Identifying Early Warning Signals from News Using Network Community Detection

Nataliya Le Vine,[1] Eric Boxer,[1] Mustafa Dinani,[2,3] Paolo Tortora,[4] Subhradeep Das[5]

[1]Advanced Analytics, Swiss Re, USA,
[2,]Life & Health Products, Swiss Re, USA,
[3]Synergy Health Advisors, USA,
[4]Data Engineering, Swiss Re, Switzerland,
[5]Data Engineering, Swiss Re, India
{nataliya_levine, eric_boxer paolo_tortora, subhradeep_das}@swissre.com, mdinani@synergyadvisors.ai

Abstract

The paper addresses the challenge of accelerating identification of changes in risk drivers in the insurance industry. Specifically, the work presents a method to identify significant news events (”signals”) from batches of news data to inform Life & Health insurance decisions. Signals are defined as events that are relevant to a tracked risk driver, widely discussed in multiple news outlets, contain novel information and affect stakeholders. The method converts unstructured data (news articles) into a sequence of keywords by employing a linguistic knowledge graph-based model. Then, for each time window, the method forms a graph with extracted keywords as nodes and draws weighted edges based on keyword co-occurrences in articles. Lastly, events are derived in an unsupervised way as graph communities and scored for the requirements of a signal: relevance, novelty and virality. The methodology is illustrated for a Life & Health topic using news articles from Dow Jones DNA proprietary data set, and assessed against baselines on a publicly available news data set. The method is implemented as an analytics engine in Early Warning System deployed at Swiss Re for the last 1.5 years to extract relevant events from live news data. We present the system's architectural design in production and discuss its use and impact.

Introduction

Many businesses are challenged with a problem of how to navigate increasing amounts of news information in a consistent way to extend what is known to their experts. Specifically, this paper investigates identifying significant news developments for Life & Health (L&H) (re-)insurance, where current processes for detecting changes in risk drivers are heavily reliant on experts siloed across the business or experience studies that could be significantly lagged. Identifying relevant signals faster will enable appropriate actions to mitigate risk or capitalize on opportunities.

Currently, L&H experts track changes in risk drivers manually, this includes setting news alerts based on keywords, searching articles on Google Scholar or PubMed, reading articles from selected reputable scientific journals, or from discussions with colleagues. This can miss important events, bring noise that happens to match search criteria, or focus on events that are still far away from impacting the business, such as early stages of drug development that can take many years before completion or turn out to be unsuccessful. Further, identified events cannot be easily shared and/or discussed with other stakeholders as groups of experts and stakeholders are typically siloed. Some examples of L&H topics of interest to (re-)insurance include tracking new cancer screening medical advances which may lead to abrupt increases in claim volumes over a shorter time horizon, and/or decrease in mortality over a longer time horizon; tracking developments and approvals of expensive drugs, e.g. Zolgensma drug costing 2 million+ US dollars for a treatment course (this can lead to extremely high drug claims); or tracking volatile developments in local regulations during the COVID-19 pandemic. Tracking important events allows for timely changes to underwriting as well as providing recommendations to our insurance clients, adjusting reserving, and staying up to date with regulatory changes.

Defining news relevancy and importance to an industry is a complex problem of translating human expertise and intuition into numeric metrics, as, for example, has been pointed out in an IAAI emerging application paper (Wang et al. 2020). Following rounds of discussions and feedback with our L&H experts, we conceptualize a signal as an event with the following characteristics: 1) relevant to a selected risk driver (topic), 2) contains new information, 3) discussed in multiple (reputable) news outlets, and 4) affects company's business. We introduce metrics to quantify the first three requirements, and provide a feedback feature in our system that enables experts and stakeholders to assess news for the fourth criteria. The metrics add explainability to identified events (”Why do I see this event?”) and allow users to react to the events most relevant to them (”How can I prioritize?”).

In this paper, we describe a graph-based event detection methodology that 1) uses a linguistic knowledge graph-based model to extract important concepts from a batch of news, 2) conducts an iterative graph community detection, and 3) scores the corresponding events using the three significance criteria described above. We illustrate the methodology using news articles from Dow Jones DNA proprietary data set and compare to baselines using publicly available data. We also describe the architecture and use of the deployed early warnings system that ingests, scores, displays

identified signals and collects feedback from users.

Related Work

The signal detection problem is similar to the important or significant event detection problem. Earlier approaches to the problem focus on sets of terms or keywords that exhibit anomalous (bursty) behavior over a time window (Ge et al. 2016; Stilo and Velardi 2016; Mele, Bahrainian, and Crestani 2019; Fung et al. 2005; Weng and Lee 2011). Such approaches calculate anomaly scores for each term's occurrence counts over time to identify an abnormal behavior (i.e. event). These approaches are usually critiqued for missing co-associations among terms and for not examining how relations among terms change over time.

More recent approaches fall into the category of graph-based event detection, where events are detected by building a graph representing relations among terms and extracting cohesive sets from the graph (Bonchi et al. 2016; Crescenzo et al. 2017). Some challenges in this approach are related to defining what constitutes a cohesive set (sub-graph), how to extract meaningful representative terms, and how to score the derived events. Our graph-based methodology attempts to addresses some of these challenges as highlighted below.

When composing a graph, some studies use full articles as graph nodes (Sidorov et al. 2018), while others use keywords extracted defined via TF-IDF scoring (Sayyadi, Hurst, and Maykov 2009), a supervised keyword labelling (Liu et al. 2017), named entities (Moutidis and Williams 2019), and word groups (Liu et al. 2020). Our method utilizes a knowledge graph-based linguistic model that has been specifically extended to include concepts from the medical domain.

Then, the keywords are arranged on a graph using weighted (Moutidis and Williams 2019; Sayyadi, Hurst, and Maykov 2009) and unweighted (Liu et al. 2017) edges based on the corresponding node co-occurrence and frequency of occurrence in documents; the node connections further can be pruned based on edge weight or node co-occurrence metrics (Sayyadi, Hurst, and Maykov 2009; Liu et al. 2017). In our work, we consider several edge weighting options that account for keyword co-occurrence, in-document keyword proximity and frequency; we also allow pruning based on in-document keyword proximity.

Further, events can be determined by extracting graph communities (Moutidis and Williams 2019; Sidorov et al. 2018), by iteratively removing edges with high "betweenness" scores (Liu et al. 2017), by depth search for graph connectivity components (Sidorov et al. 2018). We use the Infomap community detection algorithm (Rosvall, Axelsson, and Bergstrom 2009) that allows moving iteratively from coarse to fine community structure in the network.

A detected event can be reported as a small set of representative keywords (Mele and Crestani 2019), or as the document most similar to a detected community (Liu et al. 2017). We select articles that cover all nodes of identified community in order to account for potentially multiple event aspects, so that each article contains at least a pre-specified number of the community nodes.

Lastly, some studies provide scores for their events by scoring story recency (Liu et al. 2017), legal entity emergence (Moutidis and Williams 2019), and assign compatibility scores when events are inserted into a storyline (Liu et al. 2017). To reflect our stakeholder preferences, we use the three scores for each event by classifying event virality, relevance and novelty described in the previous section.

Data Description

The data powering the developed system is a live news feed from Dow Jones (DJ) DNA. While we will be focusing on news in English language, we also experimented with analyzing news from other languages in the data set (e.g. Chinese) by employing an automatic translation service (Baidu translation API). Further, we also experimented with other news data sets, such as news data set from Thomson Reuters and LexisNexis, and we were unable to identify stronger business insights for our stakeholders (Life & Health (re-)insurance) by using a different data corpus. This might change for a different line of business.

Furthermore, we benchmark our method on a subset from Mele and Crestani (Mele and Crestani 2019). Their full publicly available data set consists of news articles, RSS feeds and tweets in English. We only utilize the news article subset from the data due to the specific format of the other data types, as tweets and RSS feeds will likely require significant adjustments to the methodology.

Dow Jones DNA The news data set draws together over 8,500 licensed sources in 28 languages containing relevant metadata with continuous data delivered via a stream API. This content is licensed for text-mining and machine-learning use cases. Each news article is labeled with several fields classifying article contents (metadata) such as:

- news subjects (e.g. international politics, medical treatment);
- geographical locations/region (e.g. North America, UK, Rome);
- industry types (e.g. pharmaceutical, agricultural);
- mentioned or highly relevant to a given article companies/organizations (e.g. Amazon, UN);
- mentioned people (e.g. Barack Obama).

The full list of fields can be found on Factiva Snapshot API developer's page. In the paper, we will illustrate the methodology using article subset covering medical developments for COVID-19, extracted from DJ DNA data based on expert-defined keywords and requirements on DJ metadata.

Benchmarking Data Set Labelled news data set in Mele and Crestani (Mele and Crestani 2019) sourced from 9 different news outlets, e.g. ABC, BBC, CBC, NBC, from March 1 to June 30, 2016 consists of around 24K news documents. Each document has publishing timestamp, source, and content. Mele and Crestani (Mele and Crestani 2019) tokenized content into keywords (raw content is not available) by lower-casing, removing stop words, URLs, tokens not starting with alphabet letters, punctuation marks, and less-frequent words. Further, Mele and Crestani (Mele and Crestani 2019) randomly selected events for the time period, and used a crowd sourcing platform to determine relevance

of the news articles to the events. Each event is labelled as a list of words defining the event, e.g. ['britain', 'brexit', 'european', 'leave', 'vote', 'referendum']. We use a subset of about 1.5K news articles with corresponding 56 labelled events, one event less than labelled for the entire data set (that includes tweets and RSS feeds).

Mele et al. (Mele, Bahrainian, and Crestani 2019) used the data set for event detection and tracking; and we benchmark our approach to the performance of four methods reported in their work (Blei and Lafferty 2006; Fung et al. 2005; Weng and Lee 2011). In the study, the authors extended the labelled event set to 60 (from 57); however, we did not have access to the additional 3 events to include into the method assessment. Following the study, we assess event detection quality using recall only with respect to the labelled event set (we use 56 events from the news subset, the study used 60 events), as the entire set of events is unknown.

Method

The signal detection process requires (**Figure 1**):

1. Life & Health business experts identify a topic of interest (e.g. cancer screening) and define key search phrases (e.g. Liquid Biopsy, MRI, mammography) to be used for news filtering.
2. Data scientists create and refine a data query that pulls relevant news article set for the specified topic; this is an iterative process requiring contributions from both data scientists and business/medical experts.
3. The selected news articles are enriched with metadata – important article keywords, mentioned named entities, domains the article words belong to.
4. The metadata are used by a signal detection algorithm to identify and score events.
5. Identified signals are visualized via interactive dashboards that enable fast search, detailed drill-throughs, as well as feedback from business and subject experts.

The 'signal' is defined as relevant, novel, viral event affecting company's business (see Introduction section). News articles are scored to satisfy the first three criteria for a signal, and the highest scoring articles are further assessed for the fourth criteria by company's experts. The scoring algorithm works with connected networks of concepts/keywords extracted from articles published during different time windows. The method extracts communities, scores them using heaviness, virality and novelty scores described in the section below and selects representative articles to provide a narrative representative of the community nodes.

Figure 1: Signal detection process schematic.

Data Enrichment

We worked closely with Expert.AI to customize their proprietary NLU software Cogito to process news articles. The developed linguistic model detects relevant entities in an article by exploiting information coming from a proprietary knowledge graph (KG) Sensigrafo and textual analysis offered by a customized disambiguator. Generic Sensigrafo KG contains about 400K concepts for English language and almost 2 billion connections between them. Manual examination of hundreds of keyword extractions indicated that the generic KG has to be enriched with medical knowledge to extract meaningful medical terms and allow constructing meaningful event graphs. We customized the KG for Life & Health domain by expanding it with medical domain knowledge from selected UMLS (Unified Medical Language System) subtrees that tripled the number of concepts in the KG.

When applied to a news article, the Cogito model extracts the following relevant descriptors of the text: linguistically significant (proprietary scoring algorithm) n-grams normalized to their base word form, a list of mentioned medical diseases, legal entities, geographical entities, domains for article words (e.g. medicine, sports), and frequent article words. A selected subset of the extractions is used as article "keywords" (e.g. top 20 n-grams and frequent words), while other extractions are used for optional clustering purposes e.g. disease or domain. Extensive manual testing showed that keywords extracted by the model are superior to frequency-based (or tf-idf) n-gram extraction, as the extracted keywords represent meaningful concepts from the knowledge graph (e.g. "Food and Drug Administration").

Construction of Networks

At a high level, we are creating our network using the co-occurrence of concepts in articles during a time window. We choose a width for our time windows and a sliding step (length of time between the starting dates of consecutive windows) based on article experimentation and expert prior knowledge about the speed at which topics develop in the domain of interest. We can use a sliding step shorter than the width of our windows to have overlapping windows or use a sliding step and window width of equal length to have disjoint windows.

For a given time window, we create a network whose nodes are concepts (aggregated keywords) present in articles published during that time window. Keywords are aggregated into "concepts" using embeddings from a Sent2Vec model (Gupta, Pagliardini, and Jaggi 2019); the simple average of token-level embeddings is used for keywords composed of multiple tokens. The Sent2Vec model is trained on a corpus of articles similar in domain to the domain that we analyze during the final scoring (e.g. medical, regulatory). The clustering of keyword embeddings is carried out using a hierarchical clustering algorithm from the package fastcluster (Müllner et al. 2013). A height at which to cut the hierarchical clustering dendrogram is determined by inspection of cluster membership (e.g. does it make sense for our purposes that the keyword "mammogram" is part of the concept "screening"?), the number of dendrogram leaves, and the

number of singleton clusters; and then most frequent keyword in a cluster is selected as a representative keyword.

After aggregating from the set of keywords into a reduced set of concepts, these concepts are used to simplify the articles in the data set, so that edges between concepts can be defined. There are several ways how edges and their weights can be assigned, and we discuss a few of them below.

Let $\mathbf{C} = \{c_i\}$ denote the set of all concepts in the data set and $\mathbf{A} = \{A_i\}$ the set of all articles. Then, let $f : \mathbf{A} \to \mathbf{S}$ denote the mapping from articles to sequences of concepts $\mathbf{S} = \{\mathrm{S_i}\}$, such that $f(A_i) = S_i = \{c_j,\ c_k, \ldots\}$ where S_i is the sequence of concepts formed from A_i. Then the network time series is a sequence $\mathbf{G} = \{G_t\}_{t\in[0,T]}$, such that $G_t = (C_t,\ E_t)$ where $C_t \subset \mathbf{C}$ denotes the set of concepts present in articles published during time window t (our nodes) and $E_t \subset \mathbf{C}\times\mathbf{C}$ denotes the set of edges. We define a weighting function $w_t : \mathbf{C} \times \mathbf{C} \to \mathbb{R}$ on our edges at time t, $w_t(e) = \sum_{A_i\in\mathbf{A}_t} w_t^i(e)$, where $e \in E_t$, $\mathbf{A}_t$ denotes the set of articles published during time window t, and w_t^i is an edge weight based on article A_i. All edge weighting schemes we experimented with first create edge weights from each article in the time window, and then take a sum over articles in the window to arrive at final edge weights. Below, we define functions $w_t^i : \mathbf{C} \times \mathbf{C} \to \mathbb{R}$ for edge weights from a given article A_i during time window t. Note that we do not allow edges with coinciding source and target nodes (self-edges).

Given an article A_i, $S_i = f(A_i)$ and two distinct concepts $c_j,\ c_k \in S_i$, the simplest edge weight can be defined as

$$w_t^i(e_{jk}) = w_t^i((c_j, c_k)) = \begin{cases} 1, & \text{if } |c_j, c_k|_{S_i} < d \\ 0, & \text{otherwise} \end{cases},$$

where d is a hyperparameter representing the maximum distance allowed between two concepts to create an edge, and $|.,.|_{S_i} : \mathbf{C} \times \mathbf{C} \to \mathbb{N}$ is a symmetric function defined for sequence of concepts S_i that returns the minimum distance in S_i between the two considered concepts.

When defining an edge weight, we can also account for proximity of concepts in an article by using the inverse distance between pairs of concepts, so that an edge receives a higher weight if its respective concepts are located closer together in an article. Further, we can account for multiple appearances of concepts in an article by taking a weighted sum accounting for number of appearances of two concepts in a sequence, so that edge weight gets close to 1 when both concepts tend to co-occur next to each other in text.

Community Identification

Once we have networks for each time window, we search for node communities in each graph. After experimenting with several network clustering algorithms (e.g. Louvain method (Blondel et al. 2008), maximum clique algorithms (Boppana and Halldórsson 1992)), we decided to use the Infomap algorithm (Rosvall, Axelsson, and Bergstrom 2009). Infomap optimizes the map equation, a measure of the ability of a network partition to separate flow on the network. Flow is simulated by random walks and the optimization is carried out via a fast stochastic recursive search algorithm. Infomap has a tuning hyper-parameter, called Markov-time

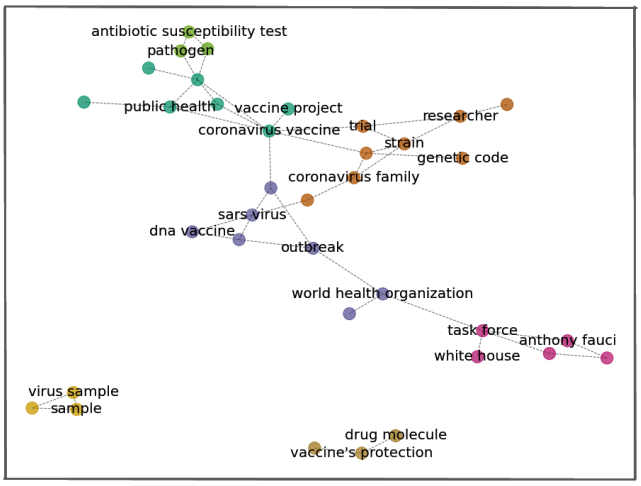

Figure 2: An example of community detection (given by different colors) on a subgraph for COVID-19. Some text labels are not shown to improve readability.

(Kheirkhahzadeh, Lancichinetti, and Rosvall 2016) that we leverage to move iteratively from coarse to fine community structure in the network. A community detection example is shown for COVID-19 graph in **Figure 2**, where different colors correspond to different communities

The iterative community refining process uses the following criteria to determine if a community is a potential signal: 1) minimum community size, 2) minimum community intersection with articles and 3) (optional) minimum community intersection with articles from reputable sources. In practice, we set minimum community size to three nodes. Minimum community intersection with articles refers to a threshold set on the proportion of a community's nodes that must intersect with concepts from a single article. This is intended to ensure that our signals are not so large or disparate that they cannot be encapsulated by a single article. In practice, this hyperparameter is set in the range $[0.3, 0.5]$.

We define a reputable source as a source publishing at least a certain percentage of its articles with a given DJ metadata tag. For example, we may define a reputable medical source as the one that publishes at least 40% of its articles under the "health care/life sciences" DJ industry tag. The criterion 3) is implemented as a minimum number of articles from reputable sources that must intersect with detected community, and/or a minimum number of nodes from reputable articles that must intersect the community.

If a community passes all the above criteria, it is considered a candidate signal and its nodes are removed from the graph for later iterations in that time window. If a community fails any of the criteria 2) or 3), it is not considered a candidate signal, and its nodes are retained for clustering with a lower Markov time. Meanwhile, if a community fails the minimum size criterion 1), it is not considered a candidate signal and its nodes will be removed from the graph.

Once we have assembled candidate signals, we define article membership in them needed when presenting output to users. We say that an article A_i belongs to a candidate sig-

nal $sign_j$, if $|A_i \cap sign_j| \geq \max(m,\ |sign_j| * r)$, where m is the minimum number of candidate signal nodes that must belong to an article, and r is a minimum proportion of such candidate signal nodes. In practice, m is set in the range $[1, 4]$ and r is set in the range $[0.2, 0.6]$.

Scoring Communities of Keywords

Let $\mathbf{Sign} = \{Sign_i\}$ denote the set of all signals in our data set, where signals are sets of concepts with a sense of ownership over articles. Let $gn : \mathbf{Sign} \rightarrow \sigma(\mathbf{C})$ denote a mapping from signals to their underlying concepts/nodes, where $\sigma(X)$ denotes the sigma-algebra of X (which can be thought of as the set of subsets of X). $gua : \mathbf{Sign} \rightarrow \sigma(\mathbf{A})$ denotes a mapping from signals to their underlying articles. $ge : \mathbf{Sign} \rightarrow \mathbf{C} \times \mathbf{C}$ denotes a mapping from signals to their underlying edges and $\mathbf{W} = \{W_i\}_{i \in [0,T]}$ denotes the sequence of time windows for our network time series. Identified signals are scored on metrics defined below.

Heaviness In order to capture the most prominent and relevant signals, heaviness is defined as the sum of all edge weights contained in a signal. Heaviness can be defined as $H : \mathbf{Sign} \rightarrow \mathbb{R}$ such that for a given signal $Sign_i$

$$H(Sign_i) = \frac{1}{2} \sum_{c_j \in gn(Sign_i)} \sum_{c_k \in gn(Sign_i)} w_t(c_j, c_k),$$

where we divide the sum by 2 to account for double-counting of edges. Heaviness assigns high scores to signals that cover frequently (co-)mentioned concepts, and can be seen as a measure of signal relevance to a topic overall.

Virality We consider a signal to be viral if many news sources publish stories about it in a short time interval. In order to consider time intervals smaller than our time windows, we introduce the concept of time frames. Given time window W_t, we define $\Omega_t^k = \{\omega \in W_t \mid |\omega| = k\}$ as the collection of time frames of length k within W_t. We define function $ns(\omega,\ Sign_i)$ as the number of sources that published articles from set $gua(Sign_i)$ during ω, and $\omega_t^* = argmax_{\omega \in \Omega_t^k} ns(\omega,\ Sign_i)$ denotes the highest number of sources publishing during any single time frame during time window W_t. We formulate virality as $Vir : \mathbf{Sign} \rightarrow \mathbb{R}$ such that for a given signal $Sign_i$

$$Vir(Sign_i) = \alpha_1 * ns(\omega^*, Sign_i) + \alpha_2 * \frac{ns(\omega^*,\ Sign_i)}{ns(W_t,\ Sign_i)},$$

where α_1 and α_2 are hyperparameters that control the emphasis placed on the highest number of sources during a time frame (α_1) and its relative value with respect to the total number of sources during the time window (α_2). In practice, we set $\alpha_1 = 0.7$ and $\alpha_2 = 0.3$.

Keyword novelty Keyword Novelty is a measure of how novel the signal concepts are with respect to concepts from historical articles. To capture the distribution of concepts from the signal and from historical articles, for a given signal $Sign_i$ and concept c_j we define

$$p(c_j) = \frac{\text{count}(c_j,\ gua(Sign_i))}{\sum_{c_k \in \mathbf{C}} \text{count}(c_k,\ gua(Sign_i))} \text{ and}$$

$$q(c_j) = \frac{\text{count}(c_j,\ A_{past})}{\sum_{c_k \in \mathbf{C}} \text{count}(c_k,\ A_{past})}$$

where A_{past} denotes articles published in the year prior to W_t and $\text{count} : \mathbf{C} \times \sigma(\mathbf{A}) \rightarrow \mathbb{N}$ is a function that counts the appearances of a concept in a set of articles. Then, p and q capture the relative frequency of a concept within a signal during W_t and during the year prior to W_t, respectively. We define the set of concepts that appear in both the signal and the historical period as $C_{\hat{t},t}$ where $\hat{t}$ denotes the index of the time window one year prior to W_t. Finally, we compute keyword novelty as the cross-entropy between p and q over their shared support $C_{\hat{t},t}$, such that for a given signal $Sign_i$

$$Nov(Sign_i) = -\sum_{c_j \in C_{\hat{t},t}} p(c_j) * \log(q(c_j)).$$

Link Novelty Edges represent co-occurrence of concepts in an article, and we assign significance to the formation of unexpected edges, which can be formalized as a link prediction task in the literature. For this, we derive node embeddings using node2vec algorithm (Grover and Leskovec 2016), which assembles sequences of nodes via biased random walks over the network, and a link prediction model with 0 denoting the most novel edges and 1 denoting the least novel edges. We set values of the two node2vec hyperparameters that balance breadth-first with depth-first in random walks based on overall good link prediction method performance for both small and large networks, specifically we set $p = 0.5$ and $q = 2$. We found node2vec performing better than two factorization-based node embedding methods we tested - Locally Linear Embedding or LLE(Roweis and Saul 2000) and High-Order Proximity preserved Embedding or HOPE (Ou et al. 2016). Due to limited computational capacity, we opted not to use deep learning methods for link embeddings (such as (Goyal and Ferrara 2018)).

The link prediction model has limited generalizability, as it is trained on links from a limited set of articles from a particular period. To capture semantic information that might be missing from the network, we use pretrained word embeddings to supplement our link novelty scores. Specifically, we use GloVe (Pennington, Socher, and Manning 2014) embeddings for general information (common-sense), and BioWordVec (Zhang et al. 2019) embeddings for medical information, if we are working with medical-related articles.

Link novelty is defined as $link\_nov : \mathbf{C} \times \mathbf{C} \rightarrow [0, 1]$ such that for a given pair of concepts $c_i,\ c_j \in \mathbf{C}$, $link\_nov(c_i,\ c_j)$ is the maximum of trained link prediction model's probability and the other two semantic similarity scores (from GloVe and BioWordVec). Then, for a given signal $Sign_i$, we can assemble scores for all of its edges as a set $LN_{Sign_i} = \{link\_nov(c_i,\ c_j) | \ c_i,\ c_j \in ge(Sign_i)\ \bigwedge i < j\}$, and define link novelty $LinkNov : \mathbf{Sign} \rightarrow [0, 1]$ as an average of the three smallest elements in the LN_{Sign_i} set. We found that using several lowest link novelty scores from LN_{Sign_i} reduces metric volatility.

Aggregate Score To simplify working with the four scores developed for the keyword communities, we also introduce an aggregate score that accounts for signal relevance (heaviness), novelty (both node and link novelties) and propagation in the news media (virality). Let $qt : \mathbb{R} \rightarrow [0, 1]$ define the mapping from one of our real-valued metrics to its quantile over a run of the method.

$$AggScore(Sign_i) = f(qt(H(Sign_i))) + f(qt(Vir(Sign_i))) + f(\max[qt(Nov(Sign_i)), 1 - qt(LinkNov(Sign_i))])$$

where $f(.)$ is an optional threshold function defined as

$$f(x) = \begin{cases} x, & \text{if } x < th_0 \\ 1, & \text{otherwise} \end{cases},$$

where th_0 is a specified threshold; we tested 0.75 and 0.9 quantiles for this threshold. This function enables a cleaner identification of the most significant articles, i.e. when signal's $AggScore$ equals to 3, which is preferred by end users. Note that link novelty is the only metric for which we are most interested in signals with low scores (low quantiles), hence we use $1 - qt(LinkNov(Sign_i))$ in the formula.

Method Testing

As a full set of is unknown, we either employ a limited number of known historical signals or use expert feedback to assess the method. At this point, it is not clear how to assess the algorithms for the cases the algorithms failed to detect signals unknown to our experts (failing to detect unknowns).

Historical signals: When a topic is set up for tracking (e.g. topic of cancer screening), experts provide a small number of known historic "events" to configure a corresponding data query, so that stories covering the events are included into the data for signal detection, provided the news is present in the DJ DNA corpus. Occasionally, business or subject experts provide some potentially significant new events that we use to 1) refine the query if needed, and 2) assess if the method identifies the events with high scores. On all such occasions, the new events have been identified with high scores without any additional changes to queries or method.

Real-time detection: Experts regularly access automatically updated dashboards with scored signals and leave feedback on signal importance. We also ask subject experts to provide a more detailed feedback for selected topics (true/-false positives, prior awareness of an event). While a number of topics is being tracked internally, for illustrative purposes, we are providing such an assessment for one of the topics - see section "Manual Expert Scoring for COVID-19".

Manual Expert Scoring for COVID-19

To evaluate performance of the method, we asked an experienced medical expert to assess top extracted signals for medical developments in COVID-19; this required identifying whether information in the signals was important (using "Yes", "No", "Maybe"), and whether the information was completely new ("I knew about it" or "New information"). The presented signals were extracted from up-to-date news relevant to COVID-19 medical developments (drugs, vaccines, tests, new technology) for the period 1-Jan, 2020 until 6-May, 2020 (about 10K articles); so that the network model was trained on the news articles published prior to 1-April, and the signals were extracted for the 1-April till 6- May, 2020 (about 6.7K relevant articles). We only provided signals with the highest aggregate score (equals 3), so that the expert was evaluating 21 top news stories out of 6.7K articles in the set, completing the evaluation on 10- May, 2020.

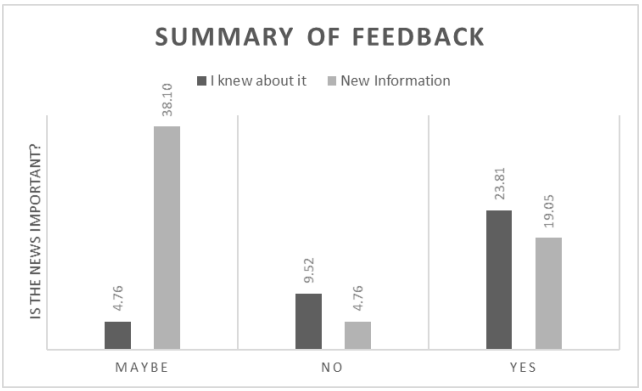

Figure 3: Summary of a medical expert's feedback for the "COVID-19 – Medical developments" topic.

Figure 3 shows the breakdown of the expert feedback for the topic, so that there are 3 signals (14.3%) that the expert did not see as important, and the other 18 signals (85.7%) considered important or potentially important. Further, 13 signals (61.9%) presented new information to the expert, and most of the novel signals (38% of all signals) were potentially important ("Maybe" tag) prompting the expert to follow the news as it develops. There was no obvious pattern identified in the three news stories deemed unimportant by the expert; two stories were covering new COVID-19 treatments, and one was covering medical community opinion on Donald Trump's administration pushing an unproven anti-malaria drug for approval to treat COVID-19.

Benchmarking on External Data

Since the data set provided by Mele and Crestani (Mele and Crestani 2019) contains only tokenized article content, we are unable to extract keywords using the described linguistic model, as it requires paragraphs with fully defined sentences. Instead, we extract keywords (up to 4-grams) with TextRank (Mihalcea and Tarau 2004) implemented in the PyTextRank library (Nathan 2016). Furthermore, to avoid bias introduced by a small number of sources (9 sources; while DJ DNA has 4K+ English language sources), we score articles using heaviness and novelty only (no virality score). Small number of sources leads to large step-like differences in virality score disproportionally affecting final scoring.

For each time window, the method builds concept graphs as well as detects and scores communities for each time window. The size of the window is to be defined by experts, or by investigating samples of historic events. As the data are not limited to a specific topic/industry, we note that the labelled events have multiple time scale, as also noted by (Mele, Bahrainian, and Crestani 2019), e.g. some events are

Method	Recall	Recall, %
This paper	51/56	91.07%
dDTM-News	55/60	91.67%
Unigram Co-Occurrences	44/60	73.33%
EDCoW	8/60	13.33%

Table 1: Method benchmarking with dDTM-News (Mele, Bahrainian, and Crestani 2019), Unigram Co-Occurances (Fung et al. 2005), EDCoWon (Weng and Lee 2011).

discussed for a small number of days, and some can evolve over several weeks. Therefore, we detect events at different time horizons (time windows): 1, 3, 7, 15 and 30 days.

To match a labelled event with detected events, first, we intersect a set of event keywords, provided as labels in (Mele and Crestani 2019), with tokens in each article in a detected event. Then, we select an event with the highest score from all matched per each labelled event, and record its score as an aggregate score quantile to be used when comparing among different time scales (note, this is different from the raw aggregate score in the Method section). Lastly, we manually inspect the matched events, and only record events that are representative of the labeled event storyline; otherwise, we perform manual matching. If an event is detected we report its highest score across the five time horizons.

The matching procedure identified 51 out of 56 labelled events as detected by the method; and **Table 1** compares this result to the three baselines taken from (Mele, Bahrainian, and Crestani 2019). Note, we are unable to compare the absolute numbers of matched events, as the total number of labelled events used by benchmarks is several events larger (see Data Description section). The five events that the algorithm missed are among the smallest labelled events (based on the labelled article count), however we are unable to precisely detect the number of relevant articles in the full corpus to further examine significance of the missed events. Furthermore, to illustrate event detection at different time scales, we provide some examples of events identified with different time windows:

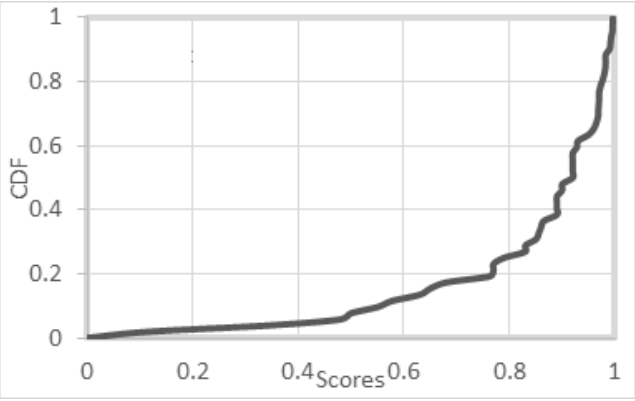

Figure 4: Distribution of the derived event scores (aggregate score quantiles) for the detected events.

- 1 day - ["brussels", "railway", "station", "evacuated", "suspicious", "suitcase"], a suspicious suitcase found at a Belgian railway station;
- 3 days - ["sharapova", "tennis", "doping", "drug", "test", "positive"], Maria Sharapova's positive drug test;
- 15 days - ["nightclub", "orlando", "shooting", "victims"], shooting in Orlando nightclub;
- 30 days - ["britain", "brexit", "european", "leave", "vote", "referendum"], coverage of Brexit referendum.

Event score is a measure of its significance, and stakeholders tend to closely examine only events with high scores. We report score distribution for the detected events in **Figure 4**; while the data set lacks any event scores to compare against. The figure shows that scores for 75% of all detected events are greater than 0.8 (80^{th} percentile); and that only 8% of articles have scores less than 0.5 (median). If used in production, users are likely to pay close attention to about 75% of labelled events with high scores.

System Deployment, Use and Maintenance

The Dow Jones data are live streamed into the system, and each additional news article is enriched with metadata by the Cogito service (path 1 in **Figure 5**). If required, document enrichment can be re-executed by re-processing a batch of historical articles saved locally (path 2 in figure). The enriched documents are indexed by Elasticsearch and accessible via API queries from the signal detection model and from a Kibana interface. Then, batches of data are pulled from the Elasticsearch storage using queries for each topic at a pre-specified frequency (e.g. daily, weekly, monthly). Events are identified and scored from the batches using the described method, stored in a file storage system (path 4 in figure), and picked up by a user-facing app (path 5 in figure). The app allows interactive experience (filtering by dates, scores, locations, clusters, etc. and search by keywords) and records user feedback. The system has been deployed at a corporate scale and has been processing the live news stream over the last 1.5 years.

The system, on average, processes 100K articles a day to extract a small number of events (the exact number depends on each tracked topic). The main goal is to complement manual information search and to bring different stakeholder groups together to enable timely decisions and actions. Experts and stakeholders triage identified events based on significance to company business, region affected and expected time line for business impact. The triaged events are then included into quarterly reports to summarize recommended actions to stakeholders. The process is enabling proactive business decisions and earlier (than by traditional methods) identification of risks, promoting growth, increased profitability and cost savings. Moreover, implementation and use of the early warning system strengthens the company's thought leadership position in the market.

Tracked topic examples include keeping up to date with the voluminous coverage of the COVID-19 pandemic that challenges manual processing protocols; this includes both volatile regulatory changes and advancements in medical

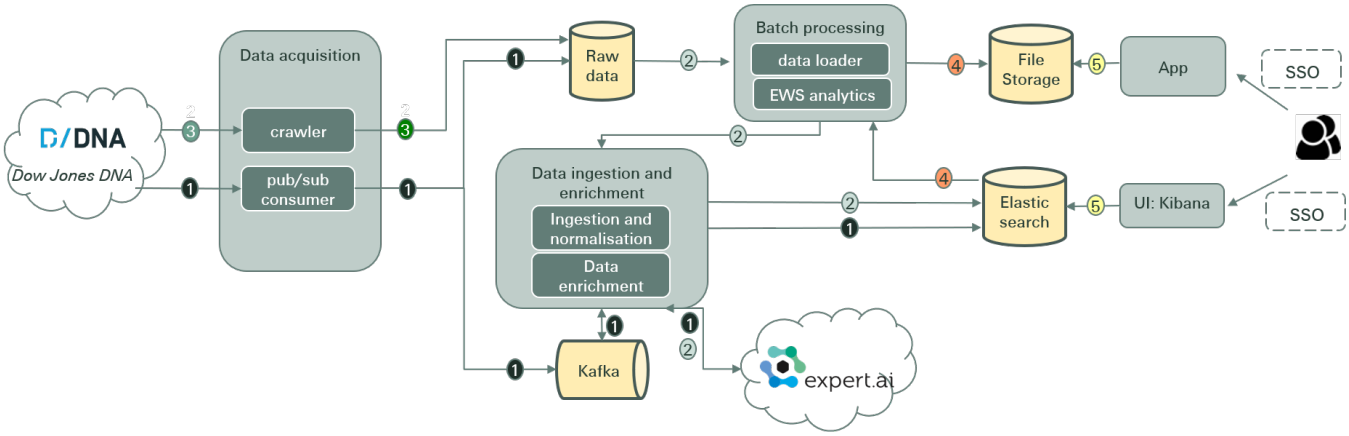

Figure 5: System design for early warning signal processing. 1 is a path for live data stream (continuous execution), 2 is a processing path of stored historical documents (executed on demand), 3 shows an initial path of a bulk historical corpus (executed once when the system is set up), 4 traces event detection steps, and 5 corresponds to user-facing app connections.

treatments and testing. Other topics cover medical and regulatory developments for high-mortality and/or costly diseases, such as different cancers and rare diseases. Signals detected by the system get shared with a selected group of insurance clients to identify the most time- and cost-efficient way to keep our clients (insurance companies) informed about significant developments in the industry.

In production, the system is able to provide about 50% of additional signals (significant events) per quarter to complement signals from expert manual search. Overall, our L&H product management estimates that timely detection of early warnings can, on average, save 200 million US dollars over a 3-year period by identifying relevant information faster and infilling gaps in manual search (due to confidentiality, we cannot provide more detailed examples of specific losses). Note, this is a soft number informed by past experience associated with delayed response in the L&H domain; this might not strictly apply to each 3-year period, so that some time periods might provide larger or smaller savings.

The deployed system requires periodic maintenance. For example, the knowledge graph underlying the Cogito model needs to be periodically (annually to bi-annually) updated to include new legal entities and medical terminology. Further, the topic queries need to be periodically adjusted to include missing information or reduce output noise. The collected feedback can help understanding whether scoring criteria need to be modified or refined, and it can help to train an additional machine learning-based scoring system that may help with a further signal prioritization.

Summary and Conclusions

The paper describes a methodology to identify changes in Life & Health risk drivers faster and more consistently than is currently done by a traditional process. The developed methodology requires business and/or medical understanding about the key risk drivers encoded as queries over a large news corpus (Dow Jones DNA). The retrieved data sets satisfying the query criteria are then analyzed to identify important events. The events are derived in an unsupervised way as graph communities and scored for the requirements of a signal: relevance, novelty and virality.

The method is illustrated on real-time data from Dow Jones and on a publicly available benchmark data set. For the purposes of this paper, a medical expert assessed signals produced by the method for medical developments related to COVID-19 and identified most (85.7%) of the signals as important, with more than half (61.9%) of all signals not known to the expert (new information). Due to the data set limitations, the method had to be adapted to the available benchmarking data by removing some of its strong features (e.g. linguistic keyword extraction, virality score), and it still performed on par or better (91.07% of events identified) than the three event detection baselines reported in the literature.

Future work will need to address the following challenges: 1) adding Knowledge Graph updates to the system to prevent keyword extraction quality deteriorating with time; 2) (semi-) automating inclusion of user feedback into queries and model parameters to reduce signal noise; 3) identifying unknown unknowns to track risk drivers that our experts are not aware of yet.

The method is implemented as an analytics engine in a corporate-scale system at Swiss Re (reinsurance company) connected to live news data stream, the system displays identified signals to stakeholders and records their feedback. The system is enabling proactive business decisions and earlier identification of the risks promoting company's growth, increased profitability and cost savings.

Acknowledgments

We thank experts and colleagues for contributions to the system development: Laura Gorrieri and Nico Lavarini from Expert.AI; Francesco Palma and Stefano Savaré from L2F; Urs Widmer, Nora Leonardi and Heather Sutherland from Life & Health, Swiss Re; Stefan Pero and Patrik Sagat from IBM; and our analytics colleagues Kongkuo Lu, Matt Zeigenfuse, Nanditha Nandy, Phoebe Sun and Boyi Xie.

References

Blei, D. M.; and Lafferty, J. D. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, 113–120.

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10): P10008.

Bonchi, F.; Bordino, I.; Gullo, F.; and Stilo, G. 2016. Identifying buzzing stories via anomalous temporal subgraph discovery. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 161–168. IEEE.

Boppana, R.; and Halldórsson, M. M. 1992. Approximating maximum independent sets by excluding subgraphs. *BIT Numerical Mathematics*, 32(2): 180–196.

Crescenzo, C. D.; Gavazzi, G.; Legnaro, G.; Troccoli, E.; Bordino, I.; and Gullo, F. 2017. HERMEVENT: a news collection for emerging-event detection. In *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics*, 1–10.

Fung, G. P. C.; Yu, J. X.; Yu, P. S.; and Lu, H. 2005. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, 181–192. Citeseer.

Ge, T.; Cui, L.; Chang, B.; Sui, Z.; and Zhou, M. 2016. Event detection with burst information networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 3276–3286.

Goyal, P.; and Ferrara, E. 2018. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151: 78–94.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864.

Gupta, P.; Pagliardini, M.; and Jaggi, M. 2019. Better word embeddings by disentangling contextual n-gram information. *arXiv preprint arXiv:1904.05033*.

Kheirkhahzadeh, M.; Lancichinetti, A.; and Rosvall, M. 2016. Efficient community detection of network flows for varying Markov times and bipartite networks. *Physical Review E*, 93(3): 032309.

Liu, B.; Han, F. X.; Niu, D.; Kong, L.; Lai, K.; and Xu, Y. 2020. Story Forest: Extracting Events and Telling Stories from Breaking News. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(3): 1–28.

Liu, B.; Niu, D.; Lai, K.; Kong, L.; and Xu, Y. 2017. Growing story forest online from massive breaking news. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 777–785.

Mele, I.; Bahrainian, S. A.; and Crestani, F. 2019. Event mining and timeliness analysis from heterogeneous news streams. *Information Processing & Management*, 56(3): 969–993.

Mele, I.; and Crestani, F. 2019. A multi-source collection of event-labeled news documents. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, 205–208.

Mihalcea, R.; and Tarau, P. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 404–411.

Moutidis, I.; and Williams, H. T. 2019. Utilizing complex networks for event detection in heterogeneous high-volume news streams. In *International Conference on Complex Networks and Their Applications*, 659–672. Springer.

Müllner, D.; et al. 2013. fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python. *Journal of Statistical Software*, 53(9): 1–18.

Nathan, P. 2016. PyTextRank, a Python implementation of TextRank for phrase extraction and summarization of text documents. https://github.com/DerwenAI/pytextrank. Accessed: 2021-12-14.

Ou, M.; Cui, P.; Pei, J.; Zhang, Z.; and Zhang, W. 2016. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 1105–1114.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Rosvall, M.; Axelsson, D.; and Bergstrom, C. T. 2009. The map equation. *The European Physical Journal Special Topics*, 178(1): 13–23.

Roweis, S. T.; and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500): 2323–2326.

Sayyadi, H.; Hurst, M.; and Maykov, A. 2009. Event detection and tracking in social streams. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 3.

Sidorov, S. P.; Faizliev, A. R.; Levshunov, M.; Chekmareva, A.; Gudkov, A.; and Korobov, E. 2018. Graph-Based Clustering Approach for Economic and Financial Event Detection Using News Analytics Data. In *International Conference on Social Informatics*, 271–280. Springer.

Stilo, G.; and Velardi, P. 2016. Efficient temporal mining of micro-blog texts and its application to event discovery. *Data Mining and Knowledge Discovery*, 30(2): 372–402.

Wang, C.; Kim, L.; Bang, G.; Singh, H.; Kociuba, R.; Pomerville, S.; and Liu, X. 2020. Discovery News: A Generic Framework for Financial News Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 13390–13395.

Weng, J.; and Lee, B.-S. 2011. Event detection in twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5.

Zhang, Y.; Chen, Q.; Yang, Z.; Lin, H.; and Lu, Z. 2019. BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific data*, 6(1): 1–9.