Exploring Online Novelty Detection Using First Story Detection Models

Fei Wang[1,2](✉), Robert J. Ross[1,2], and John D. Kelleher[1,2]

[1] School of Computing, Dublin Institute of Technology, Dublin, Ireland
d13122837@mydit.ie, {robert.ross,john.d.kelleher}@dit.ie
[2] ADAPT Research Centre, Dublin, Ireland

Abstract. Online novelty detection is an important technology in understanding and exploiting streaming data. One application of online novelty detection is First Story Detection (FSD) which attempts to find the very first story about a new topic, e.g. the first news report discussing the "Beast from the East" hitting Ireland. Although hundreds of FSD models have been developed, the vast majority of these only aim at improving the performance of the detection for some specific dataset, and very few focus on the insight of novelty itself. We believe that online novelty detection, framed as an unsupervised learning problem, always requires a clear definition of novelty. Indeed, we argue the definition of novelty is the key issue in designing a good detection model. Within the context of FSD, we first categorise online novelty detection models into three main categories, based on different definitions of novelty scores, and then compare the performances of these model categories in different features spaces. Our experimental results show that the challenge of FSD varies across novelty scores (and corresponding model categories); and, furthermore, that the detection of novelty in the very popular Word2Vec feature space is more difficult than in a normal frequency-based feature space because of a loss of word specificity.

Keywords: Online novelty detection · First Story Detection (FSD)
Unsupervised learning · Novelty score · Feature space · Word2Vec

1 Introduction

Novelty detection, also known as anomaly detection or outlier detection, is the task of identifying data that are different in some respect from other predominant chunks of data in a dataset [10]. Novelty is the property of abnormal data that usually indicates a defect (industry), a fraud (business), or a new topic in texts (media). In many cases, no definition is given for novelty before detection, and the abnormal data embedded in a large amount of normal data are not sufficient to build a class for novelty. Novelty detection is thus best conceptualized as an unsupervised machine learning application, i.e., no labels are available and detections can only be based on the intrinsic properties of the data.

H. Yin et al. (Eds.): IDEAL 2018, LNCS 11314, pp. 107–116, 2018.
https://doi.org/10.1007/978-3-030-03493-1_12

In the context of streaming data, there are two additional constraints on the design of a novelty detection model that do not generally apply in the classical case [4]: (1) the model can only use the data that has arrived before detection is performed, and (2) the detection must be made within a bounded (short) time. If these two requirements are satisfied, this type of novelty detection is called online novelty detection. When online novelty detection is applied in the context of First Story Detection (FSD), the detection is targeted at finding the very first story about each new topic, e.g. the first news report discussing the "Beast from the East" hitting Ireland. Hundreds of online novelty detection models have been applied to FSD, however, the vast majority of research aims only at improving the performance of the detection for some specific dataset, and very few of them focus on the insight of novelty itself. A general categorisation of novelty detection models was previously proposed by Pimentel [10] which includes: probabilistic-based, distance-based, reconstruction-based, domain-based, and information-theoretic models. However, this categorisation is established solely on the techniques used in the detection, and therefore does not naturally provide comparisons across the categories, nor insights into how the different categories of models define the concept of novelty.

We believe that online novelty detection, as a typical unsupervised learning application, always requires a transparent definition of novelty. Moreover, the clear exposition of this definition is the key issue in designing a good detection model and enhancing its performance. Within the context of FSD, in this paper we identify three main categories of online novelty detection models as motivated by previous research. We will see that each of these three categories is based on different definitions of novelty scores. Across these categories, we present an analysis of how each category treats the concept of novelty, and furthermore, we compare the performances of these model categories in different feature spaces. We proceed by providing further detail on the problem of First Story Detection.

2 Background

The challenge FSD was initially defined as a task within the Topic Detection and Tracking (TDT) competition series. In TDT, the definition of FSD is given specifically as "the task of identifying new events in a stream of stories. Each story is processed in sequence, and a decision is made whether or not a new event is discussed in the story, after processing the story but before processing any subsequent stories" [1]. In this context, a "topic" is "a seminal event or activity, along with all directly related events and activities" [3].

In TDT, a number of online novelty detection models were developed for FSD. Of these, the UMass model [2] and CMU model [15] achieved the best performance. The UMass model was based on a nearest neighbour approach and the CMU model employed two types of clustering models. Petrovic et al. [8] proposed an extension to the UMass nearest neighbor model that integrates Locality Sensitive Hashing (LSH) when seeking the nearest neighbour. The primary contribution of this extension was to improve the computational efficiency of the model. However, Petrovic et al.'s work is also noteworthy because it was the first FSD work applied to Twitter data, and since then Twitter has become a popular application domain for FSD research. In 2012 Petrovic [9] published the first Twitter dataset for FSD, the Edinburgh Twitter corpus. In recent years, more and more FSD models have been proposed, such as k-term hashing [14] in which the new data is compared to a look-up table to generate the novelty score, and the Nuggest-based method [11] in which only the top-k events need to be preserved as the comparison targets.

3 Three Categories of FSD Models

As claimed in our previous research [13], we believe that unsupervised learning always requires an appropriate definition of the learning target. Similarly, novelty detection always requires a definition of novelty. However, a reviewing of the literature on FSD research makes it apparent that every FSD model has its own way to calculate a novelty score, which can be considered as the definition of novelty used by that model. In other words, every FSD model aims to find the novelty represented by a specific novelty score.

In order to frame definitions of novelty in FSD, we propose (based on our analysis of FSD literature) three categories of novelty scores, and, three corresponding categories of FSD models, these are: Point-to-Point (P2P) models, Point-to-Cluster (P2C) models, and Point-to-All (P2A) models. This categorisation is based on different distances used to define novelty scores in different models. The concept of distance we are using here is a general expression that refers to the difference or dissimilarity between two objects, which can be two points, a point and a cluster, or a point and all other data. The three categories of models are detailed as follows:

Point-to-Point (P2P) models, in which the novelty score is defined as the distance from the new data to an existing data point:

$$Novelty\_Score_{P2P} \overset{\text{def}}{=} distance(data_{new},\ data_{existing}) \tag{1}$$

The nearest neighbour based model is a typical P2P FSD model, in which the novelty score is defined as the distance from the new data to the closest existing data point to it. In order to improve efficiency, P2P models usually accept an approximate nearest neighbour to each incoming data point, instead of the true nearest neighbour. For example, the UMass model [2] only seeks the nearest neighbour from the data points that have at least a single word in common with the new data.

Point-to-Cluster (P2C) models, in which the novelty score is defined as the distance from the new data to a cluster of existing data:

$$Novelty\_Score_{P2C} \overset{\text{def}}{=} distance(data_{new},\ cluster_{existing}) \tag{2}$$

The distance in defining the novelty score in P2C models is between a data point and a sub-space (or the union of sub-spaces) formed by a cluster of existing data points in the feature space. In the calculation, it could be the distance to a representative of the space, the distance to the range of the sub-space, or even the distance to a model trained by the cluster of data points in the sub-space. In the context of FSD, a cluster can be intuitively understood as a topic behind the texts. To make it simple, the cluster is usually represented by some data point within its range, e.g. the centroid of the cluster, the furthest point or the closest point to the new data point. The CMU model [15] that is based on single-pass clustering is a typical P2C model based on the distance from the new data to the centroid of the closest cluster.

Point-to-All (P2A) models, in which the novelty score is defined as the distance from the new data to all the existing data:

$$Novelty\_Score_{P2A} \overset{\text{def}}{=} distance(data_{new},\ all\_data_{existing}) \tag{3}$$

Given all existing data, the detection of novelty can be considered as a One-Class Classification (OCC) problem, in which the quantity of existing normal data is large enough to build the "normality", but the quantity of abnormal data is insufficient to build the novelty class for classification. One-Class SVM [12] is a popular model for OCC, the basic idea of which is to generate a hyper-sphere based on all existing data, and all the data points outside the hyper-sphere are considered as novel data.

It is worth highlighting that any novelty detection model that is based on a model trained on all the existing data can be viewed as a P2A model. For example, the k-term hashing model [14] compares all the terms of the new data with a look-up table created using all the existing data, and takes the proportion of new terms as the novelty score, and so it is a P2A model. The same goes for the probabilistic-based and reconstruction-based models. On the other hand, if the novelty score is defined by calculating the distance to subsets of data or using models built by subsets of data, then the model is a P2C model. For example, we can use a different distance from the incoming data point to each existing cluster in a P2C model by calculating the distance to the hyper-sphere around all the data of each cluster. Even though this model uses the same technique as a P2A model, this is still a P2C model.

Based on the descriptions given above, we can see that the differences between these three categories of models are actually dependent on the target object from which the distance of new data is defined, rather than what domain theories and/or model architectures are used. Using these three categories of models, we can analyze not only the performance of a single detection model, but also the common characteristics of models within a category, and furthermore, do cross category comparisons based on these general characteristics. As a practical example of this we can compare the performances of novelty detection in different feature spaces, and obtain deeper insights into both novelty and the appropriateness of feature spaces.

4 Experimental Design

We use the above categorization of FSD models to design our experiments. Our first experiment, compares the performance of three representative instances of the model categories. The second experiment is designed to examine the performance of different categories of models across different feature spaces.

4.1 Dataset

In our experiment we use a standard FSD benchmark dataset known as TDT5. The dataset is composed of news reports from a number of news agencies during the period from April to September of the year 2003. Compared with the Twitter FSD dataset, the contents of TDT5 dataset are all in plain text and standard English, and contain very few special expressions. Our research is to explore novelty detection across different types of models and feature spaces, rather than improve the performance of a specific model on a particular domain, so we believe the TDT5 is the better option for this work, as it contains more standard English, and hence model performance is not as affected by a model's ability to use features particular to a specific domain. For our experiments, we used the first 20,000 stories in the TDT5 dataset. Within this TDT5 subset there are 18 labeled topics (and hence 18 first stories), with a further 256 stories labeled as belonging to one of these 18 topics. The remaining stories are unlabeled.

4.2 FSD Performance Across Different Categories of Models

In the first experiment, we select a typical model for each category of model classes outlined in Sect. 3, and apply them to FSD on the TDT5 subset. For this experiment, all the text data was prepared by stemming each word into its root and removing stopwords and the words with frequency less than 3. After that, all the data was mapped into the tf-idf feature space, so the detections and analysis for this experiment are carried out within that feature space.

For P2P, we use a nearest neighbor model as the representative model. The incoming story is compared to all the existing stories to find the nearest neighbour. If the distance to the nearest neighbour exceeds a threshold, the story is declared novel. In order to improve efficiency, our implementation adopts some ideas from previous successful models - the UMass [2] model and the LSH [8] model that are different from a vanilla nearest neighbour model. Firstly, cosine distance is used to calculate the distance between two data points. Secondly, the model only seeks the nearest neighbour from the data points that have at least a single word in common with the new data. This is done by initially establishing an inverted index, and then continually updating the index as the detection process is ongoing. Thirdly, in order to improve computational efficiency only the 2,000 most recent data points obtained from the inverted index are considered when searching for the nearest neighbor.

The P2C category is represented by a single-pass (or called follow-the-leader) clustering model. In this model, we group the incoming stories into clusters.

As discussed in the introduction of P2C, the clusters represent topics behind the texts, and each cluster can be represented by its centroid, which is calculated as the mean of the vector representations of the stories in that cluster. Similarly, the incoming story is compared to the centroids of all the clusters to find the nearest cluster. If the distance to the nearest cluster does not exceed a threshold, the story is declared non-novel and assigned to the cluster after which the cluster centroid is updated; otherwise, we declare the story novel, and create a new cluster with this new story as the only data point within the cluster. Cosine distance is adopted in calculating the distance from a new data to a centroid.

For P2A, a One-Class SVM is adopted as the representative model because using this model it is easy to interpret the distance from a data point to all the existing data points. Given a parameter V between 0 and 1, all data are mapped into a hyper-space using a kernel function to generate a sphere that contains 1-V of the data inside it as normal data and V of the data outside it as novel data. In our model, we do not take the label by One-Class SVM as the label of novelty for a new data, but take the distance of the data to the sphere as the novelty score, with a positive value if the data is outside the sphere and negative value if the data inside it. Based on the results of validation tests, we select 0.1 as the value of V and use only the most frequent 300 features as the representations of stories in this part of experiment. Finally, to reduce the computational cost associated with repeatedly rebuilding a One-Class SVM, for each new query we also only use the 2000 most recent data points to build the model.

4.3 FSD Performance in Different Feature Spaces

The first experiment was conducted using tf-idf based representations of documents. For this second experiment, for all model categories, we also implemented FSD in the average Word2Vec feature space. Word2Vec [6] is a neural network model that generates vector representations of words (known as, word embeddings). It is trained with a large corpus of texts, and produces a dense vector space, typically of several hundred dimensions, in which each word is represented by a vector. Within this vector space, words with similar contexts are located in close proximity to one another, which is the most important property of Word2Vec. However, todate Word2Vec representations have not been directly applied to FSD; although, Word2Vec has been used indirectly to generate paraphrases in order to alleviate the problem caused by lexical variation [7]. In this experiment, we generate a vector representation for a story by averaging vector representations of the words in the story. We then compare the performance of the different FSD models using this word embedding based vector representations with the tf-idf representations used in the first experiment.

For this second experiment, we use the same data preparation approach that was used for the first experiment. We are aware that there are pre-trained Word2Vec embeddings available; however, for this experiment we train our own Word2Vec embeddings using all the TDT5 data so that the word embedding vectors reflect the most relevant word meanings for the TDT5 context. The number of dimensions of the Word2Vec vectors is set as 300 and the slicing window used in the training is set as 10.

4.4 Evaluation

In both experiments we evaluate FSD performance using the Detection Error Tradeoff (DET) curve [5], which is also the standard evaluation method for FSD since the TDT series. For each FSD model, we run the evaluation with a large range of thresholds. Each threshold generates a pair of False Alarms and Misses error scores, all of which are mapped to the DET curve to show the tradeoff between these two types of errors as the model threshold is varied.

5 Experimental Results

5.1 FSD Performances Across Different Categories of Models

Figure 1 presents the DET curves generated during in experiment 1, by the representatives of the different categories of models. Inspecting the DET curves there is a general trend that performance becomes worse from P2P to P2C and P2A.

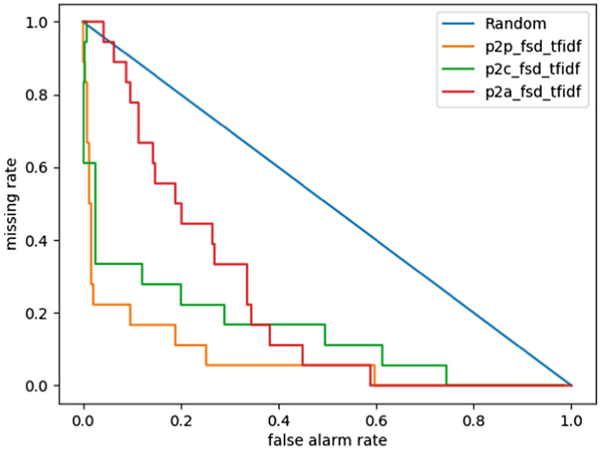
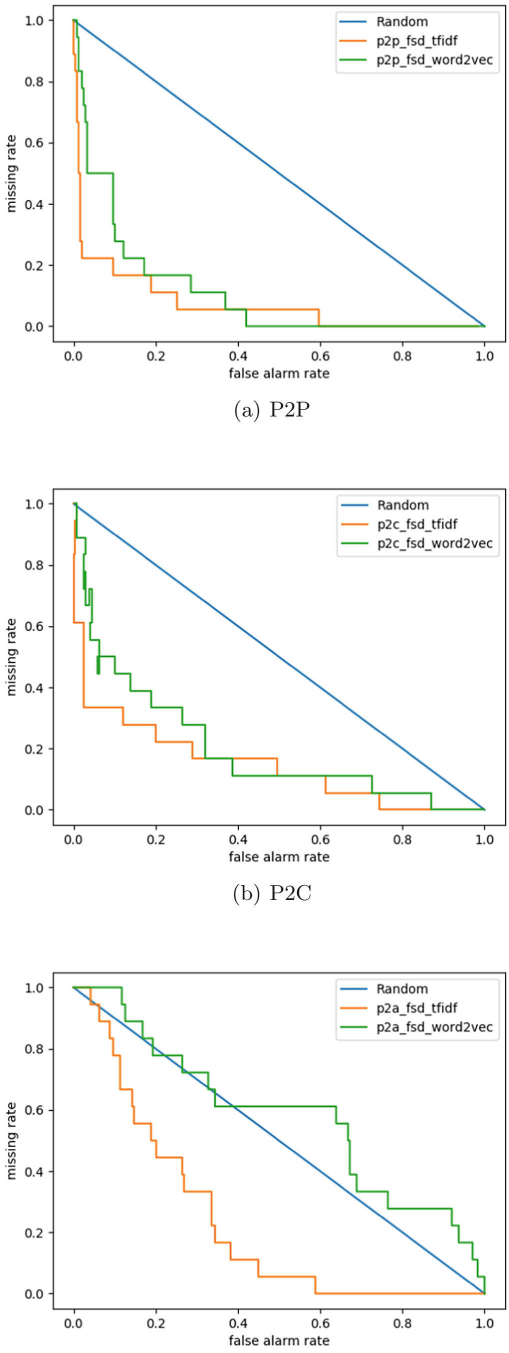


Fig. 1. FSD performances across different categories of models

5.2 FSD Performances in Different Feature Spaces

Figure 2 presents three DET graphs, one for each category of model. Within each graph two DET curves are plotted one for each feature space: tf-idf and Word2Vec. The most important finding is that, for all three categories of models, using Word2Vec representations result in worse performance on FSD, compared with tf-idf. Furthermore, the performance of Word2Vec representations decreases as we move from P2P to P2C and onto P2A. Indeed, the performance of the P2A model with Word2Vec features is worse than random selection.



(a) P2P

(b) P2C

(c) P2A

Fig. 2. Performances of P2P, P2C, and P2A models in different features spaces.

5.3 Result Analysis

Based on the experimental results, general trends in performance across different categories of models emerge, and it is possible to see clear differences in the performance of feature spaces across different categories of models. FSD performance drops as we move from P2P to P2C and P2A, and this trend is evident in the results from both experiments. Furthermore, using Word2Vec representations results in worse performance on FSD, and, again, performance drops as we move from P2P to P2C and onto P2A.

In terms of the explaining why the performances become worse from P2P to P2C and P2A, or from tf-idf to Word2Vec, one potential reason could be that the word specificity is diluted in a large number of documents or in the Word2Vec feature space, which is an important loss of information for novelty detection. The experimental results provide support for our hypothesis. For example, for the topic "Sweden rejected the euro", the P2P model in the tf-idf feature space finds the first story easily, but the P2P model in the Word2Vec feature space usually fails because the first story is considered to be very similar to a previous document that discusses another topic "Portugal and the euro". "Sweden" and "Portugal" are two different words in the frequency-based feature spaces like tf-idf, so the events in Sweden and Portugal can be clearly distinguished from each other within that representation. However, in the Word2Vec feature space, the words with common contexts are located in close proximity to one another, that is, the two words make little difference, so it is difficult to find the novelty caused by the word specificity in the Word2Vec feature space. Similarly, the loss of word specificity happens in P2C and P2A.

6 Conclusions

In this paper, we explored novelty detection by firstly identifying three categories of models, Point-to-Point (P2P), Point-to-Cluster (P2C) and Point-to-All (P2A), based on different definitions of novelty scores. We believe this categorisation leads to a good understanding of novelty and other topics across general categories of detection models. We further put this idea into practice and explored FSD based on different categories of models and different feature spaces. Our experimental results show that the challenge of FSD varies across different novelty scores and corresponding model categories; and, furthermore, that the detection of novelty in the very popular Word2Vec feature space is more difficult than in a normal frequency-based feature space because of a loss of word specificity.

The next phase of our research will be developed across two strands: the application of our categorisation to other contexts to enhance the understanding of the novelty concept; and the examination of the utility of other document embeddings like Doc2Vec to build qualitative evaluation of different feature spaces.

Acknowledgement. The authors wish to acknowledge the support of the ADAPT Research Centre. The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Funds.

References

1. Allan, J., et al.: Topic detection and tracking pilot study final report (1998)
2. Allan, J., et al.: Detections, bounds, and timelines: UMass and TDT-3. In: Proceedings of Topic Detection and Tracking Workshop. sn (2000)
3. Fiscus, J., et al.: NISTs 1998 topic detection and tracking evaluation (TDT2). In: Proceedings of the 1999 DARPA Broadcast News Workshop (1999)
4. Ma, J., Perkins, S.: Online novelty detection on temporal sequences. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2003)
5. Martin, A., et al.: The DET curve in assessment of detection task performance. National Institute of Standards and Technology, Gaithersburg, MD (1997)
6. Mikolov, T, et al.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
7. Moran, S., et al.: Enhancing first story detection using word embeddings. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM (2016)
8. Petrovic, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to Twitter. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics (2010)
9. Petrovic, S., Osborne, M., Lavrenko, V.: Using paraphrases for improving first story detection in news and Twitter. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics (2012)
10. Pimentel, M.A.F.: A review of novelty detection. Signal Process. **99**, 215–249 (2014)
11. Qiu, Y., et al.: Time-aware first story detection in Twitter stream. In: IEEE International Conference on Data Science in Cyberspace (DSC). IEEE (2016)
12. Schlkopf, B.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001)
13. Wang, F., Franco-Penya, H.-H., Kelleher, J.D., Pugh, J., Ross, R.: An analysis of the application of simplified silhouette to the evaluation of *k*-means clustering validity. In: Perner, P. (ed.) MLDM 2017. LNCS (LNAI), vol. 10358, pp. 291–305. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62416-7_21
14. Wurzer, D., Lavrenko, V., Osborne, M.: Twitter-scale new event detection via K-term hashing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (2015)
15. Yang, Y., Pierce, T., Carbonell, J.: A study of retrospective and on-line event detection. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM (1998)