WIP - SKOD: A Framework for Situational Knowledge on Demand*

Servio Palacios**[1], KMA Solaiman**[1], Pelin Angin[2], Alina Nesen[1], Bharat Bhargava[1], Zachary Collins[3], Aaron Sipser[3], Michael Stonebraker[3], and James Macdonald[4]

[1] Purdue University, West Lafayette, IN 47906, USA
{spalacio,ksolaima,anesen,bbshail}@purdue.edu
[2] METU, Ankara, Turkey pangin@ceng.metu.edu.tr
[3] MIT CSAIL, Cambridge, MA 02139, USA
{zcollins,asipser,stonebraker}@csail.mit.edu
[4] Northrop Grumman Corporation jim.macdonald@ngc.com

Abstract. Extracting relevant patterns from heterogeneous data streams poses significant computational and analytical challenges. Further, identifying such patterns and pushing analogous content to interested parties according to mission needs in real-time is a difficult problem. This paper presents the design of SKOD, a novel Situational Knowledge Query Engine that continuously builds a multi-modal relational knowledge base using SQL queries; SKOD pushes dynamic content to relevant users through triggers based on modeling of users' interests. SKOD is a scalable, real-time, on-demand situational knowledge extraction and dissemination framework that processes streams of multi-modal data utilizing publish/subscribe stream engines. The initial prototype of SKOD uses deep neural networks and natural language processing techniques to extract and model relevant objects from video streams and topics, entities and events from unstructured text resources such as Twitter and news articles. Through its extensible architecture, SKOD aims to provide a high-performance, generic framework for situational knowledge on demand, supporting effective information retrieval for evolving missions.

Keywords: Query engine · multi-modal information retrieval · knowledge base · stream processing · targeted information dissemination.

1 Introduction

The past decade has witnessed an unprecedented *volume* of data being generated by a *variety* of sources at very high *velocity*, resulting in the rise of the *big data* paradigm. Specifically, the developments in social networks and Internet of

* This research is supported by Northrop Grumman Mission Systems University Research Program.

** **Both authors contributed equally and are considered to be co-first authors.**

Things (IoT) have created a plethora of multi-modal data sources that generate billions of data records every second, only a small fraction of which readily translates into useful information. While the availability of such vast amounts of data has made it possible to build large knowledge bases, on-demand extraction of highly relevant situational knowledge for specific missions from those heterogeneous data clouds remains a difficult task for the following reasons: (1) Accurate correlation of data from different resources for billions of data items is a daunting task; (2) A knowledge base built upon a specific ontology may not cater to the needs of a mission when additional mission requirements/user interests are defined later; (3) The storage of the most relevant data in the knowledge base is essential to avoid performance degradation with growing data; (4) Generalization of knowledge bases irrespective of mission needs is a challenge.

Many critical missions will require real-time targeted dissemination of information to interested parties as information becomes available. Achieving high-performance, accurate information extraction and propagation requires (1) accurate modeling of the different users' interests; (2) application of intelligent filters on streaming data to capture and correlate the most relevant aspects; (3) triggers for communicating the gathered information to the interested parties.

In this paper, we propose SKOD, a framework for situational knowledge on demand, which provides high-performance real-time analysis of streaming data of multiple modalities from different sources to dynamically and continuously build mission-specific knowledge bases. In order to capture data most relevant to user needs, SKOD uses past user query patterns to construct the knowledge base.

Our approach provides a scalable solution for modeling different user interests over vast amounts of data while allowing flexibility for future incoming data. Additional interests can immediately be integrated by defining new queries on the knowledge base. SKOD currently handles pattern extraction from streaming video and text data, but the extensible architecture allows facile integration of additional data modalities such as audio, sensor data, signals, and others.

2 Model

2.1 Example Application Scenario

In order to clearly illustrate the objectives and operation of SKOD, we describe an example application scenario of the system in this section. Let us consider a city information system, which provides access to multiple agents (e.g., police, public works department, citizens, emergency personnel, homeland security) with varying missions, hence varying information needs. In such a system, while the police would be interested in patterns such as unsafe lane changes, locations visited by a suspicious person, to name a few; the public works department would be interested in patterns such as potholes and occluded street signs. An example query to be submitted to this system by a police officer is:

Q1: *List cars parked next to fire hydrants illegally today.*

To answer Q1, we will require detecting cars and fire hydrants in video frames and tweets, given the available data sources are city surveillance cameras and Twitter. The query response will provide information that the policeman will always be interested in, therefore as new data streams in, patterns matching the query should be communicated to the policeman and other police officers as well, due to the similarity of their profiles to the user submitting the query. A different user of the system (a firefighter) can later submit **Q2:** *Get locations of leaking fire hydrants*. While this query will be able to utilize the knowledge base created in response to Q1, it will build upon it to find patterns of the act *leak* in both data sources as they stream additional data to the system.

2.2 SKOD System Architecture

The SKOD architecture consists of three large modules - 1) streaming platform to handle the vast amount of heterogeneous incoming data, 2) multi-modal query engine to model the user interest based on their previous queries and 3) the front end with the indexing layer. The query engine also accommodates the unit for feature-analysis of heterogeneous data for identifying personalized events. SKOD includes fixed queries on data streams from multiple sources, both separate and combined. The queries are then stored to build the knowledge base, which in return models the user interests. SKOD can provide users with information similar to their previous queries as well as missing information on their existing information. This information is delivered to the user using trigger events in the relational database. Similar queries and repeated accesses to similar data are cached to provide better throughput. The front-end queries an indexing layer based on Lucene indexes to improve throughput. In Figure 1, we show an overview of SKOD's architecture. We describe the three modules below.

Streaming Broker Due to the latency-sensitive set of applications that SKOD aims to tackle to consume data from heterogeneous sources, this work relies on Apache Kafka to expose a real-time stream processing pipeline. Apache Kafka is a scalable and fault-tolerant publish-subscribe messaging system. Kafka achieves the capability to store and process data from multiple applications (producers) through a topic abstraction system. As an output, multiple applications can consume the inserted data from all the producers asynchronously and without any loss. The producers/consumers abstraction allows SKOD architecture to provide real-time analysis and recommendation capability. Apache Kafka features allow to store the raw incoming data in Postgres and consume the same data by text and video processing applications simultaneously.
Currently SKOD architecture consumes both RESTful, and streaming data from Twitter and video feeds through Kafka. SKOD is capable of integrating data from other real-time applications (i.e., sensor, audio, files, JDBC) through Kafka Clients or Kafka Connect. Kafka Clients allow to pass and retrieve messages directly to and from Kafka as long as the message can be converted to bytes. We show a detailed view of SKOD data streaming pipeline in Figure 3 for different types of Twitter data.

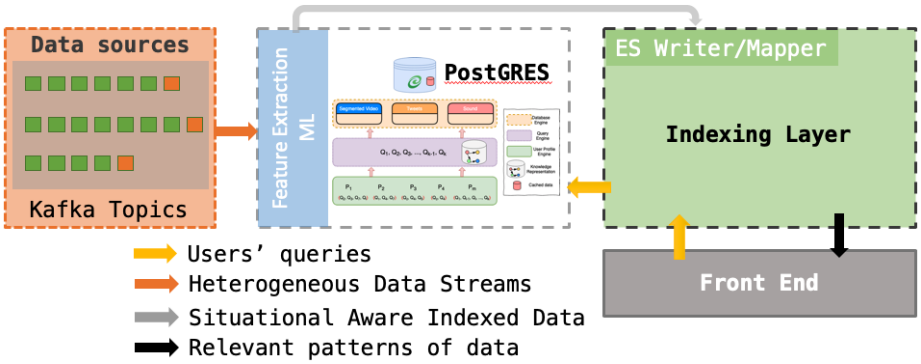

Fig. 1. SKOD Architecture. Kafka Topics Partition Layout diagram.

Multi-modal Query Engine The multi-modal query engine consists of several sub-modules. The first sub-module consumes the streams of data provided by the *streaming broker* and stores them directly in the relational database (Postgres). The second sub-module extracts features from each mode of data with a separate processing unit. For our current implementation, we focus on processing video and unstructured text to extract features relevant to most domains. We explain these processes in section 2.3 and 2.4. In the final module of the multi-modal query engine, SKOD utilizes users' SQL queries to build the knowledge base on top of a relational database and pushes relevant content to users without user intervention. We explain this module in detail in section 2.5. In Figure 1 and Figure 2 we observe the overall architecture.

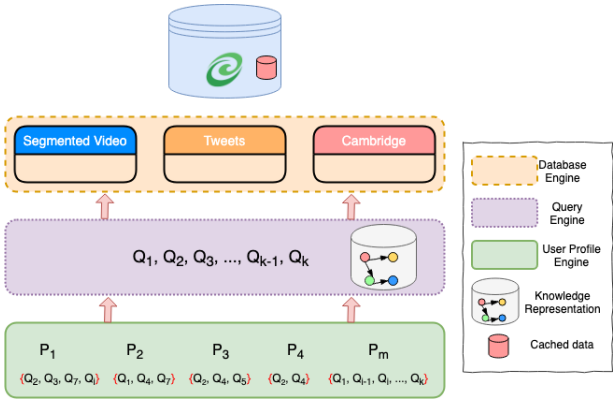

Fig. 2. Multimodal Query Engine Representation utilizing Situational Knowledge.

Indexing Elasticsearch is a distributed indexing and search engine. SKOD queries Elasticsearch through a RESTful API. Moreover, Elasticsearch utilizes Lucene indexes under the hood. Naturally, Elasticsearch achieves fast response times because it queries an index instead of querying text or video directly. The basic structure is called *Document*. Elasticsearch returns a ranked set of results according to the relevance of the query. SKOD uses Elasticsearch to rank relevant content to push to the end user.

2.3 Feature extraction from video streams

Video data represents a separate and unique modality in the SKOD multi-modal system for storing and extracting knowledge on demand. Video data comes in large amounts, unstructured, and raw video is unlabeled, frequently in need of processing, cleaning, and preparing for the next stage in the data flow.

Video can be viewed as a sequence of frames, where each frame is characterized by its bitmap that can later be transformed into a multidimensional array or a tensor. The need to work with extensive digital representations requires specific ways of storing and operating with the video data, which are different from those of text and structured data. When the knowledge must be extracted efficiently on demand from the heterogeneous multi-modal database, there are several challenges to be resolved: (1) Entities from each frame have to be accessible for user queries, user-defined stored procedures, and event triggers; (2) For connecting with other modalities in a poly-store environment, these entities must be stored in a way that they can be matched with the text data and text metadata as well as data from other modalities for further analysis; (3) There must be a way to obtain entities in an ad-hoc manner to extract knowledge from streams of video. We resolve these challenges utilizing two off-the-shelf solutions: Apache Kafka for streaming video in a scalable, fault-tolerant manner and the YOLOv3 real-time object detection system [18].

2.4 Feature extraction from unstructured text

Data Collection and Initial Processing For unstructured text, currently SKOD processes tweets. There are two types of tweet data available for scraping - RESTful data (historic data) and streaming data. SKOD uses Twitter search API to collect RESTful data and Twitter streaming API for collecting real-time tweet streams. It creates independent docker containers for the producers, which can take tags and timelines as environment variables and run simultaneously. Since there can be overlap of tweet data from multiple producers, SKOD uses the Kafka streaming platform to handle the asynchronous, scalable and fault tolerant flow of tweets using the same topic abstraction for all. After the data is in Kafka, SKOD uses two separate consumers - 1) to parse and populate Postgres with the tweet and associated metadata, and 2) to pass the raw tweets to a feature extraction engine. Figure 3 shows an overview of the architecture.

Fig. 3. Data Streaming Pipeline from Restful and Streaming tweets to applications.

Feature extraction Understanding unstructured texts has always been a daunting task. Even with the recent rise of language models it is hard to parse unstructured social texts into domain-independent features.

We first preprocess the text using Stanford CoreNLP [13], extract named entities and co-reference chains. Then we create a separate table in Postgres to save each tweet with its associated named entities i.e, LOCATION, ORGANIZATION, PERSON, saving them as text arrays and associated topic with the tweet. Further, we create another column `objects`, which are any words in the tweet except stop words and the ones identified in named entities.

2.5 Knowledge Representation

Unified knowledge representation for all streamed data is required for the query engine to extract useful knowledge and disseminate information to relevant parties efficiently. In SKOD, we represent knowledge using relational data and SQL queries on the data, which persist for the lifetime of the knowledge base and grow with additional user interests. Representation of textual data such as tweets and online news is more straightforward through the extraction of topics and keywords; which can directly be entered into the corresponding columns in the RDBMS tables. Multimedia data such as video and audio are represented both with the binary data and the text data extracted as a result of the processing performed on the binary data. The stored data also includes available metadata for all modalities, such as timestamp, geolocation, and some others. The metadata is especially useful when correlating multiple forms of data for the same events.

The schemas of the PostgreSQL tables storing the extracted features from the tweet text and video frames are as follows:

```
TWEETS(tweet_id INT,
    locations VARCHAR(100)[],
    objects VARCHAR(100)[],
    organizations VARCHAR(100)[],
    persons VARCHAR(100)[],
    dates VARCHAR(100)[],
    times VARCHAR(100)[],
    topic VARCHAR(100)[]
    created_at DATE)
```

```
VIDEO_FRAMES(video_id INT,
    frame_id INT,
    locations VARCHAR(100),
    objects VARCHAR(100)[],
    people VARCHAR(100)[],
    timestamp DATE,
    image BYTEA)
```

Here locations, organizations, and persons are different classes of named entities and other classes can be defined as necessary. Typical attributes are used to facilitate joins between the tables for data correlation. Attributes in different tables may have different names, but have commonalities, i.e., timestamp and created_at, or people and persons. Given the initial knowledge base is built upon **Q1** mentioned in Section 2.1, new streams of video data will result in running the object detector for cars and fire hydrants, and the extracted data will be inserted into the database. Similarly for streaming Twitter data, tweets that have the objects car and fire hydrant will be inserted into the relevant table.

Q1 for a system with these two data sources will translate into multiple SQL queries for the situational knowledge query engine:

```
SELECT video_id, frame_id
FROM VIDEO_FRAMES
WHERE 'car' = ANY(objects)
AND 'fire hydrant'= ANY(objects)
```

```
SELECT tweet_id
FROM TWEETS
WHERE 'car' = ANY(objects)
AND 'fire hydrant'= ANY(objects)
```

```
SELECT t.tweet_id, v.video_id, v.frame_id
FROM TWEETS t, VIDEO_FRAMES v
WHERE 'car' = ANY(t.objects) AND 'fire hydrant' = ANY(t.objects)
AND 'car' = ANY(v.objects) AND 'fire hydrant' = ANY(v.objects)
AND v.location = ANY(t.locations)
```

As data from either resource is streaming in, patterns matching these queries will create triggers for relevant data to be communicated to interested users. Note that the complete system requires translation of natural language questions into SQL queries through entity recognition, and constructs for creating all related queries given the tables for different data sources and their common attributes. Although this initial design is limited to recognition of objects, a richer knowledge base will require incorporation of activity recognition in videos and tweets.

3 Implementation

3.1 Twitter Data Collection and Feature Extraction

Since the city of Cambridge was the point-of-focus for the data used in this work, the target was to collect a million tweets that discuss events and entities

in Cambridge, MA along with all the metadata from Twitter. Twitter data can be collected by hashtags, user timelines, geo-data, and general queries. In SKOD, we chose to search by hashtags and user timelines. For that purpose, about 15 hashtags and 15 user timelines were manually selected after going through profiles in timelines and descriptions for hashtags. For example, @CambridgePolice warns about any possible crimes or street law changes, while @bostonfire talks about fire-related incidents in Boston. At a much broader scale, hashtags like #CambMA include all tweets by many Cambridge, MA departments.

For the implementation of twitter APIs, SKOD uses tweepy.api[5]. There is a class method API() which allows to search by both hashtags and timelines by providing a wrapper for twitter APIs. The Twitter streaming API is used to download twitter messages in real time. In Tweepy, an instance of tweepy.Stream establishes a streaming session and routes messages to StreamListener instance by allowing a connection to twitter streaming API.

Currently, we have around 80K tweets in Postgres. More are being accumulated as the module keeps running. The consumers inherit twitter data as JSON messages (Fig. 4). The JSON message is parsed to extract relevant metadata. Different types of tweets are identified, i.e., original, retweet, and quoted tweets. The tweet text with all the parsed metadata along with the original JSON message is saved in Postgres. With the tweet text, we obtain a social network connected by retweets and follows.

The feature extraction process from the tweet text is explained in section 2.4. We ran the pretrained 7 class NER CRFs from Stanford toolkit [13] to identify the entities. For topic extraction, SKOD uses the Latent Dirichlet Allocation (LDA) method [4]. We show the schema of the PostgreSQL table storing the extracted features from the Tweet text in section 2.5. SKOD wraps the producers and consumers in docker containers. The producers and consumers take the Kafka hostname and port number as input, along with the tags and timelines in files.


```
{'created_at': 'Mon Mar 04 20:31:28 +0000 2019', 'id': 1102667899926401025, 'id_str': '1102667899926401025', 'full_text': "RT @cambridgepl: We
are updating our strategic plan and we want to know what's important to our community! Share your ideas by taking our b…", 'truncated': False,
'display_text_range': [0, 140], 'entities': {'hashtags': [], 'symbols': [], 'user_mentions': [{'screen_name': 'cambridgepl', 'name':
'CambridgePL', 'id': 32080992, 'id_str': '32080992', 'indices': [3, 15]}], 'urls': []}, 'metadata': {'iso_language_code': 'en', 'result_type':
'recent'}, 'source': '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>', 'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None, 'in_reply_to_user_id': None, 'in_reply_to_user_id_str': None, 'in_reply_to_screen_name': None, 'user': {'id':
93219027, 'id_str': '93219027', 'name': 'CPL Teen Room', 'screen_name': 'cplteenroom', 'location': 'Cambridge, MA', 'description': 'the blog of
the Cambridge Public Library Teen Room...', 'url': 'http://t.co/QynRPPtLZX', 'entities': {'url': {'urls': [{'url': 'http://t.co/QynRPPtLZX',
'expanded_url': 'http://cambridgeteenroom.tumblr.com/', 'display_url': 'cambridgeteenroom.tumblr.com', 'indices': [0, 22]}]}, 'description':
{'urls': []}}, 'protected': False, 'followers_count': 374, 'friends_count': 454, 'listed_count': 33, 'created_at': 'Sat Nov 28 17:01:23 +0000
2009', 'favourites_count': 1551, 'utc_offset': None, 'time_zone': None, 'geo_enabled': False, 'verified': False, 'statuses_count': 2482, 'lang':
'en', 'contributors_enabled': False, 'is_translator': False, 'is_translation_enabled': False, 'profile_background_color': '1A1B1F',
```


Fig. 4. Snippet of twitter JSON data.

3.2 Feature Extraction from Video

Real-time video broadcasting in a massively scaled and distributed system requires architectural solutions that are resilient to node failures and supportive

[5] http://docs.tweepy.org/en/v3.5.0/api.html

for automatic recovery. The video data may come with metadata such as geolocation, the movement speed and IP address of the camera, and timestamp; therefore, the message broker needs to scale horizontally as well. In SKOD, Apache Kafka utilizes different topics that represent categories for different modalities of the data.

Similarly, producers transform the videos into a stream of frames and metadata with OpenCV. Then, the consumers read messages from topics and forward the data for computation and knowledge extraction. In the prototype implementation, SKOD uses a universal pre-trained neural network as a tool for object extraction and recognition in the video data. SKOD's video processing feature differentiates between 150 object classes. SKOD identifies the objects in the video on a frame-by-frame basis. Each frame is divided into several regions that are classified by the neural network to assign the most probable class along with a confidence score; this helps to establish the exact boundaries of the objects in the frame. The non-maximum suppression algorithm dismisses all the proposed bounding boxes where the confidence score is not the maximum one. Thus, the approach allows assigning classes and boundaries at the same time in one run. The result obtained for a particular video frame in the collected Cambridge dataset using the proposed neural network architecture is shown in Figure 5. For each processed frame, the recognized data and metadata are stored in the RDBMS and can be used for queries that involve the video data modality.

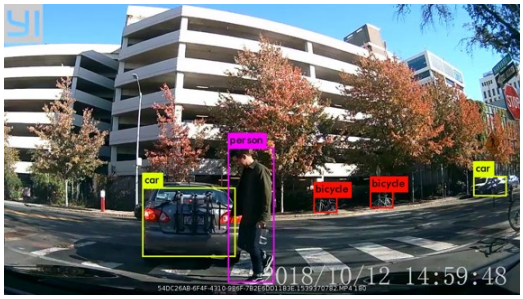

Fig. 5. Result of applying the pre-trained neural network to the Cambridge dataset.

3.3 Front End and Indexing Layer

The front-end utilizes React[6], which is a JavaScript library for building user interfaces. Also, we manage states and side effects using the Cerebral[7] library. We leverage interactive maps via the Leaflet[8] library integrated with React and Cerebral. SKOD caches the most frequent queries to provide faster response times. SKOD's architecture comprises a set of Node.js and python microservices, i.e., Docker containers. In Figure 6, we demonstrate the integration of multimodality combining the extracted Twitter data with the front-end (we utilize GPS coordinates in the Twitter data in GeoJSON format to render the Twitter data in

[6] https://reactjs.org/
[7] https://github.com/cerebral/cerebral
[8] https://leafletjs.com/

the Leaflet map). The Tweets come through the Apache Kafka broker. Then the data is stored in the backend (Postgres). Finally, the Web application queries the indexing layer and it also watches for new changes utilizing WebSockets[9]. SKOD provides an additional layer of cache storing content in the browser using PouchDB[10] similar to the OADA cache library[11]. SKOD future releases include the creation of an elastic cache-layer building a rich set of network topologies on the edge of the network utilizing Web Browsers with Real-Time Communication (WebRTC[12]) [16].

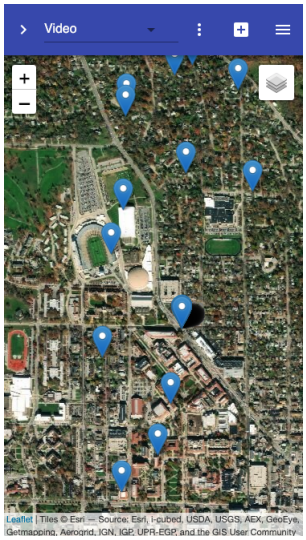

Fig. 6. Situational Knowledge on Demand proof-of-concept. Incoming streams of data shown in a Leaflet map.

4 Related Work

The rise of the big data paradigm in the past decade has resulted in a variety of approaches for processing and fusion of data of multiple modalities to extract useful knowledge. Poria et al. proposed an approach for fusing audio, visual and textual data for sentiment analysis [17]. Foresti et al. introduced a socio-mobile and sensor data fusion approach for emergency response to disasters [8]. Meditskos et al. developed a system for multi-modal fusion of data including language

[9] The Web application was developed utilizing ideas from the OATS Center at Purdue. In particular, the OADA framework https://github.com/OADA
[10] https://pouchdb.com/
[11] https://github.com/OADA/oada-cache
[12] https://webrtc.org/

analysis results, and gestures captured from multimedia data streams to provide situational awareness in healthcare [14]. Adjali et al. proposed an approach for multi-modal fusion of data from sensors to provide ambient intelligence for robots [2]. While successful for the specific domains considered, these approaches may not generalize to other domains.

One application of multi-modal data fusion that has gained increasing interest is visual question answering. Zhu et al. [21] tackle the visual question answering problem by building an external knowledge base via iterative querying of the external sources. Their system uses a neural approach where task-driven memories are actively obtained by iterative queries and produces the final answer based on these evidences. Although they take a query based approach for the QA task, their data source is just limited to images. Our approach aims to build a knowledge base integrating visual, textual, and structured data along with the relations among them.

Likewise, Wu et al. propose a method combining an internal representation of image content with information from an external knowledge base to answer complex image queries [20]. Video analytics represents a class of problems related to one of the dimensions of multi-modal systems exploration, namely efficient and fast video querying. In [10], the authors develop a declarative language for fast video analytics and enhance it with the engine that accepts, automatically optimizes and executes the queries in this language efficiently.

While many multi-modal knowledge bases are constructed using learning-based data fusion approaches on large static datasets, query-driven approaches construct knowledge bases through repeated querying of text and multimedia databases. Nguyen et al. [15] propose QKBfly, an approach for on-the-fly construction of knowledge bases from text data driven by queries. QKBfly utilizes a semantic-graph representation of sentences through which named-entity disambiguation, co-reference resolution and relation extraction are performed. Bienvenu et al. propose an approach for utilizing user queries and the associated user feedback to repair inconsistent DL-Lite knowledge bases [3]. The constructed knowledge bases will in most cases include inconsistencies and missing information. Probabilistic knowledge bases have been introduced to handle these inconsistencies by assigning belief scores to facts in the knowledge bases [7], [5], followed by approaches to fuse data from multiple probabilistic bases [19].

Traditional knowledge bases are used for information extraction to answer user queries as they are submitted. On the other hand, dynamic detection of events on streaming data is important for many systems today, due to the need to make users aware of important events in real time. This has resulted in the development of complex event processing systems for purposes such as crisis management [9], to create triggers when streaming data matches pre-defined patterns [6]. Although these systems provide real-time event notification to interested parties, their rule base in most cases is fixed, not supporting evolving mission requirements and users with different interests.

5 Conclusions and Future Work

In this paper we proposed SKOD, a situational knowledge on demand engine that aims to provide a generic framework for dynamically building knowledge bases from multi-modal data to enable effective information extraction and targeted information dissemination for missions that might have evolving requirements. In order to provide the best run-time performance and accuracy, SKOD uses a query-driven approach to knowledge base construction. Being query-driven, it is expected to enable effective information retrieval and dissemination in a variety of fields including law enforcement, homeland defense, healthcare etc., all building knowledge upon the specific interests of the system users.

The development of SKOD is in progress with components for stream data processing, feature extraction from video and text data currently in place. Our future work will involve the development of components for query processing, user similarity modeling, and user relevance feedback to achieve highly accurate real-time targeted information propagation. The system will be evaluated with multiple rich multi-modal datasets such as Visual Genome [11], COCO [12], YouTube-8M [1], and collected tweets and video data set of our own for various missions and user types.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. CoRR **abs/1609.08675** (2016), http://arxiv.org/abs/1609.08675
2. Adjali, O., Hina, M.D., Dourlens, S., Ramdane-Cherif, A.: Multimodal fusion, fission and virtual reality simulation for an ambient robotic intelligence. In: ANT/SEIT. Procedia Computer Science, vol. 52, pp. 218–225. Elsevier (2015)
3. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Query-driven repairing of inconsistent dl-lite knowledge bases. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016. pp. 957–964 (2016), http://www.ijcai.org/Abstract/16/140
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (Mar 2003), http://dl.acm.org/citation.cfm?id=944919.944937
5. Chen, Y., Wang, D.Z.: Knowledge expansion over probabilistic knowledge bases. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. pp. 649–660. SIGMOD '14, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2588555.2610516, http://doi.acm.org/10.1145/2588555.2610516
6. Cugola, G., Margara, A.: Processing flows of information: From data stream to complex event processing. ACM Comput. Surv. **44**(3), 15:1–15:62 (2012). https://doi.org/10.1145/2187671.2187677, https://doi.org/10.1145/2187671.2187677
7. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: KDD. pp. 601–610. ACM (2014)

8. Foresti, G.L., Farinosi, M., Vernier, M.: Situational awareness in smart environments: socio-mobile and sensor data fusion for emergency response to disasters. J. Ambient Intelligence and Humanized Computing **6**(2), 239–257 (2015)
9. Itria, M.L., Daidone, A., Ceccarelli, A.: A complex event processing approach for crisis-management systems. CoRR **abs/1404.7551** (2014)
10. Kang, D., Bailis, P., Zaharia, M.: Blazeit: Fast exploratory video queries using neural networks. CoRR **abs/1805.01046** (2018)
11. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L., Shamma, D.A., Bernstein, M.S., Li, F.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. CoRR **abs/1602.07332** (2016), http://arxiv.org/abs/1602.07332
12. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 740–755. Springer International Publishing, Cham (2014)
13. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 55–60. Association for Computational Linguistics, Baltimore, Maryland (Jun 2014). https://doi.org/10.3115/v1/P14-5010, https://www.aclweb.org/anthology/P14-5010
14. Meditskos, G., Vrochidis, S., Kompatsiaris, I.: Description logics and rules for multimodal situational awareness in healthcare. In: MMM (1). Lecture Notes in Computer Science, vol. 10132, pp. 714–725. Springer (2017)
15. Nguyen, D.B., Abujabal, A., Tran, N.K., Theobald, M., Weikum, G.: Query-driven on-the-fly knowledge base construction. Proc. VLDB Endow. **11**(1), 66–79 (Sep 2017). https://doi.org/10.14778/3151113.3151119, https://doi.org/10.14778/3151113.3151119
16. Palacios, S., Santos, V., Barsallo, E., Bhargava, B.: Miostream: a peer-to-peer distributed live media streaming on the edge. Multimedia Tools and Applications (Jan 2019). https://doi.org/10.1007/s11042-018-6940-2, https://doi.org/10.1007/s11042-018-6940-2
17. Poria, S., Cambria, E., Howard, N., Huang, G.B., Hussain, A.: Fusing audio, visual and textual clues for sentiment analysis from multimodal content. Neurocomput. **174**(PA), 50–59 (Jan 2016). https://doi.org/10.1016/j.neucom.2015.01.095, http://dx.doi.org/10.1016/j.neucom.2015.01.095
18. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
19. Rodríguez, M.E., Goldberg, S., Wang, D.Z.: Sigmakb: Multiple probabilistic knowledge base fusion. PVLDB **9**(13), 1577–1580 (2016)
20. Wu, Q., Wang, P., Shen, C., Dick, A.R., van den Hengel, A.: Ask me anything: Free-form visual question answering based on knowledge from external sources. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 4622–4630 (2016). https://doi.org/10.1109/CVPR.2016.500, https://doi.org/10.1109/CVPR.2016.500
21. Zhu, Y., Lim, J.J., Fei-Fei, L.: Knowledge acquisition for visual question answering via iterative querying. In: CVPR. pp. 6146–6155. IEEE Computer Society (2017)