Algorithms **2012**, *5*, 364-378; doi:10.3390/a5030364

OPEN ACCESS

algorithms

ISSN 1999-4893

www.mdpi.com/journal/algorithms

Article

Incremental Clustering of News Reports

Joel Azzopardi * and Christopher Staff

Faculty of ICT, University of Malta, Msida, MSD2080, Malta; E-Mail: joel.azzopardi@um.edu.mt

* Author to whom correspondence should be addressed; E-Mail: joel.azzopardi@um.edu.mt;
Tel.: +356-2340-2844; Fax.: +356-2144-0972.

Received: 29 June 2012; in revised form: 13 August 2012 / Accepted: 15 August 2012 / Published: 24 August 2012

Abstract: When an event occurs in the real world, numerous news reports describing this event start to appear on different news sites within a few minutes of the event occurrence. This may result in a huge amount of information for users, and automated processes may be required to help manage this information. In this paper, we describe a clustering system that can cluster news reports from disparate sources into event-centric clusters—*i.e.*, clusters of news reports describing the same event. A user can identify any RSS feed as a source of news he/she would like to receive and our clustering system can cluster reports received from the separate RSS feeds as they arrive without knowing the number of clusters in advance. Our clustering system was designed to function well in an online incremental environment. In evaluating our system, we found that our system is very good in performing fine-grained clustering, but performs rather poorly when performing coarser-grained clustering.

Keywords: clustering; news; event detection; incremental clustering

1. Introduction

Within some time of an event happening in the real world, numerous news reports will appear on news sites on the World Wide Web that describe that event. Moreover, as time passes, new reports will appear that give information on how that event developed and what its effects were. Huge quantities of information presented to user may lead to confusion on the side of the user, and some knowledge contained within this information may remain hidden since the user may not have time to go through the entire corpus of information to get each nugget of information, or may miss it as it lies "buried" in familiar information.

Automated processes such as Document Fusion and Recommendation systems can be used to assist the user in his/her quest for knowledge discovery. The tasks of such automated systems may be rendered simpler by having a mechanism that clusters news stories together by specific events (e.g., news reports on the murders in Norway by Anders Breivik). Although existing news aggregators (e.g., Google News) provide this functionality, the user would be limited to the news stories that the aggregator tracks. We anticipate that users may wish to choose their own news-feeds so we have implemented a modified version of a clustering algorithm.

In this research, we have implemented a clustering system that uses a modified k-means clustering algorithm adapted for incremental clustering to cluster news reports into event-centric clusters. News reports are downloaded continuously through RSS, and these are clustered "on the fly", resulting in the creation of new clusters or the updating of previously existing clusters. The number of clusters is not known in advance, and new events are detected automatically. Since the stream of news reports never terminates, cluster reorganisation is not feasible unlike the "standard" k-means algorithm.

Evaluation results show that the algorithm used to cluster news reports works very well when the clustering required is highly specific—*i.e.*, when each output cluster should contain only news reports describing a specific event (e.g., news reports describing a particular murder in Malta are clustered separately from news reports describing a murder in Italy). On the other hand, in cases where the requested output clusters are more generic—e.g., clustering all news reports about sports, our system obtains worse results than the baseline system (the standard k-means system). However, since our aim in this research is the clustering of reports into event clusters, our document clustering system is suitable for our purposes.

Our document clustering system was implemented to be the initial part of a larger system that performs news report fusion and issues personalised recommendations—refer to [1,2] for more details. Our system downloads RSS feeds from 9 different news sources (including Maltese local sources and international sources such as Reuters). The categorisation process of a single news report, on average, takes less than 1 second on a Linux quad-core system with CPUs running at 2.4 GHz and 4 MB of cache, and with 8 GB of RAM.

This paper proceeds as follows: In Section 2 we give an overview of related systems in literature; we describe our clustering system in Section 3; Section 4 describes how we performed our evaluation and presents the results obtained; and finally we present our conclusions in Section 5.

2. Background

Document Clustering refers to the process whereby a collection of documents is clustered in an unsupervised manner into clusters such that documents within the same cluster are more similar to each other than to documents in other clusters [3,4]. Automatic Document Clustering is part of the larger domain of knowledge management and automatic document content analysis [3,5–8]. It can serve as an intermediate step to perform other tasks [8].

The main justification behind the need for automatic document clustering systems is the information overload problem [4,6,7,9–13]. The amount of information available to users, especially textual data, has exploded with the advent of the World Wide Web and is thus becoming unlimited. There is a growing need for techniques to handle this increasing flood of incoming data and avoid delay in its distribution [6,9].

The presence of online sources of data continuously streaming out new documents, such as news sites and blogs, necessitates the use of incremental document clustering systems [13,14]. Aslam [10] distinguishes between off-line and online clustering algorithms whereby he claims that off-line algorithms are useful to organise static collections whilst incremental on-line algorithms are useful to organise dynamic corpora. In Topic Detection and Tracking (TDT) systems, the incoming stream/s of news reports are clustered in real-time as soon as each report is received [15].

The major issue encountered when performing Document Clustering is to perform effective clustering in an efficient manner [14]. Salton [16] states that more complex linguistic models are being designed to have a better understanding of the syntax and semantics of natural languages in order to perform better clustering. However, such models may be too complex to be feasible to be applied to automatic document analysis. The issue of efficiency is felt more deeply in real time systems. According to Luo [14], new event detection systems always assume that there are enough resources available to perform the necessary computation, and thus they only focus on accuracy and do not give efficiency its proper attention. Cardoso-Cachopo [17] also highlights the importance of efficient clustering especially when operating on large domains like regular news feeds.

The Document Clustering process typically consists of [7]: the extraction of features from the documents; the mapping of the documents to high-dimensional space; and the clustering of the points within the high-dimensional space.

Document features are usually represented by the set or a subset of the words they contain [4,6,7,9–14,16–22]. This approach is also known as the Bag-of-Words approach. Borko [6] advocates the use of pre-selected terms to represent each document. In contrast, [7,9,14,16,18,20,23] extract all the terms from the documents to act as content-representatives—albeit using some filtering sometimes, such as using only the highest-weighted *n* terms [7]. In the majority of cases, stop word lists are used to remove words that occur too frequently (e.g., “the”, “of”, “and”, ...), and suffix stripping routines (such as Porter’s stemming routine [24]) are used to reduce words to their stems [7,14,16,18–20,22,23,25].

The importance of a term as a representative of the document’s content may be calculated using the Inverse Document Frequency (IDF), which is the ratio of the term occurrence frequency within the document in question to the occurrence frequency of the term over the whole document collection [18]. Gulli [12] calculates the term weights by utilising the TF.IDF measure that is centred on the DMOZ (Open Directory Project) Categories. The advantage of this approach is that one does not need to have the entire document collection at hand to weight the documents’ index terms. Wang [19] utilises an incremental version of the TF.IDF, whereby the term frequency of a word is determined based on the number of times the word appears until a particular time, and the number of times this same term appears in the newly appearing stories.

After the documents’ representations have been constructed, the next step would be that of finding the similarities between the different documents, and/or between the documents and clusters. According to Salton [16], the similarity between 2 vectors can be calculated more precisely than by just a simple count of overlapping index terms. Stavrianou [8] defines two types of similarity measures, namely:

Statistical Similarity—based on term and co-occurrence frequencies (e.g., using Cosine Similarity); and Semantic Similarity—using distance between terms meanings (e.g., using WordNet). In most cases, Cosine Similarity measure is used as the distance function [4,4,7,8,10,14,16,17,19,20,23,26].

To enable the calculation of similarity between a document and a cluster, the cluster is usually represented using a centroid vector. The centroid vector is the weighted average of the documents within that cluster [4,7,17,23]. Steinbach [23] also cites the possibility of using median document weights rather than average document weights. In contrast, Cardoso-Cachopo [17] describes a method to calculate the vector centroid based on the Rocchio algorithm. In this method, the centroid vector of a particular cluster is taken to be the sum of all the document vectors for positive training examples for that cluster subtracted by the sum of all the document vectors for negative training examples for that cluster.

According to Ji [3] and Surdeanu [4], similarity measures (or distance functions) are utilised by clustering methods in order to cluster documents in such a way that the intra-cluster similarity (the similarity between the documents within the same cluster) is maximised whilst the inter-cluster similarity is minimised. Steinbach [23] divides clustering techniques into two types, namely:

- **Hierarchical Clustering**—this technique produces hierarchies and is further split into:
  - Agglomerative—whereby we start with each point being in a separate cluster, and at each step, the most similar pair of clusters are merged together (examples of a agglomerative hierarchical clustering algorithms may be found in [4,19]), and
  - Divisive—whereby we start with all the points being in a large single cluster, and at each step we split the cluster in order to maximise the intra-cluster similarity.
- **Partitional Clustering**—whereby one level of partitioning of data points is created, such as the k-means Clustering.

According to the literature reviewed in this research, the most common clustering method used is the k-means Clustering, or a variation of it. In k-means Clustering, we start with k points as the initial cluster centroids, and assign all the points to the nearest centroid. Then, a number of passes are made whereby the cluster centroid is recalculated and the cluster membership of each document is also recomputed [7,12,15,16,23,26].

Stokes [15] adapts the k-means clustering algorithm for First Story Detection such that clustering is performed on incoming documents read from a stream. In this algorithm, each document is compared to existing clusters and is placed into a cluster if the cluster-document similarity exceeds a pre-defined threshold. Otherwise, the document is set to be the seed of a new cluster. Similar methods are also applied in [14,20]. Luo [14] further adapts the k-means clustering algorithm to First Story Detection in a resource-adaptive manner for large-scale processing scenarios. Less important documents are dropped from the main document queue and placed in a lower priority queue whenever the system becomes overloaded. The documents in the lower priority queue are then processed when more resources are available.

The adaptation of clustering algorithms for incremental scenarios presents various problems. Salton [16] mentions that inverted indices used to represent documents are not easily updated. As new documents are added to the clusters with the highest similarity, cluster reorganisation will eventually be needed [16]. In principle, cluster reorganisation should be done as often as possible, however this is computationally expensive [9,16]. One should note that in cases where the vector space model with TF.IDF weighting is used, the cost of the update of the inverted index would be proportional to the current size of the inverted index and not on the size of the update [9]. This is due to the fact that the introduction of a new document into an index affects also the weights of the other documents' terms in the index. Viles [9] found that strict adherence to a term weighting scheme is not required to maintain effectiveness in most situations.

Incremental clustering systems also introduce scalability issues. As new documents keep arriving, the system will eventually run out of memory space to store all these documents. Since the real-time nature of new story detection system requires fast response, it will not be practical to compare the incoming document with all the old documents [14]. One solution to this is to use document retirement [16]. The decision to retire a document can be based on a number of factors such as: the number of citations to that document, the age of the document and/or the usage of the document [16].

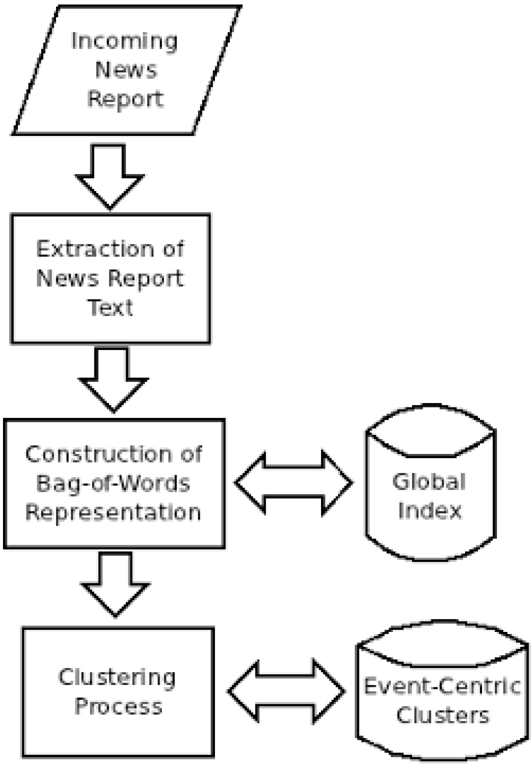
3. Methodology

The main aim within this research is the development of a Document Clustering system that is online and incremental—the system needs to continuously receive news reports, cluster them "on the fly" resulting in the creation of new clusters or the updating of previously existing clusters, and pass these clusters to the other systems as necessary for further processing. Therefore, this system cannot employ complex linguistic models or use multi-pass clustering approaches, as these will result in higher processing times that are not feasible in our operational environment. Also, cluster re-organisation is not possible, since it will result in an interruption of the operational system. The required output is a single level of event-centric clusters—there is no need to have hierarchies of news reports in our system.

The main steps performed by our clustering system are shown in Figure 1. Our clustering system obtains its source news reports via the RSS feeds. The RSS feeds from the selected sources are downloaded periodically and parsed. The relevant news reports are then downloaded, if they had not already been downloaded previously, and cleaned from the surrounding text and HTML code in such a way that only the news report content remains as simple text.

The incoming news reports are then represented in high-dimensional space using the Bag-of-Words model, following the approach taken by the majority of systems reviewed in Section 2. The set of words in each of the news reports are filtered using stop-words lists, and Porter's suffix stemming routine [24] is applied to reduce words to their stems. The reasons for the selection of the Bag-of-Words approach is that it is simple and efficient and does not require any supervision or any other intervention. Moreover, in our opinion, its wide-spread use represents sufficient proof of its effectiveness. The index terms are weighted using the TF.IDF measure. Since the collection of news reports is incremental, the document frequency of each term refers to the number of times that term has appeared in the news report collection until the time that the processing is being done. This is similar to the approach followed in [19]. Our system maintains a global index that stores the occurrence of each stemmed term within the entire news report collection. This global index is updated incrementally as the incoming news reports are being processed. A separate index is also created for each news report being processed storing the occurrence frequencies of the different terms in the corresponding news report. Term weights are not stored but are calculated dynamically when needed.

Figure 1. The Categorisation Process


Incoming News Report
Extraction of News Report Text
Construction of Bag-of-Words Representation
Global Index
Clustering Process
Event-Centric Clusters

For the case where the system has just started processing its first documents, a special procedure is performed for the weighting of those documents' terms. Before starting the categorisation process, the system waits for the first 70 documents to be available for processing, and initialises a global index with the occurrence frequencies of these documents' terms within this initial collection of 70 documents. We monitor 9 RSS feeds (4 from international news sites such as Reuters Top News, and 5 from Maltese news sites), and it takes approximately 12 h to collect 70 breaking news reports from these feeds. In our opinion, an index built from 70 news reports provides ample indication of which terms are important, and which terms are common throughout the entire collection. The clustering of these 70 documents proceeds normally with the sole exception that the global index is not re-updated whilst the clustering of these 70 documents is being performed. When processing the first document, there are no existing categories, and so it is considered to be a "new" event.

Once the news reports have been converted to their corresponding representation, the similarities between the different news reports are calculated as necessary. Each similarity is measured using the distance function that is most widely used within the reviewed document clustering systems—the Cosine Similarity measure. The use of semantic-based similarity measures would not be feasible for our system for the same reasons described before.

Rather than comparing documents (news reports) with each other, our system compares documents with the centroid vectors representing each cluster. This same approach was followed in [4,7,17,23]. The cluster centroid is represented by an index of the stemmed versions of all the terms (excluding stop-words) that appear in those documents which are members of that cluster. The weight of each term within the cluster centroid index is set to be the average weight of that term within the documents in that cluster. More specifically:

$$w_{t,c} = \frac{\sum_{d \in D_c}(w_{t,d})}{|D_c|}$$

where $w_{t,c}$ refers to the weight of term t within cluster c, D_c refers to the collection of documents in cluster c, and $w_{t,d}$ refers to the weight of term t within document d.

In Section 2, we described different clustering techniques, and how these techniques can be divided into Hierarchical Clustering and Partitional Clustering techniques. Since we do not require hierarchies, we did not consider the use of Hierarchical Clustering approaches since they have been reported to be slower than Partitional Clustering approaches [23]. Our clustering component is derived from the k-means Clustering algorithm, which is also used in [7,12,15,16,26], but adapted to work in an incremental environment. The k-means variation used is similar to the ones described in [14,15,20]. In this approach, each incoming document is compared to existing clusters, and if the cluster-document similarity is above a pre-defined threshold (in our setup, this is set to 0.4), the document is placed within that cluster.

In our system, each news report is placed within the first encountered cluster with which it has a similarity that is higher than the pre-defined threshold. If no cluster is found to have a similarity with the document that exceeds the similarity threshold, a new cluster is created with that document as its first member. This is different from the usual approach found in literature where a document is clustered with the cluster with which it has the highest similarity. In our environment, the event-clusters being created are highly specific—they are quite far apart in the vector space. Therefore, generally, each document has a relatively low similarity with unrelated clusters. We also performed some tests to compare the results between our categorisation system and a system where the document is clustered with the most similar threshold, and have found that the difference in results is negligible.

Since our system operates in an incremental environment, the terms' document occurrence frequencies of the terms are updated continuously. According to Salton [16], cluster reorganisation would eventually be needed in such incremental environments. In our system, the news reports are clustered into event clusters. Since events are usually quite distinct from each other, the clusters should be quite distinct from each other, and cluster reorganisation would not be necessary.

During the operation of our clustering system, news reports are being continuously received and clustered. This means that the amount of clusters is constantly growing. For example, in the operational setup of our clustering system that downloads news reports from 9 different news sources, there were 1369 news report clusters created during July 2012, and 10,562 news report clusters were created during 6 months of operation (mid-February till mid August 2012). When a news report describing a new event is received, it is compared with the existing clusters, and when no "similar" cluster is found, it is clustered on its own in a new clusters. Since in such cases, this report must be compared to all the existing clusters prior to being labelled as a breaking event, as more news reports are clustered and new clusters are created, the clustering procedure will become more computationally expensive. Therefore, a system where the number of clusters is continuously growing is not scalable.

To resolve this issue, the concept of “freezing” old clusters. This means that clusters that have not had new members for some time are “frozen”, and incoming documents are not compared to them at all. They are assumed to be describing events whose “influence” has now passed and are not any more of “interest” within the world of news reporting. Apart from aiding the system efficiency, “cluster freezing” is also preventing new news reports to be clustered with obsolete events. The identification of “frozen” clusters is performed by the system, which traverses the list of active (unfrozen) clusters periodically.

Reverting back to the case of our operational setup, the typical number of “active” clusters at any one time is around 135. Considering that in our current system setup, a news report is clustered in slightly less than 1 second, if there is no “freezing” of old clusters, the clustering of a single news report will start to take round 10 s after 1 month of system operation, and nearly 100 s after 6 months of system operation. Such delays may render the clustering system not feasible especially if it is just a small of part of a larger and considerably more complex operational cycle.

4. Evaluation

4.1. Evaluation Methodology

According to Sahoo [13], the clusters produced by a document clustering system can be evaluated in two ways, namely by using qualitative evaluation where the actual clusters are evaluated and quantitative evaluation by demonstrating the benefit of clustering for a particular retrieval task. For our clustering system, we performed qualitative evaluation since we evaluated the quality of the actual clusters. We utilised three different datasets of clustered news reports, and we compared how close the clusters produced by our system are to the clusters in these datasets. We then repeated the evaluation using a baseline system.

The metrics generally used for clustering evaluation are the recall, precision and F-measure—metrics that have been adapted from Information Retrieval. In Information Retrieval, recall is taken to be the fraction of the relevant documents that have been retrieved. Conversely, precision is the fraction of the retrieved documents that are relevant to the query. The F-measure metric is calculated as a weighted average of recall and precision. The use of these metrics in Document Clustering is discussed in [7,10,13,17,19]. We also utilised the recall, precision and F-measure metrics. In our evaluation, the F-measure metric gave equal weight to recall and precision.

For the evaluation of single-label text clustering, where a document may be a member of only one cluster, Ji [3] modifies the Reuters-21578 collection by discarding documents with multiple category labels, and removing clusters with less than 40 documents. Since our clustering system also performs single-label clustering, we discarded those news reports which belonged to more than one category.

We utilised three different corpora for our evaluation. The first corpus is the *Reuters-RCV1* collection—this collection has been used in [13,25] and is the successor of the *Reuters-21578* collection that was used in [3,4,7,17]. The original corpus contains about 810,000 Reuters English Language News stories published between 20th August 1996 and 19th August 1997. The filtered corpus (where the reports belonging to more than one category have been removed) contains 26,310 news reports classified into 78 categories.

The second corpus is the *Yahoo! News* collection—a collection of news reports downloaded via the RSS feeds provided by the *Yahoo! News*. *Yahoo! News* provides RSS feeds for a number of categories that are not so general. Examples of such categories are "*Apple Macintosh*", "*India*" and "*Democratic Party*". This corpus contains 7342 news reports classified into 68 categories. These reports were downloaded in December 2010 and January 2011.

The third corpus is the *Google News* collection—this is a collection of news reports download via the RSS feeds provided by *Google News*. These news reports are clustered according to the event that they describe. Therefore, the clustering in this dataset is much finer than the categories in the other datasets. This dataset is considered the most appropriate for the evaluation of our system, since the aim of our clustering component is to cluster together those news reports that are describing the same event. This dataset contains 1561 news reports downloaded in January 2011 that are covering 205 different news events.

Borko [6], Larsen [7] and Toda [11] evaluate their clustering methods by comparing their automatic clustering results with manual clustering results. Others perform comparisons with other baselines. For example, Hearst [26] compares ranked documents in a best cluster to an equivalent cutoff in the original ranked retrieval results. On the other hand, Stokes [15] compares the presented results with a basic First Story Detection system that uses the traditional vector space model to compute syntactic similarity between documents and clusters.

In our evaluation, we compared the clusters obtained by our system with a baseline system that uses a standard k-means algorithm as described in [23]. In the baseline, the documents were represented using the Bag-of-Words representation—the set of all the terms in each document were considered to be representative of that document. Each set of document terms was filtered from stop-words and had their suffix removed using the Porter's stemming routine [24]. The terms were then represented in vector space using TF.IDF weighting. Document-cluster similarities were found by calculating the cosine similarity between the document and the centroid vector of the relevant cluster (as described in Section 3). The number of clusters to produce was given beforehand to the k-means system, thus giving this system an advantage over our system. In comparison, our system does not need to know the number of clusters to produce beforehand, since it creates new clusters as it deems fit. The following sub-section presents the results obtained on each of the three corpora.

In Section 3, we described how in our system, a document is clustered with the first category it encounters with which it has a similarity higher than the threshold. To test our hypothesis that this does not affect the results greatly, we modified our clustering component to cluster documents with the most similar cluster. This modified system was then used to cluster the *Google News* corpus. The comparison results are also found in Section 4.2.

4.2. Results

Table 1 shows a summary of the results obtained by our system and the baseline system when clustering the *Reuters-RCV1* Corpus.

Table 1. Comparison of Clustering Results using *Reuters-RCV1* Corpus.

	Recall	**Precision**	**FM**
Our System	0.9076	0.0162	0.0230
Baseline System	0.4931	0.0872	0.0937

When calculating recall and precision, for every reference cluster we identified the closest cluster produced by the system being evaluated.

The results obtained by our system show a high recall and a very low precision. The clustering component produced some very large clusters containing a huge number of documents, and the remaining documents were scattered in very small clusters. The recall scores of some categories was 1, or nearly 1. In these cases the produced cluster that is closest to the reference cluster is a very large cluster that encompasses the entire or most of the reference cluster. Since all (or most) of the reference cluster's documents are in that large produced cluster, the recall is 1 (or nearly 1). However, since this closest cluster contains also a large number of documents that do not form part of the reference cluster, the resulting precision is very low. The F-measure, which in our case is the average between the recall and precision, is very low as a result of the very low precision values.

The baseline clustering system produces better results albeit still with very low F-measure values. The incidence of very large clusters, though still present, is lower in the baseline system. Therefore the recall is lower and the precision is higher. The most probable reason for better results is that the baseline system knows the number of clusters to produce beforehand, and also performs cluster reorganisation.

The results obtained by our system and the baseline system when clustering the *Yahoo! News* Corpus are shown in Table 2.

Table 2. Comparison of Clustering Results using *Yahoo! News* Corpus.

	Recall	**Precision**	**FM**
Our System	0.3851	0.2690	0.2159
Baseline System	0.4387	0.3018	0.3091

The results obtained when clustering the *Yahoo! News* corpus, both by our system and also by the baseline system, are significantly better when compared to the results obtained when clustering the *Reuters-RCV1* corpus. There are no more instances of full recall and minimal precision as before, and the F-measure values show a definite increase.

There were a number of cases where the clusters produced by our system had a precision of 1 albeit with a lower recall. A produced cluster with a precision of 1 means that it does not contain any "wrong" documents in it. However, the fact that recall is lower than 1 means that the produced cluster does not contain all the documents in the corresponding reference cluster. Despite the improvement in results, the F-measure values are still quite low—0.2159 for our clustering system, and 0.3091 for the baseline system.

Table 3 presents a summary of the evaluation results obtained when the *Google News* corpus was clustered by three systems: our system; a modification of our system where each document is placed within the most similar cluster; and the baseline system.

Table 3. Comparison of Clustering Results using *Google News* Corpus.

	Recall	**Precision**	**FM**
Our System	0.7737	0.9120	0.8019
Modified System	0.7931	0.9518	0.8370
Baseline System	0.7689	0.5935	0.6188

When viewing Table 3, one can notice that the results obtained by our clustering system when clustering the *Google News* corpus are very good compared to the results obtained by this same system when clustering the *Reuters-RCV1* corpus and *Yahoo! News* corpus. There were many cases where both the recall and precision scores are 1. This means that in these cases, the produced clusters were identical to the reference clusters. There are very few cases where the recall or precision are low. The average F-measure is 0.8019—a huge improvement on the F-measures obtained when clustering the other two corpora (0.2159 for the *Yahoo! News* corpus, and 0.0230 for the *Reuters-RCV1* corpus).

As expected, the results obtained by our modified system, where each document is clustered with the most similar category, are slightly better than the ones obtained by the original system. The difference in results between these two systems is quite low—a difference of 0.035 in F-measure, equivalent to approximately 4%. In our opinion it is worth having a faster system for the cost of 4% lower effectiveness.

The baseline results also contain cases where the produced clusters were identical to the reference clusters. However, on the whole the baseline results are significantly worse than the results obtained by our clustering system.

4.3. Discussion of Results

When analysing the results obtained from the evaluation of our clustering system, one can notice that our system attains very poor results when clustering the *Reuters-RCV1* corpus. On the other hand, it achieves very good results when clustering the *Google News* corpus. When clustering the *Yahoo! News* corpus, it obtains results that are comparable (albeit poorer) to the ones obtained by the baseline system.

The difference in the results when processing these three corpora is due to the difference in the "entropy" of the data corpora. In a data corpus with high entropy (such as *Reuters-RCV1* corpus), there is no clear similarity threshold which can be used to classify the documents into different clusters. When we calculated the similarity of the documents in the *Reuters-RCV1* corpus to their respective clusters, we found that these range from approximately 0.02 to near 0.55 with an average of less than 0.1. On the other hand, when we calculated the similarity of *Reuters-RCV1* clusters with documents that are *not* their members (we refer to this similarity as "dis-similarities"), we found that sometimes it even exceeded 0.3—this is significantly higher than the average similarity of the clusters with their respective documents. Due to this overlap between these "similarities" and "dis-similarities", one cannot establish a clear similarity threshold that would warrant membership in the respective cluster. As a result of this our system obtains poor results. One should keep in mind that the similarity threshold has a crucial importance in our clustering system since it determines when new clusters should be created. On the other hand, the baseline system has the number of clusters provided beforehand and so does not has the problem of creating too many clusters.

The problem of high entropy is also present, albeit to a lower extent, in the *Yahoo! News* corpus. In this corpus, there is still a significant overlap for similarities between clusters and their constituent documents, and between clusters and non-member documents. However, since this overlap is less than in the case of the *Reuters-RCV1* corpus, a similarity threshold can be selected that can be reasonably effective. Consequently, when clustering the *Yahoo! News* corpus, our clustering system can achieve significantly better results than when clustering the *Reuters-RCV1* corpus. However, the results obtained are still lower than the baseline, and therefore, one may conclude that our clustering component is not optimal for the clustering of documents into general categories.

In comparison to the previous two cases, our clustering approach obtains significantly better results than the baseline system when clustering the *Google News* corpus. The reason behind this is that apart from exceptional cases, the similarity between a cluster and a non-member document never exceeds the minimum similarity between a cluster and its member documents. When clustering the *Google News* corpus, our *Document Clustering* component achieves an average recall of 0.7737 and average precision of 0.9120. On the other hand, the baseline system obtains an average recall of 0.7689 and average precision of 0.5935. As one can note, there is a significant difference between these two results.

The results obtained show that our clustering system is very effective in clustering news reports into event-centric clusters. However, its use for the classification of documents or news reports into more general clusters is not recommended. The evaluation results also show that the fact that our clustering component clusters incoming news reports with the first cluster encountered that has an above-threshold similarity rather than with the most similar cluster has minimal cost. This means that our approach in choosing the faster system is justified.

5. Conclusions

In this paper, we have described the design and evaluation of our clustering system. Our clustering system reads incoming news reports from RSS streams, and clusters them "on the fly" according to the event they are describing. The clustering is performed by representing incoming news reports as Bag-of-Words with TF.IDF weighting, and using a variation of the k-means algorithm that works in a single pass without cluster reorganisation. The number of clusters to produce is not known beforehand, and new events are detected automatically. As we mentioned in Section 1, the clustering process described here is very fast—every news report is clustered in less than 1 second on average.

The evaluation results show that our system is very effective when clustering documents into highly specific clusters (such as event-centric clusters), but performs rather poorly when clustering documents (or news reports) into more general categories. The fact that the results obtained on the *Google News* clusters are very good may imply that our algorithm is similar to the one used by *Google News* for clustering. Unfortunately, we could not confirm this since we could not find information about the inner workings of *Google News*.

It is our opinion that our clustering approach can be applied in other domains apart from online news. For example, it can be applied successfully to the clustering of social media feeds to produce clusters according to the item being discussed by the different people.

References

1. Azzopardi, J.; Staff, C. Fusion of News Reports Using Surface-Based Methods. In *WAINA'12: Proceedings of the 2012 26th International Conference on Advanced Information Networking and Applications Workshops*, Fukuoka, Japan, 26–29 March 2012; IEEE Computer Society: Los Alamitos, CA, USA, 2012; pp. 809–814.
2. Azzopardi, J.; Staff, C. Automatic Adaptation and Recommendation of News Reports using Surface-Based Methods. In *PAAMS' 12 (Special Sessions): Proceedings of the 10th International Conference on Practical Applications of Agents and Multi-Agent Systems*, Salamanca, Spain, 28–30 March 2012; Springer-Velag: Berlin/Heidelberg, Germany, 2012; pp. 69–76.
3. Ji, X.; Xu, W. Document Clustering with Prior Knowledge. In *SIGIR' 06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WA, USA, 6–11 August 2006; ACM: New York, NY, USA, 2006; pp. 405–412.
4. Surdeanu, M.; Turmo, J.; Ageno, A. A Hybrid Unsupervised Approach for Document Clustering. In *KDD' 05: Proceedings of the Eleventh ACM SIGKDD International Conference On Knowledge Discovery in Data Mining*, Chicago, IL, USA, 21–24 August 2005; ACM: New York, NY, USA, 2005; pp. 685–690.
5. Kang, B.H.; Kim, Y.S.; Choi, Y.J. Does Multi-User Document Classification Really Help Knowledge Management? In *AI' 07: Proceedings of the 20th Australian Joint Conference on Advances in Artificial Intelligence*, Gold Coast, Australia, 2–6 December 2007; Springer-Verlag: Berlin/Heidelberg, Germany, 2007; pp. 327–336.
6. Borko, H.; Bernick, M. Automatic document classification. *J. ACM* **1963**, *10*, 151–162.
7. Larsen, B.; Aone, C. Fast and Effective Text Mining Using Linear-Time Document Clustering. In *KDD' 99: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, USA, 15–18 August 1999; ACM Press: New York, NY, USA, 1999; pp. 16–22.
8. Stavrianou, A.; Andritsos, P.; Nicoloyannis, N. Overview and semantic issues of text mining. *SIGMOD Rec.* **2007**, *36*, 23–34.
9. Viles, C.L.; French, J.C. On the Update of Term Weights in Dynamic Information Retrieval Systems. In *CIKM' 95: Proceedings of the Fourth International Conference on Information and Knowledge Management*, Baltimore, MD, USA, 29 November–2 December 1995; ACM Press: New York, NY, USA, 1995; pp. 167–174.
10. Aslam, J.; Pelekhov, K.; Rus, D. A Practical Clustering Algorithm for Static and Dynamic Information Organization. In *SODA' 99: Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Baltimore, MD, USA, 17–19 January 1999; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1999; pp. 51–60.

11. Toda, H.; Kataoka, R. A Clustering Method for News Articles Retrieval System. In *WWW' 05: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, Chiba, Japan, 10–14 May 2005; ACM Press: New York, NY, USA, 2005; pp. 988–989.
12. Gulli, A. The Anatomy of a News Search Engine. In *WWW' 05: Proceedings of the Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, Chiba, Japan, 10–14 May 2005; ACM Press: New York, NY, USA, 2005; pp. 880–881.
13. Sahoo, N.; Callan, J.; Krishnan, R.; Duncan, G.; Padman, R. Incremental Hierarchical Clustering of Text Documents. In *CIKM' 06: Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, Arlington, VA, USA, 5–11 November 2006; ACM: New York, NY, USA, 2006; pp. 357–366.
14. Luo, G.; Tang, C.; Yu, P.S. Resource-Adaptive Real-Time New Event Detection. In *SIGMOD' 07: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, Beijing, China, 11–14 June 2007; ACM: New York, NY, USA, 2007; pp. 497–508.
15. Stokes, N.; Carthy, J. First Story Detection Using a Composite Document Representation. In *HLT' 01: Proceedings of the First International Conference on Human Language Technology Research*, San Diego, CA, USA, 18–21 March 2001; Association for Computational Linguistics: Morristown, NJ, USA, 2001; pp. 1–8.
16. Salton, G. Dynamic document processing. *Commun. ACM* **1972**, *15*, 658–668.
17. Cardoso-Cachopo, A.; Oliveira, A.L. Semi-Supervised Single-Label Text Categorization Using Centroid-Based Classifiers. In *SAC' 07: Proceedings of the 2007 ACM Symposium on Applied Computing*, Seoul, Korea, 11–15 March 2007; ACM: New York, NY, USA, 2007; pp. 844–851.
18. Salton, G. A blueprint for automatic indexing. *SIGIR Forum* **1997**, *31*, 23–36.
19. Wang, C.; Zhang, M.; Ma, S.; Ru, L. Automatic Online News Issue Construction in Web Environment. In *WWW' 08: Proceeding of the 17th International Conference on World Wide Web*, Beijing, China, 21–25 April 2008; ACM: New York, NY, USA, 2008; pp. 457–466.
20. Braun, R.K.; Kaneshiro, R. *Exploiting Topic Pragmatics for New Event Detection in tdt-2004, Proc. of Topic Detection and Tracking Workshop*; ACM Press: New York, NY, USA, 2004.
21. McKeown, K.R.; Barzilay, R.; Evans, D.; Hatzivassiloglou, V.; Klavans, J.L.; Nenkova, A.; Sable, C.; Schiffman, B.; Sigelman, S. Tracking and Summarizing News on a Daily Basis with Columbia's Newsblaster. In *HLT' 02: Proceedings of the Human Language Technology Conference*, San Diego, CA, USA, 24–27 March 2002.
22. Arora, R.; Bangalore, P. Text Mining: Classification & Clustering of Articles Related to Sports. In *ACM-SE 43: Proceedings of the 43rd Annual Southeast Regional Conference*, Kennesaw, GA, USA, 18–20 March 2005; ACM: New York, NY, USA, 2005; pp. 153–154.
23. Steinbach, M.; Karypis, G.; Kumar, V. A Comparison of Document Clustering Techniques. In *Proceedings of the KDD Workshop on Text Mining*, Boston, MA, USA, 20–23 August 2000.
24. Porter, M.F. An Algorithm for Suffix Stripping. In *Readings in Information Retrieval*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1997; pp. 313–316.

25. Deng, S.; Peng, H. Document Classification Based on Support Vector Machine Using a Concept Vector Model. In *WI' 06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, Hong Kong, China, 18–22 December 2006; IEEE Computer Society: Washington, DC, USA, 2006; pp. 473–476.
26. Hearst, M.A.; Pedersen, J.O. Reexamining the Cluster Hypothesis: Scatter/gather on Retrieval Results. In *SIGIR' 96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zurich, Switzerland, 18–22 August 1996; ACM Press: New York, NY, USA, 1996; pp. 76–84.