CONCEPTUALIZED SHORT TEXT EMBEDDING USING KNOWLEDGE BASE

[1]Natasha J, [2]Karpaga Priyaa K, [3]Vijayrani J

[1]M.E.Scholar, [2]M.E.Scholar, [3]Teaching Fellow
[1]Department of CSE
[1] Anna University, Chennai, India

***Abstract**:* Most short text embedding models typically represent each short text only using the literal meanings of the words, which make these models indiscriminative for the ubiquitous polysemy. In order to enhance the semantic representation capability of the short texts, a novel short text conceptualization algorithm is proposed to assign the associated concepts for each short text using Probase and Concept Net and then introduce the conceptualization results into learning the conceptual short text embeddings. Hence, this semantic representation is more expressive than some widely used text representation models such as the latent topic model. Wherein, the short text conceptualization algorithm used here is based on a novel co-ranking framework, enabling the words and concepts to fully interplay to derive the solid conceptualization of short texts. Afterwards, short text embedding models are extended with this conceptualized short text words and concepts to make more efficient prediction. The experiments on the real world datasets demonstrate that the proposed short text conceptualization algorithm and the conceptualized short text embedding model are more effective than the state of the art methods.

***IndexTerms* - Short text, Conceptualization, Knowledge base, Embedding, CBOW, Skipgram**

I. INTRODUCTION

In today's many Natural Language Processing (NLP) applications require the input text to be represented as a fixed-length feature, of which the short text embedding is very important [1], [2], [3] , one of the most widely used fixed length vector representation for the text is the bag-of-words or the bag-of-n-grams. Despite that they suffer severely from data sparsity and high dimensionality and have very little sense about the semantics of the words. Recently, for short text representation and classification the Deep Neural Network (DNN) approaches have achieved the state-of-the-art results [4], [5], [6].

Despite their usefulness, recent short text embedding faces several challenges. Most short text embedding models represent each short text only using the literal meanings of the words, which makes these models indiscriminative for the ubiquitous polysemy. For short text, neither parsing nor topic modeling works well because there are simply not enough signals in the input. To resolve this, related concepts must be derived from the input short text to generate the solid semantic representation for the given short text. Hence, this study investigates how to introduce short text conceptualization into short text embedding, and proposes the Conceptual Short text Embedding (CSE): Given a short text, we first obtain its concept distribution by a novel short text conceptualization algorithm, and then utilize this concept distribution to learn the vector representation for this short text. In conclusion, the proposed CSE could be divided into the following two steps:

(i) Short text conceptualization
(ii) Short text embedding

In the overall architecture of CSE (Figure 2), the concept distribution plays an important role as a bridge between short text conceptualization and short text embedding and contributes more to the semantic representation of the given short text. A novel short text conceptualization algorithm is proposed to generate the concept distribution (θ_c) for the given short text.

Recently, short text conceptualization has played an increasingly vital role in the text understanding and other applications [7], [8], It is a task to map a piece of short text to a set of open domain concepts with different granularities. Recent probabilistic algorithms have demonstrated the remarkable success. However most of them are limited to the assumption that all the observed words in the given short text are conditionally independent ignoring the interactions among the words and the concepts, especially the beneficial reactions from the concepts to the words. Therefore, they unfortunately could not release the solid concept representation. To overcome these drawbacks above, a novel framework for co-ranking the words and their corresponding concepts is presented. As a result, the improved rankings of the words and their concepts depend on each other in a mutually reinforcing way and thus it can take advantage of the additional information implicit in such heterogeneous semantic network of the words and the concepts.

With the aforementioned concept distribution (θ_c), a novel short text embedding algorithm is derived to generate the semantic vector representation for the given short text. Overall, the proposed Conceptualized Short text Embedding is an unsupervised model that learns the distributed vector representations for conceptualized short text. These concept level vector representations for the short texts are learned to predict the target word or the surrounding words in context.

Thus an ideal co-ranking framework is introduced to address the problem of short text conceptualization. After co-ranking the words and the corresponding concepts simultaneously during an iterative procedure, the most expressive concepts and contextual keywords are obtained. The concepts are integrated to allow the resulting conceptual short text embedding to model different meanings of a word under different concepts and different contexts. The experimental results on the real world datasets demonstrate that the proposed concept level short text representation is robust and the co ranking framework for short text conceptualization is effective.

II RELATED WORK

2.1 Short Text Conceptualization

Shot text conceptualization, is an emerging task to infer the most likely concepts for the words in the given short text, which could help better make sense of the short text and extend the short text with the categorical or topical information [3], [7], [8], [12]. Since the short texts usually lack enough content from which the statistical conclusions could be drawn easily, the mining results from the traditional algorithms often have low interpretability. To assist short text conceptualization, recent work has put more emphases on using the signals from the lexical knowledge bases [7], [13], [14], [15], and has achieved great success. Typical concept mapping methodologies include the so called probabilistic conceptualization and explicit semantic analysis (ESA) [16]. Based on the Bayesian inference mechanism and utilized the probability to rank the concepts from a probabilistic knowledge base, and selected the concepts with the largest probabilities to represent the given short text. The basic assumption behind these algorithms is that, given each concept, all the observed words are conditionally independent. Aiming at mining more signals from the noisy and sparse short texts, leveraged the verbs, adjectives and attributes of the instances, which provided the valuable clues to understand the instances. Recently, some studies tried to leverage deep neural network based approaches for revealing the semantics of a short text based on its enriched representation [10], [17], [18], however failing to overcome the problems of the high computation consumption and the lack of labeled data. Overall, traditional probabilistic short text conceptualization algorithms have some major drawbacks. They assume that all the observed words in the given short text are conditionally independent, while dismissing the interactions among the concepts (and the words) and the beneficial reactions from the concepts to the words. Unfortunately, in real world, this assumption does not always hold true, and actually the words (and the concepts) interact with each other closely. Although Hua et al. [7] attempted at introducing the word-relationship to enhance the semantic representation, they relied heavily on dependency parsing or syntax analysis, which are hard for the short text because its language is usually not well-written .Furthermore, some studies somehow only ranked top-N concepts for each word in the given short text, and the resulting concept-lists for different words were disjoint [23].Hence they could not generate the global concept distribution for the entire short text. Moreover, the traditional Naive Bayes quickly boosts the general and vague concepts (e.g., Topic or Thing) co-occurred with all the observed words and dismisses the concepts partially matching the words, which would be more specific and descriptive for representing the short text. In other words, mining the contextual keywords in the current shorttext seems to contribute to conceptualization. Therefore, a framework must be derived such that it enables the signals to fully interplay. Nevertheless, to the best of their knowledge, they were not aware of any attempts to correlate the interactions among the concepts (and the words) and the reactions from the concepts to the words. Accordingly, a co-ranking framework is presented to comprehensively address these problems. Recently, heterogeneous information analysis begins to receive researchers' increasing attention [37]. By efficiently taking advantage of the additional information implicit in the heterogeneous networks, the co-ranking framework has been developed for many applications. Actually, the main intuition behind co-ranking is that there exists a mutually reinforcing relationship among the multiple typed entities that could be reflected in the rankings. However, our proposed adaptation of the co-ranking framework to the shorttext conceptualization task is novel, and could make the signals(i.e., the words and the concepts) to be fully integrated for deriving the solid conceptualization for the given short text.

2.2 Short Text Embedding

The one-hot short text representation has been widely used as the basis of the bag-of-words (BOW) text model however, it cannot take the semantic information into consideration. Recently, the deep neural network approaches achieved the state-of-the-art results in short text embedding [5], [19], [21], most of which are inspired by the word embedding approaches. Wang et al. [22] presented neural network architecture for obtaining the high-quality word embeddings directly optimized for the objective of the short text representations.Le et al. [20] proposed the Paragraph Vector (PV) model that represents each document by a dense vector which is trained to predict the words in the context. However, their models depended only on the literal meanings of the words, ignoring the semantic information suchasthe topicsor the concepts. In this paper, they have extended the PV model by introducing the concept in formation and the attention model. Aiming at enhancing discriminativeness for polysemy, [5] employed the latent topic models to assign topics for each word in corpus. It then learned the topical word embeddings and the topical short text embeddings based on the topics of words. Besides, to combine deep learning with the linguistic structures, many syntax-based embedding algorithms have been proposed to utilize the long-distance dependencies. However, short texts usually do not observe the syntax of a written language, nor do not contain enough signals for statistical inference (e.g., topic modeling). Hence, neither parsing nor topic modeling works well because there are simply not enough signals in the input and more semantic signals from the input are derived , e.g., the concepts, which has been in contestable effective in data illustration [3], [7],[8], [17]. They used short text conceptualization algorithmic rule to discriminate the concept level short text senses. Recently, the eye model has been wont to improve several neural NLP studies by selection that specializes in elements of the supply knowledge [23], [24]. during the study, the eye model was used to different influence values to different contextual words.

2.3 Knowledge Base

2.3.1 Probase

Established by Microsoft, they extracted 326,110,911 sentences from a corpus containing 1,679,189,480 web pages, after sentence deduplication. To the best of their knowledge, the scale of the corpus is larger than the antecendently known largest corpus. They then extracted 143,328,997 *isA* pairs from these sentences with 9,171,015 distinct super-concept labels and 11,256,733 distinct sub-concept labels. The inferred taxonomy contains 2,653,872 distinct concepts (down from 9.17 million after the extraction phase), 16,218,369 distinct concept instance pairs and 4,539,176 distinct concept sub concept pairs (20,757,545 pairs in total). The number of concept labels decreases since they had changed all labels to lowercases and flatten the concepts with only one instance and referred them as instances.

Given that Probase has many more concepts than any other taxonomy, it became an affordable question to ask is how many of these concepts are *relevant*. So they analyzed Bing´s query log from a two-year period, sorted the queries in decreasing order of their *frequency* (i.e., the number of times they have issued through Bing),computed the number of relevant concepts in Probase and four different general purpose open-domain taxonomies WordNet, WikiTaxonomy, YAGO, and Freebase, with respect to the top 50 million queries (**Figure 1**).

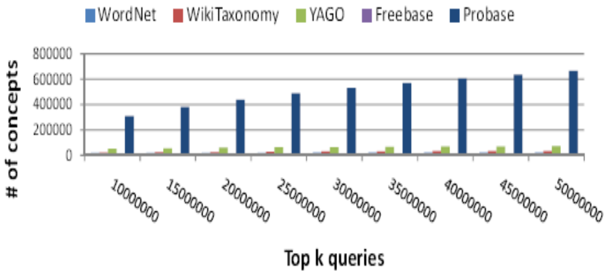


Fig 1 Number of relevant concepts in taxonomies

The framework suggested [15] focuses on understanding concepts. In several cases, semantics or linguistics is needed to supplement the syntax for correct extraction. Exploiting the knowledge to own higher understanding of the semantics, this adds to the power of extraction framework. They have specifically proposed an iterative learning process. In each round of information extraction, knowledge is accumulated for which a have high confidence to be correct is obtained. They then use this knowledge in the next round to help with extraction of data lost antecedently. This method is performed iteratively until no additional information can be extracted.

III SYSTEM ARCHITECURE

The proposed system (Figure 2) aims to achieve better understanding of short text by mapping it with knowledge base and then embedding the same. Two knowledge bases Probase and Concept Net which comprise of huge repository of words and related concepts are used. The short text mapped with the knowledge base is embedded using two popular approaches CBOW and Skipgram models of Word2vec. Figure 2 describes the overall system architecture of the proposed model. First the short texts are preprocessed by tokenizing and stemming. Then the preprocessed short text is conceptualized by mapping it with two Knowledge bases Probase and Concept Net. Next Semantic network is created to derive relationship between (word, word), (concept, concept) and (word, concept) from which affinity matrix is computed. The word and associated concepts are sorted based on their Co-ranks. At last, CBOW and Skipgram models are trained with words and their related concepts of each short text.

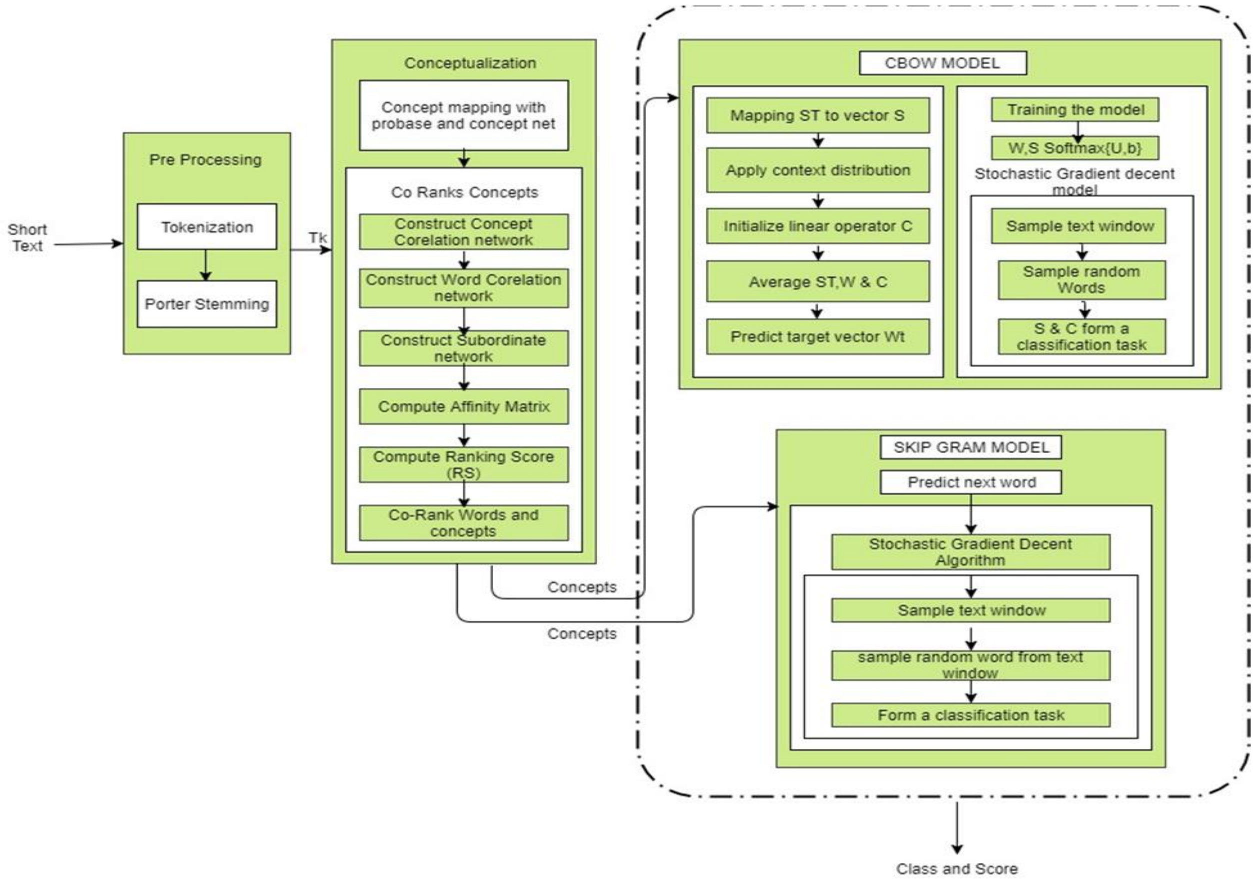


Fig 2 System Architecture

3.1 Preprocessing

In data preprocessing, the short text is tokenized, stop words are removed using a text file containing stop words and punctuation marks are removed .Finally porter stemming is applied.

3.2 Conceptualization

Tokens (words) are given as an input resulting in co ranked words and concepts. The words are mapped with knowledge base and the following steps are performed to achieve conceptualization.

SHORT TEXT CONCEPTUALIZATION (using PROBASE And Concept Net) :

Input: Stemmed Tokens$\{w_1, w_2 \ldots\ldots\ldots, w_{ST}\}$

Output: Concepts

CONCEPT MAPPING

For each word w_i that belong in the short text the word and concept pair is selected or vice versa from both the knowledge bases Probase and Concept Net, then the probability of their occurrence and the probability distribution θ_c are computed.

CO-RANK WORDS AND CONCEPTS:

Input: Words $\{w_1, w_2 \ldots\ldots\ldots, w_{ST}\}$

Concepts$\{<c_1, p_1>, <c_2, p_2> \ldots\ldots <c_c, p_c>\}$

Output: Co-Ranked words and concepts.

With words and concepts as inputs the w_i is mapped to c_i in Knowledge base for each Short text ST. Then a subset S and Co-rank w_i, c_i are created using θ_c, to enhance the semantic relationship of words. Semantic network that consists of three sub networks is constructed. They are , Concept co-relation network G_c, which has the (Concept, Concept) pair , Word co-relation network $G_w$which has the (Word, Word) pair and Word-Concept relation network G_{wc} that binds the above two network and has the (Word, Concept) pair or vice versa. Affinity Matrix(M) is computed form the parameters obtained from the semantic network and the iterative procedure is performed on G_c, G_w , G_{wc} The algorithm iteratively co-ranks the words and concepts or vice versa until the threshold is reached.

Followed by iterative procedure for co ranking

Input : M_{wc}; M_{cc} ; M_{cw}; M_{ww}; αwc; αcc ; αcw; αww;

Initialize

RSc(i) to 0,for (i= 1;...;n_c); // Ranking score vector of concept
RSw(j) to 1/n_w, for (j=1,..,n_w); // Ranking score vector of words
z=1;

ITERATIVE PROCEDURE:

while (diversity(RS_c^{z-1}, RS_c^{z})>ε) and (diversity (RS_w^{z-1} , RS_w^{z})) do //Threshold

Word Ranks Concept (RANKW->C); //αwc

Concept Ranks Concept (RANKC->C); //αcc

Concept Ranks Word(RANKC->W); //αcw

Word Ranks Word (RANKW->W); //αww

Compute RS_c^{z+1}

Compute RS_w^{z+1}

z=z+1;

end while

3.3 Short Text Embedding

CBOW Model:

In CBOW embedding module, given the words the target concept is predicted. The window size a is set based on the short text .

INPUT:

Words in contextual text window($w_{t-k,\ldots\ldots,}w_{t+k}$)

Short text ST, Short text matrix S // column of short text matrix S

Concept distribution θ_c , linear operator Ϲ

Word matrix W // column of word matrix W

OUTPUT: Conceptualized vector of ST

ALGORITHM

The output of the module 2 serves as the input for short text embedding module. Linear operator Ϲ =2k+1 is initialized for each word in the short text, which assigns a window size for each short text. Next concept distribution θ_c is converted to concept vector C and word w_i is mapped to unique vector w. Stochastic Gradient Decent algorithm is used to minimize the cost function and the Soft max function averages the short text and predicts the target Concept.

SKIP Gram Model:

The Skip gram model does the opposite of CBOW model where it predicts the target word given the related words.

Input: Short text ST

Concept vector C

Output: Short text embedding vector

Works similar to the CBOW model but here the window is set for the concepts and the target or the middle word is predicted by the Soft max function. Skip gram also makes use of the Stochastic Gradient decent algorithm for reducing the cost function.

Both embedding models make use of deep neural network.

IV. EXPERIMENTAL SETUP

4.1 Dataset

All Twitter dataset is constructed by manually labeling the previous dataset Tweet11. The dataset contains 12,378 tweets which are in four categories: food, sport, entertainment and device/IT company. The URLs and the stop-words are removed. The average length of these tweets is 13.16 words. Because of its noise and sparsity, this social media dataset is very challenging for the comparative models.

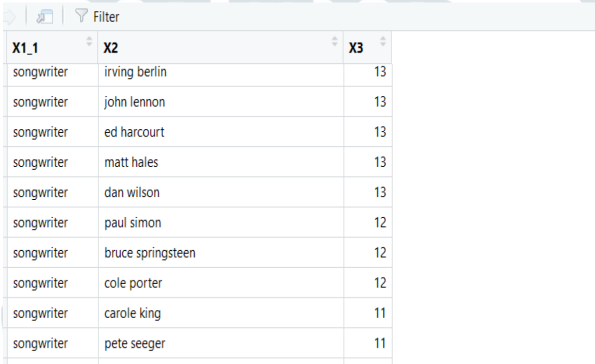
The goal of Probase is to enable machines to better understand human communication. For example, in natural language processing and speech analysis, knowledge bases can help reduce the ambiguities in language. As Probase has a knowledgebase as large as the concept space (of wordly facts) in a human mind, it has unique advantages in these applications, Besides, with the probabilistic knowledge provided by Probase, several interesting applications can be built such as topic search, web table search and document understanding.

Concept Net is a semantic network, designed to help computers understand the meanings of words that people use. Related concepts of word or phrase can be retrieved. Concept Net originated from the crowdsourcing project Open Mind Common Sense, was launched in 1999 at the MIT Media Lab. It has since grown to include knowledge from other crowd sourced resources, expert-created resources and games with a purpose.

4.2 Results

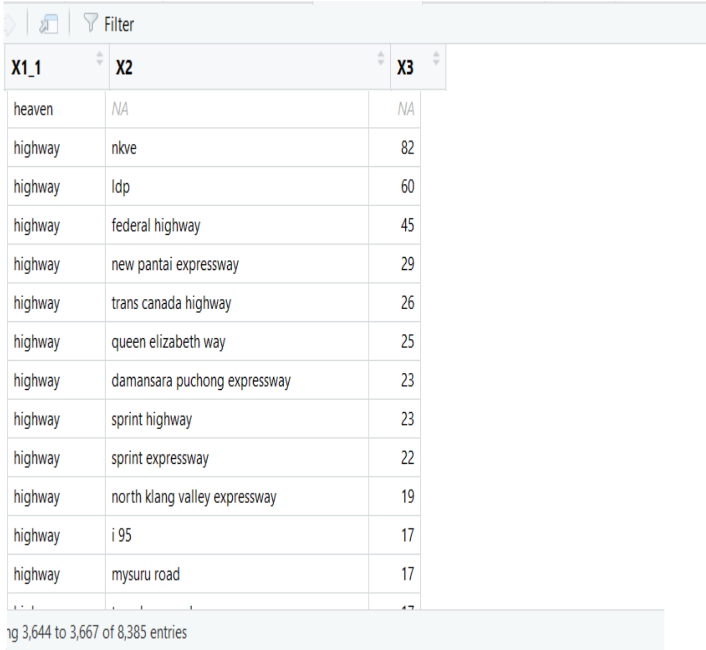
After tokenization the tokens of short text are further preprocessed by Porter stemming algorithm which removes common morphological endings from words in English and normalizes the text.

The preprocessed short text terms are mapped with their related concepts in the knowledge bases Probase (Figure 3a) and Concept Net (Figure 3b).


Filter

X1_1	X2	X3
songwriter	irving berlin	13
songwriter	john lennon	13
songwriter	ed harcourt	13
songwriter	matt hales	13
songwriter	dan wilson	13
songwriter	paul simon	12
songwriter	bruce springsteen	12
songwriter	cole porter	12
songwriter	carole king	11
songwriter	pete seeger	11

Fig 3a Concept Mapping(With Probase)



Filter

X1_1	X2	X3
heaven	NA	NA
highway	nkve	82
highway	ldp	60
highway	federal highway	45
highway	new pantai expressway	29
highway	trans canada highway	26
highway	queen elizabeth way	25
highway	damansara puchong expressway	23
highway	sprint highway	23
highway	sprint expressway	22
highway	north klang valley expressway	19
highway	i 95	17
highway	mysuru road	17

ng 3,644 to 3,667 of 8,385 entries

Fig 3b Concept mapping with (Concept Net)

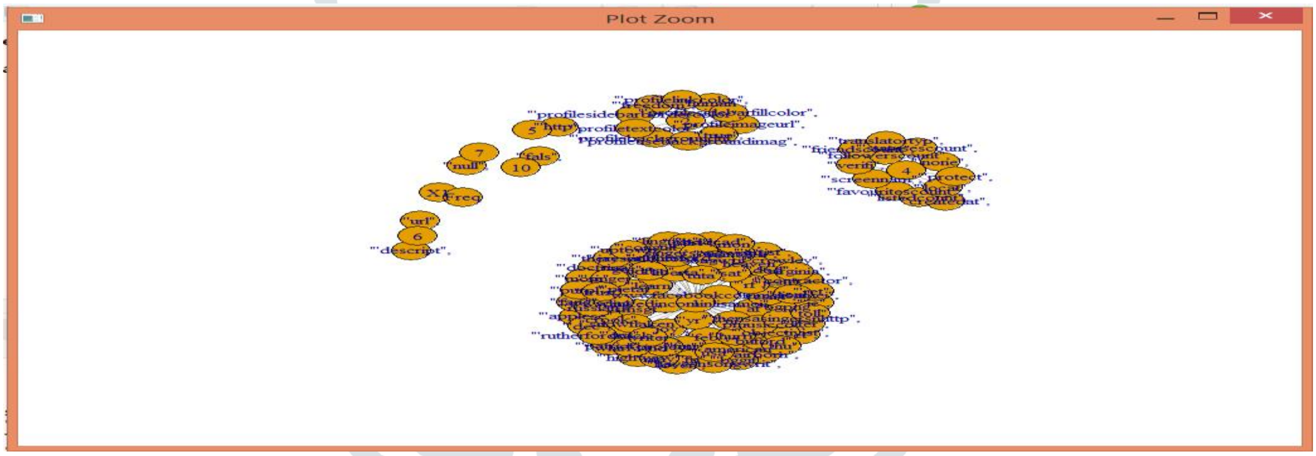


Fig 4 Semantic Network (Concept, Concept)

Semantic network or concept correlation network (Figure 4) is constructed with (Concept, Concept) pair to establish all the possible relationship of a concepts with other possible related concepts.

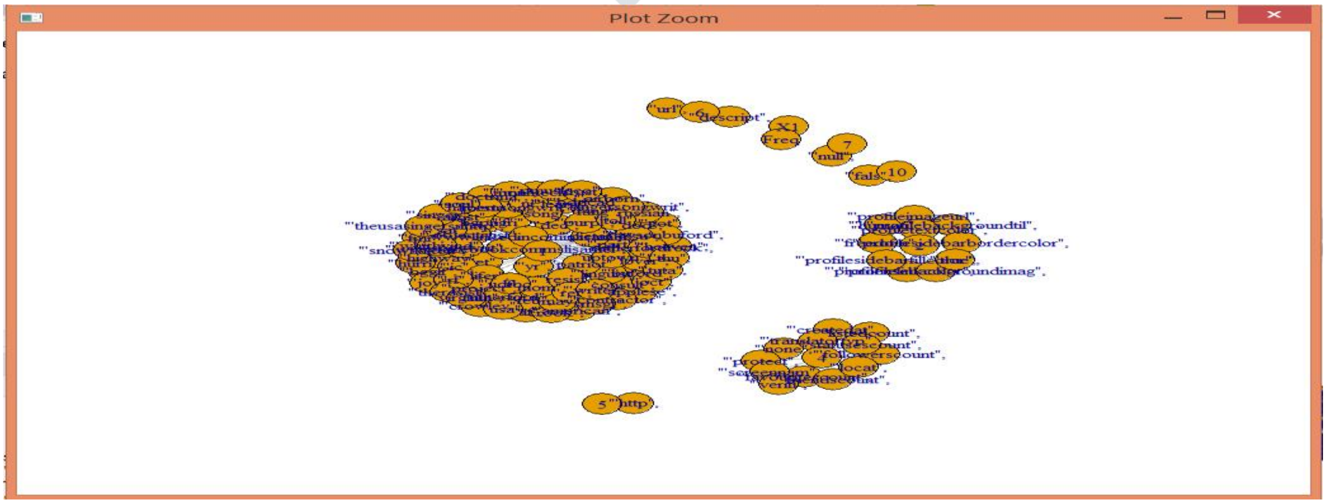


Fig 5 Semantic Network (Word, Word)

Semantic network or the word correlation network (Figure 5) is constructed with (Word, Word) pair to establish all the possible relationship of a word with other related words.

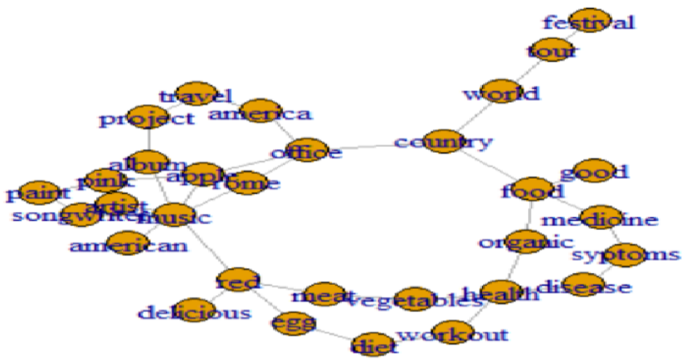


Fig 6 Semantic Network (Word, Concept)

Semantic network or the Sub ordinate network (Figure 6) is constructed with (Word, Concept) pair to establish relationship between all the words and their related concepts. All three networks listed above are used to improve the interaction of words and concepts leading to better understanding of the short text

Out[11]:

	X1_1	X2	X3	Rank_by_X1_1
3696	'hurri	NaN	NaN	1.0
3743	'mantech	NaN	NaN	2.0
3524	'png	NaN	NaN	3.0
6710	'snowflake	NaN	NaN	4.0
7369	'virginia	NaN	NaN	5.0
3567	FALSE	NaN	NaN	6.0
6594	Screen name	NaN	NaN	7.0
7354	TRUE	NaN	NaN	8.0
0	af	NaN	NaN	9.0
1	airborn	NaN	NaN	10.0
2	american	NaN	NaN	11.0
9	apple	golden delicious	31.0	20.0
8	apple	fuji	36.0	20.0
3	apple	gala	250.0	20.0
5	apple	red delicious	182.0	20.0
6	apple	granny smith	60.0	20.0
7	apple	cortland	38.0	20.0
19	apple	ida reds	12.0	20.0
4	apple	rome	218.0	20.0
17	apple	honeycrisp	13.0	20.0
16	apple	rome beauty	13.0	20.0
15	apple	empire	16.0	20.0
14	apple	jonagold	17.0	20.0
13	apple	pink lady	22.0	20.0
12	apple	jonathan	22.0	20.0
11	apple	braeburn	30.0	20.0
18	apple	granny smiths	13.0	20.0
10	apple	mcintosh	30.0	20.0
2359	artist	bobby blue bland	12.0	1779.5
2358	artist	champion jack dupree	12.0	1779.5

Fig 7 Co-ranking of Words and Concepts

Figure 7 shows the co-ranking of words and concepts that are related. The ranking is done by selecting concept for each word with highest ranking and selecting for each concept the word with highest ranking.

Console 1/A

```
   ...:     model1.similarity('apple', 'rome'))
   ...:
   ...: print("Cosine similarity between 'image' " +
   ...:           "and 'food' - CBOW : ",
   ...:       model1.similarity('image', 'food'))
   ...:
   ...: # Create Skip Gram model
   ...: model2 = gensim.models.Word2Vec(data, min_count = 1, size = 100,
   ...:                                          window = 5, sg = 1)
   ...:
   ...: # Print results
   ...: print("Cosine similarity between 'apple' " +
   ...:        "and 'rome' - Skip Gram : ",
   ...:     model2.similarity('apple', 'red'))
   ...:
   ...: print("Cosine similarity between 'image' " +
   ...:          "and 'food' - Skip Gram : ",
   ...:       model2.similarity('image', 'food'))
Cosine similarity between 'apple' and 'rome' - CBOW :  0.9629747
Cosine similarity between 'image' and 'food' - CBOW :  0.9981729
Cosine similarity between 'apple' and 'rome' - Skip Gram :  0.98349845
Cosine similarity between 'image' and 'food' - Skip Gram :  0.9696975

In [11]:
```

Fig 8 Short Text Embedding

Embedding of (Word, Concept) pair ('apple' , 'rome') and (Concept, Word) pair ('image', 'food') are computed and cosine similarity between the pair of word and concept is estimated. Without normalization, the similarity score for the short texts is computed by taking the average of vectors of words in the short text and then finding the cosine between them. But for conceptualized short texts the embedding vectors of related concepts from Probase and Concept Net too are considered for average vector computation.

4.2.1 Performance Measures - Conceptualization

Precision

$$\text{Precision} = \frac{true\ positive}{true\ positive + false\ positive} \quad (1)$$

Recall

$$\text{Recall} = \frac{true\ positive}{true\ positive + false\ negative} \quad (2)$$

F-measure

$$\text{F-measure} = 2 * \frac{Presicion * Recall}{Presicion + Recall} \quad (3)$$

Short text: I want a pet **cat** !

Table 1 Conceptualization Evaluation

		Precision	Recall	F measure
Withoutknowledge base		0.4705	0.4210	0.4443
WithKnowledge base	**Probase**	0.7692	0.8	0.7842
	Concept Net	0.75	0.6666	0.7058

Short text Embedding	CBOW model (Similarity score)	Skipgram model (Similarity score)

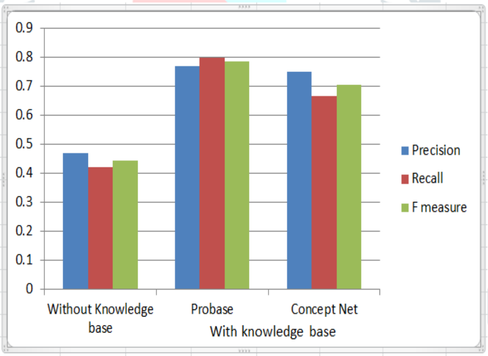


Fig 9 Performance Evaluation of Conceptualization

Figure 9 shows the evaluation of conceptualization of short text. From the graph it is understood that the Precision, Recall and F measure scores (Table1) are higher when conceptualization is done with Knowledge Bases Concept Net and Probase compared to the scores for without Knowledge Base. Among Knowledge Bases, the scores for Probase are higher compared to Concept Net.

4.2.2 Short Text Embedding Evaluation

Short text 1: The Rome apples are delicious and not too sweet.
Short text 2: I'm downloading images of food.
Preprocessed short text : ('Rome', 'apples', 'delicious', 'sweet'), ('download', 'image', 'food')

Without Conceptualization		0.2311192 0.4159336	0.2518073 0.4481765
With Conceptualization	**Probase**	**0.8629747**	**0.7834984**
	Concept Net	0.6081729	0.5696975

Table.2 Short Text Embedding Evaluation

For the evaluation of Short text embedding (Table 2) two short texts are considered and evaluated in two ways as short text embedding without any conceptualization and short text embedding with Conceptualization. When observed, the scores for embedding without Conceptualization is lower compared to embedding with conceptualization. Also the embedding Score is higher when probase is used compared to Concept Net. Thus making use of Probase for conceptualization improves the efficiency of short text embedding as well as short text similarity.

4.2.3 Test Cases

Short text 1: iTunes is Apple's music store.
Short text 2: Deep web is dangerous.
Short text 3: My friend is a total tech savvy!

Table 3 Test Cases

S.no	**Short text**	**Conceptualization**			
		Expected		**Actual**	
		Probase	**Concept Net**	**Probase**	**Concept Net**
1	**(Existing)** Apple, Music, Travel , Education, Vehicle.	90 %	85%	100%	100%
2	**(Existing , New)** Deep web, YOLO ("you only live once), Smart watch.	70%	65%	50%	50%
3	**(New)** Man's plain, Tech-savvy, Hench, ICYMI(" in case you missed it ")	30%	20%	0%	0%

For the test cases (Table 3) three short texts are considered that comprise of words that are very common in knowledge base, the words that are combination of old and new words and completely new words. There are two observations that can be seen from the test cases. Completely new words are rather difficult to find in the knowledge base compared to the other two cases and the use of Probase will give us higher percentage of words and concepts in comparison to Concept Net.

IV CONCLUSION

This paper leverages the conceptualization and embedding of short texts with Knowledge Bases Probase and Concept Net. Further, evaluation proves that embedding with conceptualization enhances the semantic relationship of words and concepts by improving the understanding of short text. Among the knowledge bases Probase and Concept Net, Probase gives more efficient result for embedding. This work can be extended for multilingual conceptualized short text embedding.

V REFERENCES

[1] X. Chen, X. Qiu, C. Zhu, S. Wu, and X. Huang, "Sentence modeling with gated recursive neural network," in Proc. Conf. Empirical Methods Natural Language Process., 2015, pp. 793–798.

[2] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in Proc. Int. Conf. Mach. Learn., 2016, pp. 2397–2406.

[3] Y. Wang, H. Huang, and C. Feng, "Query expansion based on a feedback concept model for microblog retrieval," in Proc. Int. Conf. World Wide Web, 2017, pp. 559–568.

[4] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," EprintArxiv, vol. 4, pp. 1188–1196, 2014.

[5] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical word embeddings," inProc.29thAAAIConf.Artif.Intell.,2015,pp.2418–2424.

[6] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "Towards universal paraphrastic sentence embeddings," CoRR, vol. abs/ 1511.08198, 2015.

[7] W. Hua, Z. Wang, H. Wang, K. Zheng, and X. Zhou, "Short text understanding through lexical-semantic analysis," in Proc. IEEE Int. Conf. Data Eng., 2015, pp. 495–506.

[8] F. Wang, Z. Wang, Z. Li, and J. R. Wen, "Concept-based short text classification and ranking," in Proc. ACM Int. Conf., 2014, pp. 1069–1078.

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533–536, 1986.

[10] Z. Yu, H. Wang, X. Lin, and M. Wang, "Understanding short texts through semantic enrichment and hashing," IEEE Trans. Knowl. Data Eng., vol. 28, no. 2, pp. 566–579, Feb. 2016.

[11] H. Aloulou, M. Mokhtari, T. Tiberghien, R. Endelin, and J. Biswas, "Uncertainty handling in semantic reasoning for accurate context understanding," Knowl.-Based Syst., vol. 77, pp. 16–28, 2015.

[12] W. Hua, Y. Song, H. Wang, and X. Zhou, "Identifying users' topical tasks in web search," in Proc. 6th ACM Int. Conf. Web Search Data Mining, 2013, pp. 93–102.

[13] P. Li, H. Wang, K. Q. Zhu, Z. Wang, and X. Wu, "Computing term similarity by large probabilistic isA knowledge," in Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage., 2013, pp. 1401–1410.

[14] R. Agrawal, S. Gollapudi, A. Kannan, and K. Kenthapadi, "Similarity search using concept graphs," in Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage., 2014, pp. 719–728.

[15] W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probase: A probabilistic taxonomy for text understanding," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2012, pp. 481–492.

1294 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 30, NO. 7, JULY 2018

[16]E.Gabrilovich, "Computingsemantic relatednessusingWikipediabased explicit semantic analysis," in Proc. 20th Int. Joint Conf. Artif. Intell.,2007,pp.1606–1611.

[17] J. Wang, Z. Wang, D. Zhang, and J. Yan, "Combining knowledge with deep convolutional neural networks for short text classification,"inProc.26thInt.JointConf.Artif.Intell.,2017,pp.2915–2921.

[18] Y. Song and R. Dan, "Unsupervised sparse vector densification for short text similarity," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Language Technol., 2015, pp. 1275–1280.

`[19] C. Shi, Y. Li, J. Zhang, Y. Sun, and P. S. Yu, "A survey of heterogeneous information network analysis," IEEE Trans. Knowl. Data Eng., vol. 29, no. 1, pp. 17–37, Jan. 2017.

[20] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in Proc. 31st Int. Conf. Int. Conf. Mach. Learn., 2014, pp. 1188–1196.

[21] H. Palangi, et al., "Deep sentence embedding using long shortterm memory networks: Analysis and application to information retrieval," IEEE/ACM Trans. Audio Speech Language Process., vol. 24, no. 4, pp. 694–707, Apr. 2016.

[22]Y.Wang,H.Huang,C.Feng,Q.Zhou,J.GuandX.Gao,“CSE:Conceptual sentence embeddings based on attention model,” in Proc. 54thAnnu.Meet.Assoc.Comput.Linguistics,2016,pp.505–515.

[23] T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” in Proc. Conf. Empirical Methods Natural Language Process., 2015, pp. 1412–1421.

[24] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, “An unsupervised neural attention model for aspect extraction,” in Proc. 55th Annu. Meet. Assoc. Comput. Linguistics, 2017, pp. 388–397.

[25] D. Zhou, S. A. Orshanskiy, H. Zha, and C. L. Giles, “Co-ranking authors and documents in a heterogeneous network,” in Proc. 7th IEEE Int. Conf. Data Mining, 2007, pp. 739–744.

[26]Y.Wang, H.Huang,C.FengandX.Yang,“Aco-rankingframework to select optimal seed set for influence maximization in heterogeneousnetwork,”inProc.Asia-PacificWebConf.,2015,pp.141–153.

[27] Y. Ding, “Applying weighted pagerank to author citation networks,”J.Assoc.Inf.Sci.Technol.,vol.62,no.2,pp.236–245,2011.

[28] P. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, “Learning deep structured semantic models for web search using clickthrough data,” in Proc. ACM Int. Conf. Conf. Inf. Knowl. Manage., 2013, pp. 2333–2338.

[29] L. Wang, et al., “Not all contexts are created equal: Better word representations with variable attention,” in Proc. Conf. Empirical Methods Natural Language Process., 2015, pp. 1367–1372.

[30] X. Li and D. Roth, “Learning question classifiers,” in Proc. 19th Int. Conf. Comput. Linguistics, 2002, pp. 1–7.

[31] R. Fan, K. Chang, C. J. Hsieh, X. Wang, and C. J. Lin, “LIBLINEAR: A library for large linear classification,” J. Mach. Learn. Res., vol. 9, no. 12, pp. 1871–1874, 2010.