POLARIS PROJECT

IDALab, Gachon University

PART 0 CONTENTS

POLARIS Project

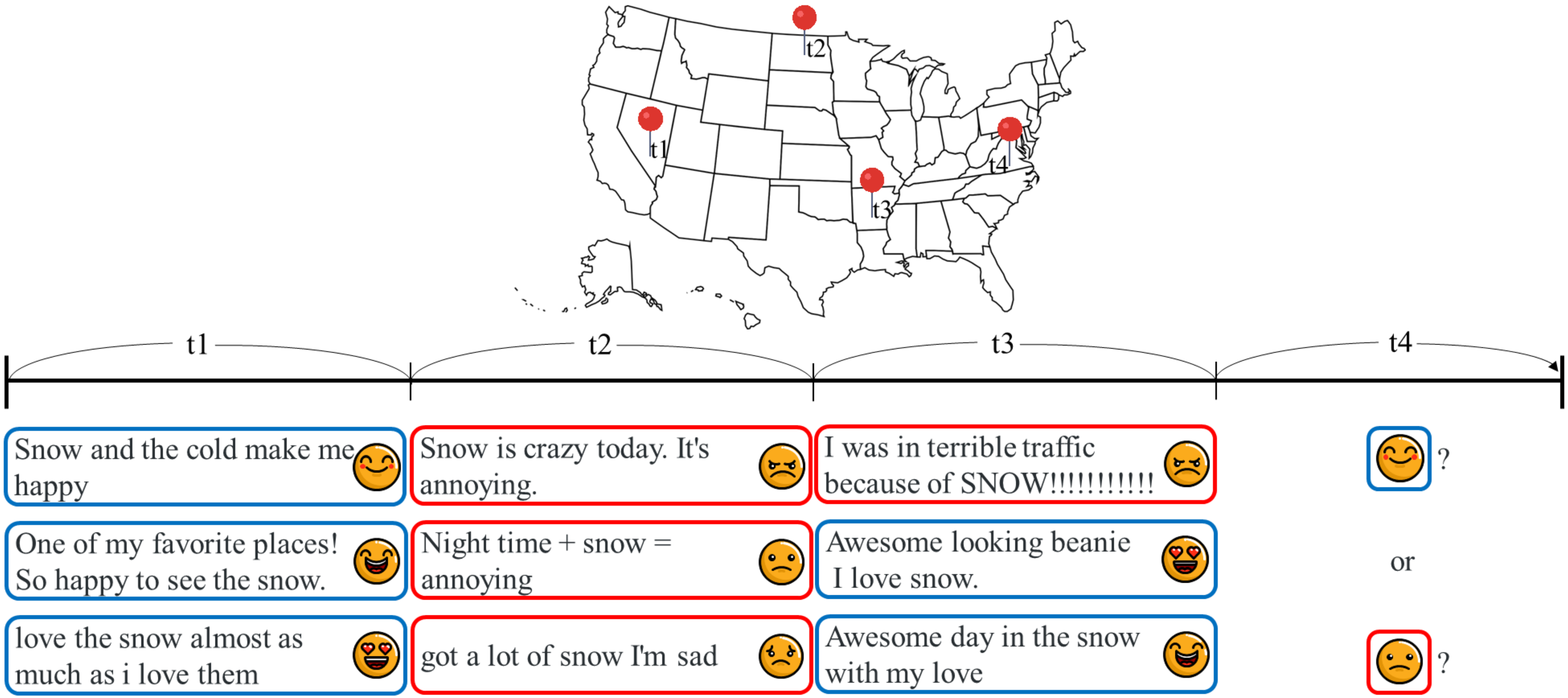
Social Media Contents based Sentiment Analysis and Prediction System (Polaris)

SoYeop Yoo, JeIn Song, and OkRan Jeong

Expert Systems With Applications 105 (2018) pp.102 – 111
https://authors.elsevier.com/c/1Wxau3PiGT7gAi

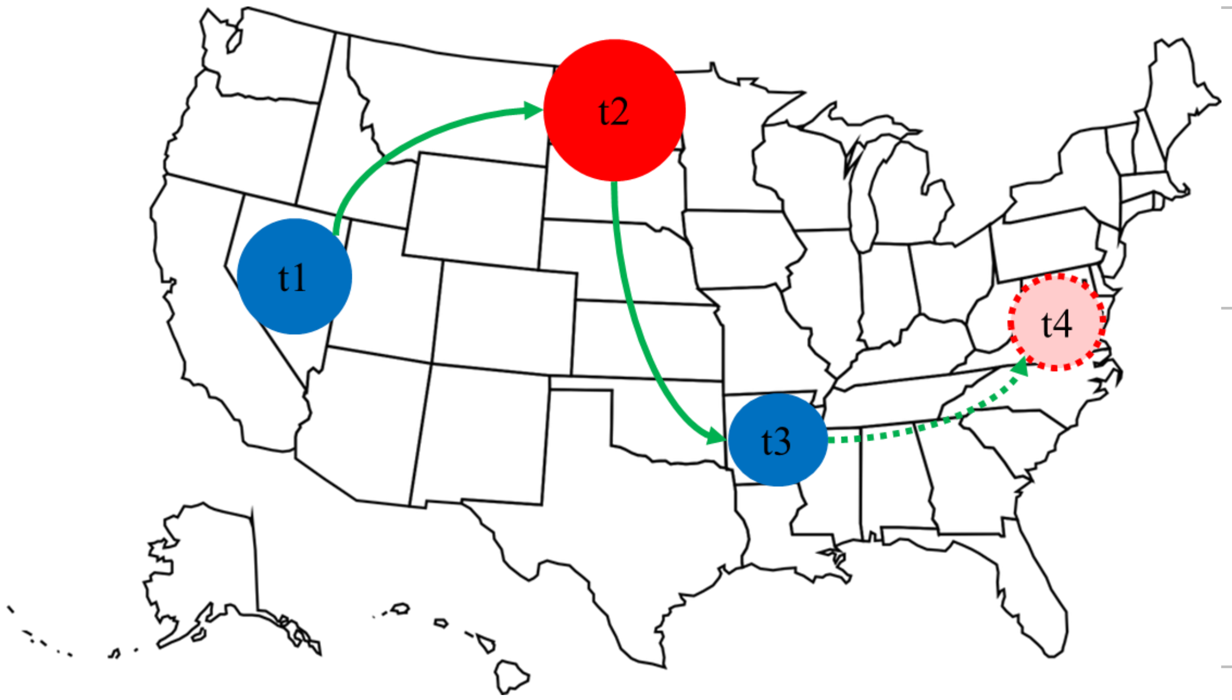
PART 1 *INTRODUCTION*

1 Motivation



1 Motivation

> Polaris: Social media contents based sentiment analysis and prediction system



Efficiency on cost
Use AsterixDB to handle social media contents efficiently

Sentimental path
Enable a user can obtain insight at a glance by analyzing trajectory and sentiment together

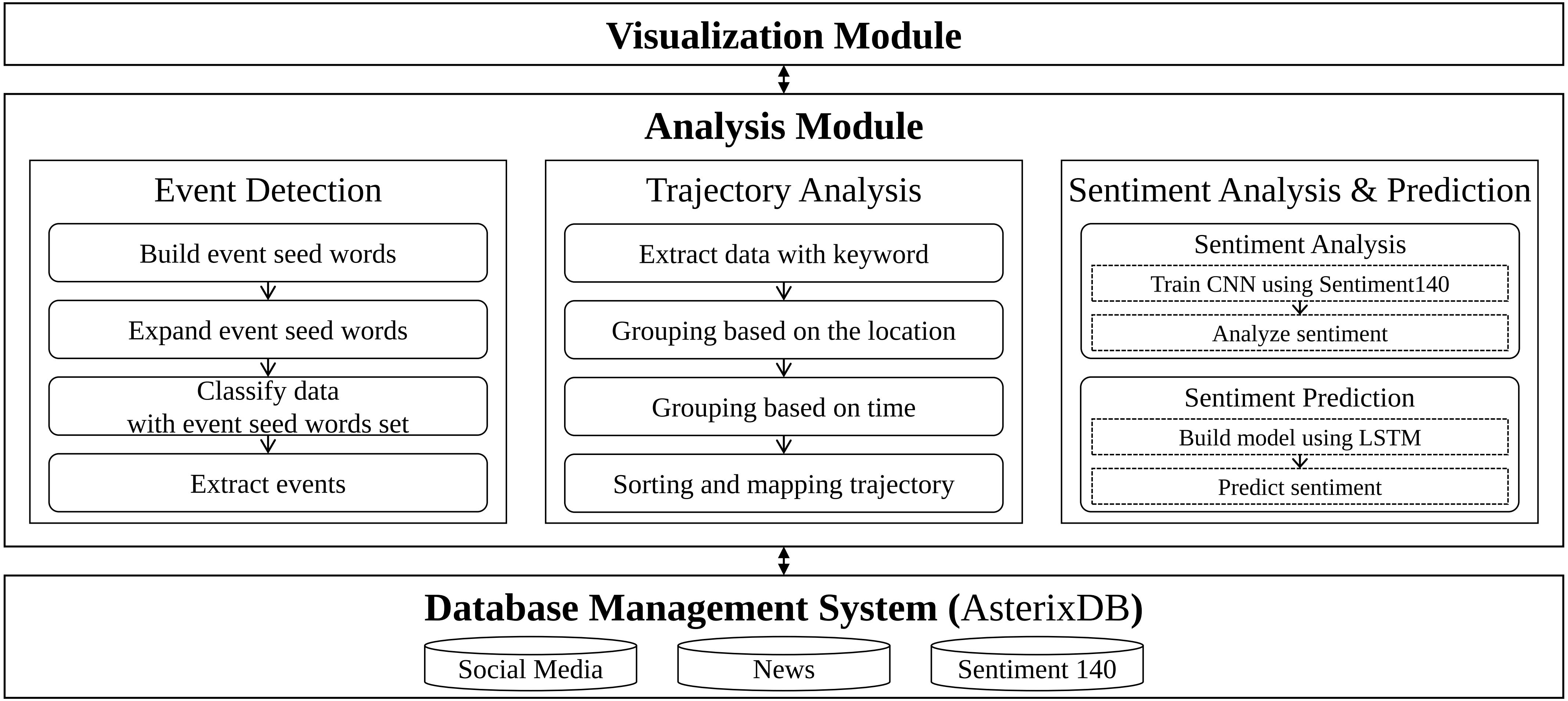
Deep learning on sentimental path
Use recent deep learning techniques (CNN and LSTM)

1 Polaris

- Polaris is the name of our proposed system
- It is a novel system for analyzing and predicting users' sentimental trajectories for events and showing the results

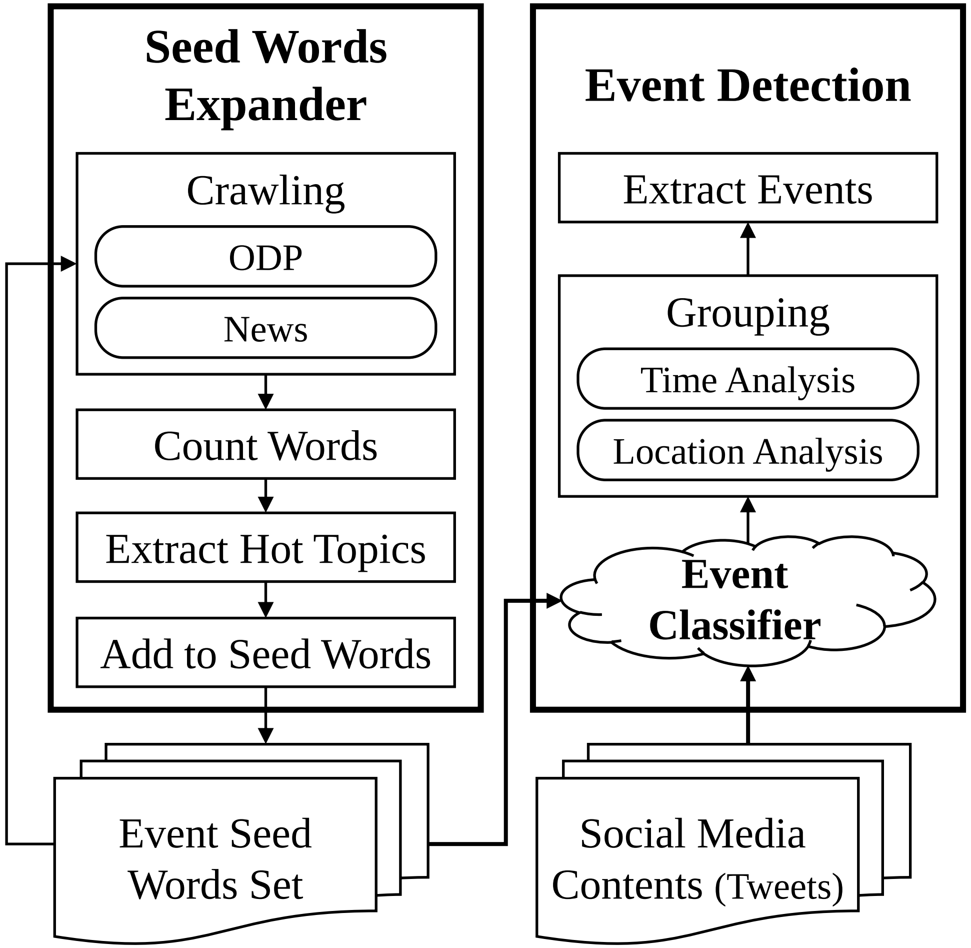
1 Polaris

> Overall Structure



2 Event Detection and Trajectory Analysis

- Event detection
  - Use social media contents as a sensor of event detection
  - Seed words are used for event detection; { crime, disaster, accident }

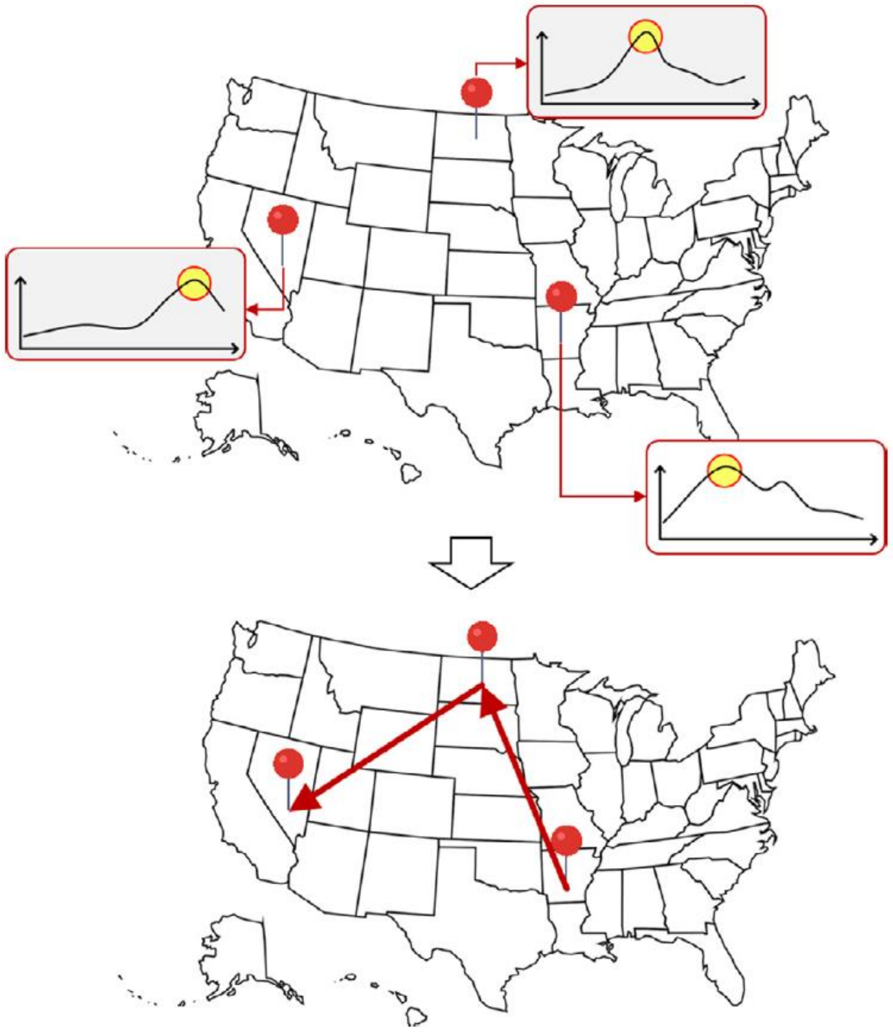


2 Event Detection and Trajectory Analysis

- Detect event based on 'Crime, Disaster, Accident' events
  - Seed words: { crime, disaster, accident }
  - "TEDAS: a Twitter based Event Detection and Analysis System" (Li, Rui, et al. 2012)
- Make event-word-set using word2vec model
- Then, weight on some features
  - Has number? (ex. 5 people died)
  - Has time? (ex. 14:30 am)
  - Has @ or #?

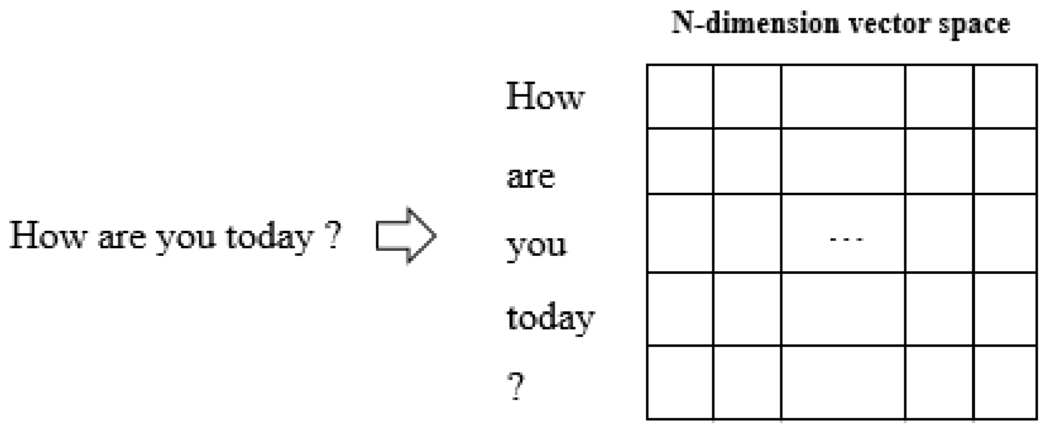
2 Event Detection and Trajectory Analysis

- Trajectory analysis
  - Event trajectories are the results of analysis of the paths which events are propagated through the analysis of the time and area where certain events are occurred



2 Sentiment Analysis

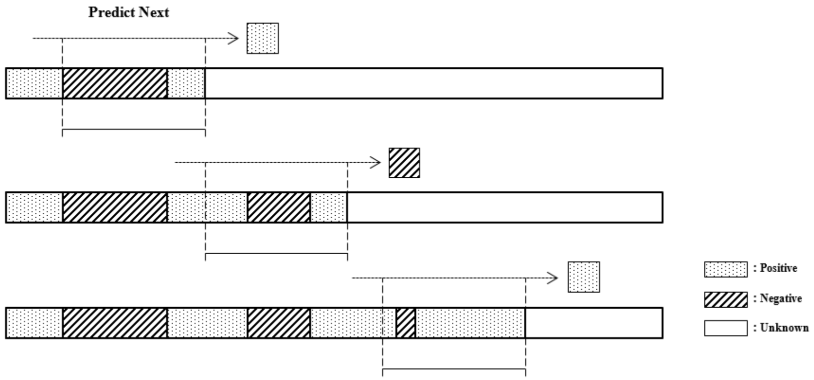
- Sentiment analysis
  - We use machine learning to analyze the sentiment of users for the events occurred
  - The sentiment classification model is trained using the CNN (Convolutional Neural Networks) for sentence classification (Kim, 2014)
- Training CNN
  - Training set: 700,000 positive, 700,000 negative
  - Data source: Sentiment140



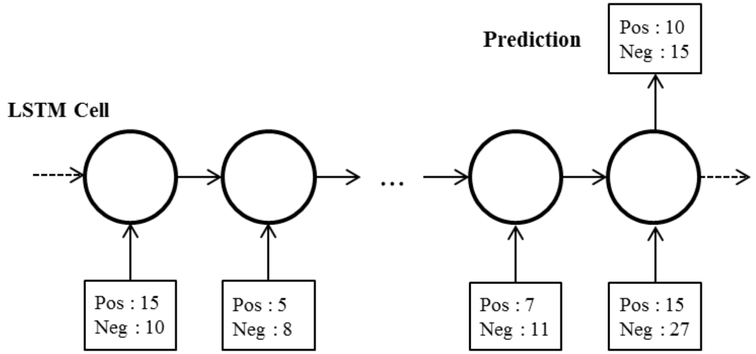
3 Sentiment Prediction

> Sentiment prediction

- The concept of time window is used for sentimental path prediction
- We set the window size to 7 days



- We predict users' sentiment based on LSTM (Long Short-Term Memory)



1 Implementation Environment and Data Set

> Environment
- Ubuntu 14.04
- Intel Xeon CPU E5-2620 v3 (X 2)
- 500GB SSD
- 32G memory
- GTX970 GPU

> Data set
- US tweet data
  - → 40 million tweets
  - → 04/01/2016 ~ 04/30/2016
- Sentiment 140
  - → 800,000 positive data
  - → 800,000 negative data

2 Implementation

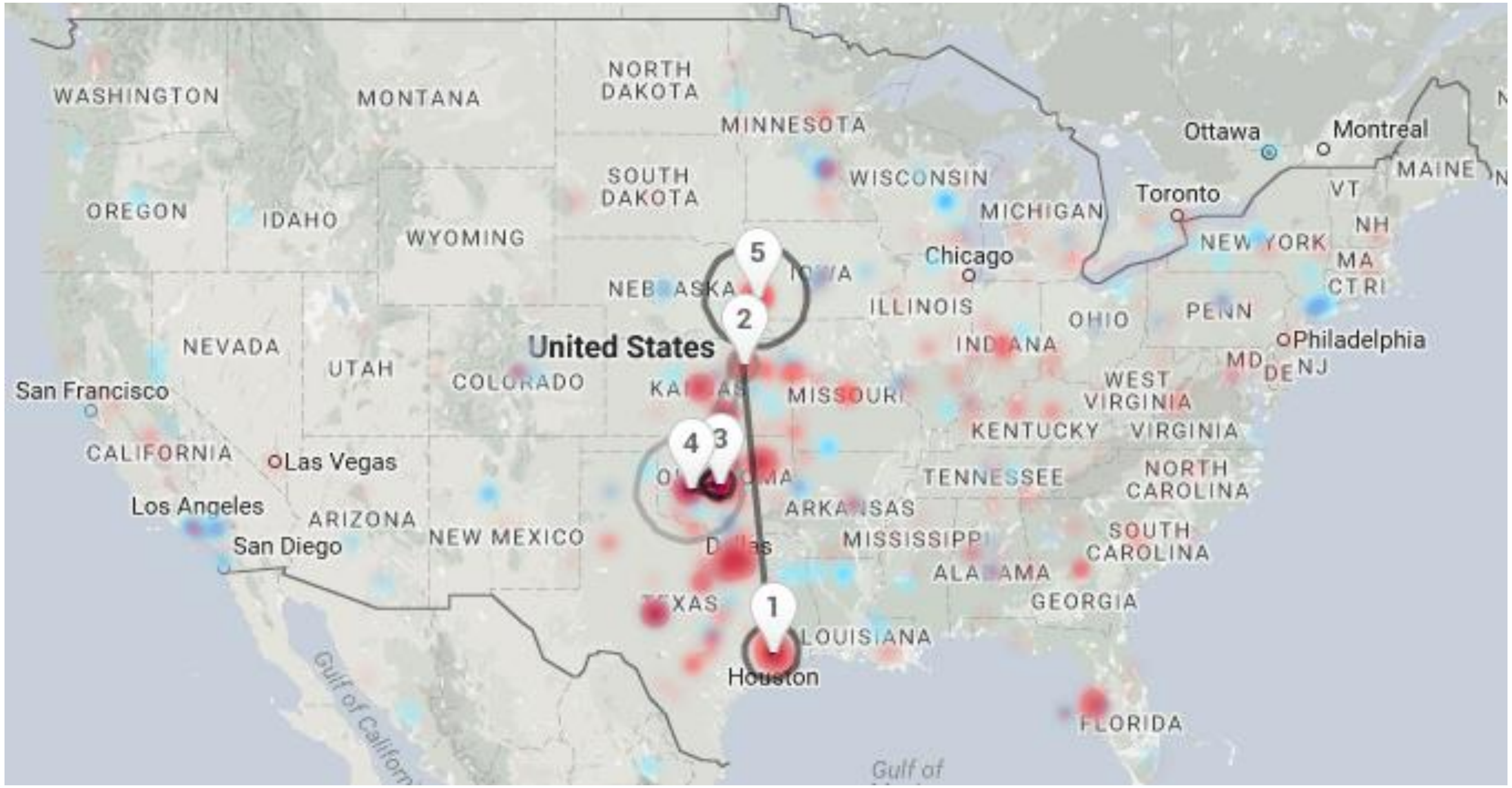
> Event detection result

2 Implementation

> Sentimental path analysis result for 'snow'

2 Implementation

> Sentimental path analysis result for 'tornado'



3 Experiments

> Experiment on sentiment analysis

— Accuracy according to vocabulary size

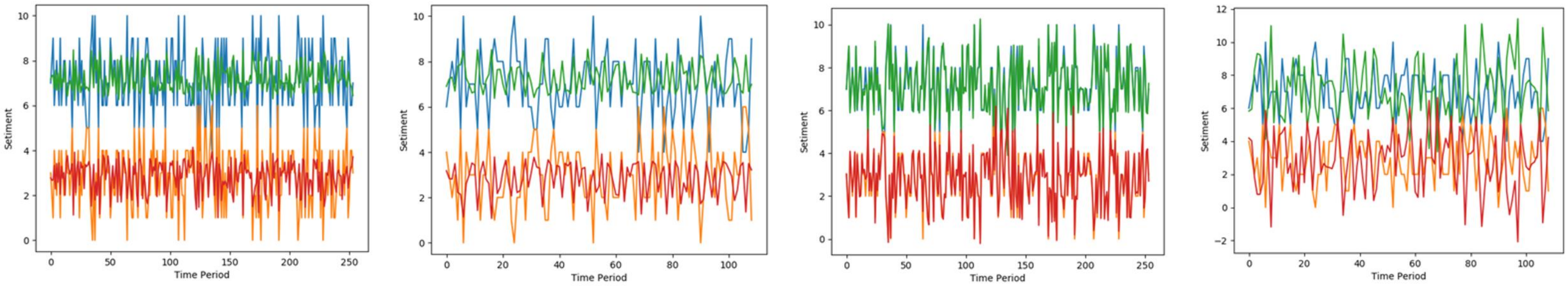
Vocabulary Size	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
20000	0.8050	0.8318	0.7611	0.7802	0.7489
30000	0.7818	0.8311	0.7921	0.7695	0.7667
40000	0.8173	0.8456	0.8012	0.7937	0.7875
50000	0.8095	0.8329	0.7808	0.7746	0.7699
60000	0.7886	0.8329	0.7749	0.7842	0.7743

— Comparison result of sentiment analysis model with other machine learning methods

Model	Naïve-Bayes	SVM	Random Forest	Proposed Model
Precision	0.76	0.77	0.76	0.839
Recall	0.76	0.77	0.76	0.845
F-1 Score	0.75	0.77	0.76	0.841

3 Experiments

- Experiment on sentiment prediction
  - Left side: training data
  - Right side: test data
  - Blue line: answer of positive
  - Green line: prediction of positive
  - Orange line: answer of negative
  - Red line: prediction of negative



(a) At iteration 1500

(b) At iteration 4500

1 Conclusion

> We propose a system to analyze social media contents in real time
> We use AsterixDB to efficiently manage the social media contents
> We propose and implement Polaris that finds events in real time to analyze and predict users' sentimental paths

2 Usage

> Polaris can be utilized for disaster notice service for earthquakes and tsunamis or real time traffic accident informing service
> It also can be applied to social events too, it can be used for diverse marketing programs

PolarisX

SoYeop Yoo, and OkRan Jeong

"Automating the Expansion of a Knowledge Graph,"
Expert Systems with Applications, Vol. 141, 2020.

1 Introduction

- Object of research
  - Expand the knowledge graph automatically not only to rapidly expand the knowledge base for any human language, but also to support neologisms
- Limitations of the existing knowledge graphs
  - The existing knowledge base is rather limited in size and scope for most of the human languages
  - It is not able to support in timely manner neologisms that form a part of the human common sense
  - Example neologisms

Word	Past	Now
Apple	a kind of fruit	IT brand
Ford	a location where a stream is shallow	motor company
Gangnam style	lifestyle associated with the Gangnam distr ict of Seoul	K-pop by PSY
Trump	playing card	the 45th president of the United States
Google	company name	search for information on the Web

1 Introduction

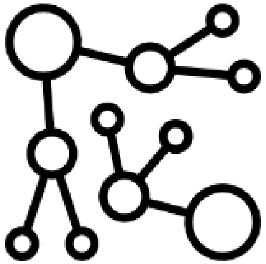
- Our approaches
  - Three major components
    - → The social crawler to expand data resource
    - → The semantic analyzer to determine new relationships using the fine-tuned BERT
    - → The knowledge miner to build and extend the knowledge graph

2 Motivation

Input: selfie

Existing Knowledge Graph-based System

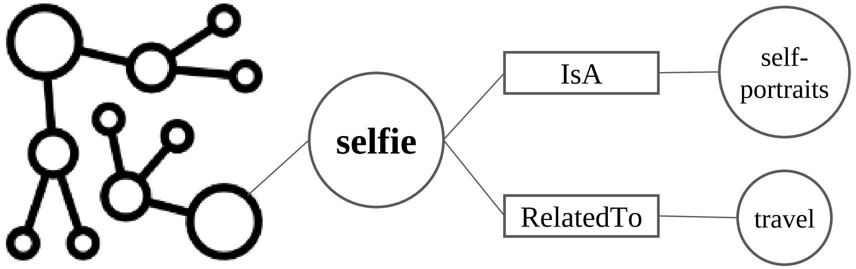
Finding 'selfie' in
the existing knowledge graph



…?

PolarisX-based System:
Auto-growing Knowledge Graph-based System

Finding 'selfie' in
PolarisX(An Auto-growing Knowledge Graph)

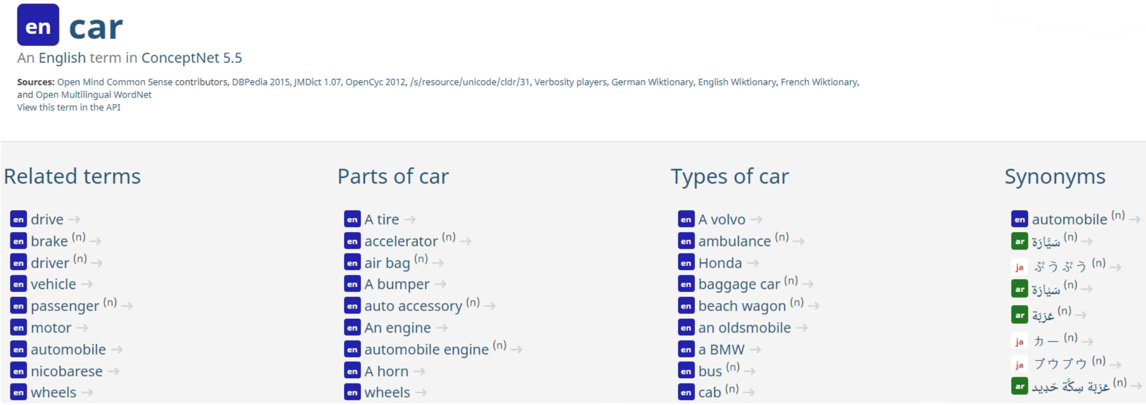


{ selfie, *IsA*, self-portraits }
{ selfie, *RelatedTo*, travel }

1 Knowledge Graph: ConceptNet

> ConceptNet

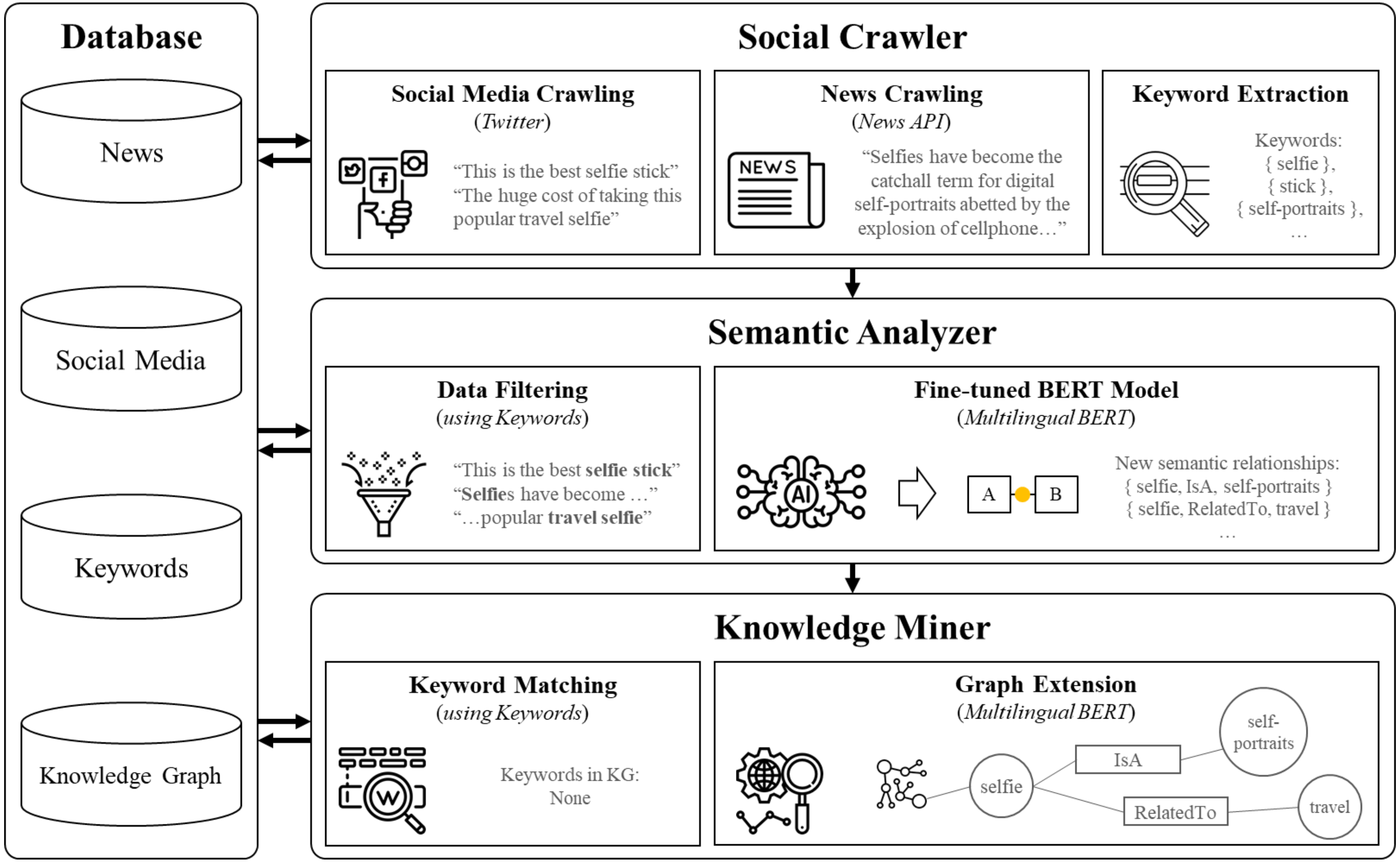
— A freely-available semantic network, designed to help computers understand the meanings of words that people use



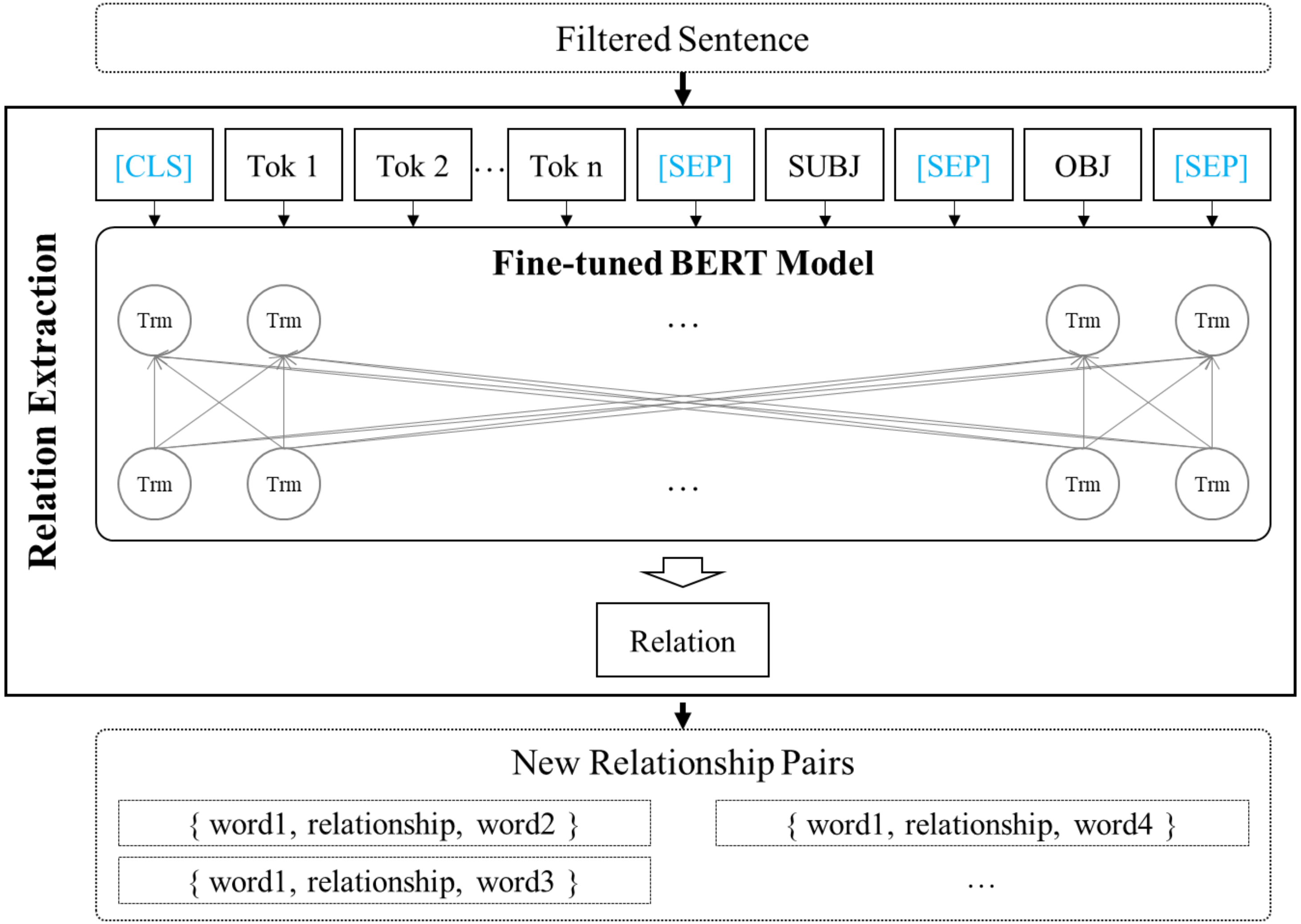
> Problem of ConceptNet

Core Languages		Common Languages	
Language	Num. of Words	Language	Num. of Words
English	1,803,873	Czech	129,183
French	3,023,144	Filipino	17,620
Italian	1,078,629	Korean	47,268
Japanese	363,663	Slovak	29,768
Chinese	242,746	Turkish	65,892

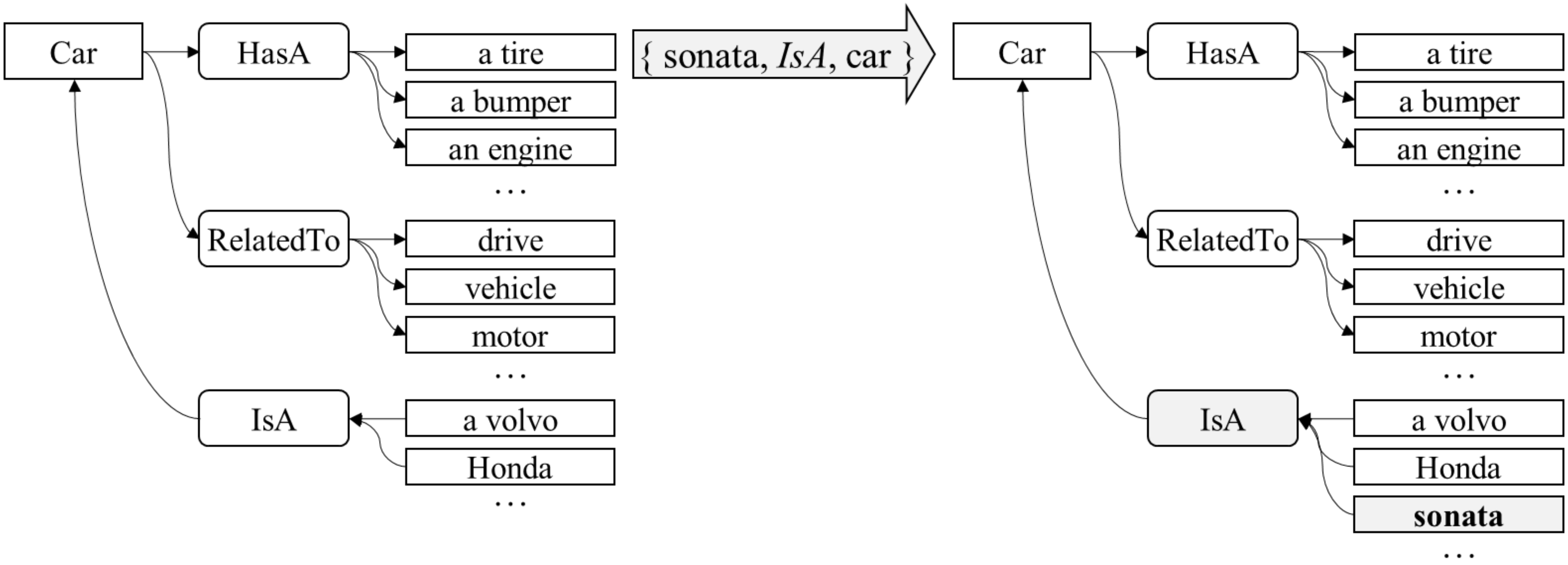
2 Architecture



3 Semantic Analyzer using the fine-tuned BERT model



4 Expanding the ConceptNet Knowledge Graph using PolarisX



1 Experiment Result

- Experiment on extension of knowledge graph
  - Using data
    - Tweets: 2.5 million (2018/08/08 ~ 2018/08/14)
    - News: 35,000
  - Comparison result of with the existing knowledge graphs

Knowledge Graphs	# of Relations	# of Edges
DBpedia (English)	2,813	176,043,129
YAGO	77	25,946,870
NELL	425	432,845
OpenCyc	18,526	2,413,894
Probase	1	20,757,545
PolarisX (+ ConceptNet)	40	32,871,573

1 Experiment Result

> Experiment on accuracy of semantic analyzer

- TACRED data
- We use 'BERT-Base, Multilingual Cased' model
- Environment: Google colab TPU

- Comparison result on TACRED dataset

Models	Precision	Recall	F1 score
Logistic Regression (Y. Zhang, Qi, and Manning 2018)	73.5	49.9	59.4
PA-LSTM (Y. Zhang et al. 2017)	65.7	64.5	65.1
C-GCN+PA-LSTM (Y. Zhang, Qi, and Manning 2018)	71.3	65.4	68.2
BERT-based model (our model)	79.1	72.6	75.7

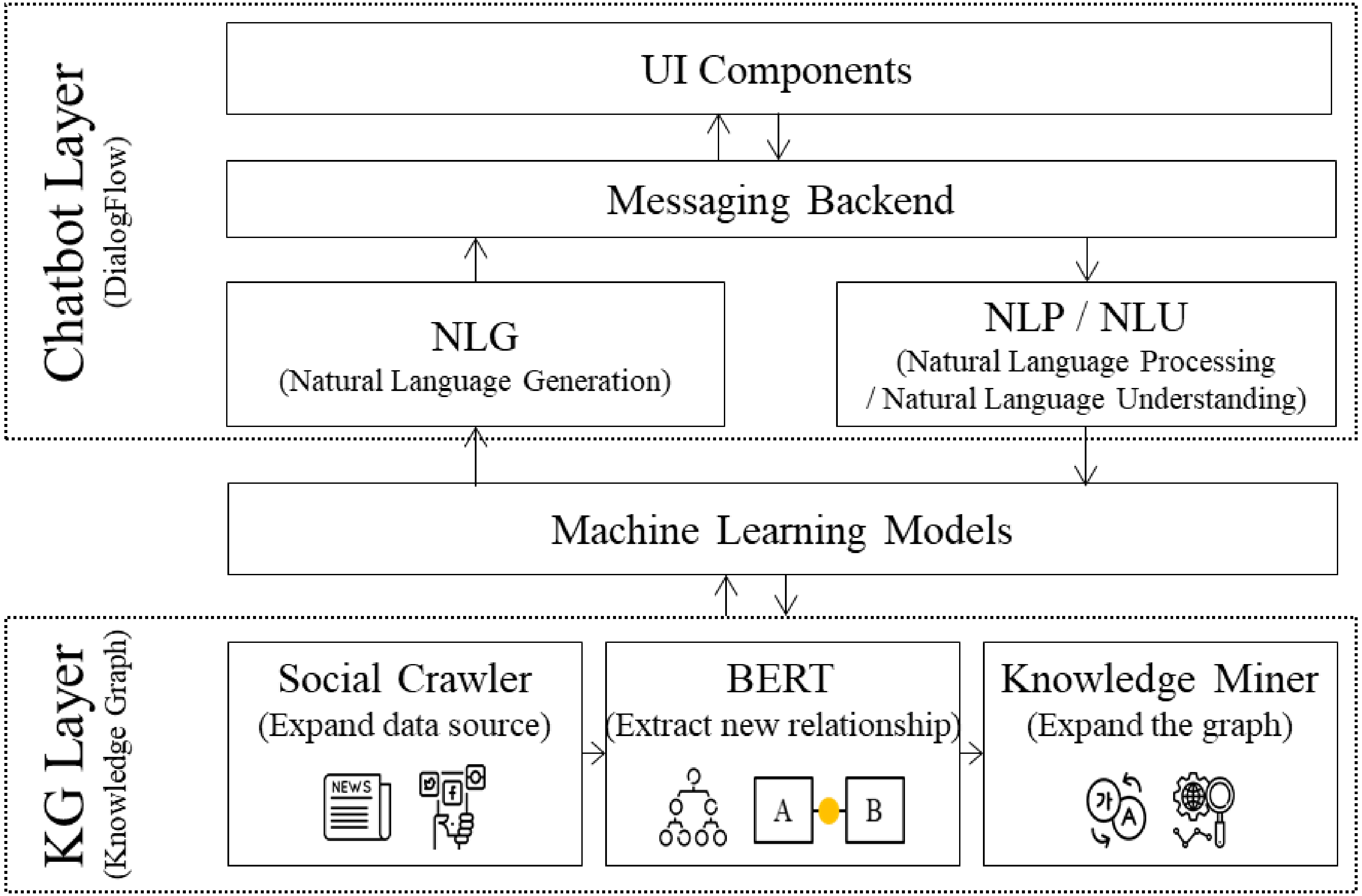
PolarisX-Bot

SoYeop Yoo, and OkRan Jeong

1 Introduction

- PolarisX-bot
  - An automatic extended knowledge graph-based intelligent chatbot
- Major layers
  - Knowledge graph layer
    - → Auto-growing knowledge graph using BERT (PolarisX)
  - Chatbot layer
    - → Google DialogFlow

1 Overall Structure



1 Dataset

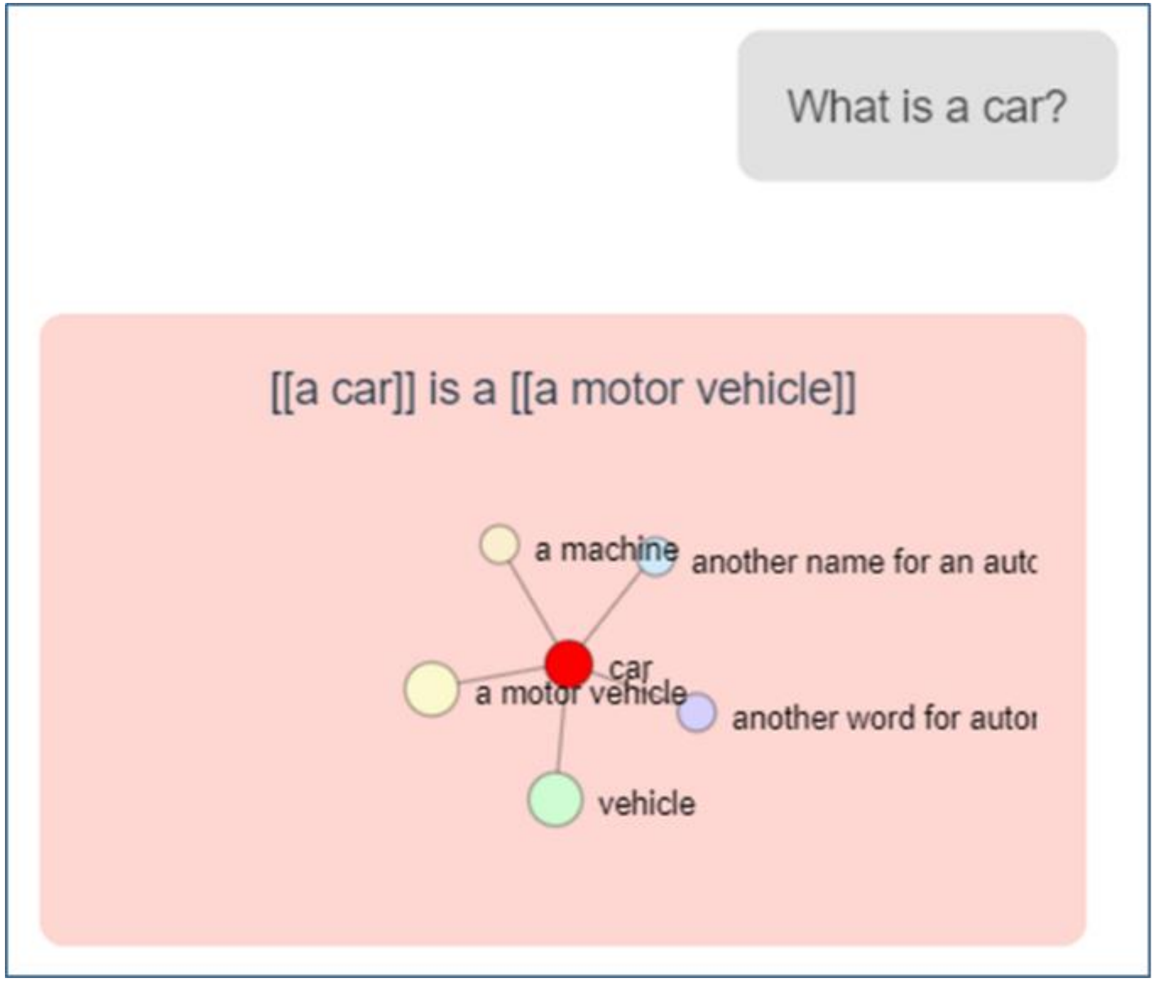
Data	Size	Collection method
Twitter	About 15 million tweets	FeedAdapter(AsterixDB)
News	About 102,000 articles	News API
TACRED	About 106,000 sentences	Linguistic Data Consortium

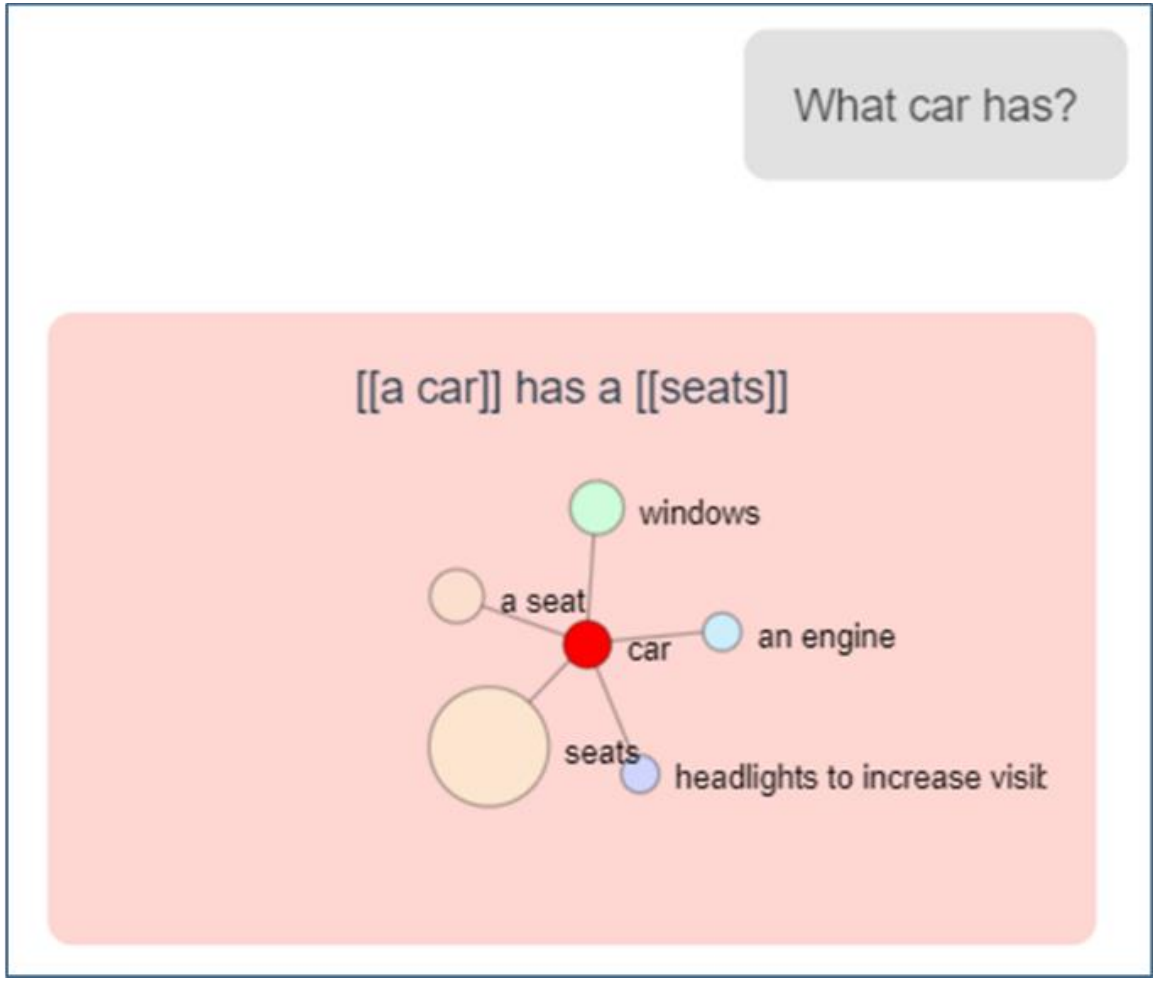
2 Experiment

> Result on BERT-based relation extraction model

BERT model	Dataset	Evaluation set	Accuracy	Loss
bert_cased_L-12_H-768_A-12	TACRED	dev set	0.7528	1.2011
		test set	0.7885	1.1412

3 Implementation

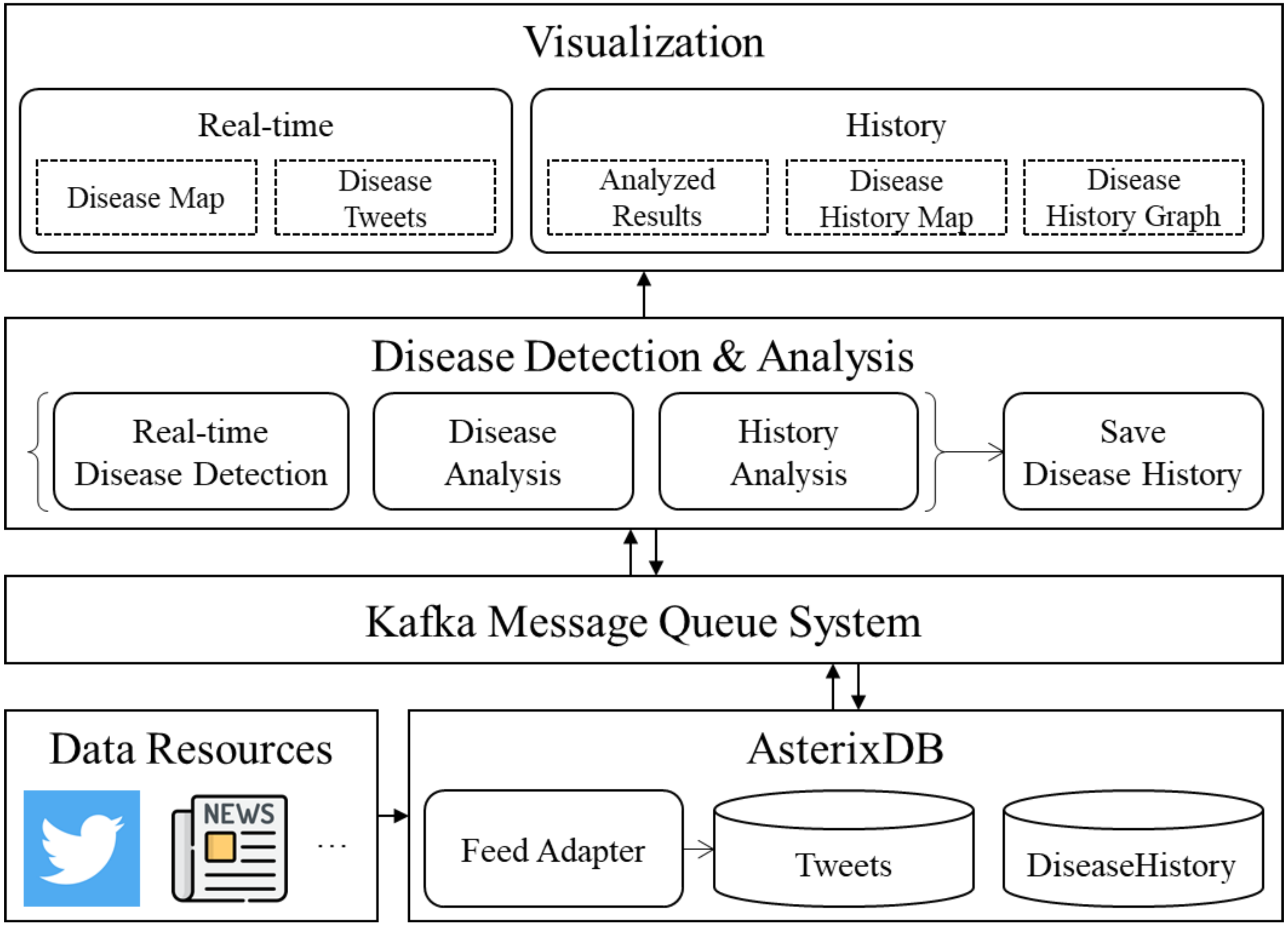




POLARIS Project for Disease Detection

SoYeop Yoo, DaeHo Kim, SungMin Yang and OkRan Jeong

1 Overall Structure



Polaris Project for Disease Detection

1 Real-time Disease Detection

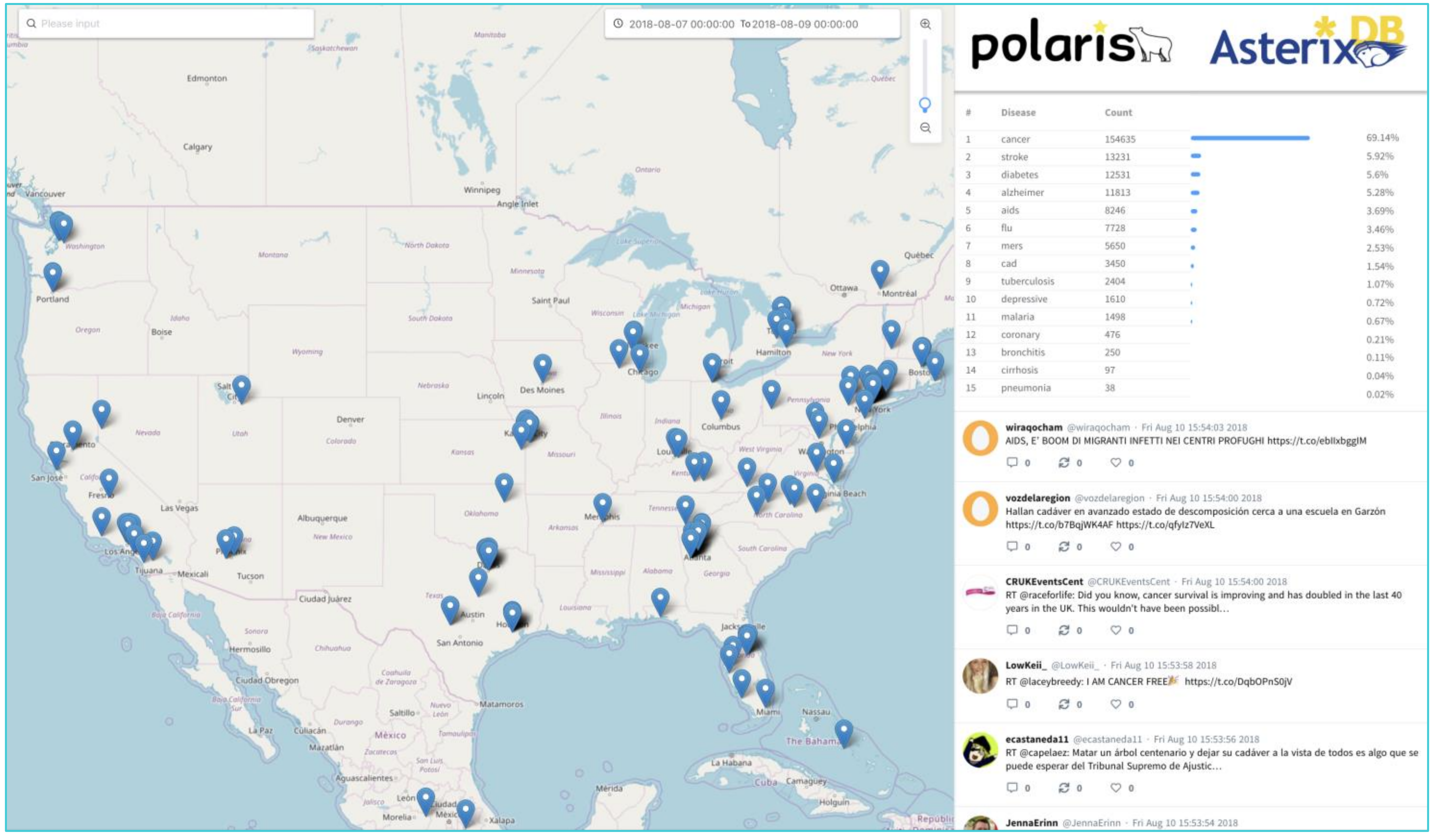
- 15 diseases
  - coronary artery disease (CAD), stroke, flu, pneumonia, bronchitis, diabetes mellitus, Alzheimer's, tuberculosis, cirrhosis, cancer, AIDS, malaria, depressive disorder, measles, MERS
  - (ref) ICD-11 from WHO, https://www.healthline.com/health/top-10-deadliest-diseases#tb, https://list25.com/25-deadliest-diseases-in-human-history/
- Real-time detection
  - Apache Kafka
  - Apache AsterixDB
    - → Feed Adapter to crawl Twitter streaming data
    - → UDF (User Defined Function) to detect diseases

PART 2 ***POLARIS FOR DISEASE DETECTION***

Polaris Project for Disease Detection

2 Disease History

> History
- Save the detected tweets into history
- Get results of the specific time

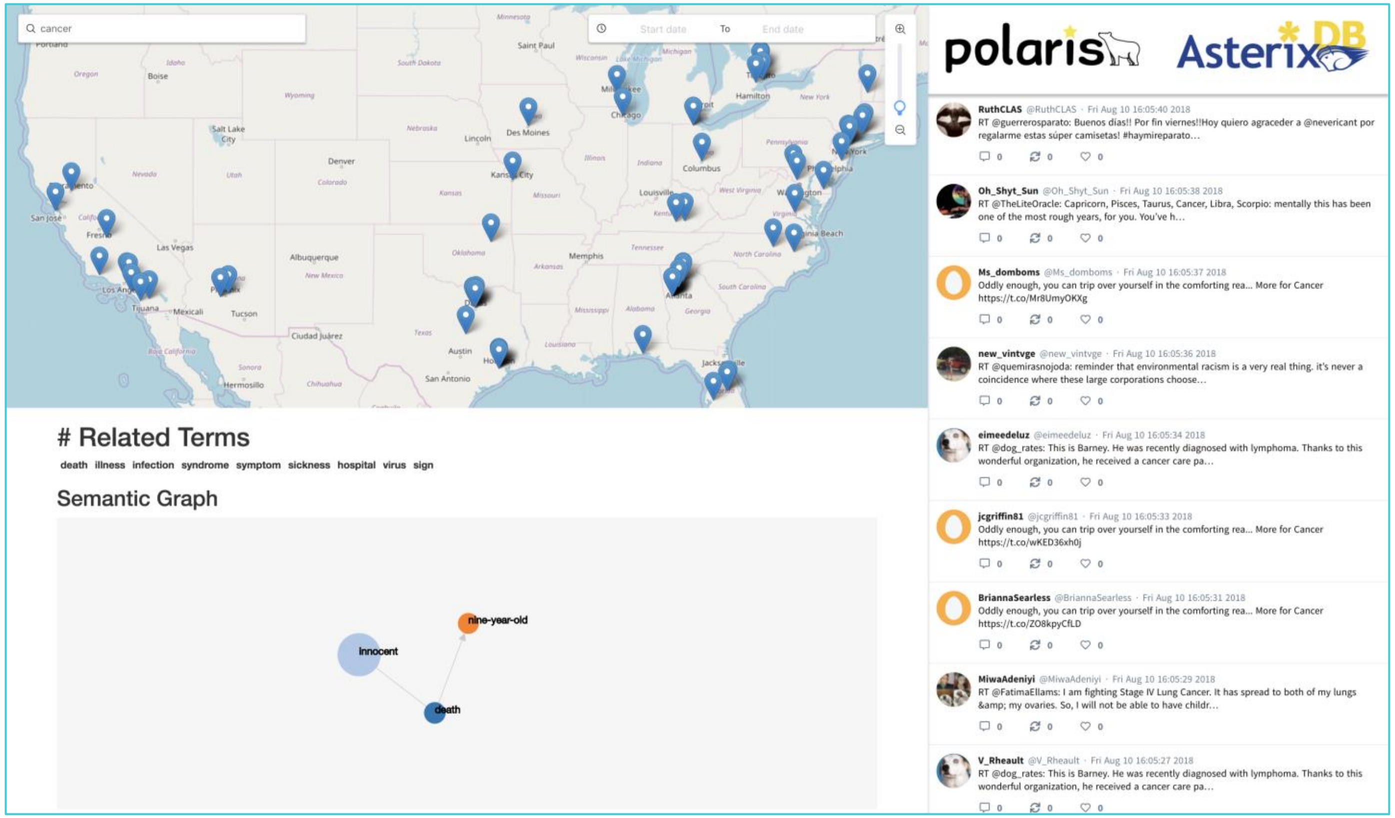


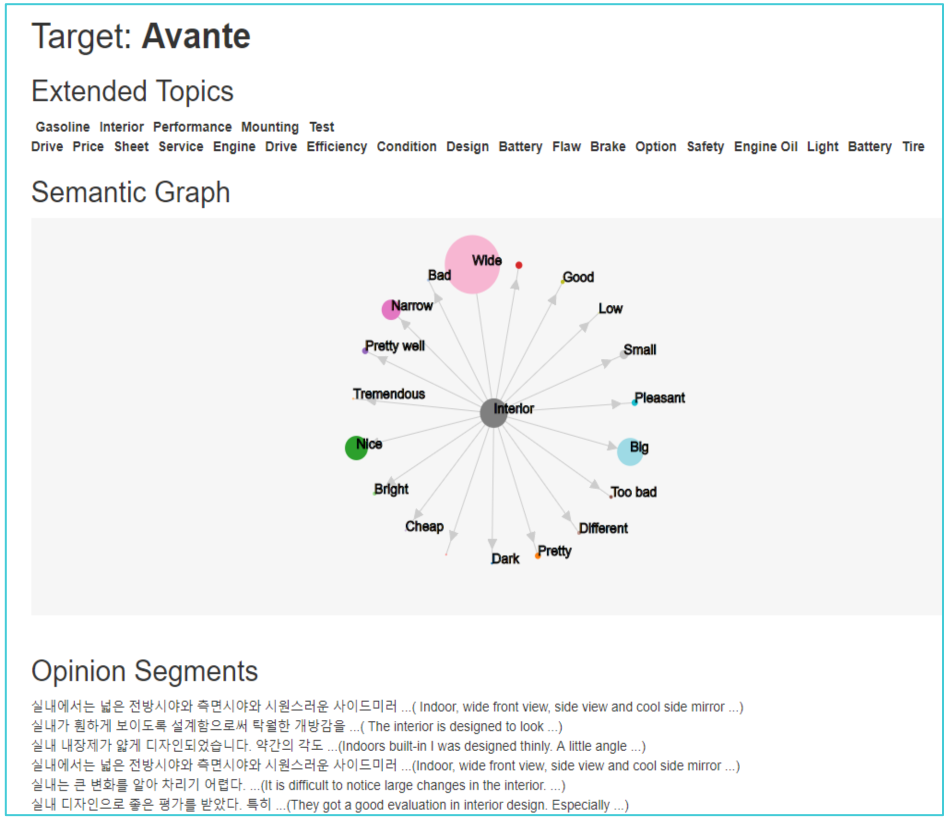
PART 2 POLARIS FOR DISEASE DETECTION

Polaris Project for Disease Detection

3 Disease Analysis

> Semantic analysis
  — Analysis opinions using semantic (PolarisX)





Future Work for POLARIS Project

SoYeop Yoo, DaeHo Kim, SungMin Yang, and OkRan Jeong

PART 1 **FUTURE WORK**

Future Work for POLARIS Project

1 Polaris & PolarisX

> PolarisX as an open source knowledge graph
> Improve opinion mining using machine learning and PolarisX

2 Conversational AI Framework

> Based on PolarisX