Using Twitter Streams for Opinion Mining: A case study on Airport Noise

Iheb Meddeb, Catherine Lavandier, Dimitris Kotzinos

Lab ETIS UMR 8051

Paris Seine University – University of Cergy-Pontoise - ENSEA - CNRS

Ecole Nationale Supérieure de l'Electronique et de ses Applications

Introduction and Problematic

Traditional Surveys methods : Phone calls, interviews, web surveys ➔ Hard work + Time consuming

Introduction and Problematic

Traditional Surveys methods : Phone calls, interviews, web surveys ➔ Hard work + Time consuming

- Share news
- Participate in discussions
- Express emotions and opinions

Social media and microblogging expansion ➔ new source of information ➔ Big Data

❑ Is it possible to offer an alternate survey method using this Data?

Problems emerging:

- How to deal with the big amount of data to extract only the relevant ones to the topic of interest?
- Which methods are used to classify data when the survey is based on opinions?

Example Tweet: "@HeathrowAirport Stop these noise sewers, my kids are not sleeping. scum" ➔ negative

Introduction and Problematic

Traditional Surveys methods : Phone calls, interviews, web surveys ➔ Hard work + Time consuming

- Share news
- Participate in discussions
- Express emotions and opinions

Social media and microblogging expansion ➔ new source of information ➔ Big Data

❑ Is it possible to offer an alternate survey method using this Data?

Problems emerging:

- How to deal with the big amount of data to extract only the relevant ones to the topic of interest?
- Which methods are used to classify data when the survey is based on opinions?

Example Tweet: "@HeathrowAirport Stop these noise sewers, my kids are not sleeping. scum" ➔ negative

Introduction and Problematic

Introduction and Problematic

Goal: understand people's sentiment towards noise (potentially coming from the airport use)

Use case: the Heathrow Airport and the area around it

Problem 1: Relevance of tweets

- capture relevant twitter messages (tweets) about airport (Heathrow airport) noise

  Challenge: Filter out trivial and irrelevant messages from the stream

Introduction and Problematic

Goal: understand people's sentiment towards noise (potentially coming from the airport use)

Use case: the Heathrow Airport and the area around it

Problem 1: Relevance of tweets

- capture relevant twitter messages (tweets) about airport (Heathrow airport) noise

  Challenge: Filter out trivial and irrelevant messages from the stream

Problem 2: Sentiment Analysis

- Extract the sentiment of the user and classify tweets based on opinions (positive, negative or neutral)

  Challenge: deal with Abbreviations, misspelling, incomplete sentences, negation, contrast, punctuations, irony, sarcasm

  Challenge: Sentiment classification close to human judgements

Introduction and Problematic

Goal: understand people's sentiment towards noise (potentially coming from the airport use)

Use case: the Heathrow Airport and the area around it

Problem 1: Relevance of tweets

- capture relevant twitter messages (tweets) about airport (Heathrow airport) noise

    Challenge: Filter out trivial and irrelevant messages from the stream

Problem 2: Sentiment Analysis

- Extract the sentiment of the user and classify tweets based on opinions (positive, negative or neutral)

    Challenge: deal with Abbreviations, misspelling, incomplete sentences, negation, contrast, punctuations, irony, sarcasm

    Challenge: Sentiment classification close to human judgements

Problem 3: Statistical analysis of the sentiments (post processing)

- Use classified tweets to:
    - pinpoint the areas mostly affected by the noise.
    - identify temporal bursts of tweets during the day and the week.
    - Trendy (hot) topics of discussion related to airport noise

State of the art

Problem	Methods	Advantages	Disadvantages
Relevance	Keyword query	simple and fast	Captures also many irrelevant tweets **[8]**
	Feature-based techniques **[12]** / Rule-based techniques **[10]**	Offer more accuracy	Specific to the topic and its characteristics
	Document-pivot techniques (*tf-idf* **[4]**, *named entity*)	Based on similarity of texts	Less suitable for tweets
Sentiment Analysis	Emoticons	Text and topic independent **[6]**	Effective only for tweets with emoticons
	POS features	Separate objective and subjective sentences	Tagger errors due to tweets
	Convolutional neural nets **[14]**	Good to learn subjective expressions (n-grams)	Time consuming
	Lexicon-based	Detect the sentiment of a tweet based the prior polarity of its words	Needs to have big lexicon dataset / Performs better with other features (negation, contrast) **[9]**

Twitter

335 M monthly active users

80% Twitter users on Mobile

500 M Tweets / Day

83% from UN countries have Twitter

Stats from https://www.statista.com and https://www.omnicoreagency.com/
(Accessed on 09/11/2018)

Twitter

335 M monthly active users

80% Twitter users on Mobile

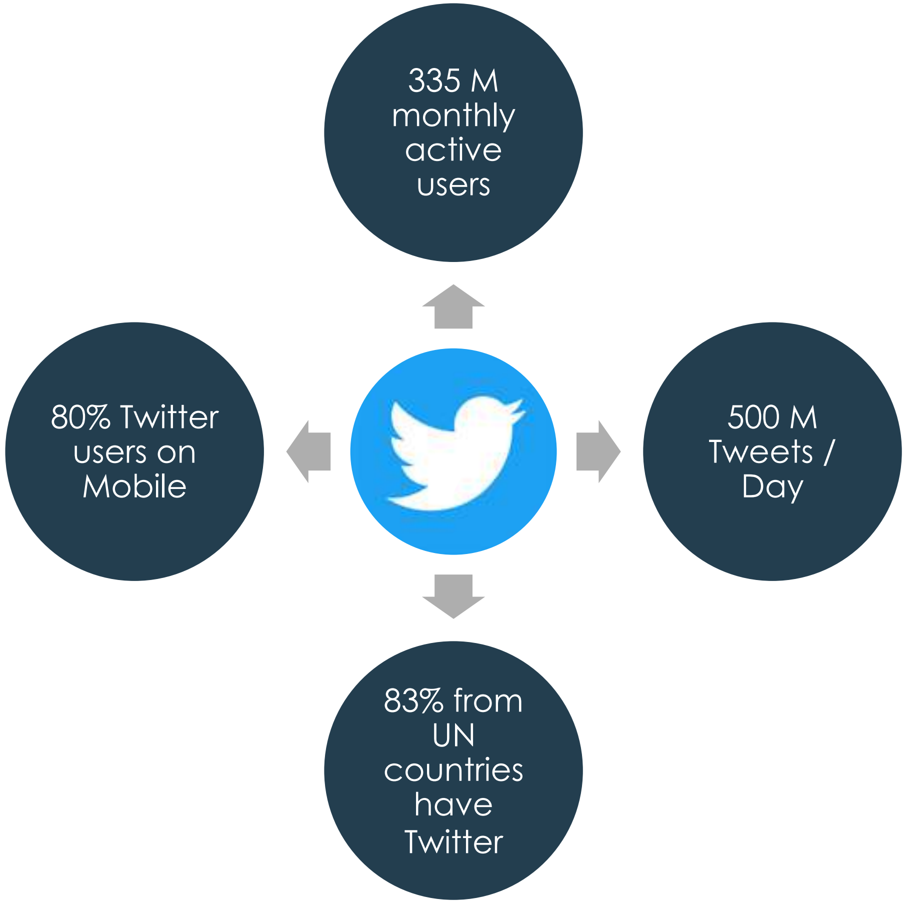
500 M Tweets / Day

83% from UN countries have Twitter

Twitter properties:
- Short messages (280 maximum characters)
- Introducing hashtags (#) and mentioning users (@)
- Share tweets of others (retweet)
- Follow users / user followers

Stats from https://www.statista.com and https://www.omnicoreagency.com/ (Accessed on 09/11/2018)

Twitter

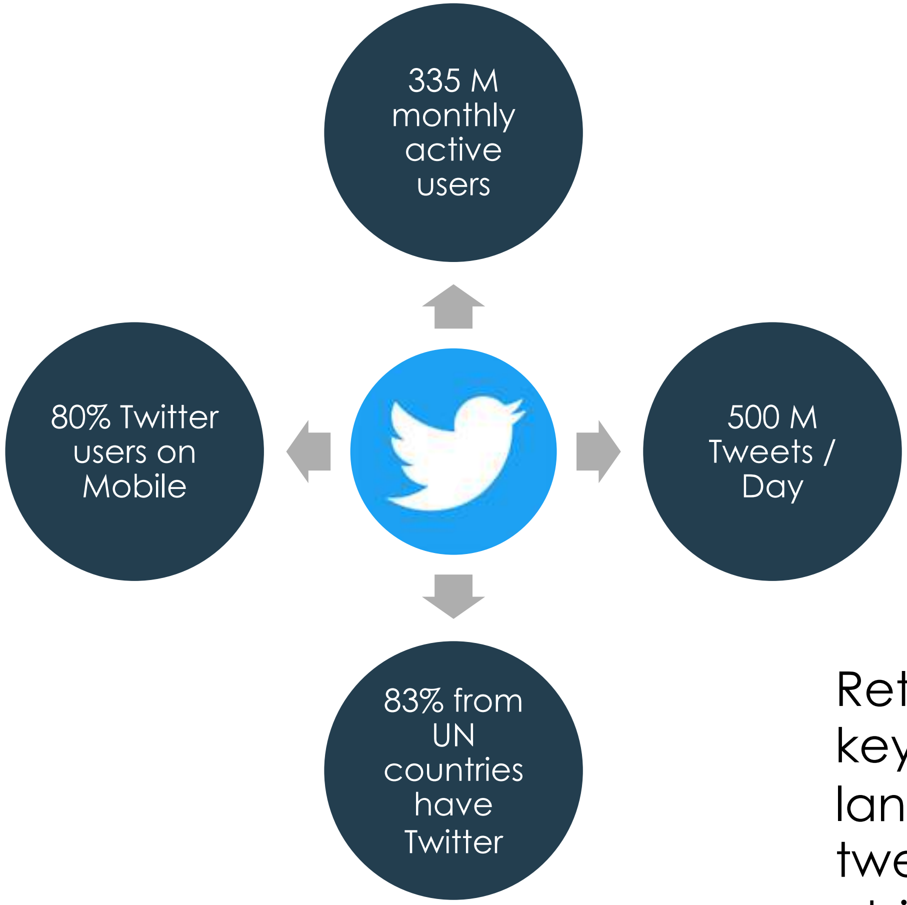


Twitter properties:
- Short messages (280 maximum characters)
- Introducing hashtags (#) and mentioning users (@)
- Share tweets of others (retweet)
- Follow users / user followers

Twitter API:
- Integrate tweets into apps websites
- Advertise on twitter
- Access publicly available tweets (real-time stream or search)

Stats from https://www.statista.com and https://www.omnicoreagency.com/ (Accessed on 09/11/2018)

Twitter



Twitter properties:
- Short messages (280 maximum characters)
- Introducing hashtags (#) and mentioning users (@)
- Share tweets of others (retweet)
- Follow users / user followers

Twitter API:
- Integrate tweets into apps websites
- Advertise on twitter
- Access publicly available tweets (real-time stream or search)

Retrieve tweets by keywords, locations, languages, users, etc. ➔ tweets presented as JSON objects


```
{
"created_at" : "Thu May 10 15:24:15 +0000 2018" ,
 "id_str" : "850006245121695744" ,
 "text" : "Here is the Tweet message." ,
 "user" : {
 } ,
 "place" : {
 } ,
 "entities" : {
 } ,
 "extended_entities" : {
 }
}
```


Stats from https://www.statista.com and https://www.omnicoreagency.com/ (Accessed on 09/11/2018)

Workflow: Main steps

Retrieve tweets with Twitter API
- Location filter
- Keywords filter

Preprocessing tweets
- Combination of natural language processing (NLP) techniques

Relevance classification
- *Tf-idf* (n-grams, hashtags, usernames)
- Supervised machine learning algorithm
- Lexicon-based classifier

Relevant tweets

Relevant tweets Preprocessing
- Other combination of NLP techniques

Sentiment classification
- Emoticons
- Lexicon
- SentiWordNet (SWN)

Positive tweets

Negative tweets

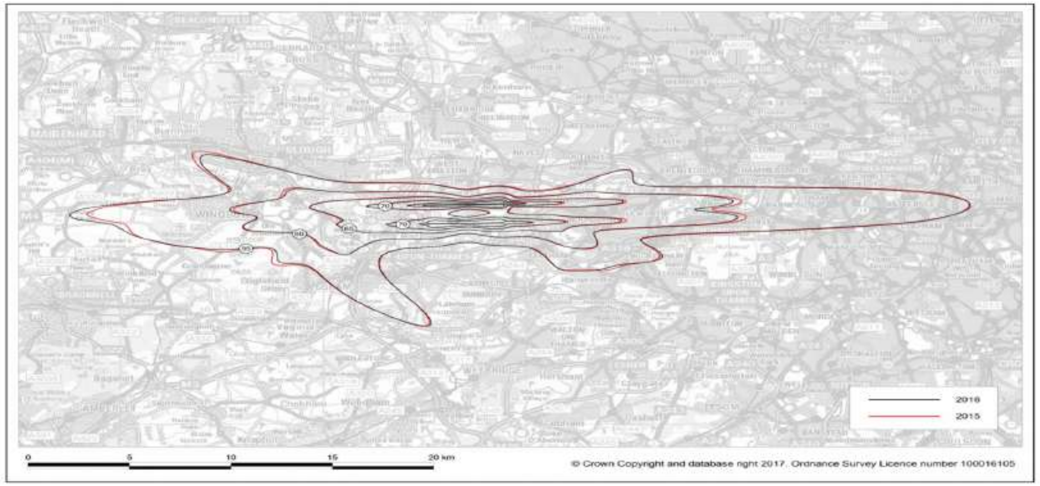
Neutral tweets

Statistical Study
- Temporal
- Geographical
- Retweets, hashtags and mentioned users

Workflow: Gathering tweets

❑ Retrieve tweets with location filter

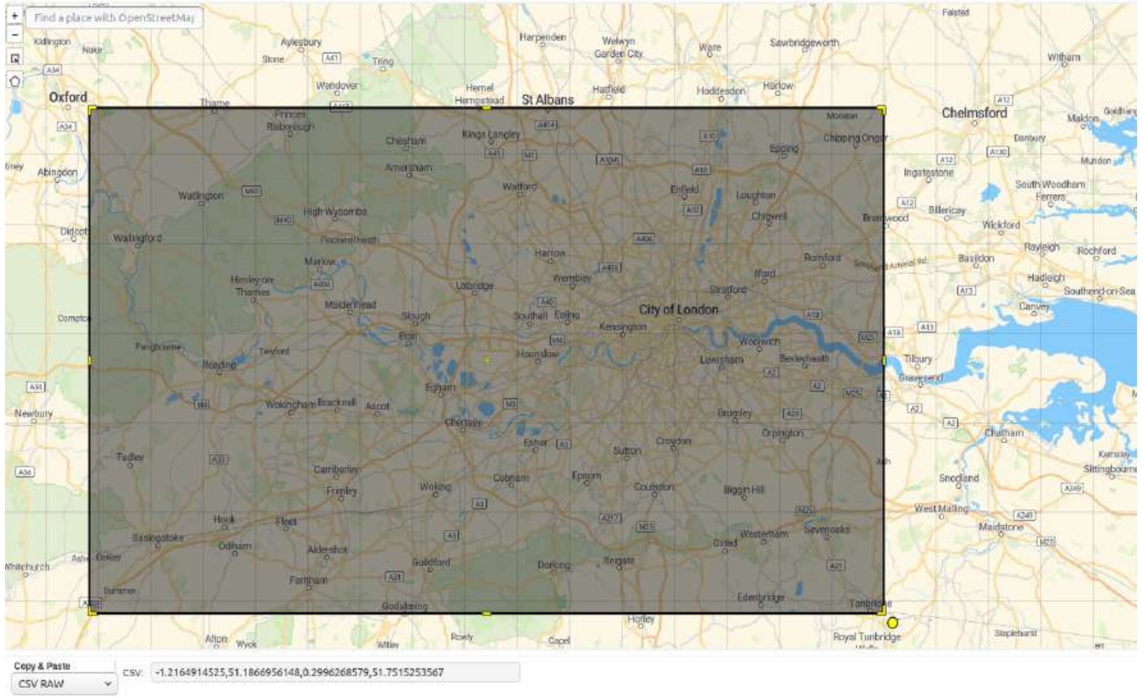
- Heathrow airport L_{den} noise contours **[3]** ➔ set the minimum size of the area
- Bounding box of 167 km wide and 73 km long



⚠ Gets only tweets with enabled location ➔ small proportion

❑ Retrieve tweets with keywords filter

- "Heathrow" and "noise" or "LHR" and "noise" to filter tweets



Both retrieval methods are configured to get only English language tweets.

➔ Two datasets: tweets with location query (TWLQ) and tweets with keywords query (TWKQ)

Workflow: Preprocessing tweets

Preprocessing tweets		
For both Sentiment and relevance classification	Specific to relevance classification	Specific to sentiment classification
URL and punctuation removal	Numbers removal	Username removal
Tokenization (split words into tokens)	Stop words removal (a, the, this, can, etc.)	Emoticons extraction
Part-of-speech (POS) tagging	Lemmatization (bring words into their respective root words)	Intensifier detection: character repetition (e.g. “happyyyy”) and all caps (e.g. “NOISE”)
Spelling correction (character repetition case)	-	Detect negation words and the respective scope
-	-	Detect contrast words

9

Workflow: Relevance classification

Tweets fed as a *tf-idf* matrix (t_i^T is tweet i in corpus T, d_j is term j):

$$\begin{array}{cc} & \begin{array}{ccc} & \mathbf{d}_j & \\ & \downarrow & \end{array} \\ \mathbf{t}_i^T \rightarrow & \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{pmatrix} \end{array}$$

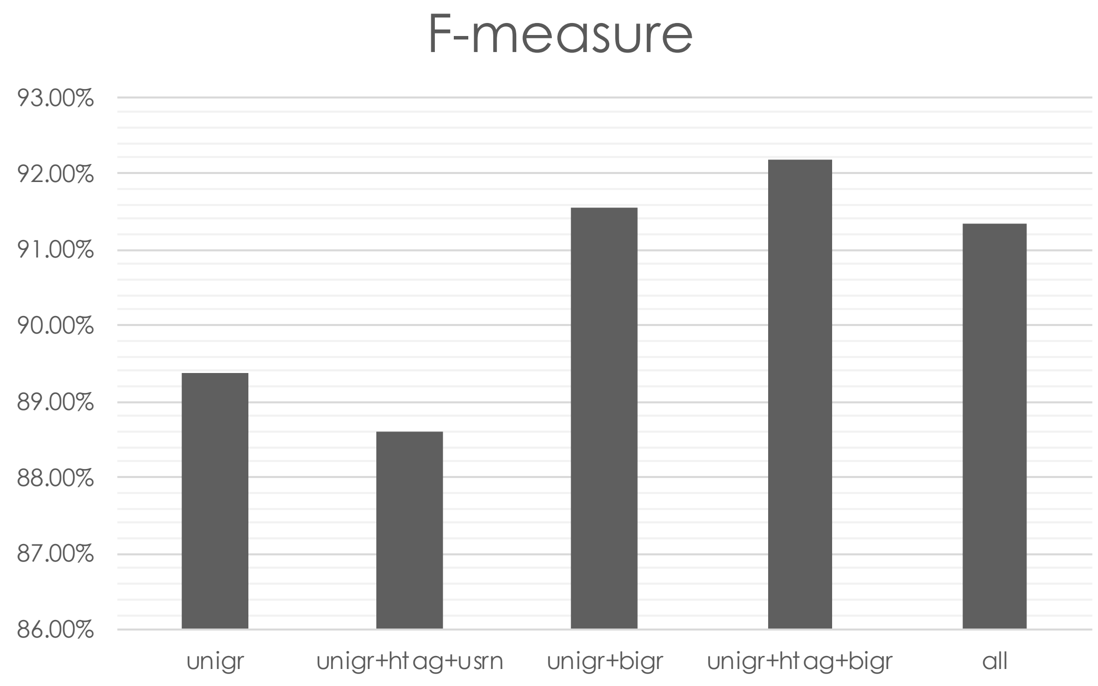
Numbers / Datasets	Relevant tweets	Irrelevant tweets	Total tweets
S_1 (training)	432	543	975
S_2 (testing)	445	524	969

Workflow: Relevance classification

Tweets fed as a *tf-idf* matrix (t_i^T is tweet i in corpus T, d_j is term j):

$$\begin{array}{cc} & \mathbf{d}_j \\ & \downarrow \\ \mathbf{t}_i^T \rightarrow & \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{pmatrix} \end{array}$$

Numbers / Datasets	Relevant tweets	Irrelevant tweets	Total tweets
S_1 (training)	432	543	975
S_2 (testing)	445	524	969



Features selection:

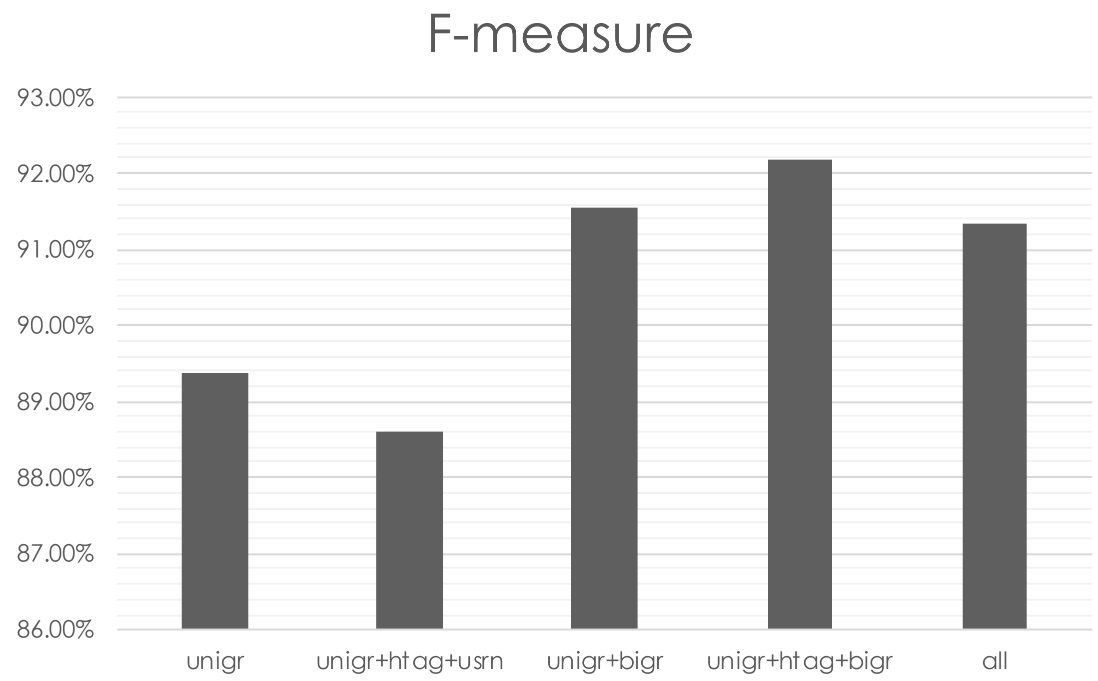
- Unigrams (**unigr**)
- Unigrams, hashtags and usernames (**unigr+htag+usrn**)
- Unigrams and bigrams (**unigr+bigr**)
- Unigrams, hashtags and bigrams (**unigr+htag+bigr**)
- Unigrams, bigrams, hashtags and usernames (**all**)

Workflow: Relevance classification

Tweets fed as a *tf-idf* matrix (t_i^T is tweet i in corpus T, d_j is term j):

$$\begin{array}{cc} & \mathbf{d}_j \\ & \downarrow \\ \mathbf{t}_i^T \rightarrow & \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{pmatrix} \end{array}$$

Datasets \ Numbers	Relevant tweets	Irrelevant tweets	Total tweets
S_1 (training)	432	543	975
S_2 (testing)	445	524	969



Features selection:

- Unigrams (**unigr**)
- Unigrams, hashtags and usernames (**unigr+htag+usrn**)
- Unigrams and bigrams (**unigr+bigr**)
- Unigrams, hashtags and bigrams (**unigr+htag+bigr**)
- Unigrams, bigrams, hashtags and usernames (**all**)

➔ Features: unigrams, bigrams and hashtags.

Workflow: Relevance classification

Tweets fed as a *tf-idf* matrix (t_i^T is tweet i in corpus T, d_j is term j):

$$\begin{array}{cc} & \mathbf{d}_j \\ & \downarrow \\ \mathbf{t}_i^T \rightarrow & \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{pmatrix} \end{array}$$

Datasets \ Numbers	Relevant tweets	Irrelevant tweets	Total tweets
S_1 (training)	432	543	975
S_2 (testing)	445	524	969

Machine learning classifier: Multinomial Naïve Bayes (MNB) **[5]**, Support Vector Machine **[2]** (with linear kernel $\mathrm{SVM}_{\mathrm{Linear}}$ and Radial Basis Function (RBF) kernel $\mathrm{SVM}_{\mathrm{RBF}}$) and Random Forests **[7]**.

	MNB	$\mathrm{SVM}_{\mathrm{Linear}}$	$\mathrm{SVM}_{\mathrm{RBF}}$	RF
F-measure	90.74%	95.53%	95.64%	94.46%

➔ $\mathrm{SVM}_{\mathrm{RBF}}$

➔ Features: unigrams, bigrams and hashtags.

Workflow: Relevance classification

Tweets fed as a *tf-idf* matrix (t_i^T is tweet i in corpus T, d_j is term j):

$$\begin{array}{cc} & \mathbf{d}_j \\ & \downarrow \\ \mathbf{t}_i^T \rightarrow & \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{pmatrix} \end{array}$$

Datasets \ Numbers	Relevant tweets	Irrelevant tweets	Total tweets
S_1 (training)	432	543	975
S_2 (testing)	445	524	969

➔ $\mathrm{SVM}_{\mathrm{RBF}}$

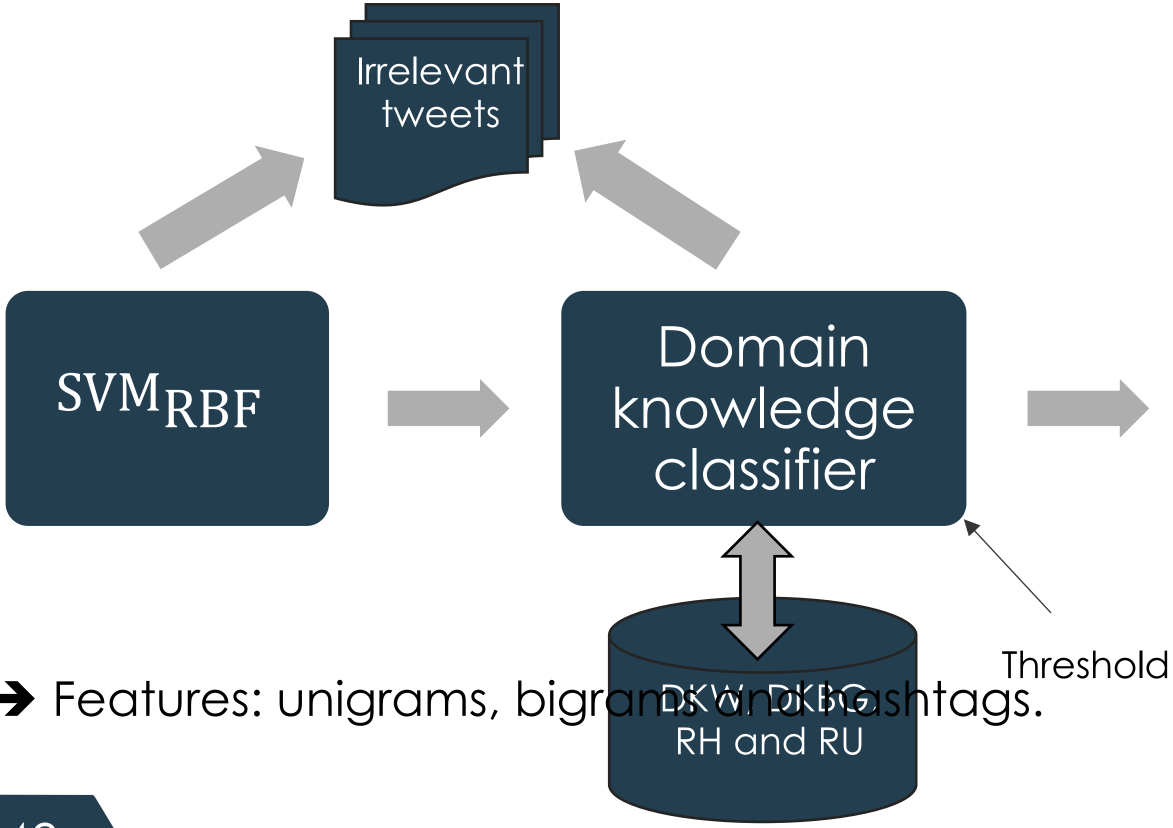
➔ Features: unigrams, bigrams and hashtags.

Workflow: Relevance classification

Tweets fed as a *tf-idf* matrix (t_i^T is tweet i in corpus T, d_j is term j):

$$\begin{array}{cc} & \mathbf{d}_j \\ & \downarrow \\ \mathbf{t}_i^T \rightarrow & \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{pmatrix} \end{array}$$

Datasets \ Numbers	Relevant tweets	Irrelevant tweets	Total tweets
S_1 (training)	432	543	975
S_2 (testing)	445	524	969



➔ Features: unigrams, bigrams and hashtags.

Domain knowledge classifier (Lexicon-based):

- Domain knowledge words (DKW) and bigrams (DKBG). ➔SVM$_{RBF}$
- Topic related hashtags (RH) and usernames (RU).

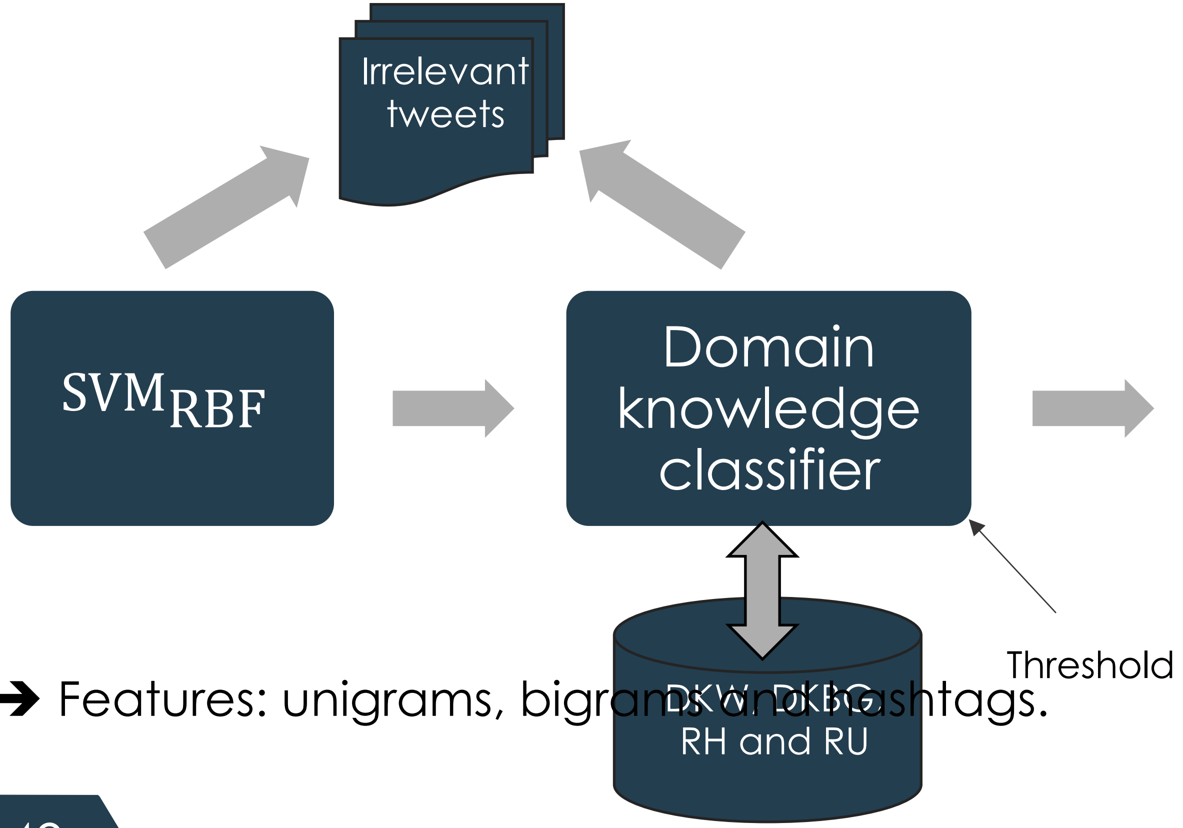
9

Workflow: Relevance classification

Tweets fed as a *tf-idf* matrix (t_i^T is tweet i in corpus T, d_j is term j):

$$\begin{array}{cc} & \mathbf{d}_j \\ & \downarrow \\ \mathbf{t}_i^T \rightarrow & \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{pmatrix} \end{array}$$

Datasets \ Numbers	Relevant tweets	Irrelevant tweets	Total tweets
S_1 (training)	432	543	975
S_2 (testing)	445	524	969



➔ Features: unigrams, bigrams and hashtags.

Domain knowledge classifier (Lexicon-based):

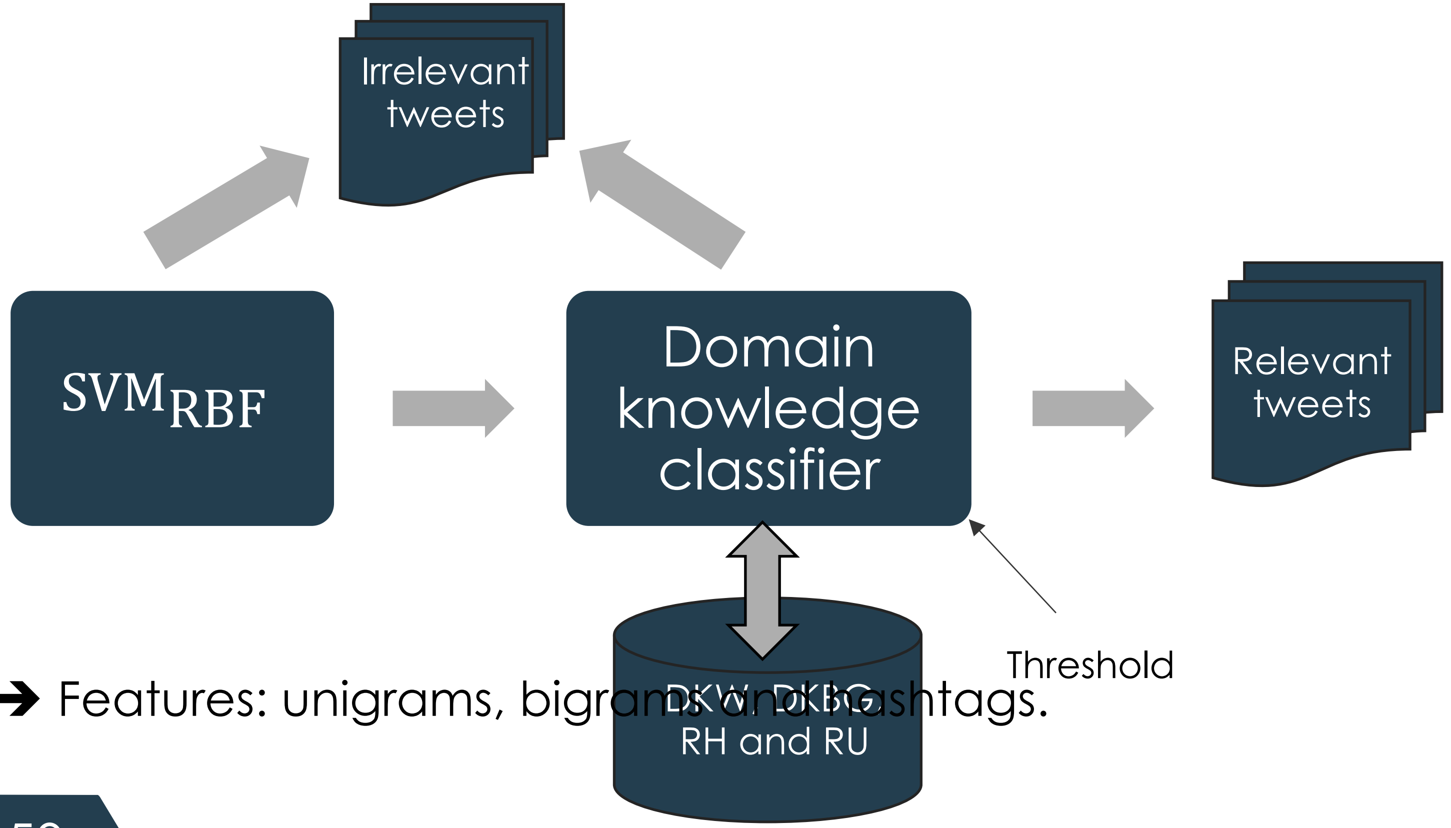
- Domain knowledge words (DKW) and bigrams (DKBG).
- Topic related hashtags (RH) and usernames (RU).

➔SVMRBF

➔ Reduce the number of false relevant tweets.

Workflow: Relevance classification

Tweet: "@HeathrowAirport Stop these noise sewers, my kids are not sleeping. scum"



➔ Features: unigrams, bigrams and hashtags.

Domain knowledge classifier (Lexicon-based):

- Domain knowledge words (DKW) and bigrams (DKBG).
- Topic related hashtags (RH) and usernames (RU).

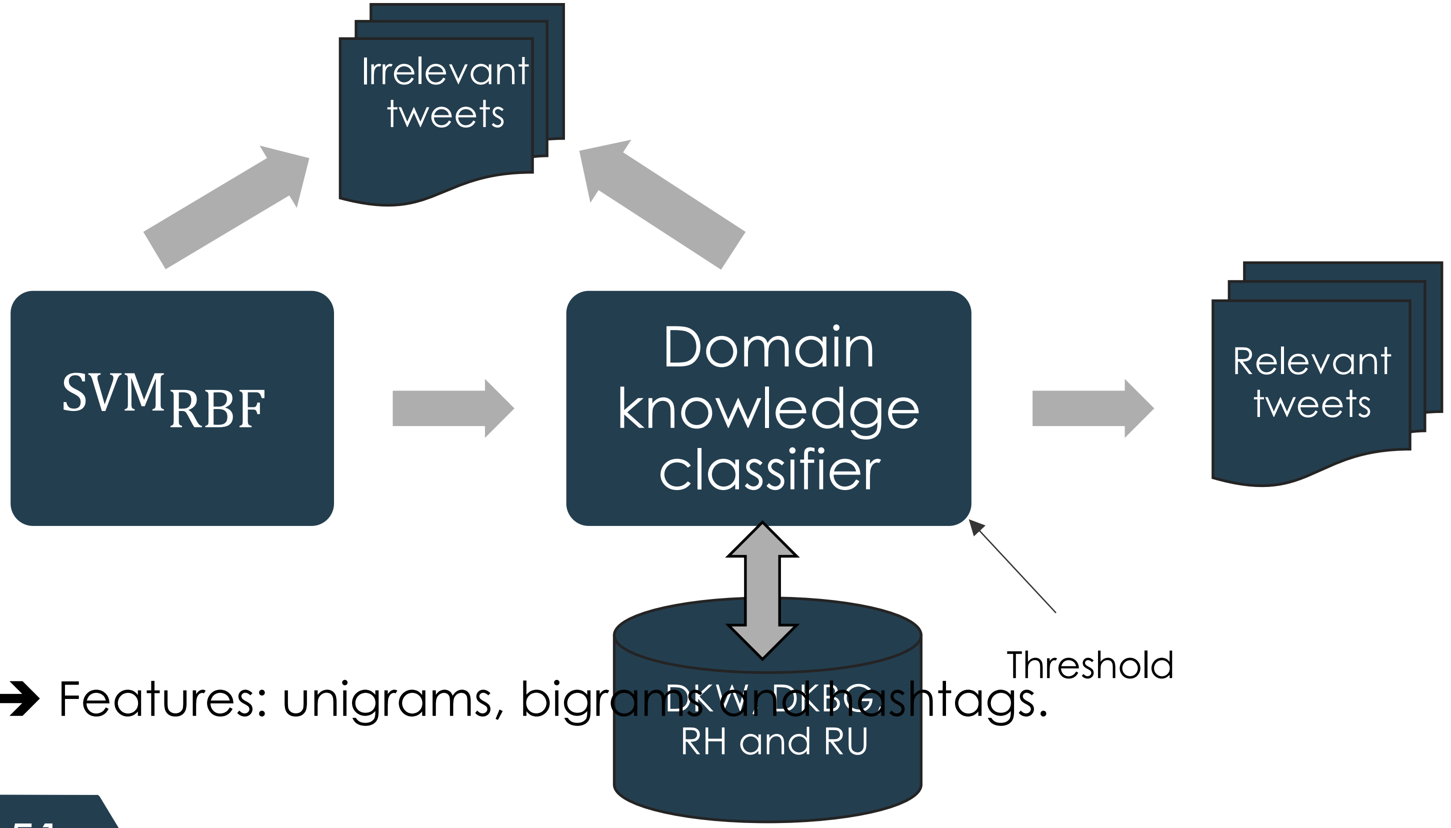
➔ SVM_{RBF}

➔ Reduce the number of false relevant tweets.

Workflow: Relevance classification

Tweet: "@HeathrowAirport Stop these noise sewers, my kids are not sleeping. scum"

$$\left.\begin{array}{l}\text{Relevance score} = \frac{4}{4} = 1 \\ \text{Threshold} = 0{,}5\end{array}\right\} \text{Relevant}$$



➔ Features: unigrams, bigrams and hashtags.

Domain knowledge classifier (Lexicon-based):

- Domain knowledge words (DKW) and bigrams (DKBG). ➔ SVM_{RBF}
- Topic related hashtags (RH) and usernames (RU).

➔ Reduce the number of false relevant tweets.

Workflow: Sentiment classification

Sentiment classification: Emoticons, lexicon and SWN

Emoticon (Em) score calculation:

- uses datasets of positive and negative emoticons (PE and NE)

$$emscore(e_j) = \begin{cases} 1, (e_j \in PE) \\ -1, (e_j \in NE) \\ 0, (e_j \notin PE) \land (e_j \notin NE) \end{cases} \quad (1)$$

Lexicon polarity (LP) score calculation:

- uses datasets of positive and negative words (PW and NW)
- Datasets are created from multiple existing lexicon (MPQA **[13]**, Bing Liu **[1]** and Bill McDonald **[11]** ➔ 10529 words).
- Subjective intensity of words: strong subjectivity, weak subjectivity and unknown subjectivity (weights)
- Intensifiers (ACI set of all caps intensifier score and CRI set of character repetition intensifier score of a tweet.

$$swscore_{LP}(w_j) = \begin{cases} 1 \times weight \times aci_j \times cri_j, (w_j \in PW) \land (aci_j \in ACI) \land (cri_j \in CRI) \\ (-1) \times weight \times aci_j \times cri_j, (w_j \in NW) \land (aci_j \in ACI) \land (cri_j \in CRI) \\ 0, (w_j \notin PW) \land (w_j \notin NW) \end{cases} \quad (2)$$

Workflow: Sentiment classification

- Negation with dynamic scope: stops when sentences ends (e.g. “,”, “.”, “-”, “!”, “?”) or when conjunction or contrast words are found (e.g. “and”, “where”, “which”, ”but”).➔ opposite scores of words in the scope.
- Contrast effect: take the opposite scores of all words before the contrast word. Is not operating at sentence level.

Tweet: “@HeathrowAirport Stop these noise sewers, my kids are not sleeping. scum”

Workflow: Sentiment classification

- Negation with dynamic scope: stops when sentences ends (e.g. “,”, “.”, “-”, “!”, “?”) or when conjunction or contrast words are found (e.g. “and”, “where”, “which”, ”but”).➔ opposite scores of words in the scope.
- Contrast effect: take the opposite scores of all words before the contrast word. Is not operating at sentence level.

Tweet: “@HeathrowAirport Stop these noise sewers, my kids are not sleeping. scum”

Workflow: Sentiment classification

- Negation with dynamic scope: stops when sentences ends (e.g. ",", ".", "-", "!", "?") or when conjunction or contrast words are found (e.g. "and", "where", "which", "but").➔ opposite scores of words in the scope.
- Contrast effect: take the opposite scores of all words before the contrast word. Is not operating at sentence level.

Tweet: "@HeathrowAirport Stop these noise sewers, my kids are not sleeping. scum"

Negation scope 1

Negation scope 2

Workflow: Sentiment classification

- Negation with dynamic scope: stops when sentences ends (e.g. ",", ".", "-", "!", "?") or when conjunction or contrast words are found (e.g. "and", "where", "which", "but").➔ opposite scores of words in the scope.
- Contrast effect: take the opposite scores of all words before the contrast word. Is not operating at sentence level.

Tweet: "@HeathrowAirport Stop these noise sewers, my kids are not sleeping. scum"

Negation scope 1

Negation scope 2

Workflow: Sentiment classification

- Negation with dynamic scope: stops when sentences ends (e.g. ",", ".", "-", "!", "?") or when conjunction or contrast words are found (e.g. "and", "where", "which", "but").➔ opposite scores of words in the scope.
- Contrast effect: take the opposite scores of all words before the contrast word. Is not operating at sentence level.

Tweet: "@HeathrowAirport Stop these noise sewers, my kids are not sleeping. scum"

Negation scope 1

Negation scope 2

$LPscore = 0$

$j = 9$

$swscore_{LP}(sleeping) = (1) * (1) * (1) * (1) = 1$

$swscore_{LP}(sleeping) = (-1) * swscore_{LP}(sleeping) = (-1) * (1) = -1$

$LPscore = 0 + (-1)$

$j = 10$

$swscore_{LP}(scum) = (-1) * (1) * (1) * (1) = -1$

$LPscore = (-1) + (-1) = -2$

Workflow: Sentiment classification

- Negation with dynamic scope: stops when sentences ends (e.g. ",", ".", "-", "!", "?") or when conjunction or contrast words are found (e.g. "and", "where", "which", "but").➔ opposite scores of words in the scope.
- Contrast effect: take the opposite scores of all words before the contrast word. Is not operating at sentence level.

Tweet: "@HeathrowAirport Stop these noise sewers, my kids are not sleeping. scum"

Negation scope 1

Negation scope 2

$LPscore = 0$

$j = 9$

$swscore_{LP}(sleeping) = (1) * (1) * (1) * (1) = 1$

$swscore_{LP}(sleeping) = (-1) * swscore_{LP}(sleeping) = (-1) * (1) = -1$

$LPscore = 0 + (-1)$

$j = 10$

$swscore_{LP}(scum) = (-1) * (1) * (1) * (1) = -1$

$LPscore = (-1) + (-1) = -2$

$$score_{LP}(\text{Tweet}) = \frac{(-2)}{10} = -0.2$$

Workflow: Sentiment classification

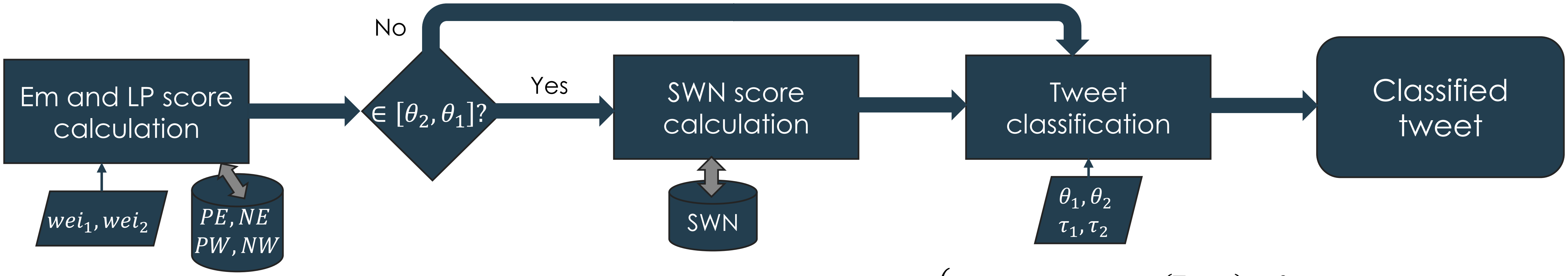
SWN score calculation: uses SWN dataset to get positive/negative score of word w_j and its synsets sy_j in SY_{w_j}

$$syscore_{SWN}(sy_j) = posscore_{SWN}(sy_j) - negscore_{SWN}(sy_j), (sy_j \in SY_{w_j}) \quad (3)$$

Proposed sentiment classifier (PC) score calculation and classification:

- Hierarchical use of Em, LP and SWN scores by weights and priority steps. First, use Em and LP score of the tweet to classify

$$score_{Em+LP}(\text{Tweet}) = wei_1 \times score_{Em}(\text{Tweet}) + wei_2 \times score_{LP}(\text{Tweet}) = 0{,}7\times 0 + 0{,}3\times(-0{,}2) = -0{,}06 \quad (4)$$



1st classification case: thresholds θ_1 and θ_2

$$sclass_{Em+LP}(\text{Tweet}) = \begin{cases} positive, score_{Em+LP}(\text{Tweet}) > \theta_1 \\ negative, score_{Em+LP}(\text{Tweet}) < \theta_2 \\ score_{SWN}(rt_j), score_{Em+LP}(\text{Tweet}) \in [\theta_2, \theta_1] \end{cases} \quad (5)$$

2nd classification case: thresholds τ_1 and τ_2

$$sclass_{SWN}(\text{Tweet}) = \begin{cases} positive, score_{SWN}(\text{Tweet}) > \tau_1 \\ negative, score_{SWN}(\text{Tweet}) < \tau_2 \\ neutral, score_{SWN}(\text{Tweet}) \in [\tau_2, \tau_1] \end{cases} \quad (6)$$

Experimental results: Classifiers comparison

	Positive tweets	Negative tweets	Neutral tweets	Total	Tweets with emoticons
D_1	26	601	26	653	18

$wei_1 = 0.7$ and $wei_2 = 0.3$

$$weight \begin{cases} \text{strong subjectivity: } 1 \\ \text{weak subjectivity: } 0{,}75 \\ \text{unknown subjectivity: } 0{,}75 \end{cases}$$

D_1			Confusion matrix			Metrics			
			Positive	Negative	Neutral	$precision$	$recall$	$F-measure$	$accuracy$
EmC	Thresholds	Positive	0	0	26	0 %	0 %	-	4.90 %
	$\tau_1 = \tau_2 = 0$	Negative	3	6	592	100%	1%	1,98 %	
		Neutral	0	0	26	4.04 %	100%	7.76%	
LPC	Thresholds	Positive	10	7	9	12.50 %	38.46 %	18.86 %	62.17 %
	$\tau_1 = 0.027$ $\tau_2 = -0.001$	Negative	66	380	155	96.69 %	63.23 %	76.46 %	
		Neutral	4	6	16	8.89 %	61.54 %	15.53 %	
SWNC	Thresholds	Positive	4	15	7	4.44 %	15.38 %	6.90 %	69.37%
	$\tau_1 = 0.015$ $\tau_2 = 0.005$	Negative	79	443	79	95.05 %	73.71 %	82.64 %	
		Neutral	7	13	6	6.52 %	23.07 %	10.17%	
PC	Thresholds	Positive	12	10	4	12.12%	46.15%	19.20 %	77.79 %
	$\theta_1 = 0.01$ $\theta_2 = -0.001$ $\tau_1 = 0.015$ $\tau_2 = 0.004$	Negative	80	491	30	95.34 %	81.70 %	87.99 %	
		Neutral	7	14	5	12.82 %	19.23 %	15.38 %	

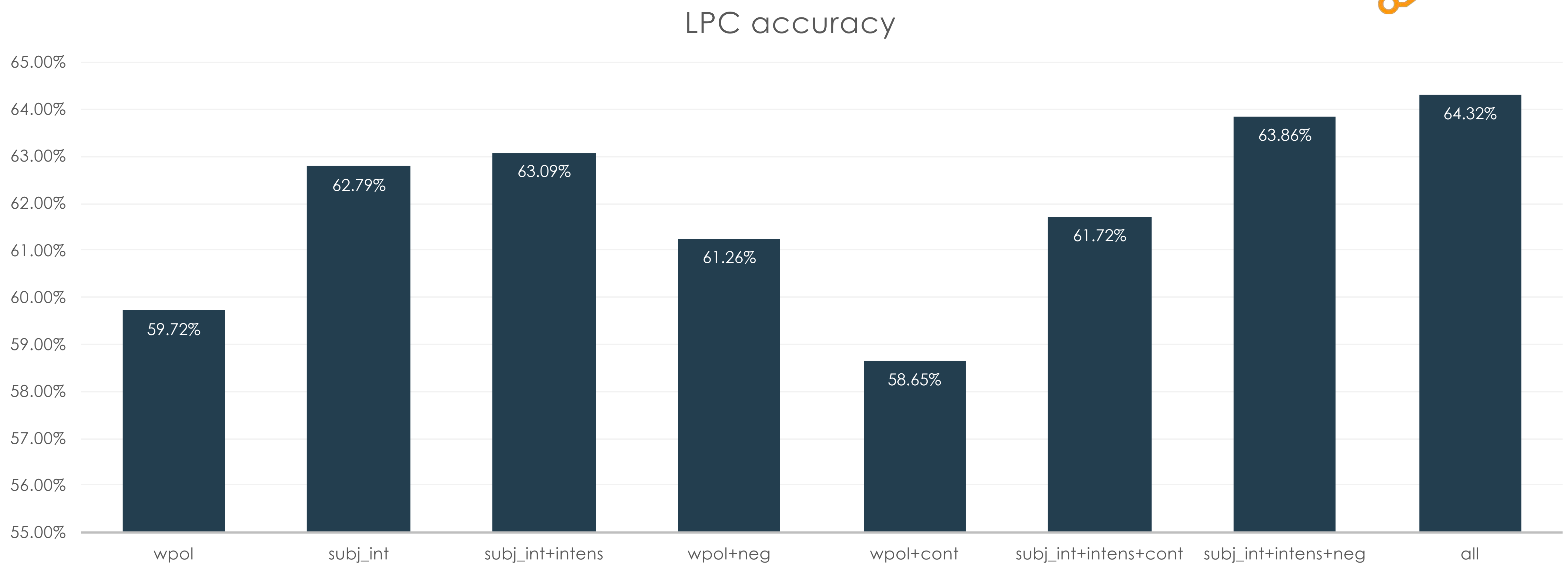
➔ Detects only tweets with emoticons

➔ Many false neutral tweets

➔ Few true positive and true neutral tweets

➔ Decrease false positive and false neutral and increase true positive

Experimental results: LPC features comparison



wpol: word polarity

subj_int: subjective intensity of words

subj_in+intens: subjective intensity of words and intensifiers

wpol+neg: word polarity and negation

wpol+cont: word polarity and contrast

subj_int+intens+cont: subjective intensity, intensifiers and contrast

subj_int+intens+neg: subjective intensity, intensifiers and negation

all: subjective intensity, intensifiers, negation and contrast

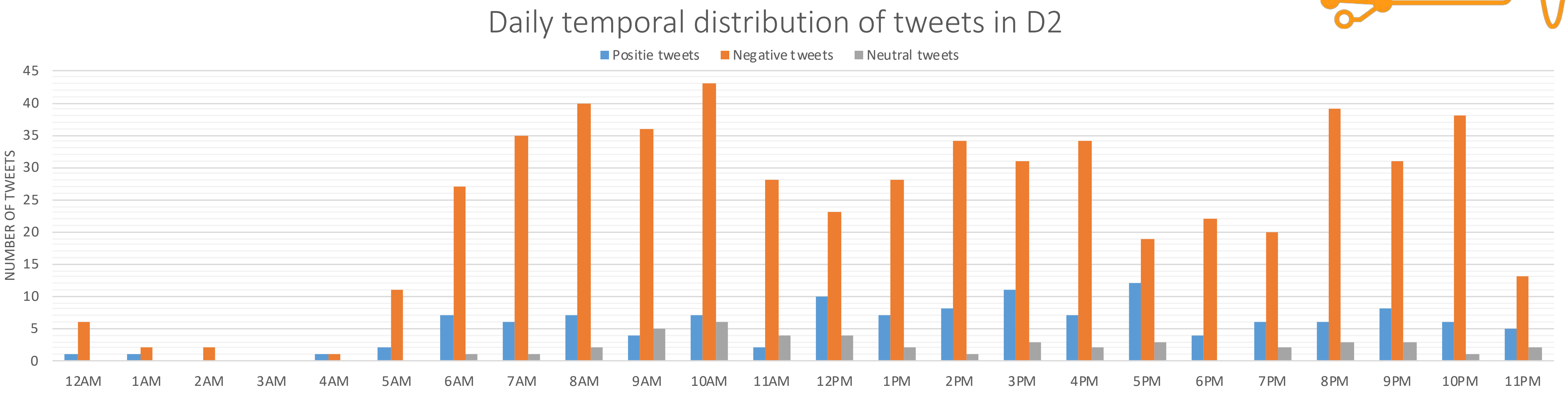
Post processing

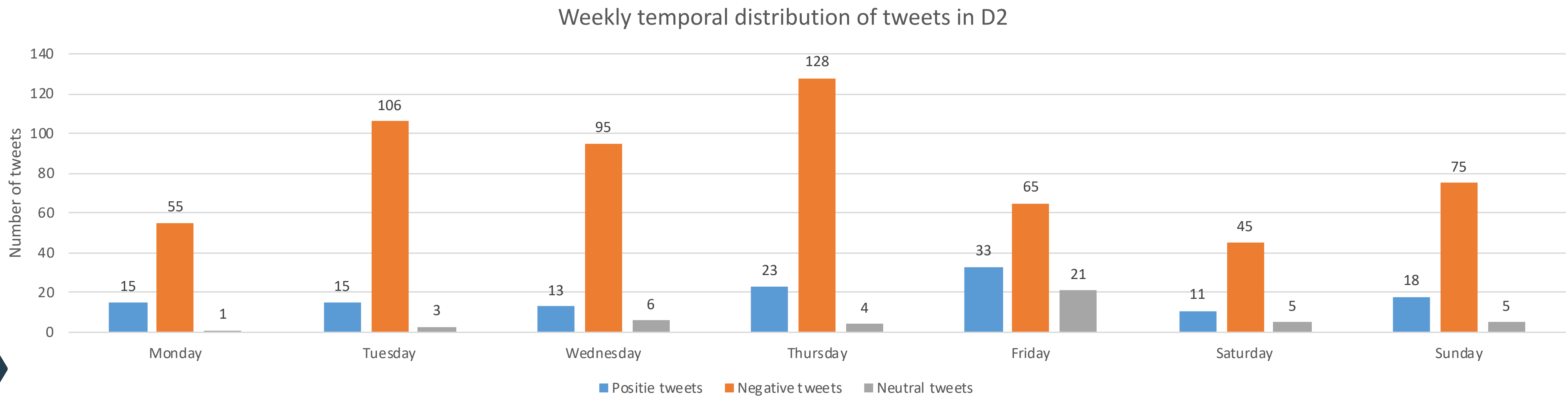
D_2	Relevant tweets	Total	Tweets with location	Positive tweets	Negative tweets	Neutral tweets
TWLQ	265	742	250	128	569	45
TWKQ	477					

D_2	Most mentioned user	Most used hashtags	Most retweeted tweets
Positive tweets	@HeathrowNoise: **17** @BBCParliament: **12** @StopHeathrowExp: **11**	#Heathrow: **8** #HeathrowExpansion: **2** #HeathrowNoise: **2**	"RT .@BBCParliament MPs who vote in favour of a third runway at Heathrow will be condemning yet more people to the torture of aircraft noise. Which ones think it's a good idea to fly another 260,000 flights pa over this densely-populated area?": **11** "RT @HeathrowNoise See? It's reversible, as we always knew: "The changes were made under an agreement to provide aircraft noise relief to residents of historic neighborhoods": **4**
Negative tweets	@HeathrowNoise: **164** @TeddingtonTAG: **45** @NeilSpurrier: **33**	#Heathrow: **52** #heathrow: **28** #care2: **23**	"RT Demand an end to noise sewers... #care2 #heathrow": **21** "RT Here's an online petition to end the 'noise sewers' caused by more concentrated flight paths out of #Heathrow… #twickenham #teddington #whitton #etc": **20**
Neutral tweets	@HeathrowNoise: **29** @NeilSpurrier: **21** @bakerainlondon: **9**	#Heathrow: **1** #care2: **1** #NoiseActionWeek: **1**	"RT @NeilSpurrier @HeathrowNoise JHK said planes at 1000ft by airport boundary, so can Matt or @Heathrownoise clarify why a procedure to 1000ft that only benefits the airport is in a community noise action plan?: **9** "RT One must question whether @yourHeathrow have the slightest intention of being truthful with the communities surrounding the airport or whether their draft Noise Action Plan bears even a passing resemblance to reality. @HeathrowCEB": **7**

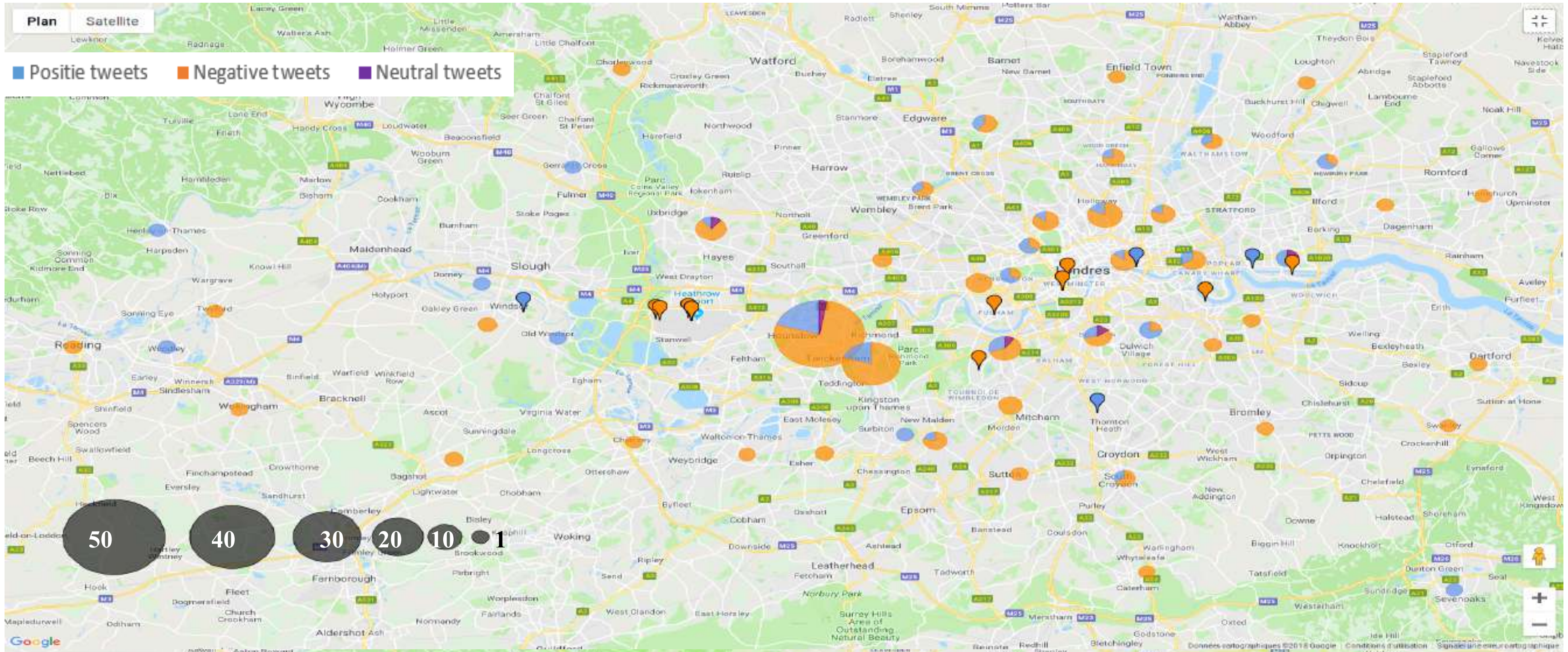
16

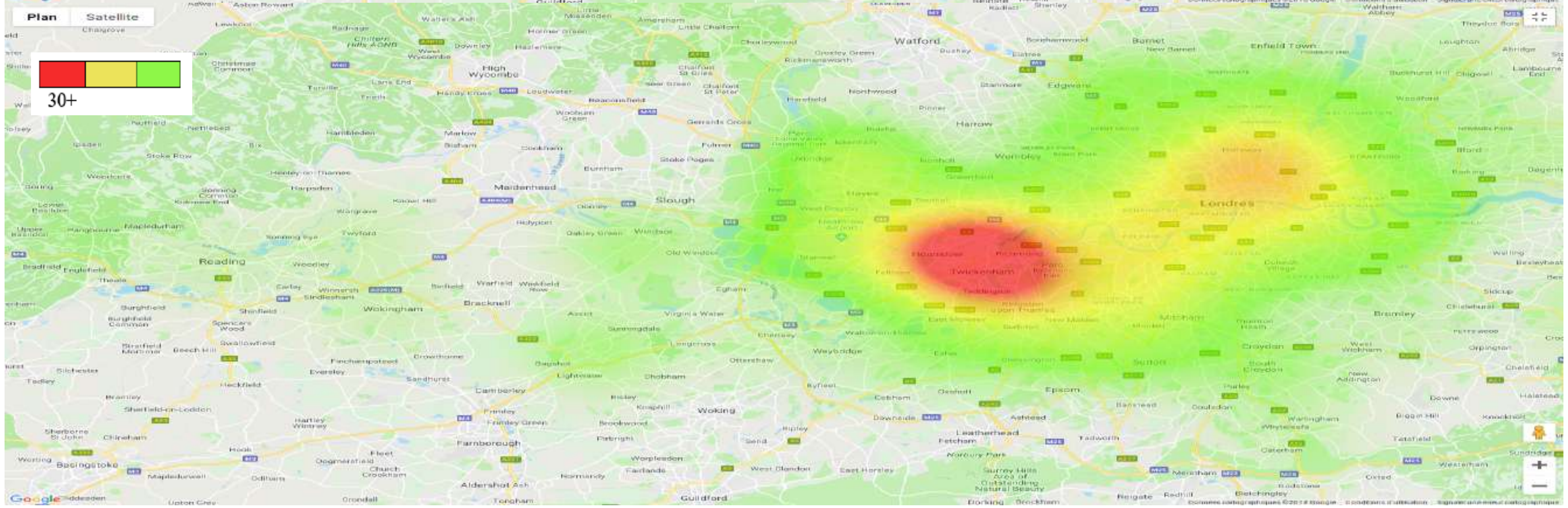
Post processing: Temporal analysis





Post processing: Geospatial analysis





Conclusion and perspectives

- Extract sentiments from twitter messages on airport noise
- Capture relevant tweets from a stream: SVM + Lexicon-based classifier
- Sentiment analysis: emoticons, subjective intensity of words, intensifiers, negation with dynamic scope, contrast and SWN
- Use hierarchy of emoticons, lexicon polarity and SWN scores to classify tweets
- Classified tweets are used to understand causes, time, areas and to extract related topics

- Improve relevance classifier: apply more domain knowledge features (e.g. presence of related expressions)
- Improve sentiment classifier: expand lexicon dataset, more types of spelling errors to correct, contrast effect at sentence level
- Post processing: compare affected areas and flight paths, extract related events not only with hashtags and create users links graph

Acknowledgements

This work has been partially supported by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 769627 - **ANIMA**.

www.anima-project.eu

Funded by the Horizon 2020 Framework Programme of the European Union

Bibliography

[1] B. Liu, M. Hu and J. Cheng, "Opinion Observer: Analyzing and Comparing Opinions on the Web."," in *Proceedings of the 14th International World Wide Web conference (WWW-2005)*, Chiba, Japan, 2005.

[2] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning,* pp. 273-297, 1995

[3] Civil Aviation Authority, "Heathrow Airport 2016 Summer Noise Contours and Noise Action Plan Contours," London, 2017.

[4] G. Salton, Automatic Text Processing: The Transformation, Analysis and Retrieval of information by Computer, Boston, Massachusetts, USA: Addison-Wesley, 1988

[5] H. Zhang, "The Optimality of Naive Bayes," in *Proceedings of Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 2004.

[6] J. Read, "Using emoticons to reduce dependency in machine learning techniques for sentiment classification," in *ACL the Association for Computer Liguistics*, 2005

[7] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[8] M. Krieck, J. Dreesman, L. Otrusina and K. Denecke, "A New Age of Public Health: Identifying Disease Outbreaks by Analyzing Tweets," in *Proceedings of Health WebScience Workshop, ACM Web Science Conference*, 2011.

[9] M. Z. Asghar, A. Khan, S. Ahmad, M. Qasim and I. A. Khan, "Lexicon-enhanced sentiment analysis framework using rule-based classification scheme," *PLoS ONE,* 2017.

[10] O. Dan, J. Feng and B. D. Davison, "A Bootstrapping Approach to Indentifying Relevant Tweets for Social TV," in *Proceedings of the Fifth International AAAI Conference on Web and Social Media*, North America, 2011.

Bibliography

[11] T. L. B. McDonald, "When is a Liability not a Liability? Textual Analysis, Dictionaries, and 10-Ks,," *Journal of Finance,* vol. 66, no. 1, pp. 35-65, 2011

[12] T. Sakaki, M. Okazaki and Y. Matsuo, "Earthquake Shakes Twitter Users: Real-Time Event Detection by Social Sensors," in *International World Wide Web Conference*, Raleigh, North Carolina, USA, 2010.

[13] T. Wilson, J. Wiebe and P. Hoffmann, " Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis," in *Proceedings of HLT-EMNLP-2005*, Vancouver, Canada, 2005

[14] Y. Ren, Y. Zhang, M. Zhang and D. Ji, "Context-Sensitive Twitter Sentiment Classification Using Neural Network," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA, 2016


That's all Folks!