601	0.6164	0.8239
Specificity	0.9108	0.9376	0.9246	0.8350

TABLE VIII
SENSITIVITY AND SPECIFICITY FOR SVM

SVM	DSSAU	DRTES	DIURB	DRDAG
Sensitivity	0.6983	0.000	0.8994	0.1006
Specificity	0.8556	1.000	0.3931	1.000

TABLE IX
SENSITIVITY AND SPECIFICITY FOR RANDOM FOREST

Random Forest	DSSAU	DRTES	DIURB	DRDAG
Sensitivity	0.9106	0.7712	0.8176	0.7925
Specificity	0.9575	0.9276	0.9104	0.9735

Table X compares general results for each algorithm showing that Random Forest presents the best results in all evaluated metrics (Kappa, F measure, Sensitivity, and Specificity). The comparison used the average of all classes to calculate the sensitivity and the specificity for each algorithm.

V. CONCLUSION AND FUTURE WORKS

This work is a proof of concept in order to verify the feasibility of the development of an automatic model for complaints screening. The proposed model was built using text mining. We applied four machine learning algorithms: SVM, Naive Bayes, Random Forest, and Decision Tree. They were evaluated with the following measures: Kappa, Specificity, Sensibility, and F-measure. The best of them was Random Forest with 0.84 F-measure and a 0.77 Kappa. Although we limited the scope of the work to just 4 units out of 82, the results obtained show that it is possible to implement an automatic classifier using text mining for complaints screening.

TABLE X
KAPPA, F-MEASURE, SENSITIVITY, AND SPECIFICITY FOR EACH ALGORITHM

Algorithms	Kappa	F-measure	Sensitivity	Specificity
Random Forest	0.7679	0.8359	0.823	0.94
Decision Tree	0.619	0.7366	0.714	0.90
Naive Bayes	0.6078	0.7218	0.70	0.9
SVM	0.2455	0.3610	0.424	0.812

However, since we have 82 different classes, and some of them are somewhat related, we will verify whether we apply multi-label classification instead of the single one used in this work. We would then forward only these complaints with more than one assigned class to be validated manually. This would decrease significantly the amount of screening that must be done manually, without compromising accuracy.

Moreover, we will try other alternatives for improving the performance of the model. Although the Random Forest model presented a good result (0.84 F-measure and 0.77 Kappa), it might benefit from others techniques such as Named Entity Recognition to recognized proper nouns such as location, organization, and personal name in the text [26]. These are often used in the manual screening, but had no special treatment in this work. Finally, we will also verify the difference in performance when considering different n-grams instead of just single words in our bag of words. This would account for terms that have a special meaning when considered together, like Government Programs, which are usually compound.

REFERENCES

[1] Fabrizio Sebastiani, "Machine learning in automated text categorization," 2002.

[2] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine learning*, vol. 3, no. 2, pp. 95–99, 1988. [Online]. Available: http://www.springerlink.com/index/rw3572714v41q507.pdf

[3] Azevedo, Ana Isabel Rojão Lourenço, "KDD, SEMMA and CRISP-DM: a parallel overview," 2008. [Online]. Available: http://recipp.ipp.pt/handle/10400.22/136

[4] S. Youn and D. McLeod, "A comparative study for email classification," in *Advances and Innovations in Systems, Computing Sciences and Software Engineering*. Springer, 2007, pp. 387–391. [Online]. Available: http://link.springer.com/chapter/10.1007/978-1-4020-6264-3_67

[5] D. Navadiay, M. Parikh, and R. Patel, "Constructure based web page classification," *International Journal of Computer Science and Management Research*, vol. 2, no. 6, 2013. [Online]. Available: http://www.ijcsmr.org/vol2issue6/paper424.pdf

[6] J. Garcia Adeva, J. Pikatza Atxa, M. Ubeda Carrillo, and E. Ansuategi Zengotitabengoa, "Automatic text classification to support systematic reviews in medicine," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1498–1508, Mar. 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0957417413006684

[7] A. Moschitti, "A study on optimal parameter tuning for rocchio text classifier," in *Advances in Information Retrieval*. Springer, 2003, pp. 420–435. [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-36618-0_30

[8] Federal Internal Control Secretariat, "Audit and Inspection," 2014, available: http://cgu.gov.br/english/AreaAuditoriaFiscalizacao/OQueE/.

[9] Transparency and Corruption Prevention Secretariat, “Preventing Corruption,” 2014, available: http://cgu.gov.br/english/AreaPrevencaoCorrupcao/OQueE/.

[10] National Disciplinary Board, “Disciplinary Action,” 2014, available: http://cgu.gov.br/english/AreaCorreicao/OQueE/.

[11] Ombudsman’s Office, “Ombudsman’s Office,” 2014, available: http://cgu.gov.br/english/AreaOuvidoria/OQueE/.

[12] I. Feinerer, “Introduction to the tm package text mining in r,” Tech. Rep., Jan. 2013. [Online]. Available: http://cran.r-project.org/web/packages/tm/vignettes/tm.pdf

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944937

[14] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” *The Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=944793

[15] L. Breiman, “Random forests,” Technical Report 567, Department of Statistics, UC Berkeley, 1999. 31, Tech. Rep., 2001, 11939. [Online]. Available: http://oz.berkeley.edu/~breiman/random-forests.pdf

[16] A. Prinzie and D. Van den Poel, “Random forests for multiclass classification: Random MultiNomial logit,” *Expert Systems with Applications*, vol. 34, no. 3, pp. 1721–1732, Apr. 2008. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0957417407000498

[17] E. Westerhout, “Definition extraction using linguistic and structural features,” in *Proceedings of the 1st Workshop on Definition Extraction*. Association for Computational Linguistics, 2009, pp. 61–67. [Online]. Available: http://dl.acm.org/citation.cfm?id=1859775

[18] S. Bhattacharyya, “Confidence in predictions from random tree ensembles,” *Knowledge and Information Systems*, vol. 35, no. 2, pp. 391–410, May 2013. [Online]. Available: http://link.springer.com/10.1007/s10115-012-0600-z

[19] H. F. Witschel, “Using decision trees and text mining techniques for extending taxonomies,” in *Proceedings of the Workshop on Learning and Extending Lexical Ontologies by Using Machine Learning Methods*, 2005. [Online]. Available: http://wortschatz.uni-leipzig.de/~fwitschel/papers/WS_ontologies.pdf

[20] D. D. Lewis, “Naive (bayes) at forty: The independence assumption in information retrieval.”

[21] A. Hotho, A. NÃijrnberger, and G. Paa\s s, “A brief survey of text mining.” in *Ldv Forum*, vol. 20, 2005, pp. 19–âĂŞ62. [Online]. Available: http://www.kde.cs.uni-kassel.de/hotho/pub/2005/hotho05TextMining.pdf

[22] J. O. Wobbrock and B. A. Myers, “Analyzing the input stream for character-level errors in unconstrained text entry evaluations,” *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 13, no. 4, pp. 458–489, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1188819

[23] Michael Gordon and Manfred Kochen, “Recall-precision trade-off: A derivation,” 1988.

[24] N. Lazarevic-McManus, J. Renno, and G. A. Jones, “Performance evaluation in visual surveillance using the f-measure,” in *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*. ACM, 2006, pp. 45–52. [Online]. Available: http://dl.acm.org/citation.cfm?id=1178790

[25] J. Carletta, “Assessing agreement on classification tasks: the kappa statistic,” *Computational linguistics*, vol. 22, no. 2, pp. 249–254, 1996. [Online]. Available: http://dl.acm.org/citation.cfm?id=230390

[26] C.-H. Ku and G. Leroy, “Automated crime report analysis and classification for e-government and decision support,” in *Proceedings of the 14th Annual International Conference on Digital Government Research*. ACM, 2013, pp. 1–âĂŞ27. [Online]. Available: http://dl.acm.org/citation.cfm?id=2479732