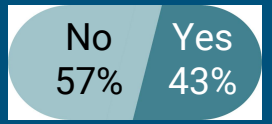
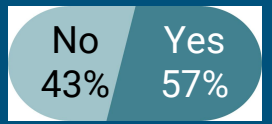
Semantic Similarity using WordVector

Rakshit Sharma
Dept. of Computer Science
IIT Kanpur Kanpur-208016

Advisor:- Prof Amitabha Mukerjee
Dept. of Computer Science
IIT Kanpur Kanpur-208016

MOTIVATION

- No proper parser available for Indian language.
- Local language user makes up to large user base on the internet.

	Internet Users	Local Language Users	Growth of Local Language user base	Penetration
ALL INDIA	269Mn	127Mn	47%	
URBAN INDIA	188Mn	81Mn	51%	
RURAL INDIA	81Mn	46Mn	41%	

Goals and Targeted Language

- Collect and create dataset in Hindi language.
- Word level similarity using WordVector.
- Sentence level similarity using word level similarity.
- Implementation using Semantic Nets.
- Target language is HINDI.

Challenges

- Data Collection:-
  - What search queries to provide to collect good mix of similar and dissimilar news articles and twitter tweets.
  - Word disambiguation in Hindi.
  - Use of some plural words meaning as singular. Eg:-
    “उसने **कौवे** को देखा”
  - More may come in future.

Approach

- The approach is similar to Yuhua Li et al [1] for “Sentence Similarity Based on Semantic Nets and Corpus Statistics”.
- Word level similarity using Word2Vec.
- Derive Semantic Vector for sentences.

Approach contd.

TABLE 1
Process for Deriving the Semantic Vector

	RAM	keeps	things	being	worked	with	The	CPU	uses	as	a	short-term	memory	store
RAM	1												0.8147	0.8147
keeps		1												
things			1					0.2802	0.4433					
being				1										
worked					1									
with						1								
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
$\vec{s}$	1	1	1	1	1	1	0	0.2802	0.4433	0	0	0	0.8147	0.8147
Weight	I(RAM) I(RAM)	I(keeps) I(keeps)	I(things) I(things)	I(being) I(being)	I(worked) I(worked)	I(with) I(with)	I(The) I(The)	I(CPU) I(things)	I(uses) I(things)	I(as) I(as)	I(a) I(a)	I(short-term) I(short-term)	I(memory) I(RAM)	I(store) I(RAM)

Progress

- Data Extraction:-
  - Tweets Extracted using frequent keywords.
  - Filtering of tweet data to make it good(a good mix of similar and dissimilar tweets) in progress
  - Clustering of news articles on the basis of topic.
    News articles on same topic will have “more similar words”.

- Semantic Similarity:-
  - Word level similarity calculated using Word2Vec.
  - Continuing the work to get sentence level similaarity.

Tweet Collection

testtweet.py - C:\Users\raksh\Desktop\NLP_Project\data\testtweet.py (2.7.10)

File Edit Format Run Options Window Help

```
import tweepy
import sys
import jsonpickle
import os
import codecs

auth = tweepy.auth.OAuthHandler('dJvGhNEaw6ivlQGals8g1mMu9', 'n5qBLoWVO77KHLfAUB1Z5wTYtO7ed62XmNbq52bmYjEFYdJNQi')
api = tweepy.API(auth)
searchQuery1 = u'\u0928\u0930\u0947\u0902\u0926\u094d\u0930 \u092e\u094b\u0926\u0940'   # this is what we're searching for
searchQuery2 = u'\u0935\u093f\u0926\u0947\u0936 \u0926\u094c\u0930\u093e'
maxTweets = 100 # Some arbitrary large number
tweetsPerQry = 100  # this is the max the API permits
fName1 = 'Narendra_Modi.txt' # We'll store the tweets in a text file.
fName2 = 'Foreign_Trip.txt'

tweetCount = 0
print("Downloading max {0} tweets".format(maxTweets))
f1 = codecs.open(fName1, encoding='utf-8', mode='w+')
f2 = codecs.open(fName2, encoding='utf-8', mode='w+')

while tweetCount < maxTweets:
    try:
        new_tweets1 = api.search(q=searchQuery1,lang ='hi', count=tweetsPerQry)
        new_tweets2 = api.search(q=searchQuery2,lang ='hi', count=tweetsPerQry)
        for tweet in new_tweets1:
            f1.write(tweet.text + chr(28) +
                     '**************************************' + chr(28))
        for tweet in new_tweets2:
            f2.write(tweet.text + chr(28) +
                     '**************************************' + chr(28))
        tweetCount += len(new_tweets1)
        print("Downloaded {0} tweets".format(tweetCount))
    except tweepy.TweepError as e:
        print("some error : " + str(e))
        break
print ("Downloaded {0} tweets, Saved to {1}".format(tweetCount, fName1))
```

Administrator: Command Prompt

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd\

C:\>cd Users

C:\Users>cd raksh

C:\Users\raksh>cd Desktop

C:\Users\raksh\Desktop>cd NLP_Project

C:\Users\raksh\Desktop\NLP_Project>cd data

C:\Users\raksh\Desktop\NLP_Project\data>python testtweet.py
Downloading max 100 tweets
Downloaded 100 tweets
Downloaded 100 tweets, Saved to Narendra_Modi.txt

C:\Users\raksh\Desktop\NLP_Project\data>
```

Results per Paper(Yuhua Li et al)

TABLE 3
Sentence Data Set Results

R&G No.	R&G Word Pair	Human Similarity (Mean)	Algorithm Similarity Measure	R&G No.	R&G Word Pair	Human Similarity (Mean)	Algorithm Similarity Measure
1	Cord smile	0.01	0.33	51	Glass tumbler	0.14	0.65
5	Autograph shore	0.01	0.29	52	Grin smile	0.49	0.49
9	Asylum fruit	0.01	0.21	53	Serf slave	0.48	0.39
13	Boy rooster	0.11	0.53	54	Journey voyage	0.36	0.52
17	Coast forest	0.13	0.36	55	Autograph signature	0.41	0.55
21	Boy sage	0.04	0.51	56	Coast shore	0.59	0.76
25	Forest graveyard	0.07	0.55	57	Forest woodland	0.63	0.70
29	Bird woodland	0.01	0.33	58	Implement Tool	0.59	0.75
33	Hill woodland	0.15	0.59	59	Cock rooster	0.86	1.00
37	Magician oracle	0.13	0.44	60	Boy lad	0.58	0.66
41	Oracle sage	0.28	0.43	61	Cushion pillow	0.52	0.66
47	Furnace stove	0.35	0.72	62	Cemetery graveyard	0.77	0.73
48	Magician wizard	0.36	0.65	63	Automobile car	0.56	0.64
49	Hill mound	0.29	0.74	64	Midday noon	0.96	1.0
50	Cord string	0.47	0.68	65	Gem jewel	0.65	0.83

References

- Yuhua Li et al “Sentence Similarity Based on Semantic Nets and Corpus Statistics” (AUGUST 2006) (IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 18, NO. 8) http://www.aicit.org/AISS/ppl/AISS1666PPL.pdf

- http://articles.economictimes.indiatimes.com/2015-08-18/news/65530379_1_second-largest-internet-base-internet-population-google-translate

- http://www.livemint.com/Industry/rad15YLFMTsWotnNAYKJbL/Local-language-Internet-users-grow-to-127-million-in-India.html

THANK YOU

QUESTIONS?