Incident Linking: Assigning Tweets to Entries in a Disaster Database

Master's Thesis Defence

Examiners

Prof. Dr. Benno Stein

Prof. Dr.-Ing. Volker Rodehorst

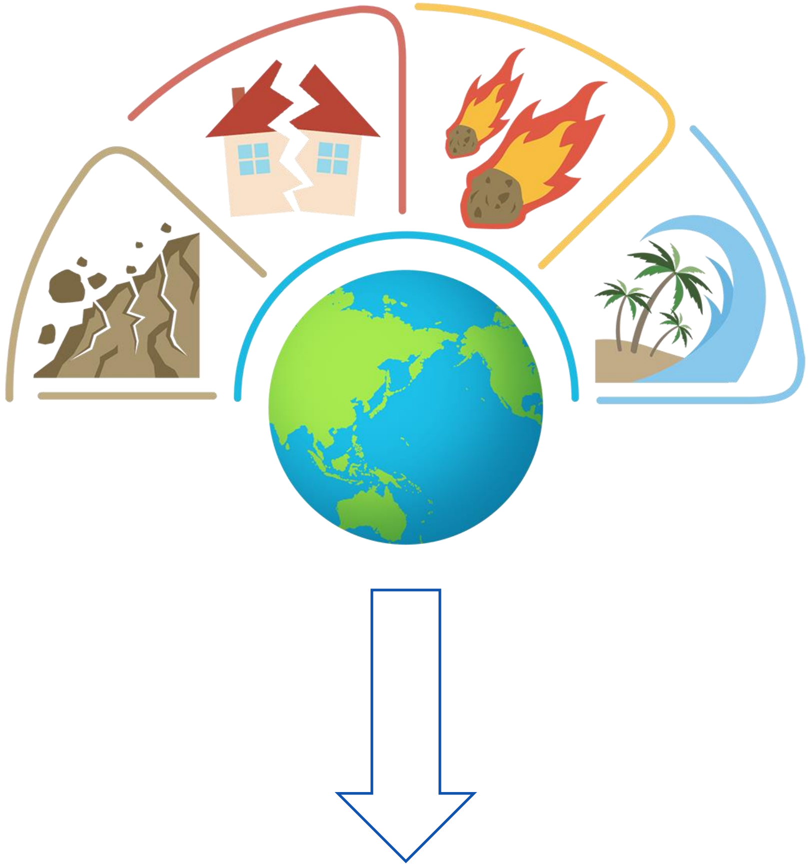
Siva Bathala
119484

CONTENTS

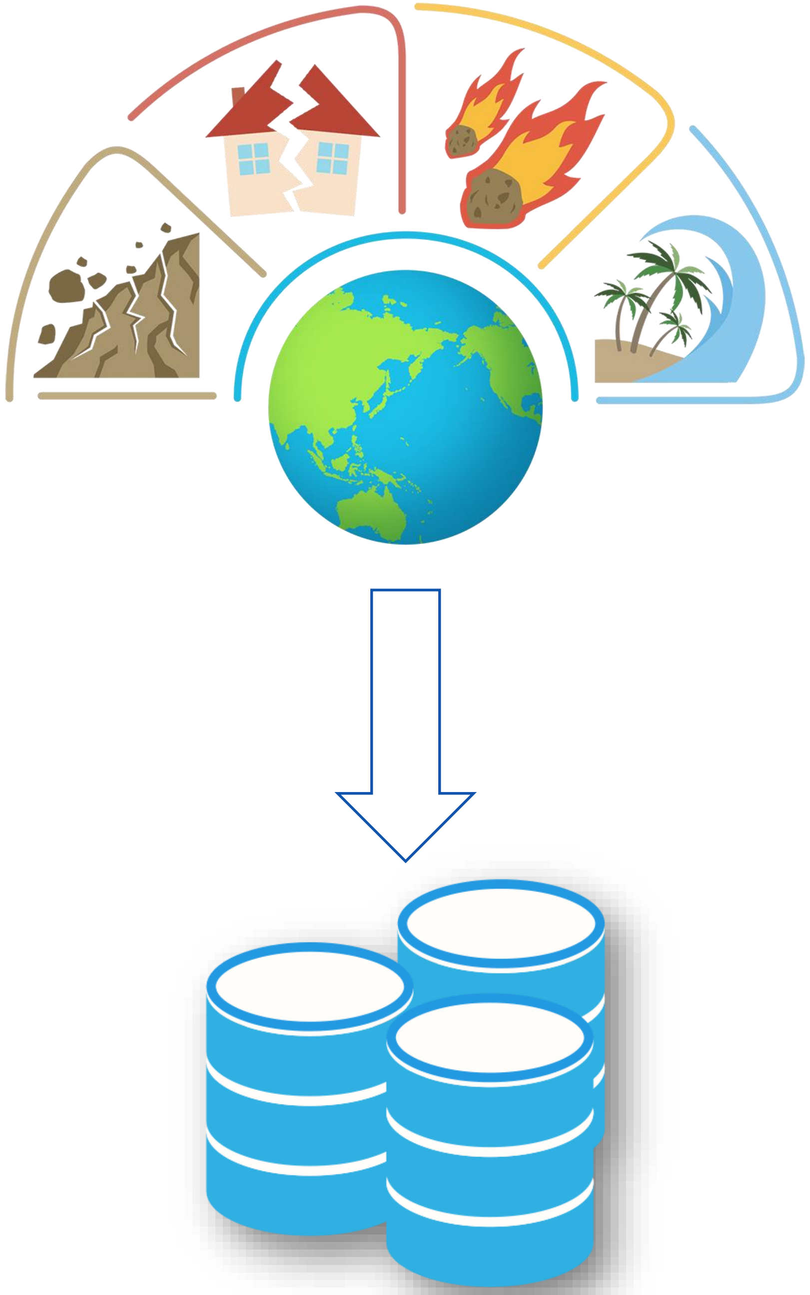
Introduction

Introduction

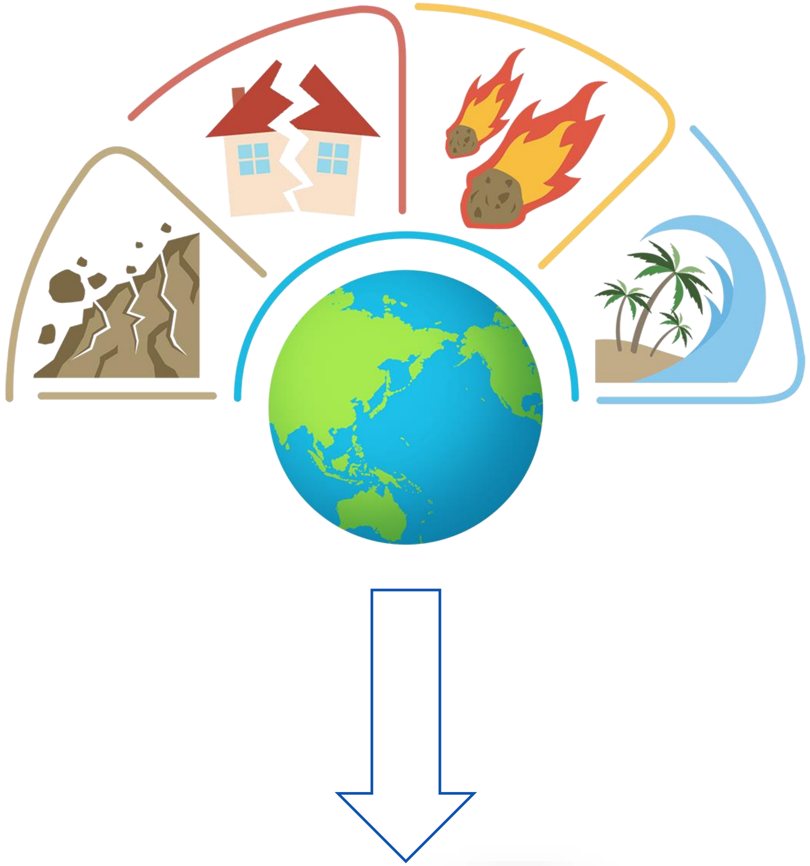
Introduction



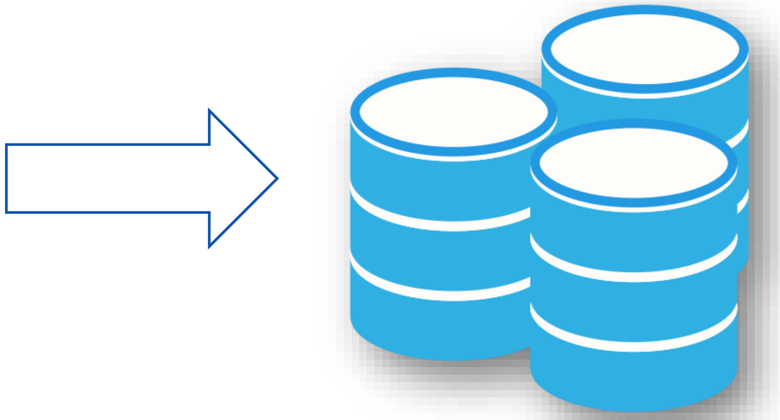
Introduction



Introduction



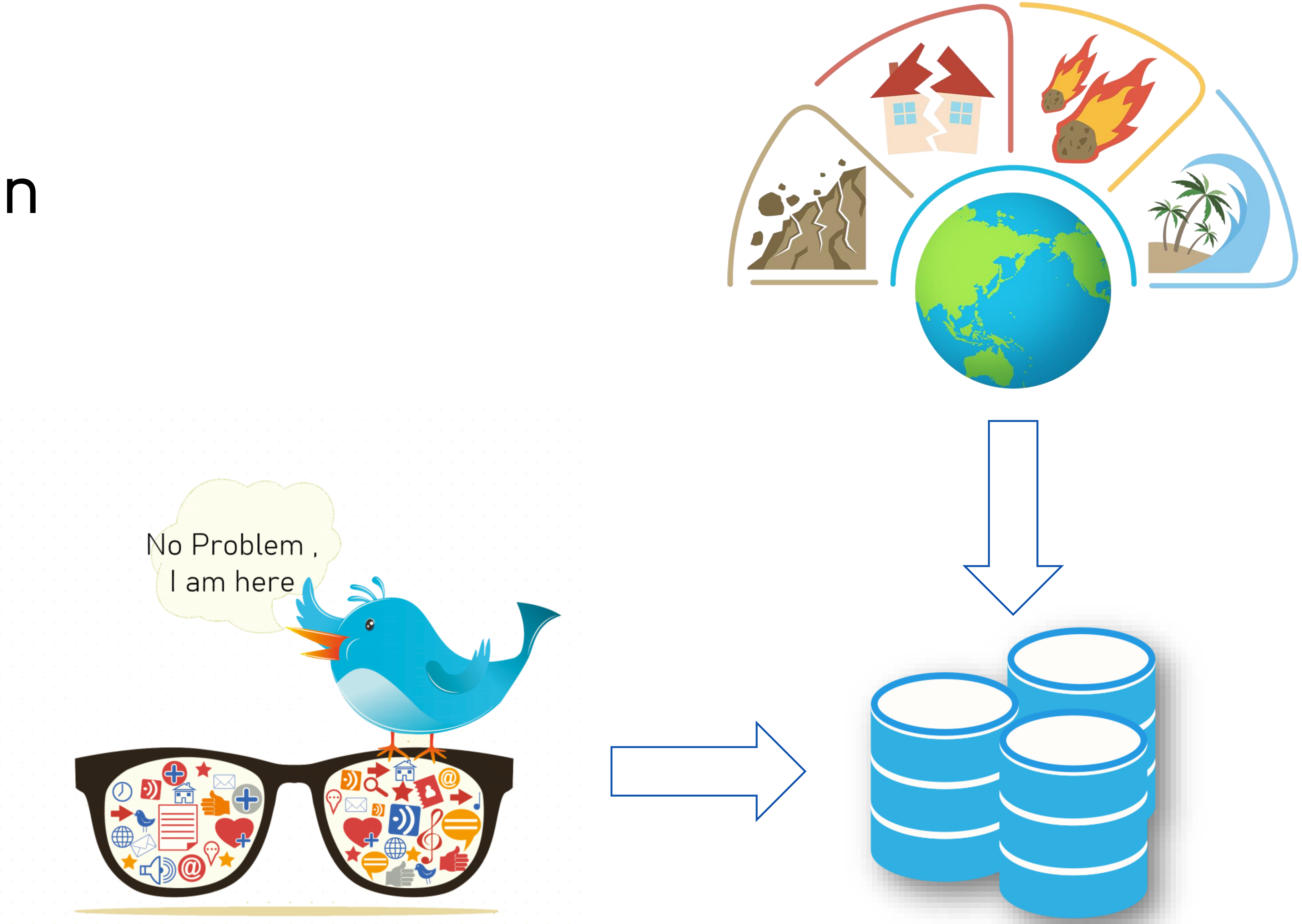
Validation ?



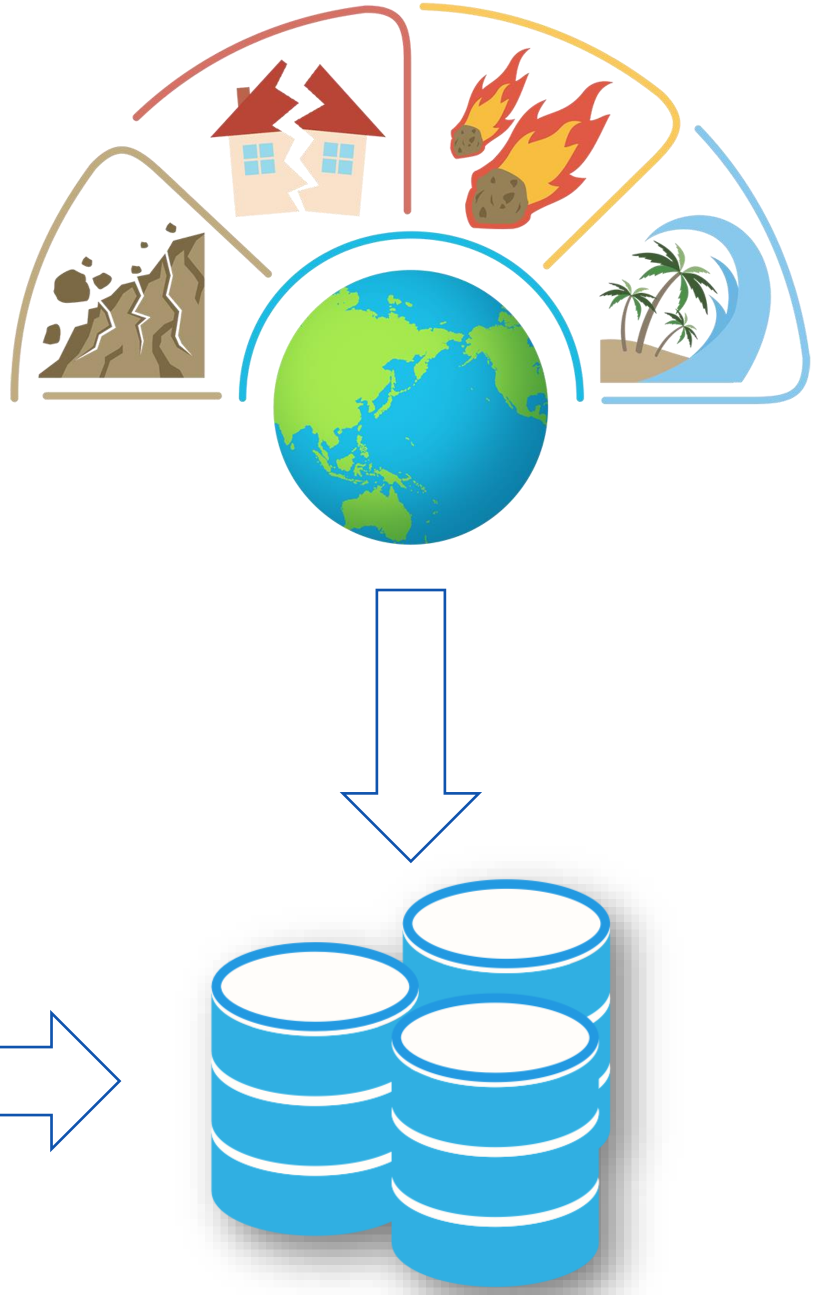
Introduction

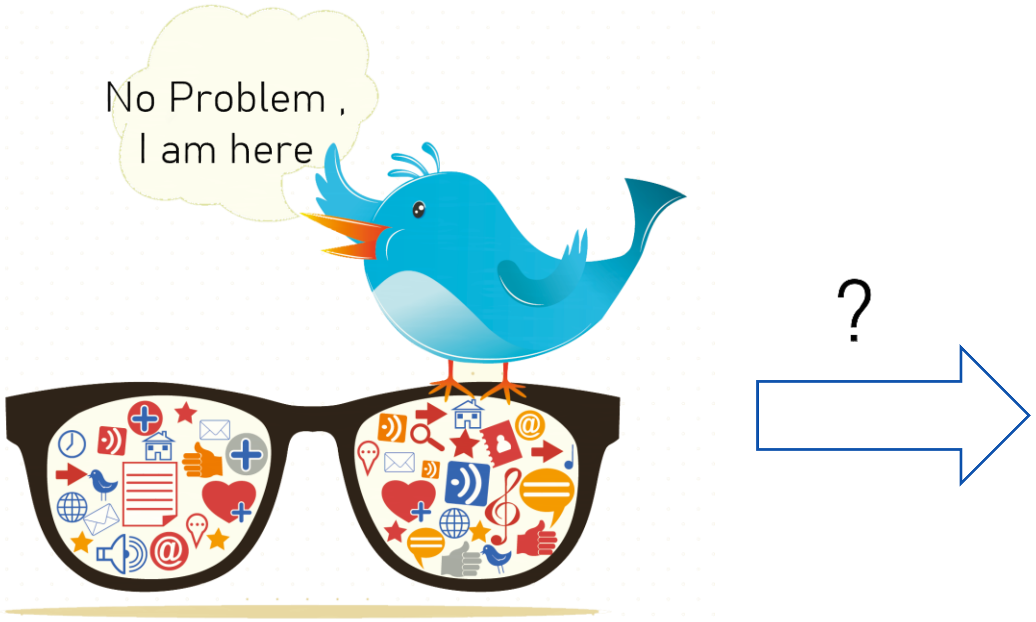
Manual Validation

Introduction



Introduction





Introduction

Our Method

Example

Sample part of incident database entry from EM-DAT	
incident_id	47108fe1-5c04-472c-b534-75a51b747489
type	landslide
start_time	2011-12-08T08:00:00.000Z
location	Colombia ; Bosa ; Bogota
deaths	6

Research Questions

- RQ 1

  What are the possible features that we can extract from tweets that match with those of typical knowledge databases?

Research Questions

- RQ 1

  What are the possible features that we can extract from tweets that match with those of typical knowledge databases?

- RQ 2

  How can we build a linking model that will link the each tweet to entries in the disaster database based on the features from RQ1?

Research Questions

- RQ 1

  What are the possible features that we can extract from tweets that match with those of typical knowledge databases?

- RQ 2

  How can we build a linking model that will link the each tweet to entries in the disaster database based on the features from RQ1?

- RQ 3

  How accurate this model to use for disaster linking?

DATASETS

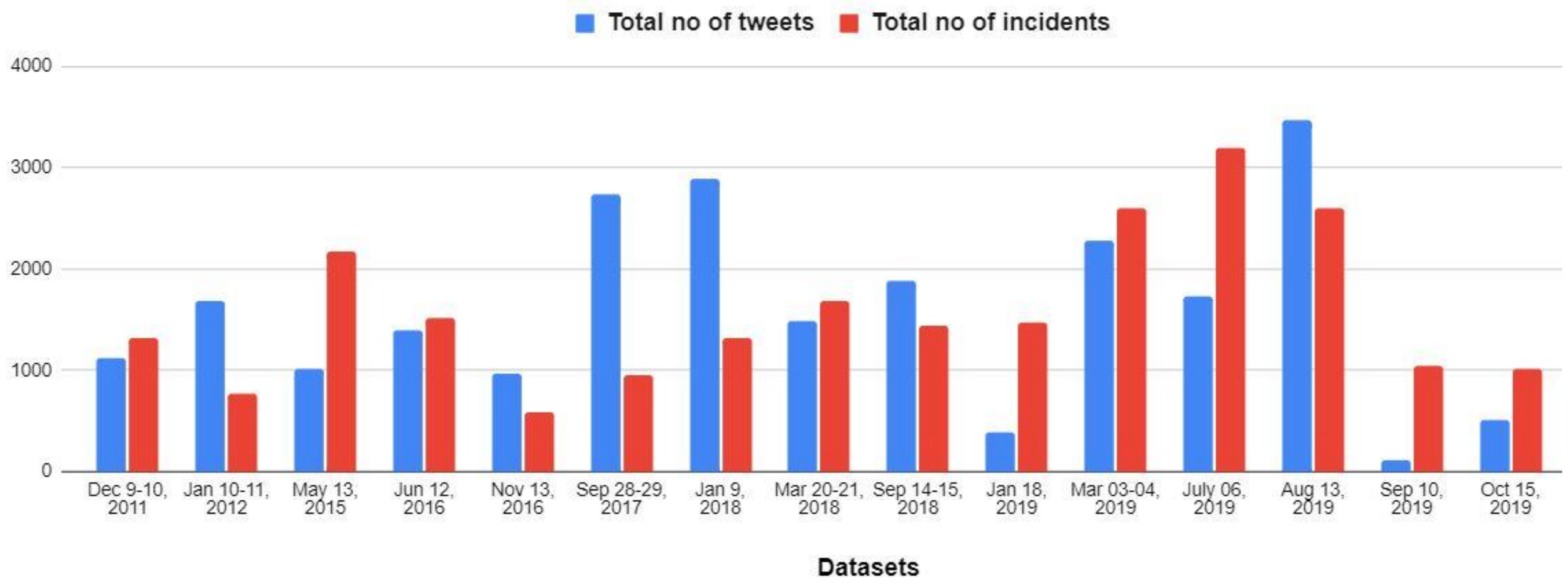
Data

- Tweets and Incidents – December 2011 to October 2019

Tweets Dataset	Incident datasets
23673 Total no of tweets	23723 Total no incidents
15 sets	15 sets
1578 Avg. tweets in each set	1581 Avg. incidents in each set

- Annotations dataset

Data statistics



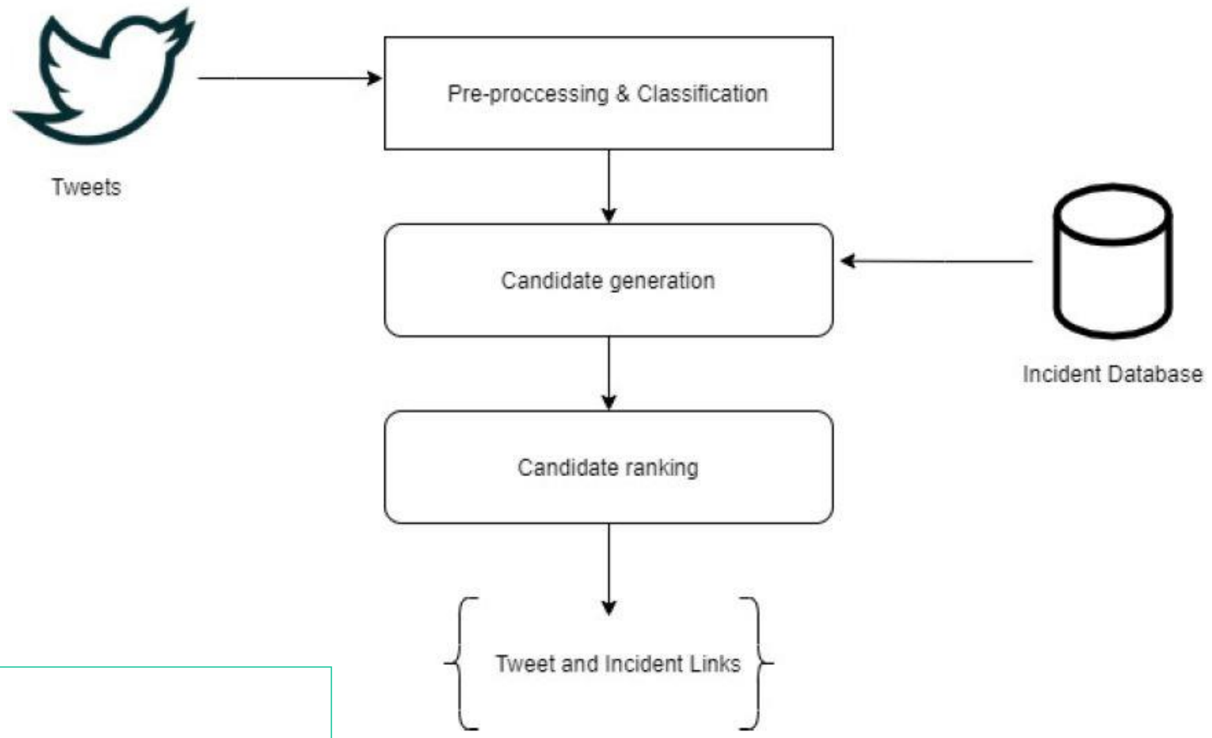
Proposed method

Incident Linking Framework(ILF)

ILF contains three different steps:

- Pre-processing and classification for tweets
- Candidate generation
- Candidate ranking

Incident Linking Framework(ILF)



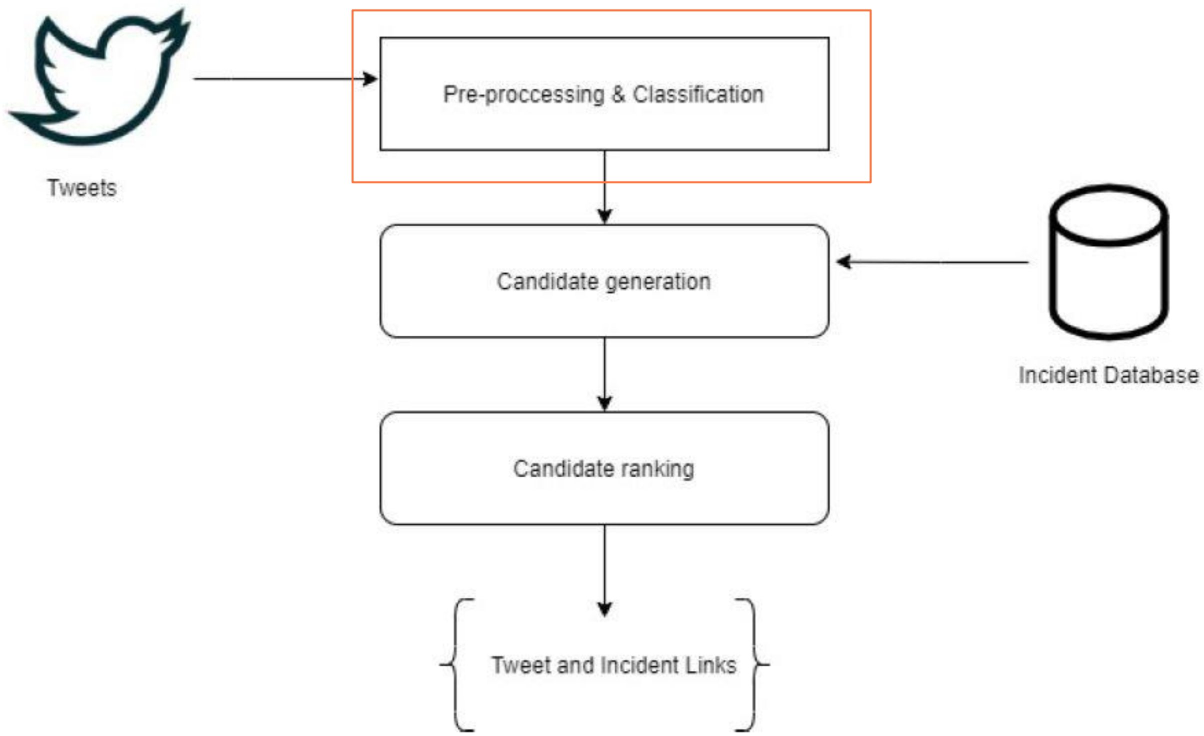
ILF contains three different steps:

- Pre-processing and classification for tweets
- Candidate generation
- Candidate ranking

Pre-processing and classification for tweets

Remove noise from the tweets

- Normalize piece of text
- Tweets that's not linkable

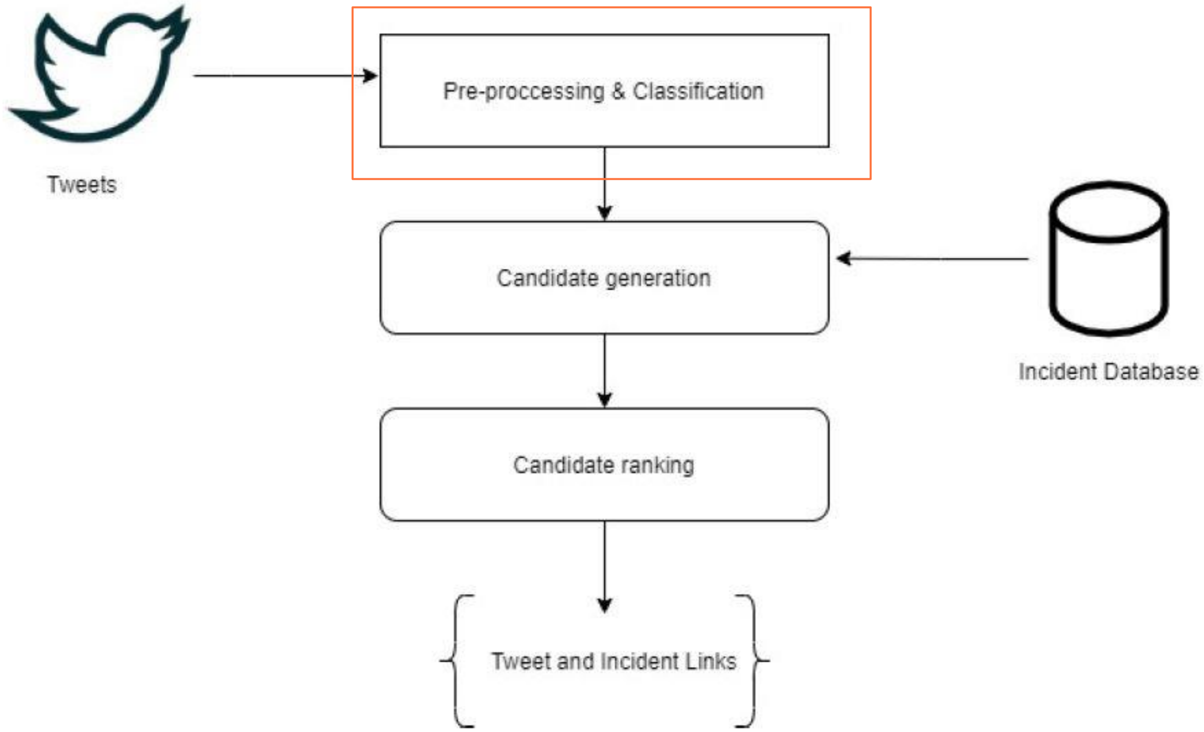


Pre-processing and classification for tweets

Remove noise from the tweets

- Normalize piece of text
- Tweets that's not linkable

- Pre-processing
    - URL's , Hashtags , Emoji's , Smileys
    - Convert text into numbers

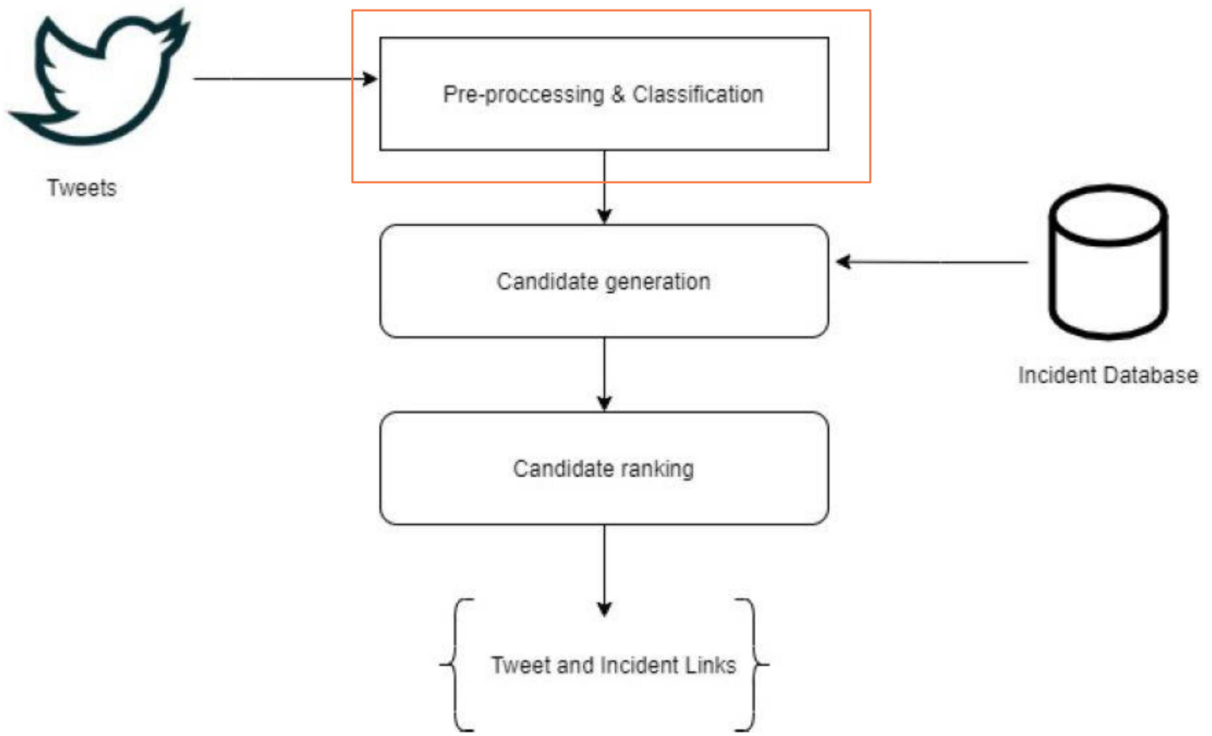


Pre-processing and classification for tweets

Remove Noise from the tweets

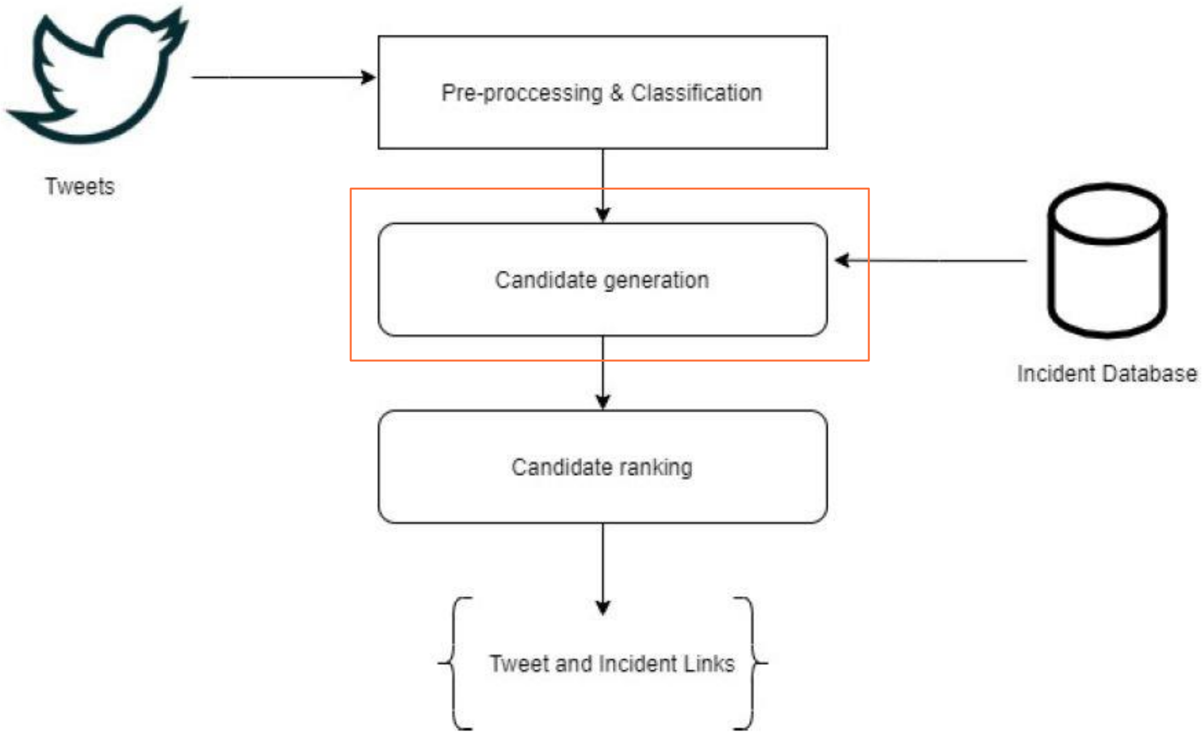
- Normalize piece of text
- Tweets that's not linkable

- **Pre-processing**
  - URL's , Hashtags , Emoji's , Smileys
  - Convert text into numbers
- **Classification**
  - Filter disaster related tweets
  - State-of-the-art pre-trained models



Candidate generation

- Input
  - Normalized tweets
  - Incident database
- Output
  - Candidate sets



Candidate generation

- Entities extraction using NER
- Generate the candidates based on the similarity between tweet entity mentions and Incidents entities
- Candidate generation divided into four steps

Candidate generation

- Entities extraction using NER
- Generate the candidates based on the similarity between tweets and Incidents to entities
- Candidate generation divided into four steps
  - Location-based candidates
  - Disaster type-based candidates
  - Impact-based candidates
  - Time-based candidates

Location- Based candidates

$$C_L = \{(t_j, i_k) \mid \forall\, j \in \{1, ..., n_t\}, k \in \{1, ..., n_i\} \;\; : SimF_L(t_j^L, i_k^L) \wedge \Delta T_{(t_j, i_k)} < \tau\}$$

where:

C_L = Location-based candidates

t = Tweet

i = Incident database entry

$SimF_L$ = Similarity function for location

t_j^L = Location mention for tweet j

i_k^L = Incident location entity

j, k = index values

$\Delta T_{(t_j, i_k)}$ = Difference between incident entry time and tweet time in hours

τ = time threshold (based on disaster type)

Location- Based candidates


```
{
  "incident_id": "47108fe1-5c04-472c-b534-75a51b747489",
  "type": "landslide",
  "start_time": "2011-12-08T08:00:00.000Z",
  "end_time": "NaN",
  "location": "Colombia ; Bosa  ;  Bogota",
  "lat": "4.6176",
  "lon": "-74.1899",
  "source_database_id": "2",
  "properties": {
    "id": "4,089",
    "landslide_": "Mudslide",
    "trigger": "Downpour",
    "storm_name": "nan",
    "fatalities": "6",
    "injuries": "0",
    "source_nam": "nan",
    "source_lin": "http://cnsnews.com/news/article/mudslide-collapses-bus-colombia-6-dead",
    "location_a": "Known_within_1_km",
    "landslide1": "Medium",
    "photos_lin": "nan",
    "cat_src": "glc",
    "countrynam": "Colombia",
    "near": "Soacha",
    "distance": "5.1765",
    "adminname1": "Cundinamarca",
    "adminname2": "nan",
    "population": "313,945",
    "countrycod": "nan",
    "continentc": "SA",
    "key_": "CO",
    "version": "1",
    "user_id": "1",
    "tstamp": "Tue Apr 01 2014 00:00:00 GMT+0000 (UTC)",
    "changeset_": "1"
  }
}
```


Disaster type-Based candidates

$$C_D = \{(t_j, i_k) \mid \forall\ j \in \{1, ..., n_t\}, k \in \{1, ..., n_i\}\ : SimF_D(t_j^D, i_k^D) \wedge \Delta T_{(t_j, i_k)} < \tau\}$$

where:

C_D = Disaster type-based candidates

t = Tweet

i = Incident database entry

$SimF_D$ = Similarity function for disaster type

t_j^D = Disaster type mention for tweet j

i_k^D = Incident disaster type entity

j, k = index values

$\Delta T_{(t_j, i_k)}$ = Difference between incident entry time and tweet time in hours

τ = time threshold (based on disaster type)

Disaster type-Based candidates


```
{
  "incident_id": "47108fe1-5c04-472c-b534-75a51b747489",
  "type": "landslide",
  "start_time": "2011-12-08T08:00:00.000Z",
  "end_time": "NaN",
  "location": "Colombia ; Bosa  ;  Bogota",
  "lat": "4.6176",
  "lon": "-74.1899",
  "source_database_id": "2",
  "properties": {
    "id": "4,089",
    "landslide_": "Mudslide",
    "trigger": "Downpour",
    "storm_name": "nan",
    "fatalities": "6",
    "injuries": "0",
    "source_nam": "nan",
    "source_lin": "http://cnsnews.com/news/article/mudslide-collapses-bus-colombia-6-dead",
    "location_a": "Known_within_1_km",
    "landslide1": "Medium",
    "photos_lin": "nan",
    "cat_src": "glc",
    "countrynam": "Colombia",
    "near": "Soacha",
    "distance": "5.1765",
    "adminname1": "Cundinamarca",
    "adminname2": "nan",
    "population": "313,945",
    "countrycod": "nan",
    "continentc": "SA",
    "key_": "CO",
    "version": "1",
    "user_id": "1",
    "tstamp": "Tue Apr 01 2014 00:00:00 GMT+0000 (UTC)",
    "changeset_": "1"
  }
}
```


Impact- Based candidates

$$C_I = \{(t_j, i_k) \mid \forall\, j \in \{1, ..., n_t\}, k \in \{1, ..., n_i\} \;\; : SimF_I(t_j^I, i_k^I) \wedge \Delta T_{(t_j, i_k)} < \tau\}$$

where:

C_I = Impact-based candidates

t = Tweet

i = Incident database entry

$SimF_I$ = Similarity function for impact

t_j^I = Impact mention for tweet j

i_k^I = Impact disaster type entity

j, k = index values

$\Delta T_{(t_j, i_k)}$ = Difference between incident entry time and tweet time in hours

τ = time threshold (based on disaster type)

Impact- Based candidates


```
{
  "incident_id": "47108fe1-5c04-472c-b534-75a51b747489",
  "type": "landslide",
  "start_time": "2011-12-08T08:00:00.000Z",
  "end_time": "NaN",
  "location": "Colombia ; Bosa  ;  Bogota",
  "lat": "4.6176",
  "lon": "-74.1899",
  "source_database_id": "2",
  "properties": {
    "id": "4,089",
    "landslide_": "Mudslide",
    "trigger": "Downpour",
    "storm_name": "nan",
    "fatalities": "6",
    "injuries": "0",
    "source_nam": "nan",
    "source_lin": "http://cnsnews.com/news/article/mudslide-collapses-bus-colombia-6-dead",
    "location_a": "Known_within_1_km",
    "landslide1": "Medium",
    "photos_lin": "nan",
    "cat_src": "glc",
    "countrynam": "Colombia",
    "near": "Soacha",
    "distance": "5.1765",
    "adminname1": "Cundinamarca",
    "adminname2": "nan",
    "population": "313,945",
    "countrycod": "nan",
    "continentc": "SA",
    "key_": "CO",
    "version": "1",
    "user_id": "1",
    "tstamp": "Tue Apr 01 2014 00:00:00 GMT+0000 (UTC)",
    "changeset_": "1"
  }
}
```


Time- Based candidates

$$C_T = \{(t_j, i_k) \mid \forall\, j = (1, ..., n_t), k = (1, ..., n_i) : \Delta T_{(t_j,i_k)} < \tau = True\}$$

where:

C_T = Time-based candidates

t = Tweet

i = Incident database entry

$\Delta T_{(t_j,i_k)}$ = Difference between incident entry time and tweet time (no of hours)

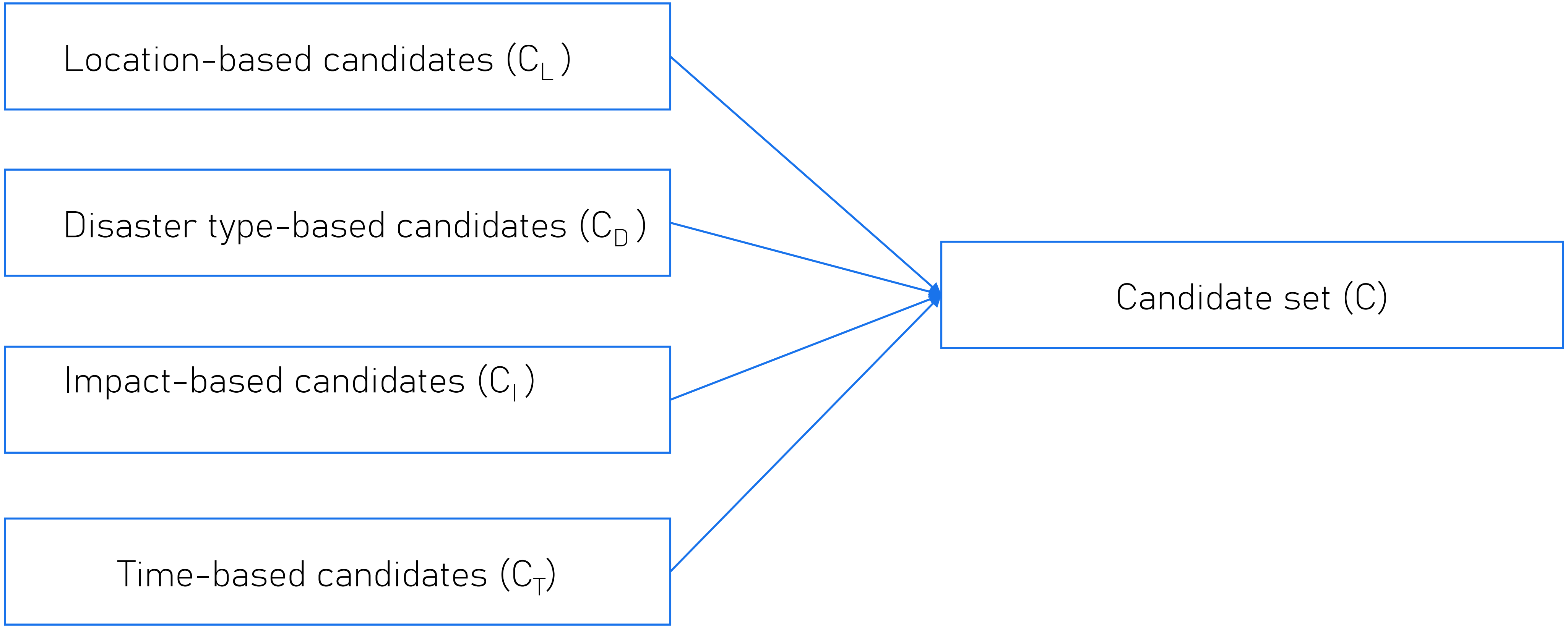
i, j = index values

τ = time threshold (based on disaster type)

Time- Based candidates

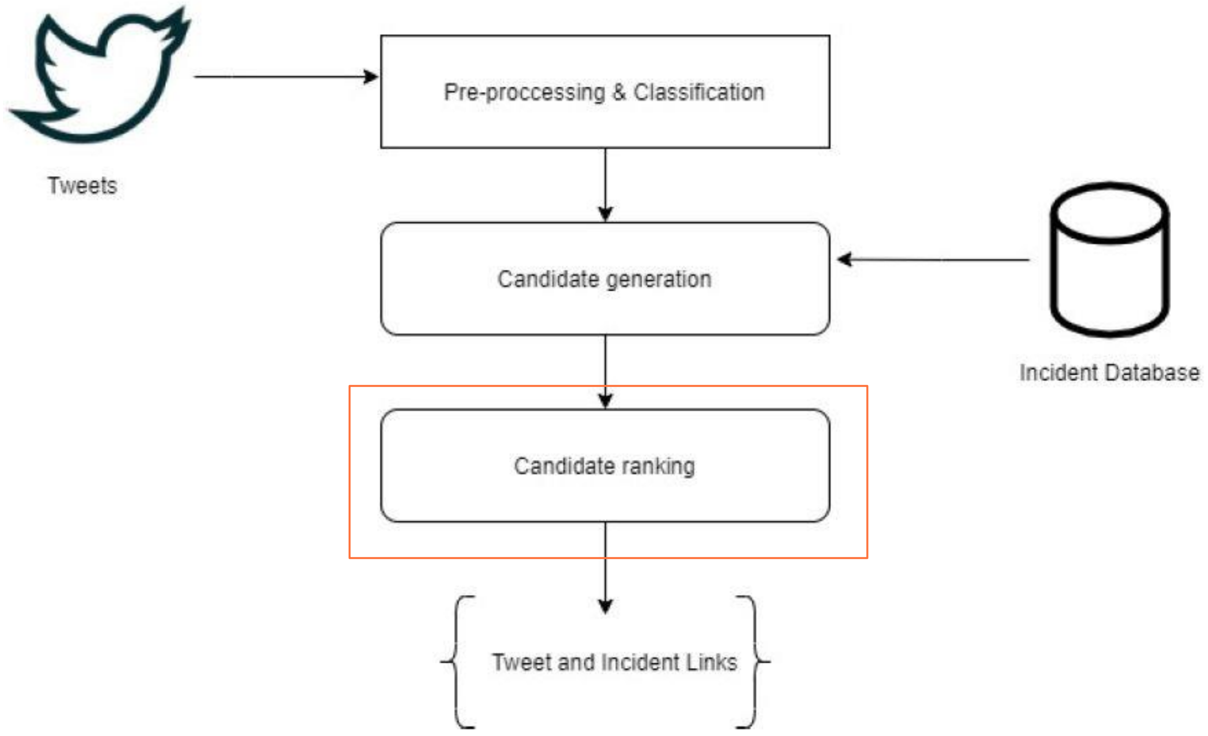

```
{
  "incident_id": "47108fe1-5c04-472c-b534-75a51b747489",
  "type": "landslide",
  "start_time": "2011-12-08T08:00:00.000Z",
  "end_time": "NaN",
  "location": "Colombia ; Bosa  ;  Bogota",
  "lat": "4.6176",
  "lon": "-74.1899",
  "source_database_id": "2",
  "properties": {
    "id": "4,089",
    "landslide_": "Mudslide",
    "trigger": "Downpour",
    "storm_name": "nan",
    "fatalities": "6",
    "injuries": "0",
    "source_nam": "nan",
    "source_lin": "http://cnsnews.com/news/article/mudslide-collapses-bus-colombia-6-dead",
    "location_a": "Known_within_1_km",
    "landslide1": "Medium",
    "photos_lin": "nan",
    "cat_src": "glc",
    "countrynam": "Colombia",
    "near": "Soacha",
    "distance": "5.1765",
    "adminname1": "Cundinamarca",
    "adminname2": "nan",
    "population": "313,945",
    "countrycod": "nan",
    "continentc": "SA",
    "key_": "CO",
    "version": "1",
    "user_id": "1",
    "tstamp": "Tue Apr 01 2014 00:00:00 GMT+0000 (UTC)",
    "changeset_": "1"
  }
}
```


Union of candidates



Candidate ranking

- Input
  - Candidates
- Output
  - Tweet and Incident links



Candidate ranking

- Implemented a scoring metric to assign similarity score for each candidate in the candidate list
- Identified Top score candidate to establish the link
- Four different scoring functions are implemented in candidate ranking

Candidate ranking

- Implemented a scoring metric to assign similarity score for each candidate in the candidate list
- Identified Top score candidate to establish the link
- Four different scoring functions are implemented in candidate ranking
  - Location score
  - Disaster type score
  - Impact score
  - Time score

Location score

$$C_{LScore} = SimS_L(t, i), \forall(t, i) \in C_L$$

where:

C_{LScore} = Location similarity score

t, i = Tweet , Incident entry

C_L = Location-based candidate set

$SimS_L$ = Similarity score function for location

Location score example

Tweet Location	Incident location	Score
Colombia bosa	Colombia	0.25
	Colombia; Bosa	0.25+0.25=0.5

Disaster type score

$$C_{DScore} = SimS_D(t, i), \forall (t, i) \in C_D$$

where:

C_{DScore} = Disaster type similarity score

t, i = Tweet , Incident entry

C_D = Disaster type-based candidate set

$SimS_D$ = Similarity score function for disaster type

Disaster type score example

Tweet disaster type	Incident disaster type	Score
landslide	mudslide	0.4
	landslide	0.6

Impact score

$$C_{IScore} = SimS_I(t, i), \forall(t, i) \in C_I$$

where:

C_{IScore} = Impact similarity score

t, i = Tweet , Incident entry

C_I = Impact-based candidate set

$SimS_I$ = Similarity score function for impact

Impact score example

Tweet	Incident no of deaths	Score
Mudslide collapses on bus in Colombia, 6 dead	10	0.2
	4	0.3
	6	0.5

Time score

$$C_{TScore} = SimS_T(t, i), \forall (t, i) \in C_T$$

where:

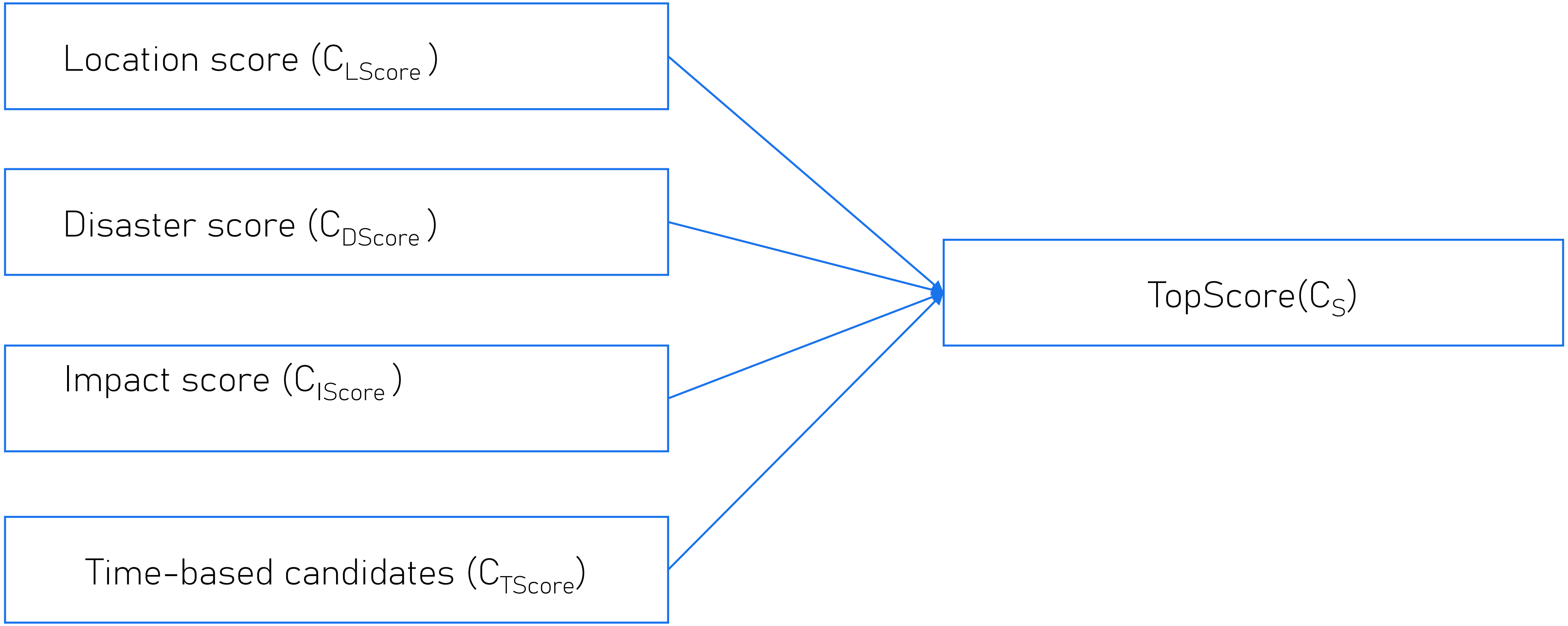
C_{TScore} = Time similarity score

t, i = Tweet , Incident entry

C_T = Time-based candidate set

$SimS_T$ = Similarity score function for time

Generate top score



Experiments

Evaluation Metrics

- Precision , Recall , F1-Score and MRR (Mean Reciprocal Rank)

Evaluation Metrics

- Precision , Recall , F1-Score and MRR (Mean Reciprocal Rank)
- Intrinsic metrics
  - Evaluate each module individually without the side effects from others
  - Candidate generation, Candidate ranking

Evaluation Metrics

- Precision , Recall , F1-Score and MRR (Mean Reciprocal Rank)
- Intrinsic metrics
    - Evaluate each module individually without the side effects from others
    - Candidate generation (recall), Candidate ranking (MRR)
- Extrinsic metrics
    - Measure the whole application with cascading errors
    - Candidate ranking (MRR)

Experimental setup

- Two experiments
  - ILF Method – 1
  - ILF Method – 2

Experimental setup

- Two experiments
  - ILF Method – 1
  - ILF Method – 2

Candidate set	ILF Method –1	ILF Method –2
Location candidates	✓	✓
Disaster type candidates	✓	✓
Impact candidates	✓	✓
Time candidates		✓

Experimental setup

- Two experiments
  - ILF Method – 1
  - ILF Method – 2
- Aim of these experiments is to check the importance of time constraints

Candidate set	ILF Method -1	ILF Method -2
Location candidates	✓	✓
Disaster type candidates	✓	✓
Impact candidates	✓	✓
Time candidates		✓

Experimental setup

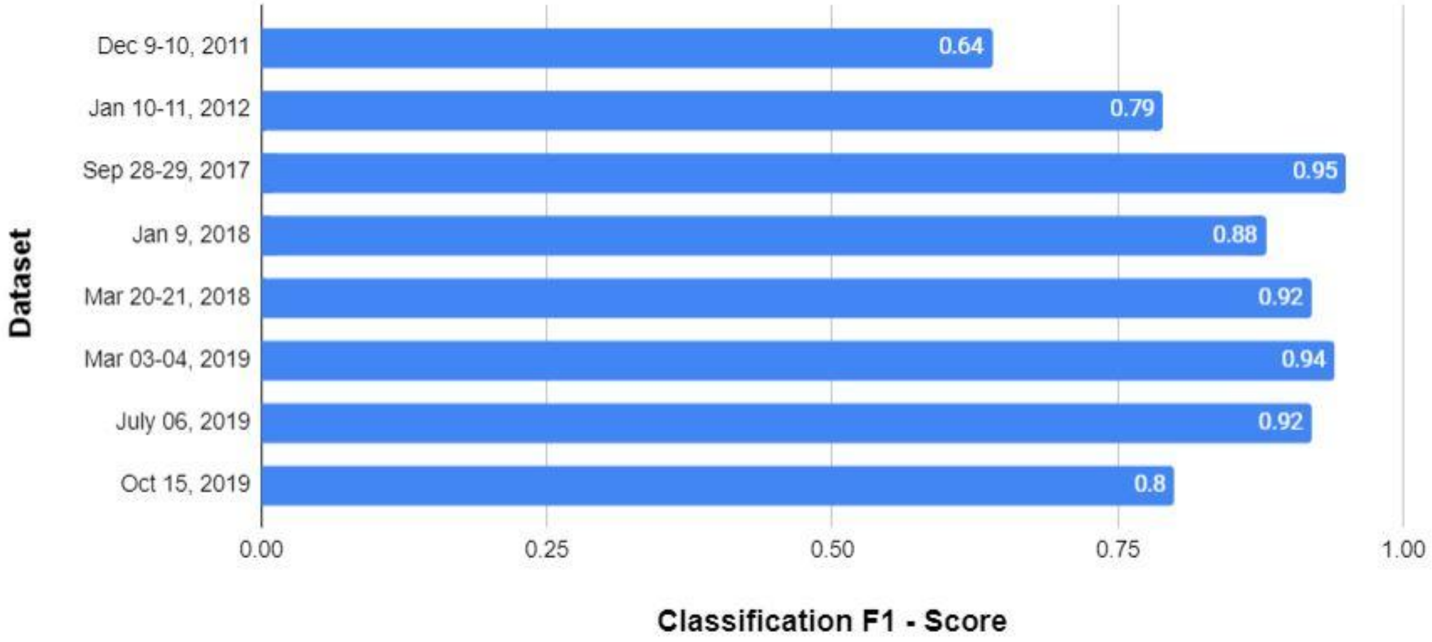
- Two experiments
  - ILF Method – 1
  - ILF Method – 2
- Aim of these experiments is to check the importance of time constraints
- Classification and ranking module will be the same for both methods

Candidate set	ILF Method -1	ILF Method -2
Location candidates	✓	✓
Disaster type candidates	✓	✓
Impact candidates	✓	✓
Time candidates		✓

Results and Discussion

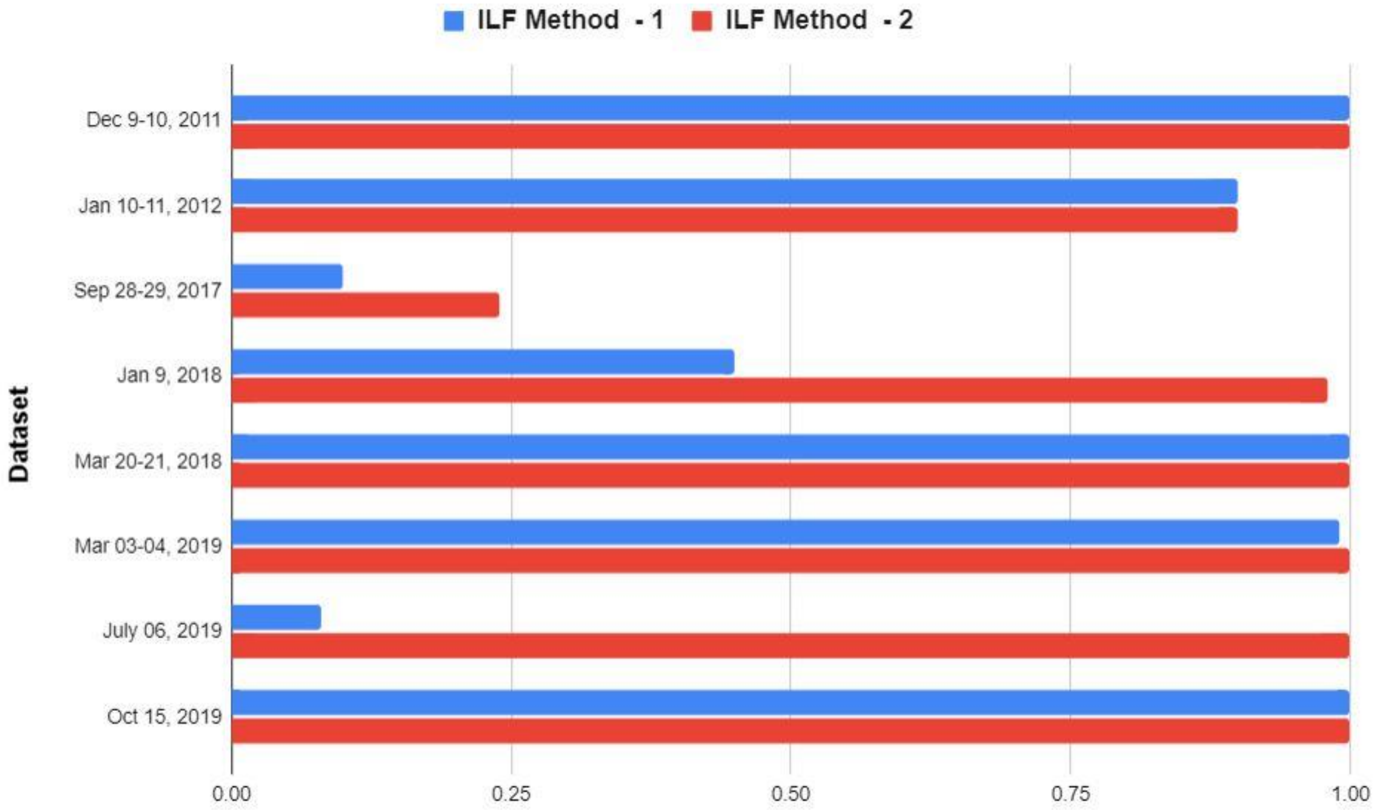
Classification (F1- Score)

- Avg. Score 0.86
- Best Score 0.95



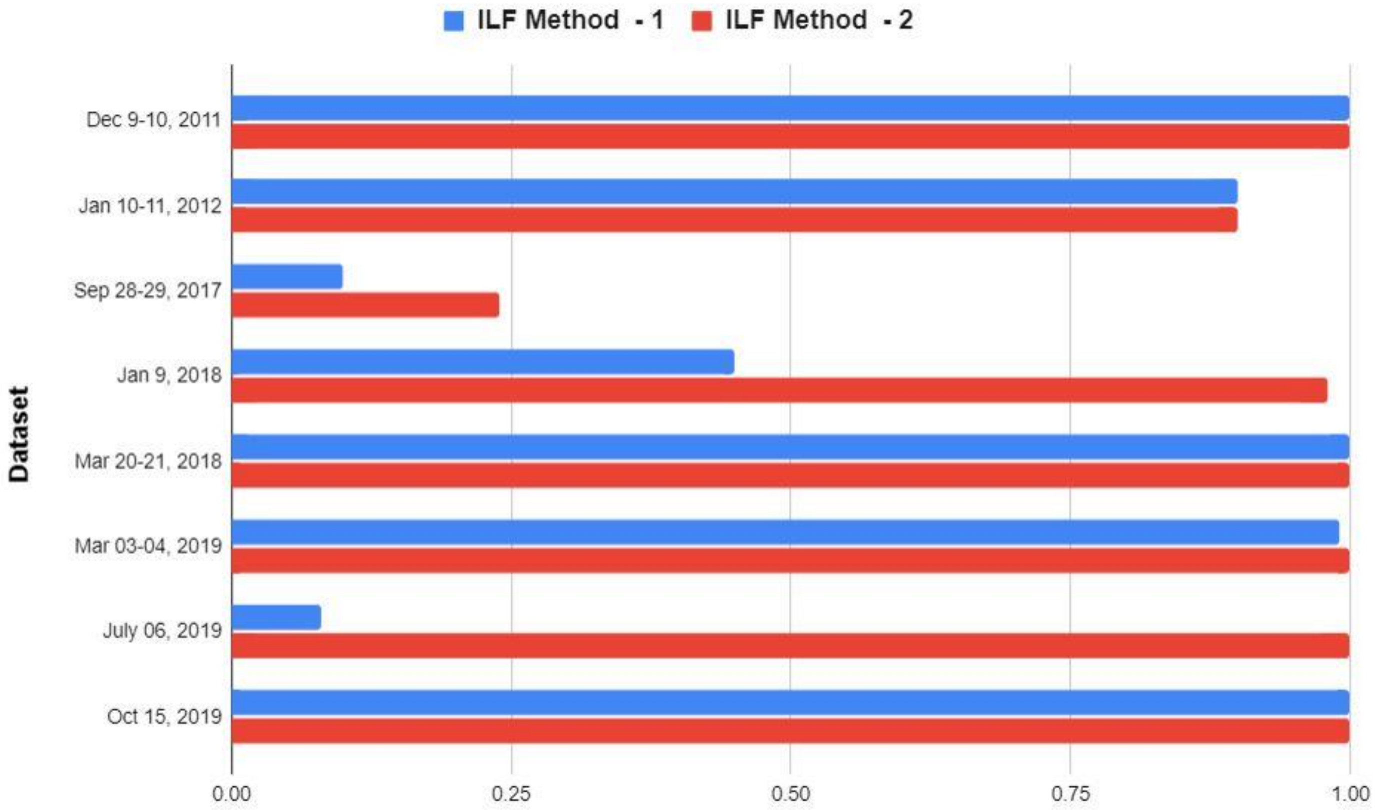
Candidate generation (recall – Intrinsic)

- Avg. recall for ILF Method – 1 & 2 is 0.69 , 0.89 respectively



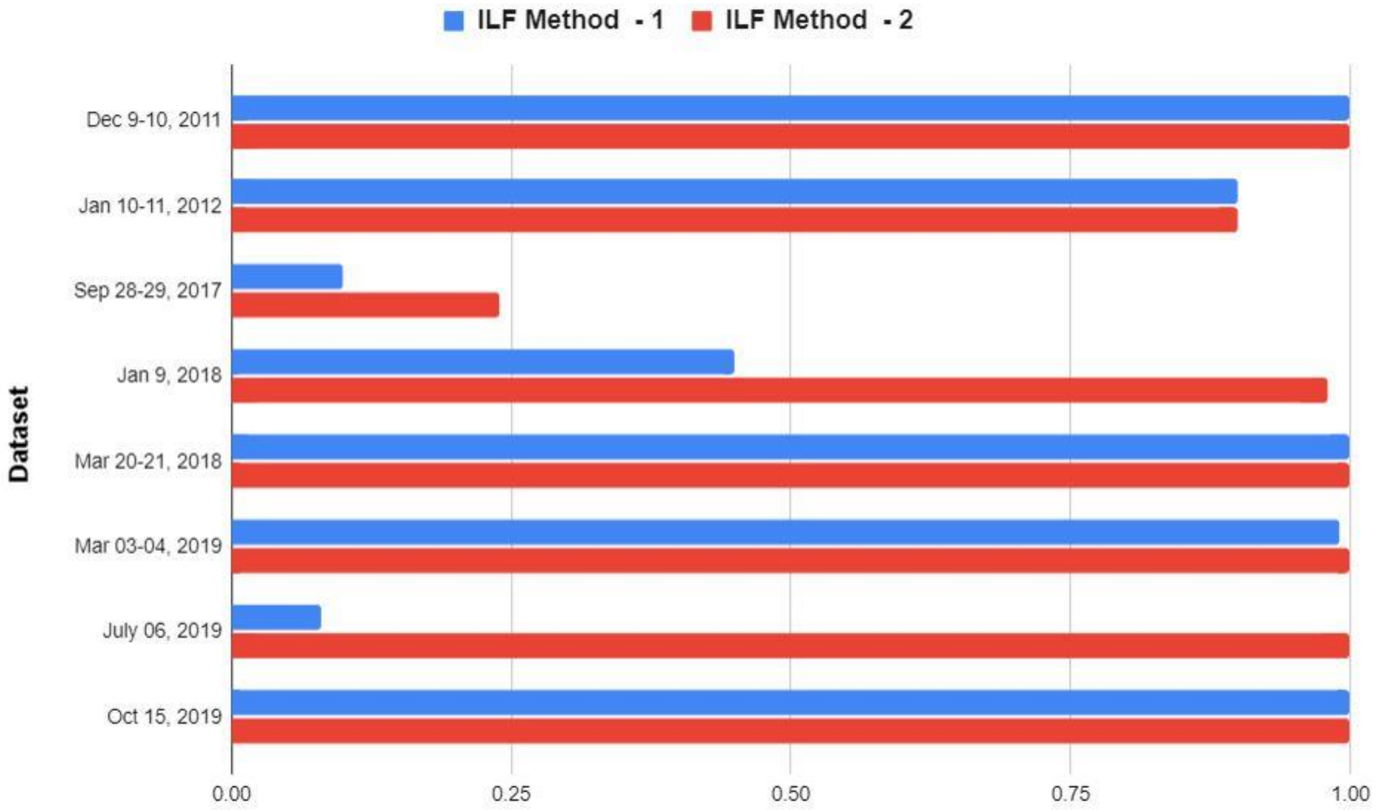
Candidate generation (recall – Intrinsic)

- Avg. recall for ILF Method – 1 & 2 is 0.69 , 0.89 respectively
- ILF Method – 2 is shown promising results than ILF Method – 1



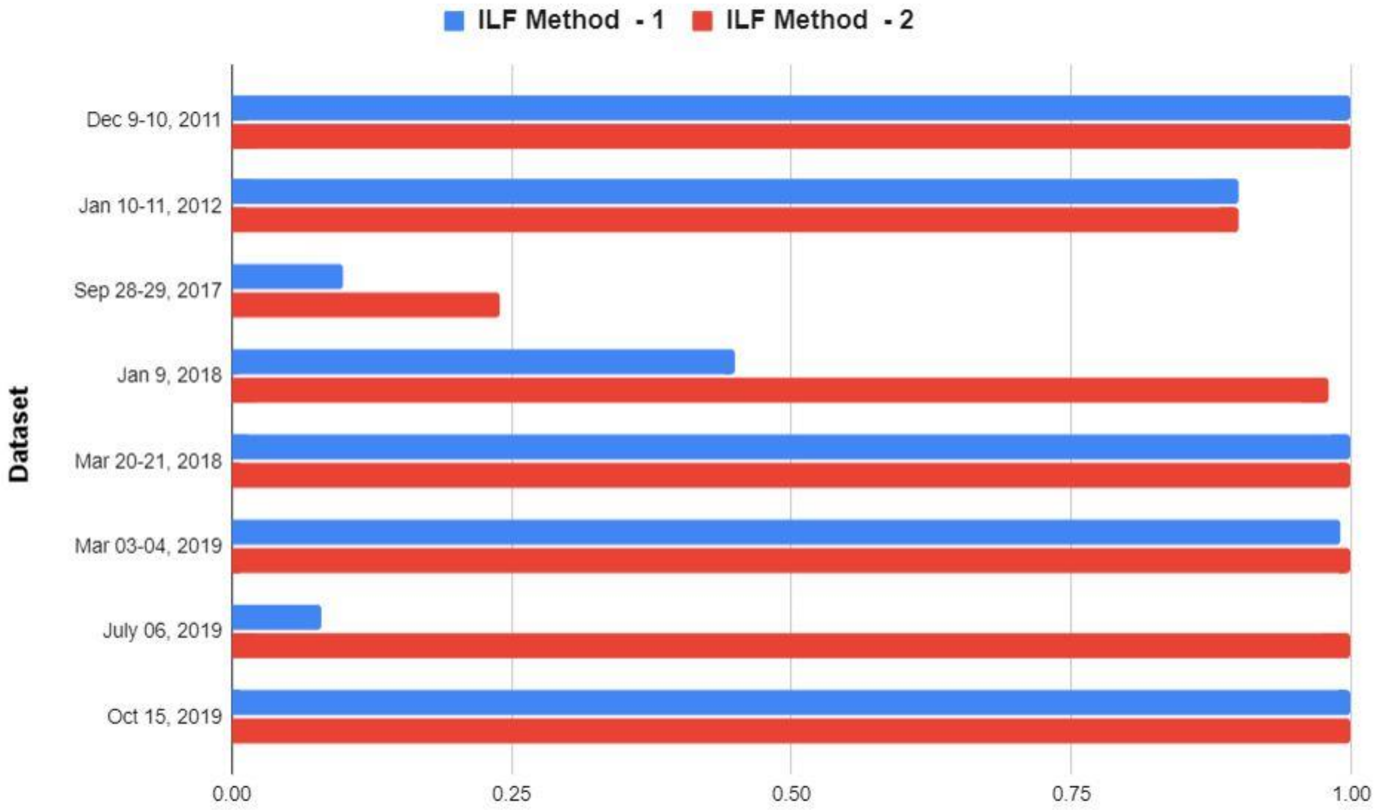
Candidate generation (recall – Intrinsic)

- Avg. recall for ILF Method – 1 & 2 is 0.69 , 0.89 respectively
- ILF Method – 2 is shown promising results than ILF Method – 1
- More no of candidates generated for ILF Method – 2



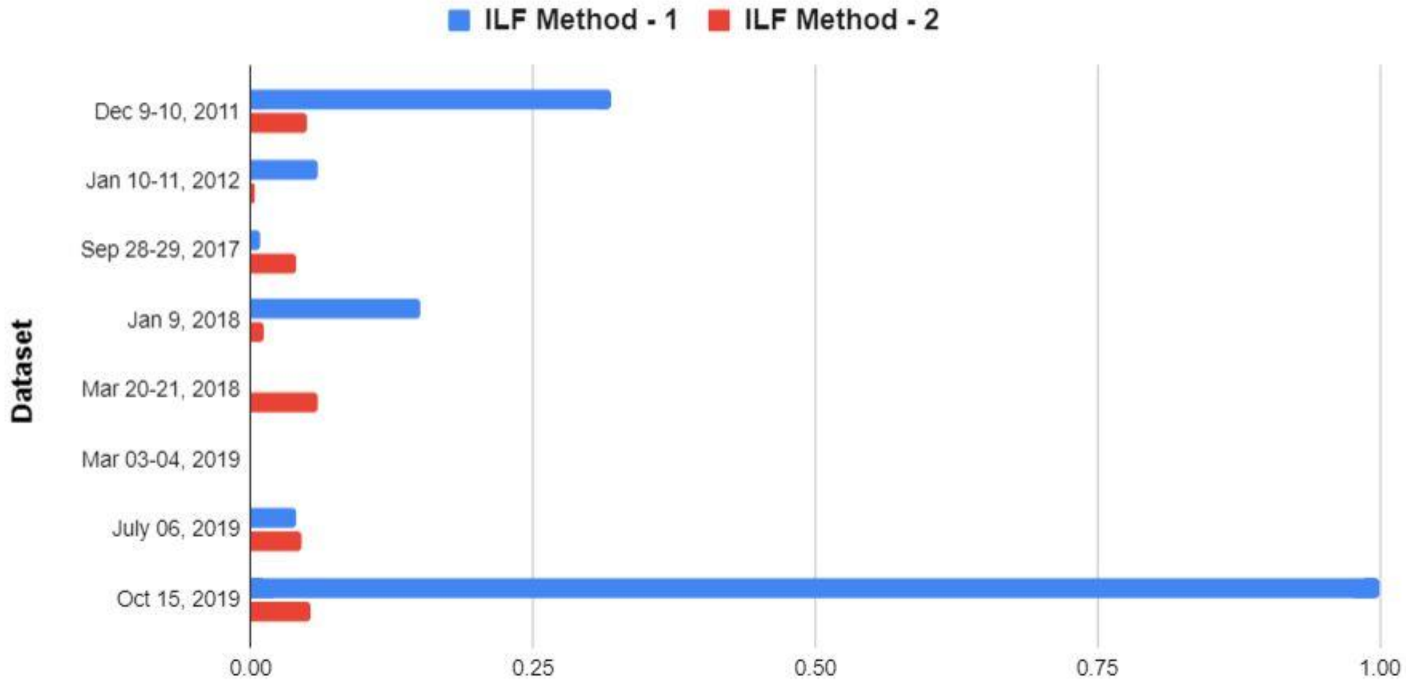
Candidate generation (recall – Intrinsic)

- Avg. recall for ILF Method – 1 & 2 is 0.69 , 0.89 respectively
- ILF Method – 2 is shown promising results than ILF Method – 1
- More no of candidates generated for ILF Method – 2
- Avg. no candidates for ILF Method – 1 & 2 are 95 , 418 respectively



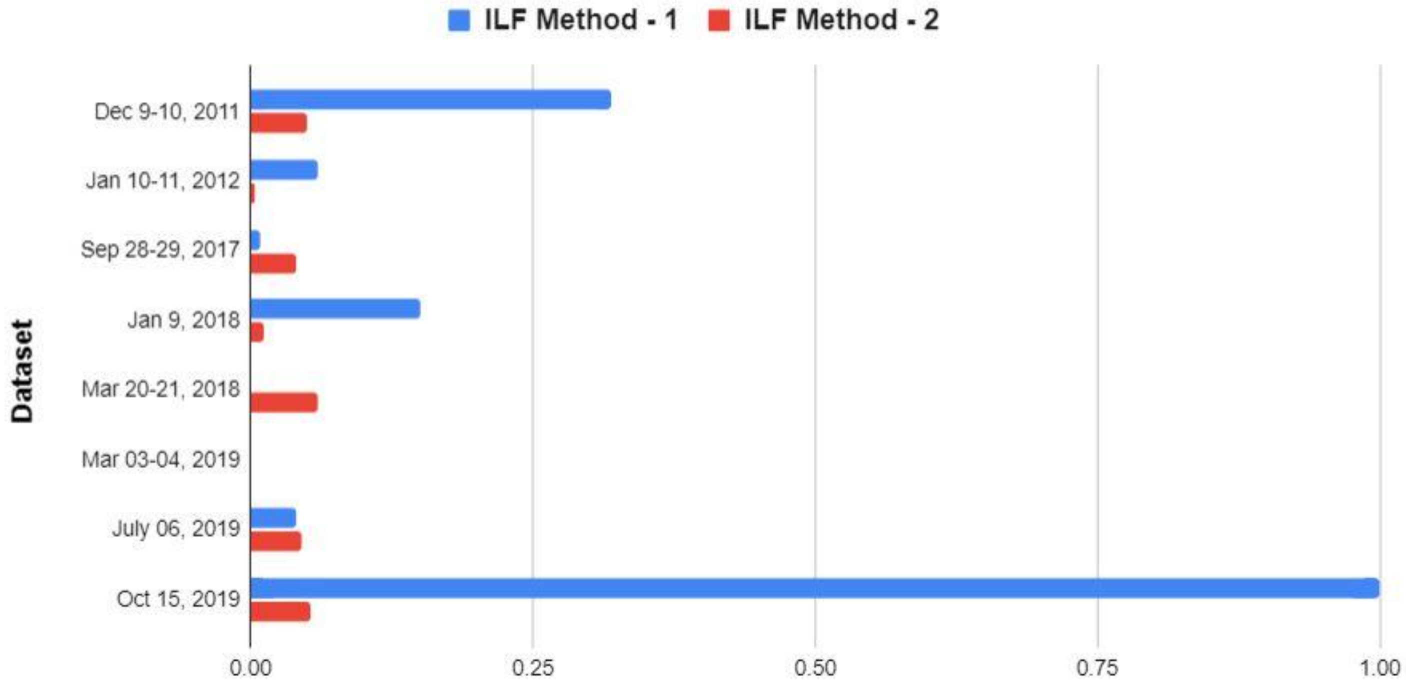
Candidate ranking (MRR – Intrinsic)

- Avg. MRR for ILF Method – 1 & 2 is 0.1972 , 0.0329 respectively



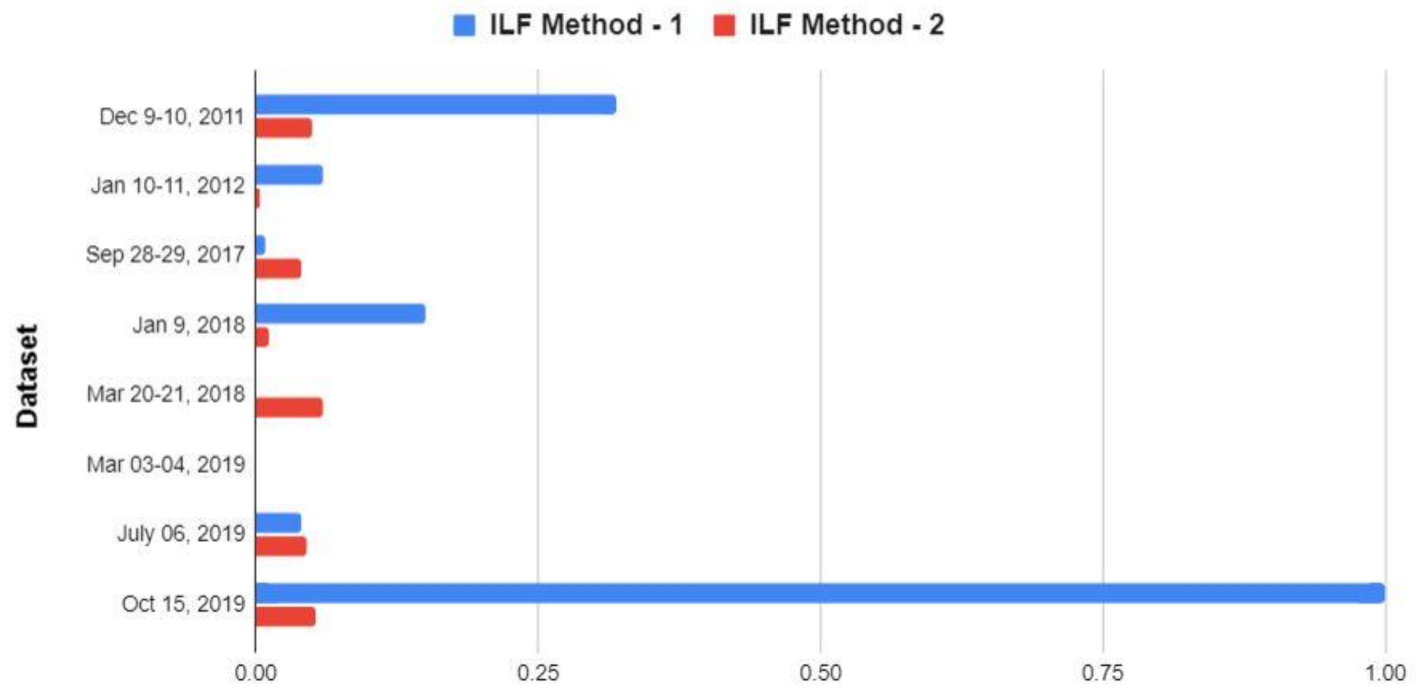
Candidate ranking (MRR – Intrinsic)

- Avg. MRR for ILF Method – 1 & 2 is 0.1972 , 0.0329 respectively
- ILF Method – 1 is shown promising results than ILF Method – 2



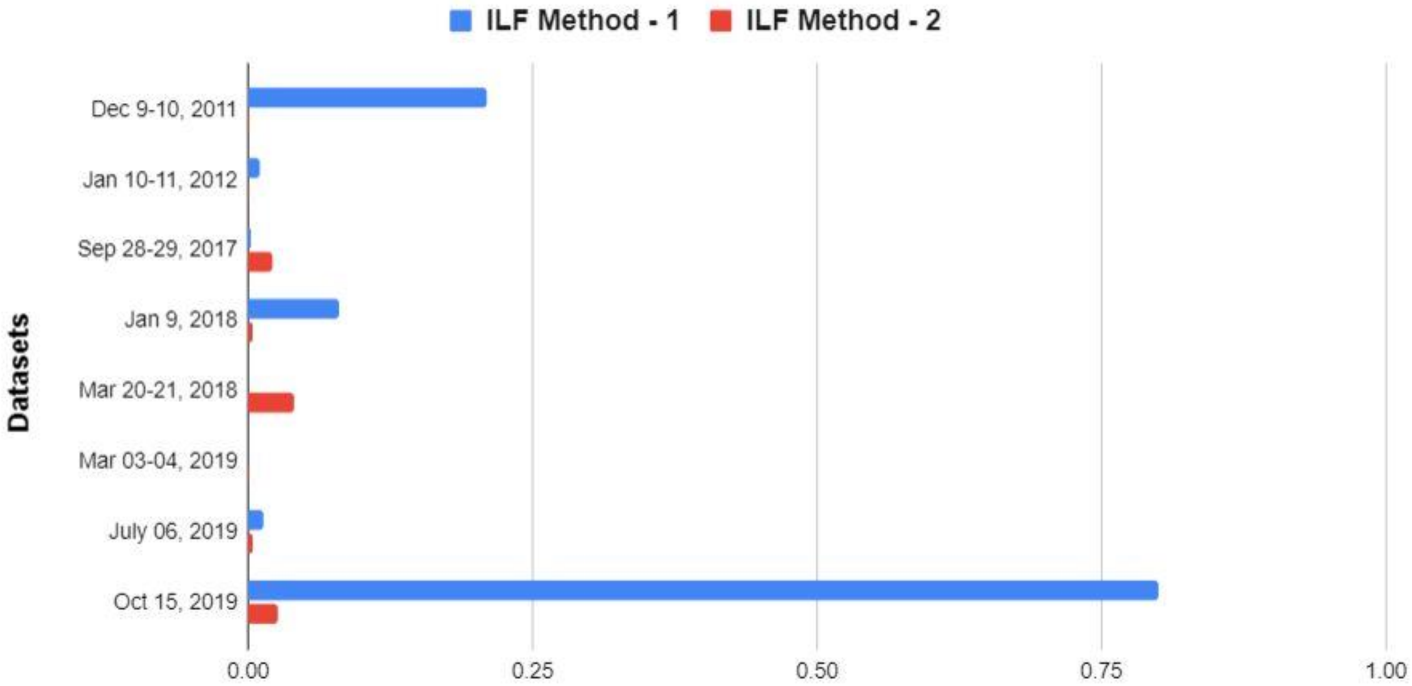
Candidate ranking (MRR – Intrinsic)

- Avg. MRR for ILF Method – 1 & 2 is 0.1972 , 0.0329 respectively
- ILF Method – 1 is shown promising results than ILF Method – 2
- ILF Method – 1 shown best results for small datasets



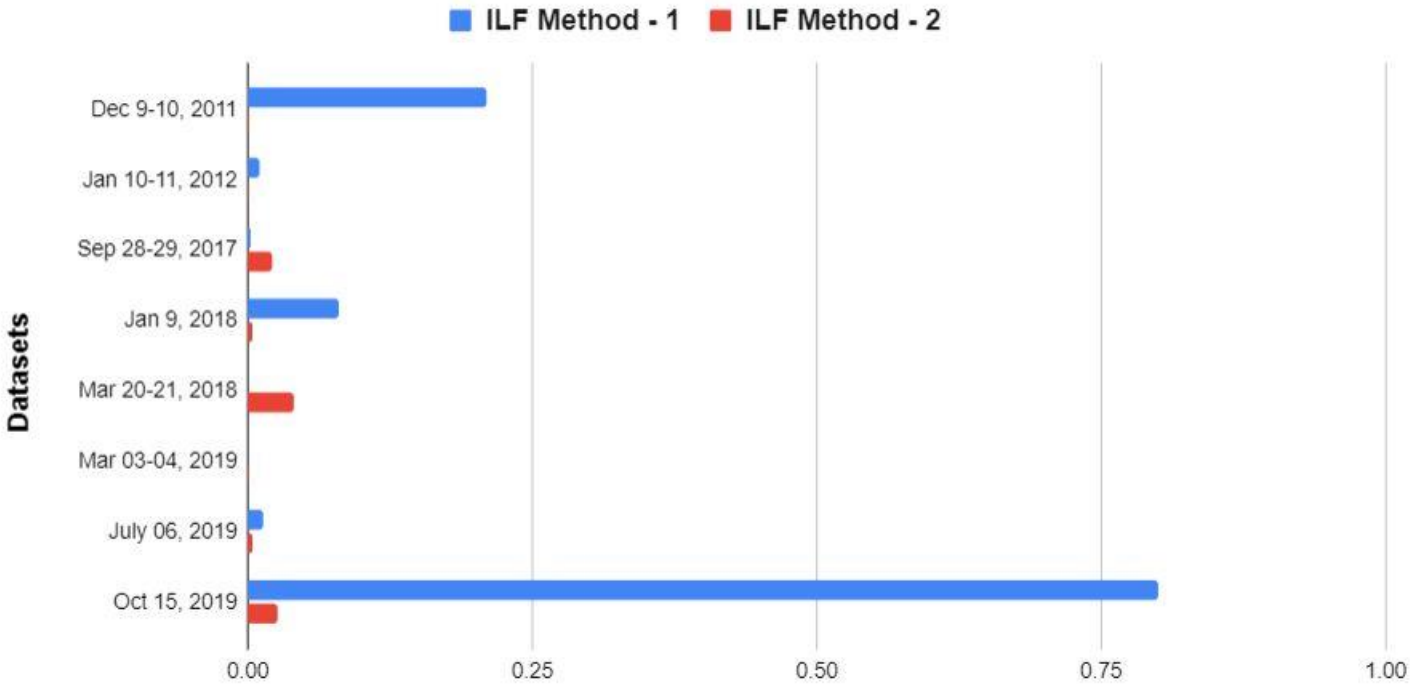
Candidate ranking (MRR – Extrinsic)

- Avg. MRR for ILF Method – 1 & 2 is 0.1395 , 0.0123 respectively



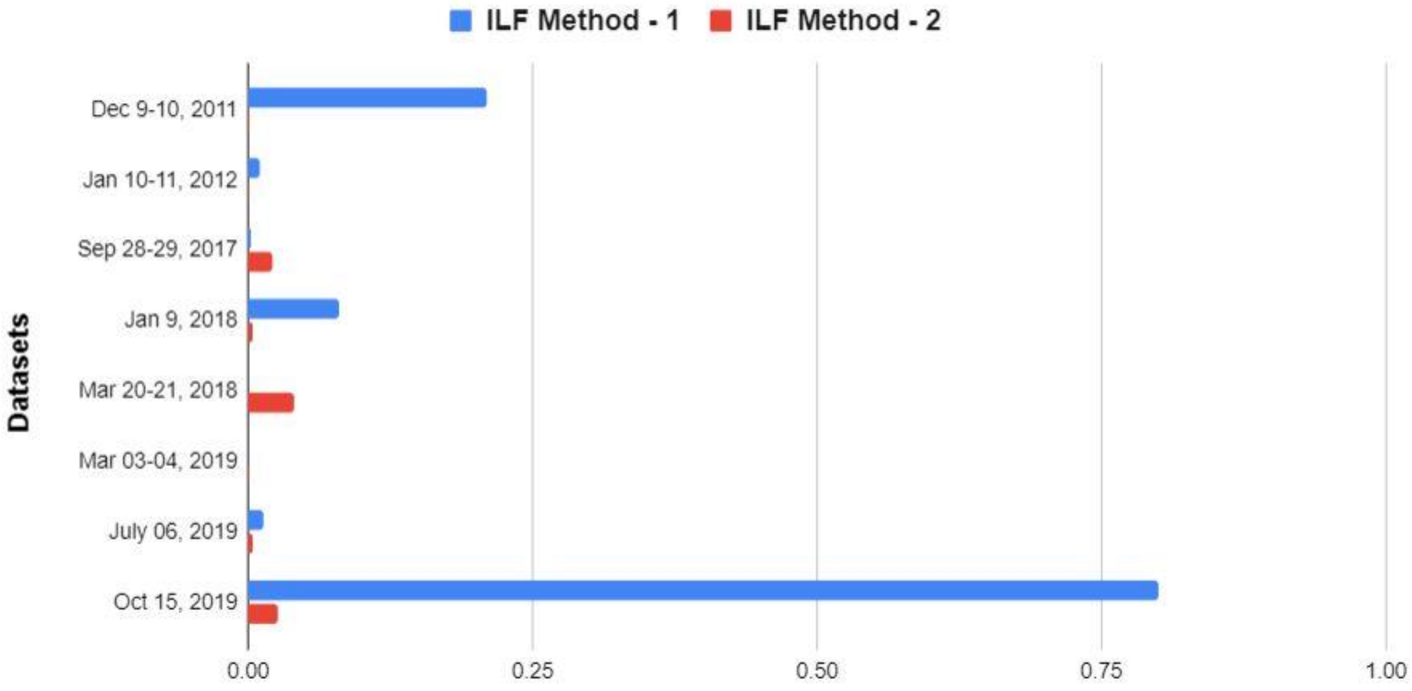
Candidate ranking (MRR – Extrinsic)

- Avg. MRR for ILF Method – 1 & 2 is 0.1395 , 0.0123 respectively
- ILF Method – 1 is shown promising results than ILF Method – 2



Candidate ranking (MRR – Extrinsic)

- Avg. MRR for ILF Method – 1 & 2 is 0.1395 , 0.0123 respectively
- ILF Method – 1 is shown promising results than ILF Method – 2.
- ILF Method – 1 shown best results for small datasets



Overview results of ILF Method – 1

- Overall performance of Dec 9-10 and Oct 15,2019 datasets shown good results
- Candidate ranking could not performed well on Mar 20-21,2018 and Mar 03-04 datasets

Datasets	Classification F1 Score (Intrinsic)	Candidate generation recall (Intrinsic)	Candidate generation count (Intrinsic)	Candidate ranking MRR (Intrinsic)	Candidate ranking MRR (Extrinsic)
Dec 9-10, 2011	0.64	1	61.62	0.3200	0.2100
Jan 10-11, 2012	0.79	0.9	47.32	0.0600	0.0100
Sep 28-29, 2017	0.95	0.1	129.03	0.0080	0.0030
Jan 9, 2018	0.88	0.45	94.56	0.1500	0.0800
Mar 20-21, 2018	0.92	1	158.63	0	0
Mar 03-04, 2019	0.94	0.99	128	0	0
July 06, 2019	0.92	0.08	108	0.0400	0.0135
Oct 15, 2019	0.8	1	52	1	0.8000

Overview results of ILF Method – 2

- Candidate generation performed well
- High candidate generation count
- Candidate ranking could not performed well

Datasets	Classification F1 Score (Intrinsic)	Candidate generation recall (Intrinsic)	Candidate generation count (Intrinsic)	Candidate ranking MRR (Intrinsic)	Candidate ranking MRR (Extrinsic)
Dec 9-10, 2011	0.64	1	977	0.0500	0.0010
Jan 10-11, 2012	0.79	0.90	462	0.0031	0.0015
Sep 28-29, 2017	0.95	0.24	557	0.0400	0.0215
Jan 9, 2018	0.88	0.98	918	0.0116	0.0044
Mar 20-21, 2018	0.92	1	145	0.0600	0.0400
Mar 03-04, 2019	0.94	1	128	0.0010	0.0004
July 06, 2019	0.92	1	108	0.0452	0.0034
Oct 15, 2019	0.8	1	52	0.0530	0.0265

Micro averages

- ILF Method – 1 performed well in candidate generation also system performance was good when compare to ILF Method – 2

Datasets	Classification F1 Score (Intrinsic)	Candidate generation recall (Intrinsic)	Candidate generation count (Intrinsic)	Candidate ranking MRR (Intrinsic)	Candidate ranking MRR (Extrinsic)
ILF Method - 1	0.84	0.69	95	0.1972	0.1395
ILF Method - 2	0.84	0.89	418	0.0329	0.0123

Research Questions Revisited

- RQ 1

  What are the possible features that we can extract from tweets that match with those of typical knowledge databases?

Research Questions Revisited

- RQ 1

  What are the possible features that we can extract from tweets that match with those of typical knowledge databases?

  We can extract Location, Disaster type , Impact and Time

Research Questions Revisited

- RQ 2

  How can we build a linking model that will link the each tweet to entries in the disaster database based on the features from RQ1?

Research Questions Revisited

- RQ 2

  How can we build a linking model that will link the each tweet to entries in the disaster database based on the features from RQ1?

  Incident Linking Framework implemented with Candidate generation and candidate ranking modules

Research Questions Revisited

- RQ 3

  How accurate this model to use for disaster linking?

Research Questions Revisited

- RQ 3

  How accurate this model to use for disaster linking?

  ILF is less accurate and need improvements in candidate generation and candidate ranking

Conclusion and Future Work

Conclusion and Future work

- Conclusion
  - Implemented Incident Linking Frame (ILF)
  - Two different NER's makes better recall for candidate generation
  - Low performance due to the heavy no of candidates generated by the system
  - Candidate ranking module needs to be improved

Conclusion and Future work

- Future work
  - Create missing entries in the database
  - Extend these system to other languages
  - Improve candidate ranking method using advanced ML (e.g. CNN)

THANK YOU