



(51) International Patent Classification:

G06F 16/35 (2019.01) G06Q 30/02 (2012.01)
G06F 40/30 (2020.01) G06Q 30/06 (2012.01)
G06Q 30/00 (2012.01) G06Q 50/00 (2012.01)

(21) International Application Number:

PCT/EP2022/051831

(22) International Filing Date:

27 January 2022 (27.01.2022)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

21155279.9 04 February 2021 (04.02.2021) EP

(71) Applicants: IMEC VZW [BE/BE]; Kapeldreef 75, 3001 Leuven (BE). VRIJE UNIVERSITEIT BRUSSEL [BE/BE]; Pleinlaan 2, 1050 Brussel (BE).

(72) Inventors: DELIGIANNIS, Nikolaos; Kapeldreef 75, 3001 Leuven (BE). HUU DO, Tien; Kapeldreef 75, 3001 Leuven (BE). BERNEMAN, Marc; Kapeldreef 75, 3001 Leuven (BE). VANDEN BROUCKE, Steven; Kapeldreef 75, 3001 Leuven (BE). CERNEJ, Vitalijus; Kapeldreef 75, 3001 Leuven (BE).

(74) Agent: PLAS, Axel et al.; IP Hills NV, Hubert Frère-Orbanlaan 329, 9000 Gent (BE).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,

(54) Title: A DISINFORMATION DETECTION SYSTEM

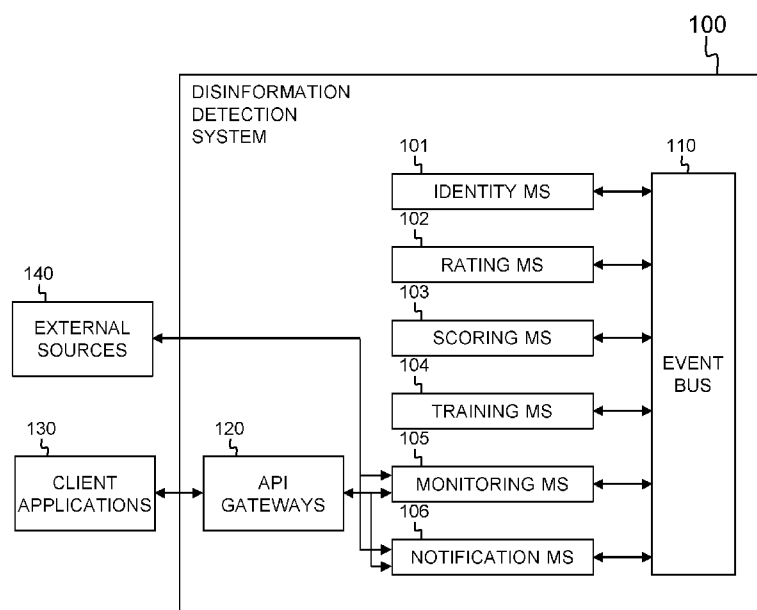


Fig. 1

(57) Abstract: A disinformation detection system (100) for automated verification of information items comprises an event bus (110) and event-driven microservices (101-106). The event-driven microservices (101-106) comprise: - a scoring microservice (103; 300) configured to execute at least one trained machine learning model (301, 302, 303, 305) adapted to generate a disinformation prediction for each information item; - a training microservice (104) configured to train the at least one machine learning model (301, 302, 303, 305) based on input obtained from researchers; and - a monitoring microservice (105; 200) configured to obtain the information items and related data, and forward them to the scoring microservice (103; 300). The monitoring microservice (105; 200) comprises: - a data storage (240); - at least one background harvester (235, 236, 237) configured to periodically fetch and store information items and/or related data from a particular information source; and - at least one on-demand harvester (221, 222) configured to fetch and store an

HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

information item and/or related data in return to a URL or other type of query of the information item.

A DISINFORMATION DETECTION SYSTEM

Field of the Invention

5

[01] The present invention generally relates to a disinformation detection system for fast and automatic detection of disinformation or fake news on a large scale.

Background of the Invention

10

[02] Disinformation or fake news is intentionally produced information, rumoured or fictitiously created, and spread rapidly in order to disinform or mislead the readers of such information. The modern civilization, equipped with mobile devices that allow
15 information consumption anytime and anyplace, has become an easy target for disinformation. The generation and spread of fake news became a business for organisations like political parties, spin doctors, marketeers and influencers that take benefit of biasing the public opinion. Fake news has made the society prone to political influences and damages to organisations like democratic institutions as it becomes
20 difficult or even impossible for information consumers to discriminate subjective opinions, rumours and intentional false information from verified legitimate facts.

[03] As a consequence of the fast-growing amount of information consumption devices, information items like news articles and stories are real-time disseminated
25 and observed without verification by information consumers. The spreaders of disinformation exploit the consumers' desire for fast news consumption and the ignorance of consumers to validate the information they absorb to rapidly spread rumours or fake news. Disinformation is not subject to fact checking or other ethical standards applied by journalistic redactions that strive to inform their readers with
30 objective information the legitimacy of which is verified. Fake news therefore seems to benefit from the perverse effect it better satisfies the overall desire of consumers to first receive and further spread news. As a result of the rapid and massive spread of disinformation, there is an increasing demand for scalable technology able to

automatically monitor the distribution of information and able to act with close to real-time performance on the spread of fake news.

5 [04] Social media networks are naturally growing graph structures that allow for the rapid dissemination of information. Although social media networks are not designed for the specific purpose of spreading fake news, it has been demonstrated that social media networks allow for a much faster spread of information when compared to traditional media like TV or printed newspapers. Despite the fact that social media networks facilitate the rapid spread of disinformation, they often also contain the
10 information that can be exploited to detect and combat fake news. Indeed, textual content of information items on its own is often insufficient to verify the legitimacy of the information item. Auxiliary information that extends beyond textual data, such as the relation between nodes and social network graphs, temporal data, geographic information, author or distributor parameters like age, credibility, etc., may be crucial
15 in detecting and combatting disinformation. There is consequently a growing demand for a scalable technological system able to exploit a variety of parameters that extend beyond the mere textual content of the information item itself in order to detect and combat the spread of disinformation with real-time or near real-time speed.

20 [05] It is an object of the present invention to disclose a disinformation detection system for automated detection of disinformation that provides an answer to the above-defined demands. More particularly, it is an object of the present invention to disclose a disinformation detection system that is scalable, able to act with close to real-time performance, and that exploits auxiliary information that extends beyond the mere
25 textual data used within existing models.

Summary of the Invention

30 [06] According to an aspect of the invention, the above-defined objective is realised by the disinformation detection system for automated verification of information items defined by claim 1, the disinformation detection system comprising an event bus and event-driven microservices coupled to the event bus, wherein the event-driven microservices comprise:

- a scoring microservice configured to execute at least one trained machine learning model adapted to generate a disinformation prediction for each information item forwarded thereto;

5 - a training microservice configured to train the at least one machine learning model based on input obtained from researchers; and

- a monitoring microservice configured to obtain the information items and related data for the information items, and configured to forward the information items and the related data to the scoring microservice, wherein the monitoring microservice comprises:

10 - a data storage;

- at least one background harvester configured to periodically fetch and store in the data storage information items from a particular information source and related data for the information items from the particular information source;

15 - at least one on-demand harvester configured to fetch and store in the data storage an information item and/or related data for the information item in return to a Uniform Resource Locator, abbreviated URL, or other type of query of the information item

wherein the at least one trained machine learning model comprise:

20 - a first Bi-directional Encoder Representations from Transformers model, abbreviated a first BERT model, trained for clickbait detection, configured to receive an information item and/or related data as input sequence and to return a probability for the information item and/or related data indicating clickbait;

25 - a second BERT model trained for sentiment detection, configured to receive an information item and/or related data as input sequence and to return a probability for the information item and/or related data indicating sentiment;

30 - a third BERT model trained for bias detection, configured to receive an information item and/or related data as input sequence and to return a probability for the information item and/or related data indicating bias; and

- a fourth BERT model trained for toxicity detection configured to receive an information item and/or related data as input sequence and to return a probability for the information item and/or related data indicating toxicity,

or, alternatively, wherein the at least one trained machine learning model (302, 302, 303) comprise:

- an N-athlon model (500), N being an integer plurality, the N-athlon model being configured to receive an information item and/or related data together with one out of N possible questions as input sequence, and to return an answer to said question and a start index (551) and end index (552) wherein the range [start index, end index] indicates where in the information item and/or related data the relevant part is found motivating said answer to said question.

[07] Thus, embodiments of the invention concern a disinformation detection system with microservice architecture that allows for scalable and rapid monitoring of information items. Microservices are a software architecture wherein applications are composed of small, independent modules that communicate with each other using well-defined API contracts. The microservice modules are highly decoupled building blocks that are small enough to each implement a single functionality. Thus, a microservice is a module that implements a specific functionality. A microservice typically has its own stack, including database and data model. The microservices are often connected via REST APIs, an event bus or message brokers. The rapid nature of information dissemination requires the disinformation detection system to correlate useful information as close as possible to real-time. Otherwise, when response times increase, the potential damages increase exponentially. Existing monolithic architectures of disinformation detection systems are easy to understand and initiate. However, when growing in size or functionality, these monolithic architectures become complex to understand, maintain and adjust. The monolithic architectures lack scalability and therefore slow down evolution of disinformation detection. The event-driven microservice architecture of the disinformation detection system according to the invention is capable of monitoring information items of as many information sources as possible and this is scalable to the number of sources it collects information items from to process. The event-driven microservice architecture with monitoring microservice to continuously collect information items and related data and scoring microservice to process the collected information items and related data allows to monitor news on large scale while being able to rapidly send notifications to information consumers once new insights are found in relation to published news. Although the

microservice-based architecture is initially more complex to design, it allows for development teams to focus on individual work packages without requiring a complete understanding of the working of the entire disinformation detection system, which finally results in significant cost savings, simplified maintenance and feature development, and enhanced scalability.

[08] Essential microservices in the disinformation detection system according to the invention are the scoring microservice, training microservice and monitoring microservice. The scoring microservice is configured to execute trained machine learning models that jointly generate predictions for information items. The predictions are then put on the event bus to be communicated to information consumers, to be used by the training microservice to improve the trained machine learning models, and/or to be stored so that they can be exploited to improve model predictions when no online training mechanism is feasible for a model. The independent scoring microservice brings the advantage that new machine learning models can be integrated easily without having to change or know anything about other components of the disinformation detection system. The training microservice is configured to allow researchers to monitor the output of the scoring machine learning models and to update or improve these scoring machine learning models automatically on the fly or in predetermined time intervals. The training microservice thus takes advantage of the prediction outputs of the scoring microservice for processed information items to improve the individual machine learning models. The monitoring microservice at last is configured to continuously monitor predefined sources of news and related data, and in addition is able to collect information items on-demand from information sources that are not continuously monitored. The collected information items and/or related data are stored in a data storage for future use and broadcasted through the event bus to be processed and scored for the presence of disinformation by the scoring microservice.

[09] The monitoring microservice is a data harvesting system that comprises one or plural background harvesters and one or plural on-demand harvesters. The main task of each harvester is to collect information items, e.g. news articles, to obtain related data for the collected information items, and to store the collected information items and related data in a specific data storage. A background harvester is configured to

periodically collect information items from a predetermined information source, like for instance labelled news articles from a fact-checking website like factcheck.org or unlabelled news articles from regular news websites like BBC, CNN, Reuters, Yahoo News, CNBC, The Guardian, The New York Times, etc., and stores the fetched information items in the specific data storage. The fetching of information items from a predetermined information source may be based for instance on keywords or topics where information consumers are interested in. A background harvester in addition collects related data for the fetched information items. The related data provides context information for a fetched information item and comprises for instance related posts collected from a social network platform like Twitter or Reddit, and/or related financial information, for example stock exchange values collected from financial information websites like Yahoo Finance for companies or organisations mentioned in the fetched information item. Multiple background harvesters operate in parallel such that all predetermined information sources are being crawled simultaneously within a predetermined time period. An on-demand harvester takes as input the Uniform Resource Locator, abbreviated URL, of an information item and fetches the content of that information item and/or related data for that information item like for instance cascades of related posts from a social network platform like Twitter. As an alternative to the URL, another type of query of the information item may be used as input by the on-demand harvester, like for example the content or parts of the content of the information item. Typically, an Application Programming Interface or API is implemented to access an on-demand harvester.

[10] Whereas disinformation detection is the ultimate target, this could be achieved by considering four different parameters: clickbait, sentiment, bias and toxicity. Clickbait corresponds to content whose main purpose is to attract attention and encourage users to click on a link to a particular webpage. Sentiment corresponds to content that refers to a feeling or emotion. Bias corresponds to content that refers to societal biases such as gender, race, etc. Toxicity corresponds to content that refers to offensive, hateful, abusive or unwelcome language, for instance in comments on social media. The detection of these four different parameters could be considered as four single tasks. In sample embodiments of the disinformation detection system according to the invention, four machine learning models may be integrated in the scoring microservice, each machine learning model being pre-trained with specific

datasets for a single task. The datasets for pre-training a single machine learning model are preferably obtained by sampling a larger dataset - either regular sampling or weighted sampling - such that all labels generated by the single machine learning model have an equal probability to be sampled.

5

[11] Thus, the disinformation detection system according to the invention comprise BERT machine learning models that are pre-trained to respectively detect clickbait, sentiment, bias and toxicity. The BERT models are transformers that have re-shaped the natural language processing domain. The BERT machine learning models can be pre-trained for general tasks on a big dataset. The pre-trained model can then be fine-tuned for specific tasks with much smaller datasets. It is noticed however that alternative embodiments of the disinformation detection system according to the invention may use other machine learning models like for instance Generative Pre-trained Transformer (abbreviated GPT) machine learning models or Bidirectional Long Short-term Memory (abbreviated BiLSTM) machine learning models for the detection of clickbait, sentiment, bias and toxicity.

[12] As an alternative to N single task machine learning models each performing a single task like for instance the detection of clickbait, it is also possible to integrate in the disinformation detection system according to the invention an N-athlon machine learning model that is able to perform N tasks at once. Such N-athlon model needs to be supplied with two things: a question and context. The output of the N-athlon model then is an answer to the question with indication of the relevant part of the context that motivates the answer to the question, wherein the relevant part of the context is indicated through start and end index logits. The N-athlon model may for instance be supplied with a sentence and the question "is this sentence clickbait or not clickbait?". The output of the N-athlon machine learning model could then be "clickbait" or "not clickbait", together with a range [start index, end index] that indicates where in the sentence the indication for "clickbait" or "not clickbait" was found. The same N-athlon machine learning model could be supplied with a sentence and the question "Is this sentence biased or not biased", and would return as answer either "biased" or "not biased" together with a range [start index, end index] that indicates where in the sentence the indication for "biased" or "not biased" was found.

[13] In embodiments of the disinformation detection system according to the invention defined by claim 2, the at least one background harvester comprise(s) one or more of:

- a Really Simple Syndication collector, abbreviated RSS collector, configured to periodically fetch and store information items from RSS sources specified in a configuration file;

- a cascade stream harvester, configured to periodically fetch and store Twitter cascades for information items whose URL is specified in a cascades link list;

- a financial data harvester, configured to periodically fetch and store financial data related to an information item from at least one predetermined financial data source;

- a social network data harvester, configured to periodically fetch and store social network data related to an information item from at least one predetermined social network;

- a fact checking data harvester, configured to periodically fetch and store fact checking data related to an information item from at least one fact checking data source.

[14] Thus, the one or more background harvesters that periodically fetch information items and/or related data from predefined information sources may comprise an RSS collector. The RSS sources and their parameters are specified in a configuration file that also specifies the webserver configuration. Using as input the RSS URL, the RSS collector fetches the news items from the provided RSS URL and stores these news items in the data storage. After fetching a news item, the RSS collector waits for a record time interval, i.e. a parameter that specifies a duration to wait before fetching the next news item.

[15] The one or more background harvesters further may comprise a cascade stream harvester that fetches Twitter cascades, i.e. cascades of posts distributed via the social network platform Twitter, in relation to an information item. The cascade stream harvester makes use of a cascades link list whereto the URLs of all harvested information items are added. The cascades link list comprises two dequeues. The URLs of all harvested information items are added to a first dequeue, the so-called "links to-do" dequeue. The cascade stream harvester pops URLs out of this to-do dequeue, fetches corresponding Twitter cascades and stores these Twitter cascades

in the data storage. The cascade stream harvester then puts the processed URL in a second deque, the so-called "links done" deque. Due to the Twitter API rate limits, the harvesting of Twitter cascades is a slower process than retrieving information items from news websites. For this reason, an implementation with two dequeues is preferred. The cascade stream harvester relies on the official Twitter API and a social network service scraper that makes a request for a given URL and returns a list of identifiers of tweets, i.e. a list of IDs of individual posts on the social network Twitter that refer to the URL. The cascade stream harvester in other words implements an ID lookup request through the official Twitter API in order to retrieve related tweets for a given information item. The number of tweets that is returned for such a lookup may be limited to N, where N is an integer parameter that may be configured manually in order to control the time spent to retrieve related tweets in case the total number of tweets related to a single news article is high. N may for instance be set equal to 100. To deal with the rate limits set by the official Twitter API, the cascade stream harvester waits a predetermined amount of time, for instance 2 seconds, in between subsequent requests. In a preferred implementation, the cascade stream harvester makes search requests through the official Twitter API using for example the Twython library. This way, the search covers the last 7 days and it is ensured that recent tweets are received. Preferably, the cascade stream harvester is configured to store tweets in the data storage only when they are not overlapping with earlier archived tweets.

[16] The one or more background harvesters further may comprise a financial data harvester that periodically collects and stores financial data related to harvested information items. The financial data are collected from a financial data source, for instance Yahoo Finance, and preferably comprise stock information for companies or organisations mentioned in the information item. The stock reference is either directly extracted from the information item or obtained from a stock dictionary that pairs names of companies and organisations with their relative stock references.

[17] The one or more background harvesters further may comprise a social network data harvester that periodically collects and stores social network data related to harvested information items. Thus, apart from Twitter cascades, social network data related to a harvested information item may be collected from other social networks

like for instance Reddit, Tumblr, Periscope, etc., through requests for posts with specified tags through the official APIs of these social networks.

[18] The one or more background harvesters further may comprise a fact checking data harvester that periodically collects and stores fact checking data related to harvested information items. The fact checking data are collected from fact checking services like for instance FactCheck, Hoax-Slayer, Politifact, Snopes, TruthOrFiction, Urban Legends, etc., and generally concern labelled news articles related to the harvested information item under consideration.

[19] The background harvesters fit into the scalable microservice architecture and contribute to the exploitation of auxiliary information that extends beyond the mere textual data used within existing models. The skilled person will appreciate that the above list of background harvesters however is non-exhaustive.

[20] In embodiments of the disinformation detection system according to the invention defined by claim 3, the at least one on-demand harvester comprise(s) one or more of:

- an on-demand information item harvester, configured to fetch and store the content of an information item in return to the URL of the information item;
- an on-demand cascade harvester, configured to fetch and store a Twitter cascade in return to the URL of the information item.

[21] Thus, the one or more on-demand harvesters that fetch and store information items and/or related data in return for a URL of an information item may comprise an on-demand information item harvester. The on-demand information item harvester takes as input the URL or link of an information item and fetches the content of the information item. The on-demand information item harvester is made accessible via an API.

[22] The one or more on-demand harvesters further may comprise an on-demand cascade harvester. The on-demand cascade harvester takes as input the URL or link of an information item and fetches a corresponding cascade of posts or tweets distributed via the Twitter social network. An API is implemented to access the on-demand cascade harvester.

[23] The on-demand harvesters fit into the scalable microservice architecture and contribute to the exploitation of auxiliary information that extends beyond the mere textual data used within existing models. The skilled person will appreciate that the above list of on-demand harvesters however is non-exhaustive.

[24] In embodiments of the disinformation detection system according to the invention, defined by claim 4,

- each of the first BERT model, the second BERT model, the third BERT model and the fourth BERT model comprises twelve layers configured to respectively generate twelve 768 dimensional feature vectors; and

- the scoring microservice is further configured to concatenate the four 768 dimensional feature vectors generated by the last four layers of the twelve layers to obtain a 3072 dimensional associated feature vector for each BERT model.

[25] The single task BERT models are able to classify an input sequence, i.e. textual data like for instance a sentence, a paragraph, a word, an article, etc., for different tasks. In the disinformation system according to the invention, four single task BERT models respectively perform the task of detecting clickbait, sentiment, bias and toxicity.

It is desired however that the single task models generate extra features as input for the machine learning model that detects disinformation or fake news using the output of the single task BERT models. A base BERT model consists of twelve layers and the output of each of these twelve layers is a 768-dimensional feature vector. It is for instance possible to provide the output of the last layer, i.e. the twelfth 768 dimensional feature vector, as input for the following fake news detecting machine learning model. Alternatively, the 768-dimensional feature vectors could be summed across the twelve layers, and the sum of these twelve feature vectors could serve as input for the following fake news detecting machine learning model. The best results however are achieved when the 768-dimensional feature vectors of the last four layers of a BERT model are concatenated to form a 3072-dimensional feature vector that serves as input for the following fake news detecting machine learning model.

[26] In embodiments of the disinformation detection system according to the present invention, defined by claim 5, the monitoring microservice is configured to forward an

information item to each one of the first machine learning model, the second machine learning model, the third machine learning model and the fourth machine learning model.

5 **[27]** The first, second, third and fourth machine learning model implement feature extraction for an information item, e.g. a news article. This is done by submitting the information item to the fine-tuned BERT models in case the first, second, third and fourth machine learning models are implemented through single task BERT models. The same information item is thus forwarded to every fine-tuned BERT model and the
10 3072-dimensional feature vector resulting from concatenating the feature vectors of the last four encoder layers of the BERT model is taken as the set features extracted by that BERT model for the information item. The four single task BERT models hence jointly produce 12288 features for the information item.

15 **[28]** In embodiments of the disinformation detection system according to the present invention, defined by claim 6:

- the monitoring microservice is configured to forward each tweet message of a Twitter cascade to the first machine learning model, the second machine learning model, the third machine learning model and the fourth machine learning model; and
- 20 - the scoring microservice is further configured to obtain for each one of the first machine learning model, the second machine learning model, the third machine learning model and the fourth machine learning model an associated feature vector that corresponds to the mean value of associated feature vectors obtained for individual tweet messages of said Twitter cascade.

25

[29] The first, second, third and fourth machine learning model also implement feature extraction for related data for an information item, e.g. a Twitter cascade. Because the number of tweets related to a single information item may be high, this is done by submitting the tweets one-by-one to each of the four machine learning models and
30 determining the mean value across all feature vectors extracted by a single machine learning model for tweets related to a single information item. Thus, each tweet of a Twitter cascade related to certain information item, e.g. a new article or event, is submitted to the fine-tuned BERT models in case the first, second, third and fourth machine learning models are implemented through single task BERT models. The

3072-dimensional feature vectors are extracted for all tweets. Then, the scoring microservice determines the mean value for each of the 3072 features across all tweets that form part of the Twitter cascade related to the information item. This results in four 3072-dimensional feature vectors respectively extracted by the four single task BERT models. The four single task BERT models hence jointly produce 12288 features for the Twitter cascade related to an information item.

[30] In embodiments of the disinformation detection system according to the invention, defined by claim 7, N corresponds to four, and the N-athlon model corresponds to a Tetrathlon model trained for clickbait detection, sentiment detection, bias detection and toxicity detection, the Tetrathlon model being configured to receive an information item and/or related data together with a question as input sequence and to return a start index and end index of parts of the information item and/or related data respectively indicating clickbait, sentiment, bias or toxicity.

[31] Indeed, as mentioned here above, the ultimate goal of disinformation detection may rely on four individual tasks, namely clickbait detection, sentiment analysis, bias detection and toxicity detection. These four individual tasks may be performed by a single Tetrathlon machine learning model, i.e. an N-athlon machine learning model that is trained with datasets to perform the four tasks.

[32] In embodiments of the disinformation detection system according to the invention, defined by claim 8:

- the Tetrathlon model comprises twelve layers configured to respectively generate twelve 768 dimensional feature vectors; and
- the scoring microservice is further configured to concatenate the four 768 dimensional feature vectors generated by the last four layers of the twelve layers to obtain a 3072 dimensional associated feature vector for the input sequence.

[33] The Tetrathlon model is able to classify an input sequence, i.e. textual data like for instance a sentence combined with a proper question, for four different tasks. It is desired however that the Tetrathlon model generates extra features as input for the machine learning model that detects disinformation or fake news using the output of the Tetrathlon model for the four tasks. When implemented as a base BERT model,

the Tetrathlon model consists of twelve layers and the output of each of these twelve layers is a 768-dimensional feature vector. It is then for instance possible to provide the output of the last layer, i.e. the twelfth 768 dimensional feature vector, obtained for the four tasks as input for the following fake news detecting machine learning model.

5 Alternatively, the 768-dimensional feature vectors could be summed across the twelve layers, and the four sums of these twelve feature vectors obtained respectively for the four tasks could serve as input for the following fake news detecting machine learning model. The best results however are achieved when the 768-dimensional feature vectors of the last four layers of the Tetrathlon model are concatenated to form a 3072-
10 dimensional feature vector for each of the four tasks, that jointly serve as input for the following fake news detecting machine learning model.

[34] In embodiments of the disinformation detection system according to the present invention, defined by claim 9, the monitoring microservice is configured to forward an
15 information item together with the question "is this sentence clickbait or not clickbait?" as a first input sequence or first pass to said Tetrathlon model, to forward said information item together with the question "is this sentence positive, neutral or negative?" as a second input sequence or second pass to said Tetrathlon model, to forward said information item together with the question "is this sentence biased or not
20 biased?" as a third input sequence or third pass to said Tetrathlon model, and to forward said information item together with the question "is this sentence toxic or not toxic?" as a fourth input sequence or fourth pass to said Tetrathlon model.

[35] Indeed, for the performance of the disinformation detection system, it is important
25 to ask the questions identifying the four different tasks in a good way to the Tetrathlon model. For clickbait detection, one could for instance consider submitting a sentence together with the question "Is this clickbait?" to the Tetrathlon model. However, in such case, when implemented as a BERT model, the Tetrathlon model would not know how to answer "not clickbait" because the BERT model can only return parts of the input
30 sequence as answer. Therefore, in a more preferred embodiment, the monitoring microservice submits the sentence together with the question "Is this clickbait or not clickbait?" to the Tetrathlon model allowing the latter to answer both "clickbait" or "not clickbait" through a range [start index, end index]. Similarly, in preferred embodiments, the monitoring microservice is configured to submit a sentence together with the

question "Is this sentence positive, neutral or negative?" to the trained Tetrathlon model in order to enable the latter to analyse the sentiment and return as answer either "positive", "neutral" or "negative" through a range [start index, end index]. In preferred embodiments, the monitoring service is further configured to submit a sentence together with the question "Is this sentence biased or not biased?" to the trained Tetrathlon model in order to allow the latter to perform the task of bias detection and return as answer either "biased" or "not biased" through a range [start index, end index]. In preferred embodiments, the monitoring microservice is at last configured to submit a sentence together with the question "Is this sentence toxic or not toxic?" in order to enable the trained Tetrathlon model to detect toxicity and return as answer either "toxic" or "not toxic" through a range [start index, end index]. A drawback of the Tetrathlon machine learning model is that this model has the ability to return a nonsensical answer. The model behaves correctly when it returns "toxic" or "not toxic". However, it is possible that the model returns a different part of the input sequence, e.g. a part of the sentence or a different part of the question, in which case no valid answer is returned.

[36] In embodiments of the disinformation detection system according to the invention, defined by claim 10:

the monitoring microservice is configured to forward each tweet message of a Twitter cascade together with the question "is this sentence clickbait or not clickbait?" as a first input sequence or first pass to the Tetrathlon model, together with the question "is this sentence positive, neutral or negative?" as a second input sequence or second pass to the Tetrathlon model, together with the question "is this sentence biased or not biased?" as a third input sequence or third pass to the Tetrathlon model, and together with the question "is this sentence toxic or not toxic?" as a fourth input sequence or fourth pass to the Tetrathlon model; and

the scoring microservice is further configured to obtain four associated feature vectors as mean value of associated feature vectors for respectively first input sequences comprising tweet messages of the Twitter cascade, second input sequences comprising tweet messages of the Twitter cascade, third input sequences comprising tweet messages of the Twitter cascade, and fourth input sequences comprising tweet messages of the Twitter cascade.

[37] Indeed, also the related data like a Twitter cascade must be submitted four times to the Tetrathlon model accompanied by the respective properly worded questions in order to allow the Tetrathlon model to generate useful responses indicating clickbait, sentiment, bias and toxicity for each tweet message in the Twitter cascade. In addition thereto, the Tetrathlon model extracts features for each of the tweet messages and each of the tasks. In the above-described example implementation wherein the 768-dimensional output vectors of the last four encoder layers of the Tetrathlon model are concatenated, a 3072-dimensional feature vector is generated for each tweet message and each task. In a preferred embodiment, the 3072-dimensional feature vectors obtained for a single task, for instance clickbait detection, and for tweet messages that belong to the same Twitter cascade, are then averaged by the scoring microservice and the mean value of these 3072-dimensional feature vectors is kept as a single 3072-dimensional feature vector for the entire Twitter cascade related to an information item and for the specific task. The Tetrathlon model applied to a Twitter cascade hence shall result in four 3072-dimensional feature vectors being generated for the respective tasks of clickbait detection, sentiment analysis, bias detection and toxicity detection.

[38] In embodiments of the disinformation detection system according to the invention, defined by claim 11, the scoring microservice is further configured to extract time series from the related data for an information item.

[39] The time series may for instance be extracted from the Twitter cascade related to an information item, and may for instance take the form of a vector x_n with n being a positive integer index indicating the amount of time (counted for instance in hours) lapsed since the publication of the information item, and x_n being the number of tweets in the related Twitter cascade posted in the n 'th time interval since the publication of the corresponding information item. Hence, x_1 represents the number of tweets posted in the first hour after publication of a news article, x_2 represents the number of related tweets posted in the second hour after publication of a news article, etc. In preferred embodiments of the disinformation detection system according to the invention, the scoring microservice is configured to extract such time series from the collected related data for an information item and feeds the time series together with the features extracted from the information item and related data to the disinformation classifying

machine learning model, for instance a deep Markov Random Field model or deep MRF model trained to classify information items as either fake or not.

[40] In embodiments of the disinformation detection system according to the invention, defined by claim 12:

- the scoring microservice is further configured to extract social network graphs from the related data for an information item; and
- the scoring service comprises a machine learning model trained to learn a representation of a social network graph.

[41] The social network graph may for instance be extracted from the Twitter cascades related to plural information items and may take the form of a user graph, i.e. a graph of correlated users for a single event, and/or an event graph, i.e. a graph of correlated information items or events. A user graph has users as nodes and the edges have values indicative for the number of common events involving both users. An event graph has events or information items as nodes and the edges have values indicative for the number of users tweeting about both events. Extracting a user graph and/or event graph relies on the insight that the correlation between users and/or events is useful information for the disinformation classification. If the same user posts tweets about two information items (i.e. two events), and one of those information items represents fake news, the other information item is more likely to be fake as well. A user graph can be generated for each event and can be forwarded to the disinformation classifying machine learning model, for instance a fully connected classifier trained to classify individual information items as either fake or not. The event graph can be used to generate an adjacency matrix for events, and the latter adjacency matrix can be forwarded to the disinformation classifying machine learning model, for instance a deep Markov Random Field model or deep MRF model trained to classify information items as either fake or not.

[42] Embodiments of the disinformation detection system according to the invention, defined by claim 13, further comprise a display unit configured to generate one or more of following visuals for an information item:

- the information item wherein parts of the information item detected as clickbait, sentiment, bias, or toxic are marked;

- related data of the information item wherein parts of the related data detected as clickbait, sentiment, bias, or toxic are marked;

- a social network graph showing relations and/or locations of users that interact with the information item;

5 - evolution in time of the social network graph.

[43] This way, embodiments of the disinformation detection system according to the invention generate visuals indicating which parts of a news article or the related social network posts were identified as clickbait, sentiment, bias or toxic. Further visuals may
10 be generated that show the graph of social network posts related to a news article and/or that visualize on a map the locations of social network users that interact with a news article and/or that visualize the relations between social network users that interact with a news article in a user graph. The evolution in time of the graph of social network posts and/or the location map of social network users interacting with a news
15 article and/or the user graph of relations between social network users that interact with a news article may be visualized as well.

Brief Description of the Drawings

20

[44] Fig. 1 illustrates an embodiment of the disinformation detection system 100 according to the present invention;

[45] Fig. 2 is a functional block scheme of an example implementation of the
25 monitoring microservice 200 in embodiments of the disinformation detection system according to the present invention;

[46] Fig. 3 is a functional block scheme of an example implementation of the scoring
30 microservice 300 in embodiments of the disinformation detection system according to the present invention;

[47] Fig. 4 illustrates a single task BERT machine learning model 400 used in example implementations of the scoring microservice that forms part of embodiments of the disinformation detection system according to the present invention;

[48] Fig. 5 illustrates a tetrathlon BERT machine learning model 500 used in example implementations of the scoring microservice that forms part of embodiments of the disinformation detection system according to the present invention;

5

[49] Fig. 6 illustrates feature extraction for an information item in example implementations of the scoring microservice that forms part of embodiments of the disinformation detection system according to the present invention;

10 [50] Fig. 7 illustrates feature extraction for a cascade of n tweets related to an information item in example implementations of the scoring microservice that forms part of embodiments of the disinformation detection system according to the present invention; and

15 [51] Fig. 8 illustrates a suitable computing system 800 for realizing embodiments of the disinformation detection system according to the invention.

Detailed Description of Embodiment(s)

20

[52] Fig. 1 shows a disinformation detection system 100 with microservice architecture comprising an identity microservice 101, a rating microservice 102, a scoring microservice 103, a training microservice 104, a monitoring microservice 105, a notification microservice 103, all coupled to an event bus 110. The disinformation
25 detection system 100 further comprises API gateways 120 enabling the monitoring microservice 105 to connect with client applications 130. Fig. 1 further also shows external sources 140 of information items and/or related data where the monitoring microservice 105 connects with.

30 [53] The identity microservice 101 implements a token-based authentication mechanism enabling users to authenticate and authorize themselves to other microservices in the disinformation detection system 100.

[54] The rating microservice 102 allows to schedule on-demand tasks. The rating microservice 102 is configured to analyse requests from external platforms or from information consumers through their browser. Once a task is registered by the rating microservice 102, an event is broadcasted on the event bus 110 to notify other microservices to start collecting and analysing the requested information.

[55] The scoring microservice 103 executes trained machine learning models to generate disinformation predictions. The generated predictions are broadcasted on the event bus 110 to be communicated back to information consumers and to be used by the training microservice 104 for further training the machine learning models. Additionally, the generated predictions may be stored in a database, not shown in Fig. 1, so that they can be exploited to improve model predictions if no online training mechanism is feasible for a machine learning model. The advantage of an independent scoring microservice 103 lies in that machine learning models can be integrated easily without knowledge of other components of the disinformation detection platform. Further, the training microservice 104 can take advantage of the predictions outputted by the scoring microservice 103 to improve the machine learning models that generate the predictions. Example implementations of the scoring microservice 103 will be described in more detail below with reference to Fig. 3 - Fig. 7.

[56] The machine learning models executed by the scoring microservice 103 to generate disinformation predictions may have the possibility to be updated on the fly or at predefined time intervals. The training microservice 104 allows researchers to monitor the prediction outputs of the machine learning models in the scoring microservice 103, and automatically improves these machine learning models upon input received from these researchers.

[57] The monitoring microservice 105 is configured to obtain the information items and related data that are subject to disinformation predictions. The monitoring microservice 105 thereto comprises one or plural so-called background harvesters that continuously or periodically collect information items like news articles and related data like tweets or financial data from predefined information sources. The background harvesters may use keywords or topics where information consumers are interested in to select the information items and/or related data that will be fetched from the

predefined information sources. In addition thereto, the monitoring microservice 105 comprises so-called on-demand harvesters that collect information items and/or related data from information sources that are beyond the scope of the background harvesters. The on-demand harvesters operate for instance upon receipt of a request or demand from a client through a client application 130. The collected information items and/or related data are broadcasted by the monitoring microservice 105 on the event bus 110 to be scored by the scoring microservice 103. The monitoring microservice 105 also comprises a data storage wherein the collected information items and/or related data are stored for future potential use. A possible implementation of the monitoring microservice 105 will be described in more detail below with reference to Fig. 2.

[58] The notification microservice 106 allows for browsers or external platforms to subscribe to information channels. When information items are detected that match with the interest profile of a subscriber, the notification microservice 106 can use for instance direct socket channels or mail notifications to such subscriber.

[59] The communication across the different microservices 101-106 is established through the event bus 110. The event bus may for instance be implemented as a RabbitMQ Bus, an Azure Service Bus, or an Amazon EventBridge.

[60] Fig. 2 shows an example implementation of the monitoring microservice 200 in embodiments of the disinformation detection system according to the present invention. The monitoring microservice 200 of Fig. 2 for instance constitutes a possible realization of the monitoring microservice 105 in Fig. 1. An essential part of the monitoring microservice 200 is its data harvesting technology 220, 230. The purpose of this data harvesting technology is to continuously and rapidly mine data, i.e. information items and related data, from various information sources, and to store the collected information items and related data in a database 240. This way, a disinformation dataset is built that is further exploited to train the machine learning models for further detecting misleading information. The data harvesting technology comprises an on-demand harvesting service 220 and a background harvesting service 230. The background harvesting service 230 periodically collects information items and/or related data from predefined data sources and stores the collected data in the

database 240 using a pre-configured database manager 234. The information sources to be crawled periodically by the background harvesters are listed in a configuration file that can be adjusted from time to time. The different background harvesters run in parallel such that all the information sources specified in the configuration file are being
5 crawled simultaneously within a predefined time period. The on-demand harvesting service 220 comprises one or plural on-demand news harvesters 221 and one or plural on-demand cascade harvesters 222. An on-demand news harvester 221 takes as input a link to a news article (link to an information item), for instance its URL, and fetches and returns the content of the news article. An API is implemented to access an on-
10 demand news harvester 221. An on-demand cascade harvester 222 on the other hand also takes as input a link to a news article (link to an information item), for instance its URL, but fetches and returns the corresponding cascade of Twitter data, i.e. the cascade of tweets referring to that news article. An API is implemented to access an on-demand cascade harvester 222. The information items and related data, harvested
15 either continuously in the background by the background service 230 or on-demand by the on-demand service 220, are stored in the database 240, whose purpose is to store the harvested data for future analysis or re-use. The database 240 may for instance be implemented as a MongoDB database.

20 **[61]** Application 210 launches the harvester starter 231 in the background harvester service 230, launches the harvesting of news articles on-demand by the on-demand news harvester 221 and launches the harvesting of Twitter cascades on-demand by the on-demand cascade harvester 222.

25 **[62]** The on-demand news harvester 221 takes the link to a news article as input, for instance the URL, and fetches the content of the news article using a predetermined library, for instance the Newspaper python library accessible via the URL <https://newspaper.readthedocs.io/en/latest/>. The on-demand news harvester 221 works on-demand, so it only starts harvesting and processing data upon receipt of a
30 request. The on-demand cascade harvester 222 also takes the link to a news article as input and using an intermediate library, for instance the Twython library accessible via the URL <https://twython.readthedocs.io/en/latest>, makes a request to the official Twitter API to fetch the corresponding cascade of tweets that reference the news article. Just like the on-demand news harvester 221, the on-demand cascade

harvester 222 works on-demand, so it only starts harvesting and processing data upon receipt of a request.

[63] In the background harvesting service 230, the harvester starter 231 is a module that loads the configuration file listing the information sources that are periodically mined, and initializes the respective background harvesters 235, 236, 237, the database manager 234 and the webserver 233 via the harvester controller 232. The latter harvester controller 232 is a central controller that reads and interprets the configuration file, initiates the database connections specified in the configuration file and starts the respective background harvesters that mine these databases. Each background harvester communicates with the harvester controller 232 for updates. The webserver 233 is a central webserver that allows to control and monitor background harvesters in runtime via a web application. The configuration of this webserver 233 is declared in the configuration file. The database manager 234 represents a single client for the database 240. This way, a single client connection to the database 240 can be reused rather than establishing a new connection for every new database operation.

[64] The RSS collector 235 is configured to periodically fetch and store news articles from an RSS source identified by its RSS URL. The RSS collector 235 has a parameter, the record interval, that specifies the duration or time interval to wait in between getting subsequent news articles. Plural RSS collectors may be instantiated to mine respective RSS sources identified by respective RSS URLs. All RSS sources that must be mined periodically in the background and their parameters are identified in the configuration file.

25

[65] The cascade stream harvester 236 is a background harvester for Twitter cascades. The cascade stream harvester 236 uses two dequeues: a first dequeue, the so-called "LinksToDo" dequeue, wherein all the links to harvested news articles are stored, and a second dequeue, the so-called "LinksDone dequeue, wherein processed links are stored, i.e. links to news articles for which the corresponding Twitter cascades have been fetched and stored in the database 240. Due to the Twitter API rate limits, harvesting Twitter cascades is a slower process than harvesting news articles. The approach with two dequeues however allows to run this slower process in the background. The cascade stream harvester 236 uses the official Twitter API and a

social networking scraper. The social networking scraper makes a request for a given URL and in return gets a list of identifiers (abbreviated IDs) of tweets that reference the news article identified through the URL. The cascade stream harvester 236 has a parameter N that is manually configurable. N is the maximum number of tweets that is returned for a single ID lookup request. This way, the time spent to retrieve all relevant tweets corresponding to a news article is restricted. In addition thereto, the cascade stream harvester 236 makes a search request to the official Twitter API that covers only a recent period, for instance the last 7 days range. The so harvested recent tweets are stored in the database 240 if they are not overlapping with already stored tweets. The cascade stream harvester 236 waits a predetermined time interval, for instance set at 2 seconds, before making the next request to avoid the rate limits set by Twitter.

[66] The harvesters 237 at last represent a variety of background harvesters corresponding to different data sources. These harvesters 237 share a common fetch-and-store function for retrieving and processing data, and further have parameters that allow to tune the behaviour of the background harvester to the type of harvester. A financial data harvester is a first example of such a background harvester, configured to periodically fetch and store financial data related to harvested news articles. The financial data harvester searches for stock names in a news article. If stock names are not mentioned in the article, the financial data harvester searches for company names in the news article and consults a dictionary that pairs stock names to the mentioned company names. Using the stock names together with the publication date of the article, the financial data harvester collects financial data like stock data from a financial data source, for example Yahoo Finance, Benzinga, Hoax-Slayer, Investing, InvestorVillage, InvestorsHub, Motley Fool, SeekingAlpha, SmallCap Network, etc., for a data range that is relevant given the publication date of the news article. A fact checking data harvester is a second example of such a background harvester, configured to periodically fetch and store fact checking data related to harvested news articles. A fact checking data harvester collects for a news article related fact checking data from a fact checking data source like for instance FactCheck, Hoax-Slayer, Politifact, Snopes, TruthOrFiction, Urban Legends, etc. The fact checking data typically represent labelled articles or other information items wherein the label indicates if the article contains disinformation (fake news) or not. For every article, the fact checking data harvester extracts stock names or company names and further harvests related

stock information or other financial data for the fact-checking articles. A social network data harvester is a third example of such a background harvester, configured to periodically fetch and store social network data related to harvested news articles. A social network harvester collects social network posts with specified tags using the official API of a social network like for instance Reddit, Tumblr, Periscope, Twitter, etc.

[67] Fig. 3 is a functional block scheme of an example implementation of the scoring microservice 300 in embodiments of the disinformation detection system according to the present invention. The scoring microservice 300 illustrated by Fig. 3 hence represents a possible implementation of the scoring microservice 103 in Fig. 1. The scoring microservice 300 comprises a feature extraction unit 301 configured to take an information item or related data as input and comprising one or plural machine learning models trained to perform four tasks: detection of clickbait, analysis of sentiment, detection of bias and detection of toxicity. Example implementations of the feature extraction unit 301 wherein these four tasks are performed by four separate single-task machine learning models, and wherein these four tasks are performed by a single multi-task machine learning model will be described further below with reference to Fig. 4 and Fig. 5 respectively. In addition to a classification for the four tasks, the machine learning models in the feature extraction unit 301 are exploited to generate a feature vector. Example implementations of the feature extraction for an information item and a cascade of tweets relating to a single information item will be described in further detail below with reference to Fig. 6 and Fig. 7 respectively. The classification for the four tasks and extracted feature vector(s) are supplied to a machine learning classifier 302, for example a softmax classifier, i.e. a machine learning model that is trained to assess the presence of disinformation in an information item based on the clickbait, sentiment, bias and toxicity classifications of that information item and the related data like the corresponding Twitter cascade, related financial data and related fact-checking articles. The classifier 302 takes the classification of the four tasks and the feature vectors as input and classifies an information item either as fake (disinformation) or not fake (no disinformation).

[68] An enhanced implementation of the classifier 302 shown in Fig. 3 further takes time series generated by the time series extraction unit 304 and a user graph generated by the user graph unit 305 as inputs. The related data, more precisely the

Twitter cascades or other social network data, are supplied to the time series extraction unit 304 which generates a time series indicating how people on social media respond to a news event, e.g. a published information item. The time series may for instance be a vector X_n of n values representing the number of social network posts (e.g. the number of tweets) associated with the news event during subsequent hours after the information item is published. Hence, X_1 would represent the number of tweets in the first hour after publication of a news article, X_2 would represent the number of related tweets in the second hour after publication of the news article, ..., and X_n would represent the number of related tweets in the n 'th hour after publication of the news article. The extracted time series contains useful information that shall enable the classifier 302 to generate a more accurate disinformation assessment for the news article. The related data, more precisely the Twitter cascades or other social network data, are also supplied to the user graph unit 305 which generates a graph of user involved with the corresponding news event, e.g. a published information item. The user graph unit 305 implements a machine learning model that is trained to learn a representation of the graph of users extracted from the social network data fetched in relation to the information items. The user graph machine learning module 305 takes as input the graph of users extracted from the social network data fetched in relation to the information items. The user graph may for instance be a graph wherein the nodes represent users and the edge between two nodes corresponds to the number of common events that involve the two users. A user or node in the user graph may for instance be represented by a vector specifying the user through a number of parameters like for instance the number of favourites the user has, the number of followers the user has, the number of friends the user has, the name length of the user, the number of statuses, etc. The representation (output of the user graph machine learning module) of the so generated user graph also constitutes useful information that shall enable the classifier 302 to generate a more accurate disinformation classification for the news article.

[69] In Fig. 3, the classifier 302 is followed by a Markov Random Field based Graph Neural Network (abbreviated MRF Based GNN) 303. The MRF based GNN 303 further exploits information available in an adjacency matrix 307 that is generated by an event graph unit 306. The adjacency matrix 307 represents a graph wherein the nodes represent information items (e.g. news articles) and the edge between two nodes

represents the correlation between the two corresponding information items. Thus, whereas the classifier 302 labels each information item independently as fake (disinformation) or not fake (no disinformation), the MRF layers of the MRF based GNN 303 exploit the correlation between information items to perform a label smoothing operation. The final disinformation labels are thus computed by an iterative mean field algorithm implemented through the MRF layers unrolled in a graph neural network (abbreviated GNN) that models the adjacency of information items. The presence of the MRF layers in the scoring microservice 300 will lead to changes in the labels of the nodes of the GNN - corresponding to changes in the disinformation assessment of information items - as a result of the node's interaction with its neighbours, in other words as a result of the correlation between information items. The scoring microservice 300 or a separate relevancy unit may further be configured to determine a relevancy or explanation for the disinformation classification of an information item. The relevancy shall for instance indicate positive contributions and negative contributions to the disinformation classification of an information item in terms of its neighbouring information items and its intrinsic features. The operation of such relevancy unit is explained in detail in a counterpart patent application of the same applicant, entitled "A Relevancy Enhanced Disinformation Detection System", incorporated herein by reference.

[70] As mentioned here above, the four tasks performed by the feature extraction unit 301 in the scoring microservice 300 may be realized through four single task machine learning models. Fig. 4 illustrates such a single task BERT (Bidirectional Encoder Representations from Transformers) based machine learning model 400. Transformers have reshaped the natural language processing (NLP) domain in recent times. A machine learning model can be pre-trained with general tasks on a big data set. The pre-trained model can then be fine-tuned for specific tasks with much smaller data sets. This is a common principle behind BERT. In a possible implementation of the feature extraction unit 301, four BERT based machine learning models may thus be fine-tuned for the respective single tasks of clickbait detection, sentiment analysis, bias detection and toxicity detection. These BERT models are fine-tuned to specific data sets.

[71] Fig. 4 further illustrates the operation of such a single task BERT model, for example the single task BERT model that is fine-tuned for the task of clickbait detection. The single task model 400 comprises a single task BERT model 420 and a single task classifier 430. The single task BERT model 420 takes a sentence as input, for instance the title of a news article. In the particular example of Fig. 4, the input sentence is: "Leading doctor reveals the No. 1 worst carb you are eating". The single task BERT model has 512 input tokens the first of which is a [CLS] token that signifies the start of a sentence. The following input tokens each correspond to a subsequent word or term in the input sentence. The single task BERT model further has twelve encoder layers 401-412. Each of these twelve layers generates a 768-dimensional feature vector. The output of the twelfth encoder layer 412 corresponding to the [CLS] token represents the entire input sentence and is forwarded as input 431 to the single task classifier 430, i.e. a machine learning model trained to output a classification 432 for a single task, e.g. "clickbait" or "not clickbait" in the example of clickbait detection.

[72] As mentioned here above, the four tasks performed by the feature extraction unit 301 in the scoring microservice 300 may alternatively be realized through a multi-task machine learning model or so-called N-athlon model. In the example of four tasks, the N-athlon model becomes a Tetrathlon model. Fig. 5 illustrates such a BERT (Bidirectional Encoder Representations from Transformers) based Tetrathlon machine learning model 500. Indeed, it is possible to train a single BERT model 520 to perform all four tasks, i.e. clickbait detection, sentiment analysis, bias detection and toxicity detection. Such tetrathlon BERT model 520 takes as input a question and context. The output of such tetrathlon BERT model 520 is then the answer to the question. It is worth noting however that the tetrathlon BERT model 520 cannot come up with an answer of its own. It must extract the answer from its input, so either from the question or the context. The tetrathlon BERT model 520 only outputs two things: a start index and an end index. This start index and end index point to the position in the input, i.e. question plus context, where the answer to the question is found. Consequently, it is important to phrase for each task that the Tetrathlon BERT model 520 must execute the question in such a manner that the possible answers for that task are contained therein.

[73] The architecture of the Tetrathlon model 500 is illustrated by Fig. 5. It includes Tetrathlon BERT machine learning model 520 trained to perform the four tasks. The

Tetrathlon BERT machine learning model 520 can receive 512 tokens as input, and has twelve encoder layers 501-512 that each generate a 768-dimensional feature vector for the next encoder layer. The outputs of the Tetrathlon BERT machine learning model for all tokens are passed to a linear decision layer 521, 522, 523, 524, ... 52n with two outputs. These two outputs represent start index logits and end index logits. The start index logits 531 are collected in a start index classifier 541 that typically implements a softmax algorithm to generate the start index 551 of the answer to the question. Similarly, the end index logits 532 are collected in an end index classifier 542 that typically implements a softmax algorithm to generate the end index 552 of the answer to the question. The range [551, 552] or [start index, end index] indicates where in the input the answer for the specific task can be found.

[74] In order to enable the Tetrathlon model 500 to execute the four tasks for the example described here above for the embodiment relying on single task machine learning models, i.e. a news article with title "Leading doctor reveals the No. 1 worst carb you are eating", the same sentence must be supplied four times to the Tetrathlon model 500, each time accompanied by a different question. In a first run wherein the Tetrathlon model 500 is used to detect clickbait, the input supplied to the Tetrathlon BERT model 520 corresponds to "[CLS] Is this sentence clickbait or not clickbait? [SEP] Leading doctor reveals the No. 1 worst carb you are eating [SEP]". Herein [CLS] signifies the start of a sentence and [SEP] represents a separator. The Tetrathlon model 500 behaves well if it extracts either "clickbait" or "not clickbait" from the input. In a second run wherein the Tetrathlon model 500 is used to analyse sentiment, the input supplied to the Tetrathlon BERT model 520 corresponds to "[CLS] Is this sentence positive, neutral or negative? [SEP] Leading doctor reveals the No. 1 worst carb you are eating [SEP]". The Tetrathlon model 500 behaves well if it extracts either "positive", "neutral" or "negative" from the input. In a third run wherein the Tetrathlon model 500 is used to detect bias, the input supplied to the Tetrathlon BERT model 520 corresponds to "[CLS] Is this sentence biased or not biased? [SEP] Leading doctor reveals the No. 1 worst carb you are eating [SEP]". The Tetrathlon model 500 behaves well if it extracts either "biased" or "not biased" from the input. In a fourth run wherein the Tetrathlon model 500 is used to detect toxicity, the input supplied to the Tetrathlon BERT model 520 corresponds to "[CLS] Is this sentence toxic or not toxic? [SEP]

Leading doctor reveals the No. 1 worst carb you are eating [SEP]". The Tetrathlon model 500 behaves well if it extracts either "toxic" or "not toxic" from the input.

[75] The single task model 400 illustrated by Fig. 4 and the Tetrathlon model 500 illustrated by Fig. 5 are able to classify sentences for different tasks. Apart from the classification outputted by these models, it is desirable to feed the classifier 302 with extra features. In embodiments implementing the Tetrathlon model 500, the start index logits 531 and end index logits 532 could serve as extra features for the classifier 302 but these logits are only two-dimensional or at best three-dimensional in case the Tetrathlon model 500 is used for the task of sentiment analysis. In the BERT machine learning models 420 and 520, the twelve encoding layers 401-412, 501-512 however each generate a 768-dimensional feature vector. In a possible embodiment of the disinformation detection system, the 768-dimensional feature vector generated by the twelfth encoder layer 412, 512 could be forwarded to the classifier 302. In an alternative embodiment of the disinformation detection system, the twelve 768-dimensional feature vectors could be summed together to form a single 768-dimensional vector that is forwarded to the classifier 302. The best results however are achieved when the four 768-dimensional vectors generated by the last four encoder layers, i.e. encoder layers 9, 10, 11 and 12, are concatenated to form a 3072-dimensional feature vector and this 3072-dimensional feature vector is forwarded to the classifier 302. This is for instance illustrated by Fig. 6 which shows a BERT machine learning model 601, that may correspond for instance to the single task BERT model 420 in Fig. 4 or the Tetrathlon BERT model 520 of Fig. 5, and wherein each encoder layer generates a 768-dimensional feature vector. The first encoder layer generates the 768-dimensional embedding feature vectors F101, F201, F301, ... or 6101, 6201, 6301, ... for respective tokens [CLS], "Leading", "doctor", ...; the second encoder layer generates the 768-dimensional embedding feature vectors F102, F202, F302, ... or 6102, 6202, 6302; ... for respective tokens [CLS], "Leading", "doctor", ...; and the twelfth encoder layer generates the 768-dimensional embedding feature vectors F112, F212, F312, ... or 6112, 6212, 6312, ... for respective tokens [CLS], "Leading", "doctor", The feature concatenator 602 that is supposed to be integrated in embodiments of the feature extraction unit 301 then collects embedding vectors F112, F111, F110 and F109 corresponding to token [CLS] and concatenates these embedding vectors into a single 3072-dimensional feature vector 603 which represents the input sequence.

[76] In order to extract features from an information item, e.g. a news article, the information item is forwarded to four fine-tuned BERT models each trained to execute an individual task. The four 3072-dimensional feature vectors corresponding to the [CLS] token are then taken as the set of features for the information item and forwarded to the classifier 302 for the disinformation assessment. In case of a multi-task Tetrathlon model, the same information item is four times inputted to the Tetrathlon model, each time accompanied by a different question. The four 3072-dimensional feature vectors corresponding to the [CLS] token are also here taken as a set of features for the information item and forwarded to the classifier 302 for the disinformation assessment.

[77] Feature extraction for a cascade of tweets related to a single information item is illustrated by Fig. 7. For each tweet and considering a single task, the feature concatenator 602 extracts a 3072-dimensional feature vector. In case of a twitter cascade of n tweets, this results in a first 3072-dimensional feature vector 701 concatenating embedding feature vectors F_{1121} , F_{1111} , F_{1101} , and F_{1091} , a second 3072-dimensional feature vector 702 concatenating embedding feature vectors F_{1122} , F_{1112} , F_{1102} , and F_{1092} , ..., and an n 'th 3072-dimensional feature vector 70 n concatenating embedding feature vectors F_{112n} , F_{111n} , F_{110n} , and F_{109n} . These n feature vectors are summed together by the adder 710, resulting in a new 3072-dimensional feature vector 70s with features F_{112s} , F_{111s} , F_{110s} , and F_{109s} , and the latter features of feature vector 70s are divided by n , the number of tweets in the Twitter cascade, by the divider 720 resulting in the mean feature vector 70a with features F_{112a} , F_{111a} , F_{110a} , and F_{109a} . Herein, as an example, $F_{112s} = F_{1121} + F_{1122} + \dots + F_{112n}$, and $F_{112a} = F_{112s} / n$. This way a single 3072-dimensional feature vector is generated per task for the complete Twitter cascade associated with a single information item by the cascade feature extractor 700, whereas each tweet is individually supplied to single task BERT models or supplied four times to the Tetrathlon BERT model with four different questions.

[78] Fig. 8 shows a suitable computing system 800 enabling to implement embodiments of the disinformation detection system according to the invention. Computing system 800 may in general be formed as a suitable general-purpose

computer and comprise a bus 810, a processor 802, a local memory 804, one or more optional input interfaces 814, one or more optional output interfaces 816, a communication interface 812, a storage element interface 806, and one or more storage elements 808. Bus 810 may comprise one or more conductors that permit communication among the components of the computing system 800. Processor 802 may include any type of conventional processor or microprocessor that interprets and executes programming instructions. Local memory 804 may include a random access memory (RAM) or another type of dynamic storage device that stores information and instructions for execution by processor 802 and/or a read only memory (ROM) or another type of static storage device that stores static information and instructions for use by processor 802. Input interface 814 may comprise one or more conventional mechanisms that permit an operator or user to input information to the computing device 800, such as a keyboard 820, a mouse 830, a pen, voice recognition and/or biometric mechanisms, a camera, etc. Output interface 816 may comprise one or more conventional mechanisms that output information to the operator or user, such as a display 840, etc. Communication interface 812 may comprise any transceiver-like mechanism such as for example one or more Ethernet interfaces that enables computing system 800 to communicate with other devices and/or systems, for example with other computing devices 881, 882, 883. The communication interface 812 of computing system 800 may be connected to such another computing system by means of a local area network (LAN) or a wide area network (WAN) such as for example the internet. Storage element interface 806 may comprise a storage interface such as for example a Serial Advanced Technology Attachment (SATA) interface or a Small Computer System Interface (SCSI) for connecting bus 810 to one or more storage elements 808, such as one or more local disks, for example SATA disk drives, and control the reading and writing of data to and/or from these storage elements 808. Although the storage element(s) 808 above is/are described as a local disk, in general any other suitable computer-readable media such as a removable magnetic disk, optical storage media such as a CD or DVD, -ROM disk, solid state drives, flash memory cards, ... could be used. It is noticed that the entire method according to the present invention can be executed centralized, e.g. on a server in a management centre or in a cloud system, or it can be partially executed on a remote electronic device, e.g. worn by the user, and partially on a central server. Computing system 800

could thus correspond to the processing system available centrally or the processing system available in the electronic device.

[79] Although the present invention has been illustrated by reference to specific
5 embodiments, it will be apparent to those skilled in the art that the invention is not
limited to the details of the foregoing illustrative embodiments, and that the present
invention may be embodied with various changes and modifications without departing
from the scope thereof. The present embodiments are therefore to be considered in all
10 respects as illustrative and not restrictive, the scope of the invention being indicated
by the appended claims rather than by the foregoing description, and all changes which
come within the meaning and range of equivalency of the claims are therefore intended
to be embraced therein. In other words, it is contemplated to cover any and all
modifications, variations or equivalents that fall within the scope of the basic underlying
principles and whose essential attributes are claimed in this patent application. It will
15 furthermore be understood by the reader of this patent application that the words
"comprising" or "comprise" do not exclude other elements or steps, that the words "a"
or "an" do not exclude a plurality, and that a single element, such as a computer
system, a processor, or another integrated unit may fulfil the functions of several
means recited in the claims. Any reference signs in the claims shall not be construed
20 as limiting the respective claims concerned. The terms "first", "second", "third", "a", "b",
"c", and the like, when used in the description or in the claims are introduced to
distinguish between similar elements or steps and are not necessarily describing a
sequential or chronological order. Similarly, the terms "top", "bottom", "over", "under",
and the like are introduced for descriptive purposes and not necessarily to denote
25 relative positions. It is to be understood that the terms so used are interchangeable
under appropriate circumstances and embodiments of the invention are capable of
operating according to the present invention in other sequences, or in orientations
different from the one(s) described or illustrated above.

30

CLAIMS

1. A disinformation detection system (100) for automated verification of information items, said disinformation detection system (100) comprising an event bus (110) and event-driven microservices (101-106) coupled to said event bus (110), wherein said event-driven microservices (101-106) comprise:
- a scoring microservice (103; 300) configured to execute at least one trained machine learning model (301, 302, 303, 305) adapted to generate a disinformation prediction for each information item forwarded thereto;
 - a training microservice (104) configured to train said at least one machine learning model (301, 302, 303, 305) based on input obtained from researchers; and
 - a monitoring microservice (105; 200) configured to obtain said information items and related data for said information items, and configured to forward said information items and said related data to said scoring microservice (103; 300), wherein said monitoring microservice (105; 200) comprises:
 - a data storage (240);
 - at least one background harvester (235, 236, 237) configured to periodically fetch and store in said data storage (240) information items from a particular information source and related data for said information items from said particular information source;
 - at least one on-demand harvester (221, 222) configured to fetch and store in said data storage (240) an information item and/or related data for said information item in return to a Uniform Resource Locator, abbreviated URL, or other type of query of said information item,
- wherein said at least one trained machine learning model (301, 302, 303, 305) comprise:
- a first Bi-directional Encoder Representations from Transformers model, abbreviated a first BERT model (400), trained for clickbait detection, configured to receive an information item and/or related data as input sequence and to return a probability (432) for said information item and/or related data indicating clickbait;
 - a second BERT model trained for sentiment detection, configured to receive an information item and/or related data as input sequence and to

return a probability for said information item and/or related data indicating sentiment;

- a third BERT model trained for bias detection, configured to receive an information item and/or related data as input sequence and to return a probability for said information item and/or related data indicating bias; and
- a fourth BERT model trained for toxicity detection configured to receive an information item and/or related data as input sequence and to return a probability for said information item and/or related data indicating toxicity,

or, alternatively, wherein said at least one trained machine learning model (302, 302, 303) comprise:

- an N-athlon model (500), N being an integer plurality, said N-athlon model being configured to receive an information item and/or related data together with one out of N possible questions as input sequence, and to return an answer to said question and a start index (551) and end index (552) wherein the range [start index, end index] indicates where in said information item and/or related data the relevant part is found motivating said answer to said question.

2. A disinformation detection system (100) according to claim 1, wherein said at least one background harvester (235, 236, 237) comprise one or more of:

- a Really Simple Syndication collector (235), abbreviated RSS collector, configured to periodically fetch and store information items from RSS sources specified in a configuration file;

- a cascade stream harvester (236), configured to periodically fetch and store Twitter cascades for information items whose URL is specified in a cascades link list;

- a financial data harvester (237), configured to periodically fetch and store financial data related to an information item from at least one predetermined financial data source;

- a social network data harvester (237), configured to periodically fetch and store social network data related to an information item from at least one predetermined social network;

- a fact checking data harvester (237), configured to periodically fetch and store fact checking data related to an information item from at least one fact checking data source.

3. A disinformation detection system (100) according to one of the preceding claims, wherein said at least one on-demand harvester (221, 222) comprise one or more of:

- 5 - an on-demand information item harvester (221), configured to fetch and store the content of an information item in return to said URL of said information item;
- an on-demand cascade harvester (222), configured to fetch and store a Twitter cascade in return to said URL of said information item.

- 10 4. A disinformation detection system (100) according to claim 1,
- wherein said first BERT model (400), said second BERT model, said third BERT model and said fourth BERT model each comprise twelve layers (401-412) configured to respectively generate twelve 768 dimensional feature vectors; and
- wherein said scoring microservice (300) is further configured to concatenate the
- 15 four 768 dimensional feature vectors generated by the last four layers of said twelve layers to obtain a 3072 dimensional associated feature vector (603) for each BERT model.

5. A disinformation detection system (100) according to one of the preceding
- 20 claims,
- wherein said monitoring microservice (105) is configured to forward an information item as input sequence to each one of said first machine learning model (400), said second machine learning model, said third machine learning model and said fourth machine learning model.

- 25 6. A disinformation detection system (100) according to one of the preceding claims,

- wherein said monitoring microservice (105) is configured to forward each tweet message of a Twitter cascade as input sequence to said first machine learning model
- 30 (400), said second machine learning model, said third machine learning model and said fourth machine learning model; and

 wherein said scoring microservice (300) is further configured to obtain for each one of said first machine learning model (400), said second machine learning model, said third machine learning model and said fourth machine learning model an

associated feature vector (70a) that corresponds to the mean value of associated feature vectors (701-70n) obtained for individual tweet messages of said Twitter cascade.

5 7. A disinformation detection system (100) according to claim 1, wherein N corresponds to four, and wherein said N-athlon model (500) corresponds to a Tetrathlon model trained for clickbait detection, sentiment detection, bias detection and toxicity detection, said Tetrathlon model (500) being configured to receive an information item and/or related data together with a question as input sequence and to
10 return a start index (551) and end index (552) of parts of said information item and/or related data respectively indicating clickbait, sentiment, bias or toxicity.

 8. A disinformation detection system (100) according to claim 7,
 - wherein said Tetrathlon model (500) comprises twelve layers (501-512)
15 configured to respectively generate twelve 768 dimensional feature vectors; and
 - wherein said scoring microservice (103; 300) is further configured to concatenate the four 768 dimensional feature vectors generated by the last four layers of said twelve layers (501-512) to obtain a 3072 dimensional associated feature vector for said input sequence.

20 9. A disinformation detection system (100) according to claim 7 or 8,
 wherein said monitoring microservice (105; 200) is configured to forward an information item together with the question "is this sentence clickbait or not clickbait?" as a first input sequence to said Tetrathlon model (500), to forward said information
25 item together with the question "is this sentence positive, neutral or negative?" as a second input sequence to said Tetrathlon model (500), to forward said information item together with the question "is this sentence biased or not biased?" as a third input sequence to said Tetrathlon model (500), and to forward said information item together with the question "is this sentence toxic or not toxic?" as a fourth input sequence to
30 said Tetrathlon model (500).

 10. A disinformation detection system (100) according to claim 7 or 8,
 wherein said monitoring microservice (105; 200) is configured to forward each tweet message of a Twitter cascade together with the question "is this sentence

clickbait or not clickbait?" as a first input sequence to said Tetrathlon model (500), together with the question "is this sentence positive, neutral or negative?" as a second input sequence to said Tetrathlon model (500), together with the question "is this sentence biased or not biased?" as a third input sequence to said Tetrathlon model (500), and together with the question "is this sentence toxic or not toxic?" as a fourth input sequence to said Tetrathlon model (500); and

wherein said scoring microservice (103; 300) is further configured to obtain four associated feature vectors (70a) as mean value of associated feature vectors for respectively first input sequences comprising tweet messages of said Twitter cascade, second input sequences comprising tweet messages of said Twitter cascade, third input sequences comprising tweet messages of said Twitter cascade, and fourth input sequences comprising tweet messages of said Twitter cascade.

11. A disinformation detection system (100) according to one of the preceding claims,

wherein said scoring microservice (103; 300) is further configured to extract time series from said related data for an information item.

12. A disinformation detection system (100) according to one of the preceding claims,

wherein said scoring microservice (103; 300) is further configured to extract social network graphs from said related data for an information item; and

wherein said scoring service (103; 300) comprises a machine learning model (305) trained to learn a representation of a social network graph.

13. A disinformation detection system (100) according to one of the preceding claims, further comprising a display unit configured to generate one or more of following visuals for an information item:

- said information item wherein parts of said information item detected as clickbait, sentiment, bias, or toxic are marked;

- related data of said information item wherein parts of said related data detected as clickbait, sentiment, bias, or toxic are marked;

- a social network graph showing relations and/or locations of users that interact with said information item;

- evolution in time of said social network graph.

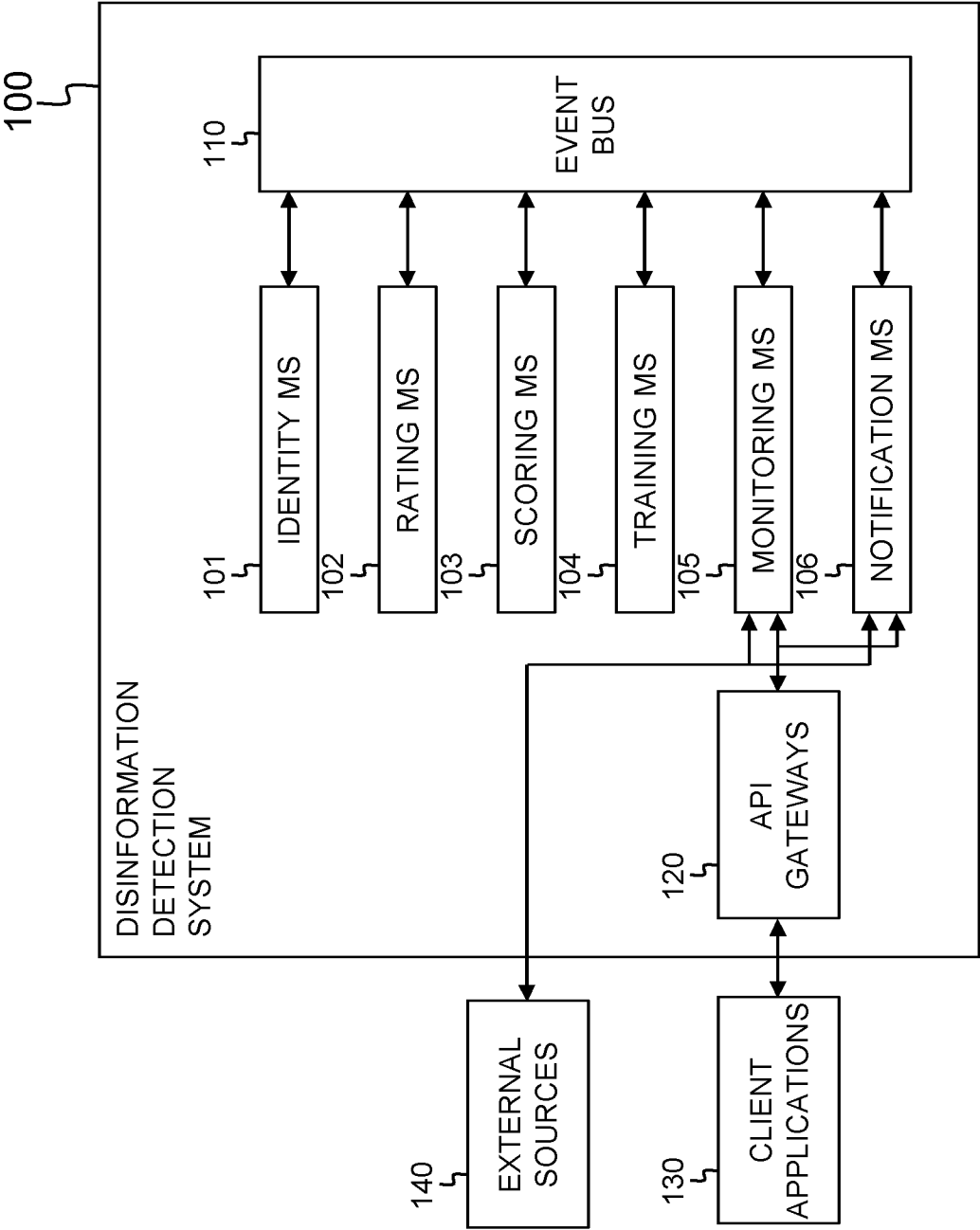


Fig. 1

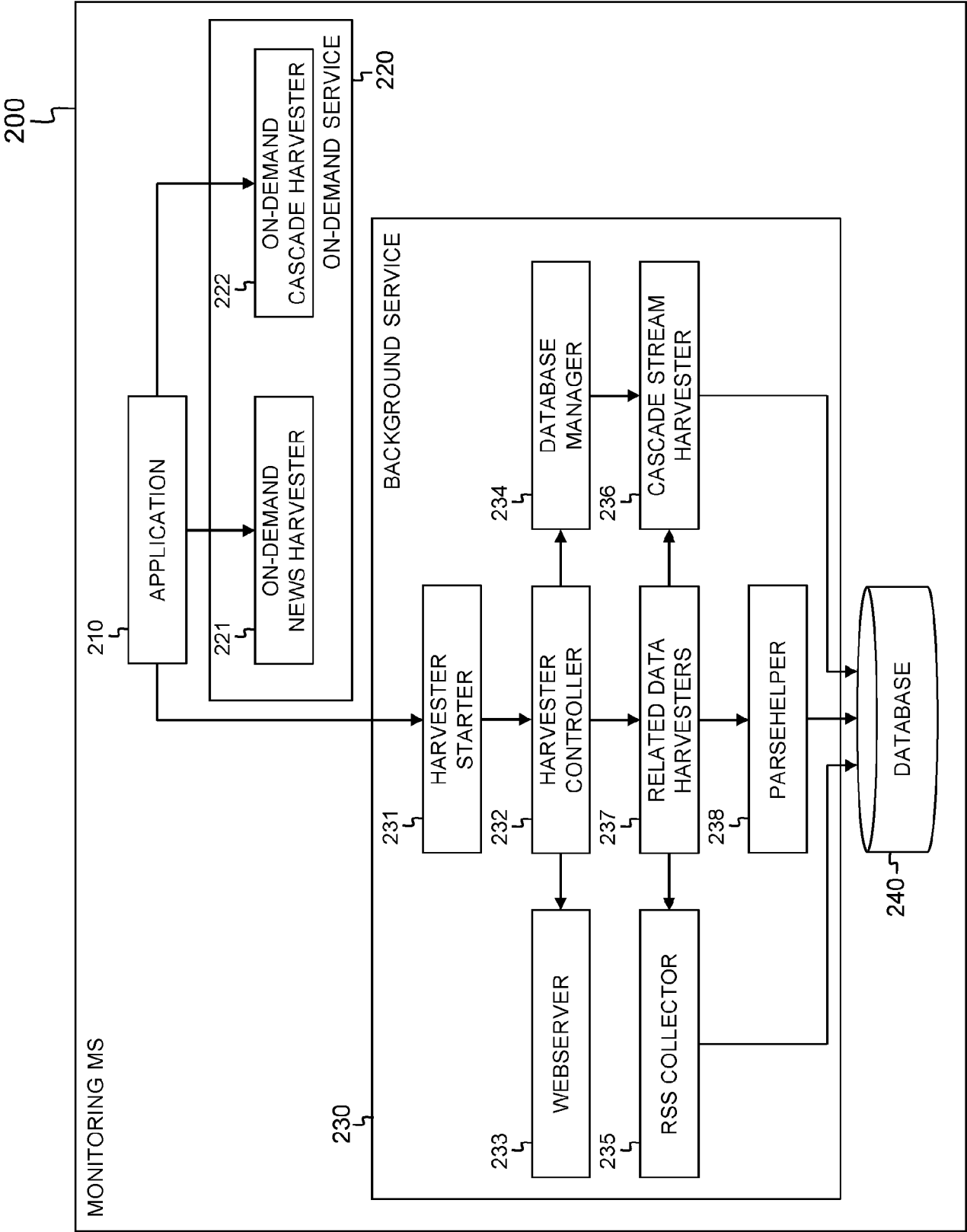


Fig. 2

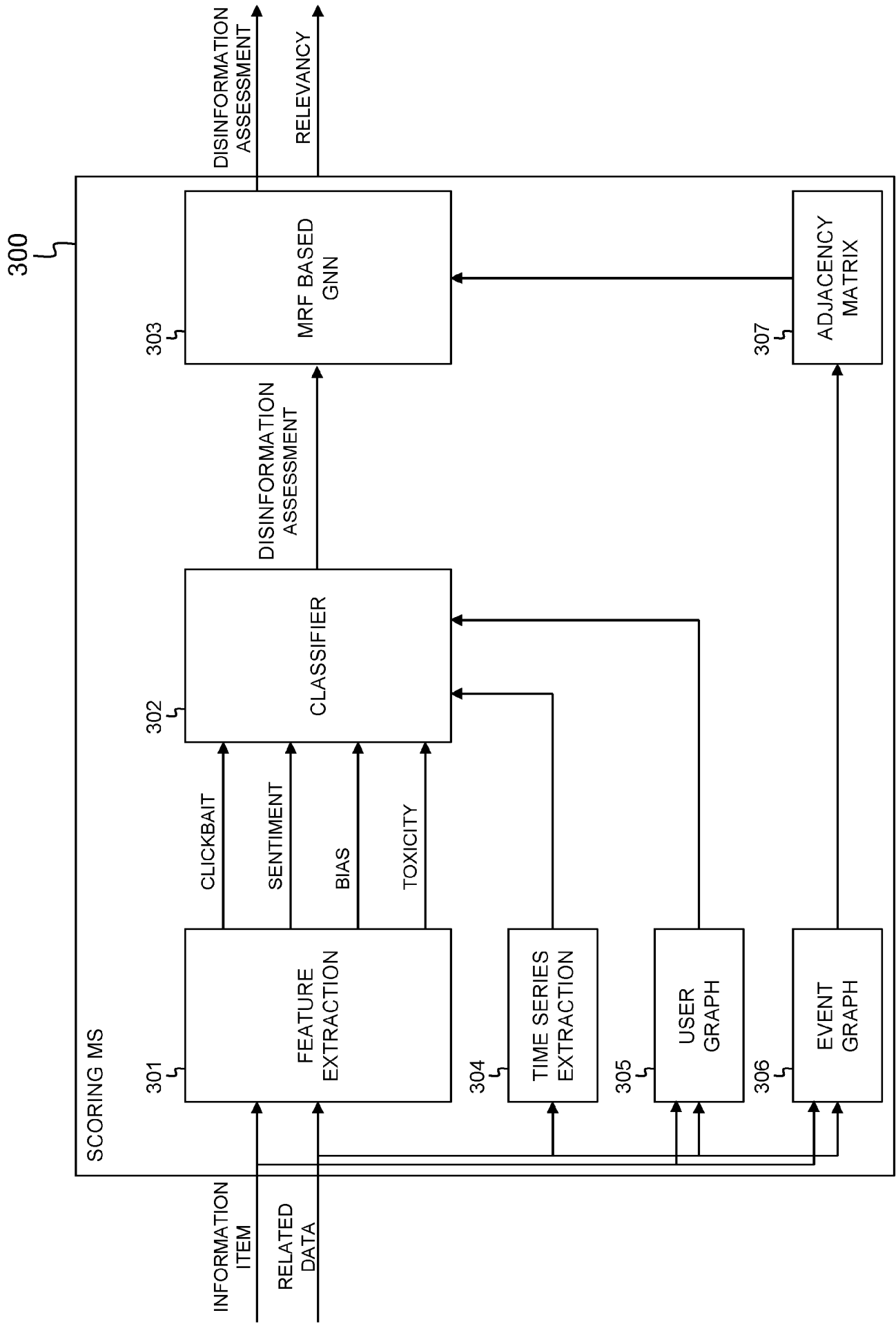


Fig. 3

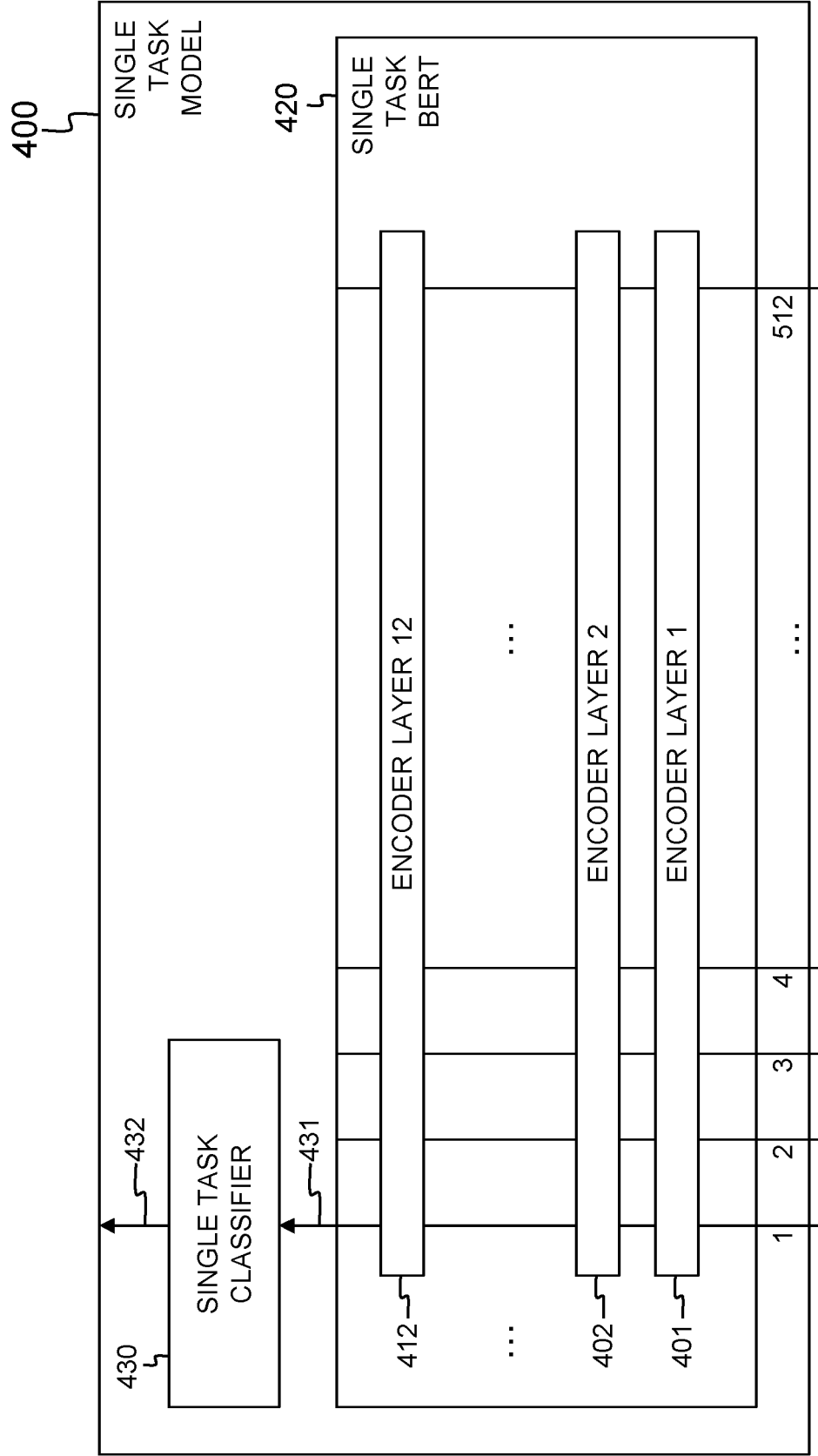


Fig. 4

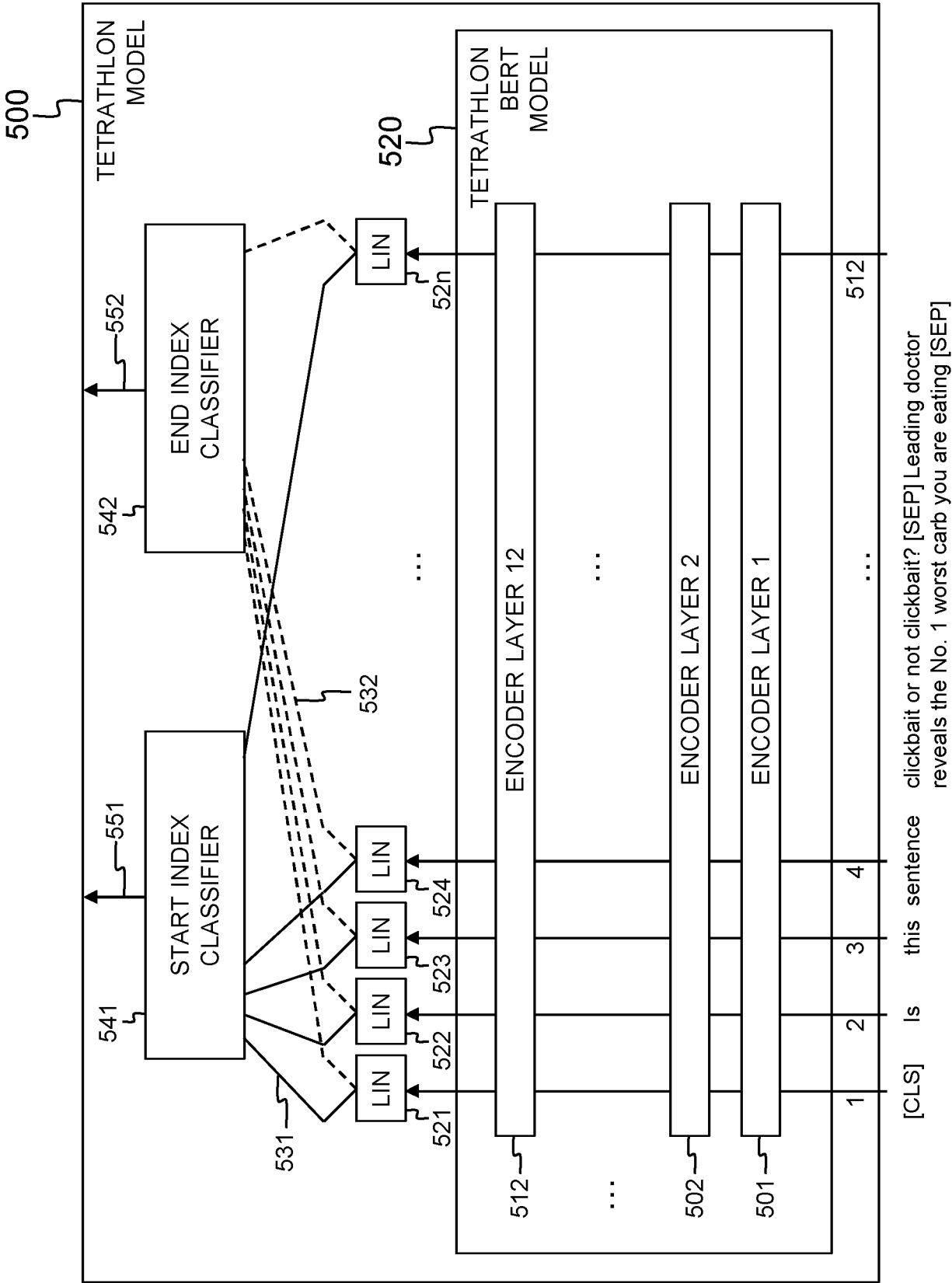


Fig. 5

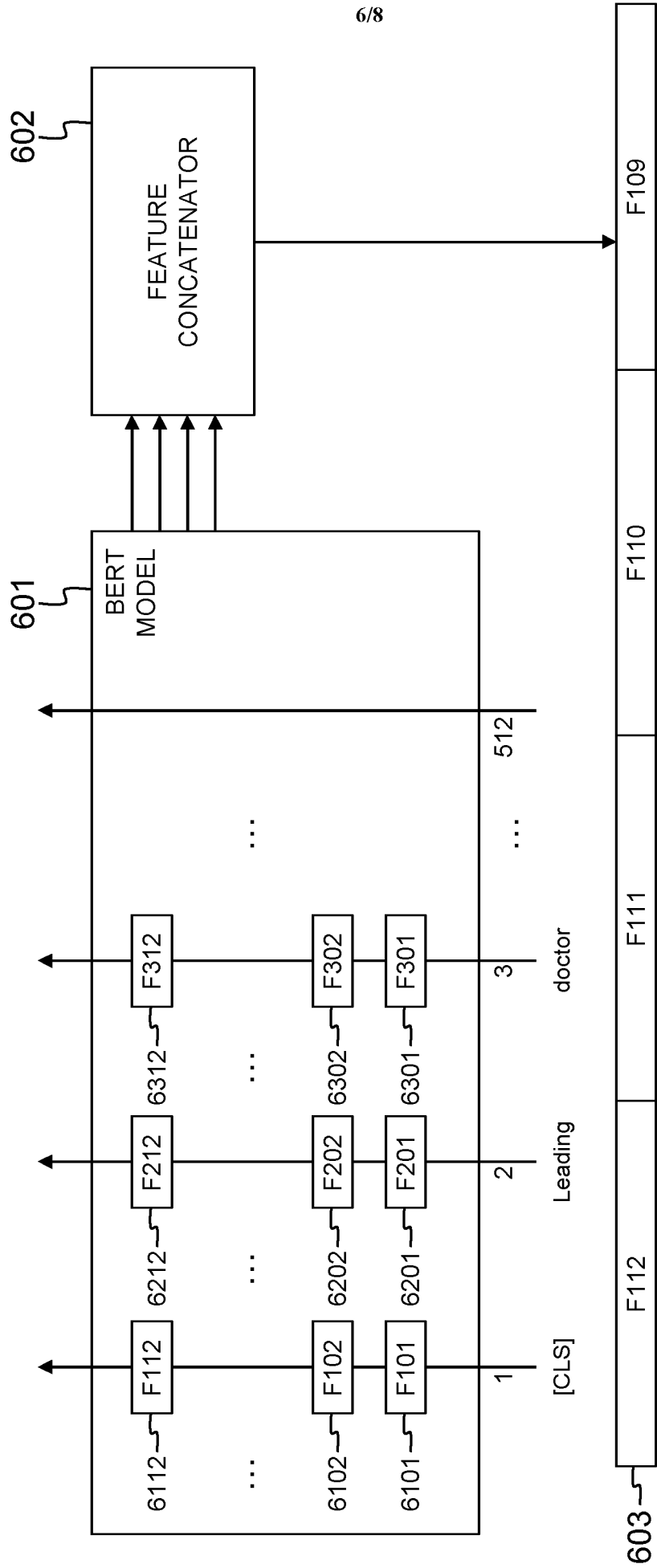


Fig. 6

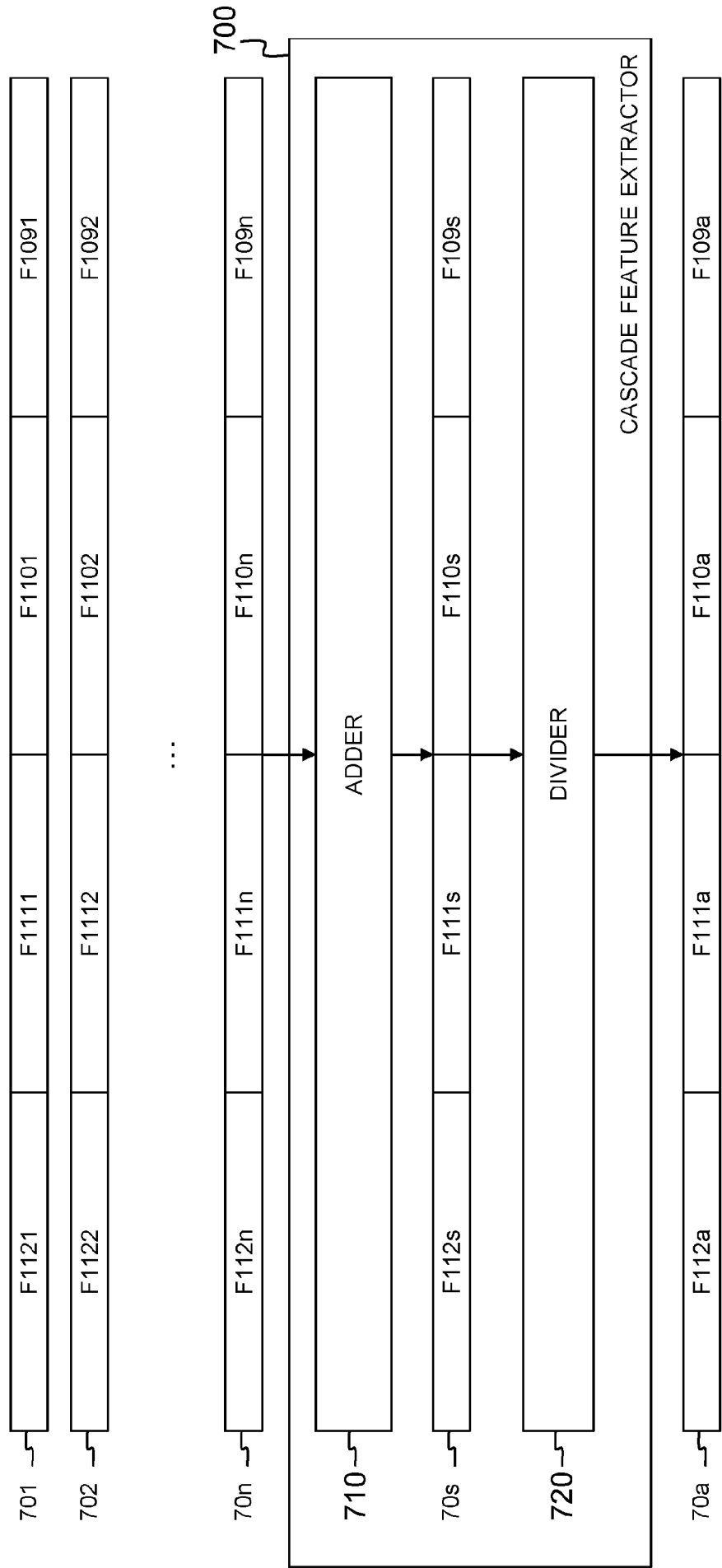


Fig. 7

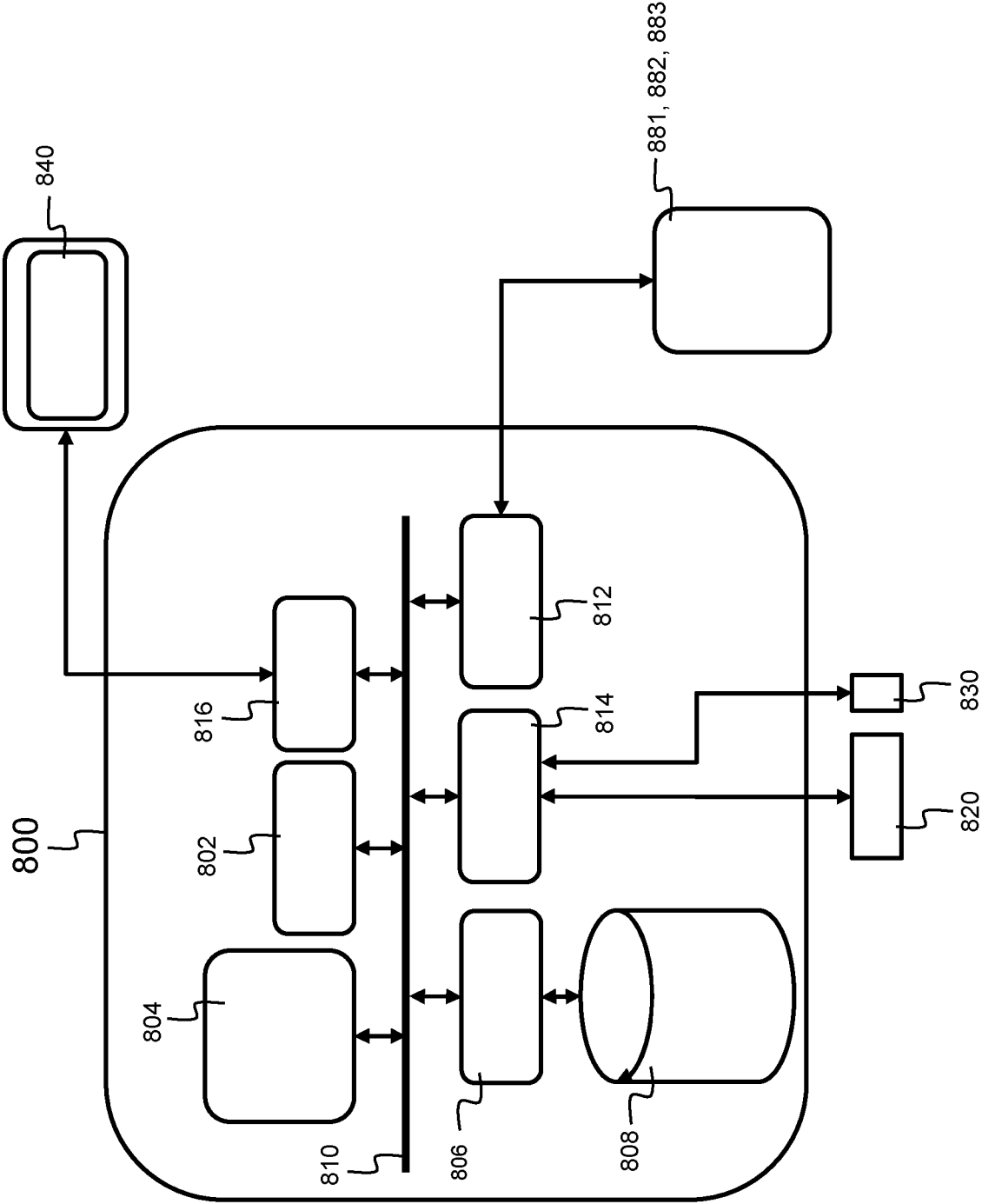


Fig. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2022/051831

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F16/35 G06F40/30 G06Q30/00 G06Q30/02 G06Q30/06 G06Q50/00 ADD. According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06F Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Pereira Rihan Stephen: "WHIRLPOOL: A microservice style scalable continuous topical web crawler", 31 December 2019 (2019-12-31), pages 1-108, XP055806830, Retrieved from the Internet: URL:https://prof.msoltys.com/wp-content/uploads/2020/02/final_draft_rihan_pereira_v1.5.pdf [retrieved on 2021-05-24] page 11 page 16 – page 17 page 23 page 50 – page 51 page 93 page 101 – page 103 -----	1-13
☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 29 April 2022		Date of mailing of the international search report 10/05/2022
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Rameseder, Jonathan