

# Fact\_Checker

## Group name : SmartBV2

Vineet Vasishta Eranki 6818911

Sandeep Varma Ganaraju 6787429

GitHub repo: [https://github.com/SmartBV2/Fact\\_Checker](https://github.com/SmartBV2/Fact_Checker)

Branch: Hits\_Algo

# Overview

- Earlier approaches
- Implemented approach
- Algorithm design
- Evaluation results
- Task Division

# Earlier approaches

## **Naïve:**

- Corpus creation: New York Times 2013
- Corpus analysis: Stanford NER, FOX

## **Graph DB**

- Inspiration from "Computational fact checking from knowledge networks" paper.
- Corpus analysis: Neo4J

# Implemented Approach

- Corpus creation: On the Fly using FOX and DBpedia
- Corpus analysis: Hits\_Algo
- Confidence value between 0 [False] and 1 [True]

# Algorithm : Hits\_Algo

- Reading of data from file  
FactID and Fact\_Statement
- Pass the Fact\_Statement as input to FOX.
- FOX uses NLP algorithms to extract RDF triples.
- Store the JSON format and identify the entities.(e.g.Person)
- Search in DBpedia by entity by considering the Infobox data of Wikipedia.
- Store the results.

- Parse the sentence and clean the text for stop words.
- Tokenization of words.
- Removal of searched entity from the tokens.
- Searching of tokens in the DBpedia Output.
- Count of number of matching words.
- Calculation of confidence value =  $\text{matched tokens} / \text{total tokens}$ .
- Writing of confidence value in TTL format.

- Text input:

Nobel Prize in Physiology or Medicine is David Baltimore's honour.

- Applying NER using FOX:

Nobel Prize in Physiology or Medicine is David Baltimore's honour.

- Search in DBpedia:

[http://dbpedia.org/data/David\\_Baltimore.json](http://dbpedia.org/data/David_Baltimore.json)

- Cleaning of stop words and Tokenization:

Nobel, Prize, Physiology, Medicine, David, Baltimore, honour

- After Removal of searched token:

Nobel, Prize, Physiology, Medicine, honour

- Finding matched words:

Nobel, Prize, Physiology, Medicine

- Matched words: 4
  - Total tokens: 5
- Calculation of confidence:

Matched words/ Total tokens = 0.8

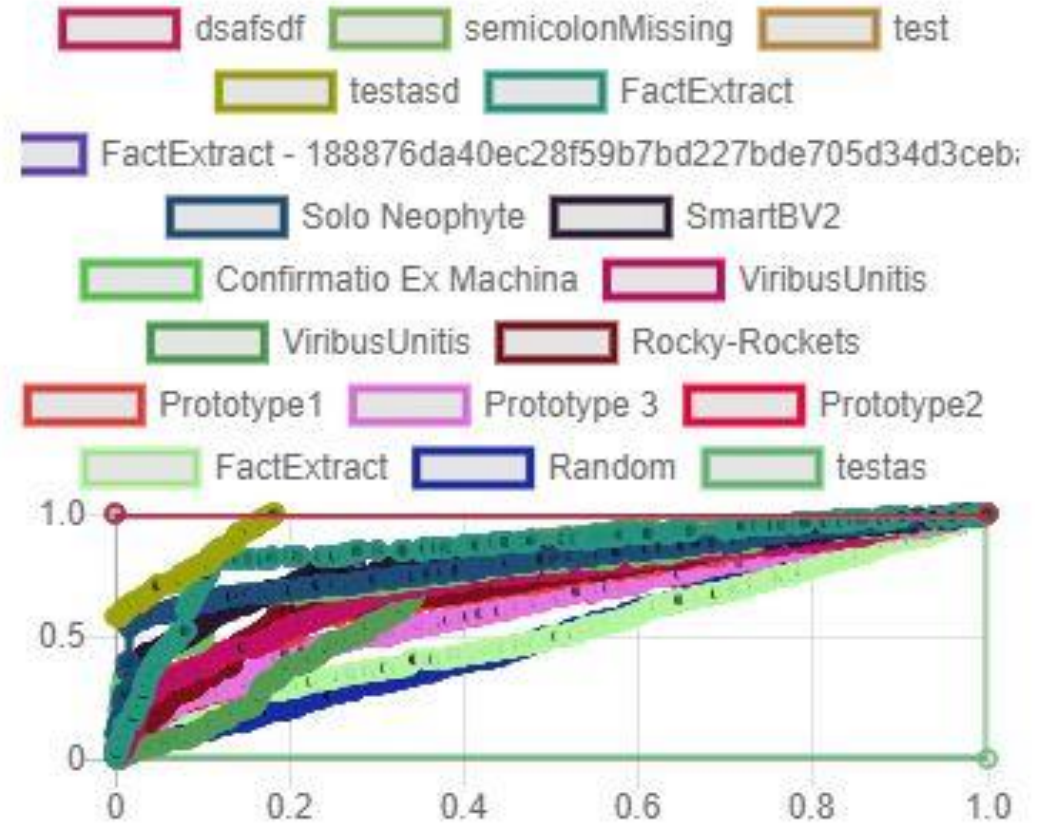
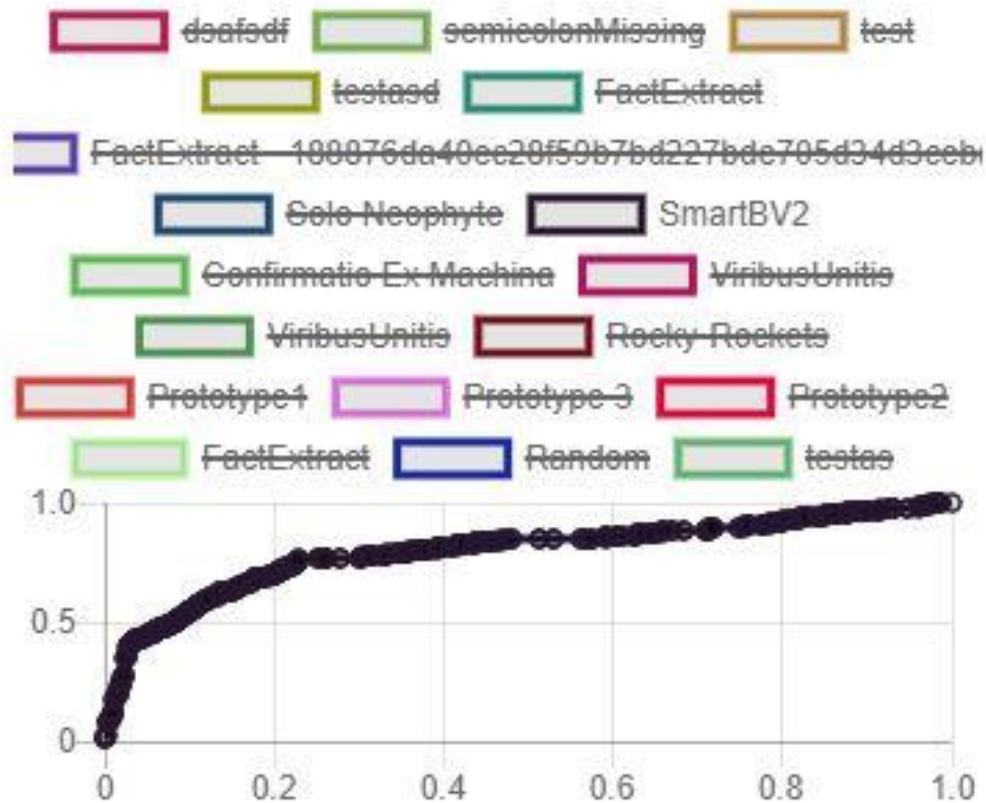
# Evaluation Results

Dataset: SNLP 2017 Train

AnnotatorName	Area Under Curve (AUC)
dsafsd	1.0
semicolonMissing	1.0
test	1.0
testasd	0.9605829150945845
FactExtract	0.8501560874089487
FactExtract - 188876da40ec28f59b7bd227bde705d34d3cebaf	0.8494741635814919
Solo Neophyte	0.8009112767548917
SmartBV2	0.7956234175945787
ViribusUnitis	0.7150077512794918
ViribusUnitis	0.7064612872551033
Rocky-Rockets	0.7024971625833821
Prototype1	0.6992078829450618
FactExtract	0.526315416905576
Random	0.5044372240745066
testas	0.0



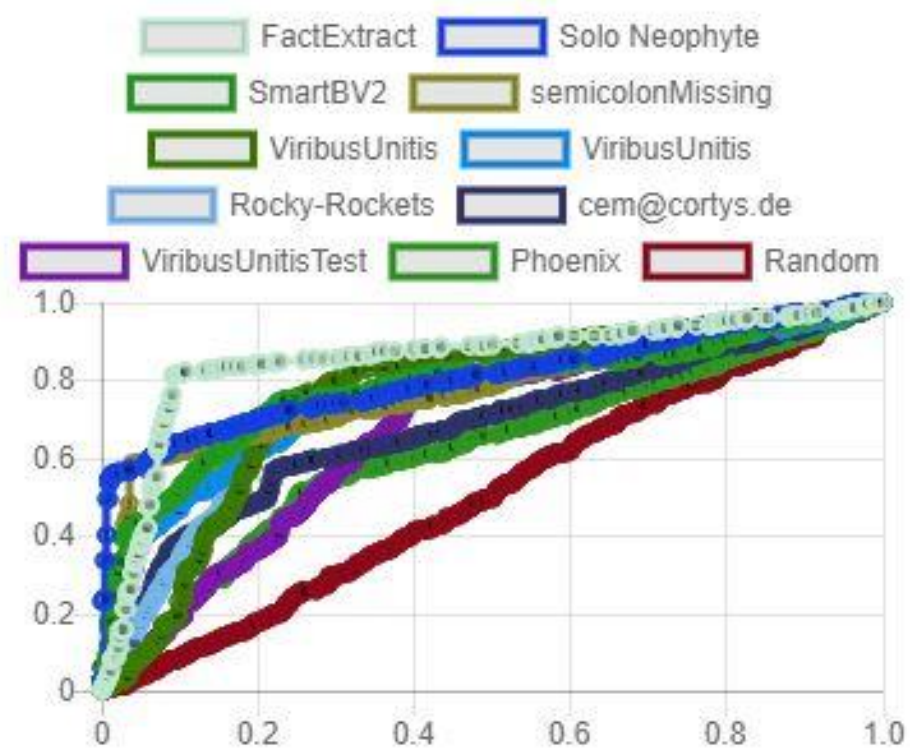
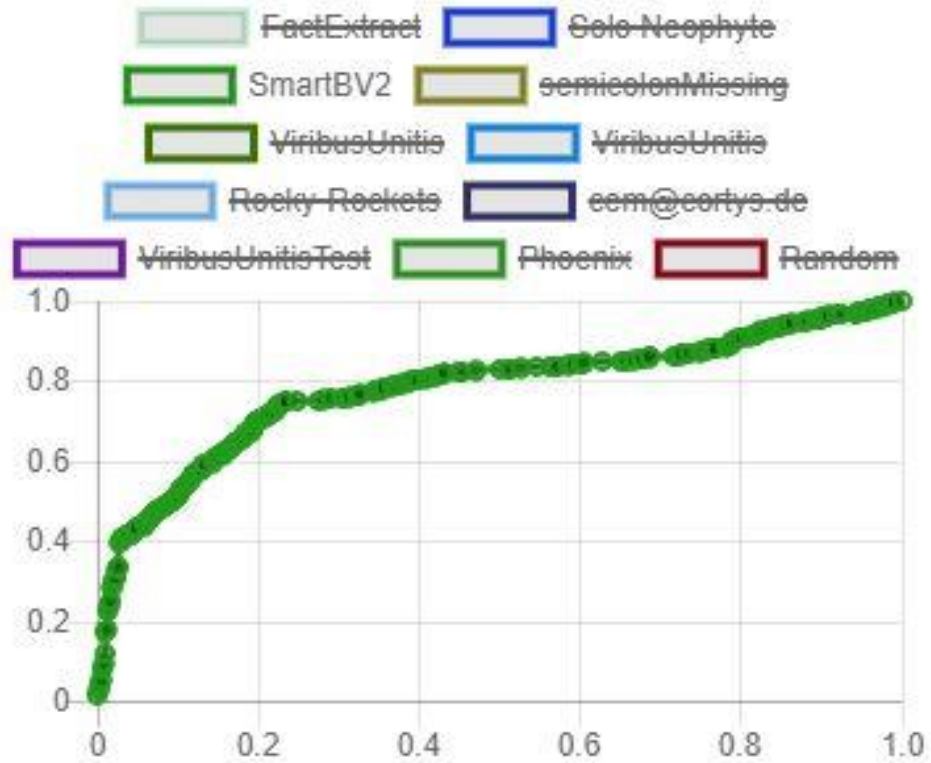
# Train Data: ROC



## Dataset: SNLP 2017 Test

AnnotatorName	Area Under Curve (AUC)
FactExtract	0.8522891670846036
Solo Neophyte	0.807378669804406
SmartBV2	0.7796333089001194
semicolonMissing	0.7757561436672972
ViribusUnitis	0.7616339068708766
ViribusUnitis	0.7540580031634583
Rocky-Rockets	0.7424699085683427
cem@cortys.de	0.680645036842714
ViribusUnitisTest	0.6794997299486901
Phoenix	0.6178291732572047
Random	0.5058784383318544

## Test Data: ROC



# Task Distribution

Tasks	Done by
Updating Wiki	Varma
Reading and Writing into file	Vineet
Parsing of data into FOX	Varma
Reading Json format and obtain entity	Varma
Search by entity in DBpedia	Varma
Tokenization of sentence	Vineet
Cleaning of stop words	Varma & Vineet
Comparison of DBpedia and token	Vineet
Calculation of confidence	Vineet

Thank you.