First Story Detection using Entities and Relations

Nikolaos Panagiotou
National and Kapodistrian
University of Athens
npanagio@di.uoa.gr

Cem Akkaya
Yahoo! Research,
Sunnyvale, CA
cakkaya@yahoo-inc.com

Kostas Tsioutsiouliklis
Yahoo! Research,
Sunnyvale, CA
kostas@yahoo-inc.com

Vana Kalogeraki
Athens Univerisity of
Economics and Business
vana@aueb.gr

Dimitrios Gunopulos
National and Kapodistrian
University of Athens
dg@di.uoa.gr

Abstract

News portals, such as Yahoo News or Google News, collect large amounts of documents from a variety of sources on a daily basis. Only a small portion of these documents can be selected and displayed on the homepage. Thus, there is a strong preference for major, recent events. In this work, we propose a scalable and accurate First Story Detection (FSD) pipeline that identifies fresh news. In comparison to other FSD systems, our method relies on relation extraction methods exploiting entities and their relations. We evaluate our pipeline using two distinct datasets from Yahoo News and Google News. Experimental results demonstrate that our method improves over the state-of-the-art systems on both datasets with constant space and time requirements.

1 Introduction

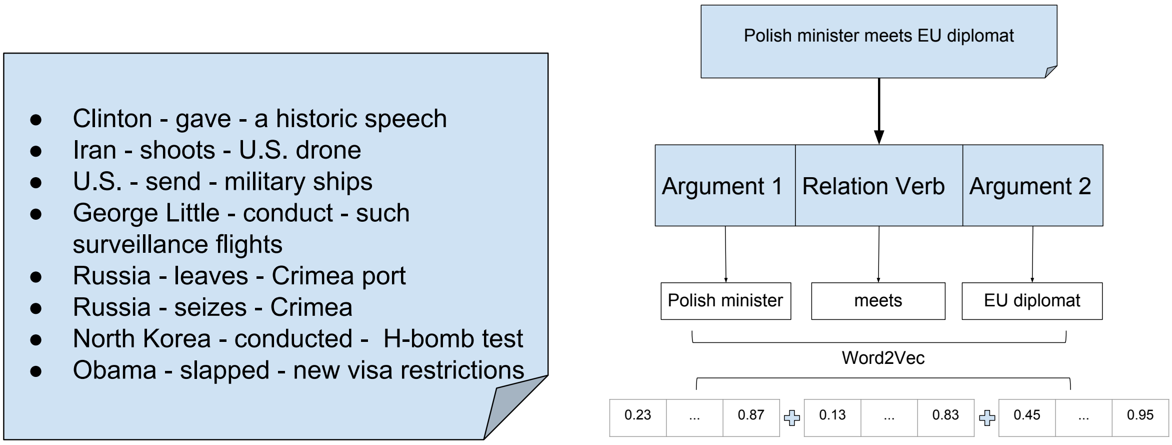
First Story Detection (FSD) is the task of detecting the first document about a new event given a stream of documents (Allan et al., 1998; Allan et al., 2000a). The task is also known as New Event Detection (NED). The problem appears in several real-world applications where news stories are accumulated and presented to users in near real-time. A FSD system should be accurate, scalable, and process a stream of articles in a single pass. The output of such a system is very valuable to news portals such as Yahoo News, Google News, since the rapid detection of a new event is crucial for the service reputation. FSD is a very challenging, if not impossible, task for human operators, since it requires inspecting millions of documents per day. As a result, accurate, automatic solutions are very desirable.

The majority of the FSD systems in the literature attempt to classify a document as a first story if the document differs significantly from those published before and thus may describe a new event. This is accomplished usually in two steps. In the first step, the nearest neighbor of a new document in the previous document stream is identified. In the second step, the similarity between the new document and its nearest neighbor is considered in order to decide if it is a first story or not. In this methodology, the selection of an appropriate similarity metric and the selection of an informative document representation are essential. As a result many different metrics have been studied in the literature, such as the cosine similarity (Petrović et al., 2012), the KL-Divergence (Karkali et al., 2013), and the named entities overlap (Kumaran and Allan, 2004). So far, terms and entities occurring in a document have been the main representation units, boiling down the task to detecting documents with novel terms and entities in a stream.

We believe that existing approaches under-utilize the semantic information present in news articles, specifically, actions performed by entities, or interactions that happened between a pair of entities. These actions and interactions are essential, since they describe events, around which a news story revolves. Our hypothesis is that often the same entities and terms may appear in documents that describe different events – sometimes related – leading to false negatives. For example, when North Korea conducted the hydrogen bomb test in January of 2016, the U.N. security council called for an emergency meeting. This resulted in two different events in news streams in the following order: (1) North Korea conducted the H-bomb test, (2) the U.N. announced a meeting about North Korea. The articles about the two events have a very similar term and entity distribution. Nevertheless, the extracted relations about North Korea and the U.N. differ in the two story lines. For the first story line, relations such as "N.Korea - concerns - U.N." are detected while for the second story line relations such as "U.N. - announces - meeting about North Korea" are present.

Motivated by this observation, we propose to utilize relations between entities when deciding if an article should be classified as a first story. For this purpose, we define a relation as an entity-action-entity triplet (see Figure 1a)

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3237–3244, Osaka, Japan, December 11-17 2016.


Clinton - gave - a historic speech
Iran - shoots - U.S. drone
U.S. - send - military ships
George Little - conduct - such surveillance flights
Russia - leaves - Crimea port
Russia - seizes - Crimea
North Korea - conducted - H-bomb test
Obama - slapped - new visa restrictions
Polish minister meets EU diplomat
Argument 1
Relation Verb
Argument 2
Polish minister
meets
EU diplomat
Word2Vec
0.23
...
0.87
0.13
...
0.83
0.45
...
0.95

(a) (b)

Figure 1: (a) Relations extracted from different news articles. (b) Our proposed relation representation.

describing events and sub-events of a story. In addition, we rely on a distributed representation to represent our relation triplets in order to overcome lexical variation.

To the best of our knowledge, our approach is the first one to integrate relations between entities in a FSD system. Our experiments demonstrate that our system improves the detection error trade-of (DET) on a Yahoo News dataset by up to 29.6% compared to the best unsupervised system and up to 17.3% compared to the best supervised system. At the same time, we show that the space and time requirements remain constant over time suggesting that the method is suitable for very high volume streams.

The contributions of this work can be summarized as follows:

- We build an efficient, stream-based pipeline for detecting fresh news that uses traditional term similarity metrics amplified by relation and entity similarity information
- We propose a novel approach to model a document as a set of named entities and their relations.
- We make annotation and preprocessing tools as well as preprocessed datasets available [1].

The rest of the paper is organized as follows. In Section 2, we briefly describe related work on FSD, as well as recent advances in relation extraction. Section 3 follows with the description of our system. First, we provide details of a basic scalable FSD system. Then, we describe our proposed entity-relation document model and present our system in detail. In Section 4, we describe the evaluation procedure, the datasets, and the systems we compared with. In Section 5, we provide experimental results and analysis. Section 6, concludes the paper.

2 Related Work

First Story Detection has been extensively studied in the literature. One of the best-performing FSD systems proposed was UMASS (Allan et al., 2000b). This system retrieves the nearest neighbour of a new document. Then the system calculates a novelty score for the new document as the cosine distance, using incremental TF-IDF weighted vectors, to its nearest neighbor. This score is used to decide if it is a first story or not. In a work proposed by (Stokes and Carthy, 2001), the authors use two distinct document representations when searching for the nearest neighbor while on (Brants et al., 2003) the authors use different TF-IDF models per document category during the search. All these approaches leverage primarily statistical information about terms found in a document.

The same principle was extended in (Kumaran and Allan, 2005) where the authors incorporated information about entities, topics and also used a supervised classifier to perform the detection. In this work, we also use features that exploit the named entities. However, we extend the entity usage by capturing also the way that the entities interact.

The above approaches are not designed to scale with massive streams. Efficient first story detection was recently studied by (Petrović et al., 2010) where the authors proposed a constant time and space solution. They redesign UMASS (Allan et al., 2000b) utilizing a locality sensitive hashing (LSH) multi-index and apply the system on the Twitter stream. We deploy the same algorithm in our system for scalability. The (Petrović et al., 2010) system was also implemented in (McCreadie et al., 2013) as a storm topology that is able to process the Twitter Firehose in

[1] Supplementary paper material available at: `https://bitbucket.org/npan1990/firststory-annotator`

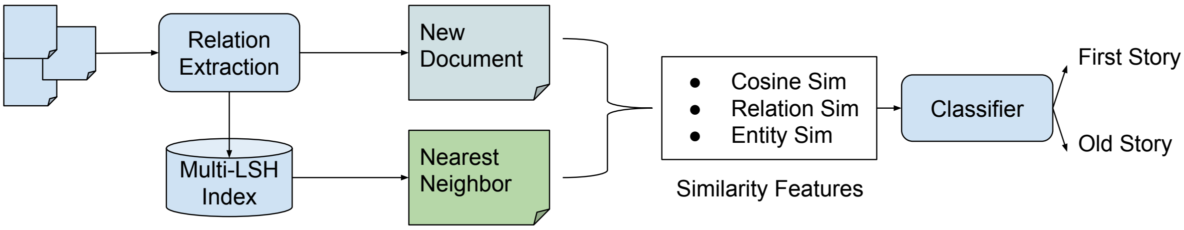


Figure 2: The First Story Detection pipeline we propose.

real-time using 70 processing units according to the author claims. On the same direction, the authors in (Karkali et al., 2013) develop an efficient approach that completely avoids the nearest neighbor identification by defining the novelty of a document as the novelty of its terms.

In addition, (Petrović et al., 2012) provides an extension of the system described in (Petrović et al., 2010) that addresses the synonymy problem by expanding the term vectors with paraphrases. The new system yielded a 13% improvement in detection error trade-of without significantly increasing the computational complexity. In our work we also address the synonymy problem, but from a different perspective through Word2Vec (Mikolov et al., 2013).

The method that we propose highly depends on a robust relation extraction mechanism. Since we are interested in generic relations independent of a predefined taxonomy, we rely on an open information extraction system. These systems detect open domain relations by self-training over a massive corpus (Banko et al., 2007) or heuristic rules (Fader et al., 2011; Etzioni et al., 2011; Schmitz et al., 2012). They allow the development of very scalable systems. The relations extracted often have a generic format of two arguments that are connected by a verb (see Figure 1a). In our work, we use OpenIE 4.1 (Etzioni et al., 2011) the successor of ReVerb (Fader et al., 2011), Ollie (Schmitz et al., 2012) and TextRunner (Banko et al., 2007).

3 The Proposed First Story Detection Pipeline

We design and implement a novel pipeline to solve the FSD task. In this section, we describe the main aspects of our pipeline. We begin by describing the basic approach and a scalable extension that uses locality sensitive hashing in Section 3.1. Then, in Section 3.2, we describe a holistic document representation based on relations and a similarity function that compares documents using this representation. Finally, in Section 3.3, we present how we identify first stories using various similarity features and a supervised classifier.

3.1 A Basic First Story Detection Pipeline

A generic pipeline for detecting first stories consists of two basic steps. In the first step, as soon as a new document arrives, the nearest neighbor is identified. In the second step, a similarity function is considered in order to measure how similar the new document is to its nearest neighbor and decide if it is a new story. This basic design was first instantiated by the UMASS system described in (Allan et al., 2000b). It uses cosine similarity and incremental TF-IDF document vectors and achieved state-of-the-art performance during the topic detection and tracking challenge (Fiscus and Doddington, 2002). Subsequently it was improved along two axes: (i) improved scalability by exploiting approximate nearest neighbor techniques, proposed by (Petrović et al., 2010), that use locality sensitive hashing, (ii) improved accuracy by addressing the synonymy problem through exploiting syntactic paraphrases (Petrović et al., 2012).

Our approach follows the basic First Story Detection paradigm. The main novel aspects of the pipeline we propose are: (i) New features are extracted improving the resolution of the similarity function. These features exploit the **relations** between the entities that appear in a document and we show how they can be obtained and incorporated in a novel similarity measure between documents. (ii) The various similarity features extracted are given to a supervised classifier that learns how to combine them, and decides if a new document is a new story. Our complete First Story Detection pipeline is illustrated in Figure 2.

3.2 Entity-Relation Document Representation

In this section, we describe how we model the entities and their relations in a document. In addition, we propose Relation Similarity (RelSim), a metric for comparing two documents in terms of their named entities and their relations. Our technique uses state-of-the-art relation extraction algorithms that extract the relationship between two arguments in a 3-tuple format (see Figure 1a).

Simplifying the Relations

The relation arguments are n-grams and in many cases consist of large text chunks or even sub-clauses. However, these large text chunks add noise to the relations and rarely reappear on other documents making relation comparison a difficult task. For that reason, we employ a set of simplification heuristics in order to convert large relation arguments to small n-grams. Here is a summary of the rules:

- We require the first relation argument to contain a simple named entity. If it does, then the argument is replaced by the named entity.
- We require that the second relation argument is not a sub-clause. If it is we remove the relation.
- If the second argument contains a named entity the argument is replaced by the named entity. If it contains more than one named entities the relation is split into multiple relations.
- From the second relation argument we keep only the nouns and the adjectives.
- From the relationship verbs we keep only the core verb that expresses the action. Modals and auxiliary verbs are removed.

During the relation extraction, we also extract the named entities along with their types[2]. We keep only the named entities of type *Person*, *Location*, and *Organization*. In many cases, we observed that an entity was mentioned explicitly only once in the text, and then was referenced implicitly through pronouns in later sentences. So, after we have identified the entities and their types, we propagate them to the following sentences while replacing the pronouns (e.g. He-met-Putin translates to Obama-met-Putin).

Relation Representation

The UMASS system represents the documents as TF-IDF weighted vectors. However, in order to compare documents in terms of their relations we need a relation-oriented representation.

Definition 1 Relation: *A relation is defined as a 3-tuple* $r = (arg1, action, arg2)$. arg1 *is the main entity or actor of the relation.* arg2 *is the recipient of that action (e.g. Putin - meets - Obama) or a preposition that describes the action (Obama - landed - Thursday). The action is a simple verb.*

Definition 2 Bag of Relations: *The relation set* $R_d = \{r_1, \ldots, r_{|R_d|}\}$ *for a document d.*

In many cases the actor of a relation could have multiple textual representations or surface forms (e.g. Obama, Barack Obama, President of US). In addition, multiple relation verbs could express the same action. For example, the relations $r_1 = (A, proposes, X)$ and $r_2 = (A, suggests, X)$ have probably the same semantic meaning. Thus, we decided to exploit Word2Vec, a technique described in (Mikolov et al., 2013) that learns a vector representation for words. All the three relation parts were converted into their corresponding vectors. If the representation of a n-gram (e.g. "U.S. President") was not directly available we used the average vectors of the n unigrams. The resulting three vectors are concatenated to a large vector. The concatenated vector of a relation $r_1 = (A, meets, X)$ will be different than the concatenated vector of the relation $r_2 = (X, meets, A)$. Under this technique, similar relations will have a similar vector representation addressing this way the synonymy problem. The procedure is also illustrated in Figure 1b.

Comparing documents using relations

So far we have described how to represent a document as a set of simplified relations. However, it is unclear how to compare two documents using their relations. Thus, we propose Relation Similarity (RelSim), a metric that compares semantically two documents using their named entities and relations.

Assume that we have two documents d_1 and d_2 that we need to compare, each with its relations R_{d1} and R_{d2}, respectively. For each relation $r_i \in R_{d1}$ the method $simRD(r_i, d_2)$ returns a relation score rs_i that is equal to the cosine similarity with the most similar relation $r_j \in R_{d2}$. While searching for the most similar relation from d_2, $SimRD(r_i, d_2)$ also checks the distance to the inverse of the relation r_i. This decision was taken since in some cases, a relation r_i may be expressed in a reverse form on the document we compare. For example, the inverse relation of $r_i = (Obama, meets, Putin)$ is the relation $r'_i = (Putin, meets, Obama)$. The method $SimRD(r_i, d_2)$ is defined on Equation 1.

The document to document similarity $SimDD(d_1, d_2)$ is the average relation score rs_i for every relation $r_i \in R_{d1}$ with the document d_2 and is defined on Equation 2. Since $SimDD(d_1, d_2)$ is not a symmetric function it is not suitable for a similarity metric. The final relation similarity $RelSim(d_1, d_2)$ defined on Equation 3 is the metric we use to compare two documents in terms of their relations. It is important to note that the computational complexity of RelSim is $O(|R_{d1}| * |R_{d2}|)$. However, the relations identified per document are on average only 10 with a standard deviation of 9. Thus, the computational complexity of RelSim won't affect the system scalability.

$$SimRD(r_i, d_2) = \max_{r_j \in R_{d2}} (max(Cos(r_i, r_j), Cos(r'_i, r_j))) \tag{1}$$

$$SimDD(d_1, d_2) = \frac{sum_{r_i \in R_{d1}}(SimRD(r_i, d_2))}{|R_{d1}|} \tag{2}$$

[2] We used the Stanford CoreNLP tool available at: `http://stanfordnlp.github.io/CoreNLP/ner.html`

Feature	LSH-RelFSD	LSH-RelEntFSD
$CosSim(d, d_n)$	✓	✓
$RelSim(d, d_n)$	✓	✓
$EntOverlap(d, d_n)$	✗	✓
$RelEntOverlap(d, d_n)$	✗	✓

Table 1: The similarity features between d and d_n used by our methods.

$$RelSim(d_1, d_2) = \frac{SimDD(d_1, d_2) + SimDD(d_2, d_1)}{2} \tag{3}$$

3.3 Entity-Relation FSD

We only focused on extracting relations from news articles, simplifying them and using them as a distance metric for document comparison. Having these ingredients, in this section we describe how to use the relation similarity (*RelSim*) discussed above in order to perform first story detection. We present two supervised approaches, *LSH-RelFSD* and *LSH-RelEntFSD*, that address the FSD task as a binary classification problem. The first uses as features the cosine similarity and the relation similarity between and new document d and its nearest neighbor d_n. The latter uses also entity similarity features. The classifier we use, since the method is supervised, is a Logistic Regression. The features used for a new document d and its nearest neighbor d_n are presented on Table 1. CosSim is the term cosine similarity employed also by UMASS. RelSim is the similarity function described in the previous section. EntOverlap is the overlap of the entities that appear on the documents and RelEntOverlap is the overlap of the entities present in the relations.

4 Evaluation Setup

In this section we describe our evaluation framework. We provide details about the datasets, the annotation process, the system configuration, and the pipeline parameters.

4.1 Datasets

We decided to evaluate the scalability and accuracy of our FSD pipeline on three datasets: (1) A Yahoo News dataset (D1) under the category “Politics and Government” that we created using an event-guided annotation procedure described in (Petrović et al., 2012). In order to simplify the annotation process for the dataset (D1) we implemented a user-friendly web interface, which allows the easy labelling of the documents in terms of events. We make the tool available[3]. This dataset consists of 652 documents, 89 are first stories and 563 are non first stories. (2) A Google News dataset (D2) that was proposed in (Karkali et al., 2013). This dataset consists of 2006 documents about “Technology” where 1491 are annotated as first stories and 515 are annotated as non first stories. Clearly, the datasets (D1) and (D2) have different label distributions. (3) A much larger synthetic dataset (D3) of $106,000$ documents from Yahoo News is used in order to evaluate the scalability of our method. However, this dataset is not annotated and thus is not used for evaluating the accuracy.

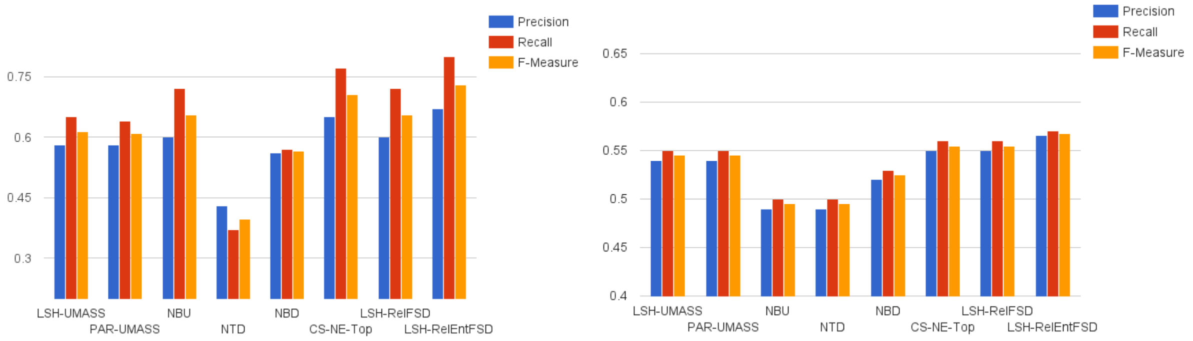
4.2 System Instantiation

In order to compare our system with existing state-of-the-art systems, we implemented to the best of our ability the following systems: (i) LSH-UMASS by (Petrović et al., 2010) and (ii) PAR-UMASS by (Petrović et al., 2012). (Karkali et al., 2013) define several measures to evaluate the novelty of a document’s terms; we use here the ones they suggest that perform best for the content, namely (iii) NTD, (iv) NBD, (v) NBU. (vi) CS-NE-Top suggested by (Kumaran and Allan, 2005). LSH-UMASS, PAR-UMASS and CS-NE-Top are similar to our method in that they also initially detect the nearest neighbor of a new document. LSH-UMASS uses the incremental TF-IDF weighted term vectors while PAR-UMASS expands the vectors according to a pool of syntactic paraphrases. CS-NE-Top uses the similarity of the term, the entity and the non-entity (topic) vectors and similarly to our technique uses a classifier in order to combine the multiple similarity features.

LSH-UMASS, LSH-RelFSD and LSH-RelEntFSD use locality sensitive hashing as described in (Petrović et al., 2010), to efficiently identify the nearest neighbor. The required multi-LSH index parameters were set to $K = 5$ and $L = 20$ which result in missing a nearest neighbor of distance 0.3 with probability $\delta = 0.025$. The maximum LSH bucket size was set to 1000. For the PAR-UMASS system paraphrases available online[4] with Precision more than 0.4 are used. For NBU, NTD, and NBD we set the required parameter N to 60.

[3] https://bitbucket.org/npan1990/firststory-annotator

[4] http://paraphrase.org/#/download



(a) Dataset (D1)

(b) Dataset (D2)

Figure 3: FSD systems comparison in terms of Precision, Recall and F-Measure.

4.3 System Description

We preprocessed our dataset (D3) on a large commercial Hadoop Cluster. The preprocessing was implemented via a Pig script in order to effectively allocate the required resources and run the Map-Reduce job. The job took less than 2 hours to extract the necessary metadata for the January 2016 dataset (D3). Datasets (D1) and (D2) are much smaller than (D3), and the preprocessing was done on a single machine. For evaluating our pipeline accuracy and performance we used a 4-core Intel i7 CPU with 32GB of RAM.

4.4 Evaluation Metrics

In order to evaluate the FSD task we used Precision, Recall, and F-Measure. Datasets (D1) and (D2) are fully labeled, so it is possible to calculate the above metrics averaged over the two classes. Also, a commonly used metric for FSD is the detection error trade-off (DET) score. DET score is defined in Equation 4 and depends on the probability of missing a first story P_{miss} and the probability of a false alarm P_{fa} for a specific threshold τ. For each system we find the threshold τ that minimizes P_{fa} and P_{miss} and achieves the lowest DET score DET_{min}. The costs of a false alarm and missing a first story C_{fa} and C_{miss} are set to 1.0, P_{target} is set to 0.5 similarly to (Karkali et al., 2013). All the evaluation metrics are calculated under 5-fold stratified cross validation.

$$DET = C_{miss} * P_{miss} * P_{target} + C_{fa} * P_{fa} * (1 - P_{target}) \tag{4}$$

5 Experimental Results and Discussion

5.1 First Story Detection Performance

In Figure 3a, we present the performance of various FSD systems on the Yahoo News dataset (D1). Clearly, the systems that incorporate entity or relation information have a significant advantage on this dataset. LSH-RelEntFSD achieves a F-Measure of .73 and CS-NE-Top, which comes second, achieves a F-Measure of .70. LSH-RelFSD and NBU systems that follow on this dataset achieve a F-Measure of .65. In terms of Precision, LSH-RelEntFSD and CS-NE-Top report .67 and .64 while their Recall is .81 and .77 respectively. The rest of the systems score F-Measure values between .40 and .61.

In Figure 3b, we present the performance on the Google News dataset (D2). LSH-RelEntFSD achieves the best performance with a F-Measure of .57. CS-NE-Top and LSH-RelFSD that follow both report a F-Measure of .55. The systems LSH-UMASS and PAR-UMASS that come next achieve a F-Measure of .54. The Precision and the Recall for LSH-RelEntFSD is $\sim$ .57. The remaining systems report F-Measure values up to .52.

The results for the DET_{min} evaluation metric are shown in Table 2. The best performance on both datasets is achieved by LSH-RelEntFSD and CS-NE-Top. On the Yahoo news dataset (D1) our LSH-RelEntFSD system achieves a 17.3% improvement over the state-of-art supervised system CS-NE-Top and a 29.6% improvement over the best unsupervised system NBU. On the Google news dataset (D2), LSH-RelEntFSD achieves a 4.4% improvement in the DET_{min} score over the best unsupervised system.

The improvement in Det_{min} score is statistically significant for the dataset (D1) at the $p < 0.05$ level using a paired t-test. However, for dataset (D2) all methods perform close to the best method (LSH-RelEntFSD) and so the results are not statistically significant. To understand our results in (D2) we explored the nature of this dataset in more detail and discovered that the dataset contains many non-news articles, such as product descriptions, reviews as well as personal opinions. The impact is two-fold: firstly, incorporating entity and relation information on these articles is not as important as in dataset (D1) about "Politics and Government" where many named entities

Method	$Det_{min}(D1)$	$Det_{min}(D2)$
LSH-UMASS	.35	.45
PAR-UMASS	.36	.45
NBU	.27	.5
NTD	.62	.5
NBD	.42	.47
CS-NE-Top	.23	.44
LSH-RelFSD	.27	.44
LSH-RelEntFSD	**.19**	**.43**

Table 2: Det_{min} scores for the Yahoo (D1) and Google (D2) datasets.

participate and interact. In addition, we discovered that for several of the articles in (D2), in particular among those that are reviews, descriptions or opinions, there is significant ambiguity whether they should have been classified as new stories or not. This clearly impacts the performance of all the techniques.

Our proposed pipeline outperforms all the methods used on both datasets. This strengthens the conclusion that a linear classifier that combines multiple similarity features is essential in order to effectively address the task at hand. The supervised system CS-NE-Top is the closest competitor reporting similar performance as LSH-RelEntFSD on dataset (D2) while LSH-RelEntFSD outperforms CS-NE-Top on dataset (D1) by 17.3%. This improvement results mainly from taking into account the entity relations in addition to the entity and term similarity features.

5.2 Space and Time requirements

Since our system should be able to process hundreds of thousands or even millions of documents per day we ensure that the algorithm's time and space requirements do not increase as the stream progresses. Figure 4 illustrates the processing time required per document on the large Yahoo News dataset (D3). Clearly, it shows that the processing time per document remains steady over time and it is less than $20ms$ on average. This result suggests that the processing time per document does not grow with the stream time making the method suitable for high volume streams. The four large spikes on the figure are caused due to the memory allocation and cleaning operations performed by the Java virtual machine. In addition, the memory usage is also steady and did not exceed $18GB$ of which $7GB$ were required by the Word2Vec model.

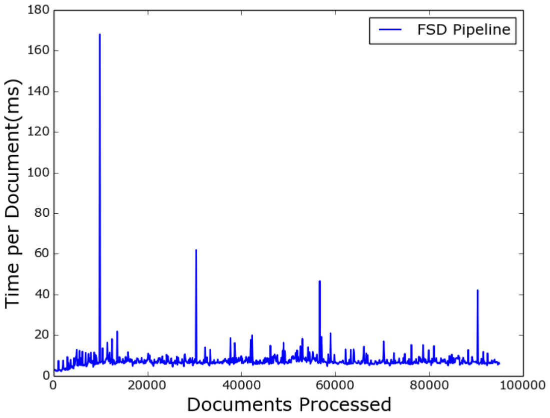


Figure 4: Processing time per document.

6 Conclusions

In this work, we proposed a scalable first story detection pipeline that exploits relations between entities in order to deduce the freshness of a document. To the best of our knowledge, our approach is the first one to integrate relations between entities in a FSD system. We proposed a novel document representation based on relation triplets and described a similarity function on that representation. The positive results on two datasets provide evidence that incorporating relation information helps to detect new events. Another advantage of our method is that it addresses indirectly the synonymy problem through the usage of Word2Vec in the relation representation. Finally, we demonstrated that our system is scalable with constant space and time requirements by investigating the behaviour on a large dataset.

Acknowledgments.

This research has been financed by the European Union through the FP7 ERC IDEAS 308019 NGHCS project, the Horizon2020 688380 VaVeL project and a Yahoo Faculty Research & Engagement Program.

References

James Allan, Jaime G Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study final report.

James Allan, Victor Lavrenko, and Hubert Jin. 2000a. First story detection in tdt is hard. In *Proc of the ninth international conference on Information and knowledge management*, pages 374–381. ACM.

James Allan, Victor Lavrenko, Daniella Malin, and Russell Swan. 2000b. Detections, bounds, and timelines: Umass and tdt-3. In *Proc of topic detection and tracking workshop*, pages 167–174.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676.

Thorsten Brants, Francine Chen, and Ayman Farahat. 2003. A system for new event detection. In *Proc of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 330–337. ACM.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. 2011. Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proc of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

Jonathan G Fiscus and George R Doddington. 2002. Topic detection and tracking evaluation overview. *Topic detection and tracking*, pages 17–31.

Margarita Karkali, François Rousseau, Alexandros Ntoulas, and Michalis Vazirgiannis. 2013. Efficient online novelty detection in news streams. In *Web Information Systems Engineering–WISE 2013*, pages 57–71. Springer.

Giridhar Kumaran and James Allan. 2004. Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 297–304. ACM.

Giridhar Kumaran and James Allan. 2005. Using names and topics for new event detection. In *Proc of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 121–128. Association for Computational Linguistics.

Richard McCreadie, Craig Macdonald, Iadh Ounis, Miles Osborne, and Slobodan Petrovic. 2013. Scalable distributed event detection for twitter. In *Big Data, 2013 IEEE Int Conference on*, pages 543–549. IEEE.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189. Association for Computational Linguistics.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2012. Using paraphrases for improving first story detection in news and twitter. In *Proc of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 338–346. Association for Computational Linguistics.

Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, et al. 2012. Open language learning for information extraction. In *Proc of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics.

Nicola Stokes and Joe Carthy. 2001. Combining semantic and syntactic document classifiers to improve first story detection. In *Proc of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 424–425. ACM.