Intrinsic and Extrinsic Evaluation of Spatiotemporal Text Representations in Twitter Streams

Lawrence Phillips and **Kyle Shaffer** and **Dustin Arendt** and **Nathan Hodas** and **Svitlana Volkova**
Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

Language in social media is a dynamic system, constantly evolving and adapting, with words and concepts rapidly emerging, disappearing, and changing their meaning. These changes can be estimated using word representations in context, over time and across locations. A number of methods have been proposed to track these spatiotemporal changes but no general method exists to evaluate the quality of these representations. Previous work largely focused on qualitative evaluation, which we improve by proposing a set of visualizations that highlight changes in text representation over both space and time. We demonstrate usefulness of novel spatiotemporal representations to explore and characterize specific aspects of the corpus of tweets collected from European countries over a two-week period centered around the terrorist attacks in Brussels in March 2016. In addition, we quantitatively evaluate spatiotemporal representations by feeding them into a downstream classification task – event type prediction. Thus, our work is the first to provide both intrinsic (qualitative) and extrinsic (quantitative) evaluation of text representations for spatiotemporal trends.

1 Introduction

Language in social media presents additional challenges for textual representations. Being able to represent texts in social media streams requires a methodology with the following properties:

1. Capable of handling *large amounts* of data.
2. In a *streaming* rather than static fashion.
3. Across *many geographic regions.*

While there has been some recent work for representing change over time in embedding spaces, these methods largely did not take into account geographic variation (Costa et al., 2014; Kim et al., 2014; Kulkarni et al., 2015; Hamilton et al., 2016b,a). Likewise, papers examining geographic variations of language tend not to examine data temporally (Bamman et al., 2014; Kulkarni et al., 2016; Pavalanathan and Eisenstein, 2015; Hovy et al., 2015). Although Kulkarni et al. (2016) incorporate temporal information, they treat each timestep as a separate corpus, learning unique representations. We propose two algorithms to learn spatiotemporal text representations from large amounts of social media data and investigate their utility both from a qualitative and quantitative standpoint.

Indeed, the broader question of how to evaluate the quality of an embedding is one which has received a great deal of attention (Schnabel et al., 2015; Gladkova et al., 2016). Previous spatial and temporal embedding algorithms have been evaluated primarily with qualitative evidence, investigating the ability of the embedding to capture a small number of known meaning shifts and providing some form of visualization (Costa et al., 2014; Kim et al., 2014; Kulkarni et al., 2015, 2016; Hamilton et al., 2016b,a). While it is important to capture known changes of interest, without some form of quantitative evaluation it cannot be known whether these embedding methods actually produce good vector spaces. Because of these issues we not only provide the first spatiotemporal algorithms for learning text embeddings from social media data, but we also evaluate our embedding algorithms through a variety of means.

For **qualitative evaluation**, we develop a set of novel visualizations[1] which allow us to investigate

[1]Live Demo: https://esteem.labworks.org/

Proceedings of the 2nd Workshop on Representation Learning for NLP, pages 201–210,
Vancouver, Canada, August 3, 2017.

word representation shifts across space and time. In particular, we demonstrate that the model captures temporal shifts related to events in our corpus and these shifts vary across distinct countries. For **quantitative evaluation**, we estimate the effectiveness of spatiotemporal embeddings through a downstream event-classification task, demonstrating that temporal and spatial algorithms vary in their usefulness. We choose an extrinsic evaluation task rather than the more standard intrinsic embedding evaluation because of recent work demonstrating weak relationships between intrinsic measures and extrinsic performance (Schnabel et al., 2015; Gladkova et al., 2016).

2 Background

Text Representations Most existing algorithms for learning text representations model the context of words using a continuous bag-of-words approach (Mikolov et al., 2013a), skip-grams with negative sampling (Mikolov et al., 2013b), modified skip-grams with respect to the dependency tree of the sentence (Levy and Goldberg, 2014), or optimized ratio of word co-occurrence probabilities (Pennington et al., 2014).

Text representations have been learned mainly from well-written texts (Al-Rfou et al., 2013). Only recent work has focused on learning embeddings from social media data e.g., Twitter (Pennington et al., 2014). Moreover, most of the existing approaches learn text embeddings in a static (batch) setting. Learning embeddings from streaming social media data is challenging because of problems such as noise, sparsity, and data drift (Kim et al., 2014; Kulkarni et al., 2015).

Embedding Evaluation There are two principle ways one can evaluate embeddings: (a) intrinsic and (b) extrinsic.

(a) **Intrinsic evaluations** directly test syntactic or semantic relationships between the words, and rely on existing NLP resources e.g., WordNet (Miller, 1995) and subjective human judgements e.g., crowdsourcing or expert judgment.

(b) **Extrinsic methods** evaluate word vectors by measuring their performance when used for downstream NLP tasks e.g., dependency parsing, named entity recognition etc. (Passos et al., 2014; Godin et al., 2015)

Recent work suggests that intrinsic and extrinsic measures correlate poorly with one another (Schnabel et al., 2015; Gladkova et al., 2016). In many cases we want an embedding not just to capture relationships within the data, but also to do so in a way which can be usefully applied. In these cases, both intrinsic and extrinsic evaluation must be taken into account.

Temporal Embeddings Preliminary work on studying changes in text representations over time has focused primarily on changes over large timescales (e.g. decades or centuries) and in well-structured text such as books. For instance, Kim et al. (2014) present a method to measure change in word semantics across the 20th century by comparing each word's initial meaning with its meanings in later years. Other work explores a wider range of corpora (all based on text from books) and embedding methods and report similar qualitative findings (Hamilton et al., 2016b). What quantitative evidence they do provide is limited to intrinsic evaluations of word similarities as well as the model's ability to recognize a small set of hand-selected known shifts. One attempt at learning over time from social media comes from Costa et al. (2014) that explore a number of online learning methods for updating embeddings across timesteps. They measure the ability of their temporal embeddings to predict Twitter hashtags, but do not compare their results against a non-temporal baseline which makes it difficult to assess the usefulness of learning temporal embeddings. Finally, more recent work learns from Twitter, among other data sources, but presents only qualitative evaluations (Kulkarni et al., 2015).

Spatial Embeddings Work on incorporating space into low-dimensional text representations has been less well researched. Only recent work presents an approach to train embedding models independently across a variety of English-speaking countries as well as US states (Kulkarni et al., 2016). Their model creates a general embedding space which is shared across all regions, as well as a region-specific embedding space which captures local meaning. Although they are able to report a number of meaning differences captured by the model, no general quantitative evaluation is given. The lack of extrinsic evaluation both for temporal and spatial representations highlights a major difficulty for future research. Although it is clear that temporal and spatial patterns can be captured by distributed text representations, unlike other approaches, our work is the first to quali-

Event Types (A)	Clusters	Tweets	Tweet / Cluster	Event Types (B)	Clusters	Tweets	Tweet / Cluster
Arts	321	18,926	144	Politics	329	36,908	270
Business	157	8,961	276	Entertainment	315	18,977	147
Politics	190	18,729	105	Business	208	9,092	97
Justice	138	1,296	27	Crime	175	16,194	233
Conflict	101	1,623	46	Terrorism	97	1,369	41
Life	93	602	20	Transportation	46	196	15
Personnel	53	8,457	412	Celebrity	43	401	22
Contact	28	226	19	Death	35	9,021	646
Transaction	20	175	872	Health	33	178	20
Nature	20	6,960	746	Natural disaster	20	6,953	873

Table 1: Distributions of event types in two annotation schemes: A from Doddington et al. (2004) and B from Metzler et al. (2012).

tatively and quantitatively evaluate whether these patterns can be useful in applied tasks.

Event Type Classification To do this, we focus on event detection, a popular NLP task to detect mentions of relevant real-world events within text documents. Some earlier efforts include TABARI (Schrodt, 2001), GDELT (Leetaru and Schrodt, 2013), TDT (Allan, 2012) challenges, and the Automatic Content Extraction (ACE) program (Doddington et al., 2004). This task can take the form of summarizing events from text (Kedzie et al., 2016), querying information on specific events (Metzler et al., 2012), or clustering together event mentions (Ritter et al., 2012) that all describe the same event.

In this work, we focus on building predictive models to classify event types from raw tweets. Only limited work in event classification has also tried to codify events into specific event types, such as "political" vs. "economic" (Bies et al., 2016). Because the desired granularity of an event type can vary depending on the end-task, we analyze our tweets using modified versions of event types from the ACE program (Doddington et al., 2004) and more topical event types defined by Metzler et al. (2012).

3 Datasets

We make use of three datasets in our experiments. First, we use a large corpus of European Twitter data captured over two weeks in order to learn text representations across time and space. For our event classification task, we chose a subset of tweets in the larger corpus which were made by news accounts. These "news-worthy" tweets were then manually annotated for event type. To leverage the additional available data annotated with real-world events, we train our models on a larger event dataset from Wikipedia and then use transfer learning to apply it to our smaller event data.

Brussels Bombing Twitter Dataset We collected a large sample of tweets (with geo-locations and language IDs assigned to each tweet) from 240 countries in 66 languages from Twitter. Data collection lasted two weeks, beginning on March 15th, 2016 and ending March 29th, 2016. Tweets were filtered based on geo-location and language tags to include only English-language tweets from a set of 34 European countries that had at least 10,000 English tweets per day in the corpus. This resulted in a set of 140M tweets we use to learn different types of embeddings.

Twitter Event Dataset We selected "news-worthy" tweets that discuss real-world events from 400M English tweets generated in 240 countries. Our criterion for selecting "news-worthy" tweets was to only select tweets that contain an action word from the connotation frame lexicon (Rashkin et al., 2016) and either come from a verified account, from a news account e.g., @bbc, @wsj, or contain the hashtag "#breaking" or "#news". We identified 600,000 English *subject-predicate-object* tuples using SyntaxNet (Andor et al., 2016).

Three annotators labeled event types for all tuples based on two previously defined lists of event categories: the ACE event categories (Doddington et al., 2004) and those from a related paper on querying event types (Metzler et al., 2012). Because of missing values for the third annotator, we used Krippendorff's alpha to judge inter-annotator agreement (like Fleiss' kappa, Krippendorff's alpha is ≤ 1 with 1 indicating complete agreement and 0 indicating random chance). This subset of labeled clusters without ties have high inter-annotator agreement: 0.71 and 0.78, respectively. Finally, we subsampled our "news-worthy" tweets to match the 34 European countries in the Brussels dataset. We show the final number of clusters and tweets per event category in Table 1.

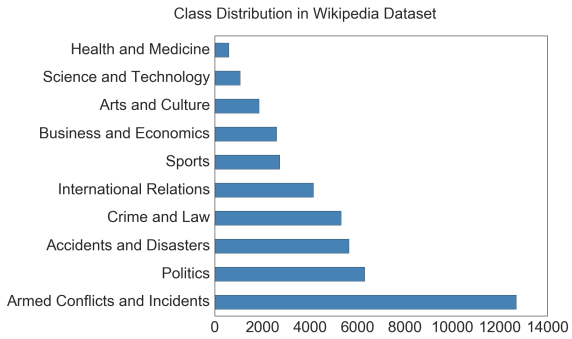

Figure 1: Top 10 classes in the Wikipedia event dataset.

Wikipedia Event Dataset Given the small size of our Twitter event dataset, we explore additional resources for training an effective event detection model. We construct a larger event dataset by scraping the English language Wikipedia Current Events Portal from the time period of January 2010 to October 2016. Each event in this portal is described in a short summary and is labeled by date along with a subject heading such as *Armed Conflicts and Attacks*. We use these summaries and headings as training data for a neural network to be used for transfer learning.

Overall, the Wikipedia event dataset contains 43,098 total event samples and 31 event type classes. For training, we use the 42,906 samples that correspond to the ten most frequent classes within the dataset, approximately 99.5% of the original data. The distribution of these ten most frequent classes is shown in Figure 1.

4 Methodology

All embeddings were trained using gensim's continuous bag-of-words word2vec algorithm (Řehůřek and Sojka, 2010). All of our embeddings were 100 dimensional with embeddings learned over the full vocabulary. For evaluation, we limit ourselves to vocabulary occurring at least 1,000 times in the Brussels dataset, resulting in a vocabulary size of 36,200.

Temporal Embeddings We build upon the state-of-the-art algorithm to learn embeddings (Mikolov et al., 2013a). In order to learn embeddings over time we separate our corpus into 8-hour windows, resulting in 45 timesteps. For each timestep, we train a model using the previous timestep's embeddings to initialize the model at time $t+1$ as shown in Algorithm 1.

This results in an embedding space specific to each timestep capturing any change in meaning which has just occurred. Because timesteps are connected through initialization, we can examine how word representations shift over time.

Algorithm 1 Temporal Text Representations

Initialize $\mathbf{W}^{(0)}$ randomly
$\mathbf{W}^{(0)}$ = LearnEmbeddings(C, t_0)
for timestep t in T **do**
    Initialize $\mathbf{W}^{(t)}$ with $\mathbf{W}^{(t-1)}$
    $\mathbf{W}^{(t)}$ = LearnEmbeddings(C, t)

Spatial Embeddings The simplest method for learning embeddings across countries is to train a separate set of embeddings for each country independently as shown in Algorithm 2. We use these spatial embeddings without time to investigate the ability of this simple method to capture task-relevant information.

Algorithm 2 Spatial Text Representations

for country c in C **do**
    Initialize $\mathbf{W}_{(c)}$ randomly
    $\mathbf{W}_{(c)}$ = LearnEmbeddings(c, T)

Spatiotemporal Embeddings We train each spatial region separately, but rather than training over the entire corpus, we train in 8 hour time chunks using the previous timestep for initialization as shown in Algorithm 3.

Algorithm 3 Spatiotemporal Embeddings

for country c in C **do**
    Initialize $\mathbf{W}_{(c)}^{(0)}$ randomly
    $\mathbf{W}_{(c)}^{(0)}$ = LearnEmbeddings(c, t_0)
    for timestep t in T **do**
        Initialize $\mathbf{W}_{(c)}^{(t)}$ with $\mathbf{W}_{(c)}^{(t-1)}$
        $\mathbf{W}_{(c)}^{(t)}$ = LearnEmbeddings(c, t)

Global2Specific Embeddings The disadvantage of training each country's embeddings independently is that countries with more tweets will necessarily possess better learned embeddings. We explore an alternative method where for each timestep, we train a joint embedding using tweets from all countries and use it to initialize the country-specific embeddings on the *following* timestep as shown in Algorithm 4.

By initializing with joint embeddings, high quality vectors for infrequent words can be retained across countries. In cases where a country's usage for a word does not differ from overall

Algorithm 4 Global2Specific Embeddings

Initialize $\mathbf{G}^{(0)}$ randomly
$\mathbf{G}^{(0)}$ = LearnEmbeddings(C, t_0)
for timestep t in T **do**
  for country c in C **do**
    Initialize $\mathbf{W}_{(c)}^{(t)}$ with $\mathbf{G}^{(t-1)}$
    $\mathbf{W}_{(c)}^{(t)}$ = LearnEmbeddings(c, t)
  Initialize $\mathbf{G}^{(t)}$ with $\mathbf{G}^{(t-1)}$
  $\mathbf{G}^{(t)}$ = LearnEmbeddings(C, t)

usage, it can still rely on the embeddings learned from a larger data. If the meaning of a word does change in a particular country, this will still be captured as the model learns from that timestep.

Aligning Embeddings Following Hamilton et al. (2016b), we use Procrustes analysis to align embeddings across space and time for more accurate comparison. Procrustes analysis provides an optimal rotation of one matrix with respect to a "target" matrix (in this case a word embedding matrix in the joint space) by minimizing the sum of squared (Euclidean) distances between elements in each of the matrices.

Predictive Models for Wikipedia Events We use subject headings from the Wikipedia event dataset as noisy labels to train a model to predict the ten most frequent classes within the dataset. Weights of the LSTM layer of this model are used to initialize the LSTM layer in the Twitter event classification models.

We divide the Wikipedia event dataset using 10-fold cross-validation, optimizing the network for F_1 score on the validation sets. We searched over hyper-parameters for dropout on all layers, number of units in the fully-connected layer, activation function (rectified linear unit, hyperbolic tangent, or sigmoid), and batch size using the Hyperas library.[2]. The model was trained for 10 epochs, implemented using Keras.[3] Hyperoptimized parameters were then used to train the model on the full dataset to be transferred to the event detection model. We compare the LSTM performance against three simpler models trained on TFIDF features using scikit-learn.[4]

Predictive Models for Twitter Events For each word in each tweet, we first look up the appropriate embedding vector. If a word does not have a corresponding embedding vector, we create an "average vector" from all the word vectors for the appropriate embedding type, and use this as the representation of the word. Preliminary results indicated that averaging produced better results than using a zero-vector. These embedding representations of tweets are fed to a fully-connected dense layer of 100 units. This layer is regularized with 30% dropout, and its outputs are then fed to 100 LSTM units whose weights have been initialized with the LSTM weights learned from our Wikipedia neural network. We tried both freezing these weights in the LSTM layer as well as allowing them to be tuned in the training process, and found that further tuning helped model performance. The output of this LSTM layer is then fed to another densely connected layer of 128 units regularized with 50% dropout, before passing these outputs to a final softmax layer to compute class probabilities and final predictions. We use rectified linear units as the activation function for both densely connected layers, and use the Adam optimization algorithm with a learning rate set to 0.001. We experimented with various other architectures, including adding 1-dimensional convolutional and max-pooling layers between the first dense layer and the LSTM layer, but did not find these to be advantageous.

Baselines and Evaluation Metrics As a baseline, we compare our spatiotemporal embeddings against openly available, pre-trained embeddings – 300-dimensional Word2Vec embeddings trained on Google News, and 100-dimensional GloVe embeddings trained on 2 billion tweets. In addition, we evaluate three simpler classifiers on the the 5- and 10-way event classification problems. We train logistic regression (LR), linear SVM, and a random forest classifier (RF) with 100 decision trees on TFIDF features from our labeled Twitter dataset, and report micro and macro F1 scores over 10-fold c.v. in Table 4 below.

5 Results

Qualitative Evaluation We analyze the results of temporal embeddings when trained over all countries of the Brussels dataset in Figure 2. Time is plotted on the x-axis with every tick indicating a single 8-hour timestep. The distance between ticks is proportional to the change in the keyword's vector representation (Euclidean distance) during that time. Vertical gray bars indicate a change greater than one standard deviation above the mean. For

[2]Hyperas: https://github.com/maxpumperla/hyperas
[3]Keras: https://keras.io/
[4]Scikit-learn: http://scikit-learn.org/

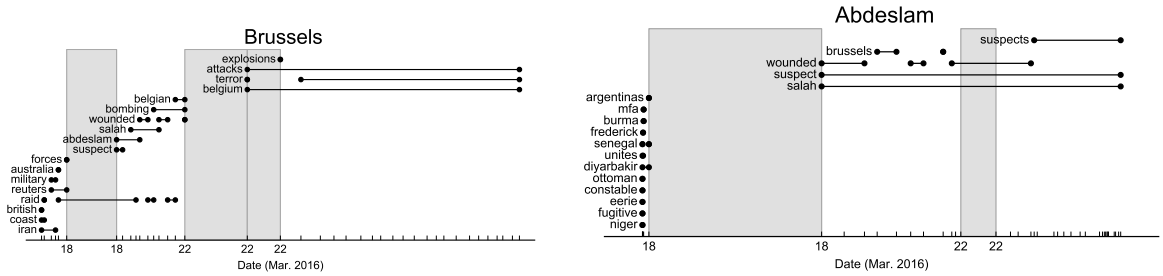

Figure 2: Visualizing temporal embeddings: Top-3 similar keywords to the concepts *Brussels* and *Abdeslam* for each timestamp (similarity is measured using Euclidean distance).

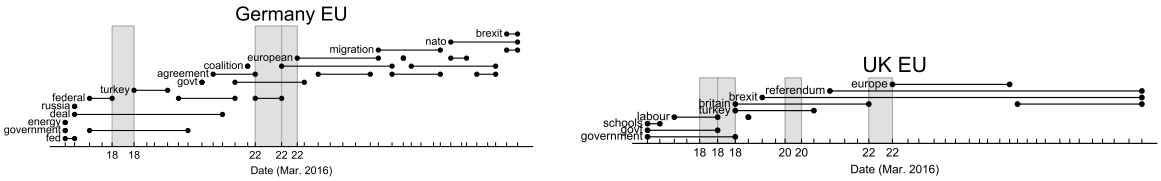

Figure 3: Visualizing spatial (country-specific) embeddings: Top-3 similar keywords to the concept *EU (European Union)* for each timestamp (similarity is measured using Euclidean distance).

each timestep, the three most similar keywords are plotted on the y-axis, with horizontal lines indicating that the keyword was in the top three over that period. We plot the keywords *Brussels* as well as *Abdeslam*, which is the last name of a suspect in the 2015 Paris bombing. For both words we see large shifts in meaning both on March 18th and 22nd. On March 18th, Salah Abdeslam was captured in Brussels during a police raid.[5] Before that date, Abdeslam was not widely mentioned on Twitter and the meaning of his name was not well learned. After his capture, Twitter users picked up the story and the embedding quickly relates Abdeslam to *Salah* (his first name), *suspect*, and *wounded*. Mention of Salah Abdeslam is also visible in the most similar keywords to *Brussels*. On March 22nd the Brussels bombing occurred.[6] and one can see that the embedding of *Brussels* quickly shifts, with *belgium*, *terror*, and *attacks* remaining as the top three similar keywords for all following timesteps.

To understand how different countries discuss a global event, we examined keywords of interest and identified for each country the top k most similar words. Table 2 presents the top-4 similar words to the keyword *Belgium* across five countries in our dataset from the spatial embeddings learned over all timesteps. Each country references the bombing which took place on March 22nd, but each country is referring to Belgium in different ways. Belgium and the UK, for instance, draw parallels to the suicide bombing in Lahore, Pakistan on March 27th,[7] while *Brussels* in France and Russia is more linked to the suspect and bomber of the Brussels attack.

Belgium	**France**	**Russia**	**UK**
killed	terrorism	bombers	pakistan
attack	suspect	belgian	bombing
isis	bombings	condolences	iraq
pakistan	turkey	bomber	lahore

Table 2: Most similar words to a query word *Belgium* for country-specific text representations.

Countries discuss topics in ways that grow more similar or distant over time. Looking at the keywords *Brussels* and *Radovan*, we calculate the cosine distance between the word vectors of any two countries and plot the three most extreme country pairs becoming more or less similar over time. For Brussels, we see that before the terror attack on the 22nd, there is a great amount of divergence between countries. After the 22nd, many of these differences disappear as can be seen in the blue lines which indicate the three most-converging country pairs. Belgium itself, however, continues to diverge from other countries even after the 22nd as can be seen in the red lines. Another event during our corpus was the conviction and sentencing

[5] https://www.theguardian.com/world/2016/mar/18/paris-attacks-suspect-salah-abdeslam-wounded-in-brussels-terror-raid-reports

[6] http://www.bbc.com/news/world-europe-35869254

[7] http://www.nytimes.com/2016/03/28/world/asia/explosion-lahore-pakistan-park.html

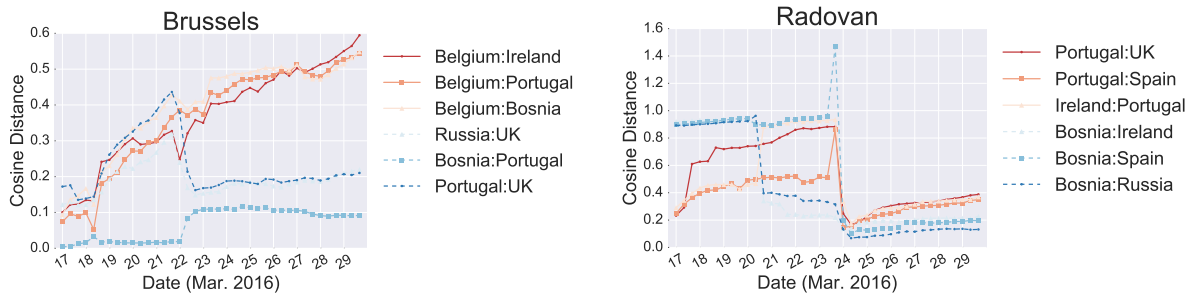

Figure 4: Converging and diverging country pairs for the keywords *Brussels* and *Radovan*. Converging country pairs (in blue) became more similar over time than other pairs. Diverging country pairs (in red) became more dissimilar over time.

Embedding	**A5 (4,325)**		**A10 (5,480)**		**B5 (5,017)**		**B10 (5,940)**	
	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro
Baseline 1: Word2Vec[6]	0.64	0.68	0.49	0.62	0.67	0.68	0.45	0.61
Baseline 2: Glove[7] (2B tweets)	0.66	0.69	0.46	0.59	0.65	0.66	0.43	0.59
Static (140M tweets; upperbound)	0.81	0.82	0.53	0.69	0.78	0.79	0.53	0.72
Temporal (1.4–4.6M tweets)	0.74	0.76	**0.51**	**0.66**	0.75	**0.77**	**0.52**	**0.67**
Spatial (0.2M–81.7M tweets)	0.62	0.66	0.36	0.55	0.65	0.67	0.39	0.57
Spatiotemporal (2K–2.8M tweets)	0.67	0.70	0.41	0.59	0.69	0.71	0.43	0.63
Global2Specific (2K–2.8M tweets)	**0.76**	**0.77**	0.46	0.65	**0.75**	0.77	0.49	0.67

Table 3: Embedding evaluation results (F1) for event detection task (best performance is marked in bold). Tweets from specific timesteps and countries make use of relevant temporal and spatial embeddings where applicable.

of Radovan Karadžić who was found guilty on the 24th for, among many other crimes, the Srebrenica massacre in 1995.[8]

While the cases listed above represent a number of real-world events that can be visualized and captured by the embedding models, we note that not all events will necessarily be captured in this same way. For instance, an event discussed many months in advance, and with many related tweets, may not see the same shifts that characterize the examples provided here. Still, our visualization techniques are able to extract meaningful relations demonstrating possible utility for social scientists hoping to better understand their data.

Quantitative Evaluation We investigate which of our embedding types are most useful for a downstream event classification task. We present a performance comparison between using our embeddings and using pre-trained Word2Vec and GloVe embeddings, as well as performance comparison between a recurrent neural network and three other models. The neural network uses the same batch size, number of epochs, and 10-fold cross-validation scheme as before. Results from our experiments are presented in Tables 3, 4 and 5. We present results of 5- and 10-way classification for each of the annotation schemes (A or B), and denote these as abbreviations of the annotation letter and number of classes e.g., A5.

Table 4 demonstrates the clear effectiveness of our LSTM model, which outperforms all other models in all classification tasks. At minimum, we see a 4.2% increase in $F1$ score over the next best model and in some cases we see an 8.5% increase in $F1$ score. While we see the largest gains in the 5-way classification task, we also see an 8.2% $F1_{macro}$ increase in the 10-way classification task for annotation B. This suggests that at least some of our embeddings are more effective at capturing information relevant to a downstream classification task than other more straightforward linguistic features such as TFIDIF weights. However, this analysis does not determine whether the increased predictive power of the LSTM or the increased information from our embeddings contributes most to this performance boost.

For a more rigorous analysis of our embedding types, we compare their effectiveness as inputs for our LSTM for this event classification task. This is the first attempt to quantitatively measure the performance of spatiotemporal embeddings on a sizable evaluation task made of non-simulated

[8] http://www.nytimes.com/2016/03/25/world/europe/radovan-karadzic-verdict.html

data. In addition to comparing the different types of embeddings, we also compare our embeddings against pre-trained static embeddings. Results for these experiments are given in Table 3. We experimented with increased embedding dimensionality but find only a modest gain and therefore report only for our 100-dimension embeddings.

We find that *static* embeddings trained on the two-week corpus of Twitter data outperform pre-trained embeddings (both Word2Vec and GloVe), even when trained on a much larger quantity of tweets e.g., 140M vs. the 2B used by GLoVe. *Static* embeddings also outperform all spatiotemporal embeddings, likely due to the amount of training data used by each embedding. The spatial algorithm has the worst performance, which also coincides with the fact that the spatial model is unable to share information between different locations. This differs from the temporal, spatiotemporal, and Global2Specific models which are able to share information across timesteps, reducing data sparsity somewhat. Although we find generally lower performance, increased data may improve the spatiotemporal models and these models have the additional advantage that they can be trained online, allowing researchers to study changes as they occur.

The difficulty for naive spatial embeddings is countered by our *Global2Specific* strategy. Recall that in the training process for these embeddings, at each timestep a joint embedding is trained using tweets from all countries. These joint embeddings are then used to initialize the embedding learned for a given country. Intuitively, this initialization should result in better learned embeddings since it leverages all of the data in the joint space (140M tweets) as well as spatiotemporal aspects of the data. While these embeddings do not outperform those learned completely in the joint space, they demonstrate that this training process transfers useful information, outperforming the *spatial* and *spatiotemporal* embeddings.

6 Summary and Discussion

Discourse on social media varies widely over space and time, thus, any static embedding method will have difficulty resolving how events influence discourse. It can be difficult to *a priori* define an effective embedding scheme to capture this without explicitly encoding space and time. In demonstrating the value of spatiotemporal embeddings, we can clearly observe the variation in discourse caused by significant events. We can pinpoint the event, such as the Brussels bombing, down to the resolution of our temporal embedding technique – 8 hours, in this case. We also observe general differences in how discourse varies over geography.

What previous work has not made clear is whether spatiotemporal embeddings also have value in a quantitative sense. Our event classification results show that simple spatiotemporal strategies are not necessarily useful. The value of spatiotemporal learning must be weighed against loss of data when multiple embeddings must be separately trained. The success of our *Global2Specific* embeddings compared to other strategies demonstrates that explicitly accounting for this loss of data is a useful strategy. Future work will need to investigate whether spatiotemporal embeddings have value only when trained on very large data or if better strategies can be incorporated to explicitly model space and time.

Annotation	F1	LR	SVM	RF	LSTM
A5	Macro	0.68	0.72	0.62	**0.80**
	Micro	0.71	0.74	0.66	**0.82**
A10	Macro	0.42	0.48	0.41	**0.53**
	Micro	0.61	0.64	0.58	**0.69**
B5	Macro	0.70	0.72	0.64	**0.78**
	Micro	0.72	0.72	0.65	**0.79**
B10	Macro	0.37	0.45	0.37	**0.53**
	Micro	0.64	0.65	0.59	**0.72**

Table 4: Results of baseline models and LSTM trained on static embeddings. Best performance is marked in bold.

A10	F1	B10	F1
Arts	0.74	Entertainment	0.79
Business	0.73	Politics	0.71
Conflict	0.73	Business	0.73
Justice	0.64	Crime	0.54
Politics	0.53	Terrorism	0.64
Life	0.56	Celebrity	0.39
Personnel	0.49	Death	0.39
Contact	0.36	Transportation	0.44
Nature	0.24	Natural disaster	0.52
Transaction	0.01	Health	0.42

Table 5: Error analysis: classification results (F1 per class) using Global2Specific embeddings.

7 Acknowledgments

The authors would like to thank Hannah Rashkin (University of Washington), Ian Stewart (Georgia Institute of Technology), Jacob Hunter, Josh Harrison, Eric Bell (PNNL) for their support and assistance with this project.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. *Proceedings of CoNNL* .

James Allan. 2012. *Topic detection and tracking: event-based information organization*, volume 12. Springer Science & Business Media.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042* .

David Bamman, Chris Dyer, and Noah A Smith. 2014. Distributed representations of geographically situated language. In *Proceedings of ACL*.

Ann Bies, Zhiyi Song, Jeremy Getman, Joe Ellis, Justin Mott, Stephanie Strassel, Martha Palmer, Teruko Mitamura, Marjorie Freedman, Heng Ji, and Tim O'Gorman. 2016. A comparison of event representations in deft. In *Workshop on Events: Definition, Detection, Coreference, and Representation*.

Joana Costa, Catarina Silva, Mário Antunes, and Bernardete Ribeiro. 2014. Concept drift awareness in twitter streams. In *Proceedings of ICMLA*. pages 294–299.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Proceedings of LREC*.

Anna Gladkova, Aleksandr Drozd, and Computing Center. 2016. Intrinsic evaluations of word embeddings: What can we do better? *Proceedings of ACL* .

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of ACL-IJCNLP*.

William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016a. Cultural shift or linguistic drift? comparing two computational measures of semantic change. *Proceedings of EMNLP* .

William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016b. Diachronic word embeddings reveal statistical laws of semantic change. *Proceedings of ACL* .

Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User review sites as a resource for large-scale sociolinguistic studies. In *Proceedings of WWW*. ACM, pages 452–461.

Chris Kedzie, Fernando Diaz, and Kathleen Mckeown. 2016. Real-Time Web Scale Event Summarization Using Sequential Decision Making. *Proceedings of IJCAI* .

Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal analysis of language through neural language models. *Proceedings of ACL* .

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *Proceedings of WWW*. pages 625–635.

Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2016. Freshman or fresher? quantifying the geographic variation of language in online social media. In *Proceedings of AAAI-WSM*.

Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA*. 4.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of ACL*. pages 302–308.

Donald Metzler, Congxing Cai, and Eduard Hovy. 2012. Structured event retrieval over microblog archives. *Proceedings of NAACL* pages 646–655.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionally. In *Proceedings of NIPS*. pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of CoNLL*.

Umashanthi Pavalanathan and Jacob Eisenstein. 2015. Confounds and consequences in geotagged twitter data. In *Proceedings of EMNLP*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*. volume 14, pages 1532–43.

Hannah Rashkin, Sameer Singh, and Yejin Choi. 2016. Connotation frames: A data-driven investigation. In *Proceedings of ACL*.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Workshop on New Challenges for NLP Frameworks*.

Alan Ritter, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from twitter. In *Proceedings of SIGKDD*.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of EMNLP*.

Philip A Schrodt. 2001. Automated coding of international event data using sparse parsing techniques. In *ISA*.