Paraphrasing Using Word2Vec

Rakshit Sharma

Department of Computer Science, IIT Kanpur

November 13, 2015

Abstract

Finding semantic similarity touches wide range of problems in real life scenarios. It may be used to summarize the news articles, to find similarities in the reviews of some product so as to get pros and cons of a product, etc. For doing semantic similarity, the problem with Indian language is that there is no proper parser available for the Indian languages. Hence we proceed through WordVector implementation.

This project employs sentence similarity of short texts based on the word order similarity which we get from the Word2Vec. We aim to exploit this sentence similarity to merge the news articles which are clustered as per topic.

Related Works

Finding sentence similarity has a huge impact on text-related research. This has been a consistent topic since last 4 years in SemEval events. Each of the papers submitted in these tasks have its own advantages and disadvantages. But most of them work on usage of parsers to preprocess the input.

Motivation

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

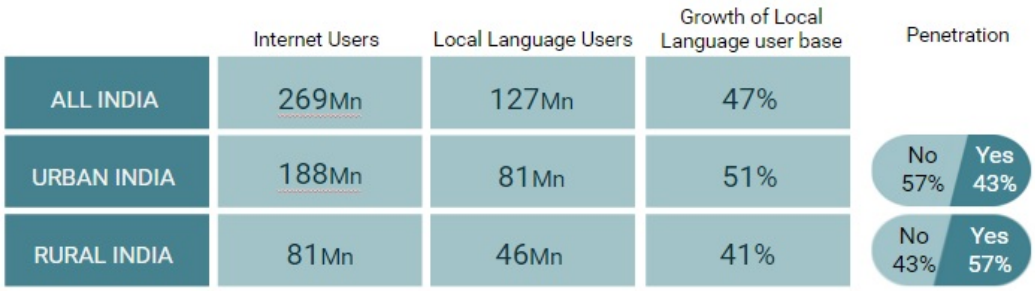

Figure 1: Internet userbase distribution in India

Dataset

Hindi news dataset of about 300 news articles

Methodology

The approach is similar to Yuhua Li et al[1] for Sentence Similarity Based on Semantic Nets and Corpus Statistics.
Get hierarchical knowledge base by using Word2Vec and then getting a raw word similarity based on common synsets.
We take the joint word set of the two sentences that we have to compare then we make an ndimensional vector each for sentence

T1 and T2 where n is the sum of number of words in T1 and T2 followed by calculating weights of each dimension as:-

$$S_1(w) = \begin{cases} 1 & \forall w \in T_1 \cup T_2,\ w \in T_1 \\ \text{similarity score} & \forall w \in T_1 \cup T_2,\ w \in T_2 \end{cases} \quad (1)$$

$$S_2(w) = \begin{cases} 1 & \forall w \in T_1 \cup T_2,\ w \in T_2 \\ \text{similarity score} & \forall w \in T_1 \cup T_2,\ w \in T_1 \end{cases} \quad (2)$$

We need to normalize the semantic vectors in the sentence by multiplying the scores by the associative info of the words. Associative info a word(I(w)) can be calculated as:-

$$I(w) = 1 - TFIDF(w) \quad (3)$$

Then we can get normalized semantic vectors as:-

$$(w) = \ S_1(w) * I(w) * I(\bar{w}) \ \ \bar{w} = \text{similar\_word(w)} \quad (4)$$

$$(w) = \ S_2(w) * I(w) * I(\bar{w}) \ \ \bar{w} = \text{similar\_word(w)} \quad (5)$$

Semantic similarity can then be calculated as:-

$$S_s = \frac{S_1.S_2}{\|S_1\|\|S_2\|} \quad (6)$$

But getting just the semantic vectors is not sufficient to find similarity.
Consider the exacmple:-

"Fox jumped over the dog"
and
"Dog jumped over the fox"

The above pair of words are considerably dissimilar than each other.
But as we are considering only the semantic vector at this point of time and considering only the amount of similar words, the above pair will result in a high similarity. Reason being the words of one sentence are just jumbled to get next sentence.
To overcome this problem, we can exploit the ordering of words in the sentence.
So, we define *"word order vectors"* for both the sentences:-

$$R_1(w) = \begin{cases} \text{index(w} \in T_1) & \forall w \in T_1 \cup T_2,\ w \in T_1 \\ \text{index(similar word} \in T_2) & \forall w \in T_1 \cup T_2,\ w \notin T_1 \end{cases} \tag{7}$$

$$R_2(w) = \begin{cases} \text{index(w} \in T_1) & \forall w \in T_1 \cup T_2,\ w \in T_2 \\ \text{index(similar word} \in T_2) & \forall w \in T_1 \cup T_2,\ w \notin T_2 \end{cases} \tag{8}$$

Word order similarity can then be calculated as:-

$$S_r = 1 - \frac{\|R_1 - R_2\|}{\|R_1 + R_2\|} \tag{9}$$

And finally overall similarity as:-

$$S(T_1, T_2) = \delta S_s + (1 - \delta) S_r \tag{10}$$

Once we have the results, recognize the threshold to classify similar and dissimilar by ROC curve. The binary classifier can then be used to merge the news articles based on what are similar and dissimilar sentences in the articles.

	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								
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
$\tilde{s}$	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)

Figure 2: Process for deriving the Semantic Vector

Experimental Results

S.No.	Sentence Pairs	Similarity	S.No.	Sentence Pairs	Similarity
1	बीजेपी की बिहार में करारी हार महागठबंधन ने बीजेपी को बिहार में 100 सीटों से हराया	0.650628	5	नरेंद्र मोदी अपने मंत्रियों के बड़बोलेपन से परेशान नरेंद्र मोदी का मंत्रियों के साथ विदेश दौरा टला	0.558449
2	बीफ की अफवाह पर भीड़ ने एक इंसान को जान से मार डाला दादरी में भीड़ ने एक मासूम को जान से मारा	0.633209	6	उमा जी कहती है नरेंद्र मोदी विनाश पुरुष नरेंद्र मोदी के गुजरात में विकास नहीं सिर्फ दंगे हुए हैं	0.319521
3	यह एक जूता है मैं कल जूता लेने गया	0.219979	7	छपरा के मढ़ौरा में प्रधानमंत्री नरेंद्र मोदी की रैली बिहार चुनाव के मद्देनज़र पार्टियों की रैली जोर-शोर पर	0.436039
4	आज NLP की कक्षा है यह एक जूता है	0.0	8	नीतीश कुमार सन्निपात रोगी की तरह परस्पर विरोधी बातें कर रहे हैं नीतीश कुमार प्रधानमंत्री नरेंद्र मोदी की यात्रा से भी परेशान होते हैं	0.4790112

Figure 3: Similarity scores on various pair of sentences

Conclusions

An approach was given to calculate sentence similarity score with the help of Word2Vec. Sentences having similar words tend to give more score irrespective of the overall sentence meaning, hence considered the impact of word order on sentence meaning. However, still without proper word disambiguation, similarity score was much higher then human score in some cases like in example 5 and vice-versa as in example 6. However in real life scenarios, sentences having many similar words tend to convey same meaning most of the time.

References

[1] Yuhua Li, David McLean, Zuhair A. Bandar, James D. OShea, and Keeley Crockett.
Sentence Similarity Based on Semantic Nets and Corpus Statistics
IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 18, NO. 8, AUGUST 2006

[2] *Userbase information1*

[3] *Userbase information2*