Applied Network Science

RESEARCH Open Access

Characterizing financial markets from the event driven perspective

Miha Torkar[1*] and Dunja Mladenic[1,2]

*Correspondence:
miha.torkar@ijs.si
[1] Jozef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia
Full list of author information is available at the end of the article

Abstract

In this work we study how company co-occurrence in news events can be used to discover business links between them. We develop a methodology that is able to process raw textual data, embed it into a numerical form, and extract a meaningful network of connections. Each news event is considered as a node on the graph and we define the similarity between the two events as the cosine similarity between their vectors in the embedded space. Using this procedure, we contribute to the literature by successfully reconstructing business links between companies, which is usually a difficult task since the data on this topic is either outdated, incomplete or not widely available. We then demonstrate possible uses of this network in two forecasting applications. First, we show how the network can be used as an exogenous feature vector, which improves the prediction of the correlation between companies in the network. This correlation is determined from their realized variance as well as using a wide set of machine learning models for prediction. Second, we demonstrate the use of network for predicting future events with point processes. Our methodology can be applied on any series of events, where we have demonstrated and evaluated its applicability on news events and large market moves. For most of the tested algorithms the experimental results show an improvement in performance when including information from our graphs. More specifically, in certain sectors using Neural Networks shows improved performance by up to 50%.

Keywords: Networks, Word embeddings, News, Realized Variance, Finance

Introduction

Finding correlations between publicly traded companies is a topic of interest for a variety of actors in the financial markets. Specifically, many financial institutions use it to predict asset returns, while the regulators want to the know how the risk of default will spread through the market in times of crises. The classical approach towards building such a network is to look at the various types of business links that exist between companies. This includes customer-supplier relationships, subsidiaries, financial loans, and others. However, the discovery of such links is extremely difficult since data on this topic is either outdated, incomplete or not widely available.

On the other hand, there is a constant stream of information available in news articles published by different media outlets. In particular, we regularly read about new business

relationships being formed or old ones being broken apart. This offers an unprecedented opportunity to extract meaningful information from news and form a network of connections between companies on the basis of it. With the help of recent advances in natural language processing techniques (NLP) we set out to fill the gap in literature and build the network of connections in an increasing order of sophistication. Our research specifically aims to enrich the literature by answering the following questions

(1) Can we reconstruct business links between companies based on news events?
(2) What is the relationship between the network of connections implied by news and the one seen in the financial markets?
(3) How accurately can we predict future news events and can they be used to explain market volatility?

Therefore the main contributions of this paper are:

- A novel methodology for reconstructing business links between companies based on news events.
- A novel approach to predicting company volatility based on news events.
- A novel approach to predicting an occurrence of new events for pairs of companies.

The starting point of our work is to simply count the number of times two companies appear together in the same news story, where a story is defined as a collection of news articles reporting on the same event. Building on this, in the next step we compare the news stories about companies based on their content as well. Therefore, if two companies are reported about in similar stories but are not mentioned explicitly, we could still detect a link between them. Once the network of relations is obtained, it is then possible to compare it to the relationship between companies on the financial markets. Stock market data is a perfect source for this kind of information, since there is a vast amount of literature studying the gradual incorporation of new information into the prices (see Hong and Stein 1999 or Hirshleifer and Teoh (2003)). We are particularly interested in investigating whether there are any similarities between the correlation matrices obtained from news and stocks. Having obtained the network of connections between companies and seeing how it changes through time, enables us to predict it's evolution into the next step as well. This can be done by assigning a point process to each node of the network. The main characteristic of the point process is its intensity function as it governs the rate of arrivals of new events. Therefore, we can leverage the learned network from the previous step and investigate whether the prediction of new market events can help us explain market variability.

The rest of the paper is organised as follows. Section 2 reviews the related literature on this topic. Section 3 defines the data that we are using in our empirical analysis. Section 4 defines the research questions which we are addressing in this work and presents the framework for our model, in which we outline how textual data gets transformed to the numeric vectors and then forms the basis for our graph construction. We conclude by presenting results of all of the used methods in Sect. 5, discuss and interpret them in Sect. 6 and present the summary as well as possible future work in Sect. 7.

Literature review

This work is part of the growing literature on studying the connection between news and financial markets. Tetlock (2014) and Tetlock (2015) provide excellent reviews of this subject, showing how media can exert causal influence on financial markets. In order to compare the related literature to our research, Table 1 provides an overview of the input data, which others have used. The majority of the related research is using only one source of information for their news data, whereas in our work, we are using multiple public sources of information and combining their content (see Sect. 3). Furthermore, the types of news stories being analysed can be classified with regards to their sources, which we categorise into general news or more finance specific news. To complete the comparison in Table 1 we also present the number of items (eg., events, news articles, documents) each research paper had analysed.

The majority of the related work is focused on extracting relevant information from news and demonstrating its predictive power. Initial studies in the area used only news counts in their analysis. With this approach (David et al. 1989) could explain less than half of the volatility in the markets. Similarly (Mitchell and Mulherin 1994) did not find a strong relation between the frequency of news announcements and market moves, since patterns in news announcements did not explain day-of-the-week seasonalities in market activity. On the other hand, Brad and Douglas (1993) found that during the two days after the publication of stock recommendations positive abnormal returns of 4% and an average volume double the normal values can be observed. Along the same lines (Schumaker and Chen 2009) obtained a directional accuracy of 57% with return of simulated portfolio of 2.06%. Wu et al. (2009) show an increased accuracy of prediction between 2% to more than 20% for certain stocks, where the positive change was even larger in returns. Moreover, Lumsdaine (2010)

Table 1 Textual Input Data in related studies of the connection between news and financial markets

Reference	Text type	Text source	No. of items
David et al. (1989)	General news	New York Times	49 events
Mitchell and Mulherin (1994)	Financial news	Dow Jones & Company	752,647 headlines
Brad and Douglas (1993)	Financial news columns	Wall Street Journal	48 + 48 stock picks
Schumaker and Chen (2009)	Financial news	Yahoo Finance	9,211
Wu et al. (2009)	General news	The Standard	?
Lumsdaine (2010)	News readership data	Bloomberg	?
Fehrer and Feuerriegel (2015)	Ad hoc announcements	Corporate disclosures	8359 headlines
Yu et al. (2013)	General news and social media	Google Blogs, Board-Reader, Twitter, Google News	52,746 messages
Hagenau et al. (2013)	Corporate announcements and financial news	DGAP, EuroAdhoc	10,870 and 3478 respectively
Shen et al. (2016)	General news	Baidu News	6250 articles
Ding et al. (2015)	Financial news	Reuters and Bloomberg	664,399 Documents
Zhang et al. (2016)	Financial news Column	NetEase	136 + 106 events

show a cumulative P&L of 60.8% on portfolio that shorts banks that were in top readership rankings and is long on others. Fehrer and Feuerriegel (2015) demonstrate a relative accuracy of prediction outperformed benchmark (random forests) by 5.66% with final accuracy of 56%. Hagenau et al. (2013) use a feedback-based feature selection combined with 2-word combinations that achieves accuracy of up to 76%. Similarly (Ding et al. 2015)'s model can achieve nearly 6% improvements on S&P 500 index prediction and individual stock prediction and has 65.08% classification accuracy. Zhang et al. (2016) show a significantly positive abnormal return as well as excessive trading volume on the event date. As an alternative source of information (Yu et al. 2013) show that social media has a stronger relationship with firm stock performance than conventional media, while both social and conventional media have a strong interaction effect on stock performance. Fan et al. (2019) include information about similarity between firm's products, using SEC fillings, to improve prediction for which companies are going to collapse and which are going to be top performers. The literature not only differs in the text type but also whether they use the whole article or only the headlines in their analysis. Peramunetilleke and Wong (2002) and Huang et al. (2010) argue that headlines are actually a better proxy for the content of the article as there is less noise in the data.

The impact of news on financial markets persists only for a limited amount time after which it is incorporated into the stock price. Shen et al. (2016) shows that incorporating information from news into regression slightly improves predictions and decreases volatility persistence, which suggest information is not absorbed immediately. Chordia et al. (2005) suggest that a reasonable market convergence rate to efficiency is more than 5 min but less than 1 h. Further research in this area by Reboredo et al. (2013) finds that the upper limit could be around 15min. This is particularly important when trying to distinguish between the effects of multiple events on one company. Additionally, Shynkevich et al. (2015) tests the underlying assumption that each news story has the same magnitude of impact on the company. They explore an alternative approach in which they define five categories for the relevant events and evaluate their impact accordingly.

The network structure that we aim to extract from the news stories has also been studied extensively on financial time series only. Rubin et al. (2019) look at how correlations of stock returns evolve through time and then uncover communities of stocks, which move in highly correlated matter and can be interpreted as groups from the same industry sector. Isogai (2017) goes a step further and applies network clustering algorithms to Japanese stock returns to detect several stock groups that extend the existing business sector classifications, while also reducing the dimensionality of the network. This allows them to extract more important information from complex correlation networks through time as well. Millington and Niranjan (2020) on the other hand, investigate whether networks based on correlation can infer spurious relationships, so they compare them to networks constructed with partial correlation on S&P 500 returns. Marti et al. (2017) offer an excellent review of the network methodology, which is applied in other academic fields as well, pointing out that the standard approach towards graph modelling is to compute its Minimum Spanning Tree and use that in further analysis. However, they find that when using textual data, the

most widely adopted approach is to use count networks and then transform them into aggregate indexes. This is something we aim to extend by not solely relying on co-occurrence of companies in news events, but also by comparing the content of the text.

Data

Financial data

We consider 75 constituents of S&P 500, of which the full list is displayed in Table 2. For each one of those companies we download the *transaction* prices from the NYSE's Transactions And Quotes (TAQ) database. We only consider data from *January 2nd 2014* through *May 31st 2018*, which translates to 1612 calendar days and 1110 trading days respectively. The raw data is cleaned according to the procedures described in Barndorff-Nielsen et al. (2009) with the sole exception of not implementing their rule T4 for removing the so called "outliers". Further description of cleaning procedures are given in the Appendix A.

News data

The textual news data used in our empirical analysis, which covers the same period as the stock data, is supplied by EventRegistry (Leban et al. 2014). EventRegistry is

Table 2 Ticker symbols and EventRegistry concepts for news retrieval

Ticker	Reference	Ticker	Reference	Ticker	Reference
AAPL	Apple	ABBV	Abbvie	ABT	Abbott
ACN	Accenture	AIG	American International Group	AMGN	Amgen
AMZN	Amazon	AXP	American Express	BA	Boeing
BAC	Bank of America	BIIB	Biogen	BLK	Blackrock
BMY	Bristol-Myers Squibb	C	Citigroup	CAT	Caterpillar
CELG	Celgene	CL	Colgate-Palmolive	COP	ConocoPhillips
COST	Costco Wholesale	CSCO	Cisco	CVS	CVS Health
DHR	Danaher	DIS	Walt Disney	DUK	Duke Energy
EBAY	Ebay	EMR	Emerson Electric	ESRX	Express Scripts
F	Ford	FB	Facebook	FOXA	Century Fox
GE	General Electric	GILD	Gilead Sciences	GM	General Motors
GS	Goldman Sachs	HD	Home Depot	HON	Honeywell
HPQ	HP	IBM	IBM	INTC	Intel
JNJ	Johnson & Johnson	JPM	JPMorgan	KO	Coca-Cola
LLY	Eli Lilly	LMT	Lockheed Martin	MA	Mastercard
MCD	McDonald's	MDLZ	Mondelēz	MDT	Medtronic
MET	MetLife	MO	Altria	MON	Monsanto
MRK	Merck & Co	MS	Morgan Stanley	MSFT	Microsoft
NKE	Nike	ORCL	Oracle	OXY	Occidental Petroleum
PEP	PepsiCo	PFE	Pfizer	PG	Procter & Gamble
PM	Philip Morris	QCOM	Qualcomm	SBUX	Starbucks
SLB	Schlumberger	T	AT&T	TWX	Time Warner
UNH	UnitedHealth	UNP	Union Pacific	UPS	United Parcel Service
USB	U.S. Bancorp	UTX	United Technologies	VZ	Verizon
WFC	Wells Fargo	WMT	Walmart	XOM	ExxonMobil

a repository of events, which are automatically identified by analysing news articles collected from numerous sources around the world. The system monitors RSS feeds of more than 100,000 news outlets globally. Using this procedure EventRegistry captures more than 200,000 articles each day, which are written in various languages, with English, Spanish, and German being the most prominent ones. EventRegistry is indeed well featured by its cross-lingual event-classification system. Furthermore, there are various NLP features created by the system for each article. Articles are clustered into so called events, which form a joint set of all articles reporting about the same news story across all languages. It then extracts relevant information from the texts about the event (entities, people, locations, topics). With this additional information we are able to look for further features of the news event that might be relevant for the market reaction. For more detailed introduction to EventRegistry, see (Leban et al. 2014) and Rupnik et al. (2016). Events for each company are being collected separately in our analysis. The exact details about event retrieval are presented in the Appendix A.

Our study uses a total of 724,310 events in the same time period as financial data set, which is from *January 2nd 2014* to *May 31st 2018*. The number of events we analyse through time is depicted in the Fig. 1 and a more specific breakdown of how many events we use per company and sector is given in Fig. 8.

Proposed methodology

In order to answer our research questions we construct a novel methodology, the workflow pipeline of which is depicted in Fig. 2 which furthermore shows how we extract the network graph from news data. The news data is obtained from EventRegistry (Leban et al. 2014), which crawls a number of news sources worldwide. The proposed methodology comprises of the following steps:

1. Collect news events provided by EventRegistry together with extracted concepts (all entities that have a Wikipedia page) and their relative weights in interval [0, 100]. This is done by EventRegistry which is why we take it as an input into our model.

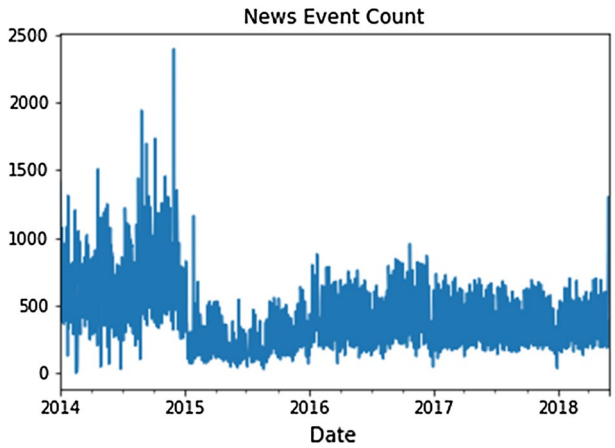


Fig. 1 Number of news event through time

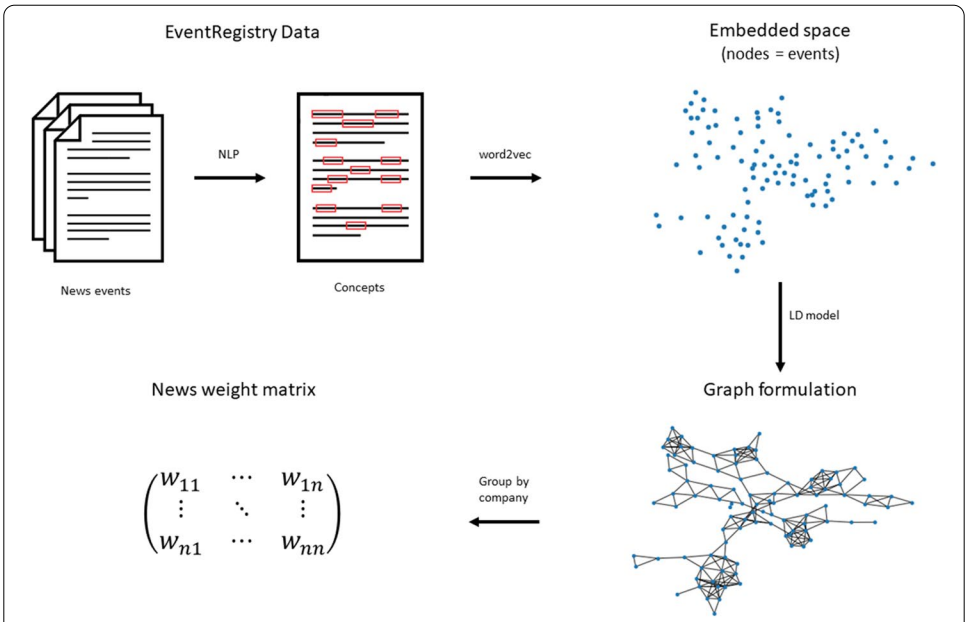


Fig. 2 Pipeline of the proposed methodology for obtaining company relationships from news data

2. Take the concepts and their weights, then apply a pre-trained word2vec model to each concept and multiply it by its weight. The entire document is then represented as a sum of these vectors in an embedded space, which is defined by the dimension chosen in the word2vec model.
3. Consider each event as a node in a graph in the embedded space, which connects events that are close together as measured by cosine similarity distance. This procedure bears a resemblance to the Latent Distance (LD) model which lends its name to the produced graph. A probabilistic time varying threshold is used to define when two nodes in the graph are connected.
4. Obtain the final weight matrix, which defines the strength of connections between companies, by taking all of the events in which the two companies are mentioned and calculate the average cosine similarity between them. In the new graph each node represents a company and weights on edges correspond to strength of the connection between corresponding companies.

Each of these steps is described into further detail in the following sections. The news weight matrix that is produced in this manner is a result in itself since it extracts the relevant business links between companies. In addition, it can also be used in various forecasting applications. We present two possible use cases for financial markets. Firstly, since we are extracting connections between companies we can use the news matrix to predict the correlation matrix between traded companies. This can be achieved by using the news matrix as an input feature vector in any machine learning model, in which each model is using a different approach to utilize the matrix values for prediction. Secondly, it is also possible to use the matrix in prediction of future news or market events. In this approach, one can incorporate the news matrix as a weight matrix in the intensity function of the process that models the arrival of new events. The type of the events

are not predetermined, so one can apply the same methodology to any kind of series of events. We show two possible applications for predicting either news events or larger financial market moves, the workflow of which is depicted in Fig. 3.

Event embedding

The first step in calculating similarity between news events is transforming the provided textual data into numerical form, in order to apply algebraic operations on it as well as use it as input for our models in the next steps. The NLP method that we use for this task is called word2vec, further description of which is given in the following section.

Word2vec word embedding

The word2vec method was originally developed by a team of experts at Google, specialized in natural language processing (Mikolov et al. 2013). It represents every word as a numeric vector, which has been a well founded practice in the literature (see Hinton et al. (1986) or Rumelhart et al. (1986)). With this method we take into account the so-called *multiple degrees of similarity* between words (Mikolov et al. 2013). It not only recognises similarity of words from the same word stem but enables us to go past the syntactic regularities. Once the words are represented as vectors, we are able to perform simple algebraic operations on the word vectors to obtain meaningful results. An intuitive example is given by Mikolov et al. (2013), where the result of *vector("King")* − *vector("Man")* + *vector("Woman")* is close to vector form of the word *Queen*.

The method takes advantage of neural network (NN) to produce the final output, which is shown to surpass other approaches such as N-gram (see Bengio et al. (2003) and Mikolov et al. (2011)). Additional details about the method are given in the Appendix B.

Events to vectors

Each news event from our data provider, EventRegistry, consists of a list of concepts, which are entities that have a Wikipedia page, and their corresponding relevance

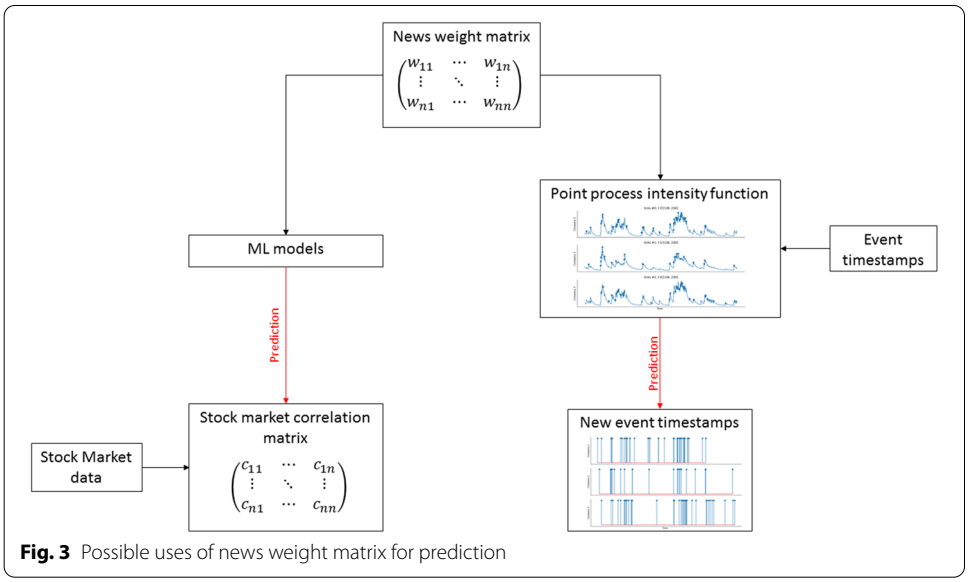


Fig. 3 Possible uses of news weight matrix for prediction

weights. We now describe how these concept weights are calculated and how we use them in combination with word2vec model to obtain the final vector representation for each event. To be more specific, let us assume that we have a set of news events E and each event has a dedicated list of concepts $c_i, i \in \{1, N\}$ associated with it. Then EventRegistry uses the following algorithm to assign a weight w_{c_i} to each concept that appears inside a particular event $e_j \in E$ with M articles $A_k, k \in \{1, M\}$:

$$\tilde{w}_{c_i} = \sum_{j=1}^{M} \bar{c}_i \cdot \mathbb{1}_{c_i \in A_j}, \quad \bar{c}_i \in \{1, 2, \ldots, 5\}, \tag{1}$$

$$w_{c_i} = 100 \cdot \frac{\tilde{w}_{c_i}}{max(\tilde{w}_{c_1}, \ldots, \tilde{w}_{c_N})}, \tag{2}$$

where $\bar{c}_i$ is an article level weight for each concept c_i. The algorithm starts by iterating through all of the articles of a specific event and sums up all of the article level weights $\bar{c}_i$ for each concept c_i. The value of $\bar{c}_i$ in each article is determined by how often that concept appears in the article and where in the article it appears (at the begging or towards the end). The concepts in the event then get normalized to be within the range [0, 100] as represented by the second equation above.

This results in having a relevance score for each concept and a word2vec embedded numerical representation. In order to combine them both and form a numeric representation for all the events we then take the weighted sum of all concepts as our event representation. Using the notation from above we then have:

$$e_j = \sum_{i=1}^{N} w_{c_i} \cdot \text{word2vec}(c_i), \quad \forall e_j \in E. \tag{3}$$

Each event is then represented as a numeric vector in $\mathbb{R}^N$, where N is dependant on the dimension of the word2vec space. In our case we choose $N = 300$, since we are using the pre-trained word2vec model by Google.

Graph formulation

With the techniques in Sect. 4.1 we are able to obtain an embedding space in which all events are placed. In order to determine the similarity between the selected events we form a matrix of similarities W with the ij^{th} element w_{ij} defined as

$$w_{ij} = d(e_i, e_j), \tag{4}$$

where $d(\cdot, \cdot)$ is the distance metric between any two events e_i and e_j under consideration. In our case, we choose the function d to be the cosine similarity, which defines the similarity between any two events. This formulation is very similar to the latent distance model in network analysis, with the only difference being that in our case, we do not have to sample the positions of elements in the latent space from a normal distribution because we obtain them from data.

The similarities matrix W is then used to build a graph of connections between the events, on which nodes correspond to the events in the embedded space.

Company graph formulation—direct comparison

So far, we have assumed that each node in the graph represents an event in the news. In order to determine the correlation between companies, we perform a graph transformation, such that nodes correspond to the companies we are considering. We achieve this by calculating the average cosine similarity between all of the events in which a pair of companies is involved. Since the distance between events corresponds to the elements in the matrix W, we just need to find all of the events that involve a specific pair of companies. In general, we define the strength of connection between any two companies A and B, with corresponding sets of events in which they appear $\hat{e}_A$ and $\hat{e}_B$, as the average of weights of all events that contain them

$$\hat{w}_{AB} = \frac{1}{K} \sum_{i,j=1}^{K} w_{ij} \tag{5}$$

where K is the number of events, calculated as $K = |\hat{e}_A| \cdot |\hat{e}_B|$, while i and j correspond to the indexes of the events in which the two companies were mentioned. Computing this for every pair yields a symmetric matrix $\hat{W}$ that represents the weights in the company graph.

An illustrative example is depicted in the Fig. 4, in which two companies A and B have one event in common and the total number of events in which both companies are mentioned is 5. The value of $\hat{w}_{AB}$ is then calculated as the mean value of all the weights between the 5 events, with 6 possible connections as calculated by multiplying the number of events for company A (3) with the number of events for company B (2).

Alternative graph formulations

Centroid network

The approach described in Sect. 4.2.1 offers a direct comparison of all the events but does not deal with the noise that is present in the data. In order to tackle this issue,

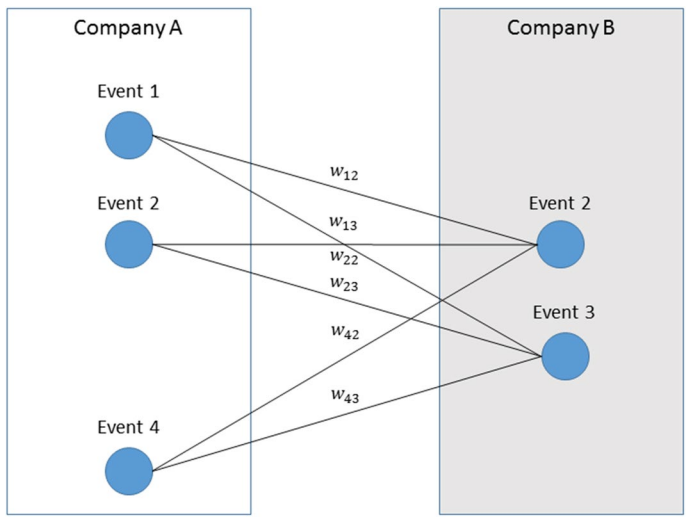


Fig. 4 Example of grouping events to companies

we present two alternative graph formulation approaches that rely on clustering events of each company into smaller subsets and then comparing only the centroids of the clusters.

The first approach is to calculate the centroid of all events for each company in a given time period. The formula for obtaining the centroid of N events $e_1, \ldots, e_N$ is a simple average over all their coordinates

$$C = \frac{e_1 + e_2 + \cdots + e_N}{N}. \tag{6}$$

This is the point that has the minimum sum of squared Euclidean distance between every event and itself. The value of the weight of the connection between two companies A and B, w_{AB}, with centroids C_A and C_B respectively is then calculated as $d(C_A, C_B)$, where $d(\cdot, \cdot)$ is the distance metric, which is the cosine distance in our case.

Events about companies are most often split into various categories, information about which is lost in calculation of a simple centroid. Therefore, we want to obtain multiple smaller sets of events that would represent different parts of the event space, which can be achieved through clustering of data and extracting the centroids of the newly obtained clusters. In order to avoid having to define the number of clusters at each step we choose the Affinity Propagation algorithm (Frey and Dueck 2007) to obtain the clusters for our data. We then calculate the cosine similarity between the clusters (their coordinates in the event space) as well as the average minimum distance between each cluster centre for the two companies. Therefore, for any two companies A and B with N and M clusters respectively, we calculate the strength of the connection between the companies in the weight matrix W by the following equation:

$$w_{A,B} = \frac{1}{N} \sum_{i=1}^{N} \min d(C_{A_i}, (C_{B_1}, \ldots, C_{B_M})). \tag{7}$$

This means that for each cluster center from company A, C_{A_i} we calculate the distance to every cluster center of company B, $C_{B_j} \forall j \in 1, \ldots, M$ and then take the minimum. The distance is defined by the distance metric $d(\cdot, \cdot)$, which is in our case a cosine similarity. The average of all these distances then represents the final value.

In order to assess the added value of our embedding models, we also define a more simple counts network, in which each element of the matrix W is an integer representing the number of times two companies appeared among the concepts of the same news event.

Financial model

In this work we take the realized volatility of the prices (Andersen et al. 2003) as our volatility measure for all of the companies under consideration. In order to define it let S denote the logarithmic price process of a stock and let $t^{(o)}$ ($t^{(c)}$) be the opening (closing) time of stock exchange on trading day-t respectively. Then, given an equidistant grid $\{S_{t^{(o)}+i\Delta} : i = 0, 1, \ldots, n\}$ of prices sampled during $[t^{(o)}, t^{(c)}]$, the realized volatility of S over such grid is defined by

$$RV(S)_t^d = \sum_{i=1}^{n} (S_{t+i\Delta} - S_{t+(i-1)\Delta})^2 = \sum_{i=1}^{n} r(S)_{t,i}^2 \tag{8}$$

where $r(S)_{t,i} = S_{t+i\Delta} - S_{t+(i-1)\Delta}$ is the i-th intra day return. The daily open-to-close return of S is simply denoted by $r(S)_t$ or equivalently written as $r(S)_t = S_{t^{(c)}} - S_{t^{(o)}}$.

We select a 5 min interval to obtain the series of intra day prices that are used in the Eq. 8. This interval is selected because we want to avoid biases by micro-structure noise that appear at higher frequencies (see Hansen and Lunde (2006) for evidence of potential bias). Additionally, choosing a time interval of 5 min for volatility estimation is supported by Liu et al. (2015), where authors provide an extensive empirical analysis. This frequency also allows us to discover what are the changes of volatility close to news events with which we can determine whether movements were correlated with the news events.

Predictive models

Regression models

The company weight matrix that is constructed from the graph of the events can be used as a feature vector for any machine learning model. Intuitively, every row in the weight matrix represents the strength of the connection of the chosen company with all the others. This feature vector can then be concatenated with the historical values of our target variable to form the final set of inputs to the models. Several regression models are used for this task, in order to show the added value of including the news matrix into analysis. The models that are being tested vary from classical machine learning models such as linear regression, Gaussian processes, and random forest, to more advanced neural networks. The target variable will be a covariance matrix of market volatility, as defined in Sect. 4.4, between all 75 companies under consideration.

Gaussian processes (GP) require some extra explanation since they are the only model that uses probabilistic approach. The model uses the observed training data to define a likelihood function and a kernel to define the covariance of prior distribution over the target function. A Gaussian posterior distribution is then defined over a target function and its mean is used for prediction. This follows from the Bayes theorem. The kernel is the most important specification of the model since it determines the shape of prior and posterior of the GP. They represent the assumptions one is making for the function the model is trying to learn since they define the similarity of two data points and assume that similar data points have similar target values. In our case we choose the Radial-basis function (RBF) kernel since it is also used in the support vector machines. The kernel is given by:

$$k(x_i, x_j) = \exp -\frac{d(x_i, x_j)^2}{2l^2} \tag{9}$$

where $d(\cdot, \cdot)$ is the Euclidean distance and l is the length scale parameter. The kernel is infinitely differentiable and is therefore very smooth.

Point process model

In order to be able to predict the future events that will occur and not only determine the connection between the current events, we need to formalise the notion of the appearance times of new events. We do this through the point processes that allow us to model and predict the occurrence of new events on the basis of the old ones. We give a brief introduction to the point processes in the Appendix C and we only describe the connection to the network model defined in previous section here. Finally, in this section we closely follow (Linderman and Adams 2014) as they use the same methodology in their work, although with different applications.

The main component of any point process is its intensity function, which determines the conditional probability that an event will occur in a specific time interval $[t, t + dt)$. One class of such processes are the Hawkes processes (Hawkes 1971), which have the property of self excitation, such that one event can trigger a whole cascade of new events. Therefore, if we denote $\{t_1, t_2, \ldots, t_k\}$ as the observed sequence of past arrival times of events, we can then write the intensity function of a Hawkes process in a discrete form as

$$\lambda^*(t) = \lambda_0 + \sum_{t_i<t} \mu(t - t_i), \tag{10}$$

where λ_0 is the background intensity and $\mu(\cdot)$ is the excitation function. The original paper (Hawkes 1971) used the exponential decay for the value of μ, that is, $\mu(t) = \alpha e^{-\beta t}$ with constants $\alpha, \beta > 0$.

In order to combine the network models with the Hawkes process that we are using, we define the impulse response function μ between two nodes in the network n and n' as follows:

$$\mu_{n,n'}(\Delta t) = a_{n,n'} \cdot w_{n,n'} \cdot \tilde{\mu}(\Delta t; \theta_{n,n'}). \tag{11}$$

Here $a_{n,n'}$ is an entry in the binary adjacency matrix, $\boldsymbol{A} \in \{0, 1\}^{N\times N}$, and $w_{n,n'}$ is the corresponding entry in the non-negative weight matrix, $\boldsymbol{W} \in \mathbb{R}_+^{N\times N}$. This split into two parts allows one to specify two components of the network, namely the *sparsity structure* and *strength* of the interaction inside the network. In order to incorporate the temporal aspect of the network as well, we include the non-negative function $\tilde{\mu}(\Delta t; \theta_{n,n'})$ parametrized by $\theta_{n,n'}$ in the formulation. Moreover $\tilde{\mu}$ is considered to be a probability density function with compact support.

In our analysis we define A from our baseline counts network, such that each element $a_{n,n'}$ corresponds to the 1 if the pair of companies had any joint news events or 0 if they did not. The value of W is defined by Eq. 5, so that the strength of connection is determined by our news similarity measure. We then follow (Linderman and Adams 2014) to choose the probability density $\tilde{\mu}$ of gamma distribution, which is defined as:

$$\tilde{\mu}(\Delta t; \theta_{n,n'}) = \tilde{\mu}(\Delta t; \alpha, \beta) = \frac{\beta^\alpha \Delta t^{\alpha-1} \exp^{-\beta \Delta t}}{\Gamma(\alpha)}, \tag{12}$$

where $\Gamma(\alpha)$ is the gamma function. The primary role of $\tilde{\mu}$ is to define the level of influence events from one company have on others at the time lag Δt. To simplify the

equation we choose the constants α and β to be equal to 2 and 1, but other values have been experimented with.

In this formulation, the background rate λ_0 from Eq. 10 has an intuitive explanation. It explains the events that cannot be classified as a reaction to the preceding events. These incorporate regularly occurring events (quarterly reports) or any other new event that has not been a direct consequence of a different event. In our case, we take this value to be constant and set it to 1, but it is possible to have a time varying background rate, which is left for future work.

Results

Business graph creation

Case study 1: most popular companies

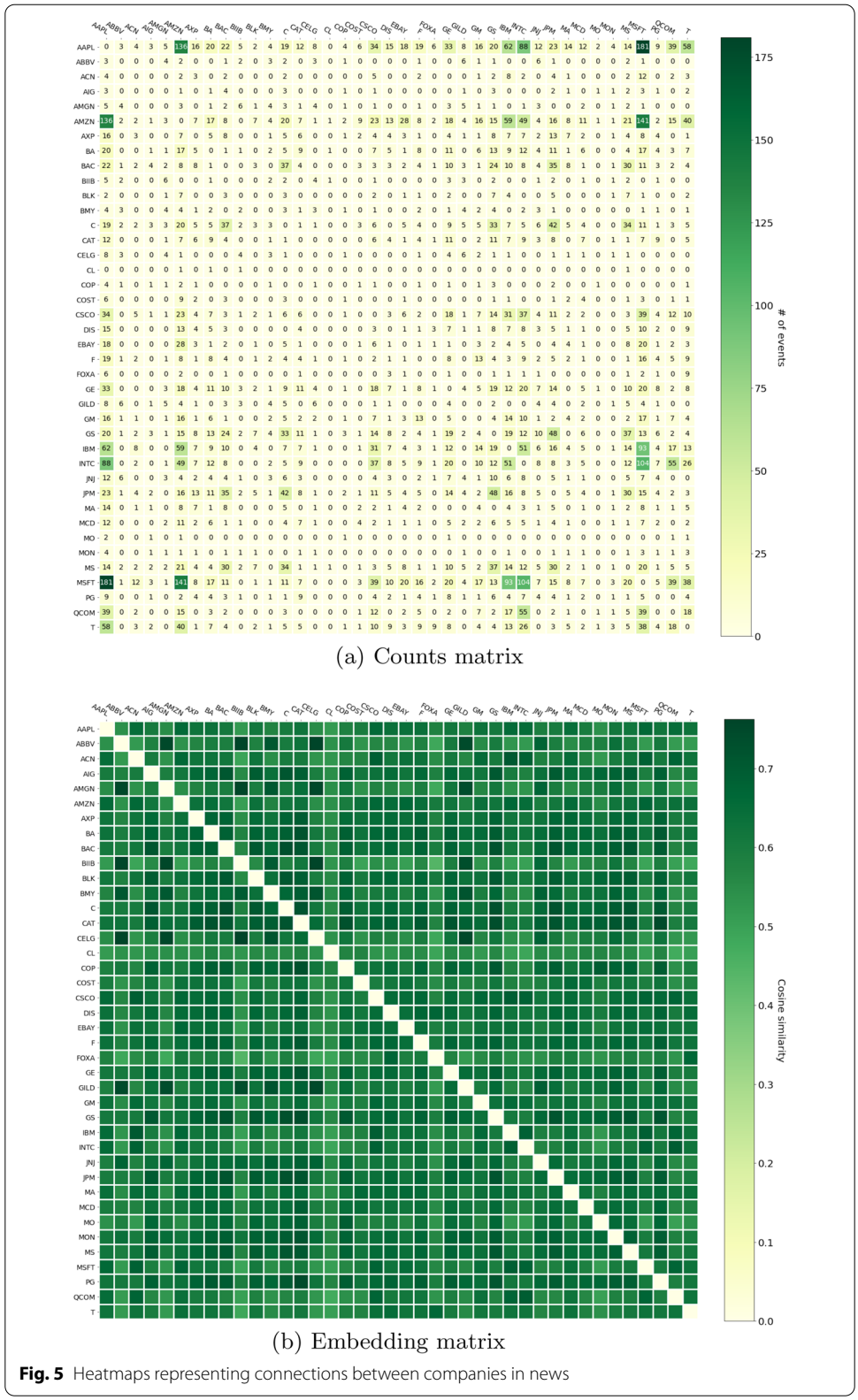
In order to asses our method for graph construction through word embeddings, we test its performance on a set of news events for companies for which there is a known clear connection.

These events were selected from the corpus by the following rules:

1. Select only events in which at least 3 companies from Table 2 are listed among concepts (ie explicitly mentioned).
2. Select only news stories in which at least 2 of the companies are mentioned in the title of the event.

After applying these strict rules, we are left with only the most relevant 657 events. Moreover, only 40 companies from the entire list appear in the events and hence in the network. The corresponding network of connections that is obtained from our model is the Fig. 5. We plot them as heat maps, so that the difference in weights can be easily seen. The first network is obtained by simply counting the number of times two companies appear in the same news story, as defined in the Sect. 4.3, and which we name 'co-occurrence network graph'. The second network represents the embedding matrix W from the direction comparison approach described in Sect. 4.2.1.

From the counts network we can see that the majority of events involve technology companies (Apple, Intel, Microsoft) therefore a substantially larger weight is placed on those connections. However, we chose these events to be similar by design, so the second plot, Fig. 5, is not too surprising since we can see that the majority of companies have a cosine similarity above 0.5. This tells us that our model does indeed manage to capture the relationship between companies from the news. We look into specific examples in the next section. To see another representation of the connections between companies, we plot network graphs from these heat maps in Fig. 6. This representation offers an insight into which companies have the highest number of links with others. We plot different embedding models as discussed in Sect. 4.2, in which we denote our counts network as a co-occurrence network graph, whereas the direct comparison graph from Sect. 4.2.1 as an embedding network graph. To count the number of events in the embedding models, we count how many events with cosine similarity above 0.5 do two chosen companies have. With this representation we see that some companies, such as Apple, retain a strong connection in all of our modelling approaches. If we then translate this relation to the stock



(a) Counts matrix

(b) Embedding matrix

Fig. 5 Heatmaps representing connections between companies in news

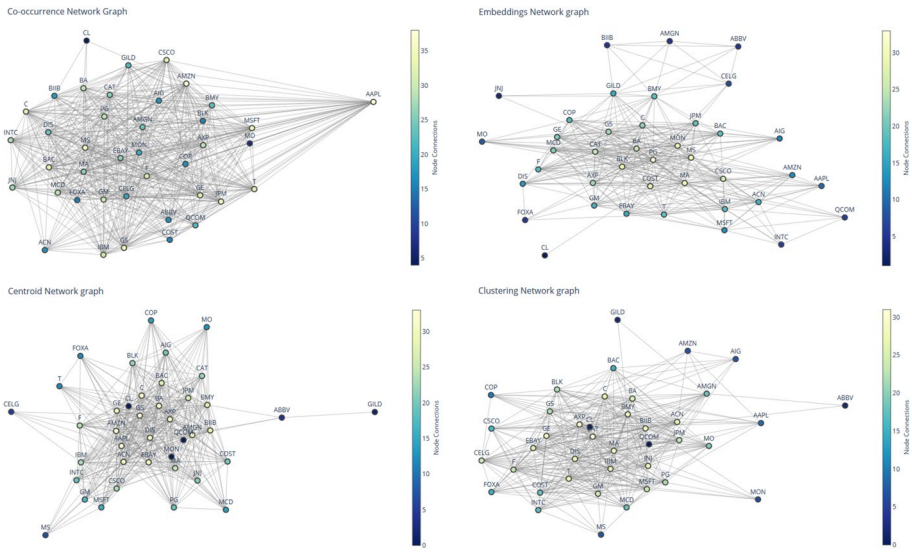
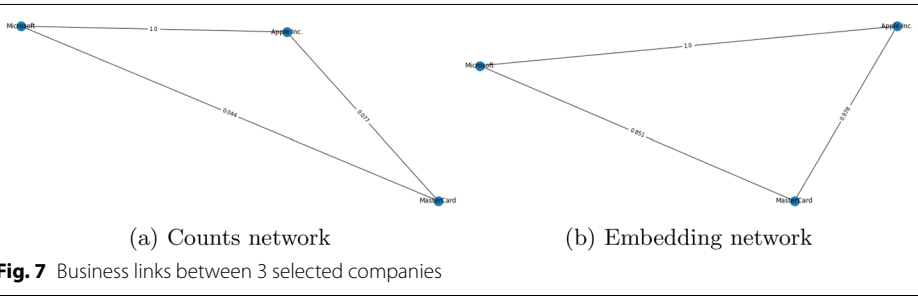


Fig. 6 Relationship graphs between subset of companies



(a) Counts network (b) Embedding network

Fig. 7 Business links between 3 selected companies

market, it would mean that whenever there are any news events affecting Apple our model would predict a market wide impact due to its connection to companies from across the sectors. Given that Apple has the largest market capitalisation such conclusion is to be expected.

In order to look at the networks more closely, we would need to look into how connections change through time as well. It is most often the case that a specific relationship between two firms only exists in certain periods of time. This is particularly evident when two companies decide to merge, since there is a greater media attention around them before the process is complete.

Case study 2: 3 companies

In order to further demonstrate that our approach can capture businesses links between companies, we take the events from Case study 1 and select 3 companies, Apple, Microsoft, and MasterCard, and investigate their relationship.

We demonstrate that our approach is able to recreate this relationship as well as the added value of using comparison on the basis of the content of the news. Specifically, both count and embedding approaches are able to capture the relationship between Microsoft and Apple, but only the embedding network is able

to capture a connection between Apple and MasterCard as well, since it places a large weight on that edge (see Fig. 7). It should be noted that the weights are standardized to the maximum value in the network to allow a fair comparison.

To better understand why the embedding network is placing such a large weight on the connection between Apple and MasterCard we look into the underlying events, where we find that there were several events announcing that Apple is first considering partnering with MasterCard for mobile payments and then later, that they eventually did. Note that in our filtering criteria we did not specify any time period, which means that we are seeing the entire evolution of the partnership here. We take two specific events for further inspection. The first one is titled *'Apple is reportedly partnering Visa, MasterCard and American Express for iPhone mobile wallet'*, with first 5 event concepts and their weights being: [('IPhone', 100), ('Apple Inc.', 86), ('Visa Inc.', 75), ('MasterCard', 69), ('American Express', 63))]. The second event is titled *Apple iPhone 6: Visa, MasterCard and American Express sign up for mobile payments* with concepts: [('Apple Inc.', 100), ('Mobile payment', 83), ('IPhone 6', 75), ('American Express', 62), ('MasterCard', 58)]. The cosine similarity between these two events is calculated to be 0.965, which demonstrates that our similarity measure is capable of recognising similar events from the embedding space. On the other hand the weight between Microsoft and MasterCard is not as large, because there were no similar type of stories. However, we do see see them appearing together in some news events, but they are not as closely related to each other as they are to Apple.

Market volatility prediction from news

In this section we demonstrate how the relation matrix obtained from news between all companies in our data set can be used as an additional feature in regression analysis, in which we choose covariance of companies' market volatility as the target variable. In Table 3 we present the errors for various combinations of the feature set that was included in the modelling. In order to shorten the graph formulation names we abbreviate the counts network as *count*, the direct comparison network as *cos*, the centroid network as *cent*, and the cluster centroid network as *clust*.

In the Table 4 we present relative errors of the same models, where the baseline is taken to be the feature set that does not include news. Specifically, the baseline feature set includes lagged values of the target variables as far as 3 months back. Two different relative error metrics are presented, namely Mean Absolute Error (MAE) and Mean Square Error (MSE). This table demonstrates the relative improvement of prediction for different models when our news matrix is included in the feature set. The reported results are the averages over all companies in our data set, with the largest improvement in error occurring for the Neural Networks.

Sector level results

In order to determine whether news events have different impacts across different industries, we grouped companies under consideration into 10 sectors. The sectorisation is based on the Global Industry Classification Standard (GICS), an industry taxonomy developed by MCSI and Standard and Poor. Since companies inside the same sector

Table 3 Errors of prediction models

Model	W	News		Finance		News + Finance	
		MAE	MSE	MAE	MSE	MAE	MSE
Decision tree	cent	0.775	35.091	0.717	3.375	0.687	3.267
	clst	0.768	35.076	0.717	3.375	0.719	3.322
	cos	0.809	35.187	0.717	3.375	0.666	3.280
	count	0.710	34.853	0.717	3.375	0.727	3.402
Gaussian process	cent	0.587	34.314	**0.570**	**2.699**	0.566	**2.689**
	clst	0.587	34.311	**0.570**	**2.699**	0.571	2.703
	cos	0.586	**34.309**	**0.570**	**2.699**	0.567	2.706
	count	0.589	34.312	**0.570**	**2.699**	**0.565**	2.730
KernelRidge	cent	2.347	50.602	0.611	3.084	1.243	7.191
	clst	1.981	42.257	0.611	3.084	2.478	18.887
	cos	2.743	57.785	0.611	3.084	1.242	11.319
	count	1.227	35.929	0.611	3.084	2.709	32.986
Linear regression	cent	2.644	55.001	0.612	3.096	3.244	40.500
	clst	2.932	62.219	0.612	3.096	4.134	57.534
	cos	2.969	62.224	0.612	3.096	3.391	62.338
	count	2.967	65.251	0.612	3.096	3.291	73.918
Nearest neighbors	cent	0.559	34.293	0.661	2.844	0.618	2.694
	clst	0.579	34.403	0.661	2.844	0.642	2.796
	cos	**0.542**	34.375	0.661	2.844	0.617	2.779
	count	0.562	34.379	0.661	2.844	0.667	2.797
Neural net	cent	0.801	35.289	1.513	21.822	0.700	3.587
	clst	0.922	36.004	1.491	20.703	0.743	3.726
	cos	0.730	34.990	1.493	20.175	0.755	4.007
	count	1.150	36.867	1.482	20.203	0.943	5.766
Random forest	cent	0.699	34.401	0.660	2.880	0.627	2.804
	clst	0.697	34.397	0.660	2.880	0.638	2.695
	cos	0.697	34.416	0.660	2.880	0.635	2.904
	count	0.662	34.325	0.660	2.880	0.662	2.861

This table presents mean absolute error (MAE) and mean squared error (MSE) of various models on variations of two input data sets, historical financial time series (Finance) and network obtained from the news events (News). Last column represents errors when both data sets were used for training of the models and bold numbers show the best performing model on each data set

received different levels of attention in the news, we first present the distribution of news counts associated with each company in a specific sector in Fig. 8.

We can see that in sectors such as Energy, Consumer Staples, or Industrials, there is one company that is represented in the majority of news stories and therefore carries the largest weight in the error analysis. We also have sectors which are represented by only two companies, such as Telecommunications and Materials & Utilities, but they are fairly evenly distributed. To demonstrate that companies inside the same sector are related to each other, we also show their location in the embedded word2vec space in Fig. 9, where we have used t-distributed Stochastic Neighbor Embedding (t-SNE) to visualize the 300 dimensional vectors. Each dot represents a specific company denoted by its ticker, which shows us that there are portions of the space where companies of the

Table 4 Relative errors of prediction models

Model	W	MAE	MSE
Decision tree	cent	0.957	0.968
	clst	1.002	0.984
	cos	0.929	0.972
	count	1.013	1.008
Gaussian process	cent	0.992	0.996
	clst	1.001	1.002
	cos	0.994	1.002
	count	0.990	1.011
KernelRidge	cent	2.034	2.331
	clst	4.053	6.123
	cos	2.032	3.670
	count	4.431	10.695
Linear regression	cent	5.298	13.083
	clst	6.750	18.585
	cos	5.538	20.137
	count	5.374	23.878
Nearest neighbors	cent	0.936	0.947
	clst	0.971	0.983
	cos	0.934	0.977
	count	1.010	0.984
Neural net	cent	**0.463**	**0.164**
	clst	0.499	0.180
	cos	0.505	0.199
	count	0.636	0.285
Random Forest	cent	0.951	0.974
	clst	0.968	0.936
	cos	0.963	1.008
	count	1.004	0.993

This table presents relative mean absolute error (MAE) and mean squared error (MSE) of various models trained on combined financial time series and network obtained from the news. The baseline for relative errors calculation is in each case the same model trained solely on financial data set. Bold number show which model had the largest decrease in error when information from news was included in the data set

same sector are grouped together. There are also areas in which companies from several different industries are grouped together, but that could be expected, since it is often the case that companies are partially involved into business activities across several sectors.

The results are presented in Table 5. To summarize them even further, they are depicted as relative errors in bar plots in Fig. 10.

The results show that the Gaussian Process regression performs best for all sectors apart from Energy, where Random Forest has the lowest MAE. The relative improvement in errors when news are included is most clearly seen with Neural Networks, which is able to improve on previous performance by up to 50%. Moreover, the added value of our methodology over simple news count indicators can also be deduced from the Fig. 10 as bottom two subplots demonstrate an improvement in almost every sector and method. Specifically, Consumer Discretionary, and Materials & Utilities sectors show the biggest relative improvement across all methods.

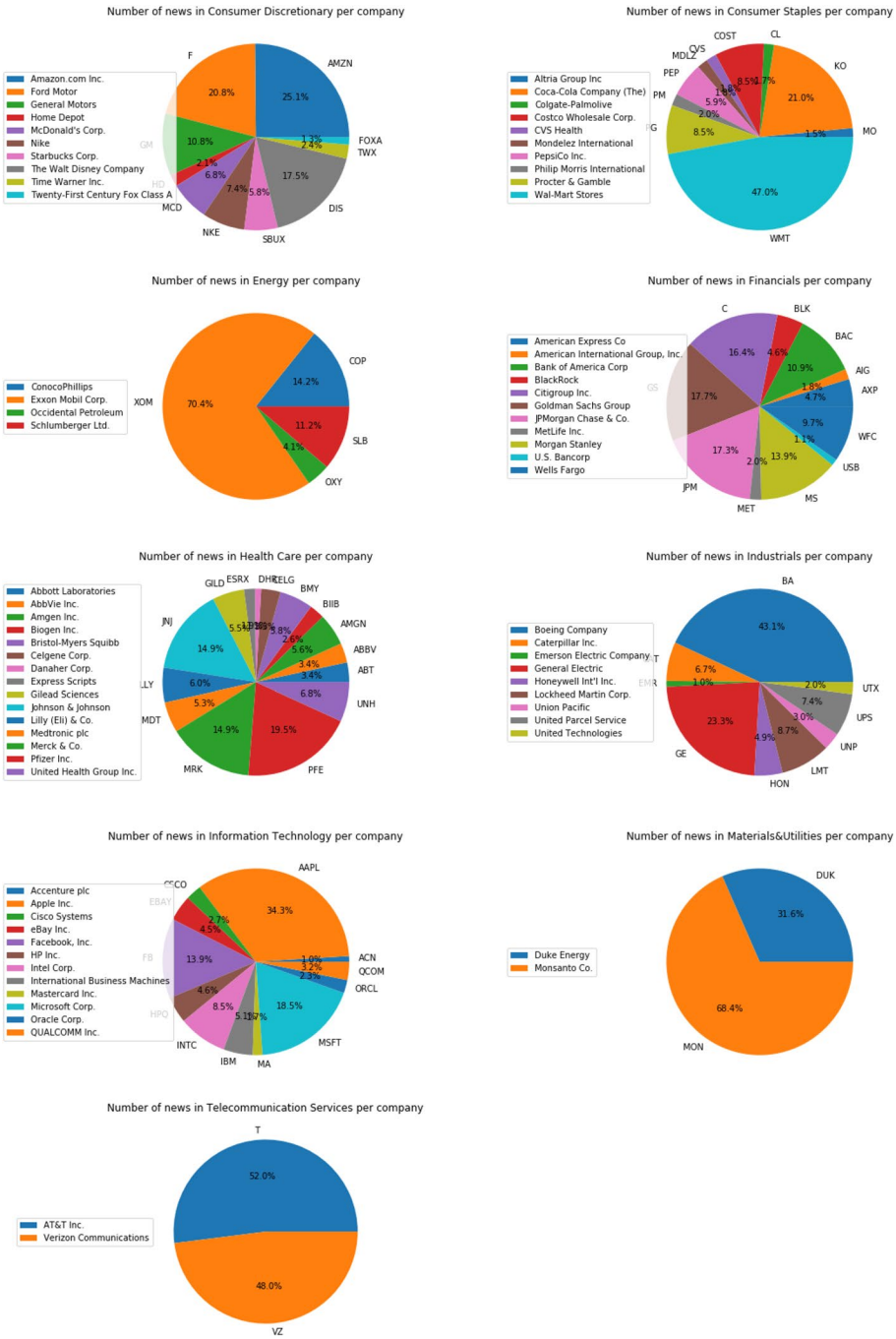


Fig. 8 News distribution inside sectors per companies

News event prediction

In this section we present results for predicting unseen news events from historical information. The models are first estimated over the training period ranging from *January 2nd 2014* until *January 1st 2018* and then tested on 5 months of held out data. We estimate the intensity function of the news arrivals for the point process model first and then make predictions accordingly, incorporating the weight and adjacency matrix that

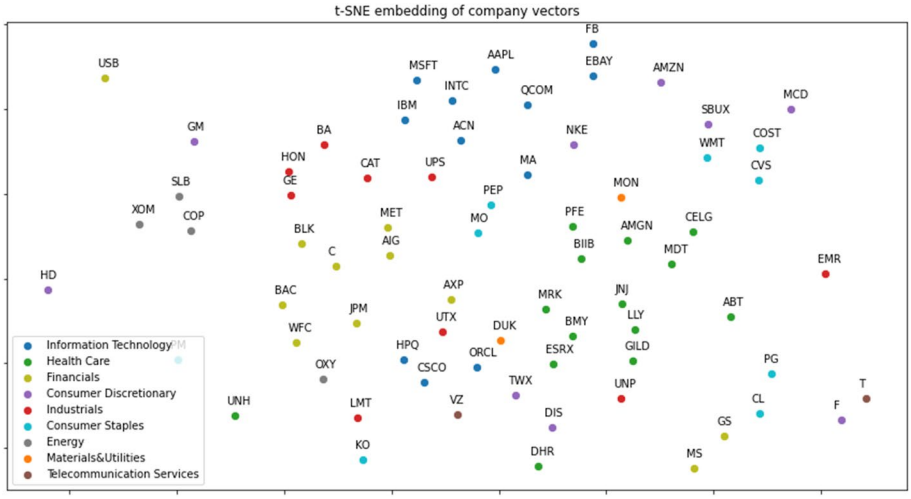


Fig. 9 t-SNE embedding of company vectors in word2vec space

was obtained from the news in the intensity function as defined in Sect. 4.5.2. To obtain a comparative measure of performance for the models we compute the log likelihood of the models for the prediction of the events on the held out data set. We define the exact formulation of the likelihood function for Hawkes processes in Sect. 10.2, in which we are using inference with Gibbs sampling to estimate the mean process of our Hawkes model as defined in Eq. 12. Our target variable is the timestamp of future news events for each company. Table 6 presents the results that we obtain in this sample, relative to the predictive log likelihood of an empty model that corresponds to the Poisson process. We are predicting the news sequence of all the companies simultaneously.

We present results for 3 different models. The first one is the standard Hawkes model, which does not contain any network structure but solely relies on a predefined function. Next, we present a Erdős–Renyi network model, in which the connections between nodes (companies) are determined at random from a binomial distribution. In other words, the values $a_{n,n'}$ from Eq. 11 are chosen to be 0 or 1 with equal probability. The final method is our Network Hawkes model, which includes the direct comparison network structure obtained from the news. We can see that in this case our latent distance Network Hawkes achieves a significant improvement at predicting future news events over the Poisson process, while the standard Hawkes process produces no improvement and Erdős–Renyi model actually gives worse results. This is not a surprising result given that the Erdős–Renyi graph chooses the connection between two nodes at random, but it can be seen as another baseline nonetheless.

News as drivers of the financial market

In similar a setting to the previous section we also test how well can our news embedding network be used to predict timings of future market shifts. We use the same train/test split of data set, in which everything before 1st Janaury 2018 is included in training and the remaining 5 months are used for testing. Specifically, we combine the news embedding matrix with the jumps in realized volatility of the

Table 5 Sector level results

Model	W	Consumer discretionary	Consumer staples	Energy	Financials	Health care	Industrials	Information technology	Materials and utilities	Telecommunication services
		mae	mae	mae	mae	mae	mae	mae	mae	mae
(a) Errors of prediction models per sector with news										
Decision tree	cent	0.617	0.521	0.649	0.767	0.799	0.836	0.606	0.385	0.774
	clst	0.641	0.563	0.697	0.777	0.826	0.867	0.655	0.444	0.798
	cos	0.590	0.502	0.621	0.733	0.780	0.831	0.594	0.337	0.759
	count	0.644	0.565	0.698	0.800	0.850	0.903	0.629	0.408	0.792
Gaussian process	cent	**0.489**	**0.399**	0.566	**0.619**	0.676	0.717	**0.500**	**0.309**	**0.642**
	clst	0.494	0.406	0.568	0.623	0.681	0.722	0.506	0.315	0.647
	cos	0.490	0.401	0.565	**0.619**	0.677	0.718	0.502	0.310	0.643
	count	**0.489**	0.400	**0.550**	0.620	**0.674**	**0.716**	**0.500**	**0.309**	0.647
KernelRidge	cent	1.131	1.122	1.200	1.286	1.345	1.454	1.167	0.897	1.364
	clst	2.336	2.415	2.430	2.509	2.569	2.709	2.382	2.129	2.625
	cos	1.113	1.170	1.189	1.269	1.355	1.445	1.149	0.909	1.342
	count	2.620	2.571	2.635	2.813	2.790	2.887	2.619	2.393	2.858
Linear regression	cent	3.020	3.127	3.185	3.296	3.358	3.618	3.118	2.758	3.504
	clst	3.938	4.196	4.050	4.142	4.188	4.405	4.037	3.746	4.275
	cos	3.165	3.465	2.979	3.394	3.405	3.824	3.340	2.843	3.772
	count	3.624	3.689	3.926	3.853	4.007	4.182	3.707	3.259	3.991
Nearest neighbors	cent	0.539	0.461	0.604	0.669	0.721	0.774	0.556	0.376	0.701
	clst	0.564	0.480	0.630	0.692	0.742	0.796	0.583	0.390	0.737
	cos	0.537	0.455	0.606	0.666	0.720	0.775	0.557	0.356	0.712
	count	0.592	0.502	0.651	0.722	0.768	0.822	0.607	0.422	0.745
Neural net	cent	0.620	0.489	0.780	0.771	0.845	0.855	0.594	0.361	0.799
	clst	0.643	0.543	0.757	0.822	0.881	0.904	0.651	0.484	0.845

Table 5 (continued)

Model	W	Consumer discretionary	Consumer staples	Energy	Financials	Health care	Industrials	Information technology	Materials and utilities	Telecommunication services
		mae	mae	mae	mae	mae	mae	mae	mae	mae
	cos	0.660	0.532	0.840	0.826	0.901	0.918	0.668	0.396	0.831
	count	0.849	0.804	0.888	0.991	1.056	1.102	0.881	0.656	1.060
Random forest	cent	0.545	0.465	0.580	0.703	0.732	0.794	0.555	0.340	0.717
	clst	0.560	0.482	0.609	0.701	0.740	0.795	0.572	0.377	0.720
	cos	0.554	0.478	0.580	0.705	0.736	0.804	0.566	0.345	0.731
	count	0.581	0.508	0.634	0.735	0.772	0.816	0.581	0.380	0.734

Model	Consumer discretionary	Consumer staples	Energy	Financials	Health care	Industrials	Information technology	Materials and utilities	Telecommunication services
	mae	mae	mae	mae	mae	mae	mae	mae	mae
Errors of prediction models per sector with no news									
Decision tree	0.630	0.564	0.680	0.817	0.841	0.881	0.612	0.367	0.772
Gaussian process	**0.494**	**0.406**	**0.568**	**0.622**	**0.680**	**0.721**	**0.505**	**0.315**	**0.646**
Kernel ridge	0.512	0.462	0.591	0.656	0.751	0.768	0.529	0.324	0.678
Linear regression	0.513	0.463	0.592	0.657	0.753	0.769	0.530	0.324	0.679
Nearest neighbors	0.563	0.497	0.577	0.762	0.747	0.858	0.592	0.390	0.730
Neural net	0.725	0.756	0.651	1.001	0.876	0.956	0.781	0.461	0.781
Random forest	0.580	0.505	0.614	0.755	0.767	0.824	0.565	0.359	0.720

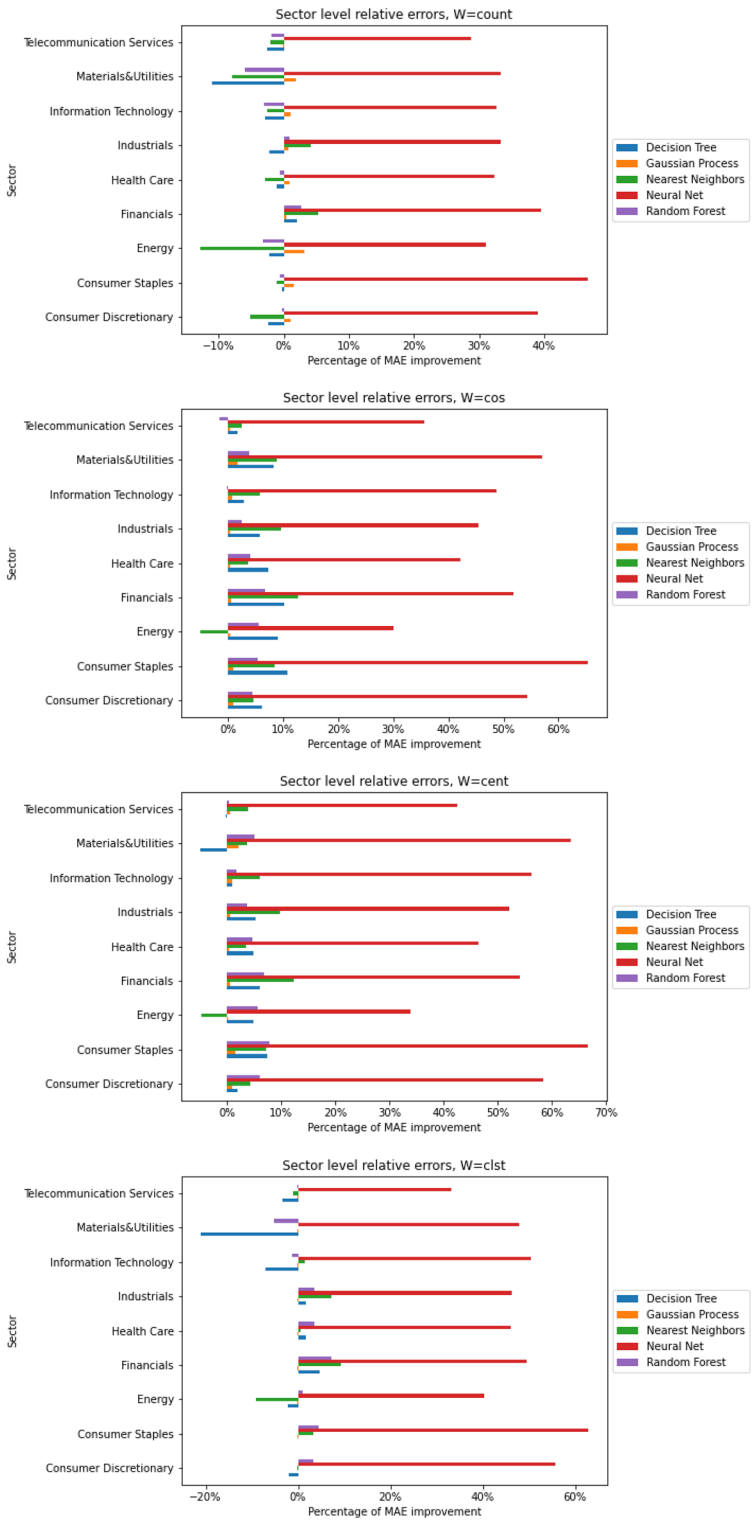


Fig. 10 Relative error improvement graphs per sectors

Table 6 Comparison of models on news prediction task

Model	Pred. log likelihood
Standard Hawkes	0.994
Network Hawkes (Erdős-Renyi)	1.087
Network Hawkes (latent distance)	**0.695**

This table represents errors of three models relative to a homogeneous Poisson process on a task of predicting next news event for a company. Bold value represents which model had the largest improvement in error over baseline

Table 7 Comparison of models on market move prediction task

Model	Pred. log likelihood	
	Market data	Market data with news
Standard Hawkes	1.32	1.25
Network Hawkes (Erdős-Renyi)	**1.28**	1.20
Network Hawkes (latent distance)	1.30	**1.18**

This table represents errors of three models relative to a homogeneous Poisson process on a task of predicting big market moves. The two columns represent comparison of relative error when news data was included into the training and when it was not. Bold values represent which model had the largest improvement in error.

financial data to obtain a better predictions of larger market shifts. We look at the changes of the realized volatility at the 5 min level and mark a significant market movement on the trading day if the realized volatility changed by at least one standard deviation from its running mean of the past 30 days. Let us denote this kind of movement with integer value 1. If the realized volatility changed by more than two running mean values we denote it as a major market shift and assign it a value of 2 and if the realized volatility did not change that significantly we give it a value of 0. Therefore, we are able to obtain a sequence of market events that takes value in $\{0, 1, 2\}$. We then use the models discussed in Sect. 4.5.2 to see if including information about news events improves the forecasts of the event sequence.

In Table 7 we present two sets of results depending on the input data set. In the first one, we only consider a time series of market shifts as defined in the previous paragraph, whereas in the second one, we add the network embedding structure obtained from the news with direct comparison in the model formulation as the weight matrix W.

We can observe that the news events do significantly improve the predictive likelihood of the future market changes, but the log likelihood is still worse than that of the baseline. However, if one takes into account that the model, which we are trying to improve by including news relationships, is performing poorly to start with, we cannot assign the under performance to news data. Since we are still seeing an improvement in relative errors, we can claim that including news stories does improve prediction of the models.

Discussion

Before going through the results let us discuss what our event feature space is embedding and why connections in this space can be useful in gauging market dynamics. The starting point of our feature space is defined by the word2vec model, in which only single words and concepts are embedded. By then adding multiple concepts together and taking their weighted mean, our goal is to obtain a position in the space that would best correspond to the overall content of the news event, a centroid of sorts. By that logic, the events that are closer together have to have similar concepts with high weights and so should be more similar in their content as well. The opposite holds for the events that are distant to each other. The added value of including the embeddings is then coming from considering the content of the news stories in the comparison and not just the fact that the two companies appeared in the same news event together. For example, if two companies are involved in specific type of news story (eg. merger or acquisition, internal scandal, release of new product, etc.), their market dynamics in terms of volatility might be similar. The approach however, does put a slight bias towards companies from the same sector, since they would be embedded closer together by default. That is one of the reasons why we present results divided by sector as well.

In order to see whether connections are being formed where we would expect them to, we perform 2 case studies analysing events that should be similar to each other and demonstrate that our model is capable of capturing these relationships better than the simple co-occurrence network. We find that the companies with larger market capitalisation, such as Apple, have the highest node degree. Some specific examples confirm that these connections between companies form different sectors are not immediately apparent, but that a business link does exist if we look at the data more closely. To assess the added value that one can obtain we then turn to prediction of market volatility and find that different machine learning models have different performance improvements. In general, the more complex the model (Gaussian Process, Random Forrest, Neural Networks), the more performance improvement we see. This highlights different models' capabilities of dealing with more features and that is why it is a good comparison to have. Gaussian Processes are showing the best results with respect to all of the different embedding methods, but the model has not seen substantial improvement in accuracy. This could be due to the probabilistic nature of this model. On the other hand, the biggest improvement was seen in the Neural Networks, which is not surprising, since the model is designed to perform better when more data is provided to it. We should also note that we have not used any of the more advanced layers that could be paired with the current setup of the Neural Network model, which leaves room for improvement in future work.

Knowing that certain sectors and companies receive more attention in the news than others, we split companies by their industry and compare the performance of all the models again. We see that the lowest errors are achieved in the Materials and Utilities sector, where we only have 2 companies. However, that is closely followed by the improvement in error for Consumer Staples, where we have 10 companies, although similarly, one company is dominating the sector; Walmart in this case. To offer a better visualisation of MAE improved for each Sector and each model separately, we plot Fig. 10. Neural Networks are seeing improvement across all sectors and with all the

embedding networks, which confirms the point that this methodology is performing substantially better when it is given more input features. More importantly, what this figure also demonstrates is the added value of our embeddings models over the simple co-occurrence count network. Almost all of the models do not see any improvement in prediction accuracy when Counts network is used, with the exception of Financial sector. However, we are unable to find a wholesome explanation for this from the data. There is no substantial difference between the three embedding models where a slight edge could be given to the Centroid network.

Our embedding network also has an alternative application to a more general problem of event prediction. We show that the counts network can be used in combination with the embedding network to obtain an improvement in accuracy when predicting the timestamp of the next news event. It should be noted that the results we present are compared to the model that has random arrival times and we leave any formulation of trading strategy on the basis of it to future work. We can see that the model is a lot more successful at predicting the arrival times of the news events versus arrival times of market events. Possible explanation for that is that the original market model is not outperforming the baseline in the first place, so introducing the news matrix does improve the results, but not enough to beat the baseline. Future work could consider changing the definition of market movement as well and then determining whether the same conclusions can be drawn.

Conclusion

In this work we show how news events can be used to detect business links between companies. We developed a cross-lingual methodology that is able to extract company relationships from textual data in the news and then transform it into a graph model through embeddings. This network is then used to improve predictions of correlations between companies on the financial markets. We also demonstrate the added value of our approach on specific cases studies, in which our approach detects a connection between companies based on the content of news events as well. Moreover, we find that including our news matrix improves the performance of the majority of standard machine learning models, where the greatest improvement is seen in neural networks. This becomes even more evident when we compare performance of the models in each sector of our target companies. Finally, we also demonstrate how our news relationship matrix can be used to improve prediction of future news events and large market movements.

Appendix A: Data pre-processing

Clearning of TAQ data

For reader's convenience we elaborate how the raw high-frequency data downloaded from the TAQ database are filtered. We follow (Barndorff-Nielsen et al. 2009), except that we do not implement their rule T4 for dropping out "outliers" (transactions with price distant from highest bid or lowest ask).

P1. Remove transactions that occur outside the interval 9:30 am to 4 pm EST.
P2. Remove transactions with zero price.
P3. Keep only transactions that occur in the stock's primary listing (see Table 2).
T1. Remove transactions with correction record, that is, trades with a nonzero correction indicator `CORR`.
T2. Remove transactions with abnormal sale condition, that is, trades where `COND` has a code other than '@', 'E' and 'F'.
T3. If multiple transactions have identical timestamp, use the median price and sum up the volumes.

Sifting news events

The python package `eventregistry` furnishes tools for retrieving articles from EventRegistry's database. The textual data are retrieved and sifted following the procedures below.

Retrieving events:

R1. Set `ConceptUri` to the second column of Table 2.
R2. Set `dateStart` to 2014-1-1 and `dateEnd` to 2018-1-1.
R3. `location` and `category` and set by default.

Appendix B: Word2vec

In this section we give a more detail explanation on how the word2vec embeddings are being calculated. Interestingly, the word2vec is not a single algorithm but split into two sub-algorithms, which determine how NN learns the word representation. These two algorithms, called Continuous Bag-Of-Words (CBOW) and Skip-Gram, are described in the next two sections and depicted in Fig. 11. In the following two subsections we closely follow (Rong 2014), who offer a detailed description of the word2vec method.

Continuous bag-of-words

The first sub-algorithm is called *Continuous Bag-Of-Words* (CBOW), in which the NN learns the vector representation of each word from the window of surrounding context words. In the simplest case this means that for a word at position n, w_n, only its predecessor w_{n-1} and successor w_{n+1} are considered. If a word appears more than once in the data set its context words in all their instances are considered in the training of the NN.

The NN that we are training only has one hidden layer between the input and output layer, which is equivalent to two transformations of the input vector. The first one occurs from the input to hidden layer and the second one from hidden layer to the final output. In order to present this more formally we need to define some hyper parameters as well as certain concepts. Let us assume that we are only considering V unique words from our texts $x_1, \ldots, x_V$, which define our vocabulary size. Any word w in the document

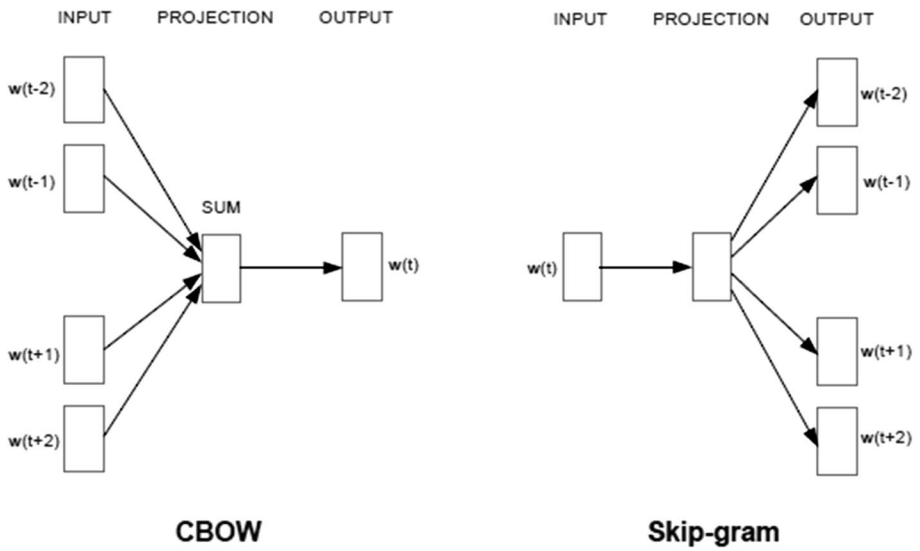


Fig. 11 Graphical representation of word2vec. *Note* The flowchart is borrowed from the original paper of Mikolov et al. (2013). It depicts the mechanism underpinning the two sub-algorithms of word2vec,CBOW and Skip-gram. In CBOW, the network learns w_n, the word at position n, by virtue of its predecessors w_{n-2} and w_{n-1} as well as successors w_{n+1} and w_{n+2}. *Skip-Gram* reverses the process in CBOW, that is, with w_n given, the neural network predicts the surrounding words ($w_{n-2}, w_{n-1}, w_{n+1}, w_{n+2}$)

is then represented as a one-hot encoded vector of dimension V with 1 at position i if $w = x_i$ for $i \in [1, V]$ and 0 everywhere else. Every word that will be used for training of our NN will be represented in this format.

As mentioned before, our NN only has one hidden layer, for which we need to choose a dimension, N. In order to transform our input vector of dimension $1 \times V$ to $1 \times N$, we multiply it with a weight matrix W of dimension $V \times N$. When implementing this procedure in practice, this matrix is first filled with random numbers and then updated during training. So given C context words for our target word, each represented as one-hot encoded vector $\mathbf{x}$, we transform each one separately and then take the average. The hidden layer can then be written as

$$\mathbf{h} = \frac{1}{C}(W^T\mathbf{x}_1 + W^T\mathbf{x}_2 + \cdots + W^T\mathbf{x}_C) \tag{A1}$$

$$= \frac{1}{C}W^T(\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_C) \tag{A2}$$

Since all of the vectors $\mathbf{x}_i$ for $i \in [1, \ldots, C]$ only have one non-zero element, we are essentially taking average of the non zero rows of W and copying them to h. We do not apply any additional functions on top of this calculation, so in terminology of NN we are applying a linear activation function at this stage.

Next, we still need to transform the hidden layer to the output layer. Here, we reverse the previous process and multiply the hidden layer with a matrix $\hat{W}$ of dimension $N \times V$. Using this procedure we can compute a score vector $\mathbf{u}$ for every word in the vocabulary.

$$\mathbf{u} = \hat{W}^T \mathbf{h} \tag{A3}$$

In order to obtain the posterior distribution of our target word given all of it's context words, we use a softmax activation function, which is also known as the log-linear classification model. Note that this is a multinomial distribution. In other words we are trying to calculate the weight for an output word w_O with position k in the vocabulary given its context words as input $w_{I,1}, \ldots, w_{I,C}$.

$$p(w_O|w_{I,1}, \ldots, w_{I,C}) = y_j = \frac{\exp(u_j)}{\sum_{j=1}^{V} \exp u_j}, \tag{A4}$$

where y_j denotes the output of the j-th unit in the output layer. This process is depicted in Fig. 12, where we take our target word to be at position k in the vocabulary and has C context words. Equation A4 is the loss function that the model is trying to maximise. So the training objective is to maximise the conditional probability of observing the actual word w_O given all of the context words as an input. This is achieved through updating the weights in the matrices W and $\hat{W}$. Details of these calculations can be found in Rong (2014).

Once the model is trained for all of the words in the vocabulary, the final output of the model is the optimised matrix W that transforms each word into its embedded space. Namely, each row of the matrix W represents the word2vec representation of every word in the vocabulary.

Skip-gram

The second sub-algorithm, called *Skip-Gram*, reverses the process in CBOW, that is, with w_n given, the neural network predicts the surrounding context words. One has to

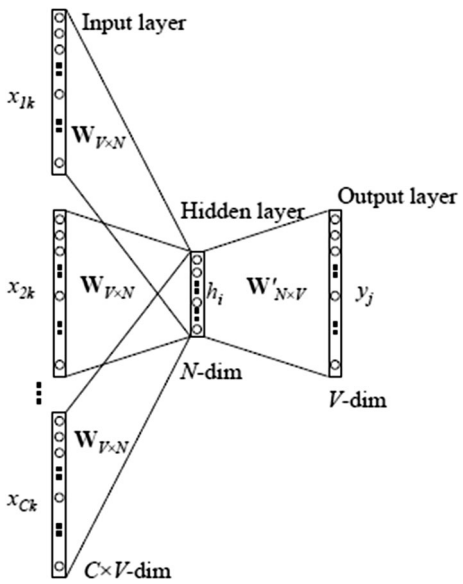


Fig. 12 Graphical representation of CBOW with C context words for target word at position k. *Note* The picture is taken from (Rong 2014)

choose the number of these context words, C, that will be predicted and in the case that we choose $C = 2$ then we are predicting (w_{n-1}, w_{n+1}), so one before and one after our target word w_n. Formally speaking the target word is now at the input layer of the NN and the context words are at the output layer.

Now let us use the same notation as in the Sect. 9.1. However the previous target word w_k with its one-hot encoding vector x_k is now used as the input and the C context words are now our targets. The first step, where we make the transformation to hidden layer, is similar to the equation Eq. A2, but we only have a single input word in this case. This means that the equation for the hidden layer then becomes

$$\mathbf{h} = W^T x = W^T_{k,\cdot}, \tag{A5}$$

where $W_{k,\cdot}$ represents the kth row of the transformation matrix W. This is also the word-2vec representation of the target word w_k. Then from hidden layer to the output layer we again have another transformation represented by matrix $\hat{W}$. However the output layer now consists of C multinomial distributions instead of just one as in CBOW. We use the same $\hat{W}$ in all calculations and the score vector $\mathbf{u}$ for the jth unit on the c^{th} context word is then written as

$$u_{c,j} = u_j = \hat{W}^T_{\cdot,j}\mathbf{h} = \hat{W}^T_{\cdot,j} W^T_{k,\cdot}, \quad \text{for } c = 1, 2, \ldots, C. \tag{A6}$$

The final output equation is then:

$$p(w_{c,j} = w_{O,c}|w_I) = y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{i=1}^{V} \exp u_i}, \tag{A7}$$

where $w_{c,j}$ is the jth word on the c^{th} panel of the output layer and $w_{O,c}$ is the true c^{th} word among the output context words. The $y_{c,j}$ is then the final output of the jth unit on the c^{th} panel. The method is depicted in Fig. 13. Further details are available in Mikolov et al. (2013) and Mikolov et al. (2013), where the method was first introduced. This is an alternative way of how to obtain the word2vec formulation of the entire vocabulary in the form of the matrix W that was seen in the previous section. Finally the authors of both CBOW and Skip gram (Mikolov et al. 2013) noted that the first is faster while the later is slower, but produces better results for infrequent words.

Appendix C: Point process

In order to define the point process we must first define a counting process as the following:

Definition 1 (*Counting Process*) A counting process is a stochastic process $N(t)_{t\geq 0}$ taking values in $\mathbb{N}_0$ with $N(0) = 0$ and is almost surely finite, right-continuous step function with an increments of $+1$. Furthermore it is adapted to the filtration $\mathcal{H}(t)_{t\geq 0}$, which is the history of values up to time t.

Now we are able to define the Point process.

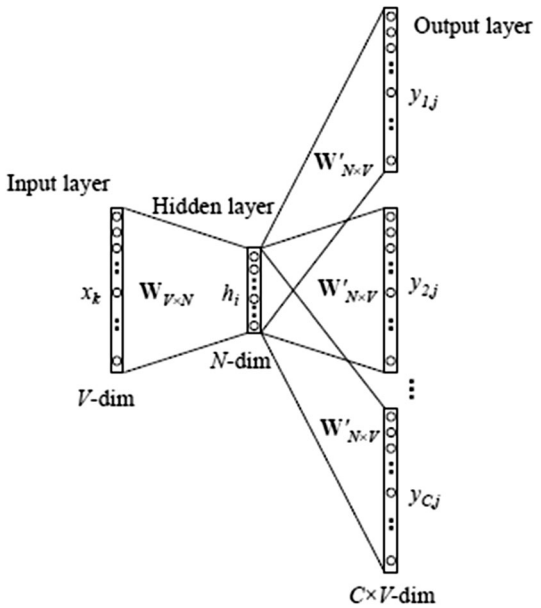


Fig. 13 Graphical representation of Skip Gram method with C context words for target word at position k. *Note* The picture is taken from (Rong 2014)

Definition 2 (*Point process*) Let $\boldsymbol{T} = \{T_1, T_2, \ldots, \}$ be a sequence of random variables taking values in $[0, \infty)$ and $\mathbb{P}(0 \leq T_1 \leq T_2 \leq \ldots) = 1$. Then we say that $\boldsymbol{T}$ is a *(simple) point process*.

In other words a point process is a set of arrival times $\boldsymbol{T} = \{T_1, T_2, \ldots, \}$ at which the counting process has jumped. Another way to characterise a particular point process is through defining the conditional probability function given the history up until the last arrival u, $\mathcal{H}(u)$.

Definition 3 (*Density function*) Let $N(t_k) = \{t_1, t_2, \ldots, t_k\}$ be a point process. Then given that $N(t)$ is $\mathcal{H}(t)$ measurable we define the conditional density function (cdf) of the next arrival time T_{k+1} as

$$F(t|\mathcal{H}(u)) = \int_u^t \mathbb{P}(T_{k+1} \in [s, s+ds]|\mathcal{H}(u))ds = \int_u^t f(s|\mathcal{H}(u))ds \tag{A8}$$

and the joint p.d.f. for $N(t)$ is then

$$f(t_1, t_2, \ldots, t_k) = \prod_{i=1}^{k} f(t_i|\mathcal{H}(t_{i-1})) \tag{A9}$$

However these functions are hard to work with and for this reason we introduce the conditional intensity function, $\lambda(t)$. This function is also referred to as hazard rate across literature (Cox 1955).

Definition 4 (*Conditional intensity function*) Let $N(t)$ be a counting process with associated history $\mathcal{H}$. Then we define the conditional intensity function as a non-negative function $\lambda(t)$ such that

$$\lambda(t) = \lim_{h \downarrow 0} \frac{\mathcal{E}[N(t+h) - N(t)|\mathcal{H}(t)]}{h} = \mathcal{E}[dN(t)|\mathcal{H}(t)] = \frac{f(t|\mathcal{H}(t))}{1 - F(t|\mathcal{H}(t))}. \tag{A10}$$

It is also $\mathcal{H}(t)$ measurable. We are assuming that such a function exists.

Frequently the integral of the conditional density function is needed, for example in parameter estimation and goodness of fit testing. Hence we define the so called compensator.

Definition 5 Let $N(t)$ be a counting process with conditional intensity function $\lambda(t)$. Then the compensator is defined as

$$\Lambda(t) = \int_0^t \lambda(s) ds \tag{A11}$$

Poisson processes

We now turn to the simplest example of the point processes, namely Poisson process. Let τ be a random variable with exponential probability density function.

We then define the arrival times of the events by $S_n = \sum_{i=1}^{n} \tau_i$, which have a gamma density function

$$g_n(s) = \frac{\lambda s^{n-1}}{(n-1)!} \lambda e^{-\lambda s}$$

and this allows us to define the Poisson process.

Definition 6 (*Poisson Process*) The Poisson process $N(t)$ is defined as the point process where the number of events in any subset A, follows a Poisson distribution with mean $\int_A \lambda(t) dt$. For the case of constant λ we can use the definition of arrival times to write,

$$N(t) = \begin{cases} 0 \;\; 0 \leq t \leq S_1 \\ 1 \;\; S_1 \leq t \leq S_2 \\ \vdots \\ n \;\; S_n \leq t \leq S_{n+1} \\ \vdots \end{cases}$$

Poisson process has stationary independent increments, which means that the occurrence of jumps depends only on length of the interval and not on history before the desired interval. So the mean and variance of the increments are

$$\mathcal{E}[N(t) - N(s)] = \sum_{k=1}^{\infty} k \frac{(\lambda)^k (t-s)^k}{k!} \lambda e^{-\lambda(t-s)} = \lambda(t-s) \tag{A12}$$

$$\mathbb{V}(N(t) - N(s)) = \lambda(t - s) \tag{A13}$$

Hawkes processes

The Hawkes process (Hawkes 1971) is a special case of point process, with the special property that it "self-excites". This means that the arrival of one event increases the chances of future arrivals, or in other words the rate of future arrivals increases with each new event. In our definition we follow (Laub et al. 2015), which provides an excellent review of Hawkes processes with sufficient mathematical rigour. We follow their notation and write $\lambda^*(t)$ as a shorthand notation of $\lambda(t|\mathcal{H}(t))$.

Definition 7 (*Hawkes process*) Let $N(t)_{t\geq 0}$ be a point process that is $\mathcal{H}(t)_{t\geq 0}$ measurable (associated history is contained in $\mathcal{H}(t)$). Assume that $N(t)$ satisfies

$$\mathbb{P}(N(t+h) - N(t) = m|\mathcal{H}(t)) = \begin{cases} \lambda^*(t)h + o(h) & m = 1 \\ o(h) & m > 1 \\ 1 - \lambda^*(t)h + o(h) & m = 0 \end{cases}. \tag{A14}$$

The we call $N(t)$ a Hawkes process if it its conditional intensity function is of the form

$$\lambda^*(t) = \lambda_0 + \int_0^t \mu(t-u)dN(u) \tag{A15}$$

for some $\lambda_0 > 0$ and $\mu : (0, \infty) \to [0, \infty)$ which are called the *background intensity* and *excitation function* respectively. In the case that $\mu(\cdot) = 0$ we obtain a homogeneous Poisson process.

The definition of the conditional intensity given in the previous definition is merely a generalised version of the one that is more common across literature. Hence if we denote $\{t_1, t_2, \ldots, t_k\}$ as the observed sequence of past arrival times, then Eq. (A15) becomes

$$\lambda^*(t) = \lambda_0 + \sum_{t_i<t} \mu(t - t_i) \tag{A16}$$

and so we only need to define the background intensity λ_0 and the excitation function $\mu(\cdot)$. In the original paper (Hawkes 1971) used the exponential decay for the value of μ, that is, $\mu(t) = \alpha e^{-\beta t}$ with constants $\alpha, \beta > 0$.

Next we define the likelihood function of Hawkes processes following (Daley and Vere-Jones 2003), Proposition 7.2.III.

Theorem 1 (Hawkes process likelihood) *Let $N(\cdot)$ be a regular point process on $[0, T]$ for some finite positive T, and let $(t_1, t_2, \ldots, t_k)$ denote a realisation of $N(\cdot)$ over $[0, T]$. Then, the likelihood L of $N(\cdot)$ is expressible in the form*

$$L = \left[\prod_{i=1}^{k} \lambda^*(t_i)\right] \exp(-\int_0^T \lambda^*(u)du) \tag{A17}$$

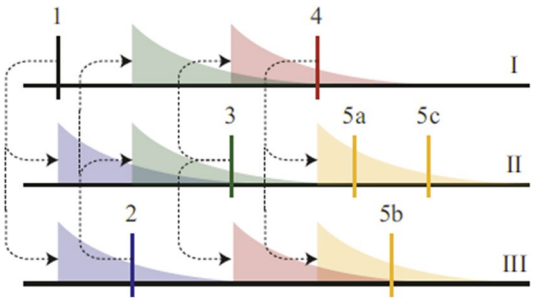


Fig. 14 An example of Hawkes process. Events from one process induce events on the other connected processes and hence cause a cascade of events. Further explanation in text. Picture taken from (Linderman and Adams 2014)

The following theorem will be used in the calculations in order to obtain the likelihood of the Hawkes process.

Theorem 2 (Poisson superposition principle) *Let* $N(\cdot)^i$ *denote the* i^{th} *Poisson process on* $[0, T]$ *with the intensity function* $\lambda_i^*(t)$ *and let* $t_i = (t_1^i, t_2^i, \ldots, t_k^i)$ *denote its realisation over* $[0, T]$. *Suppose that there are K such processes and that* $w_i \in \{1, \ldots, K\}$ *denotes the process we are considering. Denote also* $\lambda_{tot}^*(u) = \sum_{i=1}^K \lambda_i^*(t)$. *Then the likelihood of full set of arrival times is,*

$$L = \prod_{i=1}^{K} \left[\exp(-\int_0^T \lambda_i^*(u)du) \prod_{j=1}^{k} \lambda_i^*(t_j)^{\mathbb{1}[w_j=i]} \right] \tag{A18}$$

$$= \exp(-\int_0^T \lambda_{tot}^*(u)du) \prod_{i=1}^{K} \prod_{j=1}^{k} \lambda_i^*(t_j)^{\mathbb{1}[w_j=i]} \tag{A19}$$

In other words, the union of countably many Poisson processes is another Poisson process.

Figure 14 represents an example of Hawkes processes and how events are correlated among each other when we have 3 processes. The first spike of process I is caused by background rate and it causes impulse responses from the other two processes. Spike 2 originates as an impulse of the third process and causes an additional two spikes in the first two processes. One spike can even causes multiple spikes in the other two processes, which is demonstrated by spike 4, which causes spikes 5a-c. Here we can see that processes excite on another, but not themselves.

Abbreviations

CBOW: Continuous bag-of-words; GICS: Global industry classification standard; GP: Gaussian process; LD: Latent distance; MAE: Mean absolute error; MSE: Mean square error; NLP: Natural language processing; NN: Neural network; RBF: Radial basis function; TAQ: Transactions and quotes; t-SNE: t-distributed stochastic neighbor embedding.

Acknowledgements

Not applicable.

Authors' contributions
MT designed the research, wrote the code that produced results and wrote the initial version of this manuscript. DM guided the research with valuable comments and helped with editing the manuscript. All authors have read and approved the manuscript.

Funding
The research described in this paper was supported by the Slovenian research agency under the project J2-1736 Causalify and co-financed by the Republic of Slovenia and the European Union Research and Innovation programme Horizon 2020 under grant agreement No. 675044 (BigDataFinance) and No. 856632 (Infinitech).

Availibility of data and materials
The TAQ data used to support the findings of this study were supplied by the New York StockExchange under license and so cannot be made freely available. Requests for access to these data should be made to datasales@nyse.com. The news data used to support the findings of this study were supplied by EventRegistry under license and can only be freely accessed up to a limit. Requests for full access should be made to through the website "https://eventregistry.org/".

Declarations

Competing interests
The authors declare that they have no competing interests regarding the publication of this paper.

Author details
[1]Jozef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia. [2]Department of Artificial Intelligence, Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia.

Received: 10 June 2021 Accepted: 16 September 2021
Published online: 09 October 2021

References

Andersen TG, Bollerslev T, Diebold FX, Labys P (2003) Modeling and forecasting realized volatility. Econometrica 71(2):579–625
Barndorff-Nielsen OE, Hansen PR, Lunde A, Shephard N (2009) Realized kernels in practice: trades and quotes. Econom J 12(3):C1–C32
Bengio Y, Ducharme R, Vincent P, Jauvin C (2003) A neural probabilistic language model. J Mach Learn Res 3(Feb):1137–1155
Brad MB, Douglas L (1993) The "dartboard" column: Second-hand information and price pressure. J Financ Quant Anal 28(2):273–284
Chordia T, Roll R, Subrahmanyam A (2005) Evidence on the speed of convergence to market efficiency. J Financ Econ 76(2):271–292
Cox DR (1955) Some statistical methods connected with series of events. J R Stat Soc Ser B (Methodol) 17(2):129–164
Daley DJ, Vere-Jones D (2003) An introduction to the theory of point processes: volume I: elementary theory and methods. Springer, New York
David M, Cutler JMP, Summers LH (1989) What moves stock prices? J Portf Manag 15(3):4–12
Ding X, Zhang Y, Liu T, Duan J (2015) Deep learning for event-driven stock prediction. In: Proceedings of the 24th international conference on artificial intelligence. IJCAI'15. AAAI Press, pp 2327–2333
Fan J, Cohen K, Shekhtman LM, Liu S, Meng J, Louzoun Y, Havlin S (2019) Topology of products similarity network for market forecasting. Appl Netw Sci 4(1):1–15
Fehrer R, Feuerriegel S (2015) Improving decision analytics with deep learning: the case of financial disclosures. arXiv e-prints arXiv:1508.01993
Frey BJ, Dueck D (2007) Clustering by passing messages between data points. Science 315(5814):972–976
Hagenau M, Liebmann M, Neumann D (2013) Automated news reading: stock price prediction based on financial news using context-capturing features. Decis Support Syst 55(3):685–697
Hansen PR, Lunde A (2006) Realized variance and market microstructure noise. J Bus Econ Stat 24(2):127–161
Hawkes AG (1971) Spectra of some self-exciting and mutually exciting point processes. Biometrika 58(1):83–90
Hinton GE, McClelland JL, Rumelhart DE et al (1986) Distributed representations. Parallel distributed processing: explorations in the microstructure of cognition 1(3):77–109
Hirshleifer D, Teoh SH (2003) Limited attention, information disclosure, and financial reporting. J Account Econ 36(1–3):337–386
Hong H, Stein JC (1999) A unified theory of underreaction, momentum trading, and overreaction in asset markets. J Financ 54(6):2143–2184
Huang C-J, Liao J-J, Yang D-X, Chang T-Y, Luo Y-C (2010) Realization of a news dissemination agent based on weighted association rules and text mining techniques. Expert Syst Appl 37(9):6409–6413
Isogai T (2017) Dynamic correlation network analysis of financial asset returns with network clustering. Appl Netw Sci 2(1):8
Laub PJ, Taimre T, Pollett PK (2015) Hawkes processes. arXiv preprint arXiv:1507.02822
Leban G, Fortuna B, Brank J, Grobelnik M (2014) Event registry: learning about world events from news. In: Proceedings of the 23rd international conference on world wide web. ACM, pp 107–110

Leban G, Fortuna B, Brank J, Grobelnik M (2014) Event registry: learning about world events from news. In: Proceedings of the 23rd international conference on world wide web. WWW '14 companion. ACM, New York, NY, USA, pp 107–110
Linderman S, Adams R (2014) Discovering latent network structure in point process data. In: International conference on machine learning, pp 1413–1421
Liu LY, Patton AJ, Sheppard K (2015) Does anything beat 5-minute RV? A comparison of realized measures across multiple asset classes. J Econom 187(1):293–311
Lumsdaine RL (2010) What the market watched: Bloomberg news stories and bank returns as the financial crisis unfolded. https://ssrn.com/abstract=1482019
Marti G, Nielsen F, Bińkowski M, Donnat P (2017) A review of two decades of correlations, hierarchies, networks and clustering in financial markets. arXiv preprint arXiv:1703.00485
Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv e-prints arXiv:1301.3781
Mikolov T, Deoras A, Kombrink S, Burget L, Černockỳ J (2011) Empirical evaluation and combination of advanced language modeling techniques. In: Twelfth annual conference of the international speech communication association
Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
Mikolov T, Yih W-t, Zweig G (2013) Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 746–751
Millington T, Niranjan M (2020) Partial correlation financial networks. Appl Netw Sci 5(1):1–19
Mitchell ML, Mulherin JH (1994) The impact of public information on the stock market. J Financ 49(3):923–950
Peramunetilleke D, Wong RK (2002) Currency exchange rate forecasting from news headlines. Austral Comput Sci Commun 24(2):131–139
Reboredo JC, Rivera-Castro MA, Miranda JGV, García-Rubio R (2013) How fast do stock prices adjust to market efficiency? Evidence from a detrended fluctuation analysis. Phys A Stat Mech Appl 392(7):1631–1637
Rong X (2014) word2vec parameter learning explained. arXiv preprint arXiv:1411.2738
Rubin DN, Bassett DS, Ready R (2019) Uncovering dynamic stock return correlations with multilayer network analysis. Appl Netw Sci 4(1):1–13
Rumelhart DE, Hinton EG, Williams JR (1986) Learning representations by back propagating errors, vol 323, pp 533–536
Rupnik J, Muhic A, Leban G, Skraba P, Fortuna B, Grobelnik M (2016) News across languages-cross-lingual document similarity and event tracking. J Artif Intell Res 55:283–316
Schumaker RP, Chen H (2009) Textual analysis of stock market prediction using breaking financial news: the Azfin text system. ACM Trans Inf Syst 27(2):12–11219
Shen D, Zhang W, Xiong X, Li X, Zhang Y (2016) Trading and non-trading period internet information flow and intraday return volatility. Physica A 451:519–524
Shynkevich Y, McGinnity TM, Coleman S, Belatreche A (2015) Predicting stock price movements based on different categories of news articles. In: 2015 IEEE symposium series on computational intelligence, pp 703–710
Tetlock PC (2015) The role of media in finance. In: Handbook of media economics. Elsevier, vol 1, pp 701–721
Tetlock PC (2014) Information transmission in finance. Annu Rev Financ Econ 6(1):365–384
Wu D, Fung GPC, Yu JX, Pan Q (2009) Stock prediction: an event-driven approach based on bursty keywords. Front Comput Sci China 3(2):145–157
Yu Y, Duan W, Cao Q (2013) The impact of social and conventional media on firm equity value: a sentiment analysis approach. Decis Support Syst 55(4):919–926
Zhang Y, Song W, Shen D, Zhang W (2016) Market reaction to internet news: information diffusion and price pressure. Econ Model 56:43–49

Publisher's Note