iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** V **Month of publication:** May 2018

DOI: http://doi.org/10.22214/ijraset.2018.5081

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue V, May 2018- Available at www.ijraset.com

Concept Interpretation by Semantic Knowledge Harvesting

Jeena Sara Viju[1], Sruthy S[2]
[1]*PG Scholar, Computer Science and Engineering, Sree Buddha College of Engineering, Kerala, India*
[2]*Assistant Professor, Computer Science and Engineering, Sree Buddha College of Engineering, Kerala, India*

Abstract: ***Text with limited context is referred as short text. Short text understanding means detecting the concepts mentioned in a short text. Understanding short texts is important to many applications, but challenges abound. The reasons are, firstly, short texts do not always observe the syntax of a written language. Therefore, traditional natural language processing tools cannot be easily implemented. Secondly, short texts usually do not contain sufficient statistical signals to support many state-of-the-art approaches for text mining such as topic modeling. Thirdly, short texts are generated in a huge volume, which increases the difficulty to handle them. The main goal of the paper is to explore the semantics from short text and utilize it for proper decision making. A community blog is implemented in which the registered users in the community can text and the concept of the short text makes it possible to cluster the users. Online and Offline processing is performed. The instance ambiguity scoring and locating substrings in a text which are similar to terms contained in a predefined vocabulary in the offline processing increase the accuracy of the proposed system.***

Keywords: ***Shorttext, cooccurence network, termgraph, indexing, concept labeling, ambiguity.***

I. INTRODUCTION

The process of extracting information from large sets of data is termed as data mining. In the case of huge data sets, it is the computing process. Data mining is an essential process in which intelligent methods are used to extract data patterns. Data mining being an interdisciplinary subfield of computer science is used in various situations. The main objective of the data mining process is to obtain information from large data sets and transform the data into an understandable format in order to use it efficiently in future. Apart from the basic analysis step, it also involves database and data management aspects, data pre-processing, model and the inference considerations, etc. In databases process, or KDD, data mining is the analysis step of knowledge discovery. Text mining, which is also referred as text data mining, is similar to text analytics. It is the process of obtaining high-quality information from text. High-quality information is obtained through the devising of patterns and trends through means such as statistical pattern learning. Text mining mainly consists of the process of framing the input text usually parsing, along with the addition of some derived linguistic features. It also consists of removal of others, and subsequent insertion into a database, deriving patterns within the structured data, and finally evaluation and interpretation of the output.

The term 'High quality' used in text mining usually refers to certain combination of relevance, novelty, and interestingness. Text categorization, text clustering, concept or entity extraction, production of granular taxonomies, etc are certain text mining tasks. The text with limited context is termed as short text. Each day billions of short texts are generated, which will take the form of search queries, ad keywords, tags, tweets, messenger conversations, social network posts, etc. Short texts have some certain properties and characteristics which make it different from normal documents. Understanding short text means deriving the concept hidden in the short text. Even though there are many challenges while handling short text, the short text understanding is important to many of the applications. The main three reasons which is considered as a problem for short text understanding are as follows. First, short text does not follow or obey the syntax of any written language.. Second, sufficient statistical signals in order to support many state-of-the-art approaches are not contained in short text. Third, since short texts are generated in an enormous volume, they are more ambiguous and noisy. This further causes problem in handling short text. To understand short text, Semantic knowledge is essential. Some of the challenges of understanding short text are mentioned below.

A. Challenge 1 (Ambiguous Segmentation)

Consider the two short texts, “april in paris lyrics” versus “vacation april in paris”. A vocabulary contains both a term and its sub-terms which lead to multiple possible segmentations for a given short text. Semantic coherence should be maintained for a valid segmentation. For example, two possible segmentations can be derived from the short text “april in paris lyrics”, namely {april in

paris lyrics} and {april paris lyrics}. The first segmentation is better one because the word "lyrics" is more semantically related to songs ("april in paris") than months ("april") or cities ("paris").

B. Challenge 2 (Noisy Short Text)

Consider the three short texts, "new york city" versus "nyc" versus "big apple". It is essential to find out all the candidate terms for finding the semantic coherence in order to find the best segmentation for a particular text. This can be easily performed by building a hash index on the entire vocabulary. However, short texts are usually informal, full of abbreviations, etc. For example, in the above case "new york city" is usually abbreviated to "nyc" and known as "big apple". So it is important to find as much information possible about abbreviations and nicknames. Meanwhile, to handle misspellings occurred in short texts, approximate term extraction is required.

C. Challenge 3 (Ambiguous Type

Consider the two short texts "pink [singer] songs" versus "pink [adj] shoes". A word can belong to several types, and its best type in a short text depends on context semantics. For example, in the first short text, "pink" in "pink songs" refers to a famous singer and so it should be labelled as an instance, whereas in the second short text pink describes the color of the shoes and is therefore an adjective. Consider "pink songs" as an example. The probability of "pink" as an adjective and the probability of an adjective preceding a noun are relatively high, therefore traditional POS taggers will mistakenly label "pink" in "pink songs" as an adjective.

D. Challenge 4 (Ambiguous Instance)

In the three cases, "read harry potter [book] versus "watch harry potter [movie] versus "age harry potter [character]". The instance (e.g., "harry potter") can belong to multiple concepts (e.g., book, movie, character, etc.). When the context varies, these similar instances might refer to different concepts.

E. Challenge 5 (Enormous Volume)

Short texts are generated in a much larger volume when compared with normal document. Google being the most widely used search engine received over 3 billion search queries daily in 2014. Twitter reported in 2012 that it attracted more than 100 million users who posted 340 million tweets per day. Therefore, a feasible framework should be handled in real time for understanding short texts.

II. PROBLEM DEFINITION

Text segmentation, Type Detection and Concept Labelling which are the three steps for short text understanding sound quite simple, but challenges still abound. In order to face the main challenges which are being already discussed new approaches must be introduced to handle them. Given some short text, firstly text segmentation should be performed based on the semantic coherence. The best segmentation should be found out. Type Detection and Concept Label in order to understand and interpret the short text should be performed.

III.SHORT TEXT UNDERSTANDING APPROACHES

Various methods are used for understanding short text. Some of the methods are discussed below.

Transformation Based Error Driven Learning & Natural Language Processing

Linguistic information being encoded manually is challenged by automated corpus based learning. It is a method for providing a natural language processing system consisting of linguistic knowledge. Corpus based approaches have been successful and is used in many different areas of NLP, though these methods capture the linguistic information they are modelling indirectly in large opaque tables of statistics. All these make it difficult to analyze, understand and improve the ability of these approaches in order to model the underlying linguistic behaviour. In order to perform automated learning of linguistic knowledge, a simple rule-based approach is explained. The latest method for corpus based NLP called transformation based error driven learning is discussed. This algorithm was tested by applying to a number of language processing problems. Figure 1 describes the working of transformation based error driven learning. To start with firstly the unannotated text is passed through an initial state annotator. The ranges of complexity for the initial state annotator can vary from assigning random structure to assigning the output of a manually created sophisticated annotator. Various initial state annotators are used in parts of speech tagging. As indicated in the training corpus, it labels all the words with their most likely tag. Initial state annotations are explored for syntactic parsing. After passing the text

through the initial-state annotator, it is then compared to the truth. For reference of truth, a manually annotated corpus is used. To better resemble the truth, an ordered list of transformations is learned that can be applied to the output of the initial-state annotator. There are two components to a transformation: a rewrite rule and a triggering environment.

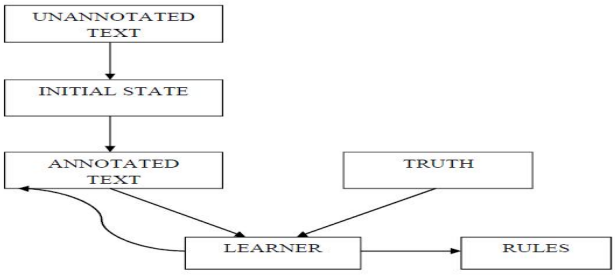


Fig-1: Transformation Based Error Driven Learning

A number of differences are present between transformation-based error driven learning and learning decision trees. One major difference between both is that during training a decision tree, the depth of the tree is increased every time, at the new depth, the average amount of training material available per node is halved (for a binary tree). While considering the case of transformation based learning, in order to find all transformations the entire training corpus is used. Transformations are being ordered, in a way that later transformations will be dependent on the result of applying earlier transformations. Therefore intermediate results help in classifying one object to be available in classifying other objects.

B. A Part-Of-Speech Tagger

A part-of-speech tagger which is based on the concept of hidden Markov model is discussed. It is a stochastic model and is used to design the randomly changing systems. In this particular case, it is assumed that the current state is responsible for the future states and not on the events that occurred previously. Most commonly, this prediction enables for reasoning and computation with the model that would otherwise be intractable. The Markov property is exhibited for this reason in the fields of predictive modelling and probabilistic forecasting. Hidden Markov Model (HMM) is a statistical Markov model and the system in which it is modelled is Markov process with neglected states. It is a generalization of a mixture model in which hidden variables that control the mixture component for each observation, are related through a Markov process rather than independent of each other. A type of Markov process consisting of either discrete state space or discrete index set is known as Markov chain. A Part-Of-Speech Tagger reads text and assign of parts of speech to each word, such as noun, verb, adjective, etc.

C. POS Tagging

Sometimes words can represent more than one part of speech at certain times and this makes the Part-of-speech tagging harder. Just having a list of words and their parts of speech is not sufficient. This condition is common because in natural languages a large percentage of words are ambiguous. For example, in the case of the sentence “the sailor dogs the hatch”, "dogs", which is usually thought as plural noun, can also be a verb.

D. Computing Term Similarity By Large Probabilistic

While considering a set of documents or terms, the idea of distance between them means the likeliness of the meaning or semantic content which is termed as semantic similarity. These are mathematical tools which used to compute the strength of the semantic relationship between words, concepts or instances. Semantic relatedness is different from semantic similarity. Semantic relatedness means any relation between two terms, while semantic similarity only includes "is a" relations. In the case of text analyses, semantic relatedness can be estimated by a vector space model to correlate words and textual contexts from a suitable text corpus. The basic structure for finding semantic similarity between two terms is discussed. Given a pair of terms <t1, t2>, first determine the type of the terms, T (t1) and T (t2), and calculate the similarity between the two contexts.

$$SIM(t1, t2) = sim(T(t1), T(t2))$$

where sim (c1, c2) is a similarity function for contexts.

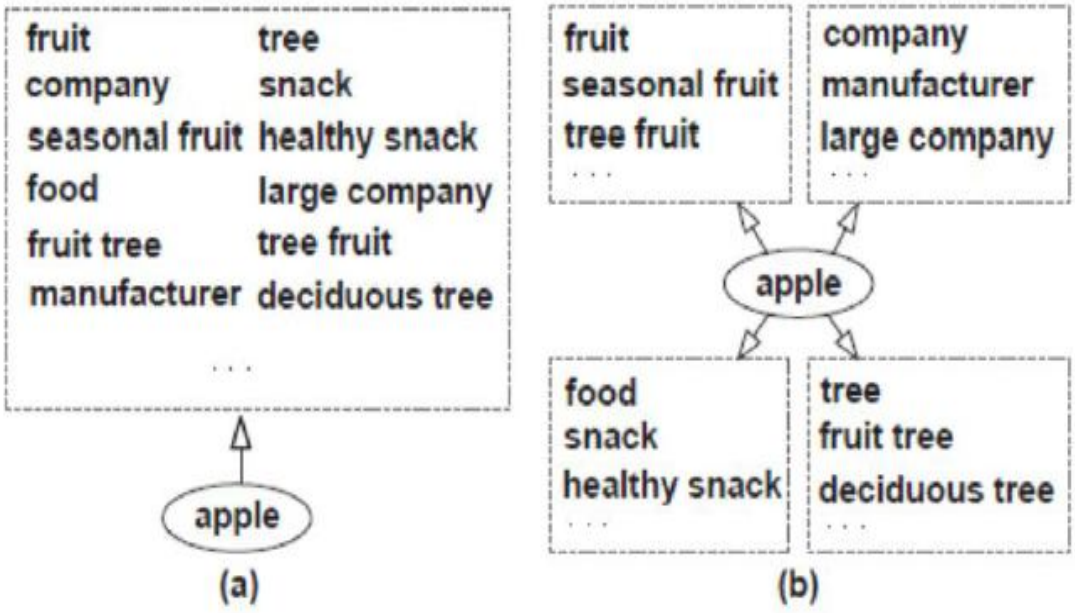


Fig-2: The concept context of "apple"

E. An Efficient Trie-Based Method For Approximate Entity Extraction With Edit Distance Constraints

Entity extraction (also known as entity recognition and entity identification) is an important operation in information extraction that locates substrings from a document into predefined entities, such as person names, locations, organizations, etc. Dictionary based entity extraction has attracted much attention from the database community, which identifies substrings from a document that match the predefined entities in a given dictionary. To quantify the similarity between two strings, many similarity functions have been proposed. Edit distance is a well-known function which is widely adopted for tolerating typing mistakes and spelling errors. The edit distance between two strings is the minimum number of single character edit operations (i.e., insertion, deletion, and substitution) needed to transform the first one to the second one. A partition scheme to partition entities into several segments is used. An efficient algorithm based on the fact that if a substring of the document is similar to an entity, the substring must contain a segment of the entity is developed. To facilitate the segment identification, use a trie structure to index the segments and develop an efficient triebased algorithm. An efficient extension-based framework to find similar pairs by extending the matching segments is proposed. A trie-based framework to find substrings from document D that approximately match entities is used. For each substring s of D, the trie structure to find its similar Entities is used. A naive method is to use every substring of s to search the trie structure. If a substring of s corresponds to a leaf node, the pair of s and every entity in the inverted list of the leaf node is a candidate pair. However this method is rather inefficient as s may have large numbers of substrings. To improve the performance, an alternative method is given. For each suffix of substring s, find the suffix in the trie structure. If a leaf node is reached, s has a substring corresponding to the leaf node. The entities are retrieved in the inverted list, which may be similar to substring s. Valid Substrings are some substrings of document D will not be similar to any entity. For instance, the substring "approximates membership" cannot be similar to any entity, as its length is too large. To address this problem, valid substrings are defined that are potentially similar to some entities. To avoid the duplicated computations on the shared segment across different substrings, a search-and-extension based algorithm is used. For each segment shared by multiple substrings, access the inverted list of the segment once. First use a SEARCH operation to locate the segment using the trie and then employ an EXTENSION operation to find similar entities.

IV. PROPOSED SYSTEM

Text segmentation, Type Detection and Concept Labelling which are the three steps for short text understanding sound quite simple, but challenges still abound. In order to face the main challenges which are being already discussed new approaches must be introduced to handle them. Given some short text, firstly text segmentation should be performed based on the semantic coherence. The best segmentation should be found out. Type Detection and Concept Label in order to understand and interpret the short text should be performed. The architecture of the proposed system is as shown in figure 1.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue V, May 2018- Available at www.ijraset.com

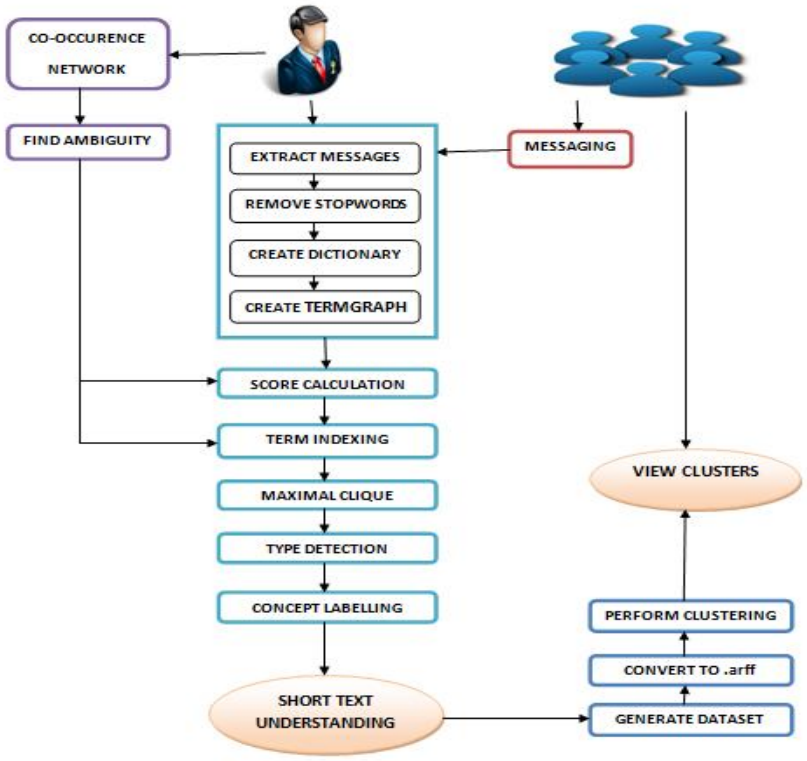


Fig. 3. Proposed System Architecture

The proposed system works in two modules – Admin module and User module. The users message each other in shorttext and the admin works on the messages and shorttext understanding is performed which determines the weight and ambiguity of the shorttexts. From the understanding, it is possible to cluster the users depending on the shorttexts. The admin performs offline and online processing. The various steps for shorttext understanding and clustering are explained.

A. Co-Occurrence Network

The co-occurrence network is an offline processing performed by admin module. Before moving to the online processing, it is important to construct the co-occurrence network. The co-occurrence network should consists of probably most of the words in English language as concept and its relatedness to other words or phrases which is taken by considering the data from web corpus. The network should therefore consist of concept, instance and the relation value. It is not easy to make such a network in a short interval of time. So, here the dataset from Probase is used. This data contains 5, 376,526 unique concepts, 12,501,527 unique instances, and 85,101,174 IsA relations.

B. Finding Ambiguity

It is essential to determine whether a concept is ambiguous or not and is part of offline processing. The easiest way to check ambiguity is to find out the number of instances which is related to a particular concept. Firstly read distinct concept and the corresponding instances. Then find the total sum of relations of the concepts chosen at each time. Read the relations corresponding to the concept and relations. Divide relations by sum and compute the ambiguity value at each time.

C. Preprocessing Shorttexts

The shorttexts messaged by the users need to be preprocessed in online processing. Firstly, the shorttexts will be extracted and the stop words will be removed from the extracted shorttexts. A set of stop words is managed and the admin as well can contribute new stop words to the set. From the shorttexts from which stop words are removed, a dictionary should be created. For that, each words from the shorttext should be extracted and using WordNet engine it should be found to which category it belongs (i.e., Noun, Verb, etc.). All the possible categories to which a word belongs should be found out and added to the dictionary. A term graph is created using Apriori algorithm. The Apriori algorithm is an influential algorithm for mining frequent itemsets for Boolean association rules. The weight corresponding to each frequent itemsets will be found out.

D. Score Calculation

Score in this proposed system refers to the relation value between concept and instance. From the shorttexts provided by the users it will check whether in there is any relation between the words in the shorttext or whether a relation value can be obtained.

E. Term Indexing

Term extraction aims to locate substrings in a text which are similar to terms contained in a predefined vocabulary. Firstly, find the n-gram of the messages and then the jaccard value. N-gram is a contiguous sequence of n items from a given sample of text or speech. It is The value for N is chosen as 3 (i.e., it's a trie gram). Then according to the proposed system the working of term indexing will be as follows:

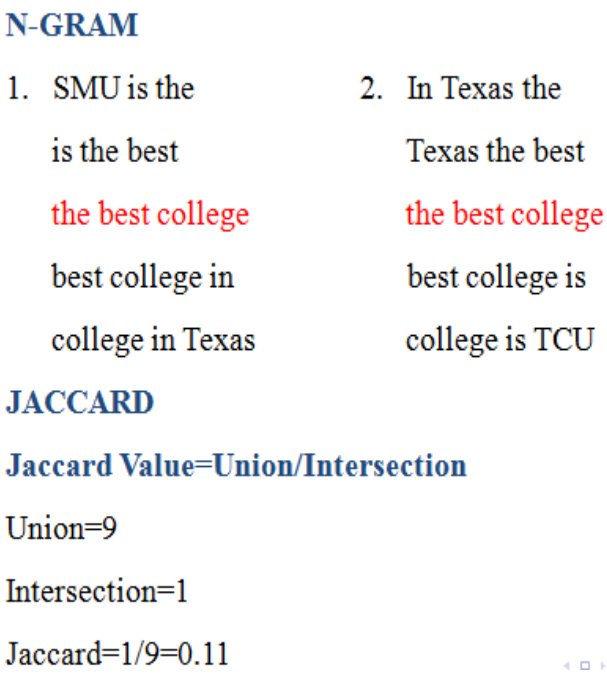


Fig.4. N-gram and Jaccard value

F. Maximal Clique

Text Segmentation is an important way to short text understanding. A maximal clique with the largest average edge weight from the original term graph is computed using Maximal Clique Algorithm by Monte Carlo is used. The algorithm works as follows : First, it randomly selects an edge with probability proportional to its weight. In other words, the larger the edge weight, the higher the probability to be selected. After picking an edge, it removes all nodes that are disconnected (namely mutually exclusive) with the picked nodes u or v. At the same time, it removes all edges that are linked to the deleted nodes. This process is repeated until no edges can be selected. After picking an edge, it removes all nodes that are disconnected (namely mutually exclusive) with the picked nodes u or v. At the same time, it removes all edges that are linked to the deleted nodes. This process is repeated until no edges can be selected. The obtained sub-graph is obviously a maximal clique of the original TG.

G. Type Detection

For each term derived from a short text, type detection determines the best typed-term from the set of possible typed-terms. Pair wise model for type detection is adopted by using a recursive function to find the paths from source to destination and each time the relation value will be found out. The best possible path from source to destination is obtained by seeing the relation value. One with greater relation value will be chosen as the best path.

H. Concept Labelling

In Concept Labelling an understanding of shorttext is performed. The weight and ambiguity is found out. This step makes it possible to understand the shorttexts in terms of weight and ambiguity. By understanding the different weights and ambiguity values, a threshold for both is set and for each shorttext it is found whether the shorttext is having high or low weight and high or low ambiguity. By performing this step a better understanding of the shorttexts will be provided.

Algorithm 1. Maximal Clique by Monte Carlo (MaxCMC)

Input:
$G = (V, E); W(E) = \{w(e)|e \in E\}$
Output:
$G' = (V', E'); s(G')$
$V' = \emptyset; E' = \emptyset$
while $E \neq \emptyset$ **do**
  randomly select $e = (u, v)$ from E with probability proportional to its weight
  $V' = V' \cup \{u, v\}; E' = E' \cup \{e\}$
  $V = V - \{u, v\}; E = E - \{e\}$
  for each $t \in V$ **do**
    if $e' = (u, t) \notin E$ or $e' = (v, t) \notin E$ **then**
      $V = V - \{t\}$
      remove edges linked to t from E: $E = E - \{e' = (t, *)\}$
    end if
  end for
end while
calculate average edge weight: $s(G') = \frac{\sum_{e \in E'} w(e)}{|E'|}$

Algorithm 2. Chunking by Maximal Clique (CMaxC)

Input:
$G = (V, E); W(E) = \{w(e)|e \in E\}$
number of times to run Algorithm 1: k
Output:
$G'_{best} = (V'_{best}, E'_{best})$
$s_{max} = 0$
for $i = 1; i \leq k; i++$ **do**
  run Algorithm 1 with ¡$G'_i = (V'_i, E'_i), s(G'_i)$¿ as output
  if $s(G'_i) > s_{max}$ **then**
    $G'_{best} = G'_i; s_{max} = s(G'_i)$
  end if
end for

I. Clustering Of Users

The users messaging similar shorttexts or shorttexts with similar meaning will be grouped together in this step. This can be done with the help of Weka Clustering. The Weka works on .arff file and therefore the input file which is the output obtained as shorttext understanding should be converted to .arff firstly. The clustering should be performed by using K-means algorithm. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

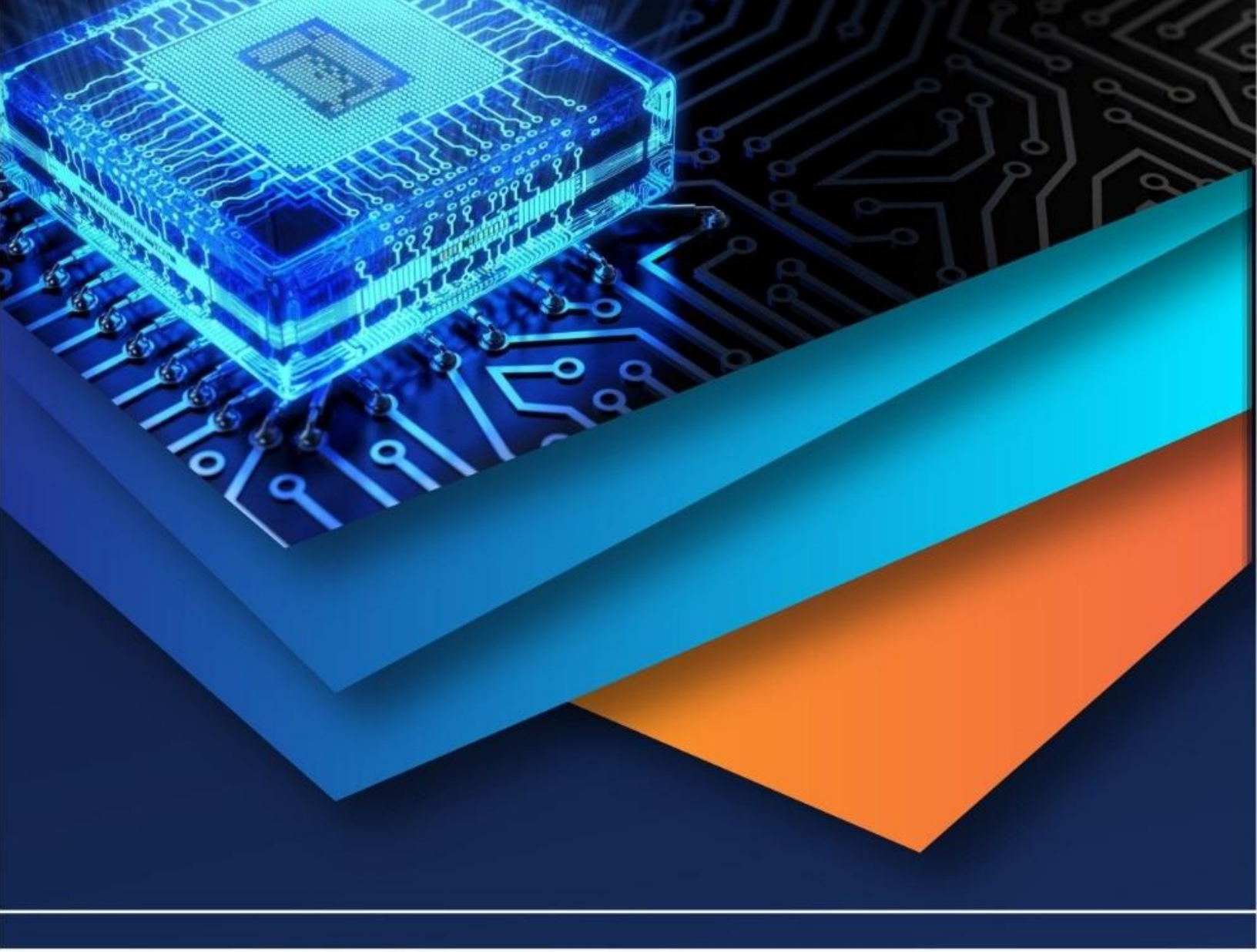
V. CONCLUSION

A community blog in which users can message in shorttexts and on the basis of the shorttexts the users will be grouped together is discussed. The shorttext understanding is performed by adopting the main three steps: text segmentation, type detection and concept labeling. The shorttexts understanding is done on the basis of weight and ambiguity. The users will be clustered together who message similar type of shorttexts.

REFERENCES

[1] E. Brill, “Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging,” Comput. Linguistics, vol. 21, no. 4, pp. 543–565, 1995
[2] E. Brill, “A simple rule-based part of speech tagger,” in Proc. Workshop Speech Natural Language, 1992, pp. 112–116.
[3] H. Schutze and Y. Singer, “Part-of-speech tagging using a variable memory Markov model,” in Proc. 32nd Annu. Meeting. Assoc. Comput. Linguistics, 1994, pp. 181–187
[4] X. Han, L. Sun, and J. Zhao, “Collective entity linking in web text: A graph-based method,” in Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval, 2011, pp. 765–774
[5] 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. Inform. #38; Knowl. Manage., 2013, pp. 1401–1410

[6] D. Deng, G. Li, and J. Feng, "An efficient trie-based method for approximate entity extraction with edit-distance constraints," in Proc. IEEE 28th Int. Conf. Data Eng., 2012, pp. 762–773
[7] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web enhanced lexicons," in Proc. 7th Conf. Natural Language Learn., 2003, pp. 188–191
[8] G. Zhou and J. Su, "Named entity recognition using an hmmbased chunk tagger," in Proc. 40th Annu. Meeting Assoc. Comput. Linguistics, 2002, pp. 473–480
[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," J. Mach. Learn. Res., vol. 3, pp. 993–1022, 2003
[10] M. Utiyama and H. Isahara, "A statistical model for domain-independent text segmentation," in Proc. 39th Annu. Meeting Assoc. Comput. Linguistics, 2001, pp. 499–506.
[11] B. Merialdo, "Tagging english text with a probabilistic model," Comput. Linguistics, vol. 20, no. 2, pp. 155–171, 1994.
[12] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen, "Short text conceptualization using a probabilistic knowledgebase," in Proc. 22nd Int. Joint Conf. Artif. Intell., 2011, pp. 2330– 2336.
[13] 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
[14] W. Wang, C. Xiao, X. Lin, and C. Zhang, "Efficient approximate entity extraction with edit distance constraints," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2009, pp. 759– 770
[15] R. Weischedel, R. Schwartz, J. Palmucci, M. Meteer, and L. Ramshaw, "Coping with ambiguity and unknown words through probabilistic models," Comput. Linguistics, vol. 19, no. 2, pp. 361–382, 1993.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY