Sādhanā (2020)45:29
https://doi.org/10.1007/s12046-019-1247-1

An algorithmic approach to rank the disambiguous entities in Twitter streams for effective semantic search operations

N SENTHIL KUMAR* and M DINAKARAN

School of Information Technology and Engineering, VIT University, Vellore 632014, India
e-mail: senthilkumar.n@vit.ac.in; dinakaran.m@vit.ac.in

MS received 5 January 2018; revised 17 February 2019; accepted 9 September 2019

Abstract. The most challenging task in any modern reasoning system is that it has been completely relied on automatic knowledge acquisition from the unstructured text and filtering out the structured information from it has turned out to be the most crucial task of Information Retrieval systems. In this paper, we have proposed a system that can recognize the potential named entities from the Twitter streams and link them to the appropriate real world knowledge entities. Besides, it has performed many semantic functions such as entity disambiguation, contextual similarity, type induction, and semantic labeling, to augment the semantic score of the entity and provide the rich entity feature space to quantitatively enhance entity retrieval accuracy. Nevertheless, we have leveraged a model to alleviate the entity imbalance present over the collected Twitter Streams and effectively utilized the contextual relatedness between the candidate entity sets. Eventually, we have proposed a probabilistic approach to deal with topic modeling and effectively disambiguate the entities by clustering the entities into its appropriate entity domain. The proposed Latent Dirichlet Allocation (LDA) model has been categorically distinguished the topics for clustering between the candidate entities and fix the exact true mentions occurred in the Knowledge Base such as DBpedia. We have also demonstrated the performance and accuracy rate of the proposed system and evaluated the results with the collected Twitter Streams for the month of August, 2016. The empirical results have shown that it has outperformed the existing state-of-the-art systems and proved that the proposed system given here has gradual accuracy rate against the conventional systems.

Keywords. Entity linking; entity disambiguation; Latent Dirichlet Allocation; semantic similarity; DBpedia ontology.

1. Introduction

Ever since the growth of social media has increased exponentially, the ambiguous nature of the text has become more complex and to derive the decision out of which is becoming more cumbersome. Hence, recognizing and linking the entities in the social media have gained the huge research attention and many of the researchers were implying to find out the atomic information in the social media content. Recently, linking the potential named entities in the social media text (such as twitter streams) to the external knowledge base (such as DBpedia, FreeBase, and YAGO) is becoming the booming research topic and many of the research initiatives such as CoNLL, SemEval, NEEL and TAC-KBP [1] had strenuously built the datasets to test, suit and enhance the entity detection and linking mechanism smoother. It is deemed important to mine the twitter streams to find the underlying thematic structure of the tweets and its relationship persists over the other tweets. By means of this, we can tap the latent structures of the real world entities and thereby facilitate to construct the semantic structures of the tweets which yield conceptual meaning of the twitter streams [2]. Further, it is paving way for inferring the conceptual grouping of the tweets and tends to mine fine-grained topical structure of the twitter streams and facilitate the way for effective searching and knowledge discovery of the Twitter streams. In this paper, we present the entity-concept discovery of Twitter content, topical hierarchy formation of the text and automated text representation. Besides, we have also illustrated the effective measures developed to handle the unstructured data and proposed the system to find the interesting information that can be discovered from the collected twitter streams.

The focal point of the research lies in the natural presumption of the chosen entity from the twitter streams which depends on its mentions, its surrounding contextual

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s12046-019-1247-1) contains supplementary material, which is available to authorized users.

*For correspondence
Published online: 24 January 2020

words and the prevalence of other entities referring in the accumulated twitter streams. To make the entity detection and disambiguation process much simpler, the authors in [3] have set the conditional probabilistic model which divides the entity into two segments: (i) find the similar candidate entities which is referring the tweets and its local contextual words and (ii) fetch the collaborative distribution of entities in the whole twitter streams. According to the paper [4], they have tentatively relied with the max product method to infer the entity sets and made the entity detection process easier. In order to empirically test the above stated cases, the corpus of entity-linked documents were downloaded from DBpedia Dump and provided the set of Wikipedia pages with its appropriate hyperlinks to probe for further details of the pages. By means of this loaded information, we can extract two sorts of inferences from the corpus. One, we can find how often the entity is referred in the corpus and its resemblance on the entire documents. Second, it is used to fetch the pairwise co-occurrence of entities in the corpus. According to the author [4], they argued that most of the related information of the entity was found in these count measures itself and proposed that it is sufficient to disambiguate the entity resemblance over the huge corpus of documents. But later, it has been evidently witnessed that it is not sufficient to tackle the huge entity detection problem and later the author [5] has proposed key measures to tackle the entity disambiguation problem with different perspective of handling. They had proposed the model which restricted its detection only on targeted entities and deduce the entities which goes out of the range or scope of domain. In many instances, they had tackled to handle the entities with the homogenous groups and disambiguate the extracted candidate mentions collectively from the documents. Perhaps, if the target entities are present in other ontologies such as FreeBase or YAGO by "is-a" property, we labeled them as homogenous if the least common ancestor (LCA) is deep from the root node. The surface forms of entities in the knowledge base are often ambiguous and give a different meaning to the context of the selected text. Identifying the true mentions of the entities (i.e., the exact fit of entity match in DBpedia) and disambiguating the entities according to the context of the situations is a seminal task of the research. Thus, the impeding task is to leverage the context similarity between the entities and choose the correct mention from the selected candidate sets.

2. Related works

According to the authors [6], the recent research has focused much on linking the entities into DBpedia mentions and named as topic annotation, entity disambiguation and entity linking. As discussed in the paper [6], there are two sources of entity linking mechanism followed in DBpedia: (i) Link Graph – which infers the semantic similarity between the entities and (ii) Anchor Text – which estimate the proximity of the entity to a given mention.

In 2011, Ritter *et al* [3] had proposed a Natural Language Processing (NLP) based system to carry out the efficient Part of Speech (POS) tagging, shallow parsing and identified the named entities in tweets using Conditional Random Field (CRF). This framework has opened a new avenue in the direction of finding the potential named entities in tweets and outperformed other standard Named Entity Recognition (NER) systems for Micro blogging systems. Later, in the same year 2011, Liu *et al* [7] had developed a system for Twitter Streams to find the potential named entity sets using K-Nearest Neighbors (KNN) classifier. Besides, his proposed system has carried out the additional task such as annotating the named entities in tweets.

The next major task of semantic web is to link the named entities into the predefined knowledge base such as DBpedia, YAGO, and Freebase. Later at the end of the year 2011, Mendes *et al* [8] had proposed a state-of-the-art framework to effectively deal the entity linking mechanism and make the entities rightly annotated with the appropriate real world entity. But earlier, Ferragina *et al* [9] 2010 had already deployed a system called TAGME which is performing the basic annotation of short texts and linking the entities into the appropriate mentions in the Wikipedia Anchor Texts. But it has encountered the problem that many of the referent entities did not occur in the chosen knowledge bases and return the NIL result. This was due to the effect that it has not checked the context of the texts and just carried the exact string match between the entity and mentions. In the next year 2012, Damljanovic *et al* [10] had developed a system called YODIE which is a yet another entity linking system that outperforms the earlier entity linking systems and it has performed the similarity between the strings, semantic similarity between entities and contextual similarity for the candidate entities. This amalgamation has made this system so popular and addressed many of the glitches faced in the entity linking system. But the major drawback of the system is twofold: one, it is too facing the same NIL results and second, it is time consuming.

Then later in 2012, Meij *et al* [11] had proposed a machine learning based system which has been dedicated to social media contents and extract the potential candidate entities from tweets using n-gram features and concept-based similarity measure. Then recently in 2014, Ibrahim *et al* [12] have given a framework called AIDA social which is centered around microblogs contents such as tweets and deployed the techniques such as normalizing the candidate mentions, contextual enrichment for the entities and temporal entity resemblance. Besides, we have listed down the standard NER tools followed to extract potential named entities from the text and given its key features in the table 1.

Table 1. Comparative analysis of major NER tools and its key features.

Tools	Proposed approach	Taxonomy followed	Domain selection	Programming languages	Licenses
ANNIE	Gazetteers and Finite State	MUC	News Content	JAVA (GATE)	GPLv3
Alchemy API	Machine Learning	Alchemy	Unstructured Text	Web Service	None
Lupedia	Gazetteers and Logic Rules	DBpedia	Unstructured Text	Web Service	None
Standard NER	Conditional Random Fields	CoNLL, ACE	Newswire	JAVA, Python	GPLv2
Ritter *et al* [3] Models	Conditional Random Fields	CoNLL, ACE	Twitter	Python	GPLv3
DBpedia Spotlight	Gazetteers and Similarity Metrics	DBpedia, Freebase, Schema.org	Semi Structured	Web Services, SPARQL	Apache 2.0
NERD-ML	KNN, Naïve Bayes	NERD	Twitter	Java, Python, Perl	GPLv3

3. Context similarity measure

According to the authors [13], the true mentions have been selected based on the similar contexts exists between multiple selected entities since all the entities have been fall into the same domain categories while the false mentions were directed to disparate domains and points to different entities. The objective here is that each mention should have associated with one real world entity. That is, there should not be any duplicates in the same source. However, if two mentions are identical and points to same real world entity, then it is called true mention. For instance, postal code and zip code are two different entities but represent the single real world entity called zonal code of any state in the country.

Now, let us take the simple mathematical notation to represent the above true mentions.

K: set of mentions in the Knowledge Base (e.g., DBpedia, Freebase, and YAGO)
H: set of relations/hyper-edges between any two mentions in KB
M: set of matches (i.e., two or more mentions which points to same entity)
N: set of non-matches (a single mention referring to different entities)
E: set of entities (selected from Twitter Streams)
L: set of links

In order to select the true mentions for the given entity sets, we have defined the condition as

$$True\ Mention = (M_{True}, N_{True}, E_{True}, L_{True})$$

where, we can obtain the true mentions of the entities only if all the properties (i.e., M, N, E, and L) are becoming true. If any single parameter became false, then the result would be inverse and completely contradictory. The decision rule to find the exact match among the available mentions has been formulated as:

Let us take K = (x, y) is an entity pair, φ is a comparison vector, M matches as given in the above notation and N non-matches.

$$Decision\ RuleR(x, y) = \frac{P(\emptyset | k \varepsilon M)}{P(\emptyset | k \varepsilon N)}$$

Here, 'k' points to the mentions which were extracted from DBpedia and checked them against the set of mentions referring to the same entity M and set of mentions individually referred N. The obtained result R is checked with the threshold value 't' which is given the value 0.75 and if $R > t$ and $R < 1$, then the entity pair (x, y) is point to the same entity. Otherwise, it is individually referred in the DBpedia.

It is assumed that the context similarity between any two mentions (x, y) in the knowledge base such as DBpedia can refer itself in any pair of occurrences in the Twitter Streams or any documents. Therefore, it is well suggested by [14] that the context similarity can be computed for all the pairs of occurrences and aggregate them for finding the true mentions out of it. That is, the context similarity between (m_i, k_j) and $(m_{i'}, k_{j'})$ can be the average similarity of every occurrence of m_i in k_j and $m_{i'}$ in $k_{j'}$. For every tweet, we evaluate the context similarity by taking the one word before and one word after the x-th occurrences of m_i in the knowledge base k_j. To measure the context similarity, we can also apply the TF-IDF cosine similarity to compare the two entities and find the true mention available in the DBpedia Knowledge Base. To utilize the cosine similarity, we should have normalize each vector and remove noisy words from it.

The other way of computing the context similarity is to use the DBpedia Ontology and compute the semantic similarity between the contexts exist in the Ontology. According to the author [14], the context similarity between two concepts can be determined by estimating the distance between two nodes in the ontological framework. It is defined as follows:

$$ContextSim\,(C1, C2) = \frac{1}{Distance(C1, C2)}$$

Here, the distance between the two nodes can be chosen based on the least number of intermediate nodes travelled between the two nodes represented in the concepts (C1, C2). That is, between the two concepts (C1, C2), which is referring in the ontology to link to the appropriate nodes (n1, n2) but the distance between these two nodes is at different levels of hierarchy in the ontological design. The context similarity is measured based on the least distance exist between the two nodes and calculated their rank score for finding the true mentions for the context and it has been depicted in figure 1. Besides, the following presumption has been taken into account for the appropriate selection of similarity score.

For example, if we want to find the semantic similarity score between the entities such as 'Singer' and 'Artist', the path for computing the similarity score would go around ('Music, Concert) and the weight of the nodes will be calculated for estimating the distance between the two entities. To compute the semantic distance between 'Singer' and 'Artist', we defined the formula as:

Semantic_Dist('Singer','Arist) = min{Semantic_Dist('Singer','Music')} + min{Semantic_Dist('Artist','Music')}.

Therefore, the Semantic_Distance('Singer','Artist') = min{(1.5,2)} + min{(2,1.5)} = {1.5 + 1.5} = 3. By applying the semantic similarity formula given above, we can get $\frac{1}{(1.5+1.5)} = 0.33$. Hence it implies that the semantic similarity between Singer and Artist is 0.33 and the proximity score is below the average.

The semantic similarity function is also measured by the following conditional probabilities and the threshold limit for the similarity score is ranged between 0 and 1 (i.e., $0 \leq sim \leq 1$).

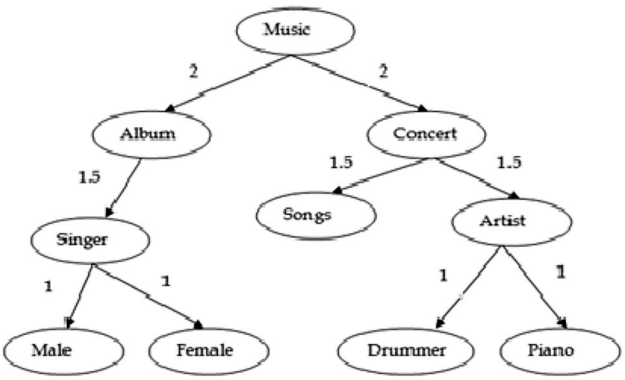

Figure 1. Ontological Hierarchical Tree for the concept mapping.

Property 1: The semantic similarity function is reflexive if the concepts C1 and C2 is identical to it and represented in the same ontology. (i.e., Sim(C1, C2) = 1).
Property 2: For any concepts C1, C2 and C3, if the semantic similarity of C1 and C2 is bigger than that of C1 and C3, then obviously the similarity score for C1 and C2 is lesser than that of C1 and C3. That is, If Sem_Sim(C1, C2) > Sem_Sim(C1, C3), then Sim(C1, C2) < Sim(C1, C3).

Presumption #1: If the semantic distance between two concepts is shorter in the given ontology hierarchy tree, the similarity score would be very high and the two concepts are very intact.

Presumption #2: Generally, the lower level nodes are semantically similar and represent the true real world entities rather than the higher level entities. In simple term, the higher level nodes describe about the generic domains and the lower level nodes truly represent the specific real world entities.

Presumption #3: The maximum similarity score between the concepts is reached only when the two concepts are conceptually identical or prevalently occurred in the same ontology.

The similarity score between the two entities that have been selected from the text is compared with the DBpedia Ontology and given the semantic similarity score which has the least common distance prevails among the other sets of nodes in the DBpedia ontology (see the below table 2).

For our case, as we given the twitter streams and identified the context similarity between the extracted named entities present in each and every tweets, the impeding task is to compute the context similarity between every entities and find the true mention of every potential entity referring in the tweet. In such cases, we both extracted the named entities and the context pinning in the tweet for processing and find the context similarity between the entities by applying the semantic similarity match. The semantic similarity measure is defined as:

$$SimMatch(C_iC_j) = \frac{M(C_i) \cap M(C_j)}{M(C_i) \cup M(C_j)}$$

Besides, the context similarity can be measured using min, max and median function.

Table 2. Entity Pair Occurrences and distance in the DBpedia ontology.

Concepts		
Entity 1	Entity 2	DBpedia Similarity Score
Singer	Artist	0.33
Concert	Music	0.50
Album	Piano	0.33

3.1 *Co-mention similarity measure*

The secondary part of the proposed approach in choosing the appropriate true mentions for the given ambiguous entities is co-mentions. If more number of targeted entities are co-mentioned in the Twitter streams and pervasively present in the tweets, then there would be a high chances of being the true mentions. But there is a slight distinction between the co-mention and co-references used in the documents. To elucidate the co-reference in the document as stated in the paper [15], we compute the vector matrix to count the number of co-referenced words occurred in the document and in this case, it need not be present in the existing knowledge base. Whereas in the case of co-mentions, every co-mention must have references (i.e., surface forms) in the knowledge base such as DBpedia and then only, we will evaluate the occurrence of co-mentions in the documents or any content. If we give the Twitter streams T_s, then the task is to compute the prior score for the potential candidate mentions which may be the co-mentions of other input entities and take the whole twitter streams as a text sliding window to effectively look for co-mentions present in the sliding window. We consider that the given tweets can be considered as a single document and searching for the co-mentioned entities were restricted within the document. There are two possibilities of occurrences of co-mentions in the document and it defines π_{ij}.

(a) Number of unique entities present in the Twitter StreamT_s.
(b) Number of similar named entities occurred in the context of Tweets m_i in k_j

When π_{ij} is determined from any one possibilities listed above and the normalized prior score 'p' can be obtained through the equation,

$$P_{ij} = \frac{\pi_{ij}}{P_{ij}} . \pi_i$$

Say for an example, in a tweet, there are three named entities identified.

Let us assume: Sachin Tendulkar, India, Cricket

Now the semantic similarity for the all the three named entities should be compared pair-wise. Like,

(Sachin Tendulkar, India) (Sachin Tendulkar, Cricket) (India, Cricket)

Similarly, every named entity should be linked to the knowledge base like DBpedia [15]. While trying to link the named entity into the DBpedia Knowledge Base, it will show many relevant mentions with similar names as given in table 3. In that case, the semantic similarity should be carry over with all other relevant mentions in the DBpedia and find the exact match of mentions to the given named entity. Using DBpedia Spotlight, we have executed the SPARQL query to obtain the concept and its DBpedia Label associated with every entity fetched by the query. Given the query search term "ACC", we have fetched the top 10 entity labels associated in DBpedia Spotlight and its relevant concepts. Entity Labeling will facilitate the entity annotation and made the entity disambiguation process easier. Given below the SPARQL query to fetch the results and portrayed in table 3.

```
SELECT DISTINCT ?term ?concept ?prefLabel
WHERE  {
  {
            ?concept skos:prefLabel ?term .
  }
 UNION
 {
   ?concept skos:altLabel ?term .
 }
UNION
{
   ?concept skos:hiddenLabel ?term .
 } .
 FILTER (regex(str(?term), "^acc", "i")) .
 ?concept skos:prefLabel ?prefLabel .
 FILTER (langMatches(lang(?prefLabel), "en")) .
}  ORDER BY ?term  LIMIT 10
```

3.2 *Propagation weight measure*

The general hypothesis about the propagation weight is that the group of similar true mentions can increase the ranking score of the entity and their edges have always pointed towards the true mentions. But, according to the author [16], sometimes, the group of similar false mentions will also provide the gradual ranking score and boost the probability of the ranking score. It is made possible when the targeted entity has contained in several ontological domain categories but with similar context. That is, although the false mentions of an entity can be similar with one another but it has been spread across to many heterogeneous domains sharing the same distinct entities. The propagation weight is a value assigned between the two vertices on the basis of its connection. Google used the PageRank algorithm to set the weight on the nodes based on its popularity. That is, how far the node has been referred by other nodes in the connected graph. Similarly, we have followed the propagation weight to know the popularity of the entity and obtained the score to evaluate the entity similarity. Hence, it was observed that a true mention can be relevant if the similar context of the mentions have been scattered across different entities of various domains rather than it is probed into the many mentions of the same

Table 3. Preferred DBpedia entity labels for the entity.

Term	Concept	Label
"ACC Asian XI One Day International cricketers"	http://dbpedia.org/resource/Category:ACC_Asian_XI_One_Day_International_cricketers	"ACC Asian XI One Day International cricketers"
"ACC Athlete of the Year"	http://dbpedia.org/resource/Category:ACC_Athlete_of_the_Year	"ACC Athlete of the Year"
"ACC Championship Game"	http://dbpedia.org/resource/Category:ACC_Championship_Game	"ACC Championship Game"
"ACC Men's Basketball Tournament"	http://dbpedia.org/resource/Category:ACC_Men's_Basketball_Tournament	"ACC Men's Basketball Tournament"
"ACC Men's Soccer Tournament"	http://dbpedia.org/resource/Category:ACC_Men's_Soccer_Tournament	"ACC Men's Soccer Tournament"
"ACC Trophy"	http://dbpedia.org/resource/Category:ACC_Trophy	"ACC Trophy"
"ACC Twenty20 Cup"	http://dbpedia.org/resource/Category:ACC_Twenty20_Cup	"ACC Twenty20 Cup"
"ACC Women's Basketball Tournament"	http://dbpedia.org/resource/Category:ACC_Women's_Basketball_Tournament	"ACC Women's Basketball Tournament"
"ACC Women's Soccer Tournament"	http://dbpedia.org/resource/Category:ACC_Women's_Soccer_Tournament	"ACC Women's Soccer Tournament"
"ACC articles by importance"	http://dbpedia.org/resource/Category:ACC_articles_by_importance	"ACC articles by importance"

domain like the example given above for zonal code. In order to limit the propagation weight to some level, we have used the following processes to effectively choose the true mentions of the context.

(i) Unlinking – Remove the propagation weight among the candidate mentions of the same real world entity.
(ii) Normalization – Curtail the link contribution of mentions on the entity.

3.3 *Mention rank*

The mention rank is totally different with the conventional PageRank algorithm proposed by Google [17] and it has followed the complete weighted and undirected graph for pruning the word analysis. Importantly, it has clustered the words which share the same mentions to point to the context and rank them according to their edges pointed to the other entities in the graph. In the PageRank algorithm [18], it refers to the inbound link calculation and calculates the probability of inbound and outbound link to the page. Whereas here in Mention Rank, the ranking score is calculated between the potential candidate mentions scattered across different entities. It just disallows the propagation weight between the mentions referring to the same entity. According to the authors [19], the normalization for the propagation weight is determined from a source node to a destined node. In Mention Rank, the propagation weight is normalized based on two strategic principles. One, number of true mentions referring to the source entity is determined as entity degree. Second, find the overall maximum contextual similarity exist between one mention and other mentions that has been scattered across different mentions.

4. Proposed system for entity disambiguation

In our proposed system, we have collected the twitter streams related to the particular event and extracted the potential named entities from the collection of streams. Then, we have given it as an input to the proposed system and determine the true mentions against the available ambiguous candidate named entities. The uniqueness of

this problem is that the entities are all represent the same event and referred to the single targeted domain. The true mentions are effectively determined by the proposed system that identifies the occurrences of candidate entities out of the given collection of twitter streams. In the given cases, an entity is appeared multiple times in the twitter collection, but the principle task underlying in the proposed approach is to identify that whether it are all referring to the same entity of same domain or not. The major problem in dealing with named entity conundrum is that the same entity can have different name variants like GM and General Motors. To fix the correct named entity which represents the true mention in the existing knowledge base such as DBpedia, Freebase, and YAGO, we have set the score between 0 and 1. Thereby, for each mention present in the KB, we indicate the likelihood of being the true mention representing the entity is given in figure 2. By means of this threshold limit, user can choose to set the cut-off value between precision and recall. Besides, it is used to extract the top-k mentions of the given twitter streams. The formal problem definition for our proposed system is defined as follows:

Definition 1 (*Targeted Entity Disambiguation*): Given the Twitter Streams T = {t1, t2, t3….. tn} for the event P, and extract the candidate entity sets from T as E = { e1,e2, e3, ….. en} and find the true mentions of the entity set given above as M = { *(ei, tj) | for each entity, fix the true mention in KB*}. We set the threshold limit to choose the true mentions for every named entity by giving the value between 0 and 1. We select the true mention of named entities by disambiguating the entities on the targeted domain but not at entity level. If two or more entities are identical and referring to the same mention in knowledge base KB, we then apply the standard NED techniques [20] to separate them into targeted domains of finer granularity.

To distinguish the ambiguous nature of the problem in our proposed definition, we have observed the following implications that affect the extraction of true mentions.

(i) The selection of true mention between the entities is similar within an entity and also with different entities across the document.
(ii) The context for false mentions is different with the results obtained and totally dissimilar with true mentions.
(iii) The context of false mentions can be similar within its group and domain but it is dissimilar with other entity sets.

To illustrate the situation, the entities such as ‘Apple’, ‘HP’, and ‘Sun’ have possess the same meaning representing the IT companies but the false mention of these entities represents different domains and lead to choose different mentions of its kind. It may also choose to select the fruit, electricity power and the planet correspondingly. Here, the context of the entities has been playing the vital role than identifying the related mentions directly from the DBpedia Knowledge Base.

Hypothesis 1 (*Context Relation Similarity*): The absolute relation exist among two or more true mentions is more similar than the two or more false mention which scattered across different entities.

Hypothesis 2 (*Co-Mention References*): If more entities are referred to the same document (i.e., Twitter Streams), then the probability of being the target domain entity is much higher.

Hypothesis 3 (*Cross-dependency*): If a mention in the KB has similar kind of relation with other true mentions, then it is likely to be a true mention.

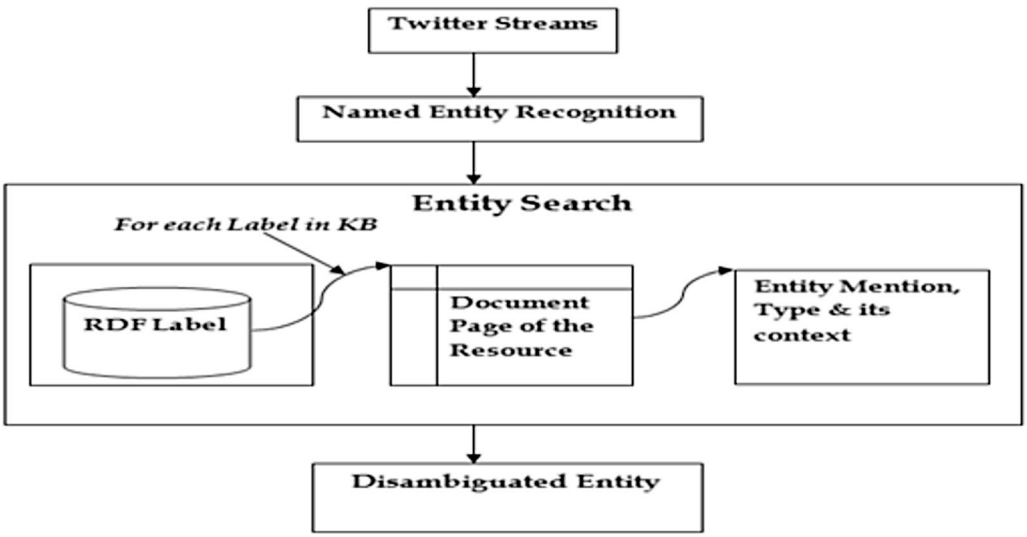

Figure 2. Proposed system for entity disambiguation and search.

4.1 *Leveraging the prior knowledge to the entity sets*

The Mention Rank model has not been utilized the existing knowledge bases to the entities other than the mentions. But in this section we are going to leverage the knowledge base entity analysis by scrutinizing the entity similarity, entity attribute selection and entity references.

4.1a *Entity Similarity*: According to the author [21], they had stated that if two or more entities belong to the same category or subcategories of the domains and possess the same properties, then they are more similar to one another. But the degree of ranking score of an entity could be lower if the interdependency between the two or more entities has not intact. Hence, the additional information about the entity such as numerical attributes or categorical attributes is extracted and performed the product context similarity as well as entity similarity to know the propagation weight as explained in the previous section. According to the multinomial distribution of word similarity [22], for each entity filtered from the tweet, we find the context similarity of the entity sets present in it. As per the multinomial distribution principle, for each entity, we may get more than two probable candidate entity sets and the chance of the true probable is to estimate the contexts based on the proximity of semantic resemblance between them. In this case, given the entity and its context, there were two sorts of hypothesis testing needed to carry out to get the similarity score of the entity sets.

H1: C(i) is a sequence of context entities representing the given entity E from tweet T.
H2: C(j) is a sequence of context entities representing the entities E' other than the E.

By these assumptions, we continue to evaluate the likelihood of the two hypotheses H1 and H2. Here, if $\Pr(C(i)|H1) > \Pr(C(j)|H2)$, then it says that the H1 context is more similar to the entity and relevance score is consistent. Otherwise, it is inconsistent. According to the multinomial distribution [23], if both the hypothesis are proportional to the probability and say that it is priori. It is defined as follows:

$$\Pr(C|H1) = Multinomial(\emptyset_E, \sum_i C_i) \propto \prod_i \emptyset_{E_i}^{C_i}$$

Assume that the corpus contains the contexts for the given entity, and it is denoted as, $E = \{(C_{11} \ldots C_{1v}), \ldots (C_{k1} \ldots C_{kv})\}$such a way that each concept inC_i, is a vector representation and estimates the $\emptyset_E$.

4.1b *Entity Attributes and Entity References*: In most of the cases [24], an entity is referenced with external knowledge bases such as DBpedia, Freebase, and YAGO. Actually, it is a database which stores the rich set of attributes about the entities and give the factual results about the entity whenever the user has asked for the information. For instance, if we take a search term 'product', it has some set of attributes like product name, manufacturer, design, logo, color, etc. and thus it is facilitating the mention in the Knowledge Base to directly link into the appropriate entity. This process of representing the attributes for entity disambiguation is termed as "representative documents". For linking the entity into its appropriate mention in the knowledge base, we have set of attribute selections in the KB and it is defined as a set $\{ei|1 < i < n\}$. We have taken the following example to choose the appropriate true mention from the candidate entity sets (table 4).

Example

"The Indian Superstar Kamal Hassan has been nominated for the prestigious French award "Chevalier" for the year 2016 who has given movies like Nayagan."

The entity detection process for the appropriate mention linking in the knowledge base can be carried out through the category and sub-category of the entities and construct the candidate set for the entity to disambiguate with the Jaccard Similarity score [25]. The algorithm for the effective comparison of candidate entity is given below for reference:

```
Algorithm 1: Rank the entities based on the category in KB
Input: Given the extracted entities from the tweets
Output: Rank the entities relevant to the context of the tweets.
Begin:
    For each entity in the tweet
        If candidate_set.size()==1 Then
            comparison_set.add(candidate.getAscendantCategory());
        Else
            Foreach c in entity.candidate_set
                c.category_set=c.getAscendantCategory();
        Endif
        c.category_rank=Jac(category_set, comparison_set);
    End loop
End
```

The above algorithm is measuring the category score of each candidate mention selected from Wikipedia data source and obtained the ascendant categories using the function getAscendantCategory(). The similarity score for the categories of each ascendant category in relation to the candidate entity using the Jaccard similarity measure [26] is given as follows:

Table 4. Entity Mapping into the respective Wikipedia links.

Entity	Detected concepts (as Wikipedia entities)
Nayagan	http://en.wikipedia.org/wiki/Film
Indian Superstar	http://en.wikipedia.org/wiki/Celebrity
Chevalior	http://en.wikipedia.org/wiki/Chevalior_Award
Kamal Hassan	http://en.wikipedia.org/wiki/Kamal_Hassan
French Award	http://en.wikipedia.org/wiki/Frech_Award

$$Jaccard(Comparison\_set, Categor\_set = \frac{|Comparison\_set \cap Category\_set|}{|Comparison\_set \cup Category\_set|}$$

If the similarity score is very high for a given candidate entity, then the chance of relationship with other related entity is increased and then performed similarity measure to check that it has mutually co-occurring with other entity sets. For each entity given, we find the related candidate entity sets from the knowledge source and rank them based on the context given in the tweet and twitter stream as a whole (see the below table 5).

That is, to the given tweet, we have extracted the potential named entities and for each entity in tweet, we find the corresponding mentions in the knowledge base and then compute the similarity score between the entities in the tweet to ensure the accuracy of candidate entities. The Word2Vector method [27] is followed for entity linking for each unique entity in the tweet and the vector of each entity is a semantic description of how each entity has been represented in the context. Suppose, if two entities in the tweets were semantically similar with one another, then the entity linking process would referred to the appropriate mention in the DBpedia knowledge source. In the Word2-Vec method, it combines all the tweets into a single document and trained the neural network to extract the vector representation of entity in the document. That is, map the entities to continuous vector representations in the N-dimensional vector space and learn the vectors from the trained neural network. The average entity vector for each tweet can be given as an input to the Artificial Neural Network (ANN) classifier and validated the model using Gensim word2vec implementation. After each entity is mapped into vector space, we have computed the cosine similarity score to find mention which has similar semantics in the KB. The semantic similarity score for the word2vector is given by cosine similarity of two entity vector. For each tweet in the twitter streams, the similarity score is calculated by measuring the vector similarity between entity and its candidate mentions.

$$CosineSim(entity, mentions) = sim\left(\frac{\sum_{entity}^{e} V_e}{|entity|}, \frac{\sum_{mention}^{m} V_m}{|mention|}\right)$$

where 'e' is an entity given in the input and 'm' is the candidate mentions from DBpedia source. It is illustrated in the following table 6:

Case 1: Considering the Named Entity itself

If we take the entity name alone to find the true mention in the knowledge base, in many instances, there were no surface forms of the mention present in the knowledge base. We measure the relative importance of Mention Rank by relying only with the knowledge base and its content match. Besides, the context in which the entity emerged out can be noted and computed the context similarity score to evaluate the Mention Rank of the true mention. Actually it is computing the average context similarity score of each candidate mention selected with other mentions and set the rank accordingly (see table 5). The Mention Rank, Context Match and Co-Mention Match are tends to observe the interdependency value by relatively propagating the ranking scores.

For evaluating the results, many authors have used precision and recall to calculate the score but in our case, we evaluated the performance with reference to the Naïve Bayes method and identified the conditional independence exists among the entities. Given the two sets, the candidate entities $E = \{e_i\}$ and its relevant candidate mentions from the DBpedia, $M = \{M_j\}$, each set consists of its own random candidate entities and the underlying task here is to measure the probabilistic association between E and M. The Naïve Bayes method for this evaluation is defined as follows:

Table 5. Candidate entity generation and ranking.

Seed entity	Candidate mention detection
Wapta Falls	Yoho National Park(1), Waterfalls of British Columbia(2), Kicking Horser River(3), WaterFall(4), Tourist Places(5)
President	Chief Executive Officier(1), Corporate Executive(2), Head of State(3), Position of Authority (4), Management Occupation(5)
Donald Trump	American Businessman(1), American President(2), American Billionaire(3), American Investor(4), American Restaurateurs(5)
Jaguar	Animal(1), Panthera(2), Keystone Species(3), Apex Predators(4), Field of Central America(5)
Bill Gates	American Billionaire(1), Director of Microsoft(2), Personal Computing (3), American Philanthropist (4), American Inventor(5)
Bank	Italian Invention(1), Legal entity(2), River Bank(3), Banking(4), Coastal(5)
Bus	French Invention(1), Bus Transport (2), Data Bus(3), Computer Terminology(4)

Table 6. Entity and mention cosine similarity using Word2Vec.

	DBpedia knowledge base						
Entity	Singer	Album	Music	Concert	Movie	Teacher	Composer
Songs	0.98	0.90	0.99	0.76	0.85	0.54	0.76
Actor	0.56	0.45	0.34	0.51	0.84	0.41	0.33
Drummer	0.77	0.89	0.91	0.82	0.65	0.45	0.65
Piano	0.79	0.71	0.86	0.85	0.55	0.92	0.84
Keyboard	0.68	0.83	0.67	0.78	0.50	0.93	0.97
Director	0.49	0.78	0.91	0.86	0.95	0.42	0.57
Lyric	0.87	0.89	0.76	0.43	0.67	0.39	0.48

Table 7. Lexical knowledge representation of entities.

Entity	Hypernyms	Hyponyms	Synonyms
Automobile	Motor Vehicle	Convertible	Car
	Locomotive	Electric Car	Auto
	Automotive Vehicle	Compact	Machine
	Four Wheels	Coupe	Motor Car
Web	Network	Spider Web	Webbing
	WWW	Net	Entanglement
	Internet	System	Computer Network
Antenna	Dipole	Electrical Device	Aerial
	Directional Antenna	Tentacle	Transmitting Aerial
	Non Directional	Sensitivity	Feeler
Protocol	HTTP	Code of Behavior	Communication
	FTP	Etiquette	Conditionals
	Transmission Control	Bittorrent	Rules
Information	Content	Details	Data
	Knowledge	Fact	Entropy
	Accumulation	Substance	Factoid

$$\Pr(M|E) = \prod_j \Pr(M_j|\{E_i\})$$

$$\Pr(M|E) = \prod_j \frac{\Pr(\{E_i\}|M_j)\Pr(M_j)}{\Pr(\{E_i\})}$$

In the context of word disambiguation, 'E' is a polysemous word (see table 7) and 'M' is representing some lexical knowledge in the predefined Knowledge Bases. Generally, $Pr(M|E)$ is measuring the relative association between the entity sets 'E' and Mentions 'M'. Related to the paper [28, 29], it has been observed that the actual estimation of $\Pr(E|M)$ is much easier than $\Pr(M|E)$ because the lexical knowledge has limited features of entities and many times, it is running out of vocabulary (OOV).

Case 2: Named Entity with set of attributes
If we provide both the entity and its attribute values to the knowledge base for the selection of appropriate true mention, then finding the true mention is made appropriate to large extent and obtained the satisfactory results. We have also performed the following comparison approaches to take over the above approach. The Content Match of the knowledge base is performed for the entity and its attributes and gets the average occurrence resemblance score for every mention that is yielding as the result. In addition to that, if we match for the Co-Mention for the same Content Match, it gives the linear combination of relevance score and paves the results to be more promising.

Case 3: Named Entity with concept alignment:
So far, we have been discussing about the named entity selection with its appropriate surface form or fetching the relevant attributes about the entities to identify the true mentions of the entities. But, as discussed in paper [30, 31], the concept vector representation of entities should be given more priorities and it almost gives the high relevance score compared to context based similarity measure and co-mention similarity occurrences. To compute the similarity between the concept vector manipulations, we must provide the actual correspondence between one concept vectors to another concept vector and align the concept vector to directly linking to the true mention occurred in the

Table 8. Semantic relatedness between the concepts.

Concepts	Scientist	Data mining	Semantic web
Data Analyst	0.62	0.67	0.74
Information Retrieval	0.34	0.56	0.69
Web of Data	0.33	0.49	0.87
Text Classification	0.31	0.72	0.68

Table 9. Twitter datasets and its topic extraction process.

Twitter streams collection	Size
Number of Raw Tweets collected	32,500
Number of Valid Tweets after pruning	20,750
Average Length of Valid Tweets (words)	4.25
Number of Valid Tweet Pairs	14,550

knowledge base. The formal definition given by the author [32] is that for each concept 'C' in an observation O_i, we choose the target concept to equally align 'C' with other concept O_j, Align(C, O_k), which will gradually increase the semantic relatedness between the two concept vector (see table 8). It is defined as follows:

$$Align(C, O_k) = \underset{C_i \in O_k}{\operatorname{argmax}} \ (C, C_i)$$

Note that the semantic relatedness between two concepts is always in range [0, 1] and the semantic similarity between the two name variants will also be in the same range [0, 1]. The value 0 indicates that the similarity between the two concepts is completely unrelated and it is absolutely a false mention. The value 1 denotes that it completely related and chance of being the true mention is highly recommended.

The average semantic relatedness between the concepts Ox and Oy is defined by the author [8] as follows:

$$SemRelatedness(O_x, O_y) = \frac{1}{2} X \left(SR(O_x \rightarrow O_y) + SR(O_y \rightarrow O_x) \right)$$

That is, we have given the semantic relatedness between the source concept vectors to target vector representation and calculated the average semantic relatedness between them.

4.2 *Empirical evaluation*

In order to streamline and validate the proposed Latent Dirichlet Allocation (LDA) model, we have collected Tweets for the period of one month and approximately, we have accumulated around 30 K of tweets. As the Tweets were possessed with noisy texts, we first shun out the low quality tweets by filter out the stop words, symbols and a words which has less than five occurrences (see table 9). We have just retained the tweets with more than 5 words after this pruning process and hold the high quality tweets around 20 K.

During the process of training the proposed LDA model, we have treated each tweet as a document and approximately about 50 topics chosen for the categorization of the topic model. Then we built the topic distribution matrix and computed the transition matrix as followed in [32] for the given datasets. Generally as discussed in [33], the LDA model yields two probability distribution for the effective categorization on topics. One, finding the probability distribution among the entities which points to the same topic and second, get the probability distribution for the topics in the entire documents. In our case, as we have taken twitter datasets, and considered the tweet as document, we find the probability distribution of every named entity present in the tweet. For instance, if we take the topic "Music", it will give some relevant terms related to it such as "Song", "Pop", "Classic", "Singer", and "Instruments" and the LDA model would find the probability distribution of these entities. In our proposed LDA model, we represented the collected twitter streams as combination of topics that further divided with words on certain probability. Then we are calculating the arguments for Φ and Θ which is followed in Gibbs Sampling [33].

*P(topic|doc) * P(word|topic) is the probability order denoting, doc → topic → word.*

Hence, successfully evaluating the Posterior distribution between topics and words, we obtained the probability distribution for topics and candidate entities respectively. This analysis steps has been shown illustriously in the below table 10.

Extracted entities (as given in table 3) were attributed to corresponding entity classes as given in the table 11 and compared the precision, recall and F1 score with different machine learning algorithms such as Support Vector Machine, Maximum Entropy Model and Hidden Markov Model. When compared with existing machine learning algorithms, our proposed LDA model has shown the accuracy rate much better than other models described. In this comparison, we have taken the entity class and its associated entities for identifying the exact match of entity-mention in the DBpedia Spotlight and filtered the candidate entities for disambiguation and ranking. Among all the three machine learning algorithms (Support Vector Machine, Maximum Entropy Model and Hidden Markov Model), our proposed model has shown the progressive results in finding the exact fit of entity referent in the DBpedia Spotlight. Using this proposed model, we have also the first to witness that the NIL referent entity problem has been solved and appropriate entity source for the candidate entity has been obtained by any of the three approaches described above.

Table 10. Topic Identification and categorization using proposed LDA model.

Domain/category	Topics selected
Disaster	Calamity, Cataclysm, Tsunami, Tidal Wave, Meltdown, Catastrophe, Tragedy, Apocalypse, Famine, Accident, Fire, Plague, Adversity, Misfortune, Destruction
Politics	Government, Profession, Social Relation, Ministers, Cabinet, Relation, Bureaucracy, Social Science, Activity, Geopolitics, Election, Assembly, Parliament
Technology	Invention, Technique, Development, Mobile Communication, Revolution, Innovation, Creativity, Scientist,
Health Care	Disease, Health Monitoring, Control, Prevention, Diagnosis, Information Technology, Assistance
Music	Songs, Album, Instruments, Sound, Vocal, Concert, Music Director, Composer, Musical Score

Table 11. Final result classifier for the proposed system.

	SVM			Maximum entropy			Hidden Markov model			Proposed model		
Entity class	Piec	Recall	Fl	Fiet	Recall	Fl	Piec	Recall	Fl	Free	Recall	PI
Country	0.811	0.782	0.796	0.755	0.621	0.681	0.891	0.787	0.835	0.897	0.789	0.839
Politics	0.891	0.713	0.792	0.824	0.728	0.773	0.823	0.762	0.791	0.879	0.741	0.804
Members	0.718	0.601	0.654	0.701	0.589	0.640	0.824	0.783	0.802	0.801	0.734	0.766
Campaisn	0.817	0.698	0.752	0.792	0.675	0.728	0.784	0.672	0.723	0.865	0.742	0.798
Prime Minister	0.876	0.716	0.787	0.895	0.769	0.827	0.834	0.745	0.786	0.871	0.756	0.309
Clean India	0.782	0.618	0.690	0.721	0.694	0.707	0.881	0.782	0.828	0.887	0.725	0.797
Environment	0.899	0.725	0.802	0.791	0.677	0.729	0.801	0.727	0.762	0.891	0.761	0.320
Mission	0.812	0.786	0.798	0.713	0.653	0.681	0.791	0.675	0.728	0.861	0.723	0.785
Organization	0.829	0.724	0.772	0.814	0.78	0.796	0.791	0.741	0.765	0.899	0.734	0.808
Volunteers	0.872	0.765	0.815	0.854	0.735	0.790	0.805	0.728	0.764	0.871	0.724	0.790
Youth Group	0.811	0.782	0.796	0.755	0.621	0.681	0.891	0.787	0.835	0.899	0.751	0.818
Program	0.891	0.713	0.792	0.824	0.728	0.773	0.823	0.762	0.791	0.879	0.709	0.784

The above table is more evident that the precision and recall of the proposed system grows consistently (i.e. a rate of 1% accuracy is increased on average for the selected entity classes) and the ranking of the overall approach of the enablement as given in the figure 3. Besides, it has also been observed that the proposed system has tackled the disambiguation problem strenuously and provide the disambiguated search environment for the potential users.

4.3 *Error analysis of entity detection and disambiguation*

From our error analysis, we have identified some errors at candidate entity detection and entity disambiguation. It has been noticed at some instances that the single token entities were very difficult to identify its original sense than the multi-token entities. In the analysis, we found that almost 63.18% of entities are single token entities and only 25.59% of entities are of multi-token entities. We observed two types of errors mostly occurring in the evaluation: One, unorthodox tokens or ill-formed tokens were find difficult to convert to its real base terms. Besides, for few abbreviation, our system did not yield the appropriate conversion and mostly incurred with dubious results. Even though, our proposed system has found the contextual information about the candidate entities, it was difficult to find the correct sense of the ambiguous entities. For example, the single-token entities such as May (Month/Surname), Washington (Location/Surname), Heathrow (Location/Organization), and few more has not been recognised by the system appropriately. Abbreviations such as GM, LOTR, ISN, and few more were not converted to its correct form by the system. Second, for the maximal sequence matching of multi-token entities, our system has followed mention ranking to propagate the weight of the entities and used the DBpedia ontology to rank the appropriate candidate entity for the surface form. For example, the multi-token entities such as Ham Sandwich (Person/Artefact), Tiger Wood (Player/Animal), Wicker Park (Movie/Park/Music) and

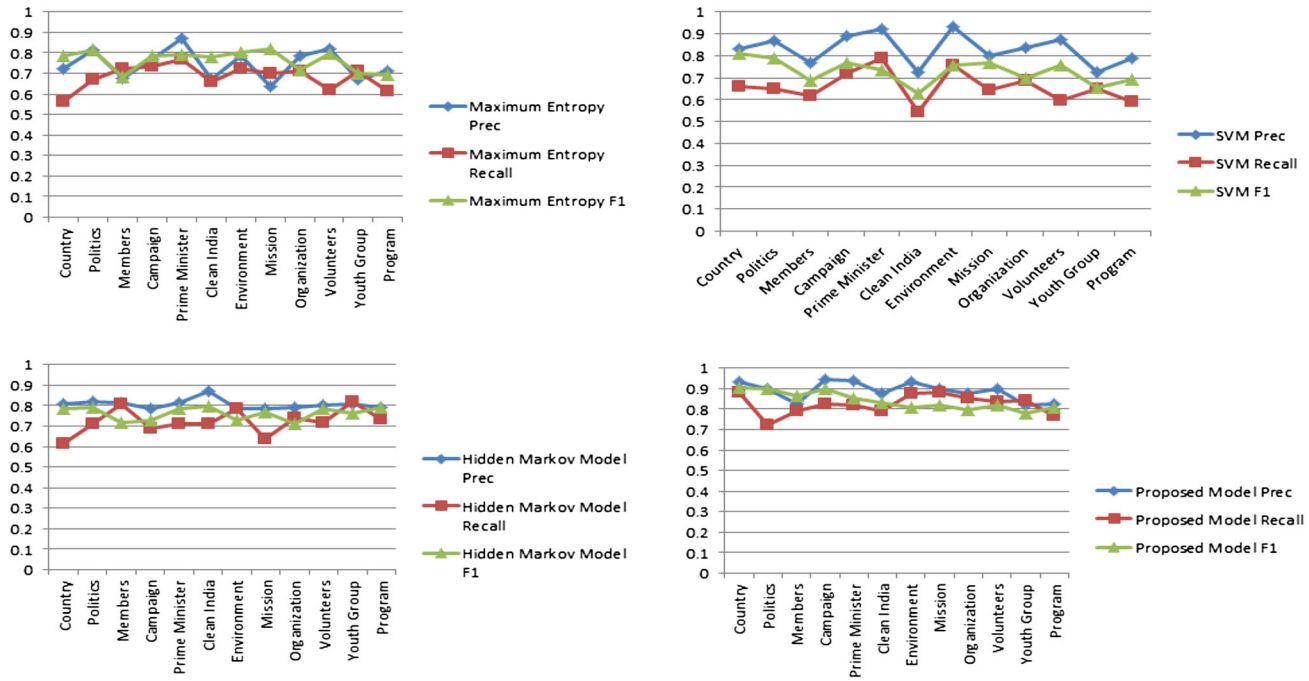

Figure 3. Statistical representation of final result.

some more has not been detected with its correct base. But for some entities, it has not given the correct results and hence we have resolved it using the statistical methods to find the proximity among the candidate entities. And also, our system has faced little challenges in entity disambiguation and yields false negatives than false positives. It was later conceived that if we had well annotated data sets with effective feature extraction techniques will improve this problem much further.

4.4 *Comparative Analysis of our proposed system with benchmarked datasets*

For further experiment analysis, we have downloaded the Ritter Twitter Datasets provided by [3] and the Ritter Twitter Datasets consists of 2400 annotated tweets with ground truth values as given in table 12. But for our experimental analysis, we have taken only 1000 tweets as the test sets due to the condition of the system and to save the computational time. We have considered this Ritter Twitter Datasets because, it has been manually annotated with research students and contains the wide range of entities such as person, location, organization, movies, and albums. Besides, it has taken the datasets from SemEval Conference and pre-processed the datasets according to the standards of the evaluations. Therefore, we benchmarked this datasets for our experimental analysis and observed the state of the performance of the proposed system.

The empirical evaluation has shown that our proposed system has outperformed the existing Ritter Model of evaluation and identified more candidate entities from the dataset. From the analysis, it has been apparently displayed that the proposed model has extracted more entities from the datasets and categorized them into the predefined categories appropriately. In this analysis, we have considered only the Entity Extraction method and extracted the entities based on the classifier that had been trained using Word2-Vec. Later, we have checked each entity in the DBpedia Ontology and count the entity if it has been present in the Knowledge Base. On that basis, the number of entities extracted has been calculated and the detailed view of the count of the analysis is given in the table 13.

When the proposed method has compared with the state of the art NER tools such as Stanford NER, NLTK Tagger, ANNIE and TwitterNLP, it has been observed that the proposed method has achieved considerably well and attained the good accuracy rate. Except in finding the appropriate Organization and categorizing the entities into Miscellaneous, in all other aspects, it has gained good

Table 12. Ritter Twitter dataset and its annotated ground truth.

Tweets	Tokens	Repeated characters	Abbreviations	Misspelled words	Total
1000	13487	672	308	1128	2108

Table 13. Proposed method for entity extraction and its accuracy rate.

NER Model	Tweets	Named Entities	Person	Location	Organization	Misc
Ground Truth	1000	682	405	129	88	60
Ritter *et al*	1000	496	303	98	54	41
Proposed Model	1000	604	386	104	62	52

Table 14. Comparison of proposed entity extraction method and NER tools.

	Person			Location			Organization			Misc		
NER systems	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Stanford NER	0.831	0.662	0.736	0.723	0.565	0.634	0.811	0.769	0.789	0.634	0.489	0.551
NLTK Tagger	0.769	0.620	0.686	0.672	0.661	0.666	0.728	0.69	0.708	0.689	0.511	0.586
ANNIE	0.726	0.546	0.623	0.788	0.689	0.735	0.851	0.719	0.779	0.571	0.435	0.493
TwitterNLP	0.725	0.648	0.684	0.768	0.675	0.718	0.788	0.696	0.739	0.557	0.471	0.510
Proposed Model	0.892	0.721	0.797	0.812	0.763	0.786	0.802	0.721	0.759	0.526	0.476	0.499

accuracy rate and F-score of the proposed system has outperformed the standard NER tools in the market as witnessed in the table 14.

Besides, we have also compared the total number of ambiguous entities that the proposed method has extracted by using the DBpedia Ontology and the finding has slated that the proposed method has deduced the ambiguous entity sets when compared to the Ritter proposed model. In the Ritter model, they have considered too many ambiguous entities for the evaluation since their model has lacked with improper entity mapping and they have not considered the supplementary entities particularly fetched from the web pages if any present in the tweets. Due to this imparity and negligence of entity context, the Ritter model has failed to supplement the entity extraction process successfully and increased the ambiguous entity list. But in the proposed model, we have extracted the entities from the tiny URLs such as web pages or blogs if any present on the tweet, identified the context of the entity using the trained classifier and mapped the candidate entities between the tweet and web pages.

5. Conclusion

In this article, we have explored the semantic aspect of extracting the potential information from the unstructured text. As detecting the named entities from the unstructured text yields high foundation for effective entity linking into any of the knowledge base, we have carried out various approaches for mention detection like propagation weight similarity measure, heuristic based approach, concept alignment and some of the linguistic features to appropriately match the named entities. Earlier research had focused much on finding the mentions without giving the due importance for the mentions and kept aside the scope of the applications. But in our proposed methods, we have formatted the method in such a way that it detects the mention according to the semantic meaning of the application. Next, we moved onto the problem of entity disambiguation, as this task finds the appropriate entity references in the given knowledge source such as DBpedia. We have devised three crucial methods to capture the entity references on the knowledge base through the features such as commonness, relatedness and topic based features. We have explored the dimension of entity references and added the context related to the co-references and modeled the system to effectively disambiguate the entities present in the tweets. Finally, we have proposed the new LDA model where we captured the topic of each and every tweet and give the overall dimension for the word distribution for effectively disambiguated the entities and linking the entities aptly to the given knowledge source.

References

[1] Van Erp M, Mendes P N, Paulheim H, Ilievski F, Plu J, Rizzo G and Waitelonis J 2016 Evaluating Entity Linking: An Analysis of Current Benchmark Datasets and a Roadmap for Doing a Better Job. In: *LREC* (Vol. 5, p. 2016)

[2] Han J, Wang C and El-Kishky A 2014 Bringing Structure to Text: Mining Phrases, Entity Concepts, Topics, and Hierarchies. In *KDD 2014 conference tutorial*. pp. 1968–1968

[3] Ritter A, Clark S and Etzioni O 2011 Named entity recognition in tweets: an experimental study. In:

Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 1524–1534). Association for Computational Linguistics

[4] Ganea O E, Ganea M, Lucchi A, Eickhoff C and Hofmann T 2016 Probabilistic Bag-Of-Hyperlinks Model for Entity Linking. In: *Proceedings of the 25th International Conference on World Wide Web* (pp. 927–938). International World Wide Web Conferences Steering Committee

[5] Wang C, Chakrabarti K, Cheng T and Chaudhuri S 2012 Targeted disambiguation of ad-hoc, homogeneous sets of named entities. In: *Proceedings of the 21st international conference on World Wide Web* (pp. 719–728). ACM

[6] Han X and Zhao J 2009 Named entity disambiguation by leveraging wikipedia semantic knowledge. In: *Proceedings of the 18th ACM conference on Information and knowledge management* (pp. 215–224). ACM

[7] Liu X, Zhou M, Wei F, Fu, Z and Zhou X 2012 Joint inference of named entity recognition and normalization for tweets. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1* (pp. 526–535). Association for Computational Linguistics

[8] Mendes P N, Jakob M, Garcia-Silva A and Bizer C 2011 DBpedia Spotlight: Shedding Light on the Web of Documents. In: 7th International Conference on Semantic Systems (I-Semantics'11)

[9] Ferragina P and Scaiella UFast and accurate annotation of short texts with wikipedia pages. IEEE software 29(1): 2012

[10] Derczynski Lnard D, Rizzo G, van Erp M, Gorrell G, Troncy R and Bontcheva K 2015 Analysis of named entity recognition and linking for tweets. *Inf. Process. Manag.* 51(2): 32–49

[11] Meij E, Weerkamp W and de Rijke M 2012 Adding semantics to microblog posts. In: *Proceedings of the 5th international conference on web search and data mining* (WSDM'12)

[12] Ibrahim Y, Amir Yosef M and Weikum G 2014 Aida-social: Entity linking on the social stream. In: *Proceedings of the 7th International Workshop on Exploiting Semantic Annotations in Information Retrieval* (pp. 17–19). ACM

[13] Jabbari S, Allison B and Guthrie L 2008 An Empirical Bayesian Method for Detecting Out of Context Words. In *International Conference on Text, Speech and Dialogue* (pp. 101–108). Springer Berlin Heidelberg.

[14] Manchanda P, Fersini E and Palmonari M 2015 Leveraging Entity Linking to enhance Entity Recognition in microblogs. In: *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)* (Vol. 1, pp. 147–155). IEEE

[15] Moro A, Raganato A and Navigli R 2014 Entity linking meets word sense disambiguation: a unified approach. *Trans. Assoc. Comput. Linguist.* 2: 231–244

[16] Turney P D and Pantel P 2010 From frequency to meaning: Vector space models of semantics. *J. Artif. Intell. Res.* 37(1): 141–188

[17] Houlsby N and Ciaramita M 2013 Scalable Probabilistic Entity-Topic Modeling. arXiv preprint arXiv:1309.0337

[18] Masada T, Kiyasu S and Miyahara S 2008. Comparing LDA with pLSI as a dimensionality reduction method in document clustering. In: *Large-Scale Knowledge Resources. Construction and Application* (pp. 13–26). Springer Berlin Heidelberg

[19] Wang J, Tong W, Yu, H, Li, M, Ma, X, Cai H and Han J 2015. Mining multi-aspect reflection of news events in twitter: Discovery, linking and presentation. In: *Data Mining (ICDM), 2015 IEEE International Conference on* (pp. 429–438)

[20] [21] Kumar A, Maskara S and Chiang I J 2015 Identifying semantic in high-dimensional web data using latent semantic manifold. *J. Data Anal. Inf. Process* 3(04): 136

[21] Wood J 2016 Source-LDA: Enhancing probabilistic topic models using prior knowledge sources. arXiv preprint arXiv: 1606.00577

[22] Wang Y, Agichtein E and Benzi M 2012 August TM-LDA: efficient online modeling of latent topic transitions in social media. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 123–131)

[23] Li, Y, Wang C, Han F, Han J, Roth D and Yan X 2013 August Mining evidences for named entity disambiguation. In; *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1070–1078). ACM

[24] Cucerzan S 2007, June Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In: *EMNLP-CoNLL* (Vol. 7, pp. 708–716)

[25] Gerber D, Hellmann S, Bühmann L, Soru T, Usbeck R and Ngomo A C N 2013, October Real-time RDF extraction from unstructured data streams. In: *International Semantic Web Conference* (pp. 135–150). Berlin, Heidelberg: Springer

[26] Kalloubi F and Nfaoui E H 2016 Microblog semantic context retrieval system based on linked open data and graph-based theory. *Expert Syst Appl* 53: 138–148

[27] Vicient C and Moreno A 2015 Unsupervised topic discovery in micro-blogging networks. *Expert Syst. Appl.* 42(17): 6472–6485

[28] Varga A, Basave A E C, Rowe M, Ciravegna F and He, Y 2014 Linked knowledge sources for topic classification of microposts: A semantic graph-based approach. *Web Semant. Sci. Serv. Agents World Wide Web* 26: 36–57

[29] Veningston K, Shanmugalakshmi R and Nirmala V 2015 Semantic association ranking schemes for information retrieval applications using term association graph representation. *Sadhana* 40(6): 1793–1819

[30] Vo, D T and Ock C Y 2015 Learning to classify short text from scientific documents using topic models with various types of knowledge. *Expert Syst. Appl.* 42(3): 1684–1698

[31] Deborah L J, Sathiyaseelan R, Audithan S and Vijayakumar P 2015 Fuzzy-logic based learning style prediction in e-learning using web interface information. *Sadhana* 40(2): 379–394

[32] Piantadosi S T 2014 Zipf's word frequency law in natural language: A critical review and future directions. *Psychon. Bull. Rev.* 21(5): 1112–1130

[33] [32] Kumar N S and Muruganantham D 2016 Disambiguating the Twitter stream entities and enhancing the search operation using DBpedia ontology: named entity disambiguation for Twitter streams. *Int. J. Inf. Technol. Web Eng. IJITWE*, 11(2): 51–63

[34] Raeesi M, Morid M A and Shajari M 2014 Trust Evaluation Using an Improved Context Similarity Measurement. arXiv preprint arXiv:1404.4592

[35] Roul R K, Asthana S R, Shah M and Parikh D 2016 Detecting spam web pages using content and link-based techniques. *Sadhana* 41(2): 193–202

[36] Sirsat S R, Chavan D V and Deshpande D S P 2014 Mining knowledge from text repositories using information extraction: A review. *Sadhana* 39(1): 53–62

[37] Kumar P P, Agarwal A and Bhagvati C 2014 A string matching based algorithm for performance evaluation of mathematical expression recognition. *Sadhana* 39(1): 63–79