Efficient Online Novelty Detection in News Streams

Margarita Karkali[1], François Rousseau[2], Alexandros Ntoulas[3,4], and Michalis Vazirgiannis[1,2,5]

[1] Athens University of Economics and Business, Greece
[2] LIX, École Polytechnique, France
[3] National and Kapodistrian University of Athens, Greece
[4] Zynga, San Francisco
[5] Institut Mines-Télécom, Télécom ParisTech, France
{karkalimar, mvazirg}@aueb.gr, rousseau@lix.polytechnique.fr, antoulas@di.uoa.gr

Abstract. Novelty detection in text streams is a challenging task that emerges in quite a few different scenarii, ranging from email threads to RSS news feeds on a cell phone. An efficient novelty detection algorithm can save the user a great deal of time when accessing interesting information. Most of the recent research for the detection of novel documents in text streams uses either geometric distances or distributional similarities with the former typically performing better but being slower as we need to compare an incoming document with all the previously seen ones. In this paper, we propose a new novelty detection algorithm based on the *Inverse Document Frequency (IDF)* scoring function. Computing novelty based on IDF enables us to avoid similarity comparisons with previous documents in the text stream, thus leading to faster execution times. At the same time, our proposed approach outperforms several commonly used baselines when applied on a real-world news articles dataset.

Keywords: novelty detection, inverse document frequency, news streams

1 Introduction

A great deal of information consumption these days happens in the form of push notifications: a user specifies a general topic or stream that he is interested in watching or following and a specific service sends updates to his email, desktop or smartphone. In certain cases, the user may be interested in following *all* the stories coming from a specific source. On the other hand, some sources like Twitter, Facebook or certain news sites allow posting of variants of a given story. In such a scenario, the user might be interested in having a way of specifying that he is interested only in stories that he is not aware of, or, in other words, only in stories that are *novel*.

This problem emerges in a variety of different settings, from email threads to RSS readers on a cell phone and is commonly called *First Story Detection* (FSD)[6]. A good novelty detection algorithm can potentially save a lot of time to the user (by hiding known stories and not only previously seen articles) and can also save bandwidth, battery and storage in the mobile setting scenario.

At a high level, previous research on novelty detection consisted of the definition of a similarity (or distance) metric that is used to compare each new incoming story (or document) to a set of previously seen stories. If the similarity of the new incoming document is below a threshold (defined differently in each work) then the document is

[6] Also known as novelty detection, novelty mining, new event detection, topic initiator detection.

considered novel and therefore some relevant action is taken on the document, otherwise it is discarded. The similarity functions used in the literature range in effectiveness and complexity from simple word counts through cosine similarity to online clustering and one-class classification [3, 30, 12, 4].

In prior work, cosine similarity has been reported to work better than most of the previously proposed approaches [3, 30, 4] and was shown to outperform even complex language-model-based approaches in most cases. The documents were represented as bag-of-word vectors with additional TF$\times$IDF term weighting applied on them.

Although previous approaches have been shown to work well in most cases, they have two shortcomings. First, the *document-to-document approaches* (such as the maximum cosine similarity ones [3]) tend to be computationally expensive as we need to compare the new incoming document with all the existing previously seen documents in order to determine its novelty. If the user wishes to have a reasonably large collection of documents to compare to, this approach can prove very costly for a system supporting millions of users or, in the case of a mobile setting, may drain the phone's battery faster. On the other hand, the *document-to-summary approaches* such as the online clustering or one-class classification [22, 12], where we compare the document to a summary (e.g. the centroid of a cluster) are faster and more appropriate for a mobile setting, but they were shown to perform worse than the document-to-document approaches [22, 3].

To this end, we propose a document-to-summary technique that is both efficient computationally and effective in performing novelty detection. Our main idea is to maintain a summary of the collection of previously seen documents that is based on the frequency of each term. We capture the specificity of each term through its *Inverse Document Frequency* (IDF) for a given incoming document and then we show how to compute its overall specificity through the definition of a novelty score. Since our approach is document-to-summary based, we do not compare to all the previous documents and thus we can compute the novelty score faster. At the same time, we show in our experimental evaluation that our approach outperforms several commonly used baseline approaches, in certain cases by a wide margin.

The main contributions of this paper are:

- A new metric for novelty detection based on inverse document frequency that captures the difference of a document's vocabulary with regard to the past.
- An extensive experimental evaluation of our proposed method and the commonly used baselines. Our results indicate that our method outperforms previous ones in both execution time and precision in identifying novel documents.
- A novel annotated corpus that can be used as a benchmark for novelty detection in text streams extracted from a real-world news stream.[7]

2 Related Work

Novelty detection is usually described as a task in signal processing. A survey on methods for novelty detection has been published on Signal Processing Journal by Markou and Singh. The survey is separated in two parts: statistical approaches [12] and neural networks [13]. Novelty detection is a challenging task, with many models that perform well on different data. In this survey, novelty detection in textual data was reported

[7] The dataset is publicly available at: http://www.db-net.aueb.gr/GoogleNewsDataset/

to be a variant of traditional text classification and it was mentioned as an alternative terminology to Topic Detection and Tracking (TDT).

In Topic Detection and Tracking (TDT) field, many papers are dealing with the problem of First Story Detection (FSD). In TDT-3 competition [1], which included a FSD task, Allan *et al*. presented a simple 1-NN approach, also known as UMass [3], that was reported to perform at least as well as the other participants. The UMass approach is constantly used as a baseline in relevant literature. An interesting report from the FSD task in the context of TDT was also published by Allan *et al*. [2], concluding that FSD based on tracking approaches bounds its performance. In our approach we do not rely on model tracking and thus such limitations do not apply.

An interesting work by Yang *et al*. [28] used topic clustering, Named Entities (NE) and topic specific stopword removal for the task of novelty detection on news. In [30], novelty detection at a document level was used in adaptive filtering. The measures tested were separated between geometric distance and language model measures. The results show that the simple approach of maximum cosine distance, introduced by Allan *et al*. in [3], work as well as complex language model measures. A recent work by Verheij et al [27] presents a comparison study of different novelty detection methods evaluated on news article from Yahoo! News Archive where language model based methods perform better than cosine similarity based ones.

Except from the TDT competition, novelty detection was also present in TREC 2002-2004 [7, 21, 20]. Novelty detection was examined at sentence level and the general goal of the track was to highlight sentences that contain both relevant and novel information in a short, topical document stream. A paper by Sobboroff and Harman [22] reported the significant problem in evaluating such tasks, by highlighting problems in the construction of a ground truth dataset.

Based on TREC novelty track, a significant amount of work was published on novelty detection at sentence level [4, 9, 8, 26]. Allan *et al*. [4] evaluated seven measures for novelty detection separating them in word count measures and language model measures. The results again showed that the simple approach of maximum cosine similarity between a sentence and a number of previously seen ones, works as well as complex language model measures. The Meiji University experiments in TREC 2003 [15] proposed a linear combination of the maximum cosine similarity measure with a metric that aggregates the *TF-IDF* scores of the terms in a sentence. This metric is similar to the one presented here, but it is tested for sentence level novelty detection which is a different task from the one we tackle in the current work.

Lately the interest in novelty detection and mainly in FSD is focused at reducing the computation time as FSD is an online task, and the prevalent 1-NN approach uses exhaustive document to document similarity computation. Petrovic *et al*. [16] approximate 1-NN with Locality Sensitive Hashing (LSH). Zhang *et al*. [29] also target in improving the efficiency of novelty detection systems introducing a news indexing-tree. [10] presents a framework for online new event detection used in a real application that focuses on improving system efficiency using indices, parallel processing etc. Our method also manages to increase the efficiency of novelty detection by avoiding exhaustive comparisons (see next section).

Benchmark Datasets for Novelty Detection Novelty detection in text streams is usually evaluated in news applications since this is the most common form of text streams and the task of finding novel news articles makes perfect sense. Most of the work on

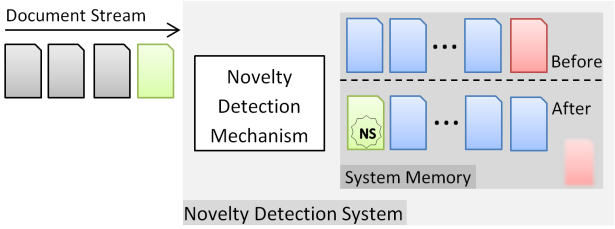

Fig. 1. The process for Novelty Detection.

novelty and first story detection use the TDT datasets for evaluation [1]. The most recent TDT benchmark collection (TDT5) is sparsely labeled: it includes 278,109 English news articles but only around 4,500 are annotated with 100 topics. The TREC novelty track dataset is another benchmark dataset, mainly used for sentence-level novelty detection. It is not suitable for the purpose of this paper as it contains novelty judgments per sentence and not per document. [25] is the only work using it at a document level considering the number of novel sentences per document but we believe that such an assumption cannot lead to safe conclusions. Other works [28, 30] use available news article collections and apply sampling and manual labeling using well-known events in a specific time span. Details for these datasets are also available in [24].

All the above evaluation datasets are manually annotated using predefined events. Thus there is always the issue of human subjective judgment that introduces a degree of uncertainty. In addition, only a small proportion of the stream is annotated.

3 Novelty scoring Methods

We consider a system that monitors a stream of documents. New documents reach the system at different times. We assume that documents arrive ordered by their creating time (timestamp). Each document d^t, with a timestamp t, is represented using a *bag-of-word* approach, as $< (q_1, w_1^C), (q_2, w_2^C), ..., (q_{|d^t|}, w_{|d^t|}^C) >$, where q_i is the i^{th} unique term in document d^t and w_i^C is the corresponding weight computed with regard to a corpus C. When a new document d^t arrives in the system, the previous N ones are already stored and indexed. We use the terms *memory* and *corpus* for this set of documents interchangeably in the paper. Assuming the corpus C, for each new document d^t, a novelty score $NS(d^t, C)$ is computed, indicating the novelty of this document for the given corpus. d^t is then stored in memory and the oldest document is flushed. This process is illustrated in Figure 1.

We define the Novelty Detection (ND) problem as the characterization of an incoming document as novel with respect to a predefined window in the past. In the described context, we declare novel a document d^t when the corresponding novelty score $NS(d^t, C)$ is higher than a given threshold θ.

3.1 Baselines

Document-to-Document using Vector Space As mentioned earlier, methods based on cosine similarity are proved to work better in similar tasks and are frequently used as a baseline in Novelty Detection. As for the *Max Cosine Similarity* baseline, it was introduced by Allan *et al* at TDT3 in [3] and is also known as the UMass. This method is used as a baseline also by [16, 29, 30] and it is considered the traditional method for document novelty detection. The intuition of this metric is that if a new document is

very similar to another in the corpus, the information it contains was seen before and thus the document cannot be considered as novel. We also introduce a second baseline using the *Mean Cosine Similarity*. Similarly, a document is marked as novel if its mean similarity to the documents in the corpus is below a threshold.

Assuming the cosine similarity between two documents d and d' is defined as:

$$CS(d, d') = \frac{\sum_{k=1}^{m} w_k(d) w_k(d')}{\sqrt{\sum_{k=1}^{|d|} w_k(d)^2 \sum_{k=1}^{|d'|} w_k(d')^2}} \tag{1}$$

where $w_k(d)$ the weight of the term k in document d and m the number of common terms among the two documents, then the respective similarity formulas for the aforementioned metrics are as follows:

$$MaxCS(d^t, C) = \max_{1 \leq i \leq |C|} CS(d^t, d_i) \tag{2}$$

$$MeanCS(d^t, C) = \frac{\sum_{i=1}^{|C|} CS(d^t, d_i)}{|C|} \tag{3}$$

Both approaches are simple to implement but their computational complexity depends on the length of the corpus used. In the worst case the complexity is $O(|d^t| \times |C|)$.

Document-to-Document using Language Models A common method to measure the similarity between two documents is using language models. A recent comparison study by Verheij et al [27], where a number of methods were used for novelty detection, reports that the best performing method was document-to-document distance based on language models.

We use minimum Kullback-Leibler (KL) divergence as a baseline approach based on LMs. We implemented the method as described in [27]. Thus, assuming the KL divergence of a document d given a document d' is as follows:

$$KL(\Theta_d, \Theta_{d'}) = \sum_{q \in d} \Theta_d(q) \log \frac{\Theta_d(q)}{\Theta_{d'}(q)} \tag{4}$$

where Θ_d is the unigram language model on document d and $\Theta_d(q)$ is the probability of term q in document d, then the respective novelty scoring formula is as follows:

$$MinKL(d^t, C) = \min_{1 \leq i \leq |C|} KL(\Theta_{d^t}, \Theta_{d_i}) \tag{5}$$

In order to avoid the problem of zero probabilities we use linear interpolation smoothing, where document weights are smoothed against the set of the documents in the corpus. Then the probabilities are defined as $\Theta_{d^t}(q) = \lambda \times \Theta_{d^t}(q) + (1-\lambda) \times \Theta_{d^1 \ldots d^{t-1}}(q)$, where $\lambda \in [0, 1]$ is the smoothing parameter and $\Theta_{d^1 \ldots d^{t-1}}$ the probability of term q in the corpus C. In our experiments, λ was set to 0.9 based on the experiments in [27].

Document-to-Summary using Vector Space Alternatively, we can maintain a summary of the previously seen documents and compare the new one only to this summary, avoiding computationally expensive comparisons with all the past documents.

To have a complete set of baselines for the evaluation of our method, we also include a document-to-summary approach based on vector space as the document representation and cosine similarity as the novelty metric. The corpus summary is defined based

Table 1. Extended SMART notations that include BM25 components

Notation	Term frequency	Notation	IDF	Notation	Normalization
b (boolean)	$\begin{cases} 1 & \text{if } tf > 0 \\ 0 & \text{otherwise} \end{cases}$	t (idf)	$\log \frac{N}{df}$	n (none)	1
n (natural)	tf	p (prob. idf)	$\log \frac{N-df}{df}$	u (# unique terms)	$\|d\|$
l (logarithm)	$1 + \log tf$			d (L^1 norm)	dl
k (BM25)	$\frac{(k_1+1)\cdot tf}{k_1 \times (1-b+b \times \frac{dl}{avdl}) + tf}$	b (BM25)	$\log \frac{N-df+0.5}{df+0.5}$	c (L^2 norm)	$\sqrt{\sum tf^2}$
				p (pivot)	$1 - b + b \times dl/avdl$

on [27], as the concatenation of all the documents in corpus, i.e. $D_C = \bigcup_{d \in C} d$. Then the novelty scoring formula for the document-to-summary baseline can be defined as follows:

$$SumCS(d^t, C) = CS(d^t, D_C) \tag{6}$$

3.2 Inverse Document Frequency for Novelty

Design Principles In this paper, we introduce a novelty score that *does not use any similarity or distance measure*. This novelty score can be considered as a way to compare a document to a corpus, which is the essence of a novelty detection task.

To do so, we capitalize on the *Inverse Document Frequency* (IDF) measure introduced in [23]. IDF is a heuristic measure for term specificity and is a function of term use. More generally, by aggregating all the IDF of the terms of a document, IDF can be seen as a function of the vocabulary use at the document level. Hence, our idea to use it as an estimator of novelty – a novel document being more likely to use a different vocabulary than the ones in the previous documents. In a way, a document is novel if its terms are also novel – i.e. previously unseen. This implies that the terms of a novel document have a generally high specificity and therefore high IDF values.

IDF was initially defined as $idf(q, C) = \log \frac{N}{df_q}$. where q is the considered term, C the collection, df_q the document frequency of the term q across C and N the size of C, i.e. the number of documents. There exists a slightly different definition known as *probabilistic* IDF used in particular in BM25 [18] where the IDF is interpreted in a probabilistic way as the odds of the term appearing if the document is irrelevant to a given information need and defined as $idf_{prob.}(q, C) = \log \frac{N-df_q}{df_q}$. Note that this IDF definition can lead to negative values if the term q appears in more than half of the documents as discussed in [17]. For ad-hoc information retrieval, it has been claimed that it violates a set of formal constraints that any scoring function should meet [5] but for novelty detection, this property could be of importance as we want to penalize the use of terms appearing in previously seen documents. We will test both versions in our experiments. Both versions also have smoothed variants for extreme cases where the document frequency could be null or equal to the size of the collection (by usually adding 0.5 to both numerator and denominator). These are the ones that we will use in practice since the collection is pretty small (memory of the last 100 documents for example) and thus subject to sparseness in the vocabulary.

Novelty Score Definition and Properties It seems then natural to define our novelty score as a TF×IDF weighting model since we are relying on a *bag-of-word* representation and a *vector space* model. The task here is more of *filtering* than ad-hoc IR, hence the TF component needs not to be concave and pivot document length normalized as

in BM25. We explored indeed a great variety of combinations for TF and IDF that we will present following the SMART notations (the historical ones defined in [19] and additional ones that include BM25 components). In general, the novelty score of a new document d for a collection C can be defined as follows:

$$NS(d, C) = \frac{1}{norm(d)} \sum_{q \in d} tf(q, d) \times idf(q, C) \tag{7}$$

where tf, idf and $norm$ can be any of the functions presented in Table 1, ranging from a standalone IDF (btn) to a BM25 score (kbn) using the SMART triplet notation. Note that because of the way BM25 is designed, the length normalization is already included in the TF component ($k\_\_$) for a slop parameter b greater than 0. Therefore, BM25 is denoted by kbn.

The aggregation (through the sum operation) of the term scores to obtain a document score reduces the impact of synonymy which is a common problem when using *bag-of-word* representation and *vector space* model. Indeed, a document that would have terms synonymous with the ones in the other documents would probably be detected as novel since its terms have high IDF values. Nevertheless, it is very unlikely that all its terms are synonymous and overall, its score should not be as high as the one of a novel document.

Unlike the approaches described in 3.1, this measure is not related to the size N of the corpus used. Its complexity is $O(|d|)$. In addition, no document vector needs to be retrieved (and a fortiori stored in an inverted index except for d) for the computation of NS. The index is only used after the score has been assigned in order to decrease the *document frequency* of the terms occurring in the oldest document (the one being flushed). Thus, the response time of the system is not affected.

4 Experiments

4.1 Datasets

Google News Dataset We wanted a dataset with ground truth judgments regarding the first story of each news cluster. Towards this direction, we worked for the construction of an annotated dataset from a real world news stream. We used the RSS feeds provided by the Google News aggregator.

The method for creating the *Google News dataset* was the following: we periodically collected all articles from the RSS, offered by Google News, for the category "Technology" published in the time period July 12 to August 12, 2012. All articles are from the English news stream. Each news unit consists of the article title, a small description (snippet), the URL for the article, the publication date and a cluster id, assigned by the aggregator, clustering threads of news. In addition, we used an open source script for main content extraction from news websites[8] to get the main content of the articles from the article URLs. We applied two standard preprocessings: stopword removal and Porter's stemming. Then for each article we store the set of unigrams and their corresponding local frequency (*TF*) for the article snippet and content separately.

Annotation Process: We take advantage of the cluster information provided by the Google News to create the ground truth dataset for our experiments. Thus, the goal set is to identify as novel the *first article* in each news cluster. Unfortunately, as the clustering in Google News is carried out via an automated mechanism, there is no guarantee

[8] http://goo.gl/LKahS

that the articles in a single cluster refer to the same real world event. To have a reliable ground truth dataset, we assign to human annotators the task of *correcting* the clusters retrieved from Google News RSS. The annotators have to assign one of the following labels to the cluster: *clean*, *separable*, *part of an existing one* or *mixed*. A clean cluster contains articles that refer to the same event (e.g. *Release of iPhone5*). A *separable* cluster contains articles from more than one event that can easily be detected and annotated. An example of such cluster contained 22 articles for the *Antitrust investigation of Microsoft by EU* and 11 articles for *Windows 8 release on October 26*. For each *separable*, the corresponding number of new clean clusters was created. When a cluster is declared as *part of an existing one*, the two clusters are merged. If the cluster mixes too many events that could not be easily distinguished by the annotator, the cluster is marked as *mixed* and it is not considered for evaluation. We do not consider *mixed* clusters for evaluation because such clusters contain more than one article that should be considered as novel.

The dataset we produced has some advantages over the other benchmark datasets such as TDT5. In those datasets (some in the scale of 10^5 articles) it is the human annotators that decide the similarity among news articles and therefore clustering before they identify the first occurrence of the cluster. Apparently the result is introduction of noise and errors with very high probability due to the diverse background of the annotators and the chance that some articles - due to human error or negligence - are left out of the thematic news clusters. In our case the dataset contains already ground truth in grouping the articles into clusters - and the annotators only improve the few (compared to the documents) clusters. In any case there is no doubt on the first article per cluster as it is the temporally first in the clusters. Thus the probability for errors and more importantly missing the first article on a cluster is much smaller. Finally, the introduced dataset does not suffer from sparseness as all of the previously used ones.

The annotation process reduced the initial data set of 3300 articles/673 clusters to 2006 articles/555 clusters. The cluster size distribution is biased as about half of the clusters (247) consist of only one article while another 261 have between 1 and 10 articles and only 47 have more than 10 articles in each cluster. Also the topics of the news clusters are quite characteristic, we refer to Table 2 for a list of the topics and sizes of the larger clusters.

Note that we use the *actual stream* including *all articles* published during the predefined one month period. We exclude *mixed* clusters only from the final evaluation of the detection task.

Table 2. Sample of Topics and Cluster sizes

Cluster Topic	**Size**	**Cluster Topic**	**Size**
Apple Considered Investing in Twitter	20	Google Nexus 7 tablet goes on sale in US	21
VMware buys Nicira for $1.05 billion	21	Google unveils price for Gbit Internet service	21
Digg acquired by Betaworks	23	Microsoft Reboots Hotmail As Outlook	27
FTC Fines Google for Safari Privacy Violations	27	Nokia cuts Lumia 900 price in half to $50	30
Apple Brings Products Back Into EPEAT Circle	31	Yahoo confirms 400k account hacks	45

Twitter Dataset To examine the potential of our method for very small documents, we used a second dataset consisting of real tweets. This synthetic dataset was constructed using the annotated proportion of the one described in [16]. The dataset contains 27

events of various lengths, from 2 to 837 tweets. The whole dataset consist of 2600 tweets. Events include "*Death of Amy Winehouse*", "*Earthquake in Virginia*" and "*Riots break out in Tottenham*". The stream created uses the actual temporal order of these tweets. Most of the events are well separated from each other with eight of them having a small overlap in time. Here, again we consider as novel only the first story in time for each event. The dataset is available from the website of the CROSS project[9] in the context of which it was created.

4.2 Evaluation Methodology

Detection Errors The performance of a Novelty Detection algorithm is defined in terms of the missed detection and false alarm error probabilities as defined in [6]. A signal detection model, variation of ROC curves, is often used for evaluation; the Detection Error Trade-off (DET) curve [14], which illustrates the trade-off between missed detections and false alarms. On the x-axis is the miss rate and on the y-axis is the false alarm rate. A system is considered to perform best when it has its curve towards the lower-left of the graph. The axes of the DET curve are on a Gaussian scale.

For the detection systems evaluation, these error probabilities are usually linearly combined into a single detection cost, C_{Det} [6, 11] defined as:

$$C_{Det} = (C_{Miss} \times P_{Miss} \times P_{Target} + C_{Fa} \times P_{Fa} \times (1 - P_{Target})) \quad (8)$$

where P_{Miss} is the number of missed detections divided by the number of target articles, P_{Fa} the number of False Alarms divided by the number of non-targets, C_{Miss} and C_{Fa} the costs of a missed detection and a false alarm respectively, P_{Miss} and P_{Fa} the probabilities of a missed detection and a false alarm respectively and P_{Target} the a priori probability for finding a target. For our experiments we set the same cost for missed detections and false alarms ($C_{Miss} = C_{Fa} = 1$) and the same probability for finding a target and a non-target ($P_{Target} = 0.5$) assuming no prior knowledge for the probability of targets.

Cross-validation Since the goal of a detection task is to minimize the detection cost C_{DET}, $minC_{DET}$ is used to define the optimum threshold, i.e. the threshold that gets the lowest C_{DET} value is the best to use for this detection model. The $minC_{DET}$ also corresponds to a certain point on the DET Curve, as the DET curve illustrates the different operating points of a detection system (i.e. the detection errors for different thresholds). In order to avoid an overfitting effect over our datasets, we used 5-fold cross validation in our experiments. We computed the $minC_{DET}$ and the corresponding threshold on the training part and then computed C_{DET} on the testing part. We will report the average C_{DET} in test for all our experiments.

Baselines We use the ground truth information of each dataset to evaluate the performance of novelty detection for our method and in comparison against the four *baseline approaches*. These methods take into account the similarity/divergence among the document under evaluation and the previous N documents or their summary and rate it as novel based on a threshold. For the experiments, the weighting model used for the baselines is BM25 (kbn in SMART notation), which is the one used in [3].

[9] http://demeter.inf.ed.ac.uk/cross/

Table 3. Average C_{DET} using 5-fold cross validation on Snippets and Content.

	Snippet					Content				
N	**20**	**60**	**100**	**140**	**180**	**20**	**60**	**100**	**140**	**180**
btd	0.439	**0.407**	0.408	0.411	0.397	0.434	0.429	0.436	0.433	**0.418**
bbd	0.432	**0.404**	0.405	0.411	0.398	0.433	0.423	0.431	**0.416**	0.416
ntd	0.287	**0.266**	0.284	0.285	0.297	0.391	0.366	**0.362**	0.396	0.386
nbd	0.294	**0.267**	0.291	0.284	0.307	0.4	**0.367**	0.373	0.4	0.394
ltd	0.413	0.382	**0.359**	0.371	0.368	0.429	0.425	0.412	**0.405**	0.413
lbd	0.393	0.381	**0.36**	0.374	0.373	0.422	0.414	**0.412**	0.419	0.415
ktd-b=0	0.394	0.367	**0.354**	0.362	0.365	0.429	0.428	0.411	**0.404**	0.412
kbd-b=0	0.392	0.368	**0.346**	0.36	0.366	0.424	0.415	**0.413**	0.419	0.422
btu	0.307	0.299	**0.293**	0.296	0.311	0.447	0.444	0.451	0.434	**0.429**
bbu	0.313	0.299	**0.294**	0.297	0.307	0.44	0.452	0.45	**0.425**	0.429
ntu	0.319	**0.293**	0.294	0.317	0.303	0.464	**0.441**	0.447	0.442	0.46
nbu	0.324	**0.288**	0.298	0.302	0.308	0.458	**0.437**	0.44	0.44	0.449
ltu	0.298	0.283	**0.281**	0.283	0.299	0.455	**0.423**	0.424	0.436	0.461
lbu	0.301	**0.276**	0.281	0.288	0.298	0.455	**0.429**	0.438	0.436	0.447
ktu-b=0	0.295	0.282	**0.268**	0.28	0.296	0.452	**0.42**	0.427	0.446	0.458
kbu-b=0	0.298	0.283	**0.279**	0.279	0.302	0.458	**0.422**	0.439	0.438	0.45
btn	0.429	0.41	0.391	**0.389**	0.398	**0.503**	0.504	0.504	0.504	0.504
bbn	0.417	0.402	**0.385**	0.388	0.403	**0.496**	0.504	0.502	0.505	0.507
ntn	0.371	**0.332**	0.337	0.336	0.346	**0.483**	0.499	0.499	0.497	0.499
nbn	0.366	0.333	**0.33**	0.34	0.339	**0.492**	0.502	0.501	0.508	0.509
ltn	0.401	0.375	**0.364**	0.37	0.371	0.502	0.509	**0.502**	0.504	0.502
lbn	0.399	0.372	0.374	**0.364**	0.371	0.498	0.515	**0.501**	0.503	0.505
ktn-b=0	0.395	0.368	0.372	0.364	**0.36**	**0.499**	0.51	0.503	0.508	0.505
ktn-b=0.75	0.385	0.342	**0.337**	0.348	0.343	0.461	**0.432**	0.434	0.439	0.447
kbn-b=0	0.39	0.363	0.363	**0.36**	0.367	0.498	0.516	0.501	**0.5**	0.505
kbn-b=0.75	0.38	0.343	**0.335**	0.347	0.343	0.441	**0.428**	0.445	0.451	0.446
maxCS	0.375	0.369	0.365	0.368	**0.353**	0.492	0.465	0.457	0.456	**0.45**
meanCS	0.368	0.362	0.344	**0.339**	0.355	0.471	0.461	0.444	0.44	**0.439**
maxKL	0.448	0.438	0.423	**0.421**	0.43	0.49	0.458	0.449	**0.434**	0.471
CSAgg	0.451	0.425	0.424	0.425	**0.423**	0.494	0.48	0.476	**0.47**	0.489

Weighting models As mentioned in section 3, we are using a variety of TF×IDF weighting models that we will refer to using the SMART notations presented in Table 1. For TF, we used the variants b (boolean term representation), n (plain term frequency), l (logarithmic saturation) and k ($BM25$ saturation) and for IDF, we considered the following variants: t (the plain IDF value) and b (the form used in $BM25$). Finally we tested three different options for document length normalization: n (none), d (the document length) and u (the number of unique terms).

5 Results

In this section we present and review the results of the experiments on the datasets mentioned in the previous sections and for all the combinations of measures and parameters values mentioned.

Google News dataset We present here the average detection cost ($avgC_{DET}$) for the cleaned dataset with memory size (i.e. length of the corpus) N ranging between 20 and 180 with step 40 for a variety of meaningful combinations of the variants of term frequency, IDF and normalization. We report these results for the snippets and the full articles versions of the dataset (Table 3).

The result table is organized in blocks of lines based on the normalization method. The top block (model SMART acronym ending in d) corresponds to normalization

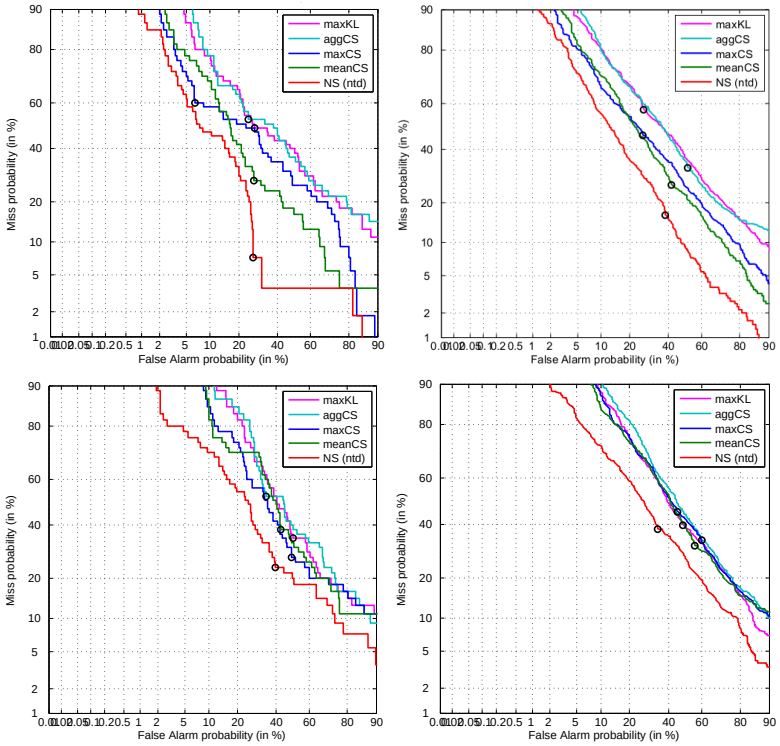

Fig. 2. DET Curves for N=100 on Clusters with size$>=10$ using snippets (top-left), all Clusters using snippets (top-right), size$>=10$ using content (bottom-left) and all Clusters using content (bottom-right).

based on the document length, the mid block (SMART acronym ending in u) to normalization based on the number of unique terms in the document and the third one (SMART acronym ending in n) is for the case where no normalization takes place. The last four rows of the table represent the results of the baseline methods (*MaxCS*, *MeanCS*, *MaxKL*, *SumCS*).

The values appearing in the cells represent the average detection cost in test (computed using 5-fold cross validation) for each combination of parameters. We excluded some combinations of the above parameters as they introduce normalization twice (ktd-b=0.75, kbd-b=0.75, ktu-b=0.75, kbu-b=0.75). This is because TF variation k, used in BM25, introduces a length normalization prior to the saturation in its formula for $b > 0$.

Given the above hints, we notice that almost all methods best results are obtained for memory size (N) either 60 or 100 thus we focus our further comments on the respective results columns.

It is evident that the proposed Novelty Scoring measure outperforms the all baselines with the best performance achieved by a L^1 normalized TF-IDF (raw TF and classic IDF – ntd) narrowly followed by the nbd model (same except for the IDF component inherited from BM25). Very good performance is achieved by the u normalization (number of unique terms) especially for the ltu and kbu models. Absence of length normalization yields to the worst results as it can be expected with documents of vary-

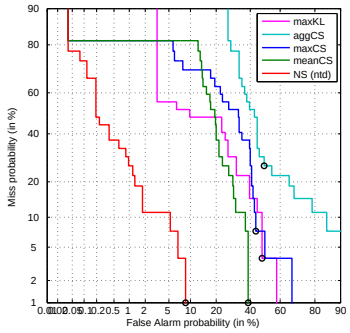

Fig. 3. DET Curve for N=100 on Twitter dataset.

ing length. Nevertheless it still outperforms the best baseline results when we consider the snippets where the variation in length is limited. Considering the content of the articles for novelty detection the no normalization weighting schemes perform much worst even than the baselines. The difference in performance of this group in comparison to the one on snippets originates at the greater differences in document lengths when the full article is taken into account (snippets tend to have a constant length – around 25 terms). Note that ktn-$b = 0.75$ and kbn-$b = 0.75$ perform better than the rest of the block. This can be easily explained as both methods use the BM25 variant of TF which includes a pivot length normalization for parameter $b > 0$. We chose to display them in that block just to be consistent in terms of SMART notations.

DET Curves:We plotted DET curves showing the evolution of performance with regards to the Miss and the False Alarm probability. These diagrams indicate the evolution of the detection cost for the best performing model (*ntd*) and all the baselines. They also depict the point on each curve that corresponds to the optimum threshold, having the $minC_{DET}$.

In figure 2 we plot the DET Curves for memory N=100 on four versions of the Google News dataset, large Clusters with size$>=10$ using snippets (top-right), size$>=10$ using content (bottom-left) and all Clusters using content (bottom-right). We compare all baseline methods and our method using the best performing weighting schema, ntd(see table 3). It is clear that overall the ntd method outperforms the others. The baseline based on maximum KL-divergence and document-to-summary baseline perform worst. The same applies for the case of *all Clusters* data set. In addition, comparing the corresponding snippet and content DET curves we confirm again our previous claims that using the full content of an article instead of a simple summary as the first few sentences of the article introduces significant noise and makes it harder to detect the first stories.

Twitter Dataset We report in Figure 3 the results on the Twitter dataset described in section 4.1. We again compare all baseline methods and our method using the best performing weighting schema, ntd (see table 3) for $N = 100$. As mentioned earlier, we exhaustively examine the performance of our method for the *Twitter Dataset* as well. Due to lack of space, we present the results of the best performing weighting scheme and N value using the DET Curves as a more concise means. The results are

Table 4. Execution times per article in microseconds for different N values, using content

	N=20	N=60	N=100	N=140	N=180
NS	**124.44**	**154.46**	**128.50**	**200.54**	**134.96**
meanCS	704.06	1798.30	2372.72	3156.27	3923.91

very encouraging, as our method outperforms by far all the baselines and manages to have a zero miss probability while maintains false alarm probability below 10%.

Execution Time We compared our method in terms of execution time with the best performing method from the baselines, *MeanCS*. We ran experiments for different values of N, using content. We used the whole stream of news. The results are shown in Table 4. The values reported correspond to the average time needed to process and assign a novelty score to an article in the dataset. The time cost for database connection and communication, indexing and index updating is not considered.

It is clear that our method is considerably faster than the document-to-document competing ones as it is at least seven times faster than *MeanCS*. The difference among the methods increases as the corpus length increases, since *MeanCS*, as any document-to-document method, have to be executed on the entire corpus to compute the similarity between all documents.

6 Conclusion

Novelty detection is an important topic in modern text retrieval systems. In this paper we proposed a new method for the novelty detection task in document streams that is accurate (i.e. performing better than several dominant baselines). We conducted extensive experiments on a real world dataset (from a news stream) where our method clearly outperforms the four baseline techniques used in the relevant literature. Moreover, as our method does not use any similarity or distance measure among documents but only stream statistics kept in memory, it is much faster and scalable than the others.

These results give strong evidence that stream statistics, such as IDF in our case, can alone be used to detect novel documents from streams. IDF is a simple yet effective indicator of both *term specificity* and *document novelty*. The first property has been known since 1972 and our work just showed the second one. In large-scale streaming, such as on Twitter that recently sparked interest in the research community, this observation may be of great importance.

7 Acknowledgments

M. Karkali has been co-financed by the EU (ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the NSRF - Heracleitus II. This work was also supported by PIRG06-GA-2009-256603.

References

1. J. Allan. Introduction to topic detection and tracking. In J. Allan, editor, *Topic Detection and Tracking*, volume 12 of *The Information Retrieval Series*, pages 1–16. Springer US, 2002.
2. J. Allan, V. Lavrenko, and H. Jin. First story detection in tdt is hard. CIKM '00, pages 374–381. ACM, 2000.
3. J. Allan, V. Lavrenko, D. Malin, and R. Swan. Detections, bounds, and timelines: Umass and tdt-3. In *Topic Detection and Tracking Workshop (TDT-3)*, 2000.
4. J. Allan, C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. SIGIR '03, pages 314–321. ACM, 2003.
5. H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. SIGIR '04, pages 49–56. ACM, 2004.

6. J. G. Fiscus and G. R. Doddington. Topic detection and tracking. In J. Allan, editor, *Topic detection and tracking*, chapter 1, pages 17–31. Kluwer Academic Publishers, 2002.
7. D. Harman. Overview of the trec 2002 novelty track. In *TREC 2002, NIST Special Publication 500-251*, pages 46–55, 2002.
8. A. T. Kwee, F. S. Tsai, and W. Tang. Sentence-level novelty detection in english and malay. PAKDD '09, pages 40–51. Springer-Verlag, 2009.
9. X. Li and W. B. Croft. Novelty detection based on sentence level patterns. CIKM '05, pages 744–751. ACM, 2005.
10. G. Luo, C. Tang, and P. S. Yu. Resource-adaptive real-time new event detection. SIGMOD '07, pages 497–508. ACM, 2007.
11. R. Manmatha, A. Feng, and J. Allan. A critical examination of tdt's cost function. SIGIR '02, pages 403–404. ACM, 2002.
12. M. Markou and S. Singh. Novelty detection a review–part 1: statistical approaches. *Signal Process.*, 83(12):2481–2497, Dec. 2003.
13. M. Markou and S. Singh. Novelty detection a review-part 2: neural network based approaches. *Signal Process.*, 83(12):2499–2521, Dec. 2003.
14. A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The det curve in assessment of detection task performance. In *5th European Conference on Speech Communication and Technology*, pages 1895–1898, 1997.
15. R. Ohgaya, A. Shimmura, T. Takagi, and A. N. Aizawa. Meiji university web and novelty track experiments at trec 2003. In *TREC 2003*, pages 399–407, 2003.
16. S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to twitter. HLT '10, pages 181–189. ACL, 2010.
17. S. E. Robertson and S. Walker. On relevance weights with little relevance information. *SIGIR Forum*, 31(SI):16–24, July 1997.
18. S. E. Robertson, S. Walker, K. Sparck Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. TREC-3, pages 109–126, 1994.
19. A. Singhal, G. Salton, and C. Buckley. Length normalization in degraded text collections. Technical report, Cornell University, Ithaca, NY, USA, 1995.
20. I. Soboroff. Overview of the trec 2004 novelty track. In *TREC 2004, NIST Special Publication 500-251*, 2004.
21. I. Soboroff and D. Harman. Overview of the trec 2003 novelty track. In *TREC 2003, NIST Special Publication 500-251*, 2003.
22. I. Soboroff and D. Harman. Novelty detection: the trec experience. HLT '05, pages 105–112. ACL, 2005.
23. K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–20, 1972.
24. F. S. Tsai. Review of techniques for intelligent novelty mining. *Information Technology Journal*, 9:1255–1261, 2010.
25. F. S. Tsai and A. T. Kwee. Experiments in term weighting for novelty mining. *Expert Systems with Applications*, 38(11):14094 – 14101, 2011.
26. F. S. Tsai, W. Tang, and K. L. Chan. Evaluation of novelty metrics for sentence-level novelty mining. *Inf. Sci.*, 180(12):2359–2374, June 2010.
27. A. Verheij, A. Kleijn, F. Frasincar, and F. Hogenboom. A comparison study for novelty control mechanisms applied to web news stories. WI 2012, pages 431–436. IEEE Computer Society, 2012.
28. Y. Yang, J. Zhang, J. Carbonell, and C. Jin. Topic-conditioned novelty detection. KDD '02, pages 688–693. ACM, 2002.
29. K. Zhang, J. Zi, and L. G. Wu. New event detection based on indexing-tree and named entity. SIGIR '07, pages 215–222, New York, NY, USA, 2007. ACM.
30. Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. SIGIR '02, pages 81–88. ACM, 2002.