IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 October 2023, accepted 26 October 2023, date of publication 8 November 2023, date of current version 16 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3331219

RESEARCH ARTICLE

Real-Time Event Detection Using Self-Evolving Contextual Analysis (SECA) Approach

SAMI AL SULAIMANI AND ANDREW STARKEY
School of Engineering, University of Aberdeen, AB24 3UE Aberdeen, U.K.
Corresponding author: Sami Al Sulaimani (s.alsulaimani1.19@abdn.ac.uk)

This work was supported by the Ministry of Higher Education, Research and Innovation-Sultanate of Oman.

ABSTRACT Dynamically monitoring and analyzing evolving real-world events (riots, earthquakes, and football matches) using publicly available short texts (social media posts) is becoming increasingly important. This content can hold critical information about various events, which can help decision-makers to make better decisions. Significant research efforts have been made in this regard. However, most of these provide solutions based on black-box engines, in which technical capabilities are required to understand their internal mechanics. Also, they offer very little information about the detected events and generally tend to answer very high-level questions, such as: "what are the main topic clusters?" "what are the main words (e.g. top ten words) of these topics?". These challenges can limit their usage in some critical domains, where the need for transparency, and more information, to analyze a particular situation is crucial. Thus, to complement and fill the gap in the direction of existing studies, in which the effectiveness and success of the proposed approaches are insufficiently determined by their performance scores, this paper presents datasets that can be used for dynamic topic detection of different frequencies over time based on real Tweets and a new transparent method for the dynamic event detection problem called Self-Evolving Contextual Analysis (SECA). It helps to answer, for any given time frame, other fundamental questions, such as: "what are the sub-topics of, and their relationship to, a topic (or a sub-topic?)", "what are the changing topics and sub-topics?", "what are the new trends?", "what are the topics no longer being discussed?" and most importantly, "why and how have these topics and changes been identified and generated?". Moreover, Performance and Carbon Footprint assessments reveal the comparative effectiveness of the proposed approach. In addition, this paper presents a practical implementation of SECA to dynamically analyze tweets collected during the FIFA World Cup 2022 Final Match.

INDEX TERMS Text analysis, event detection, contextual analysis, unsupervised machine learning, short text clustering, explainable AI, green AI.

I. INTRODUCTION

Social networking platforms such as Twitter, Facebook, and WhatsApp provide easy-to-access tools that allow people to post various topics in real-time, on varied topics such as sporting events, related to an election campaign or a natural disaster, for example. Examples already exist of where social media platforms have contributed significantly to important current events, such as facilitating communication during "Arab Spring" [1]. Capturing this valuable information as soon as it is published may help protect people's rights, deliver their voices, or, in more grave circumstances, directly support activities that can save their lives. Triggered by these needs, researchers have provided many techniques and insights for dynamically monitoring and analyzing events in real-time.

Although some service providers, such as Twitter, provide some accessibility to users' posts (tweets) for analysis, detecting events using them is a challenging task. Textual data can be very short in length and are often written in an informal language, which contains misspelled, abbreviated, or slang words, grammatically incorrect sentences, and mixed languages [2], [3]. As a result, posts may have very few lexically similar terms; therefore, identifying the relationships between

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Kashif Bashir.

similar texts can be problematic. Other challenges are related to the dynamic nature of these environments. Depending on context and time, the similarities between two keywords may change [4]. Scalability and efficiency are two other important factors that should be considered when capturing events using social media: posts arrive at high speeds and in large volumes.

Extensive studies have been conducted to overcome these issues and improve the performance of event detection tasks. However, in our recent work [5] (in addition to performance) we presented other important qualities such as Transparency and Carbon footprint, which have not received much attention in previous studies. These issues are important to consider with Explainable AI being an emerging trend in AI, and the Carbon footprint of AI also being a growing concern. Our findings showed that an approach called Contextual Analysis [6], offers some potential capabilities for the task and can satisfy the checklist of Performance, Carbon Footprint, and Transparency.

This paper focuses on complementing this work and answering its open research question: "how the Contextual Analysis algorithm can be utilized or modified (by changing its basic machinery and without harming its level of transparency), in order to provide information about the evolution of topics over time?". Based on the Contextual Analysis method, we present a new approach called Self-Evolving Contextual Analysis (SECA), and a lighter version called SECA-Light, which can help dynamically detect events. It generates a tree of word relationships that can evolve by modifying its structure for any incoming new batch when there is a need to do so.

The remainder of this paper is organized as follows. First, a summary of related work is given in Section II. Subsequently, Section III presents different definitions and highlights the main challenges of the event detection task. Section IV, concisely discusses the need for environmentally friendly and transparent algorithms. This is followed by a brief description of the Contextual Analysis method in Section V. Section VI explores the challenges of CA in the context of the dynamic nature of the event detection task. Section VII presents the Self-Evolving Contextual Analysis approach. The experiments are discussed in Section VIII. Section IX provides a demonstration of the proposed approach applied to a real-world scenario. The conclusions drawn from this study and future directions are discussed in Section X.

II. RELATED WORK

Researchers have paid considerable attention to developing algorithms and building automated solutions that can help detect events on social media. An example of the former is the effort presented in [7], in which Locality Sensitive Hashing (LSH) algorithm was employed to detect new events from Twitter. This work tends to focus more on overcoming the scalability limitations of event detection approaches that depend on comparing any new tweet with all previously seen posts. A query tweet is compared with other tweets in the same bucket. In addition, to overcome the limitations of LSH in dealing with massive tweet streams, the number of tweets per bucket and the number of comparisons required were suggested to be constant. However, the proposed system lacks sufficient detail regarding the detected events, as it produces the fastest-growing threads of tweets (that highlight significant events) represented by the top tweets. Moreover, the mathematical background required to understand the mechanism of LSH to approximate the similarity between two texts makes it difficult for non-technical users to comprehend.

The work presented in [8] builds a keyword graph based on co-occurrence to detect events. The underlying assumption is that "keywords co-occur when there is some meaningful topical relationship between them". Two keywords are linked as nodes if they are found in the same document. The proposed algorithm consists of three main parts: KeyGraph construction (mainly to create a graph with the extracted keywords), community detection (to identify the densely related terms to form a community that describes and acts as a key or proxy document of potential events), and document clustering (to map the documents to event clusters using their key documents, in which each group is considered as an event). Note: An extension of this study can be found in [9]. Although this type of work can provide a summarized and networked representation of words that highlight an event, they lack to offer a meaningful story about the word relationship for each event. This is because of the flat representation of the constructed graph.

The authors in [10] proposed an event detection approach, called EvenTweet, to detect and track localized events ("events that are important within a small geographic area") from a stream of tweets. It focuses on grouping terms that are close in spatial distance in order to describe an event. The system produces the top terms, estimated start times, and estimated locations of the generated events, which are sorted according to a scoring scheme. Yet, location-based event detection approaches that directly depend on geotagged information can suffer from low performance, due to the fact that these details are not usually available in publicly available resources. For instance, on Twitter, the proportion of posts containing geo-tag information is between 1% and 3% [11]. Thus, events that are declared only in non-geotagged posts will be undetected.

Another interesting example is the work described in [12] that provides a framework to leverage Twitter data to detect events in real-time. Based on the assumption that an event can be described by identifying its semantic descriptors ("who", "where", "what" and "when"), semantic classes were proposed to represent the tweets' terms, such as proper nouns, mentions, hashtags, verbs, common nouns, temporal expressions, and nouns. To generate event clusters, these classes, with a previously calculated weighted figure representing their importance, are then used to measure the similarity between every tweet and the already generated clusters, which are represented by their centroids. One

major limitation of this work is the high dependency on the accuracy of external resources such as TweetNLP for capturing semantics, which can limit their usage in rapidly evolving environments or where slang or mixed languages are used [12]. Moreover, word embeddings were used to solve the problem of grouping similar tweets that describe similar events, but comprised of synonyms. However, this type of technique can be biased toward the corpus that it was trained on and cannot deal with new terms that were not seen in the training dataset. This implies that tweets about similar events could be grouped into different clusters, or be missed entirely. The authors claimed that the proposed cluster merging phase could help handle these issues; however, this was not clearly explained in the original work.

The study in [4] presented an incremental clustering vector expansion technique to detect events from microblogging posts without depending on external resources. The detection process is mainly accomplished in two main phases: a clustering phase and burst detection. The vector expansion process is an important step in this approach to identify similar terms, in a temporal context, that represent a post or a cluster's centroid. This is followed by a simple incremental clustering method in which a new post is incrementally compared and added to an existing or new cluster. To detect "news worthy" events, the system identifies bursty terms in the generated clusters. Yet, each event is only represented by a few details, such as its creation time, centroid, tweets, and the "best tweet", more information is required in order to gain more knowledge about the produced outputs.

The authors in [13] proposed a framework called TwitterNews+, based on an incremental clustering algorithm to detect events in real-time for Twitter streams. The system consists of two main modules, a Search Module, and an EventCluster Module, which highly depend on indices (such as term-tweets and term-events) during their process. The Search Module maintains a term-tweets inverted index in order to fetch, from the corpus, the most similar tweets to a given query post, in order to measure its novelty and to decide whether it should be transferred to the clustering phase. Based on this decision, the EventCluster utilizes a term-events inverted index to retrieve candidate clusters, and subsequently computes the similarity between the query tweet and the centroid of every candidate cluster. It is stated that this approach can process 1336 tweets per second with high precision and recall. However, an extensive parameter analysis was conducted for the eight parameters, that are required by the algorithm, to improve its performance. Moreover, this study did not directly consider word similarities (neither synonyms nor contextual) for grouping similar tweets into the same event cluster.

Recently, the author in [14] proposed an advanced topic modeling approach called BERTopic. This approach, which is based on text embedding techniques, i.e., BERT-based, was designed to address the limitations of other prominent topic modeling methods, such as LDA and NMF, which do not consider the context of words. However, in our recent work [5], we found that BERTopic failed to meet the Transparency assessment criteria and proved to be the most carbon-intensive approach among the methods studied, even without taking into account the energy required for training the BERT model itself.

Similar to the interests of the presented studies, this paper focuses on dynamically detecting events using short texts. On the other hand, to complement them, more attention is given to incorporating two more dimensions in order to indicate the level of success of any approach intended to solve the problem, i.e., Transparency and Carbon Footprint.

III. EVENT DETECTION

A. DEFINITIONS

The term event can generally be defined as something that occurs. In Collins' dictionary, it is defined as "something that happens, especially when it is unusual or important. You can use events to describe all the things that are happening in a particular situation" [15]. According to the Oxford dictionary, an event is "Something that happens or takes place, esp. something significant or noteworthy; an incident, an occurrence" [16].

Researchers have proposed various descriptions of this term in the context of social media. Posts, time, location, topic, and people are common entities in these definitions. For example, while in [17], an event is described as "as a real-world occurrence e with (1) an associated time period T_e and (2) a time-ordered stream of Twitter messages M_e, of substantial volume, discussing the occurrence and published during time T_e", the authors in [18] derived their definition as "an occurrence causing a change in the volume of text data that discusses the associated topic at a specific time. This occurrence is characterized by topic and time, and often associated with entities such as people and location". More recently, [19] define an event as "a way of referring to an observable activity at a certain time and place that involves or affects a group of people in a social network."

Real-time event detection in social media focuses on monitoring and analyzing microblogging content as a way to detect real-world events (civil unrest, disaster, presidential elections, etc.). With the scope on the textual content only, this paper adopts the definition presented in [19], which states, "with respect to social media content, event detection describes significant happenings in real-life by systematically analyzing the content published online and addresses how an event is emerging, gaining momentum, flows and evolves.".

To that end, this paper adopts a research methodology that meets this definition and explores the problem of event detection for defined events whose frequency changes over time. The datasets are based on real-world events, whose frequency is modified in order to reflect different rates of change of discussion for that event, in order to assess the ability of an event detection method to identify these changes. The datasets are defined more fully in later sections.

B. CHALLENGES OF EVENT DETECTION IN SOCIAL MEDIA

Acquiring data is the first module in any event detection solution. Thanks to Twitter, researchers can have some accessibility to users' posts (tweets) for analysis. However, detecting events using these sources is not a trivial task. Textual data can be very short in length and are often written in an informal language, which contains misspelled, abbreviated, or slang words, grammatically incorrect sentences, mixed languages [2], [3], and other issues. This section describes the most common challenges found in the literature.

1) SPARSITY OF CONTENT

One of the important challenges is the sparsity of content problems, which generally refers to a lack of information resulting from the limited occurrence of words appearing together within a single post. This makes identifying semantic relationships based on the texts a challenging task. Many factors can lead to this issue, one of which is the short text form on social media posts. Twitter, for example, limits the number of characters in a post to 280 [20], which was 140 characters before the year 2017. Regardless of this increase, it was found that brevity in posting with less than 140 characters is the common user behavior [21].

The use of informal language in posts is another factor that can lead to sparsity. Social media users tend to use abbreviated words or phrases extensively in their posts. For example, "ur" for "your", "LIV" for "Liverpool", "btw" for "by the way", and "OT" for "Over Time" and "PK" for "penalty kick" in the context of football matches. One might argue that the driver of this habit is the constraint on the number of characters on social media platforms [19].

Slang words and word lengthening (or sometimes called words stretching) are other usage habits found in social media. Words like "belter" (means amazing goal), "banter" (means jokes), "footy" (means football), "howler" (means mistake) and "Penaltyyyyyyyyyy" (stretched word for penalty), are some examples of the types of words that can be found in tweets referring to football matches.

2) HIGH VOLUME AND SPEED

Posts on social media platforms arrive at high speeds and large volumes. For example, on Twitter, the average number of posts per second is 6000 [22], [23]. Elon Musk, the owner and CEO of Twitter, tweeted that 24,400 posts per second were sent on Twitter for one of France's goals in the final match of the 2022 FIFA World Cup [24]. Storing and processing a large volume of data that arrives at a rapid rate can be a very difficult (or sometimes impossible) task. Since a complete dataset cannot be easily acquired for researchers as Twitter allows only 1% of the data to be fetched through their API, one should carefully consider the feasibility of any proposed real-time event detection solution in terms of resource implications and other related constraints, and the problem's requirements.

3) POLLUTED INFORMATION

The success of social media platforms has attracted others to spread misinformation in their contents, whether by an individual posting misleading information (e.g. spreading incorrect instructions during a disease outbreak [25]) or by an organized group (humans or bots) promoting a certain agenda (e.g. increasing the exposure of untrustworthy content during presidential campaigns [26]). One recent example is a widely circulated video clip (with 239 thousand views on 13/02/2023) about a tsunami that devastated an Indonesian island in 2018 that was falsely connected to the 2023 Turkey and Syria Earthquake on February 6 [27]. These activities, in turn, have raised concerns related to the quality of the tremendous amount of content disseminated through these channels.

Spam (defined as "unsolicited email or text messages" in Collin's Dictionary [28]) is another issue that gained the attention of the service providers and the research community. During three-quarters of 2022, Facebook removed approximately 3.9 billion spam content from its platform [29]. Spam can negatively impact the quality of popular topic discourse by creating confusion and misunderstandings [30]. Publishing advertisements for products or services is one form of unsolicited content commonly found in trending topics.

4) DYNAMIC CONTENT

The content of social media platforms is dynamic in nature and not static. Microbloggers may use different sets of words to describe a specific event, and they may use new words (i.e. neither exist in the training dataset nor in informal or formal dictionaries) or use words in new contexts. In addition, depending on the context and timeframe, the similarity between two words can vary [4]. To give an example, words such as "Covidiot" (a person that ignores the warnings regarding public health or safety, according to Urban Dictionary) and "Covidient" (a person who takes government guidelines very seriously, according to Urban Dictionary) are new terms that emerged during the outbreak of Covid-19 [31]. Moreover, the word "bisht" (a traditional Arab cloak) newly appeared in the context of FIFA World Cup occasions, where the emir of Qatar put a "bisht" on Lionel Messi's (Argentine professional footballer) shoulder during the 2022 World Cup closing ceremony. Thus, capturing these changes and their new potential context can help reflect the word relationship in social media more dynamically to improve the accuracy of the event detection task.

5) MULTILINGUAL EVENTS

Another important challenge in the context of event detection on social media is related to dealing with more than one language. The majority of social media providers now support many other languages alongside English. According to a study conducted by [32], in which a sample of 118 billion tweets was collected throughout the period from 2009 to 2019, 173 languages were detected with high

dominance of English (24%), Japanese (12%), Spanish (6 %), Arabic (4%), and Portuguese (4%). Considering linguistic diversity is very important for analyzing content during international gatherings (like World Cups), global health crises (like Covid-19), or events that occur in multilingual regions (like Turkey and Syria earthquake in 2023).

IV. THE NEED FOR ENVIRONMENTALLY-FRIENDLY AND TRANSPARENT APPROACHES

In our previous work [5], we demonstrated two growing research fields in Artificial Intelligence (specifically in Machine Learning): Transparency and Carbon Footprint. We focused on answering the questions of what they are and why they are important (the readers are referred to this work for more details). Both of them are important to fulfill the emerging international demands and to adhere to the new regulations, such as "Right to Explanation" and "Green AI", and their value has triggered many researchers to contribute to the solution of related problems. While some of them focused on defining the concepts, others provided tools to help fellow researchers assess their proposed solutions based on these ever-demanding qualities.

Mitigating the serious impact of greenhouse gas emissions (such as carbon dioxide and methane) on the environment (such as the rise in sea levels and drought) requires urgent and collaborative participation from various fields. The machine learning community is not exempt from playing an active role in this call. Although there are different directions to address this, the primary emphasis here is limiting emissions' contribution. Complex algorithms can utilize enormous resources for days or months to complete their task; thereby releasing emissions that can harm the environment. It is found for example that the emissions produced by the NAS model during training are nearly equal to the carbon sequestered by 336 acres of U.S. forests in a year [5], [33].

Such activities have motivated many researchers in the community [33], [34], [35], [36], [37], [38] (with terminologies such as "Green AI", "Red AI", "Environmentally-Friendly AI", "Carbon Footprints of Machine Learning"), whether to prepare for any current or future recommendations or regulations, or to demonstrate ethical responsibility in providing harmless products or services to society. While some efforts recommended practices (such as computing and declaring the Carbon Footprint [35], choosing low-carbon intensity regions to train models [39]), others proposed easy-to-use tools to facilitate the quantification of the Carbon Footprint (i.e. an approximation of the carbon dioxide equivalent (CO_2eq) emissions which is "a measure of how much a gas contributes to global warming, relative to carbon dioxide" [40]), such as Green Algorithms [41] and CodeCarbon [42]. To quantify environmental costs during the computational tasks of learning algorithms in this paper, the CodeCarbon tool is used.

Other growing concerns about machine learning algorithms are related to their opaque nature and their complex internal interactions, which have sparked numerous research directions in academia (with terminologies such as "transparency", "interpretability", "explainability", "intelligibility", "(white or grey or black)-Box"), and prompted various regulations, guidelines, and standards by policymakers. However, there is no consensus on the definitions reached [43], as they vary between researchers. The author in [44] argues that interpretability is not a "monolithic concept" and has many ideas. Two notions of interpretability were proposed, such as transparency (i.e. how does the model work?) and post-hoc interpretability (i.e. what else can the model tell me?). Transparency consists of three main properties, such as Simulatability, Decomposability, and Algorithmic Transparency. Post-hoc interpretability presents techniques, such as text explanations, visualization and local explanations. In a recent work [45], the authors define "explainability" as "given a certain audience, explainability refers to the details and reasons a model gives to make its functioning clear or easy to understand".

Justifying algorithmic outputs is one of the reasons why we need to open the black-box methods particularly when they are applied directly to support activities that can save people's lives, such as analyzing social media texts during an earthquake, or in other critical domains such as emergency triage, criminal justice, and terrorist detection. Moreover, understanding the algorithmic outputs and decision-making process in detail can help to enhance them to produce better results [43], in order to serve society better.

Therefore, the work in this paper adds these two qualities to Performance as important cornerstones for the proposed event detection solution and uses one of the available measures to assess it. To the best of our knowledge, this is the first study to exhibit this level of scope in the context of real-time event detection, i.e. in terms of Performance, Carbon Footprint, and Transparency.

V. CONTEXTUAL ANALYSIS (CA)

To capture the relationships between words based on their co-occurrence in the same context, [6] proposed a novel method called Contextual Analysis. This method creates a tree-like structure called a hierarchical knowledge tree (HKT) in an unsupervised process. Depending on the strength of the relationship, this relationship can be expressed by combining the terms that appear in a similar set of sources (such as tweets) in a node in the tree and in its child nodes as a parent-child relationship. Studies in [5], [6], [46], and [47] are highly recommended for a thorough description of this strategy.

A. CA ALGORITHM

The Contextual Analysis algorithm begins by creating the first Hierarchical Knowledge Tree (HKT) Container (Seed_HKT), which initially encapsulates its first node. The node is created based on the word with the highest number of sources (the prominent word), see Fig. 1. All expected words in this HKT container, i.e. words that are strong enough to be in this HKT container according to α parameter, are checked

if they share a similar set of sources. Words that satisfy the criteria, governed by (β) parameter, are grouped in a single node. These parameters are usually set to 0.7 and 0.5 for (α) and (β), respectively. This selection is based on preliminary experiments on various datasets. Every node contains two different sets: a set of words and a set of sources.

Following the formation of the first HKT container (Seed-HKT), a set of remaining words for every node (i.e. not used in the creation of the preassessor node) is employed to construct sublevel HKT (Child-HKT). Every sub-level HKT must be linked to a parent node.

B. CA OBJECTS

According to [5], there are eight different objects in the constructed tree: Seed-Node, Seed-HKT, Child-Node, Child-HKT, Refuge-Node, Refuge-HKT, Orphan-Node, and Path (see Figure 2). These are defined as follows (note: for clarity, two more general definitions are added in this work: HKT container and Node):

Definition 1 (HKT Container): is a tree object that highlights words in related documents (tweets) in a corpus that belong to particular topics or sub-topics of the parent topic. These topics or sub-topics are represented as Nodes in a tree. A tree can have three different types of HKT containers: Seed-HKT, Child-HKT, and Refuge-HKT.

Definition 2 (Node): is a container that encapsulates one or more words as well as the documents (tweets) they appear in. This node is located in an HKT container and represents a topic or sub-topic. A tree can have four different types of Nodes: Seed-Node, Child-Node, Refuge-Node and Orphan-Node.

Definition 3 (Seed-Node): "is a container that encapsulates one or more words as well as the documents (tweets) they appear in. This node is located in Seed-HKT and represents the main topic in the corpus. A tree must have at least one Seed-Node".

Definition 4 (Seed-HKT): "is a container that highlights the most important words in related documents (tweets) in a corpus that belong to particular topics or categories. These main topics are represented as Seed-Nodes in a tree. A tree must have one Seed-HKT".

Definition 5 (Child-Node): "is a container that encapsulates one or more words as well as the documents (tweets) they appear in. This node is located in the Child-HKT, and it represents the sub-topics of the parent topic. A tree can have one or more Child-Nodes".

Definition 6 (Child-HKT): "is a container that highlights other important words in the related documents (tweets) that formed the parent's topic or category and belong to particular sub-topics or sub-categories of the parent topic or category. These sub-topics are represented as Child-Nodes in the Child-HKT. A tree can have one or more Child-HKTs, each must be linked to one parent node either a Seed-Node or a Child-Node".

Definition 7 (Refuge-Node): "is a container that encapsulates documents (tweets) that are in the corpus but none of their words appear in their Sibling-Nodes. These documents (tweets) cannot form a topic or category similar to the strength of their Sibling-Nodes. Any HKT container can have at most one Refuge-Node".

Definition 8 (Refuge-HKT): "is a container that highlights other important words in the related documents (tweets) in a corpus that belong to particular sub-topics or sub-categories of the parent's topic or category. These topics are represented as Child-Nodes in the Refuge-HKT. A tree can have one or more Refuge-HKTs, each must be linked to one Refuge-Node".

Definition 9 (Orphan-Node): "is a container that encapsulates one or more words and the documents (tweets) they appear in. This node is located in a Refuge-HKT where its parent is a Refuge-Node, and none of its ancestors is a Child-Node or a Seed-Node".

Definition 10 (Path): "is any node sequence from a starting node to any of its descendants Child-HKT or any specific descendants Child-Node in the tree along the parent-child connections. It should contain at least one node. The path represents a link between the topics and their sub-topics".

VI. DYNAMIC CONTEXTUAL ANALYSIS (CA) APPROACH CHALLENGES

The original CA approach builds a tree-like structure for the provided dataset, in a one-go procedure, to capture the relationship between words based on their appearance in the same context. However, the two questions that arise in the context of the dynamic nature of the event detection task are:

Q1: When should the created tree update itself?
To cope with the nature of the real-time event detection, careful attention should be devoted in order to guide when the update process should be conducted, i.e. should it be in an incremental basis for any incoming post individually? Or in a batch basis whenever there is a necessity for the change?

Q2: How can the tree update itself?
Up to now, there have been no attempts to describe how the trained CA incorporates the new incoming sources and their words into its created structure.

To answer the above questions, i.e. Q1 and Q2, one should consider the problems (related to the violation of the CA rules and assumptions) that can occur if the tree updates itself in an incremental or batch bases. Suppose that a new post presents to the trained tree. Incorporating this source into the tree will require finding the possible paths that the source's words can take through the tree. This is equivalent to searching through the tree to discover if there is any source in the training dataset similar, to some extent, to the new source. Taking into consideration the CA rules and assumptions, the question that may arise in this regard is: what are the required changes in the structure of the tree if the new source is to be incorporated?

Let's assume that there is an exact path found in the tree for the new post (i.e. there is at least one exactly matching source in the training dataset). Incorporating this post in the tree may

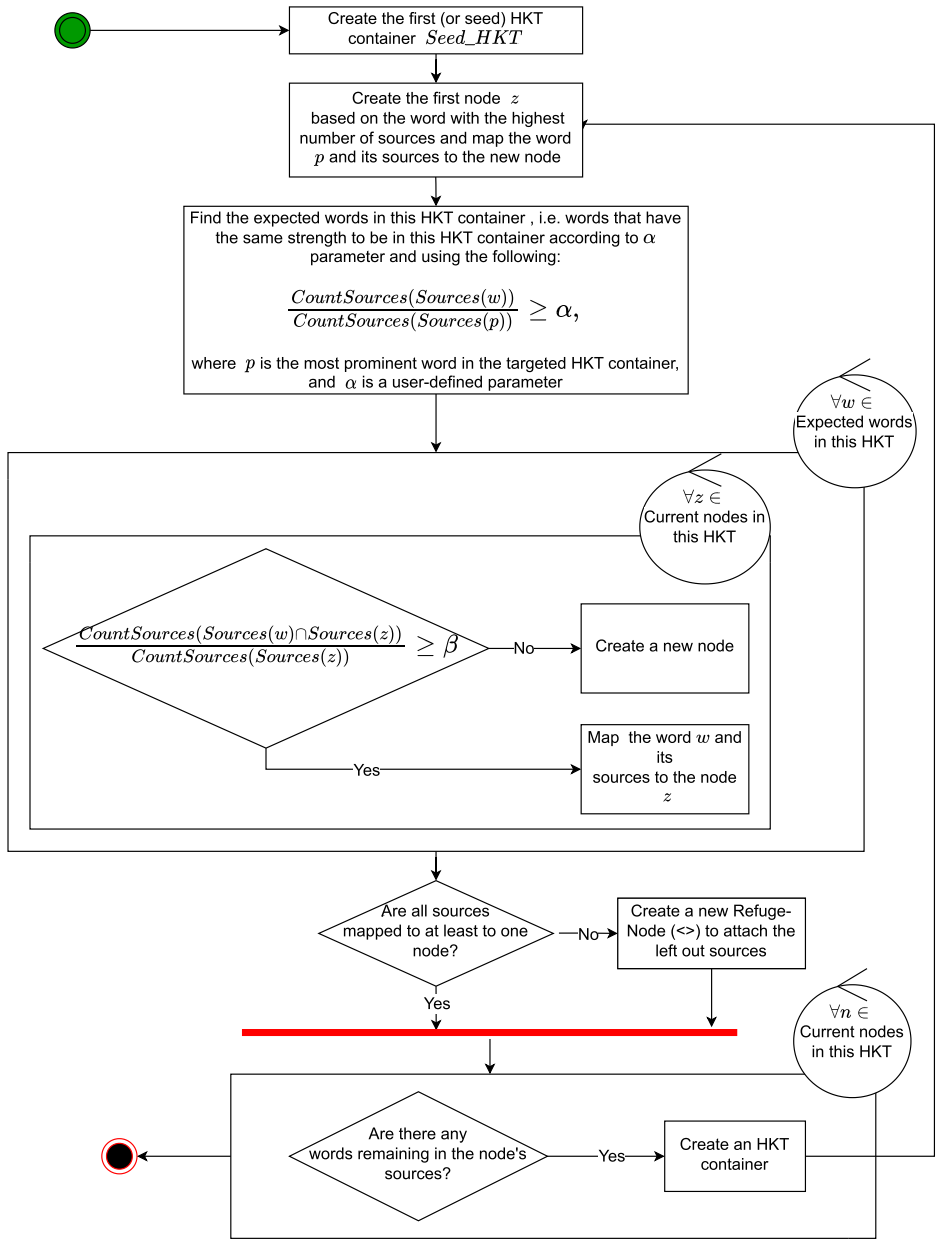

FIGURE 1. The contextual analysis (SECA)'s algorithm flowchart.

require adding it to the nodes in the matching path. This will increase the number of sources for the words in the node. According to the CA algorithm, all words in a certain HKT container should contain words of similar strength, which is governed by a threshold. Thus, increasing the number of sources for a word can affect the structure of the HKT container. This change, for instance, can boost the strength of the most prominent word in the container. Also, this can enhance the chance of other words in this level being the most prominent words. As a result, all words in this level should be checked against their eligibility, governed by the specified threshold (i.e. α and β), to be encapsulated in the current HKT container.

The CA algorithm creates its objects, i.e. HKT containers and nodes, based on the principle of aggregation over sources and their words. Changing these objects for every incoming post in real-time can be very expensive and very hard to achieve. This study focuses on developing a transparent approach that can detect emerging events. The various decisions that must be considered in order to incorporate every incoming post can lead to a less transparent approach. Whilst an individual data source can be important, the strength of the CA approach is in the aggregation of data sources, and therefore perhaps the focus should instead be on aggregating a number of sources together in order to decide whether any identified change is significant. Therefore,

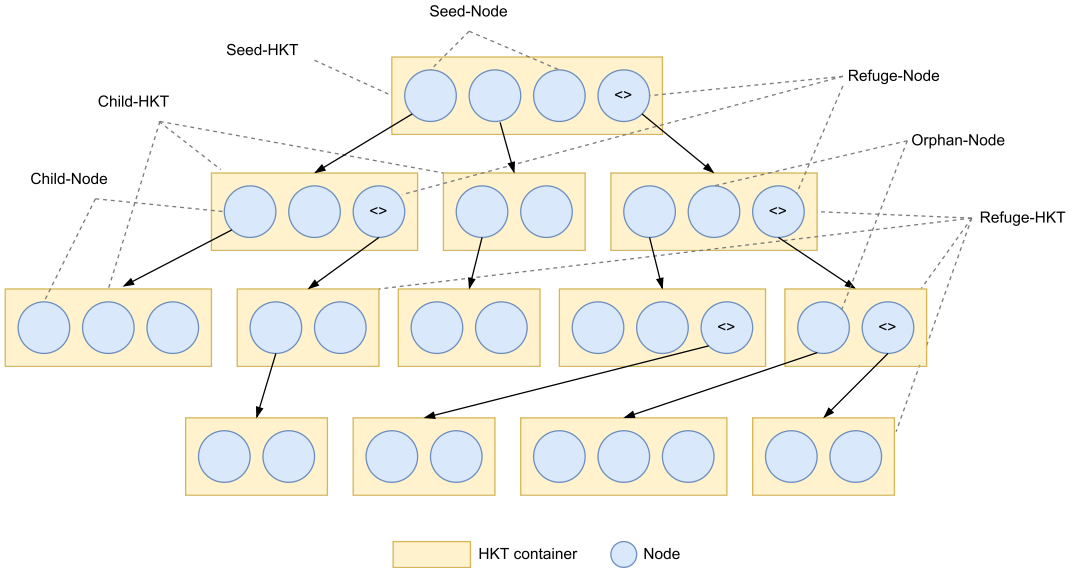

FIGURE 2. The contextual analysis (CA) components. Two main types of containers are shown: HKT containers and Nodes [5].

unnecessarily updating the tree for every incoming post can be less effective for the problem in hand.

To map the requirements of this study with the potential capabilities of the CA, a flexible approach needs to be proposed. It is hypothesised that this goal can be achieved if the tree updates itself on a batch basis, rather than an incremental, and whenever there is a necessity to do so. It is suggested that the changes in the tree should not be rigorously enforced, i.e., a change should not be performed for every incoming post. The question that arises in the context of this study, however, is what should trigger the need of change? What are the various aspects that should be considered for any change?

As mentioned above, the underlying assumption is that the words in a certain HKT container indicate the main topics (if they are located in Seed-HKT or Orphan-HKT) or sub-topics (if they are located in Child-HKT) in the training dataset. It is believed that for any new dataset, it is important to know whether these words still represent the data. In other words, with the new set of posts, do the words in the Seed-Nodes, for example, still highlight the main topics? Are there any emerging topics or sub-topics in the dataset that should be identified? Do the words that are encapsulated in a node, to represent any topic or sub-topic, still co-exist with each other (found in a similar set of sources)? Are there any emerging words for any topic or any new topics?

From all the above, in order to develop an approach that can efficiently answer all these questions, the following assumptions are made, and which all stem from the overarching hypothesis that new sets of data should demonstrate similar topics (at similar proportions) using similar words as the original dataset from which the HKT was constructed:

- The frequency at which a word is expected in a new batch of data can be predicted by the HKT container, both in terms of main topics, and also in terms of being in context with other words. These frequencies can be compared and used to highlight where in the tree changes could be required.
- The update on the tree should only affect the outdated objects whenever required. Starting from the top HKT container, one should examine if this object still represents the new dataset. This can be achieved by analyzing its underlying nodes, by which the HKT container should be reconstructed if their nodes do not effectively represent the data. Similarly, the checking process should be undertaken for each Child-HKT of every parent's Node.
- A Node does not represent the dataset if:
  - The strength of the node in an updated HKT container has changed significantly compared to its strength from the initial construction of the tree.
  - The unity of the words in the node changes significantly.

VII. SELF-EVOLVING CONTEXTUAL ANALYSIS (SECA)

The SECA approach is based on original Contextual Analysis (CA) principles [6]. However, differently from the standard CA method, the constructed tree evolves by modifying its HKT containers for any incoming new batch, and when there is a need to do so. The tree scans every HKT container and automatically determines whether a change is required for any individual object. Thus, it can be said that the tree continually adapts itself to represent the new input data, but more importantly, it can also describe what changes are being made in an automated manner and, in doing so, retains the key requirement of a transparent, explainable algorithm.

A. SECA ALGORITHM

This subsection describes the main steps of the proposed approach, i.e. SECA algorithm, which can be summarised as follows (see Figure 3):

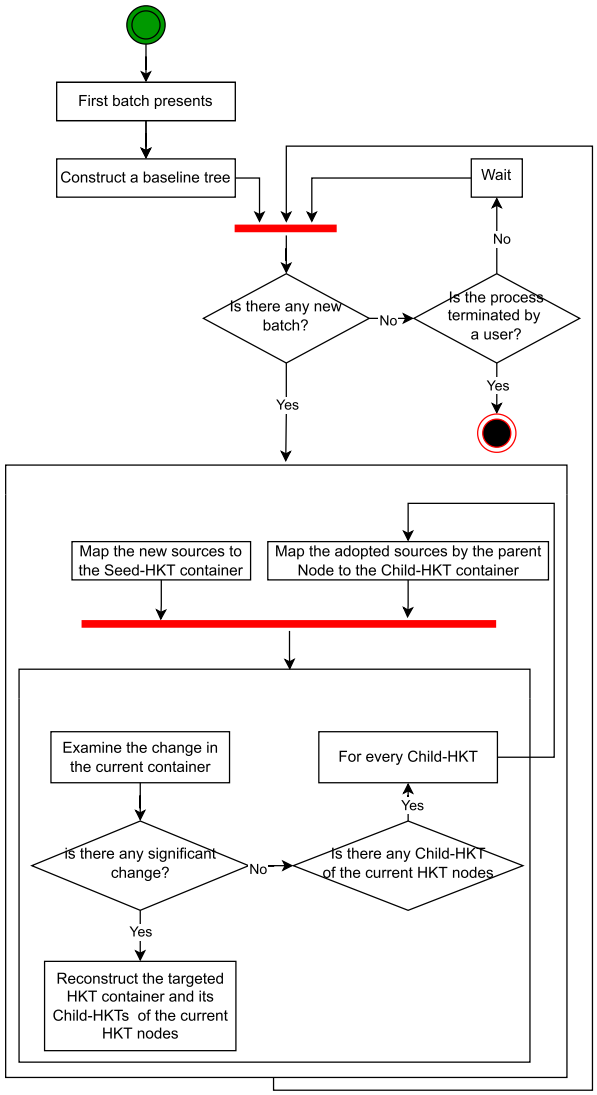

FIGURE 3. The self-evolving contextual analysis (SECA)'s algorithm flowchart.

Step 1: Construct the baseline tree. In this step, the baseline tree is constructed according to the standard CA algorithm that is presented in the original work [5], [47].

Step 2: Set the current scope to be the Seed-HKT.

Step 3: Map new sources to the targeted HKT container. Here, new data are presented in the tree. This process aims to determine the matching context paths for every source, according to the words they contain. A source is mapped to nodes that carry their words and represent the context. This procedure must follow the rules governing the construction of the original tree, in which a source can be found in more than one node of the HKT container, except for its Refuge-Node. Note: If the targeted HKT container is the Seed-HKT, then the mapping process uses all the sources in the batch; otherwise, the sources adopted by the parent node are utilized.

Step 4: Determine the significance level of the change of the current scope.
(Detailed description of this process can be found in the following section, i.e. VII-B)

Step 5: If there is a significant change, reconstruct the targeted HKT container and its Child-HKTs, otherwise, for every Child-HKTs container of the current HKT node, do steps 3, 4 and 5 recursively. If one of the errors (i.e. Word-Importance-Error, Alpha-Error, or Beta-Error) exceeds the specified threshold, reconstruct the targeted HKT container and its child-HKTs.

Step 6: For any new batch of data, repeat the steps from 2 to 5.

B. MEASURING THE SIGNIFICANCE LEVEL OF THE CHANGE IN AN HKT CONTAINER

The structure of any constructed tree and the positions of the words in the HKT container and its nodes are controlled by two important equations (equation (2) and (3)). Note: in this paper $Sources(x)$ denote a set of sources of a word or a node, more formally $\{s_1, s_2, \ldots, s_n\} \in Sources(x)$, and $CountSources$ refer to a function that counts the number of sources in a set (see equation (1)).

$$CountSources(S) = \sum_{i=1}^{n} s_i \tag{1}$$

where n is the number of sources in the set S.

Equation (2) is used to position words at the appropriate level in the tree, whereas equation (3) determines which words should be encapsulated in a single node.

$$\frac{CountSources(Sources(w))}{CountSources(Sources(p))} \geq \alpha, \quad \forall w \in HKTLevelWords \tag{2}$$

where p is the most prominent word, $HKTLevelWords$ is the list of words, in the targeted HKT container, and α is a user-defined parameter, usually 0.7 (based on our empirical observations).

$$\frac{CountSources(Sources(w) \cap Sources(z))}{CountSources(Sources(z))} \geq \beta, \quad \forall w \in NodeWords \tag{3}$$

where $NodeWords$ is the list of words, in the targeted Node z, and β is a user-defined parameter, usually 0.5 (based on our empirical observations).

It is apparent from the two equations that five important factors are directly responsible for the rules describing every HKT container in a tree. These factors are the number of sources for each word, the number of sources for the prominent word in an HKT container, the number of sources for a node, the α threshold, and the β threshold. It is hypothesized that these elements can be examined to detect any violation of the HKT container rules after the mapping process in Step 3.

Based on the above, three metrics are proposed to measure the degree of change in any HKT container: Alpha-Error, Beta-Error, and Word-Importance-Error (WI-Error). Alpha-Error is designed to determine whether there is a change in the logic that controls how the words appear in an HKT container, which places the words with similar strength (according to the number of sources they appear in) in the same container. In other words, it captures if a topic is not being discussed at a similar level or if there is a change in the strength of the words representing the topic. Beta-Error aims to identify the change in the logic that controls how the words appear in a node inside the HKT container. This can enable the identification when the representing words of a topic do not co-exist in a similar set of sources. The Word-Importance-Error seeks to detect if there are new words (a new word in this context denotes that the word did not exist in the HKT container in the baseline tree) in an HKT container. This can help identify the presence of new words to describe either an existing or new topic at that level.

Therefore, starting from the Seed-HKT (top-down), these three values are computed for each HKT container.

Note: Notions $state_0$ and $state_1$ represent HKT before and after the presence of a new batch, respectively.

1) ALPHA-ERROR

This error is intended to measure whether there is any violation of the targeted HKT container adoption rule. Every word should be checked for its eligibility to exist at the current level. Looping through all old words (i.e. words found in $state_0$) in the container, the deviation from the α threshold is measured.

Suppose that the strength of a word in the targeted HKT container in $state_0$ can be calculated using the following formula:

$$Strength_0(w) = \frac{CountSources(Sources(w))}{CountSources(Sources(p))}, \quad w \in HKTLevelWords \tag{4}$$

where p is the most prominent word and $HKTLevelWords$ is the list of words in $state_0$, in the targeted HKT container.

Also, suppose that the strength of the word in the targeted HKT container in state1 can be calculated using the following formula:

$$Strength_1(w) = \frac{CountSources(Sources(w))}{CountSources(Sources(q))}, \quad w \in HKTLevelWords \tag{5}$$

where q is the **new** most prominent word and $HKTLevelWords$ is the list of words in $state_1$, in the targeted HKT container. It is important to note that the prominent word can change after the mapping process.

Therefore, the AlphaError can be calculated using the following formula:

$$AlphaError = \frac{\sum_{i=1}^{n} |\alpha - Strength_1(word_i)|}{n} \iff Strength_1(word_i) < \alpha \tag{6}$$

where n denotes the number of words in the targeted HKT container in $state_0$. Although there is a chance that the strength of any word may differ considerably between the two states, violation of the alpha rule is the main interest here.

2) BETA-ERROR

To measure the eligibility of words in a node, every word should be examined according to the rules of that node. This can be accomplished by measuring the deviation from the β threshold between $state_0$ and $state_1$.

Suppose that the eligibility of a word in a node inside the targeted HKT container in $state_0$ can be calculated using the following formula:

$$Elig0(w) = \frac{CountSources(Sources(w) \cap Sources(z))}{CountSources(Sources(z))}, \quad w \in NodeWords \tag{7}$$

where $NodeWords$ is the list of words, in the targeted Node z in $state_0$.

In addition, suppose that the eligibility of a word in a node inside the targeted HKT container in state1 can be calculated using the following formula:

$$Elig1(w) = \frac{CountSources(Sources(w) \cap Sources(z))}{CountSources(Sources(z))}, \quad w \in NodeWords \tag{8}$$

where $NodeWords$ is the list of words, in the targeted Node z in state1.

Therefore, the BetaError can be calculated using the following formula:

$$BetaError = \frac{\sum_{i=1}^{n} |\beta - Elig_1(word_i)|}{n} \iff Elig_1(word_i) < \beta \tag{9}$$

where n is the number of words in the targeted HKT container in $state_0$. The violation of the Beta rule is the main interest here.

3) WORD-IMPORTANCE-ERROR

After the mapping phase, any HKT container may adopt new words from the dataset. To measure this change, Word-Importance-Error metric is proposed.

Suppose that the importance of a word in the targeted HKT container in $state_0$ can be calculated using the following formula:

$$Importance_0(w) = \frac{CountSources(Sources(w))}{\sum_{i=1}^{n} CountSources(Sources(word_i))} \tag{10}$$

where n denotes the number of words in the targeted HKT container in $state_0$.

In addition, suppose that the importance of a word in the targeted HKT container in $state_1$ can be calculated using the following formula:

$$Importance_1(w) = \frac{CountSources(Sources(w))}{\sum_{i=1}^{m} CountSources(Sources(word_i))} \tag{11}$$

where m denotes the number of words in the targeted HKT container in $state_1$ (i.e. old and expected words after the original mapping of the training data).

Therefore, the Word-Importance-Error can be calculated using the following formula:

$$WordImportanceError = 1 - \sum_{i=1}^{n} Importance_1(word_i) \tag{12}$$

where n denotes the number of words in the targeted HKT container in $state_0$.

Note: although equation (4), (7) and (10) are not used to measure the proposed three errors (i.e. Alpha-Error, Beta-Error, and Word-Importance-Error), they are presented to indicate the values of the different measures (i.e. strength, eligibility and importance) in $state_0$.

C. SECA-LIGHT

The above algorithm is designed to detect changes in the constructed tree in an accumulative manner, i.e. all sources are stored in memory to capture the significance of the changes between $state_0$ and $state_1$. Although this can be useful in some applications, such as the need to identify the main changes in word relationships in the presented sources since the first incoming batch, however, this can be very difficult to achieve in environments where the sources continuously arrive at a rapid rate and are characterized by a very large volume. The time and space required for SECA are expected to increase significantly. To overcome this challenge, we present a lighter version of the proposed approach referred to as SECA-Light.

We modified the algorithm by adding one more step after processing each incoming batch, i.e. after Step 5 in Section VII-A. All outdated sources will be discarded from memory. Here, we define the outdated sources as the samples that arrive before the batch number $\theta - \gamma$, where θ is the current batch, and γ is the specified number of batches to keep in memory. However, the process otherwise remains the same meaning that the generated tree's details, such as HKTs and Nodes' contents, are stored in the disk for further analysis.

D. CLUSTERING APPROACHES USING CONTEXTUAL TREE

Sources such as microblogging posts frequently develop clusters of actual events [48]. In this study, we aim to demonstrate the capability of the proposed approach, i.e. SECA, for detecting events. One way to accomplish this is to demonstrate its ability to group similar sources into coherent clusters.

We empirically investigated the generated HKT containers and nodes for various input datasets to qualitatively assess the quality of nodes' clusters. We observed that the generated trees might promote some generic words in the Seed-HKT or Orphan-HKTs, which may not represent real stories. To give an example, we found the word "news" appeared in the upper-level containers because it was presented in many sources that discussed different events in subsequent time frames. Although the event's related words were still captured in one of the descendants' nodes of the generic word and other Orphan-Nodes, one might argue that event detection solutions should limit the occurrence of these words. On the other hand, highlighting these words can provide insights into current discourse. For instance, it may reveal some details about the most important topics in the news. It should be noted that this capability has not been given sufficient attention in previous studies on the event detection task, in which the quality of generated clusters is assessed based on high-level output and ignores the hierarchical level of the details. That said, to provide solutions to satisfy the different requirements of the problem based on the generated tree, two different clustering methods (direct and indirect) are proposed and described in detail below. The direct approach is proposed to offer the full details of word relationships generated by the tree, which can provide a deep insight into the information for the event detection task. The indirect approach, however, is intended to shed light on the most important branches in the tree in order to capture the most significant topics. The main focus here is to enable quick navigation of key concepts that can be helpful in applications that require urgent action.

1) DIRECT APPROACH

We consider the nodes that appear in the Seed-Nodes and the Orphan-Nodes to be the generated clusters of the presented sources. Therefore, any source used to construct a node is deemed a cluster member.

2) INDIRECT APPROACH

Here, the constructed tree is used to generate proxies for clusters of potential events. These event descriptors are then used to cluster the sources. The following steps highlight this approach:

Step 1: After generating the contextual tree, every Seed-Node and Orphan-Node forms a potential cluster which is represented by a list of its words and their contextual value. This figure is computed based on node strength in the context of its root node (i.e. Seed-Node or Orphan-Node), as follows:

$$ContextualValue(w) = \frac{CountSources(Sources(n_x))}{CountSources(Sources(n_{Root}))}, \quad w \in NodeWords \tag{13}$$

where $NodeWords$ is the list of words in the targeted Node n_x and n_{Root} is its parent Seed-Node or Orphan-Node.

Step 2: To limit the fragmentation of clusters, any two potential clusters that share at least one of their top words are merged (governed by a threshold).

Step 3: Then, the clusters' descriptors, i.e. their words and contextual value, are used as proxies to measure

how close each source is to every potential cluster.

$$Proximity(cluster, source) = \sum_{i=1}^{n} ContextualValue((w_i)), \quad \forall w \in SourceWords \wedge w \in ClusterWords \quad (14)$$

where *SourceWords* is the list of words in the source *source* and *ClusterWords* is the list of words in the potential cluster *cluster*.

Step 4: This is followed by mapping each source to one cluster with the highest proximity value.

Note that in this approach a cluster can be linked to at least one Seed-Node or Orphan-Node in the contextual tree.

VIII. EXPERIMENTS

A. EXPERIMENT ENVIRONMENT

All codes for the experiments were developed using C#, Python, and Structured Query Language (SQL) programming languages. For details regarding the hardware and software configurations, see Table 1.

TABLE 1. Hardware and software configurations.

Platform system	Windows-10-10.0.19043.1654
Installed RAM	64.0 GB (63.8 usable)
GPU model	NVIDIA GeForce GTX 1650 Ti
Processor	Intel(R) Core(TM) i9-10885H CPU @ 2.40GHz 2.40 GHz
Hard Disk	954 GB (601 GB free)
Software	Visual Studio 2019 (community), SQL Server Express 2017, SQL Management Studio (V18.2).

B. EXPERIMENT DATASET

For the experiments in this paper, we used two datasets:

- Tweets [49]: It comprises of 30,322 tweets from TREC 2011-2015 microblog track, annotated into 269 categories. The average word count length of the tweets is 7.97.
- News [50]: It consists of 11,109 news titles, labeled with 152 categories. The average word count length of titles is 6.23.

The rationale behind selecting these datasets is related to their extensive usage in relevant literature, and their representation of real-world scenarios.

The work in [48] published a preprocessed version of the short texts from these datasets, available at [51]. We further generated eight different experimental themes, the details of which are listed in Table 2 and Fig. 4.

These setups are based on considering event detection as a cluster evolution monitoring problem in social streams, in which birth, death, growth, decay, merge and split are examples of common evolution patterns [52]. Themes A to F were carefully curated to highlight some of the possible evolutionary scenarios of events over a period of time, based on the ratio of their sources (see equation (15)). This reflects different real-world scenarios that could be encountered during an event, and the data needs to be curated in this manner in order to be able to assess how well (or not) the types of frequency changes in an event can be identified. Without the manipulation of the data in this way, it would be more difficult to understand how well the algorithm performs under different event scenarios that are changing in time. We do not believe that a dataset of this type has previously been presented for analysis for dynamic topic detection.

For example, in Theme D, there are two different groups of events, with each group consisting of three events. While the events in Group 1 evolve to a peak at Cycle 6, Group 2 events show the opposite trend. Then, both groups change their development manner to have an equal number of sources in cycle 12. In contrast, for Themes G and H, we used the published dataset with their natural evolutionary development.

$$SourceRatio(e) = \frac{AccuSourcesofEvent(e)}{AccuSources} \quad (15)$$

where *AccuSourcesofEvent* is the accumulative number of sources of an event e from the first cycle (batch) to the current one, and AccuSources is the total number of presented sources for the same interval.

To the best of our knowledge, there is no ground-truth dataset that can offer a hierarchical representation of topics and their sub-topics in the context of event detection. This limits our ability to quantitatively assess the performance of SECA and the baseline methods in capturing not only the main topics but also the hierarchical level of details.

C. EXPERIMENT DETAILS

1) TEXT PRE-PROCESSING

We used the pre-processed datasets published in [48]. We tokenized the text based only on the white space between any set of characters.

2) SECA IMPLEMENTATION

To implement the proposed SECA algorithm, a new software has been developed using C#, see Figure 5. The software primarily comprises three main engines: Crawler Engine, Source Cleaner Engine, and SECA Engine. Each of these engines is responsible for specific tasks. While the Crawler Engine primarily handles fetching raw data (e.g., tweets) from sources' providers, the Source Cleaner is responsible for the preprocessing phase and preparing the dataset for the SECA algorithm. The SECA Engine is designed for implementing the SECA algorithm with two main functions: building a tree and updating it when there is a need to do so. The software features two main dashboards: the Software Portal Dashboard and the Analyzer Dashboard. The Software Portal Dashboard facilitates access to all three engines, providing centralized access for interaction with the

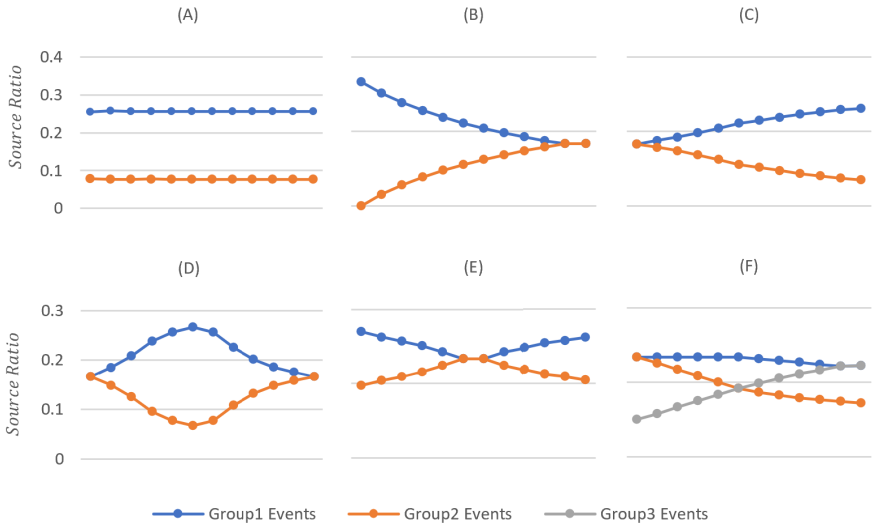

FIGURE 4. Summary of the dataset setup. A total of 72 sub-datasets, 12 cycles for each Theme A to F, were manually curated using TREC 2011-2015 microblog track. Each group comprised three events that consisted of a similar number of tweets. Details of each Theme setting are described in Table 2. Note: the x-axis indicates the cycle number (batch number) used to represent incoming batches over time.

TABLE 2. Dataset setup for Themes A-H. There were eight primary datasets, each of which was divided into 12 cycles to form its sub-datasets (in total, 96 sub-datasets were generated). Themes A-F were manually designed, using the Tweets dataset, with different settings over time, see Fig. 4. The datasets were designed to investigate different ratios of source counts against each other - for example Theme B has two groups, with group 1 having a high number of sources in the first time step, and the second not existing. Then over time the first group reduces in number, and the second increases in number until they reach the same level. Variations of ratios over time between groups are examined in the different themes. For Themes G and H, no changes were made to the datasets, i.e. News and Tweets, except for truncating them to create an equal number of sources in each cycle (batch).

Theme	Total No. Sources	No. Sources Per Cycle	Group ID	No. Events	Total No. Sources Per Event	Importance of Group's Event Per Cycle
A	2,280	190	1	3	582-585	Fig.4-A–Group1
			2	3	176	Fig.4-A–Group2
B	2,280	190	1	3	375-381	Fig.4-B–Group1
			2	3	381	Fig.4-B–Group2
C	2,280	190	1	3	596-599	Fig.4-C–Group1
			2	3	162	Fig.4-C–Group2
D	2,280	190	1	3	370-382	Fig.4-D–Group1
			2	3	382	Fig.4-D–Group2
E	2,280	190	1	3	508-511	Fig.4-E–Group1
			2	3	250	Fig.4-E–Group2
F	2,280	190	1	3	316-331	Fig.4-F–Group1
			2	3	103	Fig.4-F–Group2
			3	3	331	Fig.4-F–Group3
G	8,880	740	1	152	2-354	-
H	24,000	2,000	1	227	1-607	-

software's various engines. The Analyzer Dashboard offers comprehensive presentations of the generated trees. This includes detailed information about the structure of these trees, as well as the reasoning behind any decisions for modifications.

3) BASELINE METHODS

To test the efficacy of the proposed methods (SECA and SECA-Light), three well-known baseline methods were selected: KMeans, Nonnegative Matrix Factorization (NMF), and MStream, with the implementations found in [53] for the first two methods[1,2] and in [51] for MStream.

a: KMEANS

KMeans algorithm is one of the most frequently used clustering methods [54]. It divides a dataset into (K) distinct clusters (C). The number of clusters (K) must be predetermined and a proper distance function, such as the Euclidean distance, must be chosen. Minimizing the sum of squared distances across all clusters is the traditional KMeans objective. The key steps of this algorithm are as follows:

1) Initialise cluster centres.
2) Assign data points (N) to the closest cluster center.
3) Recompute all cluster centers as the mean of assigned data points.
4) Repeat 2 and 3, (I) number of times.

The first step in the algorithm determines the initial clusters. However, this task can be difficult in practice, because random cluster selection can produce varied clustering results. Various methods can be used to improve the initialization process, such as the KMeans++ algorithm suggested by the work in [55].

[1]https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
[2]https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html

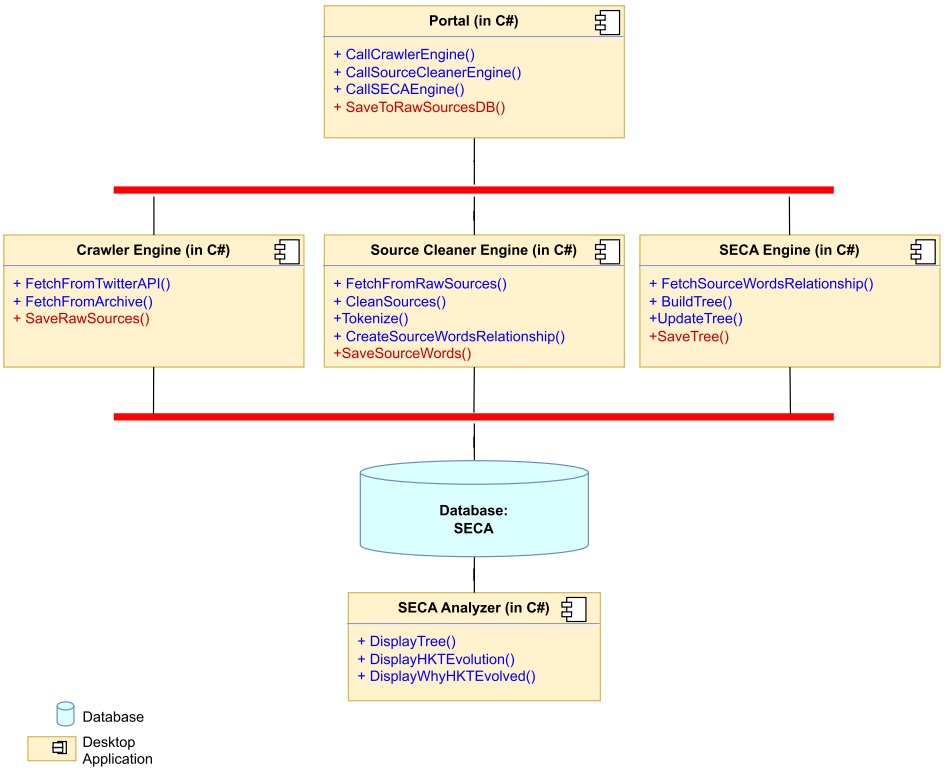

FIGURE 5. SECA software structure. The software is primarily composed of three main engines: Crawler Engine, Source Cleaner Engine, and SECA Engine.

b: NONNEGATIVE MATRIX FACTORIZATION (NMF)

Another well-known method that has been widely used for topic modeling is Nonnegative Matrix Factorization (NMF) [56]. It aims to decompose a non-negative matrix into two (approximated) lower-dimensional non-negative matrices. For a given matrix $H \in \mathbb{R}^{m \times n}$, it computes an approximate factorization such that

$$H \approx UV \tag{16}$$

where $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{k \times n}$ are two factors with the following optimization problem [57]:

$$Min_{U,V} \left\| H - UV^T \right\|_F^2 \tag{17}$$

NMF receives a document-term matrix to find document-topic and topic-term matrices. The number of k latent topics is assigned to the algorithm.

c: MODEL-BASED SHORT TEXT STREAM CLUSTERING (MSTREAM)

MStream is a short-text clustering algorithm that was proposed in [48]. It uses the Dirichlet Process to automatically generate a number of clusters in the presented documents. The authors proposed an improved version of the algorithm, called MStreamF, with forgetting rules to discard outdated documents. Each cluster is represented by a list of word frequencies, a number of documents, and a number of words.

4) EXPERIMENT SETTING

For the proposed SECA, we set α=0.7, β=0.5, $AlphaError = 0.1$, $BetaError$=0.2 and $WordImportanceError = 0.3$. For KMeans, NMF, and MStream, the default hyperparameters were adopted. We set the number of topics for KMeans and NMF to 6 for Themes A-F, 170 for Theme G, and 270 for Theme H. However, SECA and MtStream do not require specifying the number of topics; instead, they detect topics automatically. In addition, KMeans and NMF are retrospective approaches that cannot be directly applied to online event detection. We solved this problem by retraining the models from scratch using feeds from the first batch to the current batch.

D. EXPERIMENTAL DESIGN

Four experiments were designed to examine the capability of the proposed approach to detect events. The details of these are described in the following subsections.

1) EVENT CLUSTERING ALGORITHM EFFECTIVENESS - EXPERIMENTAL SETUPS

a: EVALUATION METHODOLOGY

To evaluate the quality of the generated events' clusters by SECA and the baseline methods, three metrics were used, namely, purity, normalized mutual information (NMI), and BCubed f1 (see equation (18), (19) and (25)).

Purity: By labeling each produced cluster by the dominant category (i.e. the most common category in the encapsulated

tweets), the quality is scored as follows:

$$Purity(W, C) = \frac{1}{N}\sum_{k=1}^{K}\max_{j}|w_k \cap c_j| \tag{18}$$

where W is the set of clusters, C is the set of categories, w_k: the set of tweets in the category k (i.e. according to the annotated label), c_j: the set of tweets in the cluster j and N is the number of tweets in the dataset.

NMI: To compute the tweets' clusters quality using NMI, the following formula is used:

$$NMI = \frac{2I(W;C)}{H(W)+H(C)} \tag{19}$$

where I(W;C) is the mutual information between W and C (see equation (20)), H(W) is the average entropy of the categories (see equation (21)) and H(C) is the average entropy of the clusters(see equation(22)).

$$I(W, C) = \sum_{k=1}^{K}\sum_{j=1}^{J}\frac{|w_k \cap c_j|}{N}\log\frac{N|w_k \cap c_j|}{|w_k||c_j|} \tag{20}$$

$$H(W) = -\sum_{k=1}^{K}\frac{|w_k|}{N}\log\frac{|w_k|}{N} \tag{21}$$

$$H(C) = -\sum_{j=1}^{J}\frac{|c_j|}{N}\log\frac{|c_j|}{N} \tag{22}$$

BCubed f1: The BCubed f1 of any tweet in the dataset is the average BCubed precision and BCubed recall (see equation (27), (26), (25)). The BCubed precision for a tweet captures the number of tweets in the cluster that have the same category according to the annotated label. The BCubed recall measures the number of tweets in the same category found in the cluster. The overall f1 BCubed is the average f1 of all tweets in the dataset.

$$BCubedPrecision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{23}$$

$$BCubedRecall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{24}$$

$$BCubedf1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{25}$$

Each algorithm was fed with Themes A to H sub-datasets according to the setup shown in Table 2.

b: RESULTS

All the results on the Themes A-H datasets are shown in Table 3 (with the best and worst results for each Theme highlighted in the table). On Themes A-F datasets, SECA variants achieved the highest performance across all three metrics: NMI (>92%), f1-BCubed (>89%), and purity (>99.5%). KMeans and NMF exhibited relatively high performance. MStream has the lowest performance for NMI and f1-BCubed, with scores consistently below 74%; however, it achieved competitive performance on purity. For Theme G, the best NMI (88.95%) and f1Bcubed (66.81%) scores were obtained by MStream, whereas the lowest values across all three metrics were recorded for KMeans. Both SECA-Light Indirect and MStream showed superiority over Theme H, with slight improvements by the former. Overall, the SECA variants acquired the best results in most experiments.

TABLE 3. Performance comparison between the four methods (KMeans, MStream, NMF, and SECA) on the eight subdatasets (Themes A-H)(see Table 2). For SECA, we present the results of direct and indirect clustering approaches for its two versions: complete and light. Average NMI, BCubed f1 and Puirty scores are shown. The best results are highlighted in green and the worst in red.

	Th.	K-Means	M-Stream	NMF	SECA Direct	SECA Indirect	SECA-Light Direct	SECA-Light Indirect
Avg. NMI	A	84.77	72.74	86.32	92.73	92.7	**92.74**	91.83
	B	85.06	72.85	86.52	**94.58**	94.39	94.56	93.45
	C	85.07	70.17	87.24	**92.8**	91.87	92.43	91.52
	D	87.31	71.67	89.67	93.16	**93.69**	93.51	93.02
	E	84.9	73.51	88.64	93.08	93.29	**93.79**	92.21
	F	80.83	73.29	84.65	90.96	90.17	**90.99**	89.86
	G	75.36	**88.95**	86.87	86.21	88.3	85.94	87.93
	H	73	88.4	80.76	77.75	87.52	84.3	**88.85**
Avg. f1	A	80.31	54.7	79.27	89.48	89.53	**89.66**	87.91
	B	81.09	53.38	82.01	91.82	**92.64**	92.27	91.06
	C	81.54	50.98	80.74	90.1	89.32	**90.38**	88.42
	D	84.12	56.58	86.21	91.63	92.22	**92.45**	91.34
	E	80.04	58.23	84.21	90.33	90.77	**91.53**	89.26
	F	75.53	49.03	80.63	84.89	83.45	**85.72**	82.98
	G	45.14	**66.81**	60.59	54.63	63.03	55.37	62.74
	H	43.84	73.68	56.65	53.05	71.14	64.32	**74.08**
Avg. Purity	A	94.12	99.78	97.68	99.82	99.82	99.69	**99.91**
	B	94.43	99.69	96.54	**99.87**	**99.87**	99.61	99.82
	C	95.13	**99.96**	99.61	99.87	99.87	99.65	99.82
	D	96.49	**99.91**	99.3	99.83	99.82	99.65	99.82
	E	93.51	99.74	98.38	**99.91**	99.82	99.74	99.87
	F	82.85	99.08	90.7	**99.64**	99.56	99.55	99.56
	G	68.16	74.27	79.56	71.67	**80.36**	70.73	79.2
	H	77.55	85.65	85.17	75.53	**89.18**	81.44	89.03

2) EVENT COVERAGE - EXPERIMENTAL SETUPS

a: EVALUATION METHODOLOGY

To evaluate the capability of the methods to detect events, based on the annotated dataset, we measured the recall value (the proportion of events in the ground truth that are detected by the system), see equation (26). An event is detected if 60% of a cluster's sources are about a single event. Choosing the recall value, rather than the precision (see equation (27)), is due to the possibility that more events can be detected by the algorithms that are not captured in the ground truth datasets.

$$Recall = \frac{\#DetectedEvents}{\#DetectedEvents + \#UnDetectedEvents} \tag{26}$$

$$Precision = \frac{\#DetectedEvents}{\#DetectedEvents + \#WronglyDetectedEvents} \quad (27)$$

Here, the algorithms received Themes G and H sub-datasets. We selected these two datasets because they represent real-world scenarios.

b: RESULTS

The results of this experiment are presented in Fig.6. Overall, the figure shows that the seven algorithms achieved varying levels of performance for the two Themes. While NMF achieved the highest performance on Theme G, the indirect versions of SECA obtained the best results on Theme H. The recall values of MStream and SECA have improved on Theme H, with an average increase of 0.14 and 0.17, respectively. On average, for all batches in both Themes setups, NMF performed the best; however, the difference is not significant when it is compared to SECA and SECA-Light indirect approaches (p>0.05, paired t-test).

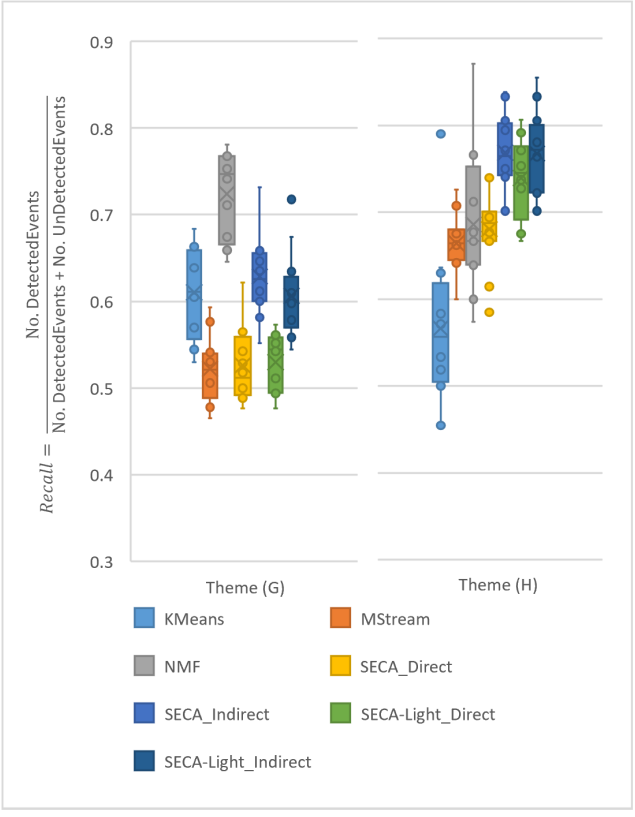

FIGURE 6. Performance comparison between seven methods (KMeans, MStream, NMF, SECA Direct, SECA Indirect, SECA-Light Direct, and SECA-Light Indirect) on Themes G and H sub-datasets. The recall scores are presented.

3) CARBON FOOTPRINT ASSESSMENT

a: EVALUATION METHODOLOGY

Recently, researchers have proposed various tools to quantify environmental costs during the computational tasks of learning algorithms. We used a tool called CodeCarbon, available in [42].

CodeCarbon is an open-source package that estimates the CO_2eq in kilograms using the following equations [42]:

$$CO_2eq = CarbonIntensity \times PowerUsage \quad (28)$$

where *CarbonIntensity* and *PowerUsage* are the consumed electricity (in kgCO_2/kWh) and power (in kWh), respectively, for computation.

The carbon intensity is calculated using a combination of energy sources, including fossil fuels (such as natural gas) and renewables (such as solar). To estimate, one of three different methods are adopted:

- by using available carbon intensity of electricity per cloud provider or per country. For example, the carbon intensity recorded for the United Kingdom in 2020 was 209 gCO_2/kWh.
- or, by computing intensities using predefined values per energy source (e.g Petroleum= 816 kg/MWh, Wind= 26 kg/MWh) and their proportional usage, according to the following equation:

$$NetCarbonIntensity = \sum_{AllEnergySources} CarbonIntensity_{Source} \times Percentage_{Source} \quad (29)$$

- or, a world average value is considered

When activating the emissions monitoring system, the power supply for the hardware is frequently observed at regular intervals, with a default duration of 15 seconds.

To estimate the carbon dioxide equivalent (CO_2eq) emissions of the six approaches, each algorithm was fed with the Theme H dataset (see Table 2) in 12 different batches, and the cumulative amount of emitted CO_2eq during the task was recorded.

b: RESULTS

The results of the Carbon Footprint assessments are presented in Table 4. As shown in the table, SECA-Light contributed the lowest emissions among the other algorithms. MStream and SECA emitted almost similar amounts of CO_2eq, 1.56 and 1.69 grams respectively. Yet, these amounts were cut down by 0.65 and 1.09 grams in their more efficient version, i.e MStreamF, and SECA-Light respectively. KMeans is the most carbon-intensive algorithm.

4) TRANSPARENCY ASSESSMENT

a: EVALUATION METHODOLOGY

For transparency assessment, we used an approach similar to that adopted in our previous studies [5], and [46]. Based on the efforts made in [44], the transparency of the methods was evaluated using the following questions [58]:

- Q1: "Is the entire model simple enough to be fully understood by a user?"
- Q2: "Is each part of the model (each input, parameter, and calculation) intuitively explainable?"

TABLE 4. Results of emissions released by six methods (i.e. KMeans, MStream, MStreamF, NMF, SECA, and SECA-Light) during the training tasks. Each method is fed with Theme H dataset (See Table 2). The accumulative results are shown.

Method	Duration (in second)	CO_2eq Emissions (in gram)
KMeans	2499	9.2
MStream	480	1.56
MStreamF	284	0.94
NMF	189	0.69
SECA	492	1.65
SECA-Light	**163**	**0.56**

- Q3: "Is the algorithm deterministic (non-stochastic) without using any random numbers?"

In [5], we further introduced two other requirements that are related to the interpretability of the results and the question of the data being asked, and whether the algorithms are capable of directly answering them, such as:

- "What words are important for a topic?"
- "What is the relationship between the words that are important for a topic?"

In other words, the algorithms are further assessed in their ability to answer these two questions, Q4 and Q5, with a binary output of yes or no. The targeted audience of these questions is non-technical users, and the sample size is limited to 10 short texts.

b: RESULTS

KMeans, NMF, and MStream require some technical capabilities to understand their internal mechanisms; mathematical and statistical skills are required to comprehend the different parts involved in their process. In addition, they are non-deterministic, each of which may generate different outputs for the same data feed. Therefore, the answer to the first binary assessments (Q1, Q2 and Q3) is negative. By contrast, SECA is considered a transparent approach due to its nature. A non-technical user can follow the development process of a tree, in order to create the HKTs and the nodes. Moreover, it can offer clear justifications for the algorithmic decisions during the tree's restructuring process, i.e. where and why has a change been made?

For Q4, the four methods can generate the top words for the detected topic. To give an example, we manually explored the top ten words related to "Kanye West" (an American rapper, producer, and songwriter) after feeding the MStream by the first three batches from Theme G. The words **"Kanye"** and **"West"** appeared on three different topics:

- Topic A: **"west"** (0.96), "Kardashians" (0.92), "Kardashian" (0.33), "verdict" (0.26), "partner" (0.24), "preach" (0.18), "low" (0.13), "exclusive" (0.11), "release" (0.009) and "square" (0.009).
- Topic B: **"west"** (0.068), "spoof" (0.058), "rogen" (0.057), "james" (0.057), "parody" (0.057), "kardashian" (0.56), "kardashians" (0.052), "franco" (0.052), "verdict" (0.044) and "kim" (0.0386).
- Topic C: "urban" (0.047), **"kanye"** (0.047), "lawson" (0.0389), "alleged" (0.0389), "revealed" (0.0195), "refused" (0.0195), "sp" (0.0195), "lock" (0.0195), "hair" (0.0078), "advancing" (0.0078) and "ahead" (0.0078).

On the other hand, SECA has advantages in capturing the relationships among words. Since it captures a hierarchal relationship of the words and their sources, it reveals the words that appear in the context of the other words, as a parent-child relationship. This provides more information on each topic. For instance, using the same input in the above example, SECA decided that the two words, "kanye" and "west", should be encapsulated in one node to represent a certain concept (see Fig. 7). Moreover, three "important" separate sub-topics in the context of the main topic were discovered, such as: "kim Kardashian", "seth franco rogen" and "nike". This is in contrast to the other approaches described above that simply return lists of words and so do not capture this additional information. Also, "day parade", "black Friday", "storm travel" and "Hanukkah" are the most important sub-topics under "thanksgiving". More granular details can be obtained by traversing the branches of each node in the tree in an intuitive fashion. Thus, SECA is the only method among the others that can positively answer Q5.

To support users' trust and understandability of the algorithmic outputs, SECA provides other information related to the decisions made during its incremental learning process. At any point in time during the process, SECA can answer two other questions: What are the changing topics and sub-topics? Why and how have these topics, and changes, been identified and generated? To give an illustration, Fig. 9 shows the affected HKT container when a change has been identified in the Child-HKT of the node "nokia lumia" in Cycle 3. The changes occurred due to the trigger of the WordImportance error, which exceeded the specified threshold of 0.3. The words "lunch" and "tablet" became more important concepts in the context of "nokia lumia", with similar strength to "window phone". This type of justification can also be provided for the other two errors when they exceed thresholds.

IX. CASE STUDY - GOAL OR PENALTY: DETECTING KEY EVENTS IN FIFA WORLD CUP QATAR 2022 FINAL MATCH (ARGENTINA VS FRANCE)

This section presents a practical application of the Self-Evolving Contextual Analysis (SECA) method within a specific scenario. The technique is employed to examine Twitter data collected during the FIFA World Cup 2022 Final Match, with the objective of identifying key events. The primary goal is to demonstrate the value of the SECA method in real-world situations.

The FIFA World Cup 2022 was the first football World Cup tournament held in the Middle East from November 21 to December 18, 2022. The competition was hosted across five cities in Qatar, with 32 participating teams from around the world. It was concluded at the Lusail Stadium in Doha,

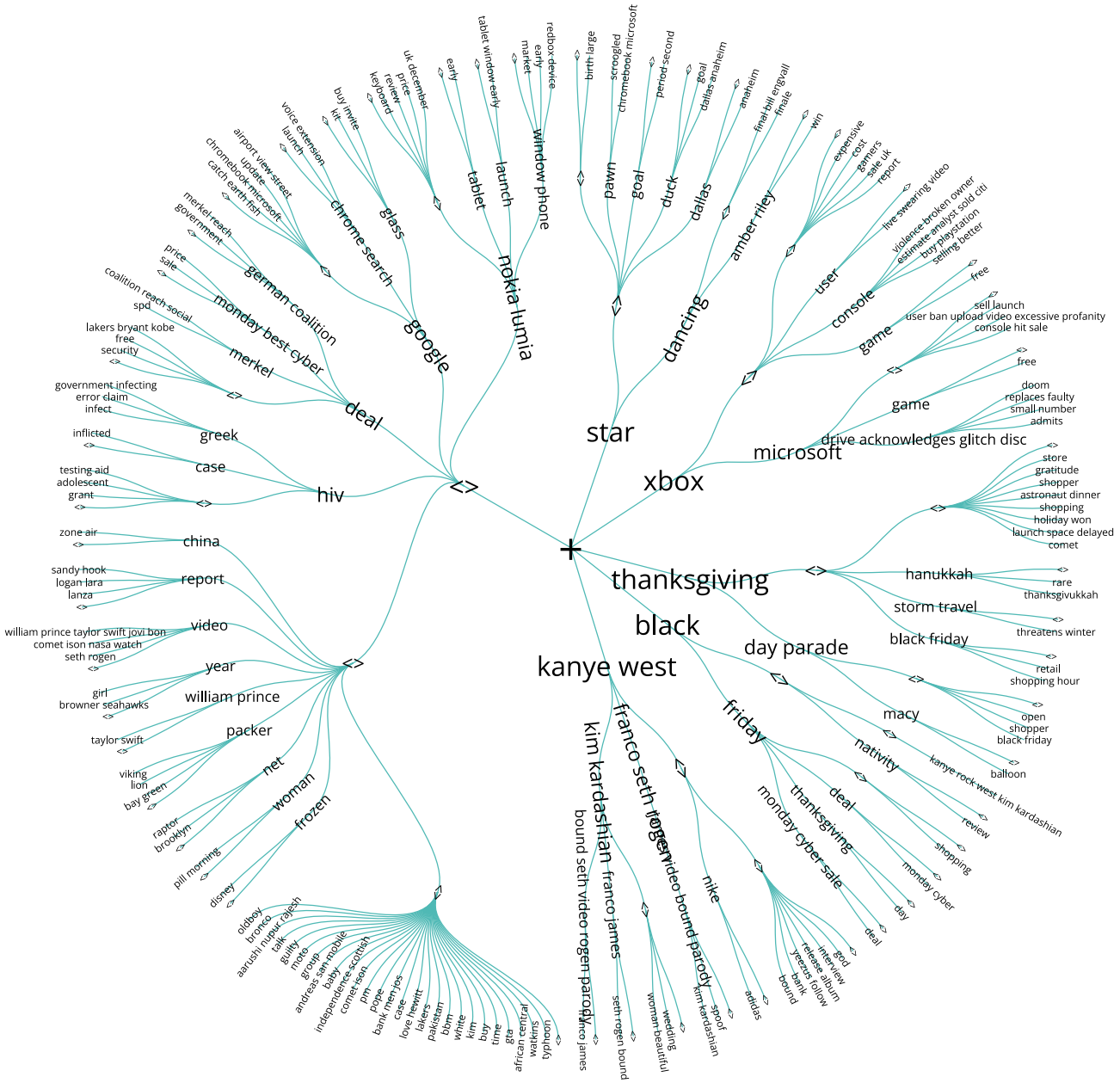

FIGURE 7. Tree representation of four levels for the topics and subtopics after feeding SECA with three batches from Theme G. The symbol "+" denotes the tree's root. "star", "xbox", "thanksgiving", "black" and "kanye west" are the Seed-Nodes that belong to particular topics. The symbol "<>" denotes a Refuge-Node. The nodes "nokia lumia", "deal", "net" are examples for Orphan-Nodes, that can also represent a seed for a particular topic but not as significant as the topics in the Seed-Nodes. There are two direct subtopics, "kim kardashian" and "bound seth video rogen parody" under the Seed-Node "kanye west". Note: this presentation was built using a source code available at https://vizhub.com/curran/.

where the final match was hosted between Argentina and France, and ended with Argentina's victory. The final match presents a good case study for applying the SECA approach for real-time tweet analysis, as it offers a huge amount of data generated by fans during a globally significant event.

A. DATA COLLECTION AND PRE-PROCESSING

Data were directly collected from the Twitter Application Programming Interface (API) during the event. Around 2.3 million tweets were downloaded over a period of three hours, starting from 15:00 GMT to 17:57 GMT on December 18, 2022. Tweets that only contain one of the terms: "Lionel", "#ARGFRA", "#WorldCupFinal", "#FRAARG", "#Mbappe", "#LionelMessi", "#Messi", "#LeoMessi", "#ArgentinavsFrance", "#FranceVsArgentina", "Argentina", "France", "#FIFAWorldCupFinal", were fetched. Moreover, a straightforward spam-filtering method was employed, relying on a list of spam words. This list was compiled through a manual examination of posts collected during earlier matches of the tournament. For example, tweets that contain words, such as "livestream", "streaming", "live", etc. were filtered. Additionally, retweets and non-English

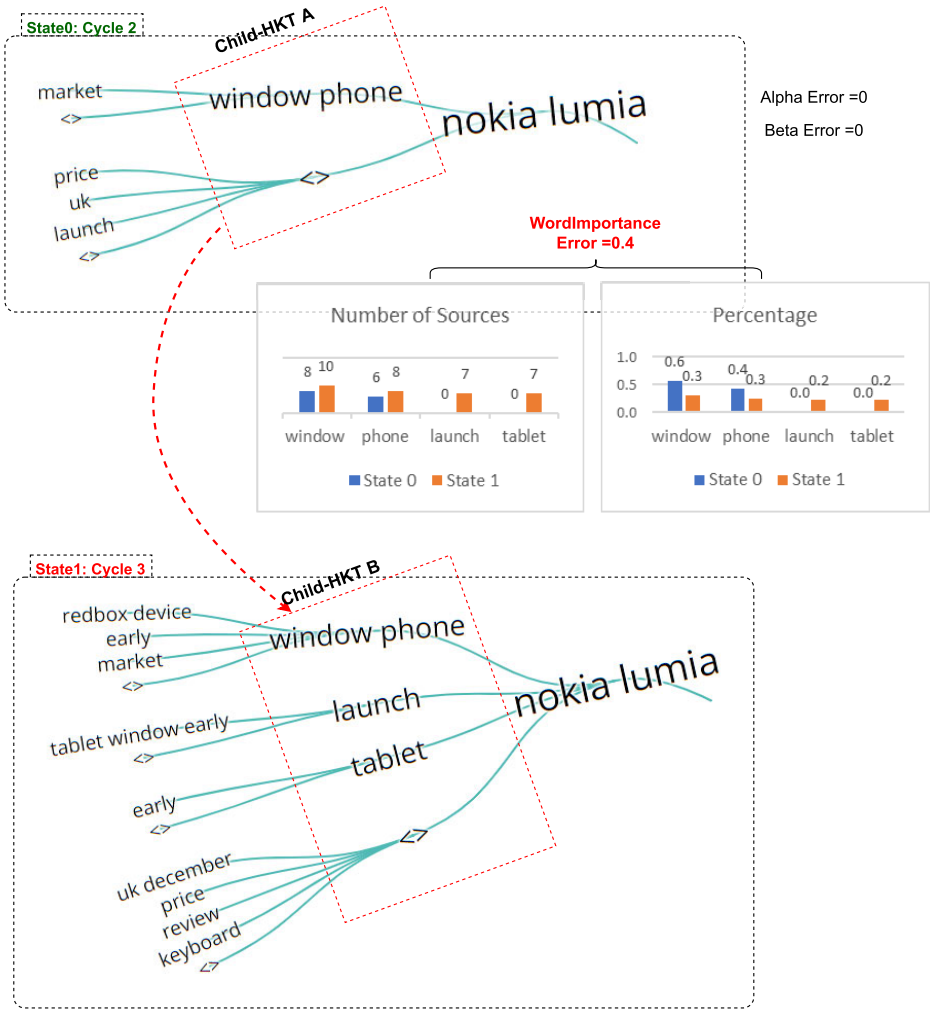

FIGURE 8. Example of the change on the Child-HKT of the node "nokia lumia" after mapping new sources to the node in cycle 3. The main sub-topic under "nokia lumia" of "window" and "phone" remains but further eligible words, i.e. "launch" and "tablet", were identified as having received more sources and so should be promoted to this level in the HKT. Note: although the word "launch" appeared as a Child-Node in cycle 2, it is counted as zero sources in the context of the targeted HKT container: {["window" "phone"], [<>]}, since the focus is to measure the changes on this container and the word regarded as a new word at this level in cycle 3.

posts were excluded. The data collection process yielded a comprehensive dataset comprising of 800 thousand tweets, providing a foundation for subsequent analyses. For the pre-processing phase, the steps described in the previous chapter were followed.

For the ground truth, key highlights from the match were selected from four different live text sources, including BBC [59], Al Jazeera Media Network [60], Sky Sports [61] and Eurosport [62]. A sample of these highlights as well as the number of crawled tweets during the match, is shown in Figure 9.

B. IMPLEMENTATION

The SECA software, which was introduced in the previous section, was used in this scenario with the following configuration: $\alpha = 0.7$, $\beta = 0.5$, AlphaError $= 0.1$, BetaError $= 0.2$, and WordImportanceError $= 0.3$. The SECA light option was enabled, and the program was set to fetch tweets every minute and store up to 3 batches in memory.

While the previous section showed that the SECA approach generates a hierarchical keyword-based summary of topics, enhancing the readability of the generated trees by producing human-like summaries of the encapsulated sources within nodes would be useful (note: the creation of this type of summary is beyond the scope of this paper). For this purpose, ChatGPT API [63] was selected for this task. This choice is based on our observations on the quality of the produced outputs. To generate a summary of each node in the tree, a random sample of 100 tweets that were clustered within each node were fed into the ChatGPT with the prompt: "Describe, as reporting live a current event in

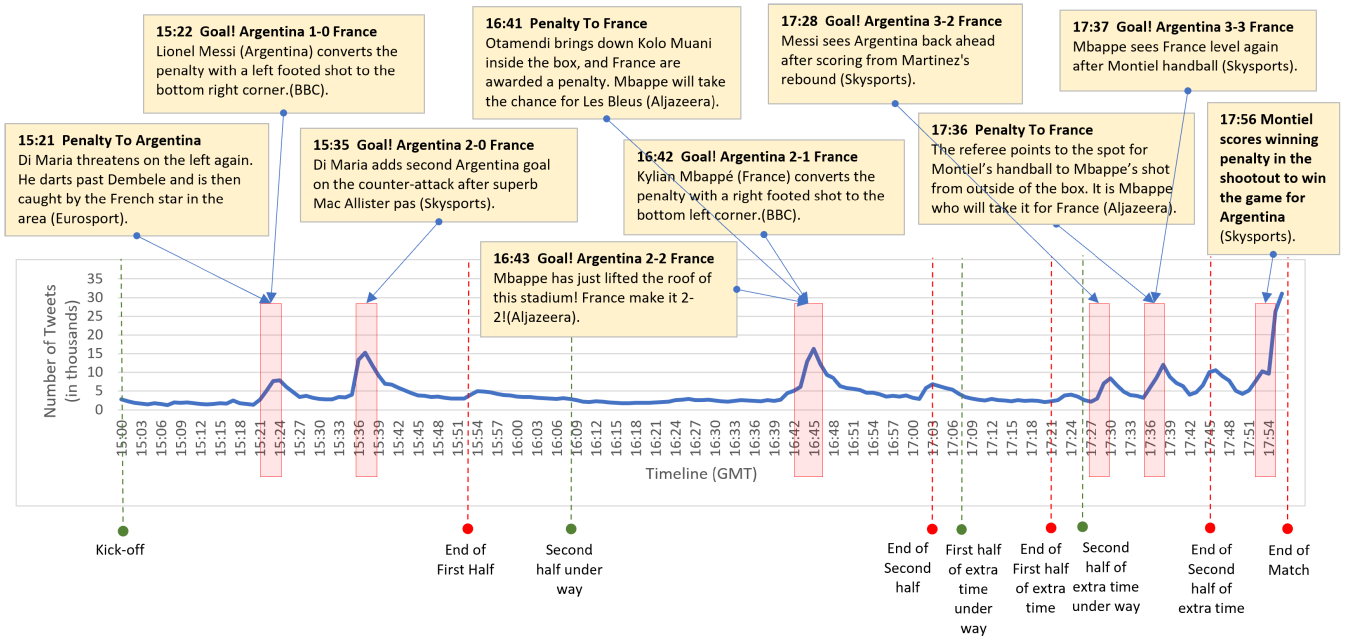

FIGURE 9. A timeline of the number of tweets downloaded from Twitter API during the FIFA World Cup 2022 Final Match (Argentina vs France). The yellow boxes represent selected posts of key events from live bogs published in media websites, such as BBC [59], Al Jazeera Media Network [60], Sky Sports [61] and Eurosport [62].

TABLE 5. Randomly selected tweets from four distinct nodes.

Node	Time	Tweet's Content
goal -> team	15:36:04	Unbelievable team goal that from Argentina
	15:36:09	What a beautiful team goal by Argentina #WorldCupFinal
	15:36:35	Argentina with the best team goal of the tournament!
	15:36:43	ARGENTINA. What an absolute team goal that was. Great finish!
	15:36:56	My word is Argentina bossing this! What a team goal!
goal -> di maria	15:36:11	Argentina looking like they want it more. Great goal Di Maria
	15:36:44	Di Maria deserves a goal.'Brilliant! #ArgentinaVsFrance
	15:36:48	Di Maria scores the second goal for Argentina.
	15:36:51	Maria !! Amazing teamwork that started from the back and led to a goal
	15:36:59	Fantastic goal by Argentina, Di Maria very cool finish! #FIFAWorldCupFinal #ArgentinaVsFrance
goal -> beautiful	15:36:50	Argentina 2nd goal was beautiful
	15:36:37	Wow that was beautiful . . . what a goal #ArgentinaVsFrance:
	15:36:42	Argentina came out to play. That goal was beautiful. What a performance by them so far
	15:36:51	Now that was a beautiful and well deserved goal! France is done.
	15:36:50	Beautiful goal by Argentina to be fair
goal -> second	15:36:12	Argentina just scored a second goal now.
	15:36:38	Argentina second goal is pure class omg . Beautiful futbol
	15:36:39	Argentina gets their second goal in the World Cup Final Qatar2022
	15:36:39	Second goal!!! Great goal from Argentina
	15:36:44	France in deep trouble what a second goal

max three lines, what is happening right now using the following tweets, with a focus on the terms *nodeTerms*", where *nodeTerms* are the encapsulated words in the node. The selection was restricted to 100 tweets because of the existing constraints on ChatGPT's prompt length capacity. The results were manually investigated and compared with the live blog comments from the four media sources mentioned above.

C. RESULTS AND DISCUSSION

In order to provide a detailed analysis of the topics identified by SECA during the match, two key events

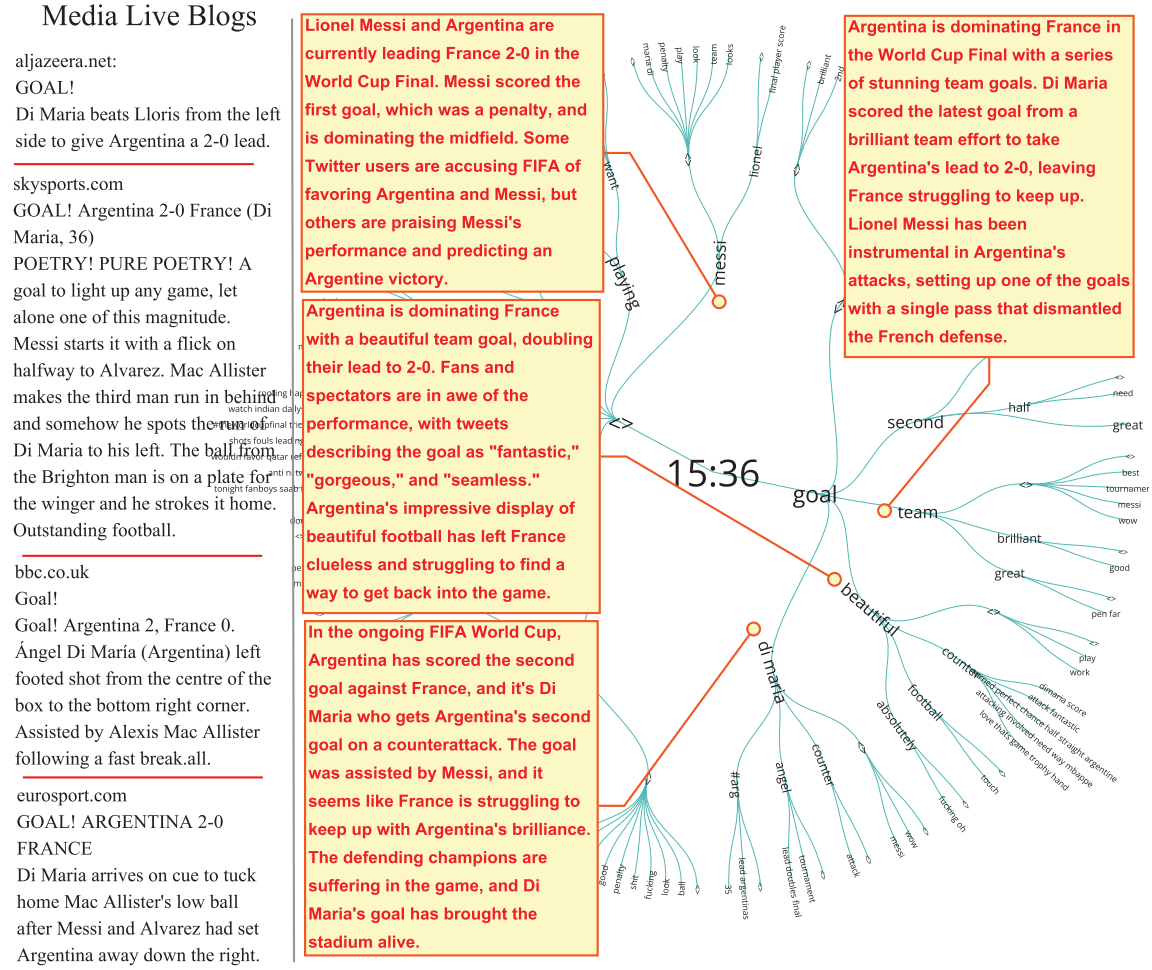

FIGURE 10. Tree representation of four levels for the topics and sub-topics after feeding SECA with three minutes batches, from 15:34 to 15:36. The timestamp "15:36" denotes the tree's root. "goal" is the Seed-Node that belongs to a particular topic. The symbol "<>" denotes a Refuge-Node. The nodes "messi", "playing" are examples of Orphan-Nodes, that can also represent a seed for a particular topic but are not as significant as the topics in the Seed-Node. There are four direct sub-topics, "di maria", "beautiful", "team" and "second" under the Seed-Node "goal". The left part of the figure shows the published live texts around 15:36 from four different sources, including BBC, Al Jazeera Media Network, Sky Sports and Eurosport. The yellow boxes represent the summary generated using ChatGPT.

are selected for illustration. Figures 10 and 11 depict two distinct trees generated by SECA at two different timestamps, 15:36 GMT and 17:36 GMT, respectively. The trees are accompanied by a summary of some nodes generated using ChatGPT. Additionally, the figures illustrate the comments posted on the live blogs during those times.

At 15:36, the 36 minute of the match, Dia Maria scored the second goal for Argentina. According to Skysport Live Blog, the move started with Messi's flick to Alvarez near the halfway line, followed by Mac Allister, who spotted Di Maria's run on the left and delivered a pass to him; the latter finished the play by sending the ball into the net. SECA captured this significant event, which is mainly represented by the Seed-Node "goal", with its four important Child-Nodes, namely: "di maria", "beautiful", "team" and "second" (see Table 5 for randomly selected tweets from these nodes). These sub-concepts and their branches provide some details about the main themes of discussions on Twitter during the event. For example, by navigating through the Child-Nodes of the word "team", additional insights on how the users described the scored goal could be obtained (such as "brilliant" and "great").

Another important event during the match, specifically at 17:36, the Argentina player, Gonzalo Montiel, received a yellow card for committing a handball (according to BBC live blog). A penalty was awarded to France, and Mbappe was set to take the penalty kick for France (according to Aljazeera live blog). SECA captured this event (see Figure 11) by generating the Seed-Node "penalty". The words "handball" and "kick" are among the significant terms that were detected for the event, which can be directly linked to the information provided by the media. In other words, the contextual path "penalty->handball->montiel" and "penalty->kick->mbappe" can be linked to the transcripts in the BBC and the Aljazeera websites, respectively.

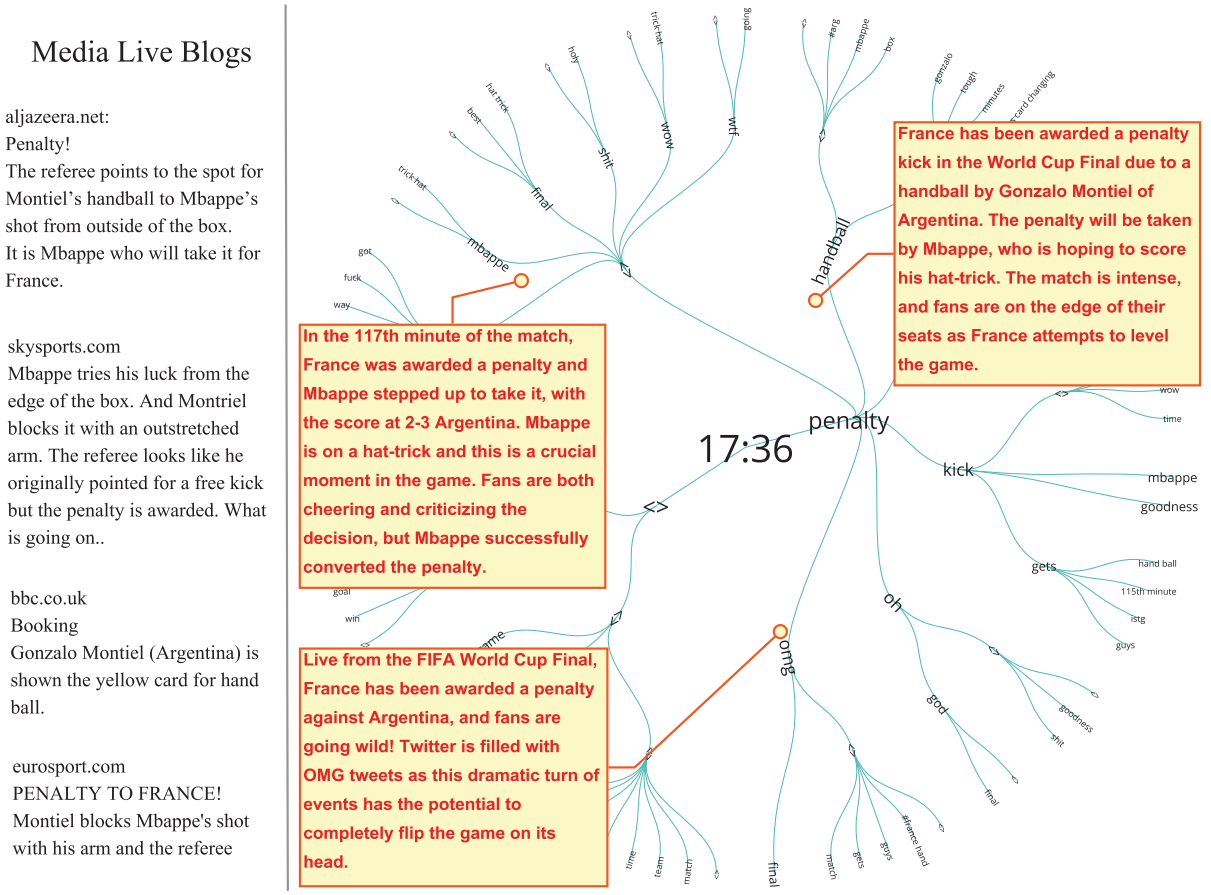

FIGURE 11. Tree representation of four levels for the topics and sub-topics after feeding SECA with three minutes batches, from 17:34 to 17:36. The timestamp "17:36" denotes the tree's root. "penalty" is the Seed-Node that belongs to a particular topic. The symbol "<>" denotes a Refuge-Node. The nodes "mbappe", "final" are examples of Orphan-Nodes, that can also represent a seed for a particular topic but are not as significant as the topics in the Seed-Node. There are five direct sub-topics, "handball", "game", "kick", "oh" and "omg" under the Seed-Node "penalty". The left part of the figure shows the published live texts around 17:36 from four different sources, including BBC, Al Jazeera Media Network, Sky Sports and Eurosport. The yellow boxes represent the summary generated using ChatGPT.

By utilizing ChatGPT's services, a more human-like summary of the generated nodes can be produced. This can offer the flexibility to direct the generated summary based on any selected scope node, whether from a higher-level concept, or from a more specific sub-concept, or from other interesting nodes like trending nodes (note: here, the trending nodes simply refer to the nodes that encapsulate a high number of sources in any specific timestamp). To illustrate, the summary generated in Figures 10 and 11 (in yellow boxes) focused on specific nodes, such as: "team", "beautiful", "di maria" and "messi" in Figure 10 and "handball", "omg" and "mbappe" in Figure 11.

X. CONCLUSION AND FUTURE WORKS

In this study, we present a new approach called Self-Evolving Contextual Analysis (SECA) for event detection. It creates a tree of word relationships that can dynamically change its structure based on input data in an automated manner. A computationally lighter version of this method was proposed to cope with fast-changing environments. We assessed SECA and the other three state-of-the-art baseline methods (KMeans, Nonnegative Matrix Factorization and MStream) with a focus on three important qualities: Performance, Transparency, and Carbon Footprint. We created events with varying frequency based on well-established datasets in literature, in order to determine the ability of the methods to these changing frequencies. We found that the proposed approach achieved highly competitive performance and overall gave the best results for all of the measures used. Transparency and Carbon Footprint assessments have also shown that SECA is the only method that meets all of the transparency requirements whilst at the same time being an environmentally friendly approach. In this regard, SECA represents a significant development for real-time monitoring of textual data for the detection of new events in an explainable manner whilst meeting the ethical requirements for Green AI.

In addition, the proposed SECA method was employed to analyze tweets published during the FIFA World Cup final match, which took place on December 18 2022. Two

key events were selected to demonstrate the ability of the approach to detect topics and present related information. The comparison analysis between the outputs of the method and posts published in media live blogs during the match reveals the usefulness of the approach in capturing the related words and their relationships. Integrating SECA outputs with ChatGPT API was utilized to offer a more readable summary of nodes from various scopes of interest.

Although the performance assessments showed very good performance of SECA in detecting the main topics, the quality of the hierarchal sub-concepts detection was not quantitatively assessed. To the best of our knowledge, there is no ground-truth dataset that can provide this type of detail in the context of event detection, and this is also a weakness in the datasets presented in this study. Our future work will focus on building a baseline corpus that can offer a hierarchical representation of topics and their sub-topics. In addition, applying the SECA method to other real-world applications, such as event detection during earthquakes and other significant occurrences, is also reserved for future work. Although ChatGPT provided interesting summaries of the generated nodes by SECA, further work is required to quantitatively assess the quality of the produced summary. Moreover, due to its nature, this approach is regarded as a black-box engine. This study focused on the transparency aspects of the event detection problem. Future work needs to focus on how to produce a more readable node summary without sacrificing the transparency aspects of the complete system.

REFERENCES

[1] W. Alhindi, M. Talha, and G. Sulong, "The role of modern technology in Arab spring," *Arch. Des Sci.*, vol. 65, no. 8, pp. 101–112, Aug. 2012.

[2] J. Eisenstein, "What to do about bad language on the internet," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, Jun. 2013, pp. 359–369. [Online]. Available: https://aclanthology.org/N13-1037

[3] S. Maity, A. Chaudhary, S. Kumar, A. Mukherjee, C. Sarda, A. Patil, and A. Mondal, "WASSUP? LOL: Characterizing out-of-vocabulary words in Twitter," in *Proc. 19th ACM Conf. Comput. Supported Cooperat. Work Social Comput. Companion*, Feb. 2016, pp. 341–344, doi: 10.1145/2818052.2869110.

[4] O. Ozdikis, P. Karagoz, and H. Oguztüzün, "Incremental clustering with vector expansion for online event detection in microblogs," *Social Netw. Anal. Mining*, vol. 7, no. 1, pp. 1–17, Nov. 2017, doi: 10.1007/s13278-017-0476-8.

[5] S. A. Sulaimani and A. Starkey, "Towards a transparent and an environmental-friendly approach for short text topic detection : A review of methods for performance, transparency and carbon footprint," *TechRxiv*, Mar. 2023, doi: 10.36227/techrxiv.22241071.

[6] A. Abdul Aziz and A. Starkey, "Predicting supervise machine learning performances for sentiment analysis using contextual-based approaches," *IEEE Access*, vol. 8, pp. 17722–17733, 2020.

[7] S. Petrovic, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to Twitter," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*. Los Angeles, CA, USA: Association for Computational Linguistics, 2010, pp. 181–189.

[8] H. Sayyadi, M. Hurst, and A. Maykov, "Event detection and tracking in social streams," in *Proc. Int. AAAI Conf. Web Soc. Media*, Mar. 2009, vol. 3, no. 1, pp. 311–314, doi: 10.1609/icwsm.v3i1.13970.

[9] H. Sayyadi and L. Raschid, "A graph analytical approach for topic detection," *ACM Trans. Internet Technol.*, vol. 13, no. 2, pp. 1–23, Dec. 2013, doi: 10.1145/2542214.2542215.

[10] H. Abdelhaq, C. Sengstock, and M. Gertz, "EvenTweet: Online localized event detection from Twitter," *Proc. VLDB Endowment*, vol. 6, no. 12, pp. 1326–1329, Aug. 2013, doi: 10.14778/2536274.2536307.

[11] I. Afyouni, Z. A. Aghbari, and R. A. Razack, "Multi-feature, multi-modal, and multi-source social event detection: A comprehensive survey," *Inf. Fusion*, vol. 79, pp. 279–308, Mar. 2022, doi: 10.1016/j.inffus.2021.10.013.

[12] Q. Li, A. Nourbakhsh, S. Shah, and X. Liu, "Real-time novel event detection from social media," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 1129–1139, doi: 10.1109/ICDE.2017.157.

[13] M. Hasan, M. A. Orgun, and R. Schwitter, "Real-time event detection from the Twitter data stream using the TwitterNews+ framework," *Inf. Process. Manag.*, vol. 56, no. 3, pp. 1146–1165, May 2019, doi: 10.1016/j.ipm.2018.03.001.

[14] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," 2022, *arXiv:2203.05794*.

[15] Collins English dictionary. (2023). *Definition of 'Event'*. Accessed: Feb. 9, 2023. [Online]. Available: https://www.collinsdictionary.com/dictionary/english/event

[16] Oxford English Dictionary Online. (2022). *Definition of 'Event'*. [Online]. Available: www.oed.com/view/Entry/65287

[17] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on Twitter," in *Proc. Int. AAAI Conf. Web Social Media*, 2011, vol. 5, no. 1, pp. 438–441.

[18] W. Dou, X. Wang, W. Ribarsky, and M. Zhou, "Event detection in social media data," in *Proc. IEEE VisWeek Work. Interact. Vis. Text Anal. Driven Anal. Soc. Media Content*, Oct. 2012, pp. 971–980.

[19] Z. Saeed, R. A. Abbasi, O. Maqbool, A. Sadaf, I. Razzak, A. Daud, N. R. Aljohani, and G. Xu, "What's happening around the world? A survey and framework on event detection techniques on Twitter," *J. Grid Comput.*, vol. 17, no. 2, pp. 279–312, Jun. 2019, doi: 10.1007/s10723-019-09482-2.

[20] Twitter. (2022). *Counting Characters*. Accessed: Jun. 24, 2022. [Online]. Available: https://developer.twitter.com/en/docs/counting-characters

[21] A. Rosen. (2017). *Tweeting Made Easier*. Accessed: Feb. 10, 2023. [Online]. Available: https://blog.twitter.com/en_us/topics/product/2017/tweetingmadeeasier

[22] Internet Live Stats. *Twitter Usage Statistics*. Accessed: Mar. 29, 2020. [Online]. Available: http://www.internetlivestats.com/twitter-statistics/

[23] Omnicore Agency. (2020). *Twitter by the Numbers: Stats, Demographics & Fun Facts*. Accessed: Dec. 2, 2020. [Online]. Available: https://www.omnicoreagency.com/twitter-statistics/

[24] E. Musk. (2022). *24,400 Tweets Per Second for France's Goal, Highest Ever for World Cup!* Accessed: Feb. 13, 2023. [Online]. Available: https://twitter.com/elonmusk/status/1604520708570517504?lang=en

[25] A. Fichera. (2020). *Viral Social Media Posts Offer False Coronavirus Tips*. Accessed: Feb. 13, 2023. [Online]. Available: https://www.factcheck.org/2020/03/viral-social-media-posts-offer-false-coronavirus-tips/

[26] C. Shao, G. L. Ciampaglia, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "The spread of low-credibility content by social bots," *Nature Commun.*, vol. 9, no. 1, p. 4787, Nov. 2018, doi: 10.1038/s41467-018-06930-7.

[27] Thomson Reuters. (2023). *Fact Check-Old Video of Indonesia Tsunami Shared as If From Turkey Earthquake*. Accessed: Feb. 13, 2023. [Online]. Available: https://www.reuters.com/article/factcheck-turkey-quake-idUSL1N34N1LX

[28] Collins English dictionary. (2023). *Definition of 'Spam'*. Accessed: Feb. 14, 2023. [Online]. Available: https://www.collinsdictionary.com/dictionary/english/spam

[29] Statista. (2023). *Actioned Spam Content Items on Facebook Worldwide From 4th Quarter 2017 to 3rd Quarter 2022*. [Online]. Available: https://www.statista.com/statistics/1013843/facebook-spam-content-removal-quarter/

[30] X. Zheng, Z. Zeng, Z. Chen, Y. Yu, and C. Rong, "Detecting spammers on social networks," *Neurocomputing*, vol. 159, pp. 27–34, Jul. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231215002106

[31] M. Asif, D. Zhiyong, A. Iram, and M. Nisar, "Linguistic analysis of neologism related to coronavirus (COVID-19)," *Social Sci. Humanities Open*, vol. 4, no. 1, 2021, Art. no. 100201. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590291121000978

[32] T. Alshaabi, D. R. Dewhurst, J. R. Minot, M. V. Arnold, J. L. Adams, C. M. Danforth, and P. S. Dodds, "The growing amplification of social media: Measuring temporal and social contagion dynamics for over 150 languages on Twitter for 2009–2020," *EPJ Data Sci.*, vol. 10, no. 1, p. 15, Mar. 2021, doi: 10.1140/epjds/s13688-021-00271-0.

[33] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3645–3650, doi: 10.18653/v1/P19-1355.

[34] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, "Quantifying the carbon emissions of machine learning," 2019, *arXiv:1910.09700*.

[35] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," *J. Mach. Learn. Res.*, vol. 21, no. 248, pp. 1–43, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-312.html

[36] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," in *Proc. AAAI*, Apr. 2020, vol. 34, no. 9, pp. 1393–13696, doi: 10.1609/aaai.v34i09.7123.

[37] L. F. Wolff Anthony, B. Kanding, and R. Selvan, "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models," 2020, *arXiv:2007.03051*.

[38] M. Yusuf, P. Surana, G. Gupta, and K. Ramesh, "Curb your carbon emissions: Benchmarking carbon emissions in machine translation," 2021, *arXiv:2109.12584*.

[39] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, and Others, "On the opportunities and risks of foundation models," 2021, *arXiv:2108.07258*.

[40] UK Government—Department for Environment and E Agency. (2014). *Calculate the Carbon Dioxide Equivalent Quantity of an F Gas*. Accessed: Jun. 23, 2022. [Online]. Available: https://www.gov.uk/guidance/calculate-the-carbon-dioxide-equivalent-quantity-of-an-f-gas

[41] L. Lannelongue, J. Grealey, and M. Inouye, "Green algorithms: Quantifying the carbon footprint of computation," *Adv. Sci.*, vol. 8, no. 12, Jun. 2021, Art. no. 2100707, doi: 10.1002/advs.202100707.

[42] Mila, BCG GAMMA, Haverford College, and Comet. (2021). *CodeCarbon*. Accessed: Apr. 15, 2022. [Online]. Available: https://codecarbon.io/

[43] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018, doi: 10.1109/ACCESS.2018.2870052.

[44] Z. C. Lipton, "The mythos of model interpretability," *Commun. ACM*, vol. 61, no. 10, pp. 36–43, Sep. 2018, doi: 10.1145/3233231.

[45] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.

[46] S. A. Sulaimani and A. Starkey, "Short text classification using contextual analysis," *IEEE Access*, vol. 9, pp. 149619–149629, 2021.

[47] A. A. Aziz, "Contextual-based approach for sentiment analysis," Ph.D. dissertation, School Eng., Univ. Aberdeen, Aberdeen, Scotland, 2020.

[48] J. Yin, D. Chao, Z. Liu, W. Zhang, X. Yu, and J. Wang, "Model-based clustering of short text streams," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Jul. 2018, pp. 2634–2642, doi: 10.1145/3219819.3220094.

[49] The National Institute of Standards and Technology (NIST)—US Department of Commerce. (2016). *2015 Microblog Track*. Accessed: Jan. 1, 2022. [Online]. Available: https://trec.nist.gov/data/microblog2015.html

[50] J. Yin and J. Wang, "A Dirichlet multinomial mixture model-based approach for short text clustering," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2014, pp. 233–242, doi: 10.1145/2623330.2623715.

[51] Jackyin12. (2022). *MStream*. Accessed: Oct. 21, 2022. [Online]. Available: https://github.com/jackyin12/MStream

[52] P. Lee, L. V. S. Lakshmanan, and E. E. Milios, "Incremental cluster evolution tracking from highly dynamic network data," in *Proc. IEEE 30th Int. Conf. Data Eng.*, Chicago, IL, USA, Mar. 2014, pp. 3–14.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2012. [Online]. Available: https://scikit-learn.org

[54] C. Zhou and Q. Zhao, "Efficient time series clustering and its application to social network mining," *J. Intell. Syst.*, vol. 23, no. 2, pp. 213–229, Jun. 2014, doi: 10.1515/jisys-2014-0005.

[55] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, vol. 8, New Orleans, LA, USA, Jan. 2007, pp. 1027–1035.

[56] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[57] C. C. Aggarwal, *Text Sequence Modeling and Deep Learning*. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-73531-3_10.

[58] T. Mori and N. Uchihira, "Balancing the trade-off between accuracy and interpretability in software defect prediction," *Empirical Softw. Eng.*, vol. 24, no. 2, pp. 779–825, Apr. 2019, doi: 10.1007/s10664-018-9638-1.

[59] P. McNulty. (2022). *World Cup final: Argentina Beat France on Penalties in Dramatic Qatar Showpiece—BBC Sport*. Accessed: Feb. 15, 2023. [Online]. Available: https://www.bbc.co.uk/sport/football/63932622

[60] R. Sharma, D. Child, S. Arunk, and J. Brownsell. (2022). *Argentina vs France: World Cup 2022 Final—As it Happened*. Accessed: Feb. 15, 2023. [Online]. Available: https://www.aljazeera.com/sports/liveblog/2022/12/18/live-argentina-vs-france-world-cup-2022-final

[61] L. Jones and J. Shread. (2022). *Qatar World Cup final 2022: Argentina vs France Live Commentary*. Accessed: Feb. 15, 2023. [Online]. Available: https://www.skysports.com/football/live-blog/12309/12768549/qatar-world-cup-final-2022-argentina-vs-france-live-commentary

[62] P. Hassall. (2022). *FIFA World Cup 2022 Final in Qatar Result: Kylian Mbappe's Hat-Trick not Enough for France as Lionel Messi Leads Argentina to Penalty Shoot-Out Win*. Accessed: Feb. 15, 2023. [Online]. Available: https://www.eurosport.com/football/world-cup/2022/live-argentina-france_mtc1287454/live-commentary.shtml

[63] OpenAI. (2023). *ChatGPT*. Accessed: Mar. 29, 2023. [Online]. Available: https://chat.openai.com/

SAMI AL SULAIMANI received the B.S. degree in computer science from Sultan Qaboos University, Oman, in 2005, and the M.S. degree in advanced software engineering from the University of Leicester, U.K., in 2014. He is currently pursuing the Ph.D. degree in computing science with the University of Aberdeen, U.K. From 2006 to 2019, he was a Software Engineer and a System Analyst.

ANDREW STARKEY is currently a Reader with the University of Aberdeen. He is also responsible for Blueflow Ltd., a spin-out company from the University of Aberdeen that proposed a solution for a wide range of data analysis areas, such as financial, textual, and web data, such as blogs and discussion threads and condition monitoring. His current research interests include explainable AI, automated AI, and autonomous learning methods. He has been awarded an Enterprise Fellowship from the Royal Society of Edinburgh and Scottish Enterprise.