Anomaly Detection in Text Data Sets using Character-Level Representation

Anomaly Detection in Text Data Sets using Character-Level Representation

Mahsa Mohaghegh[1, a, *] and Amantay Abdurakhmanov[1, b]

[1] Auckland University of Technology, Auckland, New Zealand

[a, *] mahsa.mohaghegh@aut.ac.nz, [b] amantay.abdurakhmanov@gmail.com

Abstract. This paper proposes a character-level representation of unsupervised text data sets for anomaly detection problems. An empirical examination of the character-level text representation was conducted to demonstrate the ability to separate outlying and normal records using an ensemble of multiple classic numerical anomaly classifiers. Experimental results obtained on two different data sets confirmed the applicability of the developed unsupervised model to detect outlying instances in various real-world scenarios, providing the opportunity to quickly assess a large amount of textual data in terms of information consistency and conformity without knowledge of the data content itself.

1. Introduction

Anomaly (outlier) detection is a classic problem in a variety of application domains where the necessity of determining outlying data is often crucial. The origin of the problem stems from anomaly interpretability as important, and sometimes even actionable information [1]. To extract such information, an anomaly detection model should be able to distinguish data set instances that are dissimilar to all others in a data-driven manner [2]. Some of the application fields where anomaly detection is widely used to date are: information security, e.g. network intrusion [3], identity theft [4]; e-commerce, e.g. credit-card fraud [5], price misplacing [6]; and medicine, e.g. medical devices monitoring [7].

From a business perspective, the anomaly detection problem relates to data quality business processes that operate with information about clients, products, contracts, etc. Data quality is becoming a vital business priority as an increasing number of companies aim to make their business models more digital [8]. Supervised learning generally requires a massive amount of manually labelled anomaly examples for sufficient training. However, this approach is often unsuitable in real-world business cases, especially if the consistency of data quality needs to be quickly checked. Although the unsupervised approach is usually less accurate and often less interpretable, it can be used as a first attempt towards understanding the data composition. From an interpretation perspective, text anomalies could be examined by the data owner or domain expert in terms of their nature. A typical example is a process of estimating customer data quality. Plausible anomaly types and their interpretations may include:

- Unintended typos: predominantly these would be made by a client employee during a process of customer registration (e.g. doubling some characters or confusing closely located keys on a keyboard). Typos made by customers themselves are relatively rare.

- Fraud: instances of fraud anomalies usually come by customers during self-registration (e.g. online) to hide information relating to their real identity.
- Intended modification: these anomalies may be caused by a problem with the registration process due to software defects and/or a procedural issue. For example, an employee unable to register a real name because the customer is already registered, but the employee does not have enough system privileges to use or modify it.

In this work, we aimed to develop a fully data-driven model to analyse the large textual data set with client personal information provided by one New Zealand company for scientific purposes as an example of contaminated client personal data, which can typically be found in real-world business cases. The created anomaly detection model was examined on the short message service (SMS) data set of "spam and ham" messages introduced by Almeida *et al.* [9]. We developed a character-level representation of text as numerical features for our detection model. This approach appeared to be flexible enough for business purposes and allowed us to use it on a wide variety of unlabelled textual data sets without prior model's training and tuning. We implemented a set of distinct anomaly detection classifiers able to process relatively large data sets (from hundreds of thousands to millions of records). Anomaly scores calculated by these algorithms were normalised and combined into one robust ensemble score. Finally, the ensemble score was used to order the data set from most irregular to most normal.

2. Related work

Anomaly detection is well covered in literature, including numerous books, articles, and surveys. Aggarwal [1] extensively studied multiple approaches and algorithms; Chandola et al. [2] have reviewed existing techniques; and Kwon et al. [3] conducted a survey with specific focus on modern deep learning principles. Anomaly detection might be considered as a subset of classification, as anomalies and normal instances could be considered as two special classes.

Regarding textual data, existing representation methods for text classification can be divided into three categories. The first category is classic bag-of-words (BOW) [10], n-grams and their term-frequency/inverse-document-frequency (TF-IDF) [11]. The crucial flaw of these methods is that they omit the words (n-grams) order and the context. The second category is word embeddings, which has become popular with neural networks [12] and has evolved into the actual state-of-the-art word2vec toolkit that is able to create a numerical vector for each word according to its context [13]. The third category is the recently developed character-level representation as a sequence of encoded characters as input for the models [14]. Researchers examined this with char-level convolutional neural networks (CCNN) and generative adversarial networks (GAN) methods and demonstrated its ability to capture text semantics [14-16]. Their CCNN models use backward quantization character order, therefore latest reading on characters is always placed near the beginning of the output, to make it easy for fully-connected layers to associate weights with the latest reading [14]. Authors limited their alphabet with 70 characters (26 English letters, 10 digits, 33 other characters and the new line character).

In comparison with neural networks, various other techniques can be considered in cases with time, simplicity, and relatively small available data constraints. Goldstein and Uchida [17] compared many algorithms particularly for unsupervised practical cases for their applicability in several scenarios, while Pevný [18] compared such algorithms by their time and space complexity. These researches identified unsupervised anomaly detectors that perform best in large high-dimensional data due to their computational efficiency:

- Principal Component Analysis (PCA) calculates the sum of projected distances of an instance on all eigenvectors [19].
- Histogram-based Outlier Detection (HBOS) calculates the sum of corresponding heights of the bins of created histograms for each feature [20].
- Isolation Forest (IForest) calculates the average number of splittings required to isolate instances for a randomly selected feature [21, 22].

- Cluster-Based Local Outlier Factor (CBLOF) considers the size of clusters that an instance belongs to and the distance to the nearest large cluster [23].
- Lightweight On-line Detector of Anomalies (LODA) considers the average of logarithms of probabilities, i.e. proportional to negative log-likelihood of an instance [18].

Another fast non-parametric Copula-Based Outlier Detection (COPOD) algorithm that estimates tail probabilities using empirical copula [24] has been introduced recently.

Ensemble analysis is another aspect that has recently been studied with respect to anomaly detection. Considering bias-variance trade-off Aggarwal and Sathe [25] proposed robust feature bagging methods: average-of-maximum (AOM) and maximum-of-average (MOA). These methods imply the empirical fact that different detectors might perform differently from one dataset to another, and even from instance to instance within one data set. To calculate AOM for an ensemble of components, authors divided them to several buckets and used maximization over each bucket, and then calculated scores averaged over all buckets.

3. Character-level anomaly detection ensemble model

In this section, we introduce the character-level ensemble of detectors for anomaly detection in textual data. The model was developed using Python and then incorporated in the open-source data quality framework MobyDQ [26] as an individual indicator type, which can be applied on text columns extracted from a target database.

3.1. Character representation

Our model converts input textual instances to a sequence of encoded characters. Character quantization was inspired by Zhang *et al.* [14], although we made several significant changes. Our encoding uses the predefined alphabet of size *m* for the input language and then encodes each character using indexes from the randomly shuffled alphabet. The model converts a sequence of characters to a vector of corresponding scaled alphabet indexes with length equal to the average maximal length of initial text features (each feature's maximal length limited to 95% quantile interval). Any character beyond this length is ignored. Any characters that are not presented in the alphabet are quantized as 0 values. As a result, the model converts character input to a sparse numerical high-dimensional output suitable to anomaly classifiers. The predefined alphabet used by our model comprises 84 characters, including 26 English letters, 10 digits, and 48 other characters as illustrated in Figure 1.

```
. !?:,\'%-()\/$|&;[]{}"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Figure 1. The predefined alphabet

To be applicable to proximity-based algorithms, which are based on measuring distances, an index output needs to be scaled. We used MaxAbs scaler in order to preserve the original data sparsity and prevent data from skewness reduction.

3.2. Design of the model

Our model consists of a combination set of distinct anomaly classifiers, which are known to be effective on high-dimensional sparse data obtained as a result of a character quantization, namely PCA, COPOD, HBOS, LODA, CBLOF, and IForest with different hyperparameters (15 in total). These algorithms appeared to be computationally feasible even on large data sets with hundreds of thousands of samples up to several hundreds of characters long. We used the sklearn [27] implementation for PCA, IForest and the pyod [28] implementation for COPOD, HBOS, LODA, CBLOF. Hyperparameters were chosen empirically to be adequate in capturing various anomaly patterns and provide a diverse set of outcomes in a short time. The set of detectors and used hyperparameters are shown in Table 1.

Table 1. Anomaly detectors used in the model

Detector	Hyperparameters
PCA	solver = 'randomized', 'full'
COPOD	-
HBOS	bins = 10, 200, 300
LODA	random cuts = 100, 200, 300
IForest	{number of estimators, bootstrap} = {100, False}, {200, False}, {150, True}
CBLOF	clusters = 16, 24, 32

The model standardises the set of calculated anomaly scores to the interval between 0 and 1 and then uses the combo [29] implementation for dynamic AOM and MOA ensemble methods with four buckets to obtain overall anomaly score for each record.

4. Experimental results

Since our primary data set contains personal and commercially sensitive information such as customer names and addresses, we cannot report achieved anomaly detection results directly. To demonstrate the ability of the model to reveal abnormal instances, two artificial records (one normal and one anomaly) were added to the original data set. The calculated score for the normal instance is localised inside the most populated 95% interval, whereas the score for the anomaly record appears to be within the 5% tail with the highest AOM and MOA scores. In other words, the model can unveil hidden information patterns and concentrate on inaccurate samples in a shallow distribution interval. Histograms of the distributions of AOM and MOA scores are shown in Figure 2 along with lines corresponding to artificially added records.

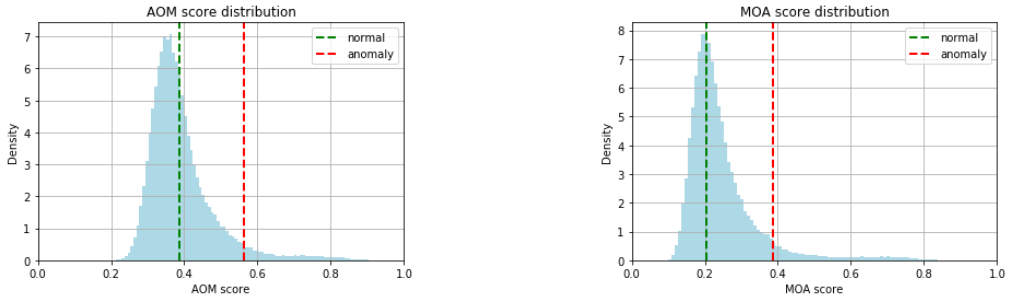

Figure 2. Anomaly score distributions for the private data set

In addition to the commercial data, we applied our model on a freely available collection of SMS messages consisting of 747 spam instances in 5574 messages total [9]. Histograms of the distributions of AOM and MOA score with corresponding ROC curve are provided in Figure 3.

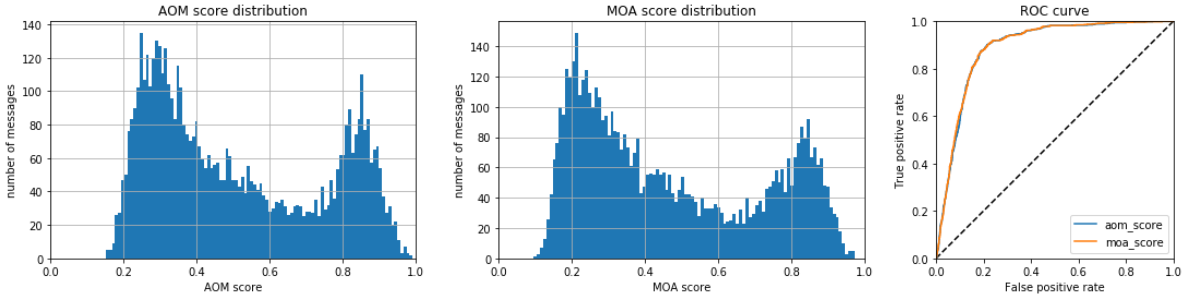

Figure 3. Anomaly score distributions and ROC curve for SMS data set

Histograms peaks correctly suggest the presence of two imbalanced classes with one (non-spam) concentrated near smaller score values and another (spam) with larger values. To evaluate our model, we used the area under the ROC curve (AUC). Reported AUC for the SMS data set was able to achieve 87%, which can be considered as a valuable initial result given the unsupervised approach with no specific tuning for a certain data set.

5. Conclusions and future work

In this work, we offer empirical research on a character-level anomaly detection ensemble model based on several computationally efficient base classifiers. We tested our approach on two unrelated data sets to demonstrate the ability of our model to meet the requirements frequently faced by business users in a real-world environment: direct evaluation of textual data consistency with no prior model tuning and training. Anomaly scores calculated by our model allow business users to rapidly explore textual data sets to see what patterns appeared to be unusual and inconsistent. As a result, they only need to observe and analyse a relatively small proportion of samples with highest anomaly scores to identify the majority of possible erroneous text patterns and then implement these patterns in rule-based indicators for future “business as usual” compliance checks.

For future work, we intend to explore more base algorithms, that are applicable as components to our ensemble approach and adapt our model to data sets with mixed-type content, e.g. combinations of numerical and textual features.

References

[1] C. C. Aggarwal, in *Outlier Analysis*: Springer International Publishing, 2017.
[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, Article vol. 41, no. 3, pp. 15.1-15.58, 2009, doi: 10.1145/1541880.1541882.
[3] D. Kwon, J. Kim, S. C. Suh, H. Kim, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, Article vol. 22, pp. 949-961, 01 / 16 / 2019, doi: 10.1007/s10586-017-1117-8.
[4] S. Garg, K. Kaur, J. J. P. C. Rodrigues, and J. J. P. C. Rodrigues, "Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective," (in English), *IEEE Transactions on Multimedia*, Article vol. 21, no. 3, pp. 566-578, 03 / 01 / 2019, doi: 10.1109/TMM.2019.2893549.
[5] F. Carcillo, Y. A. Le Borgne, G. Bontempi, O. Caelen, Y. Kessaci, and F. Oblé, "Combining unsupervised and supervised learning in credit card fraud detection," (in English), *Information Sciences*, Article 01 / 01 / 2019, doi: 10.1016/j.ins.2019.05.042.
[6] J. Ramakrishnan, E. Shaabani, C. Li, and M. A. Sustik, "Anomaly detection for an e-commerce pricing system," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1917-1926, doi: 10.1145/3292500.3330748.
[7] M. Zhang, A. Raghunathan, and N. K. Jha, "MedMon: Securing medical devices through wireless monitoring and anomaly detection," *IEEE Transactions on Biomedical circuits and Systems*, vol. 7, no. 6, pp. 871-881, 2013.
[8] M. Chien and A. Jain, "Magic Quadrant for Data Quality Tools," ed: Gartner, 2019.
[9] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: new collection and results," in *Proceedings of the 11th ACM symposium on Document engineering*, 2011, pp. 259-262, doi: 10.1145/2034691.2034742.
[10] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146-162, 1954.
[11] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, 1972.
[12] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137-1155, 2003.
[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[14] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, pp. 649-657, 2015. [Online]. Available: http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classifica.
[15] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-Aware Neural Language Models," presented at the AAAI Conference on Artificial Intelligence; Thirtieth AAAI Conference on Artificial Intelligence, 2016. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12489/12017.
[16] T. Wang, L. Liu, H. Zhang, L. Zhang, and X. Chen, "Joint Character-Level Convolutional and Generative Adversarial Networks for Text Classification," *Complexity*, Article vol. 2020, 2020, Art no. 8516216, doi: 10.1155/2020/8516216.
[17] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PloS one*, vol. 11, no. 4, p. e0152173, 2016, doi: 10.1371/journal.pone.0152173.
[18] T. Pevný, "Loda: Lightweight on-line detector of anomalies," *Machine Learning*, vol. 102, no. 2, pp. 275-304, 2016, doi: 10.1007/s10994-015-5521-0.
[19] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," University of Miami Coral Gables, FL, USA, 2003.
[20] M. Goldstein and A. Dengel, "Histogram-based Outlier Score (HBOS): A fast unsupervised anomaly detection algorithm," *KI-2012: Poster and Demo Track*, pp. 59-63, 2012.
[21] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008: IEEE, pp. 413-422.
[22] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1-39, 2012.
[23] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1641-1650, 2003, doi: 10.1016/S0167-8655(03)00003-5.
[24] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "COPOD: copula-based outlier detection," presented at the Conference: IEEE International Conference on Data Mining (ICDM), 2020. [Online]. Available: https://www.researchgate.net/publication/344306968_COPOD_Copula-Based_Outlier_Detection.
[25] C. C. Aggarwal and S. Sathe, "Theoretical Foundations and Algorithms for Outlier Ensembles," *SIGKDD Explor. Newsl.*, vol. 17, no. 1, pp. 24–47, 2015, doi: 10.1145/2830544.2830549.
[26] A. Rolland. "MobyDQ." https://github.com/ubisoft/mobydq (accessed 2020).
[27] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825-2830, 2011.
[28] Z. Yue, Z. Nasrullah, and L. Zheng, "PyOD: A Python Toolbox for Scalable Outlier Detection," *Journal of Machine Learning Research*, Article vol. 20, no. 85-96, pp. 1-7, 2019. [Online]. Available: http://www.jmlr.org/papers/volume20/19-011/19-011.pdf.
[29] Y. Zhao, X. Wang, C. Cheng, and X. Ding, "Combining machine learning models using combo library," in *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2019.