Natural Language Processing

Miles to go

Prasenjit Majumder

Natural Language

Spoken Language

Written Language

Vocal Language

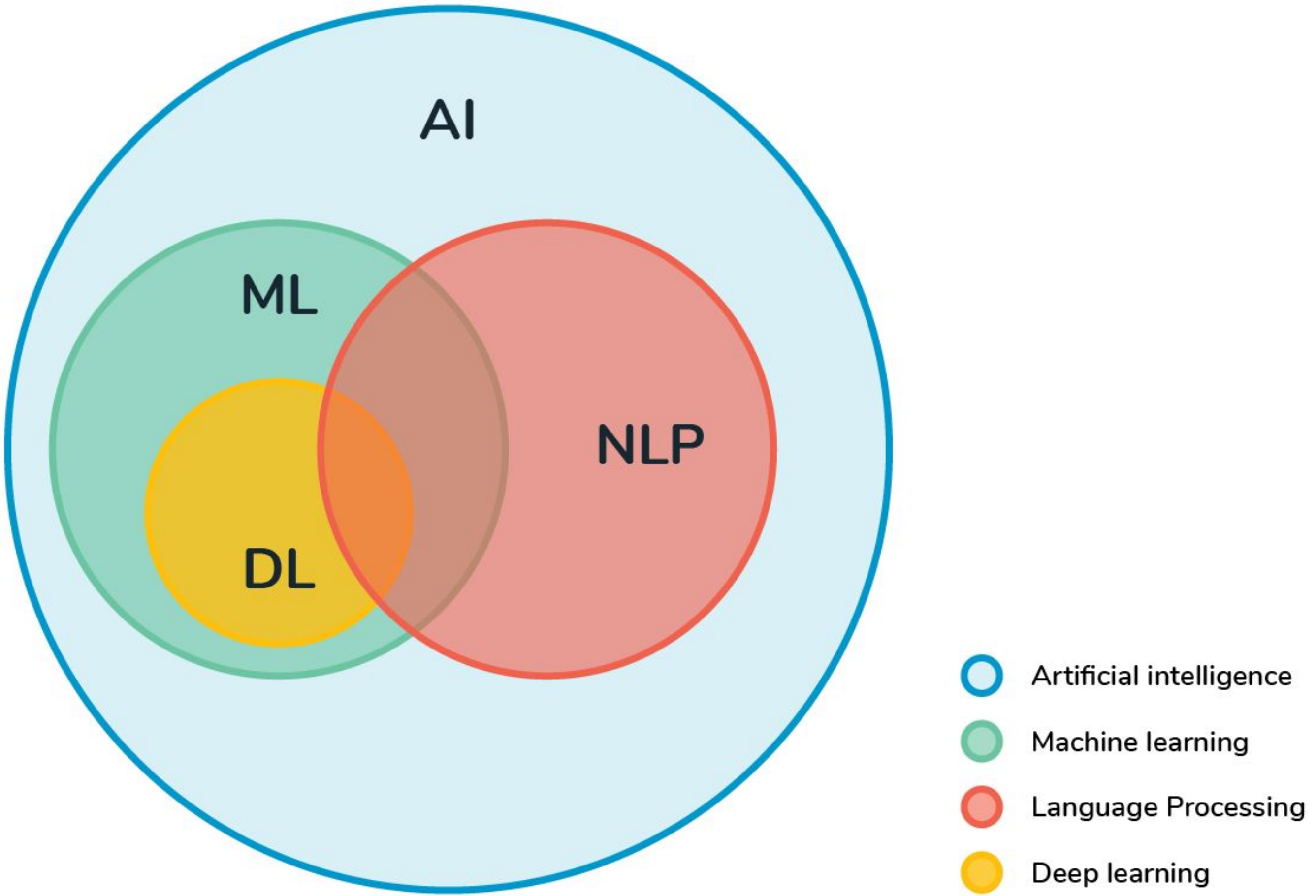
Sign Language

Writing systems

- Alphabets (English)
- Logographies (Chinese, Egyptian hieroglyphs)
- Abugida (Brahmic, Tibetan etc.)

Directions:

- Top-down,
- Left-Right,
- Right to Left


AI
ML
DL
NLP
Artificial intelligence
Machine learning
Language Processing
Deep learning

Natural Language Processing

- Natural Language Understanding
  - Information Retrieval
  - Summarization
- Natural Language Generation
  - Automatic Legal Drafting
  - Summarization

Word, Phrase, Sentences, Discourse

- Part of Speech
- Morphology
- Sense Disambiguation
- Entity Identification

Morphology

Unsupervised Root Words identification

Compute

Computer

Computing

Computerised

Computerization

Unsupervised Root Words identification

0	1	2	3	4	5	6	7	8	9	10	11	12	13
a	s	t	r	o	n	o	m	e	r	x	x	x	x
a	s	t	r	o	n	o	m	i	c	a	l	l	y

$$D_1 = \frac{1}{2^8} + \frac{1}{2^9} + \ldots + \frac{1}{2^{13}} = 0.0077$$

$$D_2 = \frac{1}{8} \times \left(\frac{1}{2^0} + \ldots + \frac{1}{2^{13-8}}\right) = 0.2461$$

$$D_3 = \frac{6}{8} \times \left(\frac{1}{2^0} + \ldots + \frac{1}{2^{13-8}}\right) = 1.4766$$

$$D_4 = \frac{6}{14} \times \left(\frac{1}{2^0} + \ldots + \frac{1}{2^{13-8}}\right) = 0.8438$$

Edit distance = 6

0	1	2	3	4	5	6	7	8	9
a	s	t	r	o	n	o	m	e	r
a	s	t	o	n	i	s	h	x	x

$$D_1 = \frac{1}{2^3} + \ldots + \frac{1}{2^9} = 0.2480$$

$$D_2 = \frac{1}{3} \times \left(\frac{1}{2^0} + \ldots + \frac{1}{2^{9-3}}\right) = 0.6615$$

$$D_3 = \frac{7}{3} \times \left(\frac{1}{2^0} + \ldots + \frac{1}{2^{9-3}}\right) = 4.6302$$

$$D_4 = \frac{7}{10} \times \left(\frac{1}{2^0} + \ldots + \frac{1}{2^{9-3}}\right) = 1.3891$$

Edit distance = 5

Unsupervised Root Words identification

y_{m-1}, but $x_m \neq y_m$).

$$D_2(X,Y) = \frac{1}{m} \times \sum_{i=m}^{n} \frac{1}{2^{i-m}} \quad \text{if } m > 0, \quad \infty \text{ otherwise}$$

$$D_3(X,Y) = \frac{n-m+1}{m} \times \sum_{i=m}^{n} \frac{1}{2^{i-m}} \quad \text{if } m > 0, \quad \infty \text{ otherwise}$$

$$D_4(X,Y) = \frac{n-m+1}{n+1} \times \sum_{i=m}^{n} \frac{1}{2^{i-m}}$$

Unsupervised Root Words identification

Table III. Retrieval Results for Various Stemmers (WSJ, queries 151–200)

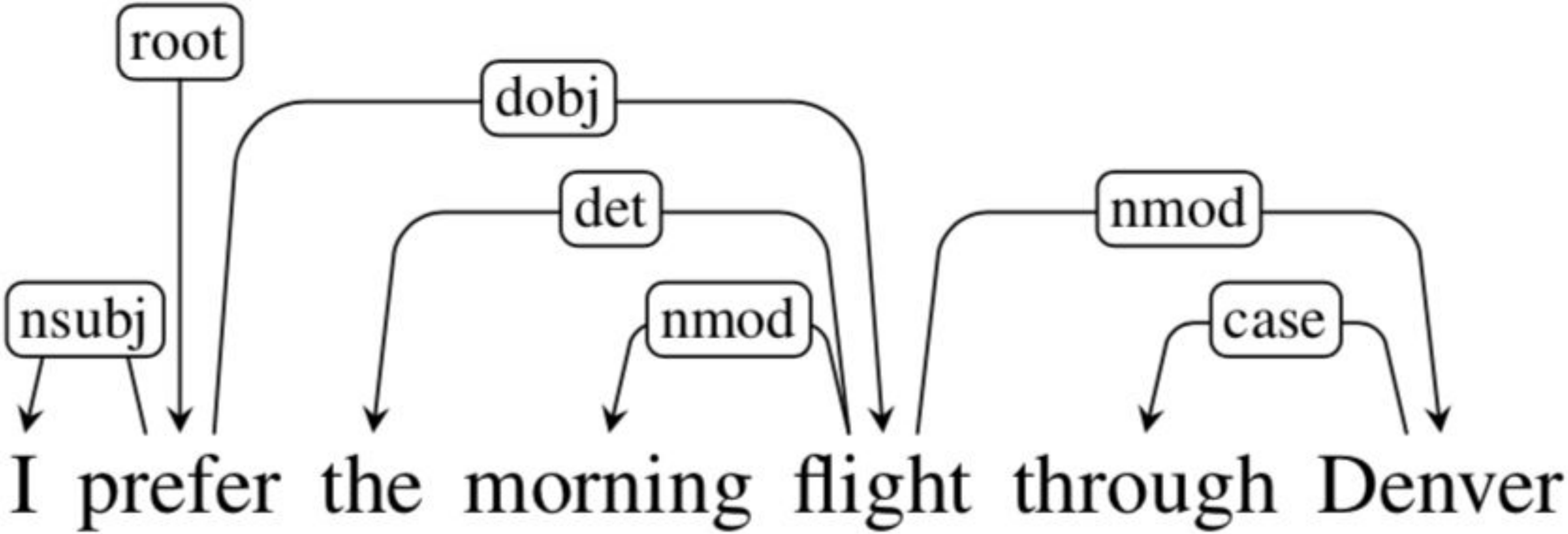
	No Stemming	$D_1 - 0.046$	$D_2 - 0.31$	$D_3 - 1.55$	$D_4 - 0.86$	Lovins	Porter	n-gram
Rel ret	3082	3235	3249	3268	3265	3318	3290	3171
P_{20}	0.4920	0.5020	0.4960	0.5090	0.5130	0.5030	0.5060	0.4960
Avg.P	0.3505	0.3732	0.3721	0.3796	0.3775	0.3746	0.3746	0.3595

Table VII. Performance of D_3-Based Stemmer on the French LeMonde Corpus

	No Stemming	$D_3(1.15)$	$D_3(1.55)$	$D_3(2.10)$	Porter
Rel ret	516	540	538	538	540
P_{20}	0.2222	0.2611	0.2578	0.2522	0.2467
Avg.P	0.3987	0.4301	0.4334	0.4153	0.4284

Parsing

Parsing


root
dobj
det
nmod
nsubj
nmod
case
I prefer the morning flight through Denver

Dependency Parsing

Basically, we represent **dependencies as a directed graph G= (V, A)** where V(set of vertices) represents words (and punctuation marks as well) in the sentence & A(set of arcs) represent the grammar relationship between elements of V.

A dependency parse tree is the directed graph mentioned above which has the below features:

- Root has no Incoming arcs (can only be Head in Head-Dependent pair)
- Vertices(except Root) should have only one incoming arc (Only one Parent/Head)
- A Unique path should exist between Root & each vertex in the tree.

Text representation

Text representation

TF-IDF

Latent semantic indexing

Word2Vec

Bidirectional Encoder Representations **from** Transformer

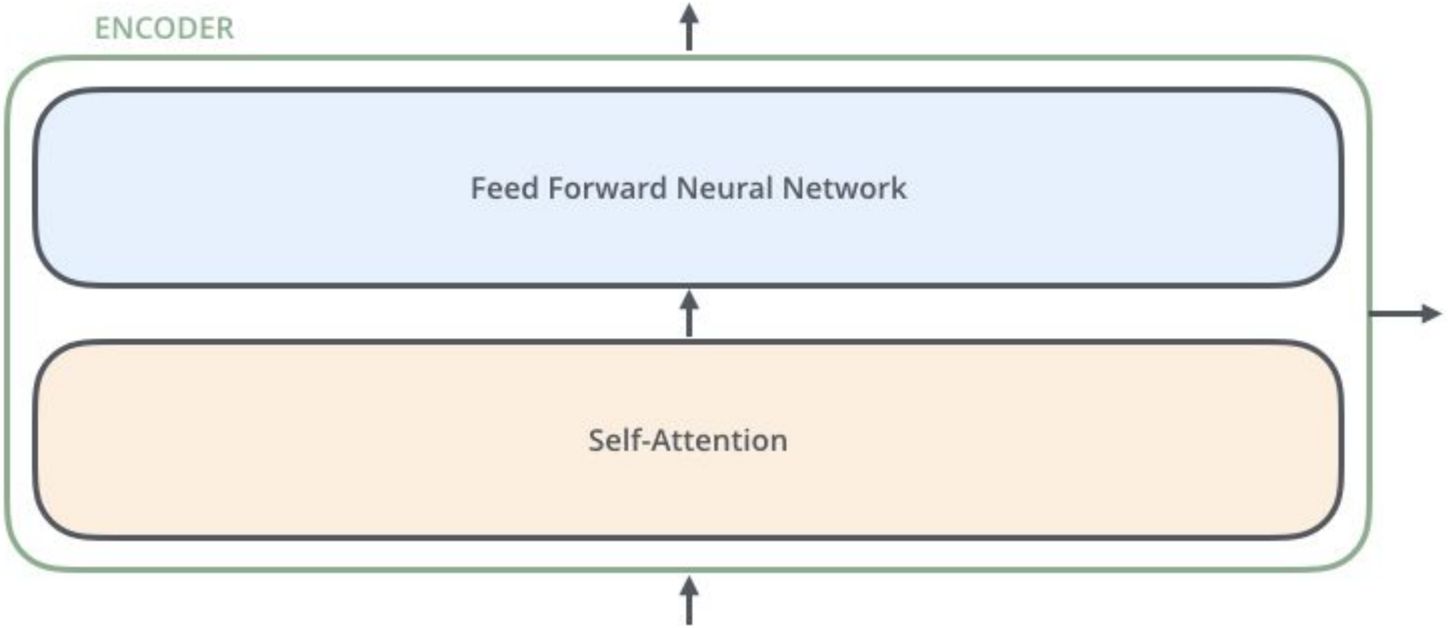
And many more...

Bidirectional Encoder Representations **from** Transformer (BERT)

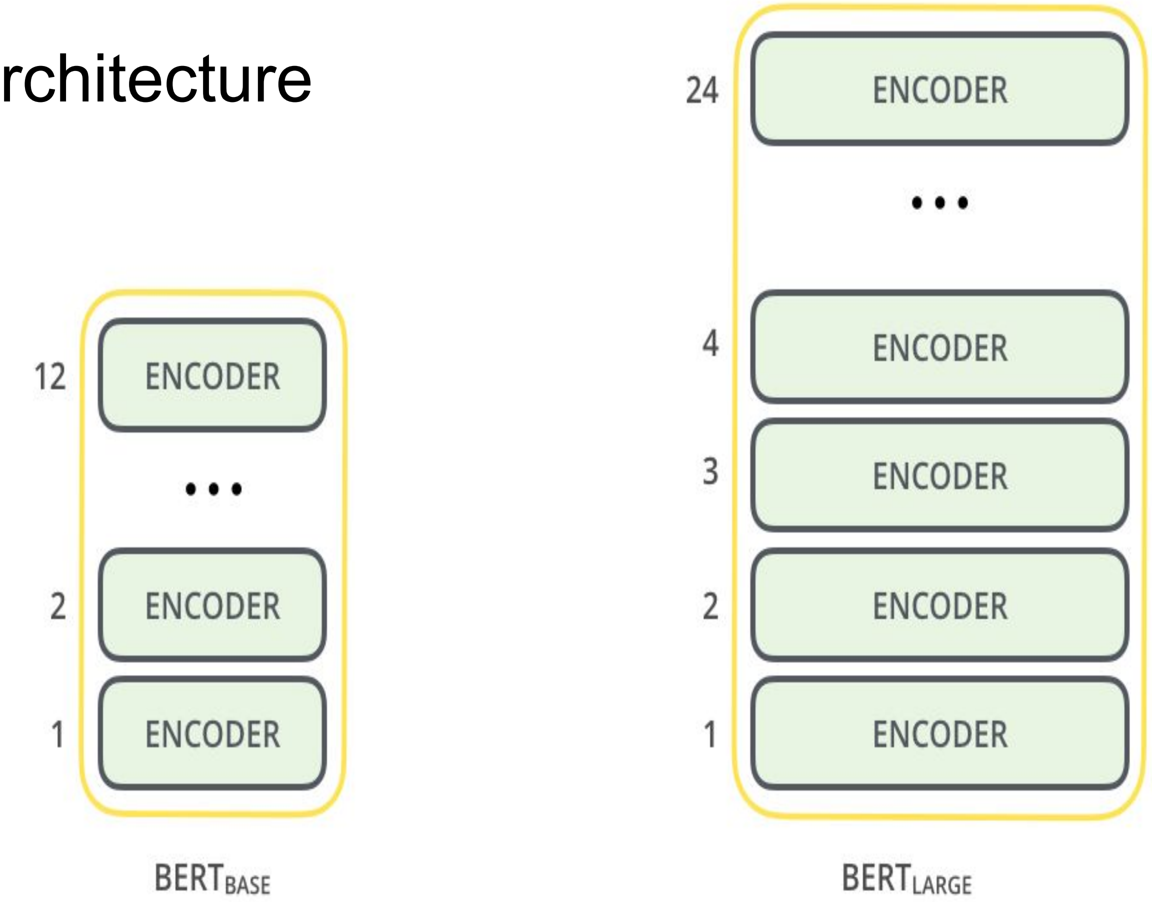
1. BERT (Bidirectional Encoder Representations from Transformers) uses Transformer, an attention mechanism that **“learns”** contextual relations between words (or sub-words) in a text.
2. BERT is pre-trained on two NLP tasks:
   a. Masked Language Modeling: Predict the masked word given the context words
   b. Next Sentence Prediction: Given a sentence predict the next sentence.
3. As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once.
4. BERT is pre-trained on a large corpus of unlabelled text which includes the entire Wikipedia (2,500 million words) and Book Corpus (800 million words).

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

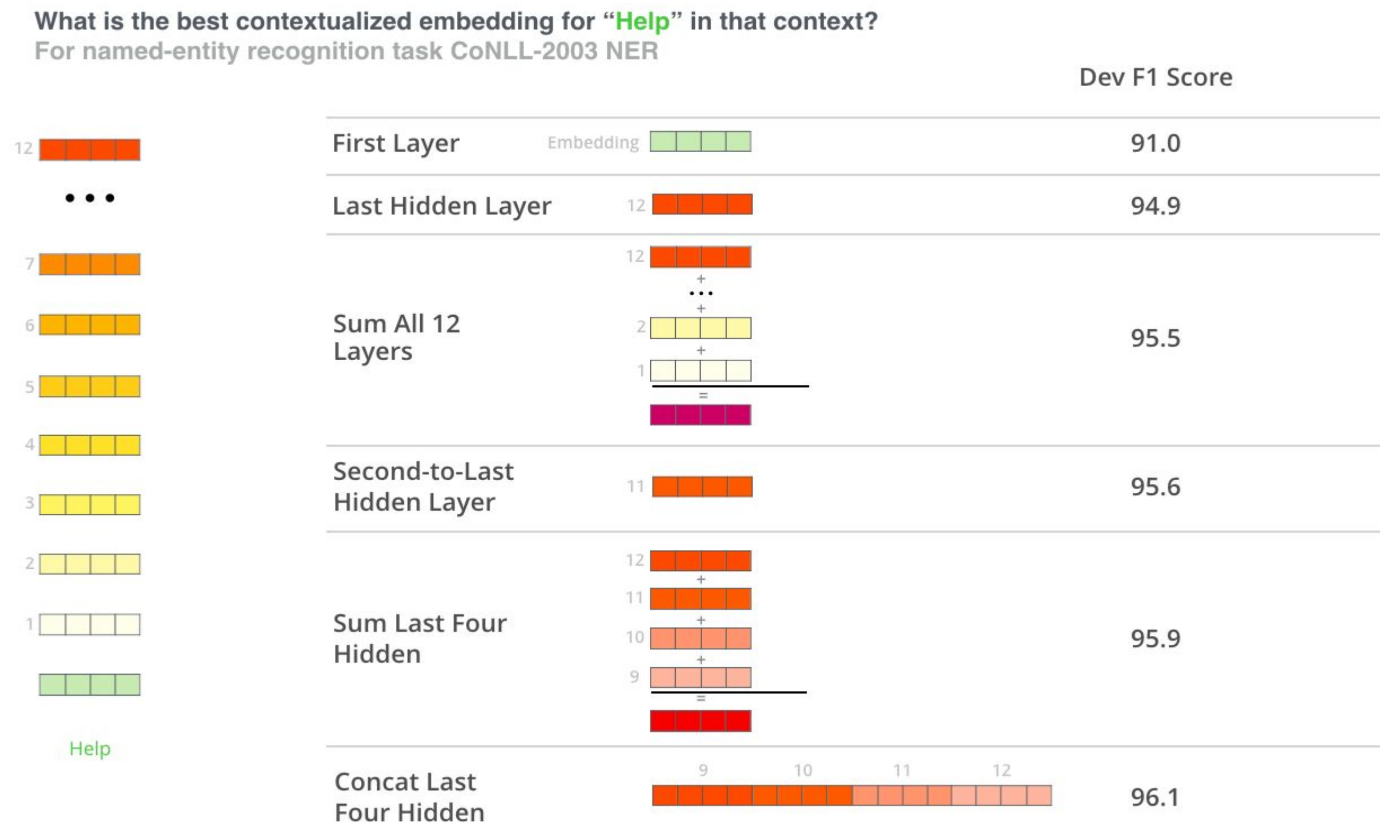
BERT Architecture



BERT Architecture



BERT Architecture



Challenges in Downstream tasks :

- Search Engines
- Hate Speech Detection
- Sentiment Analysis
- Question Answering
- Recommendation
- Summarization

Summarization

DATA

DUC 2002, DUC 2003 and DUC 2004

1. **DUC 2002: 59 clusters of around 10 documents each (TREC collection)**
2. **DUC 2003: 30 clusters of about 10 documents each (TDT Datasets)**
3. **DUC 2004   50 clusters with 10 documents per cluster. (TDT Datasets)**

**All three datasets include four manually written summaries per cluster.*

Effect of pre-processing and post-processing steps on ROUGE-1 recall.

	System	No pre/post processing	Only stemming	Only stopword removal	Only redundancy removal	Stopword + Redundancy removal
DUC 2002	Centroid	0.41783	0.42001	0.42223	0.43157	**0.44987**
	Greedy-KL	0.40173	0.40537	0.41392	0.40962	**0.41522**
	LexRank	0.42733	0.42000	0.42292	**0.44134**	0.43289
	FreqSum	0.39247	0.38120	0.40480	0.38766	**0.42522**
DUC 2003	Centroid	0.33387	0.34222	0.34382	0.35237	**0.36780**
	Greedy-KL	0.31473	0.31263	0.33892	0.31592	**0.33892**
	LexRank	0.35643	0.34900	0.34292	**0.36111**	0.35689
	FreqSum	0.29316	0.30120	0.32748	0.30486	**0.34335**
DUC 2004	Centroid	0.35399	0.35104	0.34874	0.36541	**0.37271**
	Greedy-KL	0.31913	0.32215	0.33717	0.31866	**0.34160**
	LexRank	0.35356	0.34343	0.34453	**0.36277**	0.35377
	FreqSum	0.30776	0.31500	0.34816	0.31370	**0.35851**

Hate Speech Detection

User Aggression Detection[1]

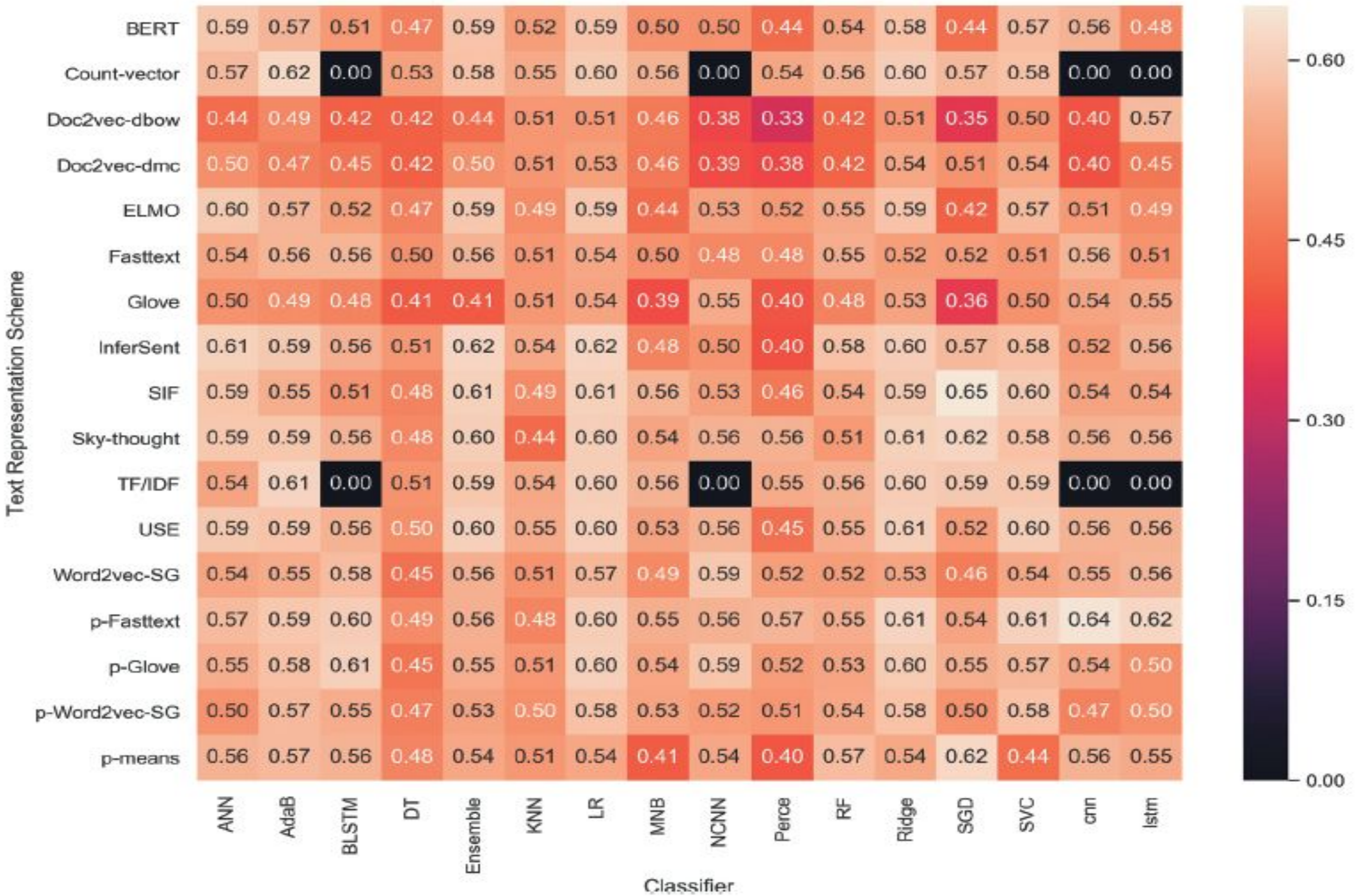
NAG: Non- Aggressive

CAG: Covertly Aggressive

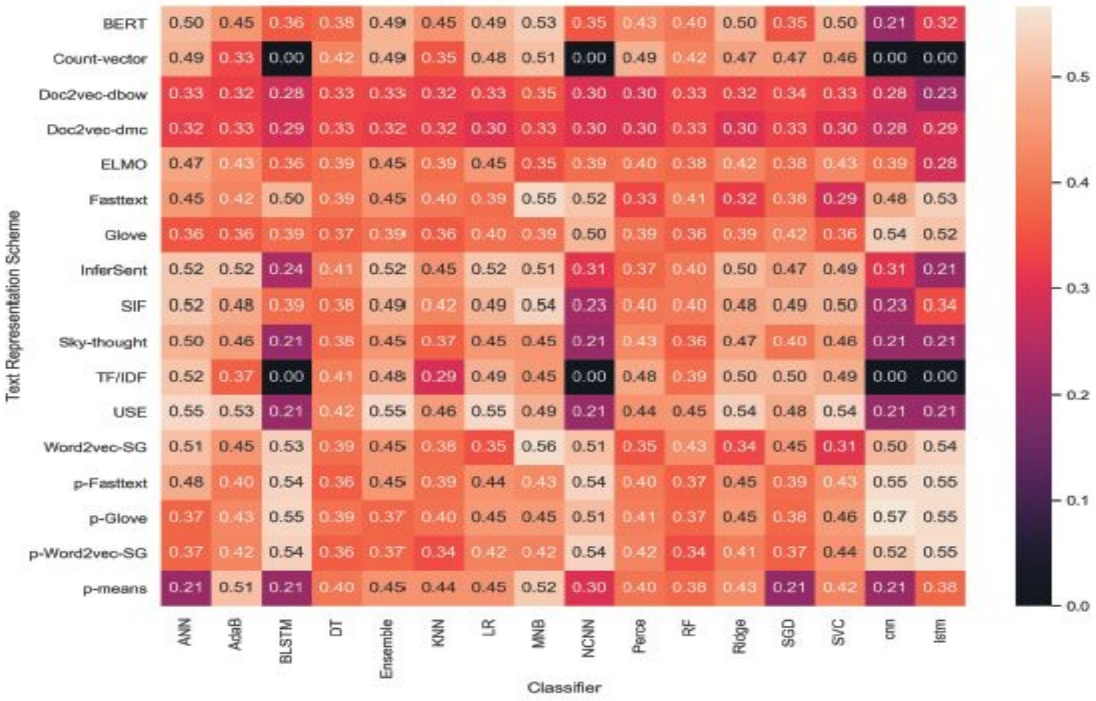
OAG: Overtly Aggressive

[1] R. Kumar, A. N. Reganti, A. Bhatia, and T. Maheshwari. Aggression-annotated Corpus of Hindi-English Code-mixed Data. In Proceedings of the 11th Language Resources and Evaluation Conference (LREC), Miyazaki, Japan, 2018.

Heatmap: Results on TRAC Facebook English Dataset

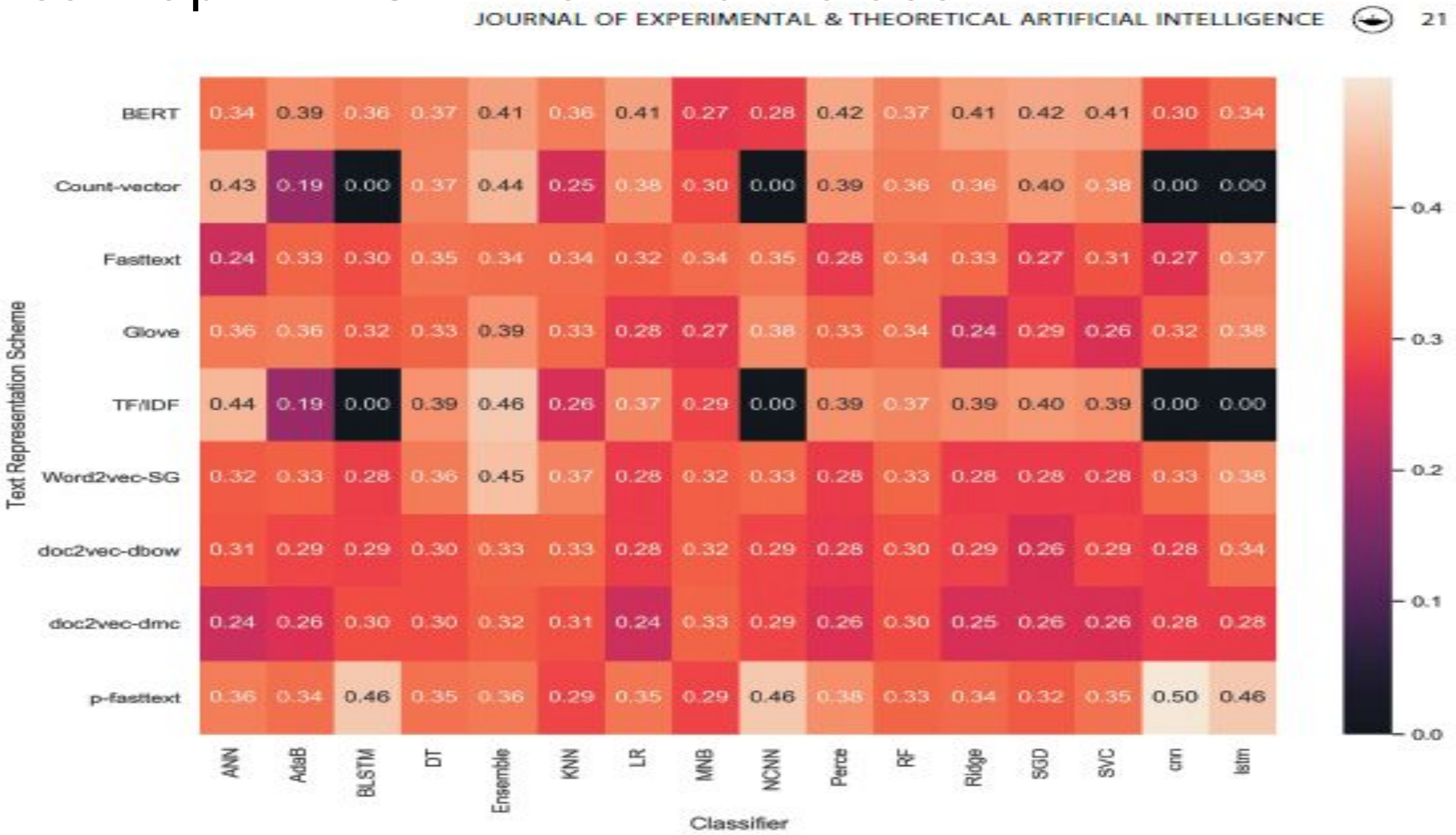


Heatmap: Results on TRAC Twitter English Dataset



Heatmap: TRAC Facebook Hindi Dataset

Heatmap: TRAC Twitter Hindi Dataset



Hate Visualization on FB : Using Browser Plugin

Hate Visualization on FB : Using Browser Plugin

Multilingual and Domain Specific

- Japanese Patent retrieval task
- Arabian Text summarization
- Bengali news recommendation systems
- English Chinese cross lingual Information retrieval systems

TREC, NIST, USA

NTCIR, Japan

FIRE, India

Evaluation

Training data (human annotated data)

+

Evaluation Metrics

+

Test and Validation Data

Its a round the year process.

Evaluation

TREC, NIST, USA

CLEF, EU

NTCIR, Japan

FIRE, India