Detecting New and Emerging Events from Textual Sources

Kirk Roberts and Sanda M. Harabagiu
Human Language Technology Research Institute
University of Texas at Dallas
Richardson TX USA 75080
{kirk,sanda}@hlt.utdallas.edu

Abstract—Recognizing new and emerging events in a stream of news documents requires understanding the semantic structure of news reported in natural language. New event detection (NED) is the task of recognizing when a news document discusses a completely novel event. To be successful at this task, we argue a NED method must extract and represent the type of event and its participants as well as the temporal and spatial properties of the event. Our NED methods produce a 25% cost reduction over a bag-of-words baseline and a 13% cost reduction over an existing state-of-the-art approach. Additionally, we discuss our method for recognizing emerging events: the tracking and categorization of unexpected or novel events.

I. INTRODUCTION

Recognizing new events from textual sources and identifying emerging events is a form of tracking information, as opposed to retrieving information, as allowed by search engines. This goes beyond de-duplication (such as the processing done by Google News) as complex events reported in news take place over the course of many days, weeks, or even longer.

New event detection (NED) requires identifying the first reporting of such an event so that it may be tracked in greater detail. For example, one event from TDT5 [1] involves a pod of whales beaching themselves on the Florida Keys on April 18, 2003. The first news article reports the location and number of whales as well as their initial rescue. Two days later, an article reports the seven survivors taking their first swim since rescue. On May 2, one of the whales dies. On July 10, the rescuers ask the government for permission to release the whales, and on July 18 the government agrees. Finally, on August 10, an article reports that the whales have been released at sea. A NED system should mark only the first document (April 18) as being new without ever seeing the subsequent documents.

New event detection requires the ability to reason over complex events. A complex event is an over-arching event that is usually un-stated in full (e.g., the beaching and release of whales on the Florida Keys in mid-2003) that may be reported in varying detail in news documents. We argue there are four primary components of such a complex event: (1) the event type (e.g., beaching and release), (2) the participants (whales and various government agencies), (3) the spatial grounding (the Florida Keys), and (4) the temporal grounding (mid-2003). These four components correspond respectively to the what, who, where, and when of an event description. We use a semantic representation for each of these components. We describe how natural language processing (NLP) techniques are applied to extract textual information from a news document into this representation. Additionally, we use similarity metrics for each of these components that take advantage of their semantic structure. Incorporating these components into a NED system shows significant improvements over both baseline and state-of-the-art approaches.

Emerging events, on the other hand, not only contain a novelty component but have several other components as well: causality, temporality, spatiality. More importantly, emerging events may be redefined as more information becomes available. To be able to detect emerging events we rely on a combination of lexical, semantic, and pragmatic resources as well as the ability to perform event coreference.

The remainder of this paper is organized as follows. Section II outlines previous work in new event detection. Section III discusses the event structure we use to represent events in a news document. Section IV provides similarity metrics to take advantage of this event structure in a semantically robust manner. Section V outlines our experiments and provides analysis about our results on new event detection. Section VI describes our algorithm for detecting emerging events. Finally, Section VII summarizes our conclusions.

II. PREVIOUS WORK

Previous methods to new event detection can generally be classified into two approaches: (1) a linear document comparison approach where new documents are compared to older documents in a pairwise fashion, and (2) a clustering approach where topically similar documents are clustered and new events are detected when a new document does not belong in the pre-existing clusters. Both methods require quality similarity metrics, and thus the second can be seen as an extension of the first.

The generally accepted baseline for NED is TF-IDF (term frequency-inverse document frequency) weighted vector comparison, some form of which is integrated into almost all NED approaches. Beyond this, numerous other similarity metrics have been proposed to complement this baseline to address its key weakness: finding and exploiting the most topically
relevant content within the document. Stokes and Cathy [2] use lexical chains derived from WordNet [3] to correlate topically similar terms (e.g., “airplane”, “pilot”, and “engine” would be found in the same lexical chain). Elsayed et al. [4] employ a character-based n-gram language model. Connell et al. [5] recognize the importance of named entities and give them preferential treatment. Braun and Kaneshiro [6] use a sentence comparison technique as well as a location overlap technique based on the location’s string value. Kumaran and Allan [7] perform a document-level classification based on the official Topic Detection and Tracking (TDT) event types and use different weighting techniques for different types of documents. For example, named entities proved more valuable in legal and science news, while non-named entities proved more valuable in news about elections, accidents, or violence.

Most similar to our approach, Makkonen et al. [8] use locations and temporal references found within the document. However, unlike our approach, they use an ontological path distance for spatial similarity and an interval overlap approach for temporal similarity. We explain why we use different approaches for these classes in Sections IV-C and IV-D. Unfortunately, we lack sufficient details for either approach to allow for fair empirical comparison.

Numerous approaches have been proposed to cluster documents by their event. Clustering reduces the problem of topic drift, where a new document might be similar to an existing document but substantially different from the original event document. For example, an election event would be followed by many documents discussing the results of the election and the issues, candidates, etc. The next election should then be considered a separate event, yet it could look quite similar to the latest documents from the previous election as there will be similar issues, candidates, parties, locations, etc. A clustering approach instead allows for new documents to be compared to a cluster centroid, and can thus avoid topic drift.

Clustering has been utilized in information retrieval for quite some time. Voorhees [9] and Willett [10] both discuss the role of hierarchical clustering for document retrieval tasks. Yang et al. [11] use both single-pass and group-average agglomerative clustering algorithms to segment documents by topic. Trieschnigg and Kraaij [12] perform re-balancing and re-branching on a hierarchical clustering in order to create a more plausible event hierarchy. Zhang et al. [13] use a non-parametric Bayesian method to build probabilistic clusters. Wang et al. [14] use UPGMA (Unweighted Pair Group Method with Arithmetic Mean), which assumes an evolutionary process such as changes in events over time.

Clustering methods, however, have not generally performed well on NED in the official TDT tasks. We suspect this is due to the evaluation method, which evaluates a list of documents ranked by their novelty. Incorrect clustering therefore will introduce greater noise into the novelty ranking than pairwise methods. We therefore follow the pairwise approach, but the similarity metrics we propose should also improve the cluster-based methods since their increased semantic robustness should result in a more precise clustering.

With their troops taking over Iraq, President George W. Bush and wartime ally Prime Minister Tony Blair plan to meet in Northern Ireland next week to discuss the battle against Saddam Hussein’s forces. White House officials said Friday. The officials, who spoke on condition of anonymity, said the meeting near Belfast on Monday and Tuesday also will focus on efforts to bring peace to Northern Ireland and the Middle East. It will be their third face-to-face talk in just over three weeks. The leaders met in the Azores on March 16, along with Spanish Prime Minister Jose Maria Aznar, to announce the end of their diplomatic campaign. Bush and Blair held private talks at Camp David March 27, more than a week after the war began.

The Swedish postal service will issue a special series of stamps to honor slain Foreign Minister Anna Lindh, a spokeswoman said Friday. Four stamps of different denominations will be engraved with Lindh’s portrait and released Nov. 11. Posten spokeswoman Aasa Ivarsson said. The total value of the four-stamp series will be 31 kronor (US$4), but Posten will add a 4 kronor (52 cents) charge that will go to a memorial fund set up in Lindh’s name by her Social Democratic Party. The popular foreign minister was fatally stabbed Sept. 10 while shopping with a friend at a crowded department store in Stockholm. Friday, a court ordered a suspect in her killing to be kept in custody for two weeks while police gather evidence.

| TABLE I | EXAMPLE DOCUMENTS FROM TDT-2004 SHOWING THE AUTOMATICALLY EXTRACTED EVENT MENTIONS (BOLD) AS WELL AS EVENT PARTICIPANTS, TEMPORAL REFERENCES, AND SPATIAL REFERENCES (ITALICS). |

Additionally, methods similar to NED have been developed based on semantic space models (see [15], [16], [17]). These methods detect new events by detecting shifts in a term vector space for a given query. While not directly applicable to NED, they demonstrate the utility of semantic space approaches. Part of our approach utilizes a semantic space method, not to identify shifts in event structures, but rather to classes of events (or event scenarios) derived in an unsupervised fashion from the data.

III. EVENT STRUCTURE

We consider a document-level event in the NED paradigm to be composed of four elements: (1) event mentions that describe the event in natural language, (2) event participants who are the actors in the event, (3) temporal references which ground the event on a timeline, and (4) spatial references which ground the event geographically. Each of these elements requires a separate representation in order to compare events in a semantically robust manner. Table I shows two news documents annotated with their corresponding event structure information.

Additionally, we consider a term vector representation similar to Kumaran and Allan [18] that is capable of capturing information missed by the four items above. Terms are composed of document tokens minus punctuation and stop words. Occasionally terms such as common nouns (e.g., “army”, “storm”, “senator”) and adjectives (e.g., “first”, “ambitious”, “legal”) provide topical clues and are best represented in TF-IDF term vectors.

A. Event Mentions

An event mention is a textual expression of a real-world event. Unlike document-level events, mentions are explicit textual references to an actual event. We use event mentions to represent the type of event that is being discussed in the document. Specifically, we follow the TimeML scheme
for events. TimeML considers events a cover term for situations that happen or occur. For instance, the following sentence contains two separate events, “left” and “attack”:

(1) John left two days before the attack.

As can be seen, TimeML events extend beyond verbs to other syntactic forms. TimeML identifies three syntactic classes of event expressions: (1) tensed verbs (e.g., “has left”, “was captured”), (2) stative adjectives and modifiers (e.g., “sunken”, “stalled”), and (3) event nominals (e.g., “merger”, “Gulf War”). Using TimeML for our event standard is appealing due to the wide range of textual expressions it recognizes. Different documents will refer to an event using different syntactic categories (e.g., “attacked” versus “attack”, “sold” versus “sale”) and TimeML offers a robust method of capturing these various events forms. We use the Evita [20] event extractor (trained on the TimeBank corpus [21]) to discover the event mentions in a news document.

Previous methods would treat the extracted event mentions as a separate term vector to be given a higher weight. But term vectors do not adequately capture the event semantics associated with a document’s event mentions. Instead, we use the event mentions to determine the event scenario [22] that best represents the type of event discussed in the document. Event scenarios are discovered using unsupervised topic modeling techniques such as latent Dirichlet allocation [23] from large numbers of event-annotated documents. Since event scenarios are the result of an unsupervised process, they can easily be built from a document collection in the same domain as the documents in a news stream. Strictly speaking, this unsupervised method learns a probability distribution over a set of event scenarios for each event mention string. A sample of the event scenarios we use in this paper can be seen in Table II. Ten scenarios were generated using the MALLET [24] implementation of latent Dirichlet allocation with the events discovered by Evita. For the event scenario corpus, we use a collection of ten thousand randomly selected documents from the English Gigaword corpus [25]. As can be seen in Table II, the first scenario corresponds to legal events, the second to sports events, the third to financial events, the fourth the violence/war events, and the fifth to political events.

B. Event Participants

The document-level event participants are the actors represented by named entities in the document. We use the Bios [26] named entity tagger, which annotates persons, organizations, locations, temporal expressions, numeric expressions, and miscellaneous named entities. We consider two configurations: (1) all the named entities annotated by Bios, and (2) just the persons, organizations, and miscellaneous entities (which are commonly demonyms such as “Italian” or “Palestinians”), which we refer to as the restricted participant configuration. The intuition behind the second, smaller configuration is that the other entity types are either handled by different elements in our event structure and therefore redundant (locations and temporal expressions) or are not as valuable for new event detection and therefore noisy (numeric expressions). After extraction, the entities are placed in TF-IDF weighted term vectors similar to Kumaran and Allan [18].

C. Temporal References

We consider two types of temporal references: (1) the publication date of the news documents, and (2) the set of temporal textual references within the news document that may be normalized to a specific date. The publication date may be derived according to the date the news article was encountered or through a source-specific timestamp. The textual references within the document, however, must be extracted using automatic methods. Here again, TimeML [19] provides a framework for the extraction and normalization of the necessary textual expressions. TimeML specifies a TIMEX3 annotation that resembles the TIMEX2 standard [27] with minor modification. TIMEX3 allows for a range of temporal specification, including exact dates (e.g., “April 29, 2011”, “1821-07-13”) and centuries (e.g., “19th century”, “1300s”). TIMEX3 can even represent temporal expressions relative to the document’s date (e.g., “now”, “three months ago”, “a year from now”). We use the TARSQI Toolkit [28] to automatically extract TIMEX3 expressions from news documents.

In order to compare temporal expressions in a semantically robust manner, we convert the normalized form to an integer value corresponding to the number of days since 1 A.D., using the half-way point for underspecified time ranges. For instance, “April 2011” becomes the value corresponding to April 15, 2011, while “19th century” becomes the value corresponding to July 1, 1850.

D. Spatial References

Spatial references in news documents serve two purposes: (1) to ground the event geographically, and (2) to ground the participant locations or origins geographically. For example, consider the sentence:

(2) Insurgents blew up a power transformer in a central Iraqi city with a rocket propelled grenade, witnesses said Friday, in the latest of a spiraling series of attacks on U.S.
soldiers and sabotage against the infrastructure needed for Iraq’s reconstruction.

There are three spatial mentions: “central Iraqi city”, “U.S.”, and “Iraq”. The first spatial mention, the unspecified Iraqi city, grounds the event itself, the attack on a power station. The second two spatial mentions ground two participants in the event, namely the United States soldiers and Iraq. Both the event spatial grounding and participant spatial grounding are important for a spatial understanding of the event. Simply representing the event location would fail to distinguish two events with the same participants in a different location. We therefore use all locations identified by Bios in the news document.

Detecting the location mentions is not sufficient to geographically ground an event. Spatial mentions may be both ambiguous (e.g., “Washington” may refer to the State of Washington, Washington D.C., or even a small village in England) and synonymous (e.g., “Washington”, “Washington D.C.”, and “District of Columbia” could all refer to the same location). To normalize spatial mentions and remove ambiguity and synonymy, we use a spatial gazetteer to find geographic locations that the mention may refer to and choose the best gazetteer entry using machine learning. We use the Geonames (www.geonames.org) geographical database to determine possible geographic groundings. When a location has multiple geographic groundings (e.g., “Washington” has 1360 possible matches in Geonames), we use a logistic regression classifier trained on SpatialML [29]. We use three features: the log of the population, the administrative level (e.g., countries get a higher weight than cities, capitals get a higher weight than non-capitals), and whether the spatial mention is an exact or partial match. We determined this method has an accuracy of 92% on a held-out set of SpatialML. Furthermore, this method allows the comparison of locations by geographic distance, which is more semantically robust than a term vector representation. For instance, we can recognize that sentences (3) and (4) are more spatially similar to each other than to sentence (5):

(3) Russian President Vladimir Putin said Friday that Tehran had assured him that their country was not producing nuclear weapons.
(4) Moscow was informed by their Iranian counterparts on Friday that they were not pursuing nuclear technology.
(5) French President Jacques Chirac said Friday that Tehran had assured him that their country was not producing nuclear weapons.

A term vector representation of these sentences would result in sentences (3) and (5) being most similar, because they share the location “Tehran”. However, Iran is more geographically similar to Tehran than France is to Russia, which can be automatically determined if we ground the locations to their spatial coordinates.

IV. Event Similarity

In the NED paradigm, news documents are compared on-the-fly based on chronological document order. Thus for document $n$, there are $n-1$ documents against which the new document must be compared in order to determine if document $n$ discusses a new event. Here we describe how various event structure elements discussed in Section III may be compared so as to conserve their semantic properties. Additionally, it is important that similarity metrics degrade gracefully, a quality that all of the following metrics have. By this we mean that minor changes in a document do not result in large differences in the similarity metric. This is important when weighing multiple similarity metrics (see Section IV-E) as it is possible for a different similarity metric to determine two documents are similar/dissimilar. Thus a similarity metric without this quality will be too noisy to be useful in combination with other similarity metrics.

A. Term and Participant Similarity

Both the document terms and event participants are represented by TF-IDF weighted vectors. However, in the NED paradigm, these TF-IDF vectors cannot be computed using any information about future documents in the news stream. We therefore adopt the Incremental IDF [11] calculation for computing TF-IDF:

$$\text{TF-IDF} = tf \times idf$$

$$tf = \log(\text{termFreq})$$

$$idf = \frac{\log(docCount_R + docCount_S)}{docFreq_R + docFreq_S}$$

Where $\text{termFreq}$ is the frequency of the term in the given document. $docCount_R$ is the total number of documents in an offline repository (we use English Gigaword [25]). $docCount_S$ is the total number of documents encountered thus far in the news stream. $docFreq_R$ is the term’s document frequency in the offline repository. $docFreq_S$ is the term’s document frequency thus far in the news stream. The two term vectors are then compared using cosine similarity:

$$\text{TermSim}(D^a, D^b) = \cosine(D^a, D^b) = \frac{\sum_{i=1}^{w} D^a_i \times D^b_i}{\sqrt{\sum_{i=1}^{w} (D^a_i)^2} \times \sqrt{\sum_{i=1}^{w} (D^b_i)^2}}$$

This technique (Incremental IDF + cosine similarity) is the most common approach in existing NED systems and forms a baseline with which the following similarity metrics can be combined.

B. Event Mention Similarity

As discussed in Section III-A, event mentions are represented by their event scenarios. Each mention in a document has a corresponding distribution of event scenarios. We refer to this distribution of the $n$th event mention in the document by its vector form $e^n_m$. Then $e^n_k$ is the probability of event mention $m$ belonging to scenario $k$. To calculate the overall event scenario distribution for a document $n$ with $M$
event mentions, we use the arithmetic mean of the individual mention vectors for each event scenario $k = 1 \ldots K$:

$$E^m_k = \frac{1}{M} \sum_{m=1}^{M} e^m_k$$

Note that the averaging of probability distributions results in a valid probability distribution. We can now compare two documents by their scenario similarity. Since $E^a_k$ is a probability-weighted vector, we compute the scenario similarity using Jensen-Shannon divergence:

$$\text{ScenarioSim}(D^a, D^b) = JS(E^a, E^b) = \frac{1}{2} KL(E^a, M) + \frac{1}{2} KL(E^b, M)$$

Where $M$ is the average of $E^a$ and $E^b$ and $KL$ is the Kullback-Leibler divergence:

$$KL(E^a, E^b) = \sum_{i=1}^{K} E^a_k \log \left( \frac{E^a_k}{E^b_k} \right)$$

Jensen-Shannon is simply a symmetric extension to Kullback-Leibler. Jensen-Shannon divergence has proved useful in calculating the similarity of two probability distributions in many NLP applications [30].

C. Temporal Similarity

As discussed in Section III-C, our normalized temporal representation is the number of days since 1 A.D. for a given temporal mention $t^m$. This allows for a straight-forward document similarity using difference-of-means similarity:

$$\text{TemporalSim}(D^a, D^b) = \frac{1}{[\text{AvgDate}(D^a) - \text{AvgDate}(D^b)] + 1}$$

Where $\text{AvgDate}(D^a)$ is the mean of the normalized temporal expressions (tnorm) in document $D^a$:

$$\text{AvgDate}(D^a) = \frac{1}{M} \sum_{m=1}^{M} \text{tnorm}(t^m)$$

We chose this difference-of-means similarity instead of an interval-based approach similar to Makkonen et al. [8] because of its increased robustness. Each date is equally weighted, so references to years are not given higher preference than exact date references. This enables our similarity metric to degrade gracefully as temporal granularity decreases, an important quality when weighing multiple similarity metrics. Additionally, an interval cover method fails to capture when similar dates with no overlap are used. This is a common occurrence with NED, as follow-up events that are still considered part of the primary event will occur on later date. An interval cover method will give a similarity of zero for these cases, whereas our method will give a high similarity score.

D. Spatial Similarity

Our spatial normalization from Section III-D determines the geographic coordinates of a location mention $s^m$. The spatial similarity that we use is the pairwise average of the geographic distances between mentions:

$$\text{SpatialSim}(D^a, D^b) = \frac{1}{M^a M^b} \sum_{i=1}^{M^a} \sum_{j=1}^{M^b} \text{dist}(s^{a,i}, s^{b,j})$$

Where $M^a$ is the number of spatial mentions in document $a$, and $s^{a,i}$ is the $i$th spatial mention in document $a$. The distance function $\text{dist}$ is the geographic distance using Vincenty’s formula [31].

We chose this method over previous approaches due to its semantic robustness. A string-based representation [6] would consider “Russia” equally dissimilar to “Tehran” as “Iran”, a problem solved by both geographic distance and ontologies. But ontology-based similarity [8] assumes all geographic locations are equally spread apart according to their hierarchical structure, so an event with locations on either side of a national border would be farther apart than two locations on opposite sides of a country.

E. Combining Similarity Metrics

Several methods have been used to combine similarity metrics for NED, usually involving a linear combination of the similarity metrics. Stokes and Carthy [2] combine two similarity metrics using an un-weighted scheme (i.e., each similarity metric was given equal weight). Makkonen et al. [32] use a Perceptron learning method to determine a weighting scheme. We follow Kumaran and Allan [18] and apply a support vector machine (SVM) to combine the similarity metrics listed above. Specifically, we use $\text{SVM}^{light}$ [33] to develop our SVM models. While Kumaran and Allan [18] show slight improvements can be achieved with non-linear kernels, we use a linear kernel to demonstrate the improvement from our similarity metrics over existing methods.

Instead of detecting a “new” event, we train the SVM to detect when two documents discuss the same event. Not only does this yield significantly more positive training instances (hundreds of thousands instead of a few hundred), but fits well within the NED paradigm. Each similarity metric described above becomes a new real-valued feature within this SVM model. The value of the feature is then the similarity of the two documents being considered according to that similarity metric. At test time, each news document is visited in chronological order and is compared pair-wise to every preceding document. If every pair-wise decision results in the SVM classifier returning a result of “different”, then the document is determined to represent a new event. Since this is an $O(n^2)$ procedure, we reduce the number of (computationally expensive) SVM-based comparisons by employing a heuristic method to detect when two documents have almost no chance of discussing the same event. If there is no overlap among the top ten words (by TF-IDF) in the two documents, we classify them as different. Otherwise, the two documents are classified
by the SVM model to determine their similarity. Experiments with the impact of this heuristic range from slight degradation to slight improvement of end-to-end performance.

V. NED EXPERIMENTS

A. Data

To evaluate our methods for new event detection we use the TDT5 corpus [1], which contains 9,812 topic-annotated documents from 250 topics. TDT5 contains 13 general categories of topics, shown in Table III. The documents are time-stamped by their date of publication and source (e.g., “NYT20030726.0000.0005” was published by the New York Times on July 26, 2003). In order to train our algorithm, we partition this data into a training set of 150 topics and a test set of 100 topics, yielding 7,681 training documents and 2,184 test documents. Thus, performing NED on this dataset requires finding the 100 documents within the 2,184 test documents that discuss a new event.

B. Evaluation

We performed a series of experiments to compare the efficacy of our similarity metrics with pre-existing methods. We use two evaluation methods for comparing results.

The official TDT metric [1] is known as the detection cost,

\[ C_{Det} = C_{Miss} \cdot P_{Miss} \cdot P_{target} + C_{FA} \cdot P_{FA} \cdot P_{non-target} \]

Where \( C_{Miss} \) and \( C_{FA} \) are costs associated with a missed new event and a false alarm (incorrectly guessed new event). TDT [1] defines these as 1.0 and 0.1, respectively, for NED. \( P_{Miss} \) and \( P_{FA} \) are the conditional probabilities for a miss and false alarm, respectively. \( P_{target} \) and \( P_{non-target} \) are the \textit{a priori} target probabilities, pre-defined as 0.02 and 0.98, respectively. To create a more intuitive score, \( C_{Det} \) is normalized to be at most one:

\[
(C_{Det})_{Norm} = \frac{C_{Det}}{\min(C_{Miss} \cdot P_{target}, C_{FA} \cdot P_{non-target})}
\]

Where \( C_{Miss} \cdot P_{target} \) is the dummy baseline of guessing nothing is a new event, and \( C_{FA} \cdot P_{non-target} \) is the dummy baseline of guessing everything is a new event. Because \( C_{Det} \) is a cost measure, better results are indicated by lower (closer to zero) scores.

Approaches are judged on their \textit{optimal} \( C_{Det} \) value. The output of a NED method is then a scored list of documents, where a threshold can be used to assign all documents above the threshold as new events, and all documents below the threshold as old events. The optimal \( C_{Det} \) is then simply the \( C_{Det} \) for the best corresponding threshold. Therefore, systems are not evaluated based on their ability to determine the best threshold, but rather their ability to rank documents by their likelihood of discussing a new event. The threshold itself could be considered an application-specific parameter (i.e., some applications may wish a strict or relaxed threshold).

Additionally, we include a more traditional information retrieval metric, average precision, which does not require any threshold-finding. Average precision combines both precision and recall for a ranked list of documents:

\[
AvgPrec = \frac{\sum_{r=1}^{N} Prec(r) \times rel(r)}{\# \text{of new events}}
\]

Where \( Prec(r) \) is the list’s precision for the first \( r \) documents; \( rel(r) = 1 \) if the document at rank \( r \) is a new event and 0 otherwise.

C. Results

The results of our experiments are shown in Table IV. We performed an additive test to determine which similarity metrics improve the baseline term similarity. The single most valuable similarity beyond the baseline was the participant metric, which uses TF-IDF vectors similar to the term similarity. After that, the spatial similarity, event scenario similarity, and document date similarity also reduced the detection cost. In total, these four additional similarity metrics, when combined with the term similarity in an SVM, reduced the baseline detection cost over 25% from 0.7855 to 0.5859.

The two similarity metrics that were unable to reduce the detection cost were the restricted participant similarity (persons, organizations, and miscellaneous named entities) and the temporal references within the document itself. However, both of these provide improvement over the baseline. As can be seen in Table IV, the restricted participant similarity improves NED even after the full participant similarity is included. This suggests that there is value to be gained by considering these entity classes separately, but either a different feature representation or similarity combination is necessary. Second, while the full temporal reference similarity is not part of the optimal configuration, it improves the baseline by 5.5%. Upon examination, we found that this similarity metric suffers when significant background information is given in the document (e.g., when a reference is made to the 18th century to help explain a current event). This distorts our difference-of-means similarity metric, but would likely prove equally problematic with an interval overlap metric such as Makkonen et al. [8]. Furthermore, given that news documents generally report events immediately after they happen, the document timestamp may be a nearly sufficient temporal representation of the event. We therefore leave the issue of handling these extraneous temporal references to future work.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
\textbf{Category} & \textbf{Count} \\
\hline
Elections & 10 \\
Scandals/Hearings & 3 \\
Legal/Criminal Cases & 30 \\
Natural Disasters & 11 \\
Accidents & 25 \\
Acts of Violence or War & 36 \\
Science and Discovery News & 11 \\
Financial News & 3 \\
New Laws & 4 \\
Sports News & 13 \\
Political and Diplomatic Meetings & 34 \\
Celebrity and Human Interest News & 22 \\
Miscellaneous News & 48 \\
\hline
\end{tabular}
\caption{TDT5 Topic Types}
\end{table}
TABLE IV

<table>
<thead>
<tr>
<th>Term/Participant model</th>
<th>Event Scenario</th>
<th>Spatial</th>
<th>Temporal (doc-date)</th>
<th>Temporal (refs)</th>
<th>C_{Det}</th>
<th>AvgPRec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event recall</td>
<td>.6750</td>
<td>.5602</td>
<td>.7016</td>
<td>.7522</td>
<td>.6984</td>
<td>.3377</td>
</tr>
<tr>
<td>Event restricted</td>
<td>.6984</td>
<td>.3477</td>
<td>.6984</td>
<td>.3477</td>
<td>.7424</td>
<td>.3158</td>
</tr>
<tr>
<td>Event scenario</td>
<td>.6502</td>
<td>.5625</td>
<td>.6593</td>
<td>.3749</td>
<td>.6154</td>
<td>.3771</td>
</tr>
<tr>
<td>Term participant</td>
<td>.6005</td>
<td>.3773</td>
<td>.5883</td>
<td>.3910</td>
<td>.6134</td>
<td>.3773</td>
</tr>
<tr>
<td>Term participant</td>
<td>.6529</td>
<td>.3479</td>
<td>.5859</td>
<td>.3911</td>
<td>.6250</td>
<td>.3619</td>
</tr>
<tr>
<td>Participant (restricted)</td>
<td>.5872</td>
<td>.3895</td>
<td>.5859</td>
<td>.3911</td>
<td>.6225</td>
<td>.3620</td>
</tr>
<tr>
<td>Participant (all)</td>
<td>.6050</td>
<td>.3700</td>
<td>.5872</td>
<td>.3895</td>
<td>.6225</td>
<td>.3620</td>
</tr>
</tbody>
</table>

Given the similarity metrics that do improve new event detection, we highlight two contributions made by this paper. First, a spatial representation based on geographic coordinates (instead of strings [6] or ontologies [8]) provides a sound semantic representation with significant flexibility, evidenced by its reduction of the detection cost by a 11.1% improvement over the baseline and a 8.6% improvement over an approach that considers only terms and named entities. Second, an event scenario representation provides a compact form for representing event types. Initially, this provides one of the least useful features, improving the baseline only 4.2%, even less than the temporal references. But after the participant and spatial similarity metrics are added, the event scenario similarity is able to reduce detection cost by a further 4.4%. Thus this type of similarity is not only highly complementary to our other similarity metrics, but should prove complementary to many other methods as well.

Unfortunately, as our results require part of the only dataset we have available for training, we are unable to compare our method to pre-existing results on the entire TDT5 dataset. However, the term/participant model, which achieves a C_{Det} of 0.6730, is largely similar to the method reported in Kumar and Allan [18], missing only a topic term feature (see [7]) that is both difficult to replicate and not generalizable beyond the TDT data. Given the significant improvements made beyond the term/participant model, we believe the similarity metrics presented here to be a valuable contribution to new event detection methods.

VI. IDENTIFYING EMERGING EVENTS

Some of the new events will also indicate the emergence of unexpected processes, states, or other series of events. Jurgens and Stevens [17] describe how the launch of a toy during the holiday season may lead to the emergence of a toy recall and eventual lawsuits due to toxicity associated with the toy. Eventually, the launch of the toy is just a new event, but later reports of toxic chemicals used in its manufacturing may lead to the development of this *emerging event*. In this case, emergence is defined by the shift or deviation from the “normal” structure of events associated with launching a product. Jurgens and Stevens associate the emergence with a topic shift, detected on a temporal ordering of events. In an attempt to define the semantics of emergence, we have considered a model that uses: (1) event scenarios; (2) event coreference; (3) scalar implicatures, to detect unexpected variations of scale, (4) temporal inference as well as (5) causal inference between events; and (6) lexico-semantic indicators of “emergence”.

Event scenarios, as described in Bejan [22], establish possible links between event mentions. These links correspond to temporal and causal relations between event mentions. Temporal relations between events are detected by an algorithm for projecting events on a timeline, as described in Bejan [34]. This algorithm builds a graphical representation of events and temporal expressions identified in textual sources and infers temporal relations between events. To be able to detect causal relations between events, we rely only on discourse-level causality. For this reason, we (a) parse text into discourse units, as described in Soricut and Marcu [35], and (b) identify causal relations, as described in Marcus and Echihabi [36]. However, most of the time temporal and causal relations hold between mentions of the same event. To be able to generate the links in the emerging event scenarios, we first perform event coreference, using the system reported in [37]. Then, for every pair of events, if non-contradictory temporal or causal relations between their mentions are detected, those relations are included in the emerging event structure.

Finally, we mine lexico-semantic patterns of emergence to identify events that are unexpected. We utilize two strategies: (1) generating temporal and causal decisions from the emerging event structures in a newswire corpus; and (2) mining lexemes whose WordNet glosses contain causal markers (e.g., the gloss for a manufacturing recall event is “cause to be returned”; the gloss for a disseminate event is “cause to become widely known”). These lexico-semantic patterns are manually pruned to remove common event types that do not indicate a true emerging event (e.g., a political candidate distributing a flyer stating her stance on various issues).

In identifying emerging events, we also try to capture the “novel components” of the event as well as the “unusual” or “disruptive” components. Whereas these components are best captured by analogical reasoning, we use scalar implicatures to capture violations of expectation. Of special interest are scalar triggers that indicate exceptional size, e.g., “large numbers
Algorithm 1 Identifying Emerging Events

1. Generate event scenarios.
2. Recognize temporal and causal relations between events.
3. Perform event coreference within and across documents.
4. Reconcile relations among events.
5. Identify events that satisfy lexico-semantic patterns.
6. Perform scalar implicatures and other novel component recognition techniques.

of casualties” or “flooding of epic proportions”. Additionally, lexical clues can be utilized for topics of interest. For example, we recognize when positive financial news for a company (e.g., “stocks rose”, “profits”) shifts to negative financial news (e.g., “bankruptcy”).

The steps for processing and identifying emerging events are summarized in Algorithm 1. Unfortunately, no data exists on which we may evaluate our methods and tune our model of emergence. While the individual components are evaluated in the reported papers, we are investigating methods for creating datasets that identify important and unexpected events within news streams.

VII. CONCLUSION

We have described an approach for the detection of new events in text streams. The approach considers the event mentions, event participants, temporal references, and spatial references within a news document. We have presented robust similarity metrics which preserve the semantics of the document’s event structure. This method achieves a 25% cost reduction over the baseline and a 13% cost reduction over a state-of-the-art system. Additionally, we have presented our method for recognizing emerging events, which combines several NLP methods such as event coreference and discourse parsing to find unexpected and novel events as they emerge.

REFERENCES