GELL: Automatic Extraction of Epidemiological Line Lists from Open Sources

Saurav Ghosh[1, 5], Prithwish Chakraborty[1, 5], Bryan L. Lewis[2], Maimuna S. Majumder[3,4], Emily Cohn[4], John S. Brownstein[4], Madhav V. Marathe[2, 5], Naren Ramakrishnan[1, 5]
[1] Discovery Analytics Center, Virginia Tech [2] Biocomplexity Institute, Virginia Tech
[3] Massachusetts Institute of Technology [4] Boston Children's Hospital
[5] Dept. of Computer Science, Virginia Tech

ABSTRACT

Real-time monitoring and responses to emerging public health threats rely on the availability of timely surveillance data. During the early stages of an epidemic, the ready availability of *line lists* with detailed tabular information about laboratory-confirmed cases can assist epidemiologists in making reliable inferences and forecasts. Such inferences are crucial to understand the epidemiology of a specific disease early enough to stop or control the outbreak. However, construction of such line lists requires considerable human supervision and therefore, difficult to generate in real-time. In this paper, we motivate Guided Epidemiological Line List (**GELL**), the first tool for building automated line lists (in near real-time) from open source reports of emerging disease outbreaks. Specifically, we focus on deriving epidemiological characteristics of an emerging disease and the affected population from reports of illness. **GELL** uses distributed vector representations (ala word2vec) to discover a set of indicators for each line list feature. This discovery of indicators is followed by the use of dependency parsing based techniques for final extraction in tabular form. We evaluate the performance of **GELL** against a human annotated line list provided by HealthMap corresponding to MERS outbreaks in Saudi Arabia. We demonstrate that **GELL** extracts line list features with increased accuracy compared to a baseline method. We further show how these automatically extracted line list features can be used for making epidemiological inferences, such as inferring demographics and symptoms-to-hospitalization period of affected individuals.

CCS CONCEPTS

• **Computing methodologies → Information extraction**;

KEYWORDS

Automated Line Listing, GELL, Word Embeddings, Dependency Parsing, Negation Detection

1 INTRODUCTION

An epidemiological line list [7, 13] is a listing of individuals suffering from a disease that describes both their demographic details as well as the timing of clinically and epidemiologically significant events during the course of disease. These are typically used during outbreak investigations of emerging diseases to identify key features, such as incubation period, symptoms, associated risk factors, and outcomes. The ultimate goal is to understand the disease well enough to stop or control the outbreak. Ready availability of line lists can also be useful in contact tracing as well as risk identification of spread such as the spread of Middle Eastern Respiratory Syndrome (MERS) in Saudi Arabia or Ebola in West Africa.

Formats of line lists are generally dependent on the kind of disease being investigated. However, some interesting features that are common for most formats include demographic information about cases. Demographic information can include age, gender, and location of infection. Depending on the disease being investigated, one can consider other addendums to this list, such as disease onset features (onset date, hospitalization date and outcome date) and clinical features (comorbidities, secondary contact, animal contact).

Traditionally, line lists have been curated manually and have rarely been available to epidemiologists in near-real time. Our primary objective is to automatically generate line lists of emerging diseases from open source reports such as WHO bulletins [22] and make such lists readily available to epidemiologists. Previous work [7, 13] has shown the utility in creating such lists through labor intensive human curation. We now seek to automate much of this effort. To the best of our knowledge, our work is the first to automate the creation of line lists.

The availability of massive textual public health data coincides with recent developments in text modeling, including distributed vector representations such as word2vec [14, 15] and doc2vec [8]. These neural network based language models when trained over a representative corpus convert words to dense low-dimensional vector representations, most popularly known as word embeddings. These word embeddings have been widely used with considerable accuracy to capture linguistic patterns and regularities, such as vec(*Paris*) - vec(*France*) ≈ vec(*Madrid*) - vec(*Spain*) [11, 16]. A second development relevant for line list generation pertains to semantic dependency parsing, which has emerged as an effective tool for information extraction, e.g., in an open information extraction context [23], Negation Detection [1, 18, 21], relation extraction [2, 9] and event detection [17]. Given an input sentence, dependency parsing is typically used to extract its semantic tree representations where words are linked by directed edges called *dependencies*.

Building upon these techniques, we formulate Guided Epidemiological Line List (**GELL**), a novel framework for automatic extraction of line list from WHO bulletins [22]. **GELL** is guided in the sense that the user provides a seed indicator (or, keyword) for each line list feature to guide the extraction process. **GELL** uses neural

KDD '17, August 13-17, 2017, Halifax, NS, Canada

ACM ISBN 978-1-4503-4887-4/17/08…$15.00
https://doi.org/10.1145/3097983.3098073

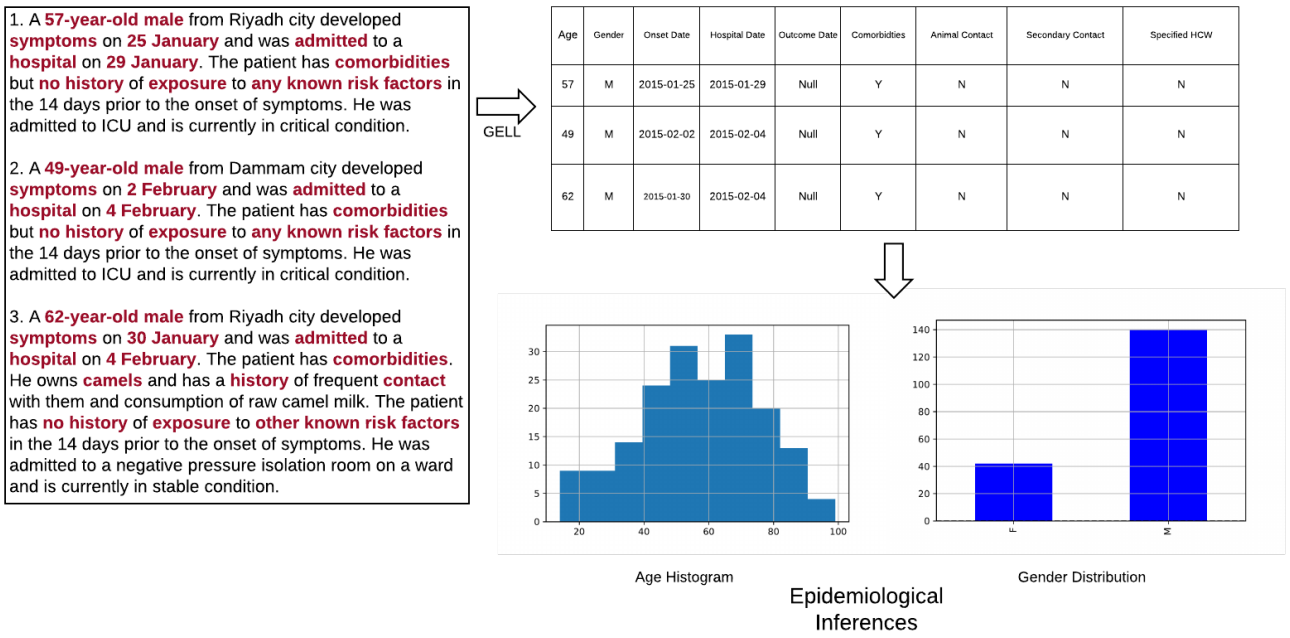


Figure 1: Tabular extraction of line list by GELL given a textual block of a WHO MERS bulletin. Each row in the extracted table depicts an infected case (or, patient) and columns represent the epidemiological features corresponding to each case. Information for each case in the table is then used to make epidemiological inferences, such as inferring demographic distribution of cases

word embeddings to expand the seed indicator and generate a set of indicators for each line list feature. The set of indicators is subsequently provided as input to dependency parsing based shortest distance and negation detection approaches for extracting line list features. As can be seen in Figure 1, **GELL** takes a WHO bulletin as input and outputs epidemiological line list in tabular format where each row represents a line list case and each column depicts the features corresponding to each case. The extracted line list provides valuable information to model the epidemic and understand the segments of population who would be affected.

Our main contributions are as follows.

- *Automated:* **GELL** is fully automatic, requiring no prior human annotation. Given a WHO bulletin, it will automatically extract the number of line list cases and the features corresponding to each case. The user only needs to provide a seed indicator for each feature to be extracted.
- *Novelty:* To the best of our knowledge, there has been no prior systematic efforts at tabulating such information automatically from publicly available health bulletins.
- *Real-time:* **GELL** can be deployed for extracting line list in a (near) real-time setting.
- *Evaluation:* We present a detailed and prospective analysis of **GELL** by evaluating the automatically inferred line list against a human curated line list for MERS outbreaks in Saudi Arabia. We also compare **GELL** against a baseline method.
- *Epidemiological inferences:* Finally, we also demonstrate some of the utilities of real-time automated line listing, such as inferring the demographic distribution and symptoms-to-hospitalization period.

2 PROBLEM OVERVIEW

In this manuscript, we intend to focus on Middle Eastern Respiratory Syndrome (MERS) outbreaks in Saudi Arabia [13] (2012-ongoing) as our case study. MERS was a relatively less understood disease when these outbreaks began. Therefore, MERS was poised as an emerging outbreak leading to good bulletin coverage about the infectious cases individually. This makes these disease outbreaks ideally suited to our goals. MERS is infectious as well and animal contact has been posited as one of the transmission mechanisms of the disease. For each line list case, we seek to extract automatically three types of epidemiological features as follows. (a) *Demographics:* Age and Gender, (b) *Disease onset:* onset date, hospitalization date and outcome date and (c) *Clinical features:* animal contact, secondary contact, comorbidities and specified healthcare worker (abbreviated as HCW).

In Figure 2, we show all the internal components comprising the framework of **GELL**. **GELL** takes multiple WHO MERS bulletins as input. The textual content of each bulletin is pre-processed by sentence splitting, tokenization, lemmatization, POS tagging, and date phrase detection using spaCy [6] and BASIS TechnologiesâĂŹ Rosette Language Processing (RLP) tools [20]. The pre-processing step is followed by three levels of modeling as follows. (a) Level 0 Modeling for extracting demographic information of cases, such as age and gender. In this level, we also identify the key sentences related to each line list case, (b) level 1 Modeling for extracting disease onset information and (c) level 2 Modeling for extracting clinical features. This is the final level of modeling in **GELL** framework. Features extracted at this level are associated with two labels:

Y or N. Therefore, modeling at this level combines neural word embeddings with dependency parsing-based negation detection approaches to classify the clinical features into Y or N. In the subsequent subsections, we will discuss each internal component of **GELL** in detail.

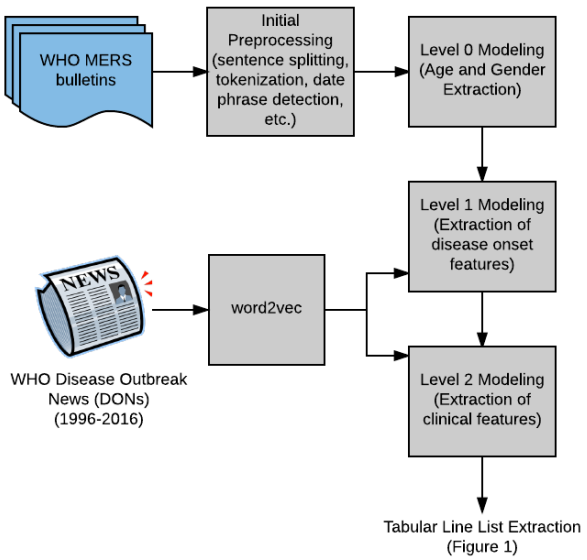


Figure 2: Block diagram depicting all components of the GELL framework. Given multiple WHO MERS bulletins as input, these components function in the depicted order to extract line lists in tabular form)

3 GELL

Given multiple WHO MERS bulletins as input, **GELL** proceeds through three levels of modeling for extracting line list features. We describe each level in turn.

3.1 Level O Modeling

In level 0 modeling, we extract the age and gender for each line list case. These two features are mentioned in a reasonably structured way and therefore, can be extracted using a combination of regular expressions as shown in Algorithm 1. One of the primary challenges in extracting line list cases is the fact that a single WHO MERS bulletin can contain information about multiple cases. Therefore, there is a need to distinguish between cases mentioned in the bulletin. In level 0 modeling, we make use of the age and gender extraction to also identify sentences associated with each case. Since age and gender are the fundamental information to be recorded for a line list case, we postulate that the sentence mentioning the age and gender will be the starting sentence describing a line list case (see the textual block in Figure 1). Therefore, the number of cases mentioned in the bulletin will be equivalent to the number of sentences mentioning age and gender information. We further postulate that information related to the other features (disease onset or critical) will be present either in the starting sentence or the sentences subsequent to the starting one not mentioning any age and gender related information ((see the textual block in Figure 1)). For more details on level 0 modeling, please see Algorithm 1. In Algorithm 1, $\mathcal{N}$ represents the number of line list cases mentioned in the bulletin and $\mathcal{SC}_n$ represents the set of sentences mentioning the n^{th} case.

Algorithm 1: Level 0 modeling

Input : set of sentences in the input WHO MERS bulletin
Output: Age and Gender for each line list case, index of the starting sentence for each case

```
n = 0;
SC_n = Null;
R_1 = \s+(?P<age>\d{1,2})(.{0,20})(\s+|-)(?P<gender>woman|man|male|female|boy|girl|housewife);
R_2 = \s+(?P<age>\d{1,2})\s*years?(\s|-)old;
R_3 = \s*(?P<gender>woman|man|male|female|boy|girl|housewife|he|she);
for each sentence in the bulletin do
    is-starting → 0;
    if R_1.match(sentence) then
        Age = int(R_1.groupdict()['age']);
        Gender = R_1.groupdict()['gender'];
        is-starting → 1;
    else
        if R_2.match(sentence) then
            Age = int(R_3.groupdict()['age']);
        else
            Age = Null;
        if R_3.match(sentence) then
            Gender = int(R_3.groupdict()['gender']);
        else
            Gender = Null;
        if Age ≠ Null && Gender ≠ Null then
            is-starting → 1;
    if is-starting then
        n += 1;
        SC_n = index of the sentence;
N = n;
```

3.2 WHO Word Embeddings

Before presenting the details of level 1 modeling and level 2 modeling, we will briefly discuss the process for providing WHO word embeddings as input to both these levels of modeling (see Figure 2). In this process, our main objective is to identify words which tend to share similar contexts or appear in the contexts of each other specific to the WHO bulletins (contexts of a word refer to the words surrounding it in a specified window size). For instance, consider the sentences $\mathcal{S}_1$ =*The patient had no contact with animals* and $\mathcal{S}_2$ =*The patient was supposed to have no contact with camels*. The terms *animals* and *camels* appear in similar contexts in both $\mathcal{S}_1$ and $\mathcal{S}_2$. Both the terms *animals* and *camels* are indicative of information pertaining to patient's exposure to animals or animal products.

Similarly, consider the sentences $\mathcal{S}_3$ =*The patient had an onset of symptoms on 23rd January 2016* and $\mathcal{S}_4$ =*The patient developed symptoms on 23rd January 2016*. The terms *onset* and *symptoms* are indicators for the onset date feature and both of them appear in similar contexts or contexts of each other in $\mathcal{S}_3$ and $\mathcal{S}_4$.

For generating word-embeddings, neural network inspired word2vec models are ideally suited to our goals because these models work on the hypothesis that words sharing similar contexts or tending to appear in the contexts of each other have similar embeddings. In recent years, word2vec models based on the skip-gram architectures [14, 15] have emerged as the most popular word embedding

models for information extraction tasks [5, 10, 12]. We used two variants of skip-gram models: (a) the skip-gram model trained using the negative sampling technique (*SGNS* [15]) and (b) the skip-gram model trained using hierarchical sampling (*SGHS* [15]) to generate embeddings for each term in the WHO vocabulary $\mathcal{W}$. $\mathcal{W}$ refers to the list of all unique terms extracted from the entire corpus of WHO Disease Outbreak News (DONs) corresponding to all diseases downloaded from http://www.who.int/csr/don/archive/disease/en/. The embeddings for each term in $\mathcal{W}$ were provided as input to level 1 modeling and level 2 modeling as shown in Figure 2.

3.3 Level 1 Modeling

The level 1 modeling is responsible for extracting the disease onset features, such as symptom onset date, hospitalization date and outcome date for each linelist case, say the n^{th} case. For extracting a given disease onset feature, the level 1 modeling takes three inputs: (a) seed indicator for the feature, (b) the word embeddings generated using *SGNS* or *SGHS* for each term in the WHO vocabulary $\mathcal{W}$ and (c) $\mathcal{SC}_n$ representing the set of sentences describing the n^{th} case for which we are extracting the feature.

Growth of seed indicator. In the first phase of level 1 modeling, we discover the top-K similar (or, closest) indicators to the seed indicator for each feature using WHO word embeddings. The similarity metric used is the standard cosine similarity metric. Therefore, we expand the seed indicator to create a set of $K+1$ indicators for each feature. In Table 1 we show the indicators discovered by *SGNS* for each disease onset feature given the seed indicators as input.

Table 1: Seed indicator and the discovered indicators using word embeddings generated by *SGNS*

Features	Seed indicator	Discovered indicators
Onset date	onset	symptoms, symptom, prior, days, dates
Hospitalization date	hospitalized	admitted, screened, hospitalised, passed, discharged
Outcome date	died	recovered, passed, became, ill, hospitalized

Shortest Dependency Distance. In the second phase, we use these $K+1$ indicators to extract the disease onset features. For each indicator $\mathcal{I}_t \forall t \in 1, 2, \ldots, K+1$, we identify the sentences mentioning $\mathcal{I}_t$ by iterating over each sentence in $\mathcal{SC}_n$. Then, for each sentence mentioning $\mathcal{I}_t$, we discover the shortest path along the undirected dependency graph between $\mathcal{I}_t$ and the date phrases mentioned in the sentence. Subsequently, we calculate the length of the shortest path as the number of edges encountered while traversing along the shortest path. The length of the shortest path is referred to as the *dependency distance*. E.g., consider the sentence $\mathcal{S}_5$ = *He developed symptoms on 4-June and was admitted to a hospital on 12-June.* The sentence $\mathcal{S}_5$ contains the date phrases *4-June* and *12-June*. $\mathcal{S}_5$ also contains the indicator *symptoms* for onset date and *admitted* for hospitalization date (see Tables 1). In Figure 3, we show the undirected dependency graph for $\mathcal{S}_5$. We observe that the *dependency distance* from *symptoms* to *4-June* is 3 (*symptoms* → *developed* → *on* → *4-June*) and *12-June* is 4 (*symptoms* → *developed* → *admitted* → *on* → *12-June*). Similarly, the dependency distance

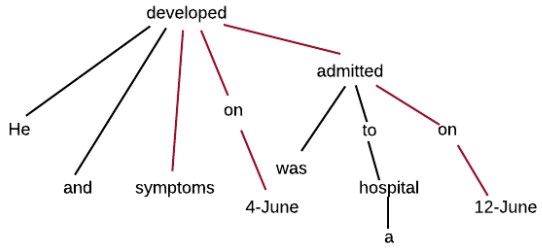


Figure 3: Undirected dependency graph corresponding to $\mathcal{S}_5$. The red-colored edges depict those edges included in the shortest paths between the date phrases (*4-June, 12-June*) and the indicators (*symptoms, admitted*)

from *admitted* to *4-June* is 3 (*admitted* → *developed* → *on* → *4-June*) and *12-June* is 2 (*admitted* → *on* → *4-June*). Therefore, for each indicator we extract a set of date phrases and the dependency distance corresponding to each date phrase. The output value of the indicator is set to be the date phrase located at the shortest dependency distance. E.g., in $\mathcal{S}_5$, the output values of *symptoms* and *admitted* will be *4-June* and *12-June* respectively. The final output for each disease feature is obtained by performing majority voting on the outputs of the indicators. For more algorithmic details, please see Algorithm 2.

Algorithm 2: Level 1 modeling

```
Input  : seed indicator, word embeddings for each term in W, SC_n
Output : date phrase
Growth of seed indicator using word embeddings to generate K + 1 indicators represented
  as I_t ∀t ∈ 1, 2, ..., K + 1;
for each I_t do
    dependency-dist = dict(); empty dictionary
    for each sentence in SC_n do
        check the mention of I_t;
        if I_t found then
            Identify the date phrases mentioned in the sentence;
            if at least one date phrase is found then
                construct the undirected dependency graph for the sentence (see
                  Figure 3);
                for each date phrase in the sentence do
                    dependency-dist[date phrase] = dependency distance (see
                      section 3.3);
            else
                continue;
        else
            continue;
    Output of I_t = date phrase in dependency-dist having the shortest dependency
      distance;
final output = majority voting on the outputs of each I_t;
```

3.4 Level 2 Modeling

The level 2 modeling is responsible for extracting the clinical features for each line list case. Extraction of clinical features is a binary classification problem where we have to classify each feature into two classes - *Y* or *N*. The first phase of level 2 modeling is similar to level 1 modeling. Seed indicator for each clinical feature is provided as input to the level 2 modeling and we extract the $K+1$ indicators

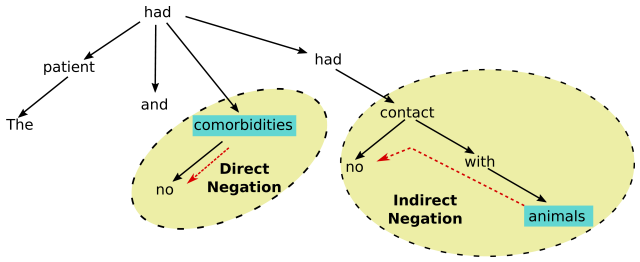


Figure 4: Directed dependency graph corresponding to $\mathcal{S}_6$ showing direct and indirect negation detection

for each such feature by discovering the top-K most similar indicators to the seed indicator (in terms of cosine similarities) using WHO word embeddings.

Dependency based negation detection. In the second phase, we make use of the $K+1$ indicators extracted in the first phase and a static lexicon of negation cues [3], such as *no*, *not*, *without*, *unable*, *never*, etc. to detect negation for a clinical feature. If no negation is detected, we classify the feature as *Y*, otherwise *N*. For each indicator $\mathcal{I}_t \forall t \in 1, 2, \ldots, K+1$, we identify the first sentence (referred to as $\mathcal{S}_{\mathcal{I}_t}$) mentioning $\mathcal{I}_t$ by iterating over the sentences in $\mathcal{SC}_n$. Once $\mathcal{S}_{\mathcal{I}_t}$ is identified, we perform two types of negation detection on the directed dependency graph $\mathcal{D}_{\mathcal{I}_t}$ constructed for $\mathcal{S}_{\mathcal{I}_t}$.

Direct Negation Detection: In this negation detection, we search for a negation cue among the neighbors of $\mathcal{I}_t$ in $\mathcal{D}_{\mathcal{I}_t}$. If a negation cue is found, then the output of $\mathcal{I}_t$ is classified as *N*.

Indirect Negation Detection. Absence of a negation cue in the neighborhood of $\mathcal{I}_t$ drives us to perform indirect negation detection. In this detection, we locate those terms in $\mathcal{D}_{\mathcal{I}_t}$ for which $\mathcal{D}_{\mathcal{I}_t}$ has a directed path from each of these terms as source to $\mathcal{I}_t$ as target. We refer to these terms as the predecessors of $\mathcal{I}_t$ in $\mathcal{D}_{\mathcal{I}_t}$. Then, we search for negation cues in the neighborhood of each predecessor. If we find a negation cue around a predecessor, we assume that the indicator $\mathcal{I}_t$ is also affected by this negation and we classify the output of $\mathcal{I}_t$ as *N*. For example, consider the sentence $\mathcal{S}_6$ =*The patient had no comorbidities and had no contact with animals.* and the directed dependency graph corresponding to $\mathcal{S}_6$ is shown in Figure 4. Sentence $\mathcal{S}_6$ contains the seed indicators *comorbidities* for comorbidities and *animals* for animal contact. In Figure 4, we observe direct negation detection for comorbidities as the negation cue *no* is located in the neighborhood of the indicator *comorbidities*. However, for animal contact, we observe indirect negation detection as the negation cue *no* is situated in the neighborhood of the term *contact* which is one of the predecessors of the indicator *animals*.

Therefore, for a clinical feature we have $K+1$ indicators and the classification output *Y* or *N* from each indicator. The final output for a feature is obtained via majority voting on the outputs of the indicators.

4 EXPERIMENTAL EVALUATION

In this section, we first provide a brief description of our experimental setup, including the models for automatic extraction of line lists, human annotated line lists, accuracy metric and parameter settings.

Algorithm 3: Level 2 modeling

Input : seed indicator, word embeddings for each term in $\mathcal{W}$, negation cues, $\mathcal{SC}_n$
Output: *Y* or *N*

Growth of seed indicator using word embeddings to generate $K+1$ indicators represented as $\mathcal{I}_t \forall t \in 1, 2, \ldots, K+1$;
for *each* $\mathcal{I}_t$ **do**
  Iterate over each sentence in $\mathcal{SC}_n$ and identify the first sentence $\mathcal{S}_{\mathcal{I}_t}$ mentioning I_t;
  Construct the directed dependency graph $\mathcal{D}_{\mathcal{I}_t}$ (see Figure 4) for $\mathcal{S}_{\mathcal{I}_t}$;
  $N_{\mathcal{I}_t}$ = set of terms connected to $\mathcal{I}_t$ in $\mathcal{D}_{\mathcal{I}_t}$, i.e. neighbors of $\mathcal{I}_t$;
  $P_{\mathcal{I}_t}$ = predecessors of $\mathcal{I}_t$ in $\mathcal{D}_{\mathcal{I}_t}$;
  Isnegation ← 0;
  if $N_{\mathcal{I}_t}$ *has a negation cue* **then**
    output of $\mathcal{I}_t$ = *N*;
    Isnegation ← 1;
    break;
  else
    Iterate over each term in $P_{\mathcal{I}_t}$ and seach for a negation cue in the neighborhood;
    if *negation cue found in neighborhood of a predecessor* **then**
      output of $\mathcal{I}_t$ = *N*;
      Isnegation ← 1;
      break;
  if ¬*Isnegation* **then**
    output of $\mathcal{I}_t$ = *Y*;
final output = majority voting on the outputs of each $\mathcal{I}_t$;

4.1 WHO corpus

The WHO corpus used for generating the WHO word embeddings (see Figure 2) was downloaded from http://www.who.int/csr/don/archive/disease/en/. The corpus contains outbreak news articles related to a wide range of diseases reported during the time period 1996 to 2016. The textual content of each article was pre-processed by sentence splitting, tokenization and lemmatization using spaCy [6]. After pre-processing, the WHO corpus was found to contain 35,485 sentences resulting in a vocabulary $\mathcal{W}$ of 4447 words.

4.2 Models

We evaluated the following automated line listing models.

- **GELL (SGNS)**: Variant of **GELL** with *SGNS* used as the word2vec model for generating WHO word embeddings.
- **GELL (SGHS)**: Variant of **GELL** with *SGHS* used as the word2vec model for generating WHO word embeddings.
- **Baseline**: Baseline model which does not use WHO word embeddings to expand the seed indicator in order to generate $K+1$ indicators for each feature. Therefore, **Baseline** uses only a single indicator (seed indicator) to extract line list features.

4.3 Human annotated line list

We evaluated the line list extracted by the automated line listing models against a human annotated line list for MERS outbreaks in Saudi Arabia. To create the human annotated list, patient and outcome data for confirmed MERS cases were collected from the MERS Disease Outbreak News (DONs) reports of WHO [22] and curated into a machine-readable tabular line list. In the human annotated list, total number of confirmed cases were 241 curated from 64 WHO bulletins reported during the period October 2012 to February 2015. Some of these 241 cases have missing (null) features (see Figure 1). In Figure 5, we show the distribution of non-null

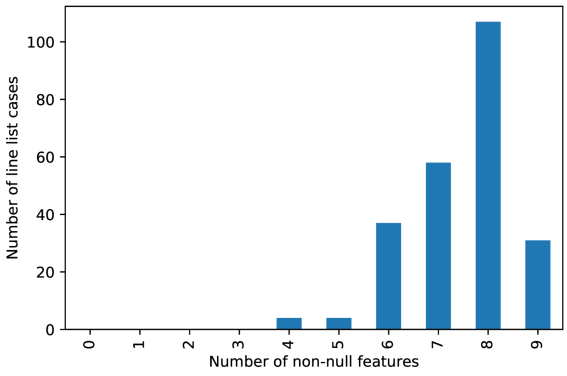


Figure 5: Distribution of non-null features in the human annotated line list

features in the human annotated list. We observe that majority of human annotated cases have at least 6 (out of 9) non-null features with the peak of the distribution at 8.

4.4 Accuracy metric

Matching automated line list to human annotated list. For evaluation, the problem is: we are given a set of automated line list cases and a set of human annotated cases for a single WHO MERS bulletin. Our strategy is to costruct a bipartite graph [20] where (i) an edge exists if the automated case and the human annotated case is extracted from the same WHO bulletin and (ii) the weight on the edge denotes the quality score (QS). Quality score (QS) is defined as the number of correctly extracted features in the automated case divided by the number of non-null features in the human annotated case. We then construct a maximum weighted bipartite matching [20]. Such matchings are conducted for each WHO bulletin to extract a set of matches where each match represents a pair (automated case, human annotated case) and is also associated with a QS. Once the matches are found for all the WHO bulletins, we computed the average QS by averaging the QS values across the matches.

Once the average QS and QS for each match are computed, we also computed the accuracy for each line list feature. For the demographic and disease onset features, we computed the accuracy classification score using scikit-learn [19] by comparing the automated features against the human annotated features across the matches. The clinical features are associated with two classes - Y and N (see Figure 1). For each class, we computed the F1-score using scikit-learn [19] where F1-score can be interpreted as a harmonic mean of the precision and recall. F1-score reaches its best value at 1 and worst score at 0. Along with the F1-score for each class, we also report the average F1-score across the two classes.

4.5 Parameter settings

GELL (*SGNS*) and **GELL** (*SGHS*) uses WHO word embeddings to generate $K + 1$ indicators for the line list columns. Therefore, these two models inherit the parameters of skip-gram based word2vec techniques, such as dimensionality, window size, negative samples, etc. as shown in Table 5. Apart from the word2vec parameters, **GELL** also inherits the parameter K which refers to the $K + 1$ indicators for disease onset or clinical features (see Section 3). In Table 5, we provide the list of all parameters, the explored values for each parameter and the applicable models corresponding to each parameter. We selected the optimal parameter configuration for each model based on the maximum average QS value as well as maximum average of the individual feature accuracies across the matches.

5 RESULTS

In this section we try to ascertain the efficacy and applicability of **GELL** by investigating some of the pertinent questions related to the problem of automated line listing.

Multiple indicators vs single indicator - which is the better method for automated line listing?

As mentioned in section 4, **GELL (SGNS)** and **GELL (SGHS)** uses multiple indicators discovered by word2vec, whereas the baseline **Baseline** uses only the seed indicator to infer line list features. We executed our automated line listing models taking as input the same set of 64 WHO MERS bulletins from which 241 human annotated line list cases were extracted. In Table 2, we observe that the number of automated line list cases (198) and the matches (182) after maximum bipartite matching is same for all the models. This is due to the reason that level 0 modeling (age and gender extraction) is the common modeling component in all the models and the number of extracted line list cases depends on the age and gender extraction (see section 3). In Table 2, we also compared the average QS achieved by each model. We observe that **GELL (SGNS)** is the best performing model achieving an average QS of 0.74 over **GELL (SGHS)** (0.71) and **Baseline** (0.67). To further validate the results in Table 2, we also show the QS distribution for each model in Figure 6 where x-axis represents the QS values and the y-axis represents the number of automated line list cases having a particular QS value. For **Baseline**, the peak of QS distribution is at 0.62. However, for **GELL (SGNS)** and **GELL (SGHS)**, the peak of the distribution is at 0.75. We further observe that **GELL (SGNS)** extracts higher number of line list cases with a perfect QS of 1 in comparison to **Baseline**.

We also compared the models on the basis of individual accuracies of the line list features across the matches in Tables 3 and 4. In Table 3, all the models achieve similar performance for the demographic features since level 0 modeling is similar for all the models (see section 3). However, for the disease onset features, both **GELL (SGNS)** and **GELL (SGHS)** outperform the baseline achieving an average accuracy of 0.45 and 0.43 in comparison to **Baseline** (0.12) respectively. **GELL (SGNS)** is the best performing model for onset date. However, for hospitalization date and outcome date, **GELL (SGHS)** is the better performing model than **GELL (SGNS)**. In Table 4, for the clinical features, we observe that **GELL (SGNS)** performs better than **GELL (SGHS)** and **Baseline** for comorbidities and specified HCW on the basis of average F1-score. Specifically, for specified HCW, **GELL (SGNS)** outperforms **GELL (SGHS)** and **Baseline** for the minority class *Y*. For animal contact, **GELL (SGHS)** emerges out to be the best performing model in terms of average F1-score, specifically outperforming

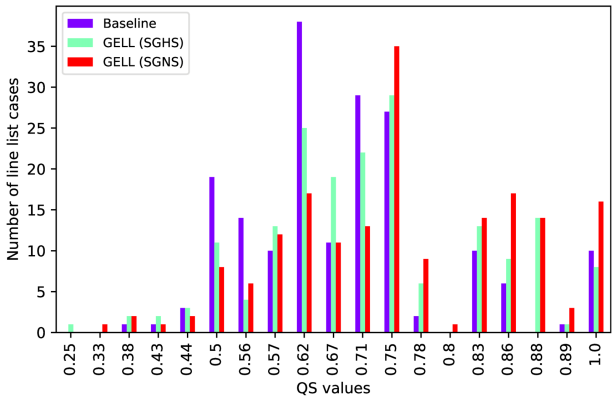


Figure 6: Distribution of QS values for each automated line listing model corresponding to MERS line list in Saudi Arabia. X-axis represents QS values and Y-axis represents the number of automated line list cases having a particular QS value

the competing models for the minority class Y. **Baseline** only performs better for secondary contact, even though the performance for the minority class Y is almost similar to **GELL (SGHS)** and **GELL (SGNS)**. Overall, we can conclude from Table 4 that **GELL** employing multiple indicators discovered via *SGNS* or *SGHS* shows superior performance than **Baseline** in majority of the scenarios, specifically for the minority class of each clinical feature.

Table 2: Average Quality Score (QS) achieved by each automated line listing model for MERS line list in Saudi Arabia. As can be seen, GELL (SGNS) shows best performance achieving an average QS of 0.73

Models	Human lists	Auto lists	Matches	Average QS
Baseline	241	198	182	0.67
GELL (*SGHS*)	241	198	182	0.71
GELL (*SGNS*)	241	198	182	**0.74**

What are beneficial parameter settings for automated line listing?

To identify which parameter settings are beneficial for automated line listing, we looked at the best parameter configuration (see Table 5) of **GELL (SGNS)** and **GELL (SGHS)** which achieved the accuracy values in Tables 2, 3 and 4. In Table 5, we explored the standard settings of each word2vec parameter (dimensionality of word embeddings, window size, negative samples and training iterations) in accordance with previous research [12]. Regarding dimensionality of word embeddings, **GELL (SGHS)** prefers 600 dimensions, whereas **GELL (SGNS)** prefers 300 dimensions. For the window size, both the models seem to benefit from smaller-sized (5) context windows. Most sentences in WHO corpus contain information about multiple columns, therefore relevant contexts of

Table 3: Comparing the automated line listing models based on the accuracy score for the demographics and disease onset features. For the disease onset features, GELL (SGNS) emerges out to be the best performing model. However, for the demographic features, all the models achieve almost similar performance

Feature type	Features	**Baseline**	**GELL** (*SGHS*)	**GELL** (*SGNS*)
Demographics	Age	0.87	**0.91**	0.87
	Gender	**0.99**	0.98	0.97
	Average	0.93	**0.95**	0.92
Disease onset	Onset date	0.01	0.01	**0.37**
	Hospitalization date	0.11	**0.63**	0.62
	Outcome date	0.48	**0.66**	0.36
	Average	0.20	0.43	**0.45**

Table 4: Comparing the performance of the automated line listing models for extracting clinical features corresponding to MERS line list in Saudi Arabia. We report the F1-score for class Y, class N and average F1-score across the two classes. For animal contact, GELL (SGHS) emerges out to be the best performing model. For comorbidities and specified HCW, GELL (SGNS) shows best performance. However, for secondary contact, Baseline achieve superior performance in comparison to GELL

Clinical Feature (Y:N)	Class	**Baseline**	**GELL** (*SGHS*)	**GELL** (*SGNS*)
Animal contact (1:3)	Y	0.33	**0.68**	0.37
	N	0.87	**0.91**	0.88
	Average	0.60	**0.79**	0.63
Secondary contact (1:3)	Y	**0.57**	0.52	0.56
	N	**0.86**	0.70	0.72
	Average	**0.71**	0.61	0.64
Comorbidities (2:1)	Y	0.52	0.52	**0.81**
	N	0.56	0.54	**0.61**
	Average	0.54	0.53	**0.71**
Specified HCW (1:6)	Y	0.26	0.35	**0.44**
	N	**0.95**	0.93	0.90
	Average	0.61	0.64	**0.67**

indicators are in their immediate vicinities leading to smaller window sizes. The number of negative samples is applicable only for **GELL (SGNS)** where it seems to prefer a single negative sample. Finally, for the training iterations, both the models benefit from more than 1 training iteration. This is expected as the WHO corpus used for generating WHO word embeddings (see section 4) is a smaller-sized corpus with a vocabulary of only $\mathcal{W} = 4447$ words. In such scenarios, word2vec models (*SGNS* or *SGHS*) generate improved embeddings with higher number of training iterations. Finally, both the models are also associated with the parameter K which refers to the number of indicators $K + 1$ used for extracting the disease

onset and clinical features. As expected, the models prefer at least 5 indicators, along with the seed indicator to be used for automated line listing. Using higher number of indicators increases the chance of discovering an informative indicator for a line list feature.

Table 5: Parameter settings in GELL (SGNS) and GELL (SGHS) for which both the models achieve optimal performance in terms of average QS and individual feature accuracies corresponding to MERS line list in Saudi Arabia. Non-applicable combinations are marked by *NA*

Models	Dimensionality (300:600)	Window size (5:10:15)	Negative samples (1:5:15)	Training Iterations (1:2:5)	Indicators (K = 3:5:7)
GELL (*SGHS*)	600	5	*NA*	5	7
GELL (*SGNS*)	300	5	1	2	5

Which indicator keywords discovered using word2vec contribute to the improved performance of GELL?

Next, we investigate the informative indicators discovered using word2vec which contribute to the improved performance of **GELL (SGNS)** or **GELL (SGHS)** in Tables 3 and 4. In Figure 7, we show the accuracies (or, average F1-score) of individual indicators (including the seed indicator) corresponding to the best performing model for a particular line list feature. Regarding onset date (see Figure 7a), **GELL (SGNS)** is the best performing model and the seed indicator provided as input is *onset*. We observe that *symptoms* is the most informative indicator achieving an accuracy of 0.36 similar to the overall accuracy (see Table 3). Rest of the indicators (including the seed indicator) achieve negligible accuracies and therefore, do not contribute to the overall performance of **GELL (SGNS)**. Similarly, for hospitalization date with the seed keyword *hospitalization* provided as input, *admitted* emerges out to be most informative indicator followed by the seed indicator, *hospitalised* and *treated* (see Figure 7b). Finally, for the outcome date, *died* (seed indicator) and *passed* are the two most informative indicators as observed in Figure 7c.

Regarding the clinical features, we show the average F1-score of individual indicators. For animal contact, the seed indicator provided as input is *animals*. We observe in Figure 7d that the most informative indicator for animal contact is *camels* followed by indicators such as *animals* (seed), *sheep* and *direct*. This shows that contact with *camels* is the major transmission mechanism for MERS disease. The informative indicators found for comorbidities are *patient*, *comorbidities* and *history*. Finally, regarding specified HCW, the informative indicators discovered are *healthcare* (seed), *tracing* and *intensive*.

Does indirect negation detection play a useful role in extracting clinical features?

In level 2 modeling for extracting clinical features, both direct and indirect negation detection are used. For more details, please see section 3. To identify if indirect negation detection contributes positively, we compared the performance of **GELL** with and without indirect negation detection for each clinical feature in Table 6 by reporting the F1-score for each class as well as average F1-score. We observe that indirect negation detection has a positive effect on the performance for animal contact and secondary contact. However, for comorbidities and specified HCW, indirect negation detection plays an insignificant role.

Table 6: Comparing the performance of GELL on extraction of clinical features with or without indirect negation for MERS line list in Saudi Arabia. It can be seen that indirect negation improves the performance of GELL for animal contact and secondary contact.

Clinical Feature	Class	Direct Negation	Direct + Indirect Negation
Animal contact	Y	0.56	**0.63**
	N	0.80	**0.90**
	Average	0.68	**0.77**
Secondary contact	Y	**0.55**	0.54
	N	0.65	**0.72**
	Average	0.60	**0.63**
Comorbidities	Y	**0.86**	0.82
	N	**0.64**	0.62
	Average	**0.75**	0.72
Specified HCW	Y	**0.44**	**0.44**
	N	**0.90**	**0.90**
	Average	**0.67**	**0.67**

What insights can epidemiologists gain about the MERS disease from automatically extracted line lists?

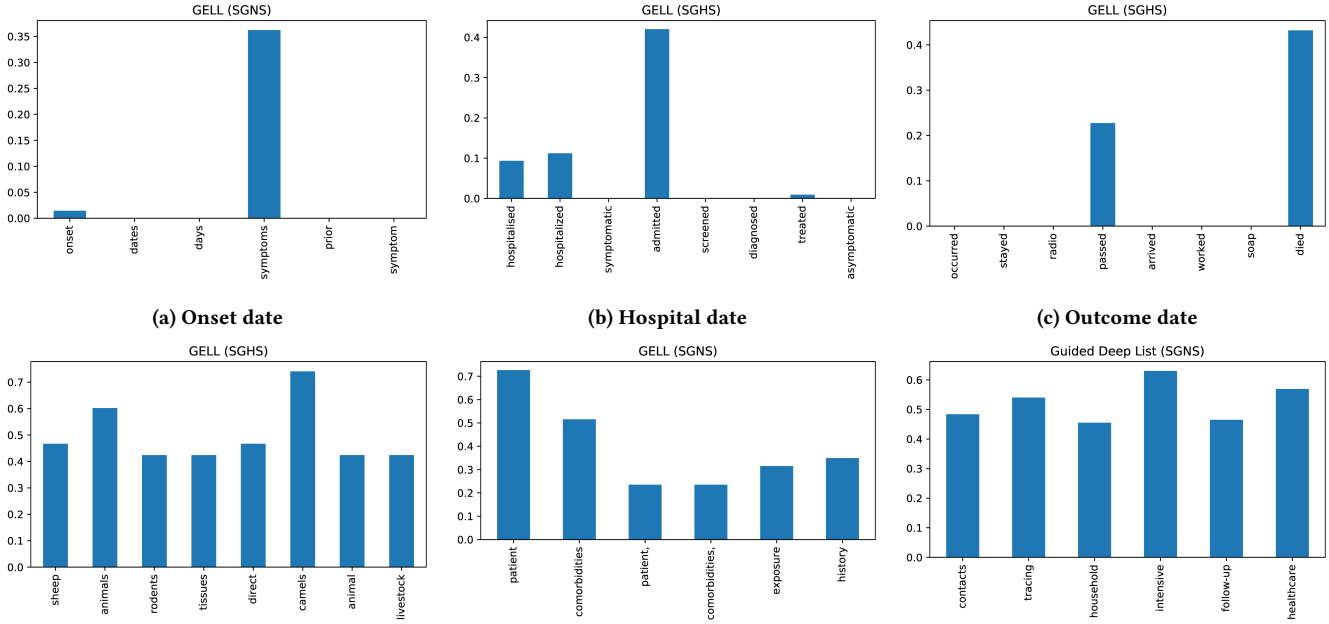
Finally, we show some of the utilities of automated line lists by inferring different epidemiological insights from the line list extracted by **GELL**.

Demographic distribution. In Figure 1, we show the age and gender distribution of the affected individuals in the extracted line list. We observe that males are more prone to getting infected by MERS rather than females. This is expected as males have a higher probability of getting contacted with infected animals (animal contact) or with each other (secondary contact). Also individuals aged between 40 and 70 are more prone to getting infected as evident from the age distribution.

Analysis of disease onset features. We analyzed the symptoms-to-hospitalization period by analyzing the difference (in days) between onset date and hospitalization date in the extracted line list as shown in Figure 8a. We observe that most of the affected individuals with onset of symptoms got admitted to the hospital either on the same day or within 5 days. This depicts a prompt responsiveness of the concerned health authorities in Saudi Arabia in terms of admitting the individuals showing symptoms of MERS. In Figure 8b, we also show a distribution of the hospitalization-to-outcome period (in days). Interestingly, we see that the distribution has a peak at 0 which indicates that most of the infected individuals admitted to the hospital died on the same day indicating high fatality rate of MERS case.

6 CONCLUSIONS AND FUTURE WORK

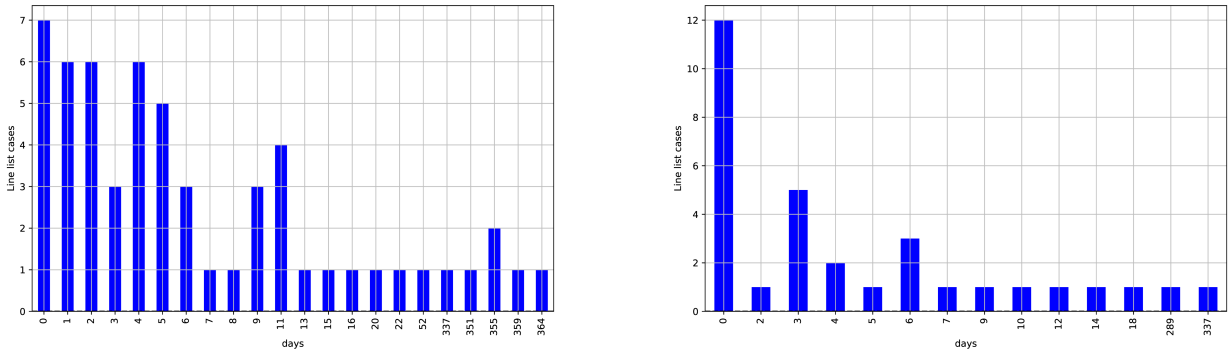
In this manuscript, we have introduced **GELL**, the first automated framework for building epidemiological line lists from open source reports of emerging diseases. **GELL** uses word2vec techniques (*SGNS* or *SGHS*) to discover multiple indicators for each line list feature and these indicators were subsequently used to guide dependency parsing based shortest distance and negation detection approaches for final feature extraction. We demonstrated the superior performance of **GELL (SGNS)** and **GELL (SGHS)**, specifically for disease onset and clinical features by comparing it against a baseline model **Baseline** which doesn't use any word embedding



(a) Onset date **(b) Hospital date** **(c) Outcome date**

(d) Animal contact **(e) Comorbidities** **(f) Specified HCW**

Figure 7: Accuracy of individual indicators (including the seed indicator) discovered via word2vec methods in GELL (SGNS) or GELL (SGHS) for each line list feature. For clinical features, we show the average F1-score. This figure depicts the informative indicators (indicators showing higher accuracies or F1-scores) which contribute to the improved performance of GELL (SGNS) or GELL (SGHS) for a particular feature. E.g. for animal contact, the most informative indicator contributing to the superior performance of GELL (SGHS) is *camels* followed by *animals* (seed), *sheep* and *direct*



(a) Symptoms-to-hospitalization period distribution

(b) Hospitalization-to-outcome period distribution

Figure 8: Analysis of disease onset features in the extracted line list

model and only utilizes the seed indicator to extract line list features. Our results showed that relative performance improvement of **GELL** over **Baseline** is dependent on the discovery of informative indicators using word2vec.

Our future work will focus on adapting **GELL** to extracting line lists from highly unstructured open sources (compared to WHO) such as HealthMap [4] using advanced NLP techniques, such as CRFs and LSTMs specifically for negation detection in Level 2 Modeling. Moreover, we aim to extract line lists using **GELL** for modeling other emerging diseases, such as Ebola, H7N9 at different geographical regions of the world.

ACKNOWLEDGEMENTS

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D12PC000337, the US Government is authorized to reproduce and distribute reprints of this work for Governmental purposes notwithstanding any copyright annotation thereon. This work has also been partially supported by DTRA CNIMS and DTRA BSVE (contract number HDTRA1-11-D-0016-0005), National Science Foundation grant NRT-DESE-154362, NSF DIBBS Grant ACI-1443054, NIH MIDAS Grant 5U01GM070694, NSF BIG DATA Grant IIS-1633028 and the National Institutes of Health grant 1R01GM109718.

SUPPLEMENTARY CODE AND DATA

The codes and datasets associated with this paper can be found in https://github.com/sauravcsvt/KDD_linelisting. Confusion matrices supporting the results in Table 4 can be found in https://github.com/sauravcsvt/KDD_linelisting/tree/master/data/confusion_matrix.

REFERENCES

[1] M. Ballesteros, A. Díaz, V. Francisco, P. Gervás, J. C. De Albornoz, and L. Plaza. 2012. UCM-2: a rule-based approach to infer the scope of negation via dependency parsing. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 288–293.

[2] R. C. Bunescu and R. J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, 724–731.

[3] A. Diaz, M. Ballesteros, J. Carrillo-de Albornoz, and L. Plaza. 2012. *UCM at TREC-2012: Does negation influence the retrieval of medical reports?* Technical Report. DTIC Document.

[4] Clark C Freifeld, Kenneth D Mandl, Ben Y Reis, and John S Brownstein. 2008. HealthMap: global infectious disease monitoring through automated classification and visualization of Internet media reports. *Journal of the American Medical Informatics Association* 15, 2 (2008), 150–157.

[5] S. Ghosh, P. Chakraborty, E. Cohn, J. S. Brownstein, and N. Ramakrishnan. 2016. Characterizing Diseases from Unstructured Text: A Vocabulary Driven Word2vec Approach. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 1129–1138.

[6] M. Honnibal and M. Johnson. 2015. An Improved Non-monotonic Transition System for Dependency Parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 1373–1378. https://aclweb.org/anthology/D/D15/D15-1162

[7] E. HY. Lau, J. Zheng, T. K. Tsang, Q. Liao, B. Lewis, J. S. Brownstein, S. Sanders, J. Y. Wong, S. R. Mekaru, C. Rivers, et al. 2014. Accuracy of epidemiological inferences based on publicly available information: retrospective comparative analysis of line lists of human cases infected with influenza A (H7N9) in China. *BMC medicine* 12, 1 (2014), 88.

[8] Q. V. Le and T. Mikolov. 2014. Distributed Representations of Sentences and Documents.. In *ICML*, Vol. 14. 1188–1196.

[9] O. Levy and Y. Goldberg. 2014. Dependency-Based Word Embeddings.. In *ACL (2)*. 302–308.

[10] O. Levy and Y. Goldberg. 2014. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the ACL*. 302–308. http://aclweb.org/anthology/P/P14/P14-2050.pdf

[11] O. Levy and Y. Goldberg. 2014. Linguistic Regularities in Sparse and Explicit Word Representations. In *Proceedings of the Eighteenth Conference on CoNLL*. 171–180. http://aclweb.org/anthology/W/W14/W14-1618.pdf

[12] O. Levy, Y. Goldberg, and I. Dagan. 2015. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *TACL* 3 (2015), 211–225. https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/570

[13] M. S. Majumder, C. Rivers, E. Lofgren, and D. Fisman. 2014. Estimation of MERS-coronavirus reproductive number and case fatality rate for the spring 2014 Saudi Arabia outbreak: insights from publicly available data. *PLOS Currents Outbreaks* (2014).

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). http://arxiv.org/abs/1301.3781

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *26th Annual Conference on Neural Information Processing Systems*. 3111–3119.

[16] T. Mikolov, W. Yih, and G. Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In *Human Language Technologies: Conference of the NAACL*. 746–751. http://aclweb.org/anthology/N/N13/N13-1090.pdf

[17] S. Muthiah, B. Huang, J. Arredondo, D. Mares, L. Getoor, G. Katz, and N. Ramakrishnan. 2015. Planned Protest Modeling in News and Social Media.. In *AAAI*. 3920–3927.

[18] Y. Ou and J. Patrick. 2015. Automatic negation detection in narrative pathology reports. *Artificial intelligence in medicine* 64, 1 (2015), 41–50.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[20] N. Ramakrishnan, P. Butler, S. Muthiah, N. Self, R. Khandpur, P. Saraf, W. Wang, J. Cadena, A. Vullikanti, G. Korkmaz, et al. 2014. 'Beating the news' with EMBERS: Forecasting civil unrest using open source indicators. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1799–1808.

[21] S. Sohn, S. Wu, and C. G. Chute. 2012. Dependency parser-based negation detection in clinical narratives. *AMIA Summits on Translational Science proceedings AMIA Summit on Translational Science* 2012 (2012), 1–8.

[22] WHO. 2016. Coronavirus infections: Disease Outbreak News. (2016). http://www.who.int/csr/don/archive/disease/coronavirus_infections/en/

[23] F. Wu and D. S. Weld. 2010. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 118–127.