Information Processing and Management 59 (2022) 102892

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Why does the president tweet this? Discovering reasons and contexts for politicians' tweets from news articles

Ziyue Li [a], Hang Hu [a], He Wang [a], Luwei Cai [a], Haipeng Zhang [a,*], Kunpeng Zhang [b]

[a] ShanghaiTech University, 393 Huaxia Road, Shanghai, 201210, China
[b] University of Maryland, College Park, MD, 20742, USA

ARTICLE INFO

Keywords:
Social media analysis
Politicians' tweets
Causality inference
Open information extraction
Term weighting
Tweet interpretation

ABSTRACT

Politicians' tweets can have important political and economic implications. However, limited context makes it hard for readers to instantly and precisely understand them, especially from a causal perspective. The triggers for these tweets may have been reported in news prior to the tweets, but simply finding similar news articles would not serve the purpose, given the following reasons. First, readers may only be interested in finding the reasons and contexts (we call causal backgrounds) for a certain part of a tweet. Intuitively, such content would be politically relevant and accord with public's recent attention, which is not usually reflected within the context. Besides, the content should be human-readable, while the noisy and informal nature of tweets hinders regular Open Information Extraction systems. Second, similarity does not capture causality and the causality between tweet contents and news contents is beyond the scopes of causality extraction tools. Meanwhile, it will be non-trivial to construct a high-quality tweet-to-intent dataset.

We propose the first end-to-end framework for discovering causal backgrounds of politicians' tweets by: 1. Designing an Open IE system considering rule-free representations for tweets; 2. Introducing sources like Wikipedia linkage and edit history to identify focal contents; 3. Finding implicit causalities between different contexts using explicit causalities learned elsewhere. We curate a comprehensive dataset of interpretations from political journalists for 533 tweets from 5 US politicians. On average, we obtain the correct answers within top-2 recommendations. We make our dataset and framework code publicly available.

1. Introduction

More and more politicians use social media to directly interact with the public and avoid being quoted out of context by other media (An, Cha, Gummadi, Crowcroft, & Quercia, 2012; Papakyriakopoulos, Shahrezaye, Serrano, & Hegelich, 2019). 6,437 politicians from 26 countries post 6,281,684 tweets in 2018 and the total keeps growing from 2017 to 2020 (van Vliet, Törnberg, & Uitermark, 2020). These tweets, by nature, often have economic and political impacts. For instance, after Trump complains about the cost of jet fighters in a tweet, the manufacturer's stock price drops 5% within 13 min and the tweet is later interpreted to suggest a cancellation of purchase. As a more recent example, Trump's tweets on the 2020 US presidential election are widely believed to be associated with the US Capitol riots (Chen, Deb, & Ferrara, 2021). Though being important, politicians' tweets are often so intricate that they cannot be fully understood by the general public without comprehending the reasons and contexts behind

* Corresponding author.
E-mail addresses: lizy@shanghaitech.edu.cn (Z. Li), huhang@shanghaitech.edu.cn (H. Hu), wanghe@shanghaitech.edu.cn (H. Wang), cailw@shanghaitech.edu.cn (L. Cai), zhanghp@shanghaitech.edu.cn (H. Zhang), kpzhang@umd.edu (K. Zhang).

https://doi.org/10.1016/j.ipm.2022.102892
Received 30 June 2021; Received in revised form 27 December 2021; Accepted 7 February 2022
Available online 21 February 2022

them. When Biden tweets merely four words: '*Listen to the scientists.*', readers may wonder what the trigger is. Later confirmed by Biden, it is a sarcastic response to Trump who is diagnosed with COVID-19 on the very day – as being previously reported, Trump has blamed that Biden's advice on totally complying with scientists' suggestions would sacrifice America's economy. As another example, Trump tweets '*Jay Powell and the Federal Reserve have allowed the Dollar to get so strong*' without explaining its reasons and contexts. Political analysts later conclude that a slowing US economy is Trump's concern and the worst manufacturing survey in a decade reported in a news article before he tweets is the direct trigger. A natural question arises - can machines help people understand politicians' tweets efficiently and effectively? Though the follow-up analysis by political journalists sometimes infer the tweets' reasons and contexts, they usually appear hours later and the lags prevent readers from immediately reacting to the possible coming impacts. From another angle, as the above examples suggest, the reasons and contexts of the tweets may have been reported in prior news. However, simply connecting the tweets with similar news articles without identifying focal contents of the tweets and news articles or considering the causality would be problematic. People are usually only interested in certain parts of the tweets (Kanhabua & Nejdl, 2013). These parts need to be identified in human-readable forms to ease the readers' burden in digesting the tweets. However, classic methods of Open Information Extraction (Open IE) are designed for formal text and they become unsuitable when processing tweets with commonly seen informal rule-free textual expressions. Besides, to identify the focal contents, their temporal and political aspects should be considered, since the public's attention changes over time and they would naturally seek political relevant contents in politicians' tweets (Raza, Habib, Ashraf, & Javed, 2019). Nevertheless, these aspects are overlooked in term-frequency based approaches. Where the causality is concerned, text matching methods based on textual or semantic similarity would not be sufficient. The causality extraction tools may not be applicable in our scenario, since they deal with causality within same contexts, whereas we seek the causality between contents in tweets and these in news articles.

Since only the author of a tweet knows its exact reason and context, the ground-truth data is almost impossible to collect and it poses another challenge to this thread of research. Fortunately, politicians' tweets, especially these from 'star' politicians, are sometimes interpreted by political journalists in follow-up analysis. For instance, the abovementioned tweets from Trump and Biden are both interpreted in later news articles. In this article, we call any inferred reasons and contexts that trigger the politicians to tweet by ***'causal backgrounds'***, to differ from the genuine ground truth.

1.1. Research objectives

We aim at finding in previous news articles causal backgrounds for politicians' tweets that accord well with political journalists' follow-up analysis.

1.2. Prior studies and their limitations

Researchers train machine learning models to classify tweets by their general motivations, such as to criticize or to advise (Kozlowski et al., 2020), and 'to favour' or 'to oppose' (Mohammad, Zhu, Kiritchenko, & Martin, 2015), and relevant news articles have been linked with tweets using textual features to incorporate external knowledge (Tsagkias, de Rijke, & Weerkamp, 2011). However, to genuinely understand tweets, people have to be informed of the specific causal backgrounds and these studies have not touched the 'causality'.

Establishing the specific causal relationships between the content of tweets and that of news articles, as introduced earlier, breaks down into two tasks: extracting human-readable focal contents and identifying cross-context causal relationships. The former task is challenging due to tweets' noisy and informal nature (Derczynski et al., 2015; Ritter, Clark, Etzioni, et al., 2011). Regular Open IE systems that work on formal text often overlook rule-free representations of information which are prevailing in informal expressions and this leaves an open space for enhancements. Identifying the focal contents in tweets appears to be similar to finding check-worthy parts in tweet fact-checking tasks (Hansen, Hansen, Alstrup, Grue Simonsen, & Lioma, 2019; Zhao, Resnick, & Mei, 2015). However, these tasks focus on finding factual claims as a first step (Konstantinovskiy, Price, Babakar, & Zubiaga, 2021), regardless of whether the content is of interest to readers. Furthermore, as matured tasks, they can easily train on large annotated datasets from experts whereas we have to invent weighting schemes based on external open knowledge to reduce the intensive human labor involved. Causal relationship inference is a main topic in textual causality extraction domain, where there are by definition two types of causal relationships: explicit causality and implicit causality (Ittoo & Bouma, 2011). Explicit causality is explicitly expressed with causal cue words such as 'lead to' and 'to cause', while implicit causality exists without being marked by any cue words and is intrinsically hard to detect. One simple example is 'Congratulations. You won.' in which 'won' is the implicit cause of 'congratulations'. The implicit causality can sometimes be learned from explicit causality. For instance, the sentence 'Congratulations, because you won.' with the cue word 'because' indicates the relationship which is otherwise implicit. For tweets, as we need to find their causal backgrounds in news, all the causal relationships under discussion here are implicitly across different contexts, which have rarely been studied.

1.3. Contributions of present work

We initiate the first attempt to uncover causal backgrounds behind politicians' tweets from previous news articles by establishing a tweet interpretation framework. To tackle tasks in Fig. 1, we develop a clause-based open information extraction approach (Section 3.1) to extract meaningful structured tweet clauses (Task 1) from an unstructured tweet. We then identify the clause worth interpreting (Task 2) using a scoring model (Section 3.2), which incorporates novel scoring features, such as edit-popularity on Wikipedia and a deep-neural-network based term weighting scheme to assess the political relevance of terms (Section 3.4). To

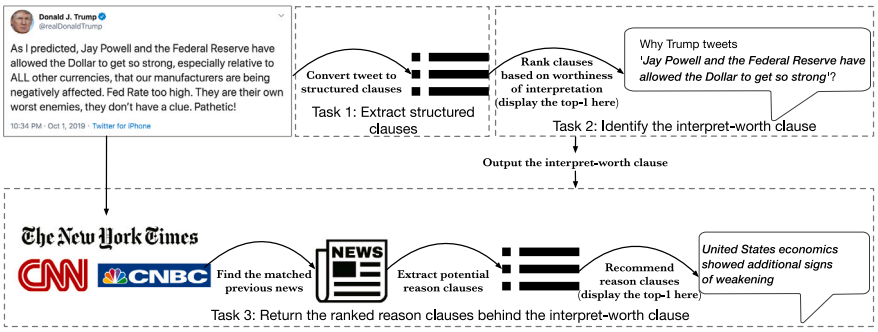

Fig. 1. An illustration example of Tweet Interpretation framework with input and outputs.

recommend reason clause (Task 3), the input tweet is further matched to previous news based on textual similarity, from which we then obtain potential reason clauses and link them with the interpret-worth tweet clause (Section 3.3), applying causal associations learned through dependency analysis.

Since there is no existing labeled data for our task to the best of our knowledge, we construct a dataset and make it publicly available.[1] The dataset contains the annotated contents of interest for 533 tweets from 5 US politicians, including Biden and Trump, and the political journalists' follow-up interpretations obtained through Google News search, as a *golden standard* for reason clause recommendation.

The proposed clause-based open information extraction approach (Task 1) outperforms the state-of-the-art ClausIE (Del Corro & Gemulla, 2013). The clause scoring model (Task 2) finds the content of interest within the top 2 recommendations on average. To evaluate the reason clause recommendations (Task 3), we feed the system with news from major agencies, and recruit 9 qualified annotators to annotate the generated result according to the *golden standard*. The evaluation results indicate that the true causal backgrounds can be found in its top 2 recommendations on average.

To sum up, this paper makes the following contributions:

- We initiate the first attempt to implement an end-to-end framework that automatically discovers causal backgrounds for politicians' tweets. Built and tested on real-world datasets, our framework yields results that correspond well with political journalists' analysis – the true causal backgrounds can be found within its top 2 recommendations.
- We propose a clause-based Open IE system with specific considerations on rule-free representations, outperforming classic clause-based systems in tweet extractions.
- To find the contents of interest, we overcome the lack-of-labeled-data problem by utilizing news and Wikipedia data to measure the contents' popularity and political relevancy. Specifically, we showcase how Wikipedia linkage can be used in a weakly-supervised fashion with a neural network model to weight a term's relevancy to a specific domain which can be particularly useful for text classification tasks without much labeled data.
- As a first attempt, we find implicit causalities between different contexts, namely tweets and news articles, using explicit causalities learned elsewhere. This approach surpasses common Q&A methods based on text similarity.
- To facilitate future research, we open source our framework and make a first well-curated tweet-intent-interpretation benchmark dataset publicly available.

The remaining parts are organized as follows. Section 2 briefly introduces the related work. Section 3 presents the tweet interpretation framework. The data used is described in Section 4. In Sections 5–8, we present the benchmark dataset, evaluate the framework, and conduct a case study, followed by a discussion on results and their implications in Section 9. We conclude this paper in Section 10.

2. Related work

As mentioned before, our framework extracts structured information from sources including news and social media, identifies the content of interest to the audience (term weighting), and performs causality inference. Accordingly, this section organizes and reviews relevant work from these three perspectives.

[1] https://anonymous.4open.science/r/TweetInterpretation-37C7.

2.1. Open information extraction

The aim of Information Extraction (IE) is to extract structured information from unstructured or semi-structured text. Much existing work is dedicated to a specific task that extracts two arguments with a set of predefined relationships (Zhang, Sun, Feng, & Li, 2020). However, whenever the domain of the documents changes, extensive work in relation annotation has to be done. To achieve domain-independence, Open IE is introduced as a new extraction paradigm to extract all types of relationships and their arguments from open domain text to express facts (Yates et al., 2007). Open IE systems represent natural languages as *n*-arity propositions, where a proposition consists of a subject, a relation, and any number of other arguments. When there are only two arguments and their relationship in a proposition, this 3-arity proposition is called a triple in the form of *<subject, predicate, object>*. There exist three types of Open IE systems: 1. learning-based systems (e.g., ReNoun (Yahya, Whang, Gupta, & Halevy, 2014)) which are self-supervised or supervised learning on labeled data; 2. rule-based systems (e.g., PredPatt (White et al., 2016)) which make use of hand-crafted extraction rules; 3. clause-based systems (e.g., ClausIE (Del Corro & Gemulla, 2013) and Stanford Open IE (Angeli, Johnson Premkumar, & Manning, 2015)) which decompose a sentence to generate propositions. Different from the first two types of systems, clause-based systems do not rely on labeled data, which is costly to create or hand-crafted extraction rules that are hard to exhaust on heterogeneous web texts. ClausIE utilizes linguistic knowledge of English grammar to extract *n*-arity propositions instead of triples, which can capture more complete results. In contrast, Stanford Open IE traverses the dependency paths of a syntax tree to construct clauses and outputs content consisting of discrete fragments of text.

While the target of Open IE is to generate machine-readable representations of the information, our system aims to present content similar to a human-generated query, which tends to be a continuous meaningful text piece (Chali & Golestanirad, 2016). Moreover, although automatically mining social media content is an active research topic, the usage patterns of some common informal rule-free textual expressions (e.g., idioms) have been somehow rarely extracted and utilized.

2.2. Term weighting for text mining

In text mining, it is a common task to identify the important content in a piece of text (Domeniconi, Moro, Pasolini, & Sartori, 2015) and it is often accomplished by term weighting. The earliest and most widely used approaches calculate term weights based on word frequency, such as TF-IDF (Ramos et al., 2003) and TF-ICF (Reed et al., 2006). When labeled training documents are available, derived approaches weight terms by employing metrics of feature selection (Debole & Sebastiani, 2004), such as information gain.

These approaches rely on pre-computed term frequencies and these are not available for new words or new named entities with more than one word. This is particularly common for web texts where new words and new named entities emerge all the time (Li et al., 2012), and, fortunately, they are often timely captured and further documented with links to related terms in crowd-sourced services such as Wikipedia (Wu et al., 2017). To incorporate Wikipedia's rich information in weighting terms with less human intervention, a promising option may be the Convolutional Neural Network (CNN) that succeed in many recent text classification tasks (Kim, 2014) with the attention mechanism that automatically focuses on important parts of the feature representations (embeddings) given the context (Yin, Schütze, Xiang, & Zhou, 2016). However, neural network models always require large amount of labeled training data which can be expensive and time-consuming to obtain.

2.3. Causality inference

We review studies of tweet intention classification and causality reasoning that are relevant to our task.

Previous intention classification studies classify the tweet types into fixed categories, usually very general (Kozlowski et al., 2020; Mohammad et al., 2015), instead of figuring out the specific reasons for posting the tweet.

Causality reasoning finds causality between two pieces of text. Both explicit causal relationships expressed with causal cue words (e.g., '*lead to*') and implicit relationships without indicators are explored. To extract explicit cause–effect pairs, most of the work relies on probability models (Wang & Chan, 2011), dependency structure (Hashimoto et al., 2014), and directed acyclic causal graph (Peters & Bühlmann, 2015). Recently, neural networks have been used for implicit causality reasoning (Zhao, Ji, He, Liu, & Ren, 2021) (Oh, Torisawa, Kruengkrai, Iida, & Kloetzer, 2017). While these studies find the cause and effect within a same document, tweets' context often cannot provide enough information. A similar problem occurs in the fact-checking studies where the claims need to be verified with facts, and one effective approach is to mine external knowledge and link the facts with the claims (Wang, Yu, Baumgartner, & Korn, 2018). However, causality inference across different media has been suggested (Wallace, Choe, Kertz, & Charniak, 2014) but rarely touched. A possible reason is that it is almost impossible to label general statements with their external reasons, unless the statements are triggered by important issues with traces on the Web, which is often the case in our scenario of politicians' tweets. Existing approaches for explicit causality inference are not applicable since trigger words are not available and neural network based implicit causality inference requires very large annotated data, which is not feasible in our task.

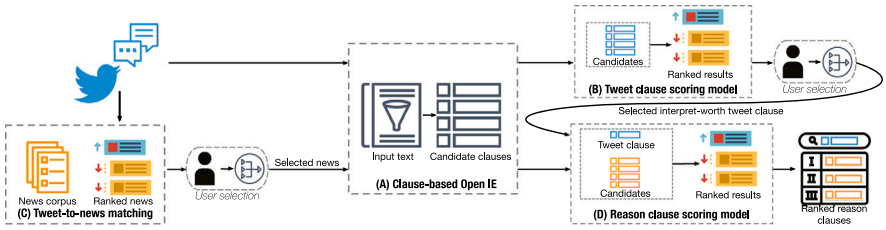

Fig. 2. Tweet interpretation framework.

3. Tweet interpretation framework

We consider the tweet interpretation problem as a procedure consisting of multiple tasks that can be pipe-lined in an interactive tweet interpretation framework (Fig. 2). We explain the framework by following the three tasks it attempts to accomplish for each input tweet:

Task 1: Extract structured clauses in the tweet (Section 3.1). This is done by Module (A) in the figure. The candidate clauses will be inputs for Task 2.

Task 2: Rank tweet clauses based on the worthiness of interpretation (Section 3.2). Module (B) performs the ranking. The top results or the ones picked by the end user will be interpreted in the next task.

Task 3: For an interpret-worth tweet clause, return its ranked reason clauses from previous news (Section 3.3). This involves matching the original tweet that contains the tweet clause with previous news articles (Module (C)). From the automatically matched article or user-picked article, the framework extracts clauses (Module (A)), ranks them (Module (D)) to recommend reason clauses.

In the following subsections, we will introduce each task in turn. Before that, we formally define '*clause*' and '*reason*' in our context:

A *clause*, similar to the definition in the classic ClausIE, is a part of a sentence that expresses some meaningful information. It consists of at least one proposition that contains a verb or a preposition as the core element.

A causal background of an interpret-worth tweet clause is a past event that triggers or explains tweeting (e.g., '*Trump defended his abrupt decision to withdraw American troops from northern Syria*' explains why Trump tweets '*I am trying to end the ENDLESS WARS*'), represented by a *reason clause*.

3.1. Clause-based open IE system

In our pipeline, a tweet is decomposed into clauses in ***Task 1*** and these clauses will be further selected and interpreted. To end-users of the system, these clauses are expected to be meaningful to imitate human-generated queries (Chali & Golestanirad, 2016), such that they can be easily made sense of. Similarly, in ***Task 3***, the reason clauses from the news articles also need to be well understood by humans. As mentioned in Section 2.1, traditional Open IE systems that pursue machine-readability and content completeness are not designed for this specific purpose and may not perform well when compared with human annotations. Therefore, we propose a novel clause-based Open IE system (Module (A) in Fig. 2) with two stages: 1. generating *meaningful propositions* (Section 3.1.1); 2. combining multiple propositions into a clause to achieve higher content completeness.

3.1.1. Proposition extraction

Based on previous studies, we define the standard for a *meaningful proposition*. First, it has to be related to at least one specific entity. Traditional Open IE systems, such as ClausIE (Del Corro & Gemulla, 2013) and Stanford Open IE (Angeli et al., 2015), require the extracted propositions to contain at least a noun. However, in our scenario where we try to discover specific causal backgrounds, represented by events behind tweet clauses, a noun is often too vague and general. In event extraction studies, an event is usually defined to contain at least one named entity (Zhou, Chen, & He, 2015). Hence, it is intuitive that named entities can help specify events. To find more propositions that contain information of interest, we relax the constraint to also include the nouns that are once bounded to named entities in the context. For instance, in the news content in Fig. 3, *'fire'* alone is no longer a vague noun as the context contains *'fire on Monday'* where *'Monday'* is identified as a named entity. Therefore, we require any extracted proposition to contain at least one named entity or a noun related to a named entity, to be *meaningful*. This applies to proposition extraction in both tweets and news articles, with the difference being that, for the extraction in news articles, the nouns are relaxed to contain the ones from tweet proposition extraction for them to be linked up to the tweets. To be specific, we denote the extracted name entities and related nouns in a tweet as its *tweet_keyword_set*.

Second, the form of a *meaningful proposition* should allow us to extract all meaningful facts contained in a text. As discussed in Section 2.1, traditional 3-arity propositions are concise and easy to understand compared to the over-specified information provided by n-arity propositions with n over 3. Previous approaches incorporate syntactic constraints to encourage the meaningfulness of extracted propositions, such as (Fader, Soderland, & Etzioni, 2011). The 3-arity propositions use verbs as a bridge indicating the relationships between the parts. However, the frequency of verbs is relatively lower than other parts of speech (e.g., prepositions) (Stevenson, 2010), resulting in a lower coverage of textual fragments with relationships. On the other hand, prepositions, as the

Table 1
Patterns and examples of prepositional triples.

No.	Pattern	Example
1	N-P-N	A lot of **winning** in Kentucky.
2	N-P-VL	Looks like Pelosi and Co. are putting on **a show** to **appease the Democratic base** . . .
3	V-P-N	Looks like Pelosi and Co. **are putting** on **a show** to appease the Democratic base . . .
4	V-P-VL	It is not clear if Iran **will be permitted** to **export oil** in exchange for EU.
5	VL-P-N	. . . cut rates again unless United States economics **showed additional signs** of **weakening**.
6	VL-P-VL	Jay Powell and the Federal Reserve **have allowed the Dollar** to **get so strong**.

[1] P: preposition, V: verb, VL: verb-like structure.

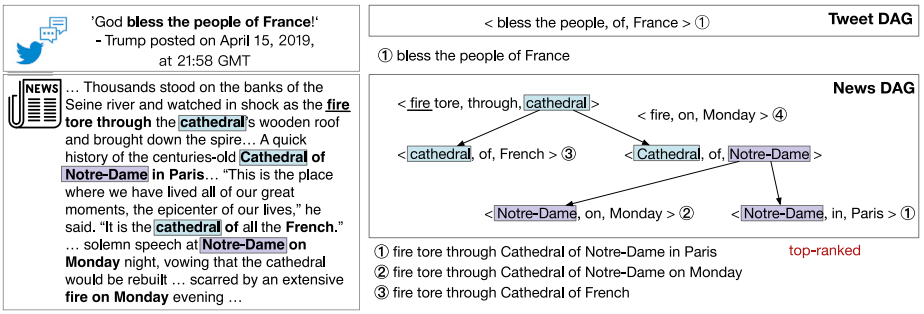

Fig. 3. From extracted propositions to clauses, with one tweet example (top left) and one news article example (bottom left). The two corresponding directed acyclic graphs (DAGs) are on the right. Underlined text represents the supplemented subject of the proposition.

most frequent part of speech, capture the relationships between nouns/pronouns and other words, and guarantee a higher coverage. Therefore, we keep the classic three-element form and use prepositions as the central element instead to extract propositions in the pattern of *<left element, preposition, right element>*. As in Fig. 3, *'fire on Monday'* is converted to *<fire, on, Monday>*.

We further customize the proposed form to extract a larger variety of meaningful propositions. Previous research suggests that about 40% of the propositions in web texts contain more than 3 arities (Christensen, Mausam, Soderland, & Etzioni, 2011). This inspires us to enrich the prepositions' left and right elements by adding more syntax rules. Like ClaueIE, we require the left elements to be nouns, verbs, or verb-like structures and right elements to be nouns or verb-like structures, where a verb-like structure corresponds to a verb followed by any combination of nouns, adjectives, and adverbs (e.g., '*bless the people*' in Fig. 3 or '*get so high*'). Examples of these patterns are presented in Table 1. We name this new form of three-element representation as a **prepositional triple**, which is one of the propositions' forms in our setting.

We apply some common pre-processing techniques before extracting the **prepositional triples**, including treating noun phrases (e.g., '*current economic weakness*') as nouns, merging words of the same part of speech connected by a conjunction (e.g., '*A and B*'), and combining consecutive verbs (e.g., '*have allowed*'). Identifying noun phrases and part of speech, as well as named entity recognition, is done by spaCy,[2] a widely used package, in the whole framework.

Tweets are often mixed with daily-life idioms (Grzeça, Becker, & Galante, 2020) and we encounter them frequently, especially phrasal verbs, which may have varied usage patterns. For instance, *'bring home'* as a phrasal verb, can be used either in the form of *'bring home somebody'* or *'bring somebody home'*, while another phrasal verb *'grow up'* can only be used in an intransitive manner. Formally, a phrasal verb is a verb made up of the main verb followed by an adverb or a preposition, or both. We start with a list of phrasal verbs from UsingEnglish[3] to look for their example sentences in Youdao dictionary[4] and we combine them with similar usage instances from the crowdsourced Urban Dictionary.[5] From these examples, we extract usage patterns of phrasal verbs.[6]

3.1.2. Clause generation

A human-generated query usually spans over multiple propositions (Athreya, Bansal, Ngomo, & Usbeck, 2021). To approximate human annotations, our extracted propositions with overlaps can be further concatenated through a 'head-to-tail' fashion into a more complete clause. A similar merging strategy is used to extract semantic relations (Al-Zaidy & Giles, 2018). However, direct concatenations often lead to clauses without subjects, because, by design, the subjects may not be present in propositions. Therefore, before concatenating overlapped propositions, we supplement the subjects for propositions that start with a verb by searching their nearby named entities and related nouns on the left. For instance, proposition *'tore through Cathedral'* in Fig. 3 is supplemented with

[2] https://spacy.io/.
[3] https://www.usingenglish.com/reference/phrasal-verbs/list.html.
[4] http://www.youdao.com.
[5] http://www.urbandictionary.com.
[6] A detailed list is available in our public repository.

the subject *'fire'*. For any overlapped content, the proposition on left is denoted as the left-block, while the one on the right is denoted as the right-block. We use a directed acyclic graph (DAG) structure to represent the connections between propositions by setting the left-block as the incoming neighbor of the right-block, and output all feasible paths from each source node to its connected sink nodes and all isolated nodes as clauses. Fig. 3 shows the DAGs constructed for a tweet and a news article, respectively. Through the DAGs, the clauses, such as *'fire tore through Cathedral of Notre-Dame in Paris.'*, are extracted.

As a side note, our system can also solve traditional Open IE tasks by removing the keyword set.

3.2. Scoring model for finding the focal clause

To help end-users quickly find the focal clause out of the candidate clauses, the framework needs to rank them (***Task 2*** and ***3***). In ***Task 2***, given a set of tweet clauses, we aim to rank them by the worthiness of interpretation. In ***Task 3***, given clauses from a news article matched with the tweet, the goal is to find the clause most likely to be the reason for an interpret-worth tweet clause.

This process is similar to finding the check-worthy sentences in fact-checking related research, whereas no labeled data is available for our task that aims at a different target. Consequently, we cannot select features in a data-driven fashion. Instead, we carefully design features based on a semantic analysis (Section 3.2.1) to form a clause scoring model (Section 3.2.2). Besides these features, the recommended reason clause in ***Task 3*** is also measured by its causality to the interpret-worth clause, which will be described in Section 3.3.3.

3.2.1. Features to score clauses

This section presents five features to score clauses, including *popularity in the news corpus*, *popularity of Wikipedia editing*, *political relevance*, and *verb frequency*, as well as a *sentence importance* feature particularly for scoring clauses in news articles. For all features, we conduct the ablation study, which evaluates a feature by removing it and examining the change in model performance.

News-corpus-popularity: Intuitively, popular named entities in the news are the ones that get more media attention and may be of interest to the audience (Abbar, Castillo, & Sanfilippo, 2018). Hence, the news-corpus-popularity feature is designed to represent the frequency of a clause's named entities in our news corpus.

Wiki-page-edit-popularity: Wikipedia's editing history reflects entities' saliency, since it provides a view of editors' reactions to salient topics (Al Tamime, Giordano, & Hall, 2018). We calculate the Wiki-page-edit-popularity of a clause by summing up the number of valid recent edits and editors of all named entities in the text.

***Political-relevance*:** While much of the content in a tweet is noisy (Liu, Fu, & Chen, 2020), politically relevant parts of politicians' tweets may be relatively more attractive in our scenario. We modify the Attention-based Convolutional Neural Network (ABCNN) (Yin et al., 2016) to model the relevancy degree as the political-relevance feature, which will be elaborated in Section 3.4.

Verb-frequency: Since the verb is the main component that reveals the causal relation between events (Riaz & Girju, 2013), focal clauses are prone to contain verbs. Besides, some other parts of speech can be converted into verbs, such as *'completion'* means *'complete'* in *'completion in the spring'*. Therefore, while calculating verb-frequency, we also include the frequency of words that can be converted into verbs according to WordNet (Soergel, 1998).

Sentence-importance: The classic TextRank (Mihalcea & Tarau, 2004) method computes sentence importance within a document. This helps us identify the important content of a news article from a textual perspective at the time of its publication. We are curious to see whether it could still help given that tweets to be interpreted are generated later with possible different focuses.

3.2.2. Scoring tweet clauses

Based on the first four features, we score a tweet clause as summation of its normalized feature values across all clauses in the tweet:

$$Saliency\ score\ of\ the\ i\text{-th}\ clause = w \sum_k a_k \frac{x_i[k]}{\sum_j x_j[k]} \tag{1}$$

where $x_i[k]$ is the kth feature score of the ith clause, a_k is the coefficient of each feature k, w is the inverse of the word count of the ith clause, k and j are the number of features and clauses, respectively. The weight a_k for each feature is determined by a grid search on a validation set of tweets and the best weight combination corresponds to the best overall ranking of the clauses deemed correct by humans. After scoring, the clause with the highest saliency score is automatically selected as **the interpret-worth clause**. We also provide the top results as a recommendation list for human annotators to choose from, given human preferences may vary.

3.3. Finding reason clauses from news

To help end-users understand an interpret-worth tweet clause with external knowledge, the framework recommends reason clauses by mining previous news. This is divided into three stages: 1. match the tweet with previous news (Module (C) in Fig. 2); 2. extract candidate reason clauses from the matched news article (Module (A)); 3. rank candidate reason clauses (Module (D)).

3.3.1. Tweet-to-news matching

As mentioned in Section 2.3, previous news articles may contain causal backgrounds for tweeting. To find the most relevant news article, we rank those published within 15 days before the tweet. Though widely used in recent text ranking tasks, BERT can be computationally intensive for long text (Nogueira & Cho, 2019) and therefore is less suitable for time-critical tasks. The traditional ranking method Okapi BM25 only considers textual similarity weighted by TF-IDF (d. manning, raghavan, & schütze, 2008), while in our scenario, other weighting factors may be at play. We design our approach based on two observations: 1. Relevant news articles usually contain more keywords in the *tweet_keyword_set*. 2. These keywords' appearances in different parts of a news article imply differences in importance. For instance, titles are one of these 'spotlight' locations. Hence, a news article's relevancy to the tweet is measured as a weighted sum of *tweet_keyword_set*'s element occurrences in distinct parts of the news, namely *URL*, *title*, *abstract*, *lead paragraph*, *body*, *news keywords*, and *section name*, provided by news media. The news article with the highest relevancy score is matched with the tweet.

3.3.2. Reason clause extraction

From the matched news article, we extract structured clauses using the algorithm proposed in Section 3.1. Since the extracted clauses should be tweet-related, we restrict the clause to contain at least one keyword from the *tweet_keyword_set*. However, same meanings can be expressed differently without using the exact same words. Therefore, we introduce synonyms (provided by conceptnet.io) to supplement the *tweet_keyword_set*. Here, we only extend politics relevant keywords (identified by the model in Section 3.4) to ensure the quality.

3.3.3. Reason clause recommendation

We rank the extracted reason clauses based on their saliency scores. The saliency score for a reason clause is a summation of various features, similar to that of a tweet clause (Section 3.2.2), but with two differences: (1) We use the ***sentence-importance*** value to replace the original w to give the important sentences in a news article more weights whereas in short text like tweets, there are often very few sentences and their importance is hard to measure based on word co-occurrences (Gao et al., 2019); (2) One additional feature, ***causality***, is proposed, since we are trying to link news contents to tweet contents and we prefer the ones that may cause the tweet clauses with causal word-pair evidences.

Because we are finding causal connections between two sources of text, namely tweets and news articles, the explicit causality that only exists within a same context and is identified by cue words such as 'lead to' is not what we are looking for. As mentioned in Section 1, though difficult to find, implicit causality can be captured by a set of causally related words that are explicitly extracted. For example, '*congratulations*'-'*win*' explicitly extracted from '*Congratulations, because you won*' can help reveal the implicit causal relationship between the sentences '*Congratulations.*' and '*You won.*'.

We use the causal cue words provided by (Kayesh, Islam, & Wang, 2019) to recognize explicit cause and effect phrases in historical news corpus, from which we form causal word pairs. However, these phrases usually contain unnecessary information that generates low-quality word pairs. Fortunately, research shows that the root word of a phrase's syntax tree is the most important word (Li et al., 2008) and, accordingly, we only keep the root word and its largest subtree's root word as the core content of a phrase. We then further filter out the words whose stems are '*be*' and words whose part-of-speeches are pronouns, conjunctions, and numerals. A causal word pair contains one word from the cause phrase and one word from the effect phrase, and a pair is generated for each cause-word and effect-word combination. Given the interpret-worth clause and a candidate reason clause, the ***causality*** score is incremented whenever a cause-word and its paired effect-word appear in the reason clause and the tweet clause respectively.

3.4. Scoring domain relevancy

In Section 3.2.1, the *political-relevance* feature is designed to score clauses based on their relevancy to politics. To score a term's relevancy to a specific domain including politics in our scenario, a scoring scheme is needed. Most of the existing scoring scheme weights a term according to its frequency in a corpus. However, most corpora do not reflect the most up-to-date terms, which could be problematic when dealing with tweets. Wikipedia pages, on the other hand, are created and updated in a timely manner, capturing emerging terms.

Thus, given a term, we can obtain information, often in summary (Tang, Chen, Cui, & Wei, 2019), from its real-time Wikipedia page to form its features and feed them into a neural network, which generates a political relevancy score. Moreover, Wikipedia provides a ranked list of page backlinks (links from other terms' pages to the current page) for a term's page, where important pages with similar topics have higher weights in a PageRank-like method (Nie, Davison, & Qi, 2006). Therefore, we use the page summary and the titles of its top backlink pages to incorporate text information and relevancy between terms.

Although the summary in a Wikipedia page may include important information that can help us distinguish the domain relevancy of its title, it also contains unnecessary information. As established in Attention-based CNN (ABCNN) proposed in (Yin et al., 2016), words grouped by embedding similarities can be used to find sentence pair relations and identify the key words in both sentences. This inspires us to further find the important content in a summary and in backlink page titles. We obtain a matrix with rows and columns respectively representing words in the summary and backlink page titles, each location recording the corresponding similarity. We use this matrix through an attention mechanism to adjust the weights for the summary word embeddings. Similarly, we construct a matrix that records the pairwise similarity within the backlink page titles to adjust their embeddings.

We construct features for ABCNN and also modify its architecture to adapt to our scenario, where we want to compute the domain relevancy for each term.

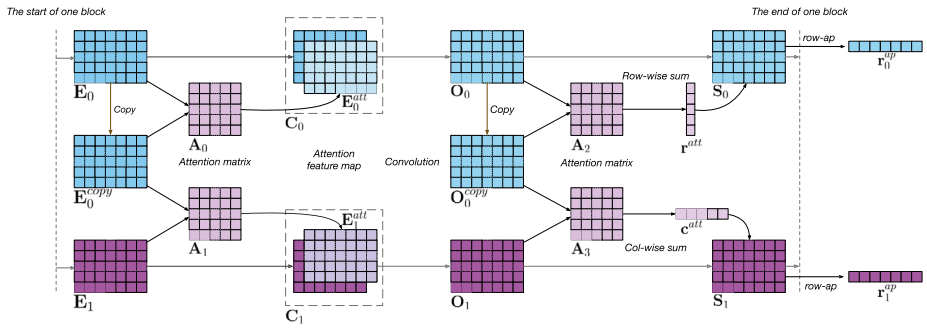

Fig. 4. One block of the modified Attention-based Convolutional Neural Network model.

Input: ABCNN takes two sentences of the same length represented by concatenated word embeddings as its input. In our case, we want to score a term by its domain relevancy. We select top-s_0 backlinks (denoted as *backlink-features*) and first s_0 words in the summary (denoted as *summary-feature*) of the term's Wiki page, and convert them into embeddings (denoted as $\mathbf{X}_0$ and $\mathbf{X}_1$) as the input of our model. $\mathbf{X}_0$ is generated by concatenating the embeddings of *backlink-features*, where a *backlink-feature*'s embedding is the average of its word embeddings. $\mathbf{X}_1$ is a concatenation of word embeddings of the summary. These word embeddings are given by the pre-trained GloVe model (Pennington, Socher, & Manning, 2014).

Network architecture: Our network architecture is built with two identical blocks and we present the design for one of them in Fig. 4.

The inputs to the block are $\mathbf{E}_0 \in \mathbb{R}^{s\times d_l}$ and $\mathbf{E}_1 \in \mathbb{R}^{s\times d_l}$, where d_l is the dimension of input to block l and $l \in \{0, 1\}$. Instead of generating an attention matrix for two inputs at one layer in the original network design, we create two different attention matrices, $\mathbf{In}_1$ and $\mathbf{In}_2$ for them:

$$Attention(\mathbf{In}_1, \mathbf{In}_2)_{i,j} = \frac{1}{1 + \sqrt{\sum_k (\mathbf{In}_1[i,k] - \mathbf{In}_2[j,k])^2}} \tag{2}$$

For $\mathbf{E}_0$ and $\mathbf{E}_1$, two attention matrices, A_0 and A_1, are calculated with $Attention(\mathbf{E}_0, \mathbf{E}_0^{copy})$ and $Attention(\mathbf{E}_0^{copy}, \mathbf{E}_1)$ respectively, where $\mathbf{E}_0^{copy}$ is a copy of $\mathbf{E}_0$. We then generate attention feature maps:

$$\mathbf{E}_0^{att} = \mathbf{A}_0 \cdot \mathbf{W}_0, \mathbf{E}_1^{att} \quad = \mathbf{A}_1^T \cdot \mathbf{W}_1$$

where $\mathbf{W}_0, \mathbf{W}_1 \in \mathbf{R}^{s\times d_l}$ are weight matrices. For each $t \in \{0, 1\}$, we stack $\mathbf{E}_t$ and $\mathbf{E}_t^{att}$ to generate a 3-dimensional vector denoted by $\mathbf{C}_t \in \mathbb{R}^{2\times s\times d_l}$ as the input to the convolution layer which outputs $\mathbf{O}_t \in \mathbb{R}^{s\times d_{l+1}}$, with the convolution weight $\mathbf{W}_t^{conv} \in \mathbb{R}^{d_{l+1}\times d_l}$ and bias $\mathbf{b}_t^{conv} \in \mathbb{R}^{d_{l+1}}$:

$$\mathbf{O}_t = \tanh(\mathbf{W}_t^{conv} \cdot \mathbf{C}_t + \mathbf{b}_t^{conv})$$

We then make a copy of $\mathbf{O}_0$, denoted as $\mathbf{O}_0^{copy}$, and calculate $\mathbf{A}_2$ as $Attention(\mathbf{O}_0, \mathbf{O}_0^{copy})$ and $\mathbf{A}_3$ as $Attention(\mathbf{O}_0^{copy}, \mathbf{O}_1)$, respectively. Let $\mathbf{r}_i^{att} = \sum_j \mathbf{A}_2[i,j]$ and $\mathbf{C}_i^{att} = \sum_j \mathbf{A}_3[j,i]$. We generate feature maps $\mathbf{S}_0$ and $\mathbf{S}_1$ by:

$$\mathbf{S}_0[i,j] = \mathbf{r}_i^{att}\mathbf{O}_0[i,j], \mathbf{S}_1[i,j] \quad = \mathbf{C}_i^{att}\mathbf{O}_1[i,j]$$

where $\mathbf{S_0}$ and $\mathbf{S_1} \in \mathbb{R}^{s\times d_{l+1}}$. We perform the *row-ap* (col-wise averaging over all rows) on $\mathbf{S_0}$ and $\mathbf{S_1}$ and store the outputs as $\mathbf{r}_0^{ap}$ and $\mathbf{r}_1^{ap}$, respectively. Finally, the lth block returns $\mathbf{S}_0$ and $\mathbf{S}_1$ as the inputs to the $(l+1)$-th block.

For the first block, the inputs $\mathbf{X}_0$ and $\mathbf{X}_1$ have word embedding dimension $d_0 = 300$, and $s = s_0 = 50$. The output dimension of this two-block network d_2 is 2, corresponding to the probability that a term belongs to politics-relevant and politics-irrelevant, respectively. While the input dimension d_1 of the second block is adjusted during training. We stack all *row-ap* outputs of these two blocks, as well as *row-ap* outputs of $\mathbf{X}_0$ and $\mathbf{X}_1$. The aggregated feature map fed into the output layer, which is a fully connected layer with a softmax layer, generates two values representing the probability of being politically relevant or irrelevant, respectively. We use the binary cross entropy as the objective function for training. The model selects the class with higher probability as the predicted class.

It is worth noting that collecting labeled data for training the ABCNN is time consuming. To address the problem, we use a weak supervision strategy to train our model based on pseudo-labels generated under a strong assumption that a Wikipedia page and its internal links (hyperlinks on the page to other pages) are related to a same domain. Accordingly, we can obtain plenty weak supervision data by selecting pages under politics relevant/irrelevant titles, together with the pages their internal links point to as positive/negative samples. The details on data acquisition will be described in Section 4.3.

While our goal is to assess political relevancy, the model can be applied to assess relevancy for other domains by replacing the selected samples.

Table 2
The descriptive statistics of the collected data.

Origin	Size	Time range	Avg length	Source
Twitter	22,840 tweets	03/01/2019 - 02/29/2021	38.55 words	Five most tweeted US politicians
News	147,619 articles	01/01/2019 - 02/29/2021	877.01 words	NYT, CNN, and CNBC
Wikipedia	13,222 pages	NA	228.08 words in the summary	15 seed Wikipages about US politics

4. Data collecting and processing

As stated in Section 3, our framework is built upon tweets, news, and Wikipedia data. We describe how to obtain and process them in this section. Table 2 summarizes their descriptive statistics.

4.1. Twitter

We collect tweets from five politically active US politicians that are also active on Twitter,[7] namely Joe Biden, Donald Trump, Bernie Sanders, Hillary Clinton, and Alexandria Ocasio-Cortez, for two years (March 1, 2019 to February 28, 2021). After discarding reposted tweets and merging consecutive tweets (those end or start with '...' or '..'), we obtain 9,458 tweets from Trump and 13,382 tweets from the other four. To construct a high-quality dataset, we keep the tweets with more public attention, measured by numbers of comments and search each sentence of them on Google News to find the tweets that are interpreted by political journalists in follow-up analysis. This leaves us 533 tweets, which will serve as inputs to our framework and be annotated for evaluation purposes. These tweets appear to be more informative – on average, each contains 46.1 words and 5.0 Named Entities, versus 38.5 words and 2.8 Named Entities for the tweets filtered out.

4.2. News

To develop a reliable news database for finding causal backgrounds behind tweets, we crawl 'US politics' and 'world' sections from *CNN* and all sections from *New York Times* (*NYT*) and *CNBC*. As a result, we obtain 147,619 valid articles from January 1, 2019 to February 28, 2021. The content of news including release time, title, main body, URL, section name, the leading paragraph, abstract, and ordered keywords is obtained. In addition, abstract and ordered keywords are only available in the *NYT* data.

Besides, one million news articles provided in Corney, Albakour, Martinez, and Moussa (2016) are used to extract a total of 441,341 **causal word pairs**.

4.3. Wikipedia

We obtain edit history, backlinks, internal links, and summaries of Wikipedia pages. To collect training and validation sets for the ABCNN model in Section 3.4, we collect politics related terms and non-politics related terms. We hand-pick 15 pages about US politics (e.g., *Politics of the United States*) and have internal links that point to more Wikipedia pages. We use the titles of these pages as positive samples. Similarly, negative samples are obtained through 5 random non-political related pages' internal links. As a result, we get a training set of 10,000 samples (5000 positive and 5000 negative) and a validation set of 3222 samples (1339 positive and 1883 negative). Since these samples are weakly supervised, we manually label a test set with 156 positive samples and 157 negative samples for more accurate evaluation.

5. Ground truth and evaluation metrics

In this section, we describe the construction of ground truth for each evaluation task and present the evaluation metrics. The tasks to be evaluated are: (1) structured clause extraction (Section 3.1); (2) focal clause identification (Section 3.2); (3) matching of the tweet and news articles (Section 3.3.1); and (4) reason clause identification (Sections 3.3.2 and 3.3.3). Terms' domain relevancy scoring (Section 3.4) as a standalone component in (2) and (4) is evaluated separately as a fifth task.

5.1. Tweet clause generation

For a tweet, the framework decomposes it into clauses to be interpreted. Ideally, these clauses should be what humans would care about in the tweet and further seek causal backgrounds for. We measure their quality by quantifying their overlap with the tweet content highlighted by human annotators.

[7] https://blog.twitter.com/en_us/topics/insights/2019/ThisHappened-in-2019.html.

5.1.1. Ground truth

For each of the 533 tweets in Section 4.1 , two undergraduate students with English proficiency work independently to highlight one piece of content that is of interest. A graduate student verifies these results and when the two results for a tweet are not consistent, the graduate student uses her own judgement to pick one result.

5.1.2. Metrics

We use Jaccard similarity to measure how a generated clause overlaps with the ground truth clause. It computes the percentage of shared n-grams among all unique n-grams in both the clause and its corresponding ground truth:

$$Jaccardsim(a,b) = \frac{|n_a \cap n_b|}{|n_a \cup n_b|} \tag{3}$$

where n_a is the set of n-grams in the generated clause a and n_b is the set for the ground truth b. We put a threshold on the Jaccard score to identify correct generated clauses and calculate their ratio in all candidate clauses (i.e., precision) for each tweet.

5.2. Interpret-worth tweet clause recommendation

After we obtain the candidate clauses for a tweet, we will rank them according to their worthiness of interpretation using the scoring mechanism in Section 3.2. We examine how the correct clause judged by humans is ranked among all candidates using classic metrics for evaluating recommender systems.

5.2.1. Ground truth

We reuse the human-highlighted content of interest for evaluating clause generation in Section 5.1.1. For each tweet, we manually examine the generated clause that has the highest Jaccard similarity to the human-highlighted content and decide whether it is a match. If it is not a match, the algorithm fails to generate the correct interpret-worth clause for the tweet. We record the position of the correct clause on the list. In cases where the correct interpret-worth clause is not generated, the ranking position of the correct result is set to infinity.

5.2.2. Metrics

We use two common metrics, percentage of questions answered correctly (PerCorrect) and mean reciprocal rank (MRR) (Shah, Ravana, Hamid, & Ismail, 2019), to measure how the correct results are ranked on recommendation lists. PerCorrect(k) calculates the percentage of test samples with at least one correct result in the top-k recommended items. MRR provides the mean reciprocal rank of the first correct result:

$$MRR = \frac{1}{|\mathcal{T}|} \sum_{t_i \in \mathcal{T}} \frac{1}{z_i} \tag{4}$$

where z_i is the first correct result's ranking position for tweet t_i in corpus $\mathcal{T}$.

5.3. Tweet-to-news matching

The tweet-to-news matching module (Section 3.3.1) matches tweets with previous news articles, from which further procedures will locate causal backgrounds for interpret-worth tweet clauses. We need to evaluate the list of news ranked by similarity score for each tweet and a good algorithm will rank the human-picked correct match higher. We manually traverse the ranked news list from the top to see where there is a match and record the ranking position of the first match. If no news article can match the tweet, the ranking position is set to infinity. Similar to Section 5.2, the performance is evaluated with MRR and PreCorrect(k) scores based on the obtained ranking positions.

5.4. Reason clause recommendation

To recommend causal backgrounds for a tweet's interpret-worth clause, the framework mines reason clauses from the matched news article and ranks them to output a reason clause recommendation list. We expect the model to rank the correct reason clauses high on the list, where the correct clauses are obtained by crowdsourcing annotations for an unbiased evaluation. Thus, similar to the aforementioned recommendation tasks, the ranking position of the correct clause is evaluated by MRR and PerCorrect(k).

5.4.1. Ground truth

Even with the follow-up experts' interpretation provided as benchmark, judging the correctness of reason clauses would require analytical proficiency in international politics and economics. Moreover, multiple causal backgrounds can be considered as generally correct for each interpret-worth clause. Therefore, qualified annotators from the above mentioned backgrounds shall be recruited, and the annotation task is designed as selecting the correct ones, if any, to generate the ground truth. For each tweet clause, we find the reason clause(s) voted by the majority as **top-voted-reason(s)**. A special case is that no candidate reason clauses receive more than one vote . In this scenario, to ensure the fairness of the MRR and PerCorret(k) which seeks 'the ranking position of the first correct result' from a list of ranked results, this ranking position is computed as the average ranking of all the reason clauses with 1 vote.

We further measure the degree of consensus among the annotators. For each annotator, we calculate the probability of picking the clause(s) voted by majority across all his or her annotations. This probability is averaged by annotators to compute the Average Probability of choosing the Majority Class (APMC).

5.5. Term domain relevancy scoring

The module in Section 3.4 scores a term's relevancy to politics. Though it outputs relevancy probabilities as scores used for ranking the tweet clause candidates, we evaluate it as a binary classifier since we only have binary labels. The datasets for training, validation, and testing are from Wikipedia as described in Section 4.3, which contains terms (i.e., Wikipedia page titles) and their binary labels (1 for political VS 0 for non-political). We put a threshold on the model's output probability to get the binary classification for each term in the test set.

6. Benchmark dataset

The benchmark dataset contains 533 tweets from five most tweeted about American politicians, namely, Joe Biden, Donald Trump, Bernie Sanders, Hillary Clinton, and Alexandria Ocasio-Cortez. For each tweet, the dataset provides: (1) the interpret-worth tweet clause (Section 5.1.1); (2) follow-up analysis by political journalists (Section 4.1); and (3) the correct reason clause(s) marked by each annotator (Section 5.4.1). We use (1) to evaluate interpret-worth clause extraction and (2) for annotators to evaluate the correctness of generated reason clauses. (3) is for other studies to compare with our results.

7. Evaluation results

In this section, we evaluate the framework's 5 tasks using the setup in Section 5 and discuss the results.

The 533 annotated tweets in Section 4.1 are the inputs for both Sections 7.1 and 7.3, while the other input to Section 7.3 is the news articles collected in Section 4.2. The generated clauses from Section 7.1 are delivered to Section 7.2 to produce tweet clauses. These clauses as well as the matched news from Section 7.3 are then fed to Section 7.4 to generate the reason clause. Additionally, on the Wikipedia data (Section 4.3), we learn to score terms' relevance to politics, evaluated in Section 7.5. The evaluation process mimics its usage in the real world with user interactions, in which users choose from recommended interim results and the system proceeds with the users' selections.

The framework's core code, the complete test data with human annotations and reference content are made publicly available.[8]

7.1. Tweet clause generation results

We evaluate our proposed clause-based Open IE system (Section 3.1) on generating clauses for each of the 533 tweets, using the settings in Section 5.1. We list the methods to be compared with:

- **Stanford Open IE.** State-of-the-art clause-based method based on dependency paths in syntax trees to generate machine-readable discrete text.
- **ClausIE.** Another state-of-the-art clause-based method that uses linguistic knowledge to generate human-readable clauses.
- **TextRank.** A simple and classic sentence-level ranking method.
- **Proposed method.** It differs from Stanford Open IE and ClausIE, as it introduces prepositions as the bridge of **prepositional triples** for higher coverage of textual fragments, instead of verbs. Besides, it is customized to process tweets which are often informal, by capturing phrasal verbs.
- **Proposed method without concatenating propositions.** The proposed method concatenates multiple propositions to form a clause. We want to verify whether this helps improve the results.

As described in Section 5.1, Jaccard is calculated for each tweet clause to measure how it overlaps with the ground truth. Thresholds are then put upon it to identify correct clauses and calculate the precision for each tweet. The precision scores are then averaged over all tweets as shown in Fig. 5. Our method outperforms others in comparison by large margins, at all Jaccard thresholds. Specifically, when the Jaccard threshold is set to 0.8, our method's average precision is 13.4%, 5.6% higher than our method without concatenation and 8.2% higher than the classic ClausIE which comes third in this comparison. Though much more effective than other methods, the average precision for our method does not seem high (36.1%@0.5 and 10.8%@1). This is not necessarily a big issue, since this step only tries to generate candidate clauses. They will be further ranked in the next step and it is the top ones that the users would care about. On average, our method generates 2.81 clauses for each of the 533 tweets.

[8] https://anonymous.4open.science/r/TweetInterpretation-37C7.

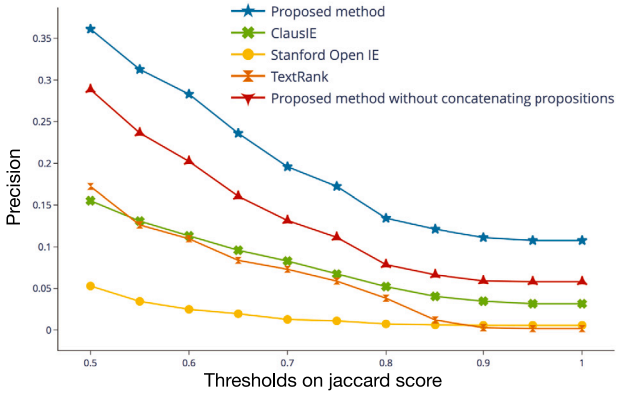

Fig. 5. Average percentage of correct extractions at different Jaccard thresholds.

Table 3
The MRR and PerCorrect scores for tweet clause recommendation.

Method	MRR	PerCo(1)	PerCo(2)	PerCo(3)	PerCo(4)	PerCo(5)
BERT-based TextRank	72.4%	55.7%	77.0%	86.7%	91.7%	93.7%
Our method	75.5%	60.4%	80.2%	88.3%	92.4%	93.7%

7.2. Interpret-worth tweet clause recommendation results

We rank the candidate clauses for each tweet from the last step, using the clause scoring model in Section 3.2. As described in Section 5.2, we examine how the 'correct' results are ranked by calculating the MRR and the PerCorrect(k) scores. As far as we know, scoring the worthiness of interpretation has rarely been touched in the literature. The classic TextRank algorithm serves a similar task. It tries to find the important pieces of text in an article, considering word co-occurrences. However, these measurements are not feasible in short text like tweets. A study shows that scoring the text similarity based on contextual embeddings generated by the BERT model can improve TextRank's performance (Zou et al., 2021). Therefore, we use word embeddings from a fine-tuned BERT model (Gupta, Pagliardini, & Jaggi, 2019) as the state-of-the-art in representing semantic relatedness for TextRank.

If each correct clause is ranked second among all candidate clauses for the corresponding tweet, the MRR would be 50.0%. As shown in Table 3, we achieve an MRR of 75.5%, which is 3.1% higher than BERT-based TextRank. It also suggests that our algorithm finds the correct clause within top 2 results on average. Besides, a PerCorrect(2) of 80.2% indicates that for 80.2% of the tweets, our algorithm finds the correct clauses within top 2 suggestions.

Furthermore, to examine how each scoring feature impacts the performance, we conduct an ablation study. Table 7 shows that the removal of any feature will lead to decrease in MRR. Specifically, the MRR score drops by 2.0%, 1.3% after removing **political-relevance**, **Wiki-page-edit-popularity**, respectively, suggesting that they are relatively more important than the other two features.

In this experiment, 520 out of 533 samples will be further fed to the next step. For 13 tweets, no correct clauses are found by humans and therefore the reason clauses cannot be suggested accordingly.

7.3. Tweet-to-news matching results

To find the most relevant news for a tweet, the tweet-to-news matching module (Section 3.3.1) ranks news articles from the dataset in Section 4.2 that are published within 15 days before the tweet was posted.

We compare our method with the widely-used probabilistic document retrieval algorithm Okapi BM25 (Robertson & Zaragoza, 2009) and a BERT-based approach (Nogueira & Cho, 2019), which take a tweet as the query and returns news articles ranked by text similarity.

With 520 tweets as inputs, we calculate the MRR and PerCorrect metrics as displayed in Table 4. Our method achieves an MRR of 63.9% which is 11.5% and 9.1% higher than the BM25 and BERT-based methods, respectively, indicating the correct matches are ranked higher by our method. Besides, our overall PerCorrect metrics are much higher. For instance, our PerCorrect(5) is 86.3%, with 19.4% and 7.1% improvements over the BM25 and BERT-based baselines, meaning that 86.3% of the correct matches can be found within top 5 results. It is worth mentioning that the BERT-based approach is about 300 times slower than our method, with 56 CPU cores, 256 GB of RAM, and 4 RTX2080 GPUs.

The matched news articles picked by humans will be passed onto the final step in which we generate ranked reason clauses from these articles. Since 13 out of the 520 samples do not have any matched article, the rest 507 tweet clauses with their corresponding news articles will be used.

Table 4
The MRR and PerCorrect scores for tweet-to-news matching.

Method	MRR	PerCo(1)	PerCo(2)	PerCo(3)	PerCo(4)	PerCo(5)
Our method	63.9%	46.7%	66.9%	77.9%	82.9%	86.3%
BERT-based	54.8%	39.0%	55.4%	64.6%	72.5%	79.2%
Okapi BM25	52.4%	41.0%	53.1%	59.2%	62.3%	66.9%

Table 5
Annotators' background information.

No.	1	2	3	4	5	6	7	8	9
Bachelor in	Political Science				International Relations		Economics	History	Markets & Culture
Country English	Spain bilingual	US native		Venezuela bilingual	Hungary bilingual	UK native	Denmark bilingual	US native	

Table 6
The reason clause recommendation methods' MRR and PerCorrect scores.

	MRR	PerCo(1)	PerCo(2)	PerCo(3)	PerCo(5)	PerCo(10)
Seq2Seq ranking	34.4%	19.3%	31.8%	39.8%	49.9%	66.5%
Our method	52.1%	36.3%	49.5%	58.6%	73.4%	89.3%

Table 7
Ablation tests for features on tweet/reason clause recommendation.

Feature	Performance change on MRR	
	Tweet clause	Reason clause
Political-relevancy (from ABCNN)	−2.0%	−0.8%
Wiki-page-edit-popularity	−1.3%	−4.6%
News-corpus-popularity	−1.1%	−1.5%
Verb-frequency	−0.7%	−0.4%
Sentence-importance	/	−3.3%
Causality	/	−0.5%

7.4. Reason clause recommendation results

For each of the 507 tweet clauses, we generate its reason clauses from the tweet's matched news article (Section 3.3.2) and rank these candidate reason clauses based on predefined features (Section 3.3.3). The ground truth, as stated in Section 5.4.1, provides correct clauses judged by humans and we compute MRR and PerCorrect(k) to measure whether these clauses are ranked high.

Our recommendation task is similar to query-based question-answering, where candidate answers are ranked according to their relevancy to the query. As a comparison, taking each tweet clause as a query, the generated reason clauses are ranked by a state-of-the-art question-answering model (Nogueira, Jiang, Pradeep, & Lin, 2020), which applies a sequence-to-sequence (Seq2Seq) transformation to compute their semantic relatedness as the relevancy score.

To generate the ground truth, we recruit nine annotators with qualified education backgrounds (Table 5) from *Upwork*, and each tweet will be annotated by five annotators. Following the guidelines in Section 5.4.1, **top-voted-reasons** are constructed based on the fact that each annotator can mark multiple reason clauses as correct for each tweet and the clause with the most votes is selected. The **top-voted-reasons** shows considerable inner-consensus, as Average Probability of choosing the Majority Class (APMC) scores is 92.0%, comparing with a random selection baseline of merely 4.0%.

Evaluation of recommendation: For our method and Seq2Seq model, we calculate their MRR and PerCorrect (k) scores. Table 6 shows that our model outperforms the Seq2Seq model, with regard to all metrics. Specifically, our method achieves MRR of 52.1%, compared with 34.4% for the Seq2Seq model. This means our method finds the correct causal background within top 2 result on average. Besides, the PerCorrect(5) score suggests that for 73.4% of the tweets, our method finds the correct causal backgrounds within top 5 results.

Evaluation of feature effectiveness: We evaluate the impact of all scoring features on system performance by ablation tests, with results shown in Table 7. All features appear helpful, with **Wiki-page-edit-popularity**, **news-corpus-popularity**, and **sentence-importance** corresponding to the largest performance changes. Though the **causality** feature's impact seems modest under this measurement, the cause-words appear much more in correct reason clauses than in incorrect ones, given that their paired effect-words are in the corresponding tweet clauses. 67.2% of the correct clauses have at least one cause-word, comparing with 46.3% for the incorrect ones. This difference in percentage is significant according to a two-sample T-test (p-value = 2.02e−24). It is possible that other features may have also captured some causality.

Table 8
Test accuracy and area under the curve (AUC) of different models. All experiments are repeated 5 times with different random seeds.

Method	BERT + TextCNN	BERT + TextRCNN	BERT + LSTM	Our Model
Accuracy	$87.63\%_{\pm 1.74}$	$86.92\%_{\pm 1.34}$	$88.27\%_{\pm 0.87}$	$\mathbf{94.48\%}_{\pm 0.88}$
AUC	$87.59\%_{\pm 1.7}$	$86.91\%_{\pm 1.34}$	$88.24\%_{\pm 0.88}$	$\mathbf{94.51\%}_{\pm 0.88}$

Table 9
Cases incorrectly predicted by ABCNN.

False negative cases		
Inflation (E)	Vladimir Vladimirovich Putin	Premier
The World Bank (E)	The United Nations Educational, Scientific and Cultural Organization	The Intermediate-Range Nuclear Forces Treaty
Gross domestic product (E)	China	Central Empires
Gross national income (E)	Afghanistan	Navy
Economics (E)	Speaker	The machinery of government
False positive cases		
Courage	University	Loan

7.5. Political relevancy scoring results

As a component of the clause scoring modules (Sections 3.2 and 3.3), the attention-based CNN model proposed in Section 3.4 tries to score a term's political relevancy. To evaluate our model, we convert the task into a text classification problem and compare with popular deep neural network text classification models, including TextCNN (Kalchbrenner, Grefenstette, & Blunsom, 2014), TextRCNN (Lai, Xu, Liu, & Zhao, 2015), and LSTM (Hochreiter & Schmidhuber, 1997). We choose the language model BERT (Devlin, Chang, Lee, & Toutanova, 2019) pre-trained on a large corpus by Google to generate text embeddings for these baseline models.

For the methods in comparison, their input unit for a term is a single piece of text consisting of the term's Wiki page summary and the titles of that page's backlink pages. All models, including ours, are trained, validated, and tested on the same dataset with same split (Section 4.3). We train all models with 5 epochs, using early stopping and weight decay strategies. We adjust the input dimension of the second block for our model (d_1) to 50 based on the validation set performance. As shown in Table 8, our model achieves an accuracy of 94.48% on the test set, outperforming the second best method by 6.21%. This also suggests that besides the enhancement in this relevancy scoring scenario, crowd-sourced linkage data can be similarly integrated to improve other text classification methods.

We examine these cases in which our model fails. For one of the five trained models, we see 18 such cases — 15 false negatives (politically relevant, but predicted to be not relevant) and 3 false positives (not politically relevant, but predicted to be relevant) out of 313 predictions (94.25% accuracy) and we list them in Table 9. One third of the false negatives seem to be also related to economic concepts such as 'Gross domestic product' and 'Gross national income' and we mark them with (E) in the table. The reason might be that human annotators are more likely to consider these as politically relevant while the training data from Wikipedia may weigh them less. Among these 18 cases, only 3 ('Gross domestic product', 'Afghanistan', and 'Courage') overlap with the second-best BERT-LSTM model. The BERT-LSTM model, on the other hand, has a much lower overall accuracy of 88.27%, with 34 failed cases in total (19 FNs and 15 FPs). However, similar to our model, 4 of its false negatives are economic concepts and the ratio is comparable (31.58% VS 33.33%). This consistent pattern may also root from the Wikipedia data on which the models are trained.

8. Case study

We perform a case study on one of Trump's tweets, demonstrating how data flows through the framework and how the results are evaluated. Fig. 6 shows the tweet's content, intermediate results, and the output reason clauses, with human annotations and interpretations from political news.

On Oct. 1, 2019, Trump posts a tweet, complaining that the federal funds rate set by the Federal Reserve is '*too high*' and affects the manufacturers negatively. For the general public who do not follow political and economic events closely, they may be confused about the causal backgrounds behind the tweet statements, given that the tweet lacks necessary background information and contains emotional and informal expressions such as '*Pathetic!*'.

To help the public understand the tweet, our framework first identifies interpret-worth parts in the tweet, since contents such as *'They are their own worst enemies, they don't have a clue.'* do not convey too much information. To this end, our framework decomposes the tweet into ranked clause(s), which is/are concatenated *meaningful proposition(s)* to imitate human-generated queries. As shown in Fig. 6, only one clause is formed in this case – *'Jay Powell and the Federal Reserve have allowed the Dollar to get so strong'*. It is the same as what human annotators select as the interpret-worth clause.

Our framework finds the causal backgrounds behind the interpret-worth tweet clause from news articles published within 15 days before the tweet is posted. To do this, it first seeks news articles with high textual similarity to the target tweet. The top 3

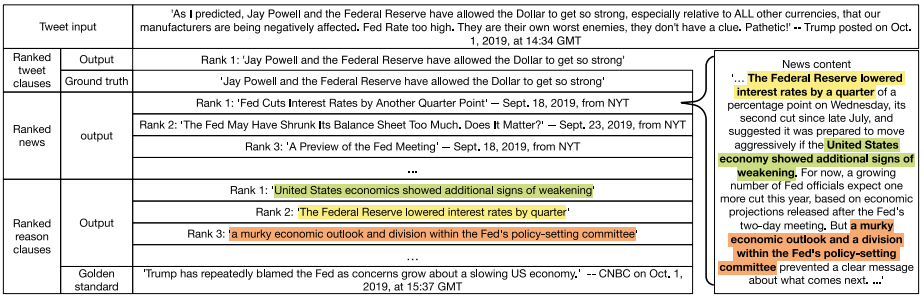

Fig. 6. A case study of the tweet interpretation framework.

news articles, as listed in Fig. 6, are all relevant to the *'Fed'* (Federal Reserve). The top one article (*'Fed Cuts Interest Rates by Another Quarter Point'*, Sept. 18, 2019, from NYT) is chosen by the annotators to be a 'match', since it discusses 'interest rates' which is a core component of the tweet.

From this matched news article, the framework tries to further locate the reason clauses. The causal relationship between the tweet and the news content is often implicit, since the framework only looks at existing knowledge (previous news in this step) for clues, instead of waiting for political analysis on the tweet that comes hours later. The framework follows the mechanisms described in Sections 3.2.1 and 3.3.3 to recommend top reason clauses, taking the implicit causality into consideration. The top reason clause among 9 generated candidates is *'United States economy showed additional signs of weakening'*. To verify its correctness, we use the tweet content as a query on Google News and find a post-tweet analysis article published one hour after the tweet – *'Trump has repeatedly blamed the Fed as concerns grow about a slowing US economy', CNBC*, which is consistent with the framework's top pick. Besides, the crowdsourced annotators also vote this reason clause as the top one. It deciphers Trump's accusatory statement, as he believes that the US economy is slowing down due to the Federal Reserve's high federal funds rate.

9. Interactivity in our design

In pipeline frameworks, errors from each step often multiply, resulting in unacceptable outcomes. In our scenario, step 1 and 2 introduce errors. However, we design our framework to be interactive — as can be seen in Section 8 and Fig. 2, the users (or annotators) will pick their contents of interest in step 1 and related news article in step 2 from the ranked recommendations. By doing this, we avoid the errors from multiplying and make an effort in improving user experience.

10. Conclusions and future work

We present a novel framework to discover the causal backgrounds behind politicians' tweets, which addresses the challenges in open information extraction, term weighting, and causality inference. The framework is a first attempt to infer specific causal backgrounds behind interpret-worth clauses in tweets, which captures contents of interest to humans and generates valid corresponding causal backgrounds from news. Its core code and the entire data including the annotated tweets which take an expert 1000 hours to label, are made public.

Our framework is in the broader domains of intent-discovery and causality-detection, with its own design considerations that may supplement relevant research. Regular intent-discovery studies try to find the intent within an utterance by classification or extraction based on the context itself which can be quite limited. We propose to seek intents from external knowledge. This may be useful when the contexts do not directly contain the intents but provide clues for finding them elsewhere. Similarly, it can supplement causality-detection within a single context by finding implicit cause-and-effect across different contexts. Applications like advertisement recommendation, behavior prediction, and dialogue generation can be potentially improved with specific intents discovered.

For future work, we plan to explore two directions. One is to find the connections between the tweet motivations and their relevant real-world consequences. For instance, we are curious to quantify how stock markets react to politicians' tweets. The other direction aims at broadening the generality of our approach. Currently, we focus on political figures' tweets with customization in various steps and we would like to apply the framework to other influential people and other types of information sources, such as TV interviews and public speeches.

CRediT authorship contribution statement

Ziyue Li: Methodology, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Hang Hu:** Data curation, Writing – original draft, Writing – review & editing. **He Wang:** Data curation, Writing – original draft. **Luwei Cai:** Data curation. **Haipeng Zhang:** Conceptualization, Supervision, Project administration, Writing – review & editing, Funding acquisition. **Kunpeng Zhang:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This project is sponsored by ShanghaiTech University start-up fund, Shanghai Sail Program (No. 19YF1433800), and the Key Projects of Shanghai Soft Science Research Program (No. 20692194300).

References

Abbar, S., Castillo, C., & Sanfilippo, A. (2018). To post or not to post: Using online trends to predict popularity of offline content. In *HT '18*, *Proceedings of the 29th on hypertext and social media* (pp. 215–219). New York, NY, USA: ACM, http://dx.doi.org/10.1145/3209542.3209575.

Al Tamime, R., Giordano, R., & Hall, W. (2018). Observing burstiness in wikipedia articles during new disease outbreaks. In *WebSci '18*, *Proceedings of the 10th ACM conference on web science* (pp. 117–126). New York, NY, USA: ACM, http://dx.doi.org/10.1145/3201064.3201080.

Al-Zaidy, R. A., & Giles, C. L. (2018). Extracting semantic relations for scholarly knowledge base construction. In *ICSC '18*, *2018 IEEE 12th international conference on semantic computing* (pp. 56–63). New York, NY, USA: IEEE.

An, J., Cha, M., Gummadi, K., Crowcroft, J., & Quercia, D. (2012). Visualizing media bias through Twitter. In *ICWSM '12*, *Proceedings of the 6th international AAAI conference on web and social media*. Palo Alto, California, USA: AAAI Press, URL https://www.aaai.org/ocs/index.php/ICWSM/ICWSM12/paper/view/4775.

Angeli, G., Johnson Premkumar, M. J., & Manning, C. D. (2015). Leveraging linguistic structure for open domain information extraction. In *ACL '15*, *Proceedings of the 53th annual meeting of the association for computational linguistics* (pp. 344–354). Baltimore, Maryland, USA: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/P15-1034.

Athreya, R. G., Bansal, S. K., Ngomo, A.-C. N., & Usbeck, R. (2021). Template-based question answering using recursive neural networks. In *ICSC '21*, *IEEE 15th international conference on semantic computing* (pp. 195–198). IEEE, http://dx.doi.org/10.1109/ICSC50631.2021.00041.

Chali, Y., & Golestanirad, S. (2016). Ranking automatically generated questions using common human queries. In *INLG '16*, *Proceedings of the 9th international natural language generation conference* (pp. 217–221). Baltimore, Maryland, USA: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/W16-6635.

Chen, E., Deb, A., & Ferrara, E. (2021). # Election2020: the first public Twitter dataset on the 2020 US presidential election. *Journal of Computational Social Science*, 1–18. http://dx.doi.org/10.1007/s42001-021-00117-9.

Christensen, J., Mausam, Soderland, S., & Etzioni, O. (2011). An analysis of open information extraction based on semantic role labeling. In *K-CAP '11*, *Proceedings of the 6th international conference on knowledge capture* (pp. 113–120). New York, NY, USA: ACM, http://dx.doi.org/10.1145/1999676.1999697.

Corney, D., Albakour, D., Martinez, M., & Moussa, S. (2016). What do a million news articles look like? In *ECIR '16*, *Proceedings of the first international workshop on recent trends in news information retrieval co-located with 38th European conference on information retrieval* (pp. 42–47). CEUR-WS.

Debole, F., & Sebastiani, F. (2004). *Text mining and its applications* (pp. 81–97). Berlin, Heidelberg: Springer.

Del Corro, L., & Gemulla, R. (2013). Clausie: clause-based open information extraction. In *WWW '13*, *Proceedings of the 22th international conference on world wide web* (pp. 355–366). New York, NY, USA: ACM, http://dx.doi.org/10.1145/2488388.2488420.

Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., et al. (2015). Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 32–49. http://dx.doi.org/10.1016/j.ipm.2014.10.006.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL '19*, *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1423.

Domeniconi, G., Moro, G., Pasolini, R., & Sartori, C. (2015). A comparison of term weighting schemes for text classification and sentiment analysis with a supervised variant of tf. idf. In *Proceedings of the 4th international conference on data management technologies and applications* (pp. 39–58). Berlin, Heidelberg: Springer, http://dx.doi.org/10.1007/978-3-319-30162-4_4.

Fader, A., Soderland, S., & Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 1535–1545). Edinburgh, Scotland, UK.: Association for Computational Linguistics, URL https://www.aclweb.org/anthology/D11-1142.

Gao, W., Peng, M., Wang, H., Zhang, Y., Xie, Q., & Tian, G. (2019). Incorporating word embeddings into topic modeling of short text. In *Knowledge and information systems* (pp. 1123–1145). Berlin, Heidelberg: Springer, http://dx.doi.org/10.1007/s10115-018-1314-7.

Grzeça, M., Becker, K., & Galante, R. (2020). Drink2Vec: Improving the classification of alcohol-related tweets using distributional semantics and external contextual enrichment. *Information Processing & Management*, Article 102369. http://dx.doi.org/10.1016/j.ipm.2020.102369.

Gupta, P., Pagliardini, M., & Jaggi, M. (2019). Better word embeddings by disentangling contextual n-gram information. In *NAACL '19*, *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics* (pp. 933–939). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1098.

Hansen, C., Hansen, C., Alstrup, S., Grue Simonsen, J., & Lioma, C. (2019). Neural check-worthiness ranking with weak supervision: Finding sentences for fact-checking. In *WWW '19*, *Companion proceedings of the 2019 world wide web conference* (pp. 994–1000). New York, NY, USA: ACM, http://dx.doi.org/10.1145/3308560.3316736.

Hashimoto, C., Torisawa, K., Kloetzer, J., Sano, M., Varga, I., Oh, J.-H., et al. (2014). Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and association features. In *ACL '14*, *Proceedings of the 52th annual meeting of the association for computational linguistics* (pp. 987–997). Baltimore, Maryland, USA: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/P14-1093.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 1735–1780.

Ittoo, A., & Bouma, G. (2011). Extracting explicit and implicit causal relations from sparse, domain-specific texts. In *NLDB '11*, *International conference on application of natural language to information systems* (pp. 52–63). Berlin, Heidelberg: Springer, http://dx.doi.org/10.1007/978-3-642-22327-3_6.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *ACL '14*, *Proceedings of the 52th annual meeting of the association for computational linguistics* (pp. 655–665). Baltimore, Maryland, USA: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/P14-1062, URL https://www.aclweb.org/anthology/P14-1062.

Kanhabua, N., & Nejdl, W. (2013). Understanding the diversity of tweets in the time of outbreaks. In *WWW '13*, *Proceedings of the 22th international conference on world wide web* (pp. 1335–1342). New York, NY, USA: ACM, http://dx.doi.org/10.1145/2487788.2488172.

Kayesh, H., Islam, M. S., & Wang, J. (2019). Event causality detection in tweets by context word extension and neural networks. In *PDCAT '19*, *2019 20th International conference on parallel and distributed computing, applications and technologies* (pp. 352–357). New York, NY, USA: IEEE, http://dx.doi.org/10.1109/PDCAT46702.2019.00070.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *EMNLP '14*, *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1746–1751). Baltimore, Maryland, USA: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/D14-1181.

Konstantinovskiy, L., Price, O., Babakar, M., & Zubiaga, A. (2021). Toward automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. In *Digital threats: Research and practice* (p. 16). New York, NY, USA: ACM, http://dx.doi.org/10.1145/3412869.

Kozlowski, D., Lannelongue, E., Saudemont, F., Benamara, F., Mari, A., Moriceau, V., et al. (2020). A three-level classification of french tweets in ecological crises. *Information Processing & Management*, Article 102284. http://dx.doi.org/10.1016/j.ipm.2020.102284.

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent convolutional neural networks for text classification. In *AAAI '15*, *Proceedings of the 29th AAAI conference on artificial intelligence*. Palo Alto, California, USA: AAAI Press, URL https://dl.acm.org/doi/10.5555/2886521.2886636.

Li, H., Liu, T., Ma, W.-Y., Sakai, T., Wong, K.-F., & Zhou, G. (2008). *Information retrieval technology*. Berlin, Heidelberg: Springer.

Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., et al. (2012). Twiner: Named entity recognition in targeted twitter stream. In *SIGIR '12*, *Proceedings of the 35th international ACM special interest group on information retrieval conference* (pp. 721–730). New York, NY, USA: ACM, http://dx.doi.org/10.1145/2348283.2348380.

Liu, X., Fu, J., & Chen, Y. (2020). Event evolution model for cybersecurity event mining in tweet streams. *Information Sciences*, 254–276. http://dx.doi.org/10.1016/j.ins.2020.03.048.

d. manning, C., raghavan, p., & schütze, h. (2008). *Introduction to information retrieval* (pp. 192–195). Cambridge, England: Cambridge University Press.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *EMNLP '04*, *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 404–411). Baltimore, Maryland, USA: Association for Computational Linguistics, URL https://www.aclweb.org/anthology/W04-3252.

Mohammad, S. M., Zhu, X., Kiritchenko, S., & Martin, J. (2015). Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing & Management*, 480–499. http://dx.doi.org/10.1016/j.ipm.2014.09.003.

Nie, L., Davison, B. D., & Qi, X. (2006). Topical link analysis for web search. In *SIGIR '06*, *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 91–98). New York, NY, USA: ACM, http://dx.doi.org/10.1145/1148170.1148189.

Nogueira, R., & Cho, K. (2019). Passage re-ranking with BERT. arXiv preprint arXiv:1901.04085.

Nogueira, R., Jiang, Z., Pradeep, R., & Lin, J. (2020). Document ranking with a pretrained sequence-to-sequence model. In *EMNLP '20*, *Proceedings of the 2020 conference on empirical methods in natural language processing* (pp. 708–718). Baltimore, Maryland, USA: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.findings-emnlp.63.

Oh, J.-H., Torisawa, K., Kruengkrai, C., Iida, R., & Kloetzer, J. (2017). Multi-column convolutional neural networks with causality-attention for why-question answering. In *WSDM '17*, *Proceedings of the 10th ACM international conference on web search and data mining* (pp. 415–424). New York, NY, USA: ACM, http://dx.doi.org/10.1145/3018661.3018737.

Papakyriakopoulos, O., Shahrezaye, M., Serrano, J. C. M., & Hegelich, S. (2019). Distorting political communication: The effect of hyperactive users in online social networks. In *INFOCOM '19*, *IEEE INFOCOM 2019 - IEEE conference on computer communications workshops* (pp. 157–164). IEEE, http://dx.doi.org/10.14778/3329772.3329778.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *EMNLP '14*, *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1532–1543). Baltimore, Maryland, USA: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/D14-1162.

Peters, J., & Bühlmann, P. (2015). Structural intervention distance for evaluating causal graphs. *Neural computation*, 771–799.

Ramos, J., et al. (2003). Using tf-idf to determine word relevance in document queries. In *ICML '03*, *Proceedings of the first instructional conference on machine learning* (pp. 133–142).

Raza, A. A., Habib, A., Ashraf, J., & Javed, M. (2019). Semantic orientation based decision making framework for big data analysis of sporadic news events. *Journal of Grid Computing*, 367–383. http://dx.doi.org/10.1007/s10723-018-9466-y.

Reed, J. W., Jiao, Y., Potok, T. E., Klump, B. A., Elmore, M. T., & Hurson, A. R. (2006). TF-ICF: A new term weighting scheme for clustering dynamic data streams. In *ICMLA '06*, *Proceedings of the 5th international conference on machine learning and applications* (pp. 258–263). Los Alamitos, CA, USA: IEEE Computer Society, http://dx.doi.org/10.1109/ICMLA.2006.50.

Riaz, M., & Girju, R. (2013). Toward a better understanding of causality between verbal events: Extraction and analysis of the causal power of verb-verb associations. In *SIGDIAL '13*, *Annual meeting of the special interest group on discourse and dialogue* (pp. 21–30). Baltimore, Maryland, USA: Association for Computational Linguistics, URL https://www.aclweb.org/anthology/W13-4004.

Ritter, A., Clark, S., Etzioni, O., et al. (2011). Named entity recognition in tweets: an experimental study. In *EMNLP '11*, *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 1524–1534). Baltimore, Maryland, USA: Association for Computational Linguistics, http://dx.doi.org/10.5555/2145432.2145595.

Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. In *Foundations and trends in information retrieval* (pp. 333–389). Hanover, MA, USA: Now Publishers Inc., http://dx.doi.org/10.1561/1500000019.

Shah, A. A., Ravana, S. D., Hamid, S., & Ismail, M. A. (2019). Accuracy evaluation of methods and techniques in web-based question answering systems: a survey. *Knowledge and Information Systems*, 611–650. http://dx.doi.org/10.1007/s10115-018-1203-0.

Soergel, D. (1998). *Language, speech, and communication, WordNet: An electronic lexical database*. Cambridge, MA: MIT Press.

Stevenson, A. (2010). *Oxford dictionary of English*. USA: Oxford University Press.

Tang, X., Chen, L., Cui, J., & Wei, B. (2019). Knowledge representation learning with entity descriptions, hierarchical types, and textual relations. *Information Processing & Management*, 809–822. http://dx.doi.org/10.1016/j.ipm.2019.01.005.

Tsagkias, M., de Rijke, M., & Weerkamp, W. (2011). Linking online news and social media. In *WSDM '11*, *Proceedings of the 4th ACM international conference on web search and data mining* (pp. 565–574). New York, NY, USA: ACM, http://dx.doi.org/10.1145/1935826.1935906.

van Vliet, L., Törnberg, P., & Uitermark, J. (2020). The Twitter parliamentarian database: Analyzing Twitter politics across 26 countries. *Plos One*, Article e0237073. http://dx.doi.org/10.1371/journal.pone.0237073.

Wallace, B. C., Choe, D. K., Kertz, L., & Charniak, E. (2014). Humans require context to infer ironic intent (so computers probably do, too). In *ACL' 2014*, *Proceedings of the 52th annual meeting of the association for computational linguistics* (pp. 512–516). Baltimore, Maryland, USA: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/P14-2084.

Wang, Z., & Chan, L. (2011). Using bayesian network learning algorithm to discover causal relations in multivariate time series. In *ICDM '11*, *Proceedings of the 11th IEEE international conference on data mining* (pp. 814–823). New York, NY, USA: IEEE, http://dx.doi.org/10.1109/ICDM.2011.153.

Wang, X., Yu, C., Baumgartner, S., & Korn, F. (2018). Relevant document discovery for fact-checking articles. In *WWW '18*, *Companion proceedings of the 2018 world wide web conference* (pp. 525–533). New York, NY, USA: ACM, http://dx.doi.org/10.1145/3184558.3188723.

White, A. S., Reisinger, D., Sakaguchi, K., Vieira, T., Zhang, S., Rudinger, R., et al. (2016). Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 1713–1723). Baltimore, Maryland, USA: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D16-1177.

Wu, Z., Zhu, H., Li, G., Cui, Z., Huang, H., Li, J., et al. (2017). An efficient wikipedia semantic matching approach to text document classification. *Information sciences*, 15–28. http://dx.doi.org/10.1016/j.ins.2017.02.009.

Yahya, M., Whang, S., Gupta, R., & Halevy, A. (2014). Renoun: Fact extraction for nominal attributes. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 325–335). Baltimore, Maryland, USA: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/D14-1038.

Yates, A., Banko, M., Broadhead, M., Cafarella, M., Etzioni, O., & Soderland, S. (2007). TextRunner: Open information extraction on the web. In *NAACL-HLT '07, Proceedings of the annual conference of the north american chapter of the association for computational linguistics* (pp. 25–26). Rochester, New York, USA: Association for Computational Linguistics, URL https://www.aclweb.org/anthology/N07-4013.

Yin, W., Schütze, H., Xiang, B., & Zhou, B. (2016). ABCNN: Attention-based convolutional neural network for modeling sentence pairs. In *Transactions of the association for computational linguistics* (pp. 259–272). Baltimore, Maryland, USA: Association for Computational Linguistics, URL https://www.aclweb.org/anthology/Q16-1019.pdf.

Zhang, J., Sun, M., Feng, Y., & Li, P. (2020). Learning interpretable relationships between entities, relations and concepts via bayesian structure learning on open domain facts. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8045–8056). Baltimore, Maryland, USA: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.717.

Zhao, K., Ji, D., He, F., Liu, Y., & Ren, Y. (2021). Document-level event causality identification via graph inference mechanism. *Information sciences*, 115–129. http://dx.doi.org/10.1016/j.ins.2021.01.078.

Zhao, Z., Resnick, P., & Mei, Q. (2015). Enquiring minds: Early detection of rumors in social media from enquiry posts. In *WWW '15, Proceedings of the 24th international conference on world wide web* (pp. 1395–1405). New York, NY, USA: ACM, http://dx.doi.org/10.1145/2736277.2741637.

Zhou, D., Chen, L., & He, Y. (2015). An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In *AAAI '15, Proceedings of the 29th AAAI conference on artificial intelligence*. Palo Alto, California, USA: AAAI Press, URL https://dl.acm.org/doi/10.5555/2886521.2886664.

Zou, Y., Lin, J., Zhao, L., Kang, Y., Jiang, Z., Sun, C., et al. (2021). Unsupervised summarization for chat logs with topic-oriented ranking and context-aware auto-encoders. In *Proceedings of the international joint conference on artificial intelligence*. Palo Alto, California, USA: AAAI Press, URL https://arxiv.org/abs/2012.07300.