Automatic Identification of Economic Activities in Complaints

Luís Barbosa[1], João Filgueiras[1,2], Gil Rocha[1,2], Henrique Lopes Cardoso[1,2](✉), Luís Paulo Reis[1,2], João Pedro Machado[3], Ana Cristina Caldeira[3], and Ana Maria Oliveira[3]

[1] Departamento de Engenharia Informática, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal
{up201405729,gil.rocha,filgueiras,hlc,lpreis}@fe.up.pt

[2] Laboratório de Inteligência Artificial e Ciência de Computadores (LIACC), Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal

[3] Autoridade de Segurança Alimentar e Económica (ASAE), Rua Rodrigo da Fonseca, 73, 1269-274 Lisboa, Portugal
{jpmachado,accaldeira,amoliveira}@asae.pt

Abstract. In recent years, public institutions have undergone a progressive modernization process, bringing several administrative services to be provided electronically. Some institutions are responsible for analyzing citizen complaints, which come in huge numbers and are mainly provided in free-form text, demanding for some automatic way to process them, at least to some extent. In this work, we focus on the task of automatically identifying economic activities in complaints submitted to the Portuguese Economic and Food Safety Authority (ASAE), employing natural language processing (NLP) and machine learning (ML) techniques for Portuguese, which is a language with few resources. We formulate the task as several multi-class classification problems, taking into account the economic activity taxonomy used by ASAE. We employ features at the lexical, syntactic and semantic level using different ML algorithms. We report the results obtained to address this task and present a detailed analysis of the features that impact the performance of the system. Our best setting obtains an accuracy of 0.8164 using SVM. When looking at the three most probable classes according to the classifier's prediction, we report an accuracy of 0.9474.

Keywords: Text categorization · Natural language processing · User-generated text · Complaint analysis

1 Introduction

Several countries have public administration institutions that provide public services electronically. Moreover, such institutions are responsible for processing citizen requests, also performed by electronic means, often materialized through

C. Martín-Vide et al. (Eds.): SLSP 2019, LNAI 11816, pp. 249–260, 2019.
https://doi.org/10.1007/978-3-030-31372-2_21

email contacts or by filling-in contact forms in so-called virtual counters. In specific types of public institutions, such as those in charge of enforcing compliance of citizens or economic agents, a significant number of such requests are in fact complaints that need to be appropriately dealt with.

The amount of complaints received can reach the thousands in a short period of time, depending on the size of the country/administrative region. The Portuguese Economic and Food Safety Authority (ASAE), for instance, receives more than 20 thousand complaints annually, more than 30% of which are usually found not to be in the jurisdiction of ASAE; the rest are sent to the ASAE Operational Units. Given the high amount of complaints, the use of human labor to analyze and properly handle them quickly becomes a bottleneck, bringing the need to automate this process to the extent possible. One of the obstacles to do it effectively is the fact that contact forms typically include free-form text fields, bringing high variability to the quality of the content written by citizens.

This work focuses on automatically identifying economic activities in complaints written in Portuguese, through the use of natural language processing (NLP) and machine learning (ML) techniques. Portuguese is a low-resourced language in terms of NLP. We employ different features and analyze which ones give the best results using different ML algorithms. We start by discussing related work in Sect. 2. Section 3 describes the dataset used in this work. We detail the employed preprocessing and feature extraction techniques in Sect. 4. Using different ML models, Sect. 5 describes several experiments, including those related with feature selection and data balancing techniques. In Sect. 6, we provide an error analysis and make pertinent observations on the difficulty of the task. Finally, Sect. 7 concludes and presents some lines of future work.

2 Related Work

Although several works exist on analyzing user-generated content, they mostly study social media data [1], focusing on tasks such as sentiment analysis and opinion mining [15], or predicting the usefulness of product reviews [4]. Forte and Brazdil [6] focus on sentiment polarity of Portuguese comments, and use a lexicon-based approach enriched with domain specific terms, formulating specific rules for negation and amplifiers. Literature on (non-social media) complaint analysis is considerably more scarce, mainly due to the fact that such data is typically not publicly available. Nevertheless, the problem has received significant attention from the NLP community, as a recent task on consumer feedback analysis shows [11]. Given the different kinds of analysis one may want to undertake, however, the task concentrates on a single goal: to distinguish between comment, request, bug, complaint, and meaningless. In our work, we want to further analyze the contents of complaints, with a finer granularity.

Ordenes et al. [14] propose a framework for analyzing customer experience feedback, going beyond sentiment analysis and using a linguistics-based text mining model. The approach explores the identification of activities, resources and context, so as to automatically distinguish compliments from complaints, regarding different aspects of the customer feedback. This is made possible through a manual annotation process. The work focuses on a single activity domain, and in the end aims at obtaining a refined sentiment analysis model. In our case, we aim at distinguishing amongst a number of economic activities, without entering into a labor-intensive annotation process of domain-specific data.

Traditional approaches to text categorization employ feature-based sparse models, using bags-of-words and TF-IDF metrics. In the context of insurance complaint handling, Dong and Wang [17] make use of synonyms and Chi-square statistics to reduce dimensionality.

Dealing with complaints as a multi-label classification problem can be effective, even when the original problem is not, due to the noisy nature of user-generated content. Ranking algorithms [10,12] are a promising approach in this regard, providing a set of predictions sorted by confidence. These techniques have been applied in complaint analysis [5], although with modest results.

Kalyoncu et al. [9] approach customer complaint analysis from a topic modeling perspective, using techniques such as Latent Dirichlet Allocation (LDA) [2]. This work is not so much focused on automatically processing complaints, but instead on providing a visualization tool for mobile network operators.

3 Data

The dataset under study has been provided by ASAE. It contains a total of 48,850 complaints received by this governmental entity between 2014 and 2018, submitted by citizens, economic operators, public organizations or other organizations either by email or through a contact form in an official website. Each complaint contains its textual content and is classified with a single economic activity. This is the focus of this work, i.e., to train a classifier that is able to predict this activity (or a generalization thereof).

The economic activity taxonomy used by ASAE is hierarchical in nature. The first level contains 11 classes, and its imbalanced distribution is shown in Table 1. Generally, each class is composed of a number of sub-classes, which have a further decomposition level. Given the large number of second and third-level classes, we decided to train our classifiers to predict first-level classes only.

Since our goal is to aid ASAE staff in handling complaints, we have decided to base our classifications on their textual contents alone. The average complaint is $1,664$ characters long after removing HTML tags and other artifacts, containing information on its subject matter, the targeted economic agent and contact information of the claimant.

4 Data Preprocessing and Feature Extraction

We have gone through a typical preprocessing pipeline, including tokenization and lemmatization. Based on [13], we have chosen to use NLTK[1], StanfordNLP[2]

[1] https://www.nltk.org/.

[2] https://stanfordnlp.github.io/stanfordnlp/.

Table 1. Distribution per classes

Class	# examples	# 2nd level subclasses
I - Primary Production	134	7
II - Industry	2031	26
III - Restoration and beverages	20899	4
IV - Wholesalers	299	4
V - Retail	5951	23
VI - Direct selling establishments	1	1
VII - Distance selling (by Catalog and Internet)	2856	1
VIII - Production and Trade	3335	69
IX - Service Providers	9933	85
X - Safety and Environment	696	62
Z - No activity identified	2715	*N/A*

and spaCy[3]. Given the lack of conclusive data on their performance for Portuguese, the non-exhaustive experiments shown in Table 2 were performed to analyze which were better to identify the economic activity of a complaint.

StanfordNLP was chosen for most experiments, given its competitive contribution to the task and because it is able to identify punctuation marks. Additionally, StanfordNLP provides specific and complete support for Portuguese and presents the data using the CoNLL-U format [16], which increases interoperability with other tools. After obtaining the lemmas, we remove punctuation marks and stop words (using NLTK's stop word list for Portuguese) before performing TF-IDF counts. Given that we have a single example for class VI, as per Table 1, we decided to leave it out of our classification problem.

To perform feature extraction, different data representation techniques were used: count, hashing and TF-IDF, as provided by scikit-learn [3]. The count technique transforms a collection of texts into a matrix of token counts. Hashing obtains a matrix of either token counts or binary occurrences, depending if we want counts or one-hot encoding. We used it to obtain token counts and compare the difference with the count technique because it has a few advantages, like low memory scalability. TF-IDF obtains features representing the importance of each token in the collection of all documents. For these three techniques, we present results obtained by using bags-of-words of 1-grams, 2-grams, 3-grams and intervals of 1 to 2-grams, 1 to 3-grams and 2 to 3-grams.

5 Predicting Economic Activity

The classification task addressed in this paper concerns predicting the economic activity targeted in a complaint. We focus on the first level of the hierarchy, as explained in Sect. 3. In order to find out which classifiers would allow us to obtain

[3] https://spacy.io/.

the best results, we decided to use Random Forests, Bernoulli NB, Multinomial NB, Complement NB, k-Nearest Neighbors, SVM, Decision Tree, Extra Tree and Random (stratified). The latter will be used as baseline. All of them were implemented using scikit-learn[4] and the default parameters are used (for version 0.22), except those explicitly stated.

To split the original dataset into training and test set, we use 30% of the data for testing, while keeping the distribution of classes of the original dataset in both training and test sets. Following this procedure, we ensure that the trained classifier learns the real distribution of the data, and that the distribution is kept in the test set. Cross-validation was considered but given the considerable amount of training data it was deemed unnecessary to ensure consistency. This is important not only to ensure proper training but also to ensure that, when applying over/under sampling, no over/underfitting occurs in a class.

Our main performance metric was the accuracy score instead of the average macro-F1 score. We aim to provide a list of classes sorted by confidence and it is not critical to correctly classify minority classes. As a baseline we used a stratified random classifier that yielded an accuracy of 0.2504.

Table 2. Economic Activity Multiclass Classification accuracy scores using different tokenizers/lemmatizers

Classifier	StanfordNLP (baseline)	NLTK	spaCy - pt_core_news_sm	spaCy - xx_ent_wiki_sm
Random Forests	0.6787	0.6924	0.6818	0.6911
Bernoulli NB	0.5115	0.5363	0.5110	0.5185
Multinomial NB	0.4603	0.4719	0.4613	0.4648
Complement NB	0.5914	0.6263	0.5965	0.6066
K-Neighbors	0.6283	0.3146	0.6328	0.6180
SVM (linear)	**0.8075**	**0.8164**	**0.8093**	**0.8135**
Decision Tree	0.6659	0.6698	0.6669	0.6703
Extra Tree	0.5056	0.5228	0.5185	0.5166

In Table 2 we present the accuracy scores obtained using different tokenizers and lemmatizers to preprocess the text of the examples in the dataset. For this experiment, we used 1-gram TF-IDF to represent the features extracted. NLTK obtains the best scores overall, followed by spaCy and, finally, StanfordNLP. Nevertheless, we chose to continue using StanfordNLP because the performance loss is negligible and it provides PoS information, including punctuation marks. This proved useful to remove punctuation on all experiments and also experiment with removing adjectives. Furthermore, it has the advantage of having specific support for several languages, several more than the ones supported by NLTK and spaCy (although for now we are focusing on Portuguese).

[4] https://scikit-learn.org/stable/.

Table 3. Economic Activity Multiclass Classification accuracy scores using different feature extraction techniques

Classifier	Count	Hashing	TF-IDF
Random Forests	0.6958	0.6561	0.6787
Bernoulli NB	0.5115	0.4415	0.5115
Multinomial NB	0.6329	Error[a]	0.4603
Complement NB	0.6790	Error[a]	0.5914
K-Neighbors	0.5359	0.5750	0.6283
SVM (linear)	**0.7784**[b]	**0.7953**	**0.8075**
Decision Tree	0.6786	0.6671	0.6659
Extra Tree	0.4968	0.4865	0.5056

[a] Hashing may generate negative feature values, not supported by some classifiers.
[b] Failed to converge after 1,000 iterations.

Table 3 presents accuracy scores obtained using the different feature representation techniques discussed in Sect. 4. We used StanfordNLP for preprocessing and represent only 1-grams. Accuracy scores vary considerably depending on the classifier used, the best being obtained using SVM and TF-IDF. For that reason, subsequent experiments make use of TF-IDF.

Table 4. Economic Activity Multiclass Classification accuracy scores using different n-grams

Classifier	1-gram	1 to 2-grams	2-grams	1 to 3-grams	2 to 3-grams	3-grams
Random Forests	0.6737	0.6503	0.6323	0.6230	0.6127	0.5663
Bernoulli NB	0.5115	0.4763	0.4703	0.4622	0.4561	0.4495
Multinomial NB	0.4603	0.4568	0.4700	0.4568	0.4683	0.4733
Complement NB	0.5914	0.5432	0.5978	0.5381	0.5922	0.6320
K-Neighbors	0.6283	0.6152	0.5821	0.5950	0.5631	0.5413
SVM (linear)	**0.8075**	**0.8098**	**0.7640**	**0.8004**	**0.7396**	**0.6532**
Decision Tree	0.6659	0.6729	0.6121	0.6717	0.6120	0.5413
Extra Tree	0.5056	0.5338	0.5462	0.5541	0.5398	0.5298

In Table 4 we present the accuracy scores obtained using different n-grams when performing feature extraction with TF-IDF. It is not possible to conclude which is the best interval of n-grams because it depends on the classifier, but, for SVM, 1 to 2-grams is the best choice, followed by 1-gram. Because the difference between 1-grams and 1 to 2-grams in small for SVM, but higher for Random Forests, the following experiments use only 1-grams.

Taking into account the potential usage of the classifier, which is meant to help humans on analyzing complaints by providing likely classification labels

Table 5. Economic Activity Multiclass Classification accuracy scores for top-k predictions. Acc@k: accuracy scores considering the top-k (Acc@k) predicted classes, according to the confidence of the classifier predictions

Classifier	Acc@1	Acc@2	Acc@3
Random Forests	0.6787	0.8322	0.8885
Bernoulli NB	0.5115	0.7533	0.7913
Multinomial NB	0.4603	0.6790	0.7936
Complement NB	0.5914	0.8214	0.8873
K-Neighbors	0.6283	0.7699	0.8447
SVM (linear)	**0.8075**	**0.9031**	**0.9474**
Decision Tree	0.6659	0.7086	0.7226
Extra Tree	0.5056	0.5627	0.5703

(as opposed to imposing a definitive one), we looked at the performance of the classifier considering the ranking provided. In Table 5 we present accuracy scores obtained by accepting the 1st, 2nd and 3rd best probabilities. The second column shows the accuracy scores accepting as correct only the option with the highest probability. The third/fourth column shows the accuracy scores when accepting as correct one of the two/three options with the highest probabilities. For most classifiers, the accuracy of the top-2 is considerably higher than the accuracy considering the top-1. The 0.9474 score with SVM and top-3 demonstrates that presenting a set of classes sorted by confidence will be an effective help.

5.1 Feature Selection

We noticed that TF-IDF using 1-gram extracted 252,000 features, while only 101,159 are of interest when analyzing feature importance with Random Forests. As such, although a lot of features are extracted, a considerable part will be of no use to a classifier. For that reason, we explored feature selection via Latent Dirichlet Allocation (LDA), with the aim of bringing the number of features down while improving classification and training speed by clustering the features that are more important for the classification problem. However, as shown in Table 6, the use of LDA largely reduces the effectiveness of the classifiers. Moreover, although Random Forests presents an increase of 6% when raising the number of LDA components, most other classifiers maintain or even decrease accuracy scores. For this experiment, we used StanfordNLP and TF-IDF, extracting only 1-grams and analyzing top-1 predictions.

Based on these results, we concluded that performing LDA is not effective for this classification task. Applying Principal Component Analysis (PCA) [18] has led to a similar result.

Finally, we performed experiments using the Recursive Feature Elimination and Cross-Validated selection (RFECV) approach [7]. This technique consists in training a classifier multiple times with different features and yielding the

Table 6. Economic Activity Multiclass Classification accuracy scores using LDA

Classifier	No LDA (baseline)	10 components	100 components
Random Forests	0.6787	0.4053	**0.4612**
Bernoulli NB	0.5115	0.4278	0.4278
Multinomial NB	0.4603	-	0.4306
Complement NB	0.5914	0.4157	0.3561
K-Neighbors	0.6283	0.3995	0.4146
SVM (linear)	**0.8075**	**0.4484**	0.4424
Decision Tree	0.6659	0.3320	0.3525
Extra Tree	0.5056	0.3300	0.3419

feature matrix that generated the best classifier according to a chosen metric. RFECV was tested with Complement NB because it is fast to train, resulting in a classifier with significantly better accuracy. On the other hand, testing with SVM has shown that this classifier does not benefit from further optimization.

5.2 Over and Under Sampling

As shown in Table 1, the class distribution for our problem is very imbalanced. To improve the overall classification performance and, more specifically, the performance on minority classes, we explore two widely used techniques to deal with imbalanced datasets [8]: *random under sampling* and *random over sampling*.

We have chosen to use the "imblearn" Python package[5]. There were three alternatives to perform the over sampling: RandomOverSampler (ROS), SMOTE and ADASYN. ROS duplicates some of the examples of the classes, increasing the number of examples of all classes to the number of examples of the class with the highest number of examples, as indicated in the documentation of "imblearn". SMOTE generates new samples by interpolation, not distinguishing between easy and hard examples. ADASYN generates new samples by interpolation, focusing on generating samples based on the original samples which are incorrectly classified using a k-Nearest Neighbors classifier. Because we were testing several different classifiers, including a k-Nearest Neighbors classifier, we decided to use the RandomOverSampler to reduce bias in the results. For random under sampling, RandomUnderSampler (RUS) was chosen to be comparable to the RandomOverSampler. RandomUnderSampler randomly selects a subset of data for the targeted classes, reducing the number of examples of each class to the number of examples of the class with the smallest number of examples.

Table 7 presents the accuracy and average macro-F1 scores obtained by performing random over sampling and random under sampling on the dataset. For these experiments, we used StanfordNLP for preprocessing and TF-IDF to represent the features extracted. Only 1-grams were extracted and only the top-1

[5] https://imbalanced-learn.readthedocs.io/en/stable/.

was analyzed. As shown in Table 7, when performing random over sampling the accuracy scores related to Naive Bayes increased significantly and the accuracy of Random Forests also increased, but for all others it decreased. A similar situation can be observed regarding the corresponding average macro-F1 score. This is demonstrative that repeating the same data in the classes with a lower number of examples does not help distinguishing the different classes (except for Naive Bayes) and indicates that the classifiers are not predicting mostly the more frequent classes due to their amount of examples.

Table 7. Accuracy scores and average macro-F1 score using over or under sampling

Classifier	Accuracy (baseline)	Accuracy ROS	Accuracy RUS	Avg macro-F1 (baseline)	Avg macro-F1 ROS	Avg macro-F1 RUS
Random Forests	0.6787	0.7137	0.4402	0.42	0.49	0.30
Bernoulli NB	0.5115	0.6477	0.4703	0.18	0.48	0.28
Multinomial NB	0.4603	0.7299	0.5223	0.09	0.56	0.37
Complement NB	0.5914	0.7130	0.5258	0.28	0.52	0.37
K-Neighbors	0.6283	0.5456	0.3959	0.46	0.46	0.29
SVM (linear)	**0.8075**	**0.7985**	**0.5555**	**0.63**	**0.62**	**0.43**
Decision Tree	0.6659	0.6294	0.3678	0.45	0.44	0.26
Extra Tree	0.5056	0.4942	0.2074	0.33	0.32	0.15

On the other hand, when performing random under sampling, only the accuracy scores related to Bernoulli NB and Multinomial NB increased, while for all the other classifiers it has decreased significantly. All average macro-F1 score are relatively low, but 6 of them decreased and 3 of them increased. This is demonstrative that reducing the amount of examples for the classes with a higher number of examples reduces the ability of distinguishing the different classes.

5.3 Additional Experiments

An experiment performed to analyze the impact of the removal of adjectives identified by StanfordNLP was performed to identify if they were important for the classification task. This experiment was interesting because strong adjectives are apparently important for the classification task, but other weaker adjectives should not be. Depending on the amount and type of adjectives present in the dataset, their removal could reduce the amount of features that are irrelevant for the problem. Comparing the accuracy scores of all classifiers with the accuracy scores obtained by not removing the adjectives (baseline), as is the case in Table 5, the percentage was always the same, differing only on the permillage. These results are indicative that adjectives are partially important for the classifiers, although most of them have a low or even null importance/coefficient.

Experiments performed to increase the accuracy of SVM (with linear kernel) generating different class weights and balanced class weights (hyperparameterization) [8] obtained accuracy and average macro-F1 scores close to the ones obtained using the default parameters: a maximum accuracy of 0.8096 with a macro-F1 score of 0.63. Also, the different kernels available for SVM (linear, poly, rbf, sigmoid, precomputed) were tested and it was found that the linear kernel is the best in terms of accuracy, immediately followed by the sigmoid kernel, and that the sigmoid kernel is the best in terms of average macro-F1 score, immediately followed by the linear kernel. Finally, experiments performed to test the use of ensembles based on decision trees, which usually have interesting performances, provided accuracy scores higher than the ones obtained using Random Forests, but considerably lower than the ones provided by SVM.

6 Error Analysis

Based on the different accuracy and average macro-F1 scores obtained, we decided to focus on SVM for the sake of error analysis. We show the obtained confusion matrix in Table 8, when considering top-1 classification only. The influence of the majority class III is visible, but also of the second majority class IX. Class Z, where there is no identified economic activity, seems to be the most ambiguous for the classifier.

Table 8. Confusion matrix of the baseline SVM (Top-1)

		Predicted									
		I	II	III	IV	V	VII	VIII	IX	X	Z
Actual	I	14	5	10	1	5	0	1	0	0	4
	II	1	324	155	2	61	2	8	26	0	30
	III	0	37	5935	1	72	7	30	160	2	24
	IV	0	9	16	22	22	0	3	7	0	11
	V	1	26	184	3	1454	16	32	42	1	26
	VII	0	0	16	0	7	722	26	62	1	23
	VIII	1	18	126	1	61	31	596	114	6	46
	IX	0	5	314	0	26	30	83	2479	10	33
	X	0	0	17	1	6	8	31	52	81	12
	Z	2	35	181	3	72	55	93	163	6	204

To better understand in which situations the classifier was making erroneous predictions, we randomly sampled 50 examples from the dataset where the classifier was not capable of correctly predicting (from the top-3 predictions) the gold-standard class. Based on a manual analysis of such cases, we were able to draw the following observations:

- The dataset includes some short text complaints, not providing enough information to classify their target economic activity. Furthermore, a small number of complaints are not written in Portuguese. Some complaint texts are followed by non complaint-related content, sometimes in English.[6]

[6] Complaints received by e-mail often include "think twice before printing" appeals.

- Some classes exhibit semantic overlap (to a certain degree), thus confusing the classifier. For example, class VIII apparently overlaps with classes II and V. Moreover, while being labeled with a given class, some complaints contain words that are highly related with a different class.
- A non-negligible number of examples refer to previously submitted complaints, either to provide more data or to request information on their status. These cases do not contain the complaint itself, the same happening when a short text simply includes meta-data or points to an attached file.
- Finally, we were able to identify some complaints that have been misclassified by the human operator.

7 Conclusions and Future Work

For the imbalanced complaints dataset of ASAE, SVM with a linear kernel proved to be the best option among the experimented models. It is reasonably fast, allows to get probability scores and gives the best accuracy scores and average macro-F1. It is particularly valuable if we need a ranked output, given its high accuracy when aggregating the top-3 predicted classes. It is interesting to note that removing punctuation and stop words after lemmatization, using TF-IDF and training the SVM generates better accuracy scores than using additional techniques like feature selection and different quantities of n-grams.

After analyzing misclassified examples, several improvements have been planned. Non-Portuguese complaints need to be ignored, as the number of examples is too low to warrant a multilingual classifier. Furthermore, we aim to further assess how to discard texts that are simply not informative enough to consider as valid complaints (besides empty complaints, which the system correctly classifies). We also aim to tackle additional classification problems exploring this rich dataset. The ideas presented in this work will be the baseline for these future classifiers. We intend to explore recent advances on word embeddings approaches and deep learning techniques, and compare the results obtained with the models presented in this paper. The end goal is to create a system that will greatly assist ASAE personnel when handling these complaints.

Acknowledgements. This work is supported by project IA.SAE, funded by Fundação para a Ciência e a Tecnologia (FCT) through program INCoDe.2030. Gil Rocha is supported by a PhD studentship (with reference SFRH/BD/140125/2018) from FCT.

References

1. Batrinca, B., Treleaven, P.C.: Social media analytics: a survey of techniques, tools and platforms. AI Soc. **30**(1), 89–116 (2015)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)

3. Buitinck, L., et al.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108–122 (2013)
4. Diaz, G.O., Ng, V.: Modeling and prediction of online product review helpfulness: a survey. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 698–708 (2018)
5. Fauzan, A., Khodra, M.L.: Automatic multilabel categorization using learning to rank framework for complaint text on bandung government. In: 2014 International Conference of Advanced Informatics: Concept, Theory and Application (ICAICTA), pp. 28–33. Institut Teknologi Bandung, IEEE (2014)
6. Forte, A.C., Brazdil, P.B.: Determining the level of client's dissatisfaction from their commentaries. In: Silva, J., Ribeiro, R., Quaresma, P., Adami, A., Branco, A. (eds.) PROPOR 2016. LNCS (LNAI), vol. 9727, pp. 74–85. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41552-9_7
7. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. **46**(1), 389–422 (2002)
8. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009). https://doi.org/10.1109/TKDE.2008.239
9. Kalyoncu, F., Zeydan, E., Yigit, I.O., Yildirim, A.: A customer complaint analysis tool for mobile network operators. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 609–612. IEEE (2018)
10. Li, H.: Learning to rank for information retrieval and natural language processing. In: Synthesis Lectures on Human Language Technologies, 2nd edn. Morgan & Claypool Publ., San Rafael (2014)
11. Liu, C.H., Moriya, Y., Poncelas, A., Groves, D.: IJCNLP-2017 task 4: customer feedback analysis. In: Proceedings of the IJCNLP 2017, Shared Tasks. Asian Federation of Natural Language Processing, Taipei, Taiwan, pp. 26–33, December 2017
12. Momeni, E., Cardie, C., Diakopoulos, N.: A survey on assessment and ranking methodologies for user-generated content on the web. ACM Comput. Surv. **48**(3), 41:1–41:49 (2015)
13. Omran, F.N.A.A., Treude, C.: Choosing an NLP library for analyzing software documentation: a systematic literature review and a series of experiments. In: Proceedings of the 14th International Conference on Mining Software Repositories, pp. 187–197. MSR 2017. IEEE Press, Piscataway (2017)
14. Ordenes, F.V., Theodoulidis, B., Burton, J., Gruber, T., Zaki, M.: Analyzing customer experience feedback using text mining: a linguistics-based approach. J. Serv. Res. **17**(3), 278–295 (2014)
15. Petz, G., Karpowicz, M., Fürschuß, H., Auinger, A., Stříteský, V., Holzinger, A.: Opinion mining on the web 2.0 – characteristics of user generated content and their impacts. In: Holzinger, A., Pasi, G. (eds.) HCI-KDD 2013. LNCS, vol. 7947, pp. 35–46. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39146-0_4
16. Qi, P., Dozat, T., Zhang, Y., Manning, C.D.: Universal dependency parsing from scratch. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 160–170. Association for Computational Linguistics, Brussels, Belgium, October 2018
17. Dong, S., Wang, Z.: Evaluating service quality in insurance customer complaint handling throught text categorization. In: 2015 International Conference on Logistics, Informatics and Service Sciences (LISS), pp. 1–5. IEEE (2015)
18. Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. J. Roy. Stat. Soc. B **61**(3), 611–622 (1999)