Structured Event Entity Resolution in Humanitarian Domains

Mayank Kejriwal(✉), Jing Peng, Haotian Zhang, and Pedro Szekely

USC Information Sciences Institute, Marina Del Rey, USA
kejriwal@isi.edu

Abstract. In domains such as humanitarian assistance and disaster relief (HADR), events, rather than named entities, are the primary focus of analysts and aid officials. An important problem that must be solved to provide situational awareness to aid providers is automatic clustering of sub-events that refer to the same underlying event. An effective solution to the problem requires judicious use of both domain-specific and semantic information, as well as statistical methods like deep neural embeddings. In this paper, we present an approach, AugSEER (Augmented feature sets for Structured Event Entity Resolution), that combines advances in deep neural embeddings both on text and graph data with minimally supervised inputs from domain experts. AugSEER can operate in both online and batch scenarios. On five real-world HADR datasets, AugSEER is found, on average, to outperform the next best baseline result by almost 15% on the cluster purity metric and by 3% on the F1-Measure metric. In contrast, text-based approaches are found to perform poorly, demonstrating the importance of semantic information in devising a good solution. We also use sub-event clustering visualizations to illustrate the qualitative potential of AugSEER.

Keywords: Events · Structured event resolution
Hybrid embeddings · Crisis informatics
Humanitarian and disaster relief · Clustering

1 Introduction

As the devastating consequences of recent disasters such as Hurricanes Irma and Harvey illustrate, effective mobilizing of resources and personnel is an important problem, with technology playing an increasingly important role, both in taking preventive action (e.g., evacuations) and dealing with the disaster's aftermath [6], [8]. The impact of disasters, and other events with a humanitarian dimension, is global: according to the 2016 Human Development report [7], conflicts, disasters and natural resources constitute key global concerns, with more than 21.3 million people (roughly the population of Australia) being affected by the refugee crisis alone. Technology can play an important role in alleviating this suffering by equipping HADR analysts with *situational awareness* [20]. Situational awareness

D. Vrandečić et al. (Eds.): ISWC 2018, LNCS 11136, pp. 233–249, 2018.
https://doi.org/10.1007/978-3-030-00671-6_14

is a broad notion, involving analytics that can cover text, sentiments, entities and spatio-temporal information. Examples include *entity-centric search* and aggregate sentiment analyses that help pinpoint emerging hotspots [10]. In some cases, *posthoc analysis* also needs to be conducted, perhaps by performing batch analytics on newswire or social media collected over a time interval.

For a continuously deployed system to conduct even basic *event-centric* analysis, at global scales of space and over arbitrary periods of time, the *structured event entity resolution* (SEER) problem needs to be solved. Along with named entities, HADR ontologies (whether simple or complex), also include *event entities* as first-class citizens. Event entities tend to be semi-structured objects that are sometimes extracted from documents, but (in the HADR space) can also be entire document fragments. This is especially the case when considering heterogeneous corpora such as specialized newswire (e.g. an article describing a single incident or event), social media and SMS. Events can span multiple days, week or in some cases (such as the Syrian refugee crisis), years. For posthoc analysis (the batch mode), users input their own heterogeneous corpus, usually collected over a multi-year period of time, and desire semi-automatic non-overlapping event clustering as a first step. In this sense, each data item is a 'sub-event', and a collection of sub-events represent a 'resolved' event.

Adequately solving the SEER problem involves several challenges not completely addressed by modern or classic text classification and clustering approaches. First, in addition to being relatively robust to errors, a good SEER system must handle the *topical flux* (more generally, called *concept drift*) that an evolving event exhibits across documents, space and time, often in unprecedented ways. As an example, consider the case of the Haiti earthquake in 2010. In an initial set of documents describing this disaster, the topics were primarily along the lines of earthquakes and landslides. In later documents, the key issues were humanitarian aid, politics and an unfortunate Cholera outbreak due to waste mismanagement by rescuers. Experiments described later show that topic modeling methods (or more recently, document embeddings) yield poor performance by themselves as they are not able to deduce that all of these circumstances relate to the same situation, namely a localized disaster in Haiti that has its origins in the earthquake.

The case above suggests that, barring large quantities of training data, a multi-pronged i.e. *statistical-semantic* approach may be necessary to address the SEER problem. In this paper, we present AugSEER, an approach that can judiciously accommodate both domain expertise and recent advances in neural representation learning to respond to users in online and batch modes. AugSEER is continuously running and minimally supervised. It interfaces directly both with a Neon engine that powers an interactive GUI, and with a NoSQL database that stores a knowledge graph of both named and event entities, (translated and original) texts, and NLP analytics such as sentiment analysis (Fig. 1). The GUI and the overall system (called THOR[1]) is already undergoing user studies with

[1] Text-enabled Humanitarian Operations in Real-time.

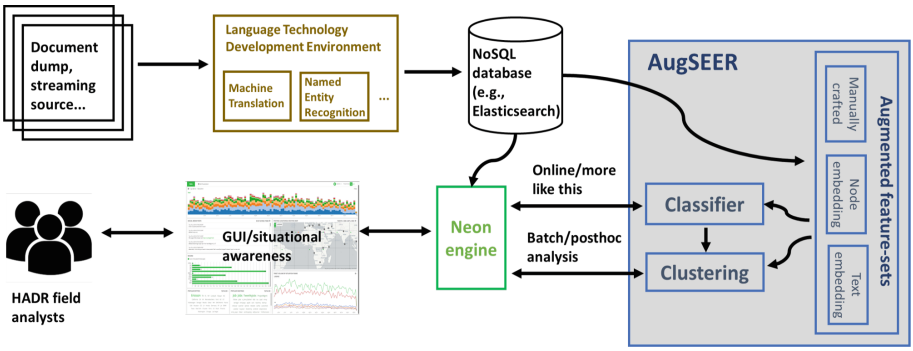

Fig. 1. A schematic of the overall HADR situational awareness system (THOR) within which AugSEER (the focus in this paper) is embedded.

real-world analysts, and is able to incorporate NLP outputs from independent state-of-the-art systems.

Contributions. We introduce and model the Structured Event Entity Resolution (SEER) problem, motivated by rapid mobilization of resources in the HADR domain. To the best of our knowledge, SEER is a difficult, socially consequential AI challenge not addressed by existing work. Second, we present AugSEER, which uses a hybrid combination of feature sets, both manually defined and automatically constructed using neural vector space embeddings, to address the SEER problem in both online and batch modes. AugSEER supports the online *more like this* mode by framing the SEER problem as a probabilistic binary classification task. To support the batch setting (e.g., for posthoc analyses), AugSEER uses a combination of classification and spectral clustering. AugSEER is also *minimally supervised*, being able to achieve reasonably accurate results using 30% (or fewer) training labels. To the best of our knowledge, this is the first application to demonstrate empirical utility from combining feature subspaces in a manner that has not been attempted in prior work on neural embeddings. Third, we rigorously evaluate multiple aspects of AugSEER on five HADR datasets encompassing diverse events, using clustering and classification metrics in tandem with visualizations.

2 Related Work

Feature embeddings have become popular in the AI and knowledge discovery communities in recent years, with vector space embeddings developed for words, sentences, documents, nodes in networks and graphs, particularly knowledge graphs, along with embeddings of the entire graph itself. Many recent models either adapt or extend the skip-gram model, used first for word2vec [13], or in the case of knowledge graph embeddings, surveyed by [21], use hand-crafted energy functions to optimize performance on applications such as triples ranking.

Other similar kinds of graph embeddings have also been proposed in the broader community (see [1] for a recent synthesis).

Our work is different from the above for several reasons. First, none of the embedding papers cited above attempt to combine manual features with graph and text-based feature embeddings in an effort to improve performance as well as allow the domain expert (in an unusual domain like HADR) to exert a level of control over the machine learning process. In general, AI research in the HADR domain has been limited; far more attention has been paid instead to good *data management* techniques [6], [8]. As [8] describe, only a handful of free systems exist for powerful HADR analytics, and none cover the SEER problem. Examples of specific work in HADR, but with much narrower scope than this paper, include 'social sensing' of earthquakes [18], and location extraction [9], both on Twitter data. To the best of our knowledge, no existing HADR system has fully leveraged recent advances in neural embeddings.

Second, existing work on entity resolution and linking is typically limited to resolving *atomic* entities like persons or organizations [4]. In contrast, we are attempting to resolve an entire event, which is a complex data structure with auxiliary information sets like words and entities. To the best of our knowledge, this is the first paper that presents a minimally supervised approach for addressing the SEER problem in a socially consequential domain like HADR.

We also note that, in contrast with graph-theoretic communities, the NLP community majorly focuses on text-centric techniques for a similar problem, namely *event co-reference* resolution [15], [11]. Events in the NLP community tend to be strictly typed according to a shallow schema, and are extracted from documents with corresponding information such as actors and dates. In contrast, our techniques make no such assumptions, since they are unrealistic in HADR. For example, a news article may discuss an event several hours or days after it strikes, while social media could be instantaneous. Often, location information is not available, and many document fragments that our approach takes as input may not even be 'events' in the NLP sense. Most importantly, we are clustering entire semi-structured objects, and not just sentences or triggers that are embedded within a larger textual context. This makes the problem more challenging, and as we describe later, text-only methods perform poorly in many cases.

3 Structured Event Entity Resolution (SEER)

We assume a set of *situation frames*, where a situation frame is intuitively defined as the finest-grained unit of data collected in that HADR problem domain. A situation frame may include such artifacts as SMS messages, intelligence fragments or even social media. Many NLP tasks are performed at the level of situation frames, following which the outputs (such as named entities) are used to enrich the situation frame further. A simple, but representative illustration, of this enrichment and the various artifacts involved, can be seen in Fig. 2. In

particular, the situation frame is itself part of an *event ontology*, which captures the core elements of the analyses[2].

Given a set of situation frames, the SEER problem can be defined as inferring (whether automatic or not) *Same Event* relationships between situation frames. The *Same Event* relationship is currently assumed to have equivalence class (i.e. reflexive, symmetric and transitive) semantics, although future work may relax the transitivity assumption. Given these assumptions, each connected component (in the knowledge sub-graph where situation frames are nodes, and edges exist between frames if they are part of the same event) is called a resolved event cluster. The ultimate goal of a *batch* SEER system is to recover such clusters from a given dump of situation frames. In an (alternative) *online* setting, also called *more like this*, users (typically interactively) select a single situation frame as query, sometimes preceded by keyword search, and desire related frames that provide more insight into the broader event described by the query.

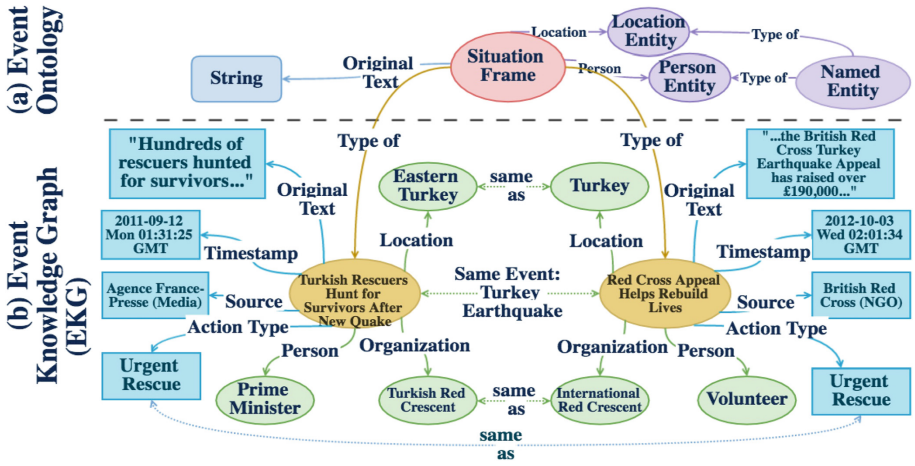

Fig. 2. A schematic illustrating the key representational details of the event ontology and event knowledge graph (EKG) for supporting solutions to the SEER problem.

While the online and batch modes are related, there are several challenges in solving either one. First, raw HADR frames are not only highly heterogeneous in terms of information content and quality, but in low-resource regions of the world (where such technology would have maximal impact), come in a computationally under-studied language like Uighyur [5]. As a first step, machine translation (MT) algorithms have to be executed to automatically translate the text into English [19]. The resulting translated text is noisy, because MT algorithms for such languages are not as well developed as for English. Next, because

[2] Although not fully described herein, the ontology is quite rich in practice, and includes inferential elements like sentiments and offsets (for extraction provenance).

named entities are important, both for SEER and for situational analytics, a named entity recognition system has to be executed [3], following which, *duplicate* named entities have to be *resolved*. However, highly accurate, automatic *entity resolution* [4] is far from solved, despite decades of research from the AI community.

Finally, because each disaster event tends to be *unique* compared to other disaster events, building a representative training set, and automating the solution completely using static machine learning modules, is also difficult. An example illustrating how text, topics and entities are collectively important, but can also naïvely interact to give misleading results, is in the case of the earthquake in Turkey in 2011. Around the time the earthquake struck Turkey, the country was also dealing with the Syrian refugee crisis. Frames describing either crisis tended to have similar statistical, entity and word profiles. For example, aid agencies, like the UN, or governmental entities like the Turkish army, were common to both crises. In the next section, we describe AugSEER, which is an approach that attempts to capture the important interactions between various situation frame attributes that can lead to accurate *Same Event* inference even when distinctions are fine-grained.

4 Approach

We note that the clustering in SEER is challenging (and different from ordinary non-semantic clustering) precisely because of the arbitrary scales of time and space involved, since at such scales, multiple, unrelated disasters are present in the corpus. In the example we described earlier, the earthquake that hit Turkey in 2011 was contemporaneous with the (still ongoing) Syrian refugee crisis. Also, not every disaster is consequential enough to make international headlines, or is in an English-speaking region. An important HADR problem is to generate meaningful results even in *low-resource*, *minimally supervised* environments. Ideally, an analyst would like to obtain robust situational awareness on each HADR-relevant event in such an environment with little technical expertise.

In order to learn good representations for addressing the HADR-specific challenges of the SEER problem, AugSEER relies heavily on an *augmented feature set* that relies on recent advances in latent space embedding models (both for text and graphs [2], [1]) as well as on a small set of similarity features that captures the intuitions of domain experts. More details are provided below.

Manually Crafted Features. Domain experts, who have studied the HADR problem over several years, understand that the text alone does not adequately convey all relevant information about an event to statistical methods. Instead, one must also rely on auxiliary *information sets*, such as extracted entities. Based on initial data exploration and feature engineering, we devised ten real-valued feature functions (Table 1), where each feature function is a similarity function that applies to some information set of a *pair* (D_1, D_2) of situation frames.

We consider three similarity functions, namely cosine similarity on TFIDF, cosine similarity on latent space embeddings derived using the paragraph2Vec

Table 1. Manually crafted feature descriptions. Each feature is computed on a pair of situation frames (D_1 and D_2).

Name	Description
$TFIDF_{\{W,E\}}$	The respective cosine similarities based on bag-of-words and bag-of-entities TFIDF representations of the text fields of D_1 and D_2
$TFIDF_{avg}$	The average of $TFIDF_W$ and $TFIDF_E$
$DV_{\{W,E,avg\}}$	Same as above, except using text embedding (rather than TFIDF) representations [2].
$JAC_{\{L,O,P\}}$	The Jaccard similarity between the {location, organization, person} extracted entity sets of D_1 and D_2
JAC_{all}	The Jaccard similarity between the set of all entities extracted from D_1 and D_2

algorithm [2], and Jaccard similarity. We consider two information sets, namely the set of entities extracted from each frame, and the tokens in the text. In the case of Jaccard similarity, we do not consider the text as an information set, but we do consider finer-grained sets like *differently typed* entities. Descriptions are provided in Table 1.

Importantly, unlike the (subsequently described) node embedding and text embedding features, the manually crafted features are computed for the texts in each *pair* of situation frames. This makes the features inherently more suited to the *more like this* online setting than to the clustering setting, to which their application and scalability is not obvious.

Node Embedding Features. Entities play an important role in the HADR domain, as many key events revolve around a specific set of persons, locations and organizations, some of which might be *latent* (i.e. not explicitly mentioned in the text). On the other hand, some entities might be wrongly extracted or typed due to imperfections in the underlying extraction system. Features relevant to explicitly extracted entities can be captured by the manual features. However, those features cannot capture latent information, and are also not good at distinguishing which entities might prove to be more important to the problem at hand. Instead, to capture the special nature of entities, we construct an *undirected entity-SF bipartite graph* from the corpus by (1) assigning a unique node in the *situation frame (SF) layer* to each frame D, and (2) assigning a unique node in the *entity layer* to the pair (E, T), where T is the type (e.g. person) of an entity E extracted from the text of at least one situation frame. Edges in this bipartite network are created by linking an entity node to each frame node from which the entity was extracted.

Next, we execute a model inspired by the skip-gram based DeepWalk algorithm on the constructed network to obtain an embedding for each situation frame and each entity [17]. DeepWalk was originally designed for learning node representations in unweighted social networks like YouTube and Flickr. In this paper, we use its philosophy for learning entity-centric frame embeddings by

first sampling nodes from the bipartite graph and initiating a constant number of random walks from the node; and then treating each random walk like a list of tokens that can be embedded using skip-gram. More details on the skip-gram model and on DeepWalk may be found in the respective papers [13], [17]. We denote the entity-centric node embedding of D, obtained through the procedure described above, as $\boldsymbol{D_N}$ (boldface indicating vectorization). Note that, because of connectivity and co-occurrence information about extracted entities across the corpus, entities that have not been explicitly extracted can also influence $\boldsymbol{D_N}$ owing to the continuous representations learned by DeepWalk in a dense real-valued vector space.

Text Embedding Features. Finally, to capture statistical signals in the text, we use skip-gram based document embeddings (also called Paragraph2Vec or PV) first described in [2]. Specifically, we tokenize the machine-translated (if in a foreign language) text of a situation frame using a standard set of delimiters, convert all words to lower-case, and feed each list of tokens to the PV algorithm. For a frame D, we denote the text embedding feature vector as $\boldsymbol{D_T}$. These embeddings are also used in computing DV features in Table 1 for frame pairs.

4.1 Classification and Clustering

AugSEER supports the SEER problem both in batch and online settings. The latter is a pairing problem, whereby a domain expert uses the system in a *more like this* manner by first specifying a situation frame as input and then expecting the system to retrieve other situation frames (possibly with other constraints specified in the GUI, like keywords or entities, but not discussed herein) that refer to the same underlying event. In AugSEER, we frame this as a *probabilistic binary classification problem* on pairs of frames, whereby the pair should have higher probability of a positive label if they represent two sub-events resolving to the same underlying event.

In a supervised setting, given a labeled set of positive and negative pairs, we construct an augmented feature vector for a pair (D_1, D_2) by (1) computing the ten manual features on the pair, (2) concatenating the node embedding feature vectors of D_1 and D_2, and (3) concatenating the text embedding feature vectors of D_1 and D_2. The final feature vector is itself a concatenated combination of all three feature sets. A classifier $\mathcal{C}$ is trained using the labeled data, and applied on the test data. Based on these scores (i.e. the positive class probability output by $\mathcal{C}$ per test item), a ranked list of relevant situation frames can be interactively shown to the HADR domain expert using the system.

In a supervised *batch* setting, the user inputs a document dump into THOR and expects clusters of situation frames, such that each cluster describes an event. As Fig. 1 illustrates, the documents first undergo processing through various components (e.g., NLP components like entity recognition and machine translation) that precede THOR. While clustering can generally be either supervised or unsupervised, it is supervised in this case because a user has specific cluster semantics (and granularity) in mind. If this were not the case, one could

also achieve a 'good' clustering by executing a topic modeling algorithm like LDA. In early trials, this was found to yield poor results in terms of capturing events, due to topical flux within event clusters; see, for example, the case of the Haiti earthquake in the introduction.

Instead, AugSEER combines *spectral clustering* with the classification scheme described earlier in a supervised setting [14]. Given a set $\mathcal{D}$ of frames, the input to AugSEER is a $|\mathcal{D} \times \mathcal{D}|$ affinity matrix. We assume training sets T_P and T_N respectively of positive and negative pairs, exactly like with classification. As a first step, we train the classifier $\mathcal{C}$ on the training sets. For efficiency reasons, we use either the (concatenated) node embedding or text embedding feature representations (not both) and we do not use the manual features[3]. The second step is to construct a *symmetric* affinity matrix $\mathcal{A}$ as follows. For a cell $\mathcal{A}_{ij}$ in the matrix indexed by (i, j), we use the following assignment function:

$$\mathcal{A}_{ij} = \begin{cases} 1 & \text{if } (D_i, D_j) \in T_P \\ 0 & \text{if } (D_i, D_j) \in T_N \\ \mathcal{C}(D_i, D_j) & \text{if } (D_i, D_j) \notin T_P \cup T_N \end{cases} \quad (1)$$

We assume that the classifier $\mathcal{C}$ outputs the probability of the statement $(D_i, D_j) \in T_P$. Note that spectral clustering, like many other well-known clustering algorithms like k-Means, requires the desired number of clusters as a hyperparameter. Because AugSEER is a tunable system designed to assist users in exploring events (not in giving final single-point outputs), we allow the user to set this number, but also provide guidance through validation. In evaluations, this value is set at the number of clusters in the ground-truth, both for AugSEER and baselines.

5 Experiments

AugSEER has been in development for almost a year, and several evaluations have been conducted. We evaluate the algorithmic potential of AugSEER on the SEER task, both quantitatively and through qualitative visualizations.

5.1 Datasets

We evaluate AugSEER on five HADR datasets described in Table 2. Each dataset is derived from real-world disasters, of which details were publicly published on, and scraped from, the *Relief Web Processed* portal[4]. The datasets describe different HADR categories and are quite diverse in their information content. In addition, we also consider a *global* dataset that combines the information in Datasets 1–5. We use this dataset both for exploring the generalization potential of the system, as well as the loss in performance when we do not combine

[3] A more technical reason is that we can visualize the representations this way using an algorithm like t-SNE [12], as we illustrate subsequently.

[4] http://reliefweb.int/.

Table 2. Dataset (and gold standard) details. *pos.* stands for *positively labeled*. Column 5 separately breaks down PER/ORG/LOC entity mentions. The average number of frames per cluster in datasets 1–5 are 3, 11, 4, 6 and 5 resp.

ID	Dominant themes	Unique frames	Unique pos. pairs	Unique entity mentions	Unique words	Clusters
1	Floods	234	535	972/1,069/1,398	13,108	74
2	Earthquakes/landslides	424	11,425	1,559/1,855/1,394	18,735	38
3	Cyclones/hurricanes	101	276	372/534/440	7,479	25
4	Disease-related/tropical	135	1,401	508/513/434	8,495	21
5	Miscellaneous	461	5,117	1,554/1,512/1,576	18,689	85

feature sets into an ensemble. Note that, to ensure a fair evaluation, the machine translation and named entity recognition outputs are already provided by the program for each situation frame in the datasets, in addition to the (not used) original, non-translated text.

Negatively labeled pairs for the classification task were generated as follows. Using each frame D in the corpus as a 'query', we computed a ranked list of all other frames in the corpus using a simple bag-of-words approach on the translated text. We computed the rank of the last frame D_i that describes the same event as D. All documents between rank 1 and i not describing the same event as D were paired with D and assigned a negative label. After computing such pairs using all documents as queries, and removing duplicate pairs, we sampled about 400,000 negative pairs (20x the total number of positive pairs in Datasets 1–5) as the negatively labeled evaluation corpus, shared among Datasets 1–5, as described subsequently.

5.2 Preliminaries

We simulate the *more like this* use-case by using each frame in an event cluster as a query, and by framing the problem of 'pairing' the query frame with relevant sub-event frames as a binary classification task (described earlier in Sect. 4.1).

Parameter Tuning. We used the Python sklearn library implementations for Random Forest (RF) and Logistic Regression (LogReg) classifiers, and for spectral clustering. The gensim package in Python was used both for paragraph2Vec, as well as the word2vec model that feeds into the DeepWalk node embedding. The best hyperparameters for LogReg were found using the LogisticRegressionCV class in sklearn that uses cross-validation (on the training set), and using grid search with cross-validation for RF.

Training Protocol. Training percentages vary with the experiment as described later, but training is always *balanced*. Namely, once $|T_D|$ is fixed for a given experiment, we sample $|T_N|$ pairs from the large negative pairs corpus described earlier in Sect. 5.1. The rest of the corpus is always used for testing. Because of sampling,

all experiments are conducted over ten trials, and averages are reported. We use the unpaired (two sample) Student t-test for computing statistical significance of the best performance against the *next* best alternative.

Metrics. Like with other entity resolution scenarios, *precision, recall* and their *F1-Measure* metrics on the positive class are used to report classification accuracy. For evaluating clustering, we use both the *cluster purity* and *F1-Measure* metrics. Given a cluster where each data item (i.e. a frame) has a label (withheld during clustering), in this case the underlying event that the frame is a part of, we compute cluster purity by taking the ratio of the number of frames having the majority label divided by the cluster size. F1-Measure can be computed by using the set of all pairs of frames sharing a cluster as the set of positives, and comparing against the known set of true positives to obtain the precision and recall (and by extension, their F1-Measure), similar to classification. We note that for all metrics, the higher the score, the better the performance.

5.3 Baselines

AugSEER involves a number of different interacting components both in classification and clustering settings. To illustrate that many of these components are jointly necessary for achieving good performance, we considered a range of competitive alternatives. We note that, because the SEER problem has not been studied in detail in the research literature (see Sect. 2), especially in the HADR domain, there are no direct SEER baselines available.

Classification. We consider three alternative feature-sets (or combinations) as baselines: only the manual features (M), only the DeepWalk features on the bipartite entity-frame network (N), and a combination of the two (MN). We also consider the PV text embedding baseline (T) in isolation, along with other text-only baselines like bag-of-words and topic models (using LDA), but all text-only baselines consistently under-performed the alternatives described above by significant margins. The full system includes all three feature sets (MNT).

Clustering. We tried several alternate clustering models, including Gaussian mixture models and agglomerative clustering, and found the latter to work best. We use both average (*agg-avg*) and complete (*agg-c*) linking when performing agglomerative clustering. Results are reported separately for node embedding and text embedding features. We also use *unsupervised* spectral clustering using node embeddings in a cosine similarity affinity space (*spec-N*) as a baseline, to investigate the effects of supervision in AugSEER's model of supervised spectral clustering. We also explored using the latter with TFIDF representations, but performance significantly declined, and we do not report those results herein.

5.4 Results

Four different sets of quantitative experiments, described below, were conducted to test the online and batch potential of AugSEER.

Experiment 1. For the very first set of experiments, we drew on standard findings that focus primarily on text and textual contexts, whether using embeddings or bag-of-words baselines. We considered both the classification and clustering settings, and describe the latter here (results were consistent for both). First, we built a supervised affinity matrix in the manner described in Sect. 4.1, using text embeddings, followed by spectral clustering. Across ten trials, the F1-Measure was only 10.73%, while cluster purity was higher at 71.6%. We also used cosine similarity to build an unsupervised affinity matrix, and while F1 was better for TFIDF (21.48%), the F1 for text embeddings was only 9.24%, almost 1.5% lower than for the supervised setting. Compared to the results described later, these results illustrate the non-viability of using text only, whether in low or high dimensional spaces, for addressing the challenges of SEER in the HADR domain. Alternatives like topic models, as well as alternate choices of word embeddings (e.g., PV vs. fastText), did not yield significant differences.

Experiment 2. For the second set of experiments, we tested the performance of AugSEER by using 30% and 15% of the positive samples in the *global* dataset for (balanced) training, and the rest for testing. We used both the Logistic Regression and Random Forest classifiers (with best hyperparameters determined using cross validation) with all the baseline feature sets mentioned earlier. The average best[5] F1-Measures over ten trials are reported in Table 3.

Table 3. F1-Measure results on the global dataset. MNT is the full feature set ensemble implemented in AugSEER.

Classifier (Training %)	MNT	MN	M	N
LogReg (30%)	**0.4982**	0.4924	0.4185	0.2570
LogReg (15%)	**0.5120**	0.5075	0.4439	0.2847
RF (30%)	0.7725	**0.7737**	0.4165	0.7729
RF (15%)	**0.7423**	0.7296	0.4359	0.7155

To test how the performance varied by the disaster theme, we used 30% of each dataset in Table 2 for training, and the other 70% for testing (over 10 trials). While we do not reproduce the full table herein, an absolute F1-measure improvement, using RF, was achieved by AugSEER (MNT) in the range of 0.8–18% for all five parts over the next best baseline (MN). We note that these results far outperform the text-only results[6] presented in Experiment 1.

Of the results in Table 3, RF (15%) and LogReg (15%) are significant at the 99% and 90% levels respectively. In other cases, there is no significant difference between MN and MNT. This provides some indication that all three feature

[5] By best, we mean that we chose the classifier threshold for all systems such that F1-measure achieved by that system was maximized in that trial at that threshold.

[6] Using average best F1 reporting and the 30% training methodology.

Table 4. *Precision/recall/F1-Measure* scores testing generalization of AugSEER (MNT and MN). All results are statistically significant at the 99% confidence level. LogRef (MNT), which is all bold, performs uniformly worse than LogReg (MN), omitted here due to space.

Training Dataset	Test Datasets	RF (MNT)	RF (MN)	LogReg (MN)
1	2+3+4+5	0.223/0.494/0.307	**0.320/0.494/0.393**	**0.228/0.296/0.275**
2	1+3+4+5	0.587/0.150/0.238	**0.614/0.163/0.258**	**0.272/0.228/0.248**
3	1+2+4+5	0.294/0.474/0.363	**0.339/0.475/0.395**	**0.271/0.300/0.284**
4	1+2+3+5	0.166/**0.457**/0.243	**0.205**/0.390/**0.268**	**0.257/0.205/0.228**
5	1+2+3+4	0.186/0.483/0.268	**0.209/0.506/0.296**	**0.156/0.225/0.184**

Table 5. Cluster *purity* scores using either *node embeddings/text embeddings* in a cosine similarity space (agg-*) or affinity matrix (AugSEER), except spec-N (only node embedding results reported).

ID	agg-av	agg-c	AugSEER	spec-N
1	0.611/ 0.415	0.633/ 0.402	**0.671/ 0.633**	0.556
2	0.718/ 0.384	0.723/ 0.410	**0.920/ 0.880**	0.678
3	0.644/ 0.426	0.634/ 0.416	**0.792/ 0.822**	0.624
4	0.748/ 0.496	0.704/ 0.578	**0.963/ 0.852**	0.644
5	0.639/ 0.475	0.641/ 0.456	**0.755/ 0.592**	0.522

sets have merit, with the effects more dramatic for Logistic Regression than for Random Forest. Overall, the node embedding feature vectors $\boldsymbol{D_N}$ are found to be especially instrumental, illustrating the importance of entities, both latent and explicit, for the SEER task. The good absolute performance of RF over LogReg, even after cross-validation, provides further evidence for the importance of robust feature combinations. Additionally, RF is able to generalize without overfitting, when given more training data (unlike LogReg, which clearly starts overfitting in the 30% setting, compared to the 15% setting). We tried other classifiers like SVM, and found that they underperformed RF as well.

Experiment 3. We isolate the generalization ability of different feature sets in a setting resembling *transfer learning* [16]. We used one of the datasets in Table 2 as positively labeled training data, and the others for testing. We used the same negatively labeled dataset described in Sect. 5.1 for all experiments. Maximal performance was found to be achieved across all settings with balanced training. This resulted in five training/testing paradigms. We report the average (over ten trials[7]) best F1-Measure achieved, along with corresponding precision and recall

[7] Because of balanced training, we had to randomly sample the negative training set; the positive set remained constant per trial.

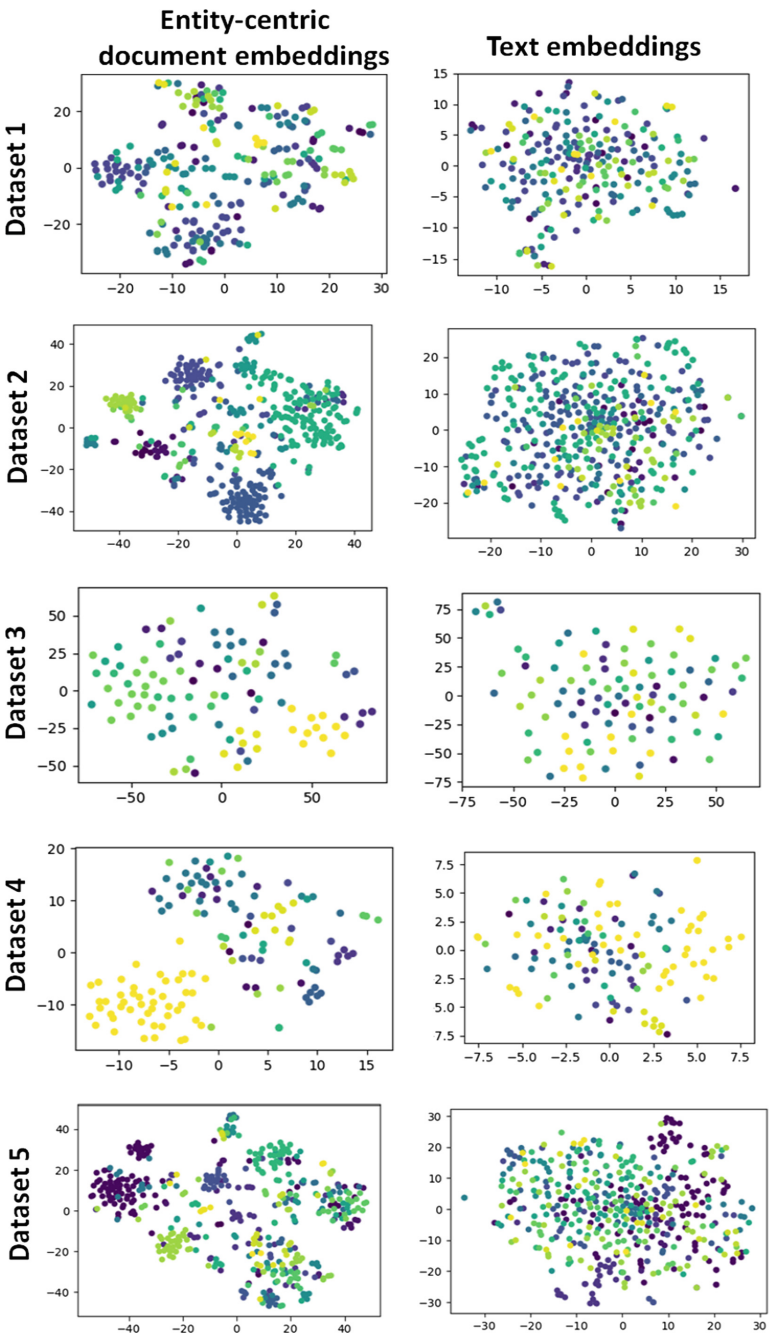

Fig. 3. t-SNE visualizations of all datasets using node/text embeddings (for visual purposes, the same color is sometimes re-used to represent different events). Dimensions have no intrinsic meaning in t-SNE [12].

in Table 4, limiting results to only the two best performing systems, which were always MN and MNT.

Similar to Table 3, we find that the two feature combinations perform similarly, but the trend is reversed. The text embeddings, which *weakly* increased the power of the classifier in the first experiment, have negative influence in this experiment. This experiment offers a cautionary lesson in naively transferring text embeddings, even in domains that seem somewhat similar (every dataset is from an HADR domain). If the data in the training phase does not sufficiently represent the test data (true in this experiment, but not the previous experiment), text embeddings can reduce F1-Measure by as much as 5%.

Experiment 4. We evaluate AugSEER in the batch/posthoc analysis setting. Using 30% positively labeled pairs in a (balanced training) supervised setting, and the RF classifier, we test AugSEER's performance against the agglomerative clustering baselines (using both average and complete link functions) as well as unsupervised spectral clustering (spec-N). In all cases, AugSEER outperforms rival methods on the cluster purity metric by a considerable margin[8], both when using node and text embeddings. When using the F1-Measure metric, a similar trend is observed, but with narrower improvements (3% average improvement, rather than the 15% achieved using cluster purity). In the next section, we use visualizations to emphasize that the latent space model and representation that AugSEER employs for entities has considerable influence on performance.

Visualization Experiments. Visualization is an important function in AugSEER as it is primarily a cognitive system designed to facilitate rapid situational awareness in both military and civilian situations. All visualizations described in this section employ the unsupervised t-SNE algorithm [12]. In an actual deployment, we use THOR (Fig. 1) for an interactive interface. Figure 3 shows that clusters for all datasets achieve an intuitive separation into different events when using the entity-document node embedding representation, but not the text embedding representation, supporting the hypothesis that entities and semantics are fundamental in addressing SEER challenges.

6 Conclusion

This paper presented AugSEER, a statistical-semantic approach for addressing structured event-entity resolution. AugSEER supports a combination of graph and text embeddings, and manually devised feature sets to achieve 77% highest F1-Measure on a challenging classification problem, using only 30% labeled training data. Similar results are achieved in the clustering scenario. AugSEER has also been implemented into a broader HADR system called THOR (Fig. 1) that is designed to ingest noisy NLP outputs and assist HADR field analysts in real-time in low-resource environments[9].

[8] All results in Table 5 are statistically significant at the 99% level, except AugSEER node embedding results on Dataset 1.

[9] THOR was recently demonstrated in an academic venue also: https://www2018.thewebconf.org/program/demos-track/.

Acknowledgements. The authors gratefully acknowledge the ongoing support and funding of the DARPA LORELEI program, and the aid of our partner collaborators and users in providing detailed analysis. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, AFRL, or the U.S. Government.

References

1. Cai, H., Zheng, V.W., Chang, K.: A comprehensive survey of graph embedding: problems, techniques and applications. In: IEEE Transactions on Knowledge and Data Engineering (2018)
2. Dai, A.M., Olah, C., Le, Q.V.: Document embedding with paragraph vectors. arXiv preprint arXiv:1507.07998 (2015)
3. Finkel, J.R., Manning, C.D.: Nested named entity recognition. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 1, pp. 141–150. Association for Computational Linguistics (2009)
4. Getoor, L., Machanavajjhala, A.: Entity resolution: theory, practice & open challenges. Proc. VLDB Endow. **5**(12), 2018–2019 (2012)
5. Hahn, R.F.: Modern uighur language research in China: four recent contributions examined. Cent. Asiat. J. **30**(1/2), 35–54 (1986)
6. Hristidis, V., Chen, S.-C., Li, T., Luis, S., Deng, Y.: Survey of data management and analysis in disaster situations. J. Syst. Softw. **83**(10), 1701–1714 (2010)
7. Jahan, S.: Human Development Report 2016: Human Development for Everyone. United Nations Development Programme (UNDP), New York (2016)
8. Li, T., et al.: Data-driven techniques in disaster information management. ACM Comput. Surv. (CSUR) **50**(1), 1 (2017)
9. Lingad, J., Karimi, S., Yin, J.: Location extraction from disaster-related microblogs. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1017–1020. ACM (2013)
10. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Aggarwal, C., Zhai, C. (eds.) Mining Text Data. Springer, Boston (2012). https://doi.org/10.1007/978-1-4614-3223-4_13
11. Lu, J., Ng, V.: Joint learning for event coreference resolution. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 90–101 (2017)
12. Maaten, Lvd, Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013)
14. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems, pp. 849–856 (2002)
15. Ng, V.: Machine learning for entity coreference resolution: a retrospective look at two decades of research. In: AAAI, pp. 4877–4884 (2017)
16. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
17. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710. ACM (2014)

18. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web, pp. 851–860. ACM, 2010
19. Vandeghinste, V., Schuurman, I., Carl, M., Markantonatou, S., Badia, T.: METIS-II: machine translation for low resource languages. In: Proceedings of LREC 2006 (2006)
20. Verma, S., et al.: Natural language processing to the rescue? extracting "situational awareness" tweets during mass emergency. In: ICWSM (2011)
21. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)