International Journal of Information Technology and Web Engineering
Volume 11 • Issue 2 • April-June 2016

Disambiguating the Twitter Stream Entities and Enhancing the Search Operation Using DBpedia Ontology:
Named Entity Disambiguation for Twitter Streams

N. Senthil Kumar, SITE, VIT University, Vellore, India
Dinakaran Muruganantham, SITE, VIT University, Vellore, India

ABSTRACT

The web and social web is holding the huge amount of unstructured data and makes the searching processing more cumbersome. The principal task here is to migrate the unstructured data into the structured data through the appropriate utilization of named entity detections. The goal of the paper is to automatically build and store the deep knowledge base of important facts and construct the comprehensive details about the facts such as its related named entities, its semantic classes of the entities and its mutual relationship with its temporal context can be thoroughly analyzed and probed. In this paper, the authors have given and proposed the model to identify all the major interpretations of the named entities and effectively link them to the appropriate mentions of the knowledge base (DBpedia). They finally evaluate the approaches that uniquely identify the DBpedia URIs of the selected entities and eliminate the other candidate mentions of the entities based on the authority rankings of those candidate mentions.

KEYWORDS

DBpedia, Named Entities, OWL, RDF, SPARQL

1. INTRODUCTION

The real world entities have always been deal with one-to-many cardinality of mapping in the context of information retrieval. Most of the instances, it has been witnessed that one entity is linked with one or more real world entities or the entity can be referred with multiple entities in the knowledge base. This sort of ambiguity prevalence is very large in the Information Retrieval context and it can be further analyzed in Named Entity Recognition (NER). In order to facilitate the Named Entities identification processes much easier, the Markov Network (Andrea Varga, et al, 2014) was represented where entities were denoted with nodes and edges were the conditional dependencies between the mentions of the selected two entities. The principle task of the Markov Network is resembled with Bayesian Network except with the fact that Bayesian Network is acyclic and well directed. If we had given any document containing a group of potential named entities, then every single named entity mention in the document will be mapped in the Markov Network by forming the appropriate node and suitable conditional dependencies which have the sheer interpretations of the named entities. In some cases, the named entity has not linked with the appropriate nodes of the knowledge graph and that paves the way for ambiguous connection between entities. Hence, the Hidden Markov Model

DOI: 10.4018/IJITWE.2016040104

Copyright © 2016, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

was proposed and the Viterbi decoding methods were employed to fix the correct link between the mentions and the possible knowledge base named entities.

The Hidden Markov Model has used many languages processing task such as POS tagging, Named Entity Detection, and Classification etc. In this proposed approach, we have taken twitter as a social media site and carry out the process of identifying the potential named entities from the twitter streams. As the tweets are very short and noisy, finding the named entities is the challenging task and linking the named entities into the appropriate knowledge base mentions is a yet another cumbersome process to deal with. Hence, in this proposed system, we have explained the mechanism to link the entities into the knowledge base, removing the ambiguity persist over the extracted named entities and enhance the capabilities of searching much easier than before using semantic web technologies like RDF/SPARQL.

2. RELATED WORKS

According to authors in (Leon Derczynski, et al, 2014), they have identified that whenever the system deal with different types of named entities, the primary task is to recognize the entities out the document collections and then classify the entities into their respective category of domains. Besides, it has to find the suitable relationship exists between the entities. Some of the Named Entity Taggers have been employed to find out the entities in different types of documents and categorized it. Further to that, in order to find the category of the entity, the favorable approach described in (R. Bunescu et al, 2006), is that they have generalized the entity types as locations, persons, organizations, timestamp, etc. While doing so, the majority of the entity types fall under the category afore mentioned. It has facilitated the process of fixing the appropriate domains to the entities and linking the entities to the correct level of meaning.

In the paper (Valentina Presutti, et al, 2014), the author has identified the candidate entities from the collection of documents, and used Wikipedia has the useful resource for identifying the potential candidate entities out of the documents. According to (Andrea Varga, et al, 2014), Once, the named entities have been extracted, ranking the entities is the crucial task which requires high governing over the entity sets. Therefore, the author has done the entity level ranking using INEX and TREC. Several approaches had been attempted thereafter in ranking the named entities but failed to fulfill the major changes to be incorporated in this methodology. Besides, the lexical similarity measures have been taken to categorize the entities to the targeted group when the entities are too ambiguous. The relevance propagation was used to filter out the entities that do not link to the set of allowed categories.

Mendes et al, had developed DBpedia Spotlight for annotating the text documents with DBpedia URIs. The core function of the DBpedia Spotlight is that it first identifies the noun phrases in the given sentences and matches that mention with the DBpedia entity. Disambiguating the entity against the set of DBpedia mention is the challenging task and for that it had followed the Vector Space Model (VSM). The major disadvantage in this proposed approach is that it has not covered the out of the vocabulary sets (OOV) and NULL entity sets.

In this paper [Romil et al, 2014], the author has discussed about the difficulty of handling the disambiguated entity sets in the twitter streams. They have proposed an approach to disambiguate the entity sets using three features. 1) Find the similarity between the entity in tweet and its corresponding entity in Wikipedia URL. 2) Find the Jaccard Similarity between entity and anchor text string across multiple web pages. 3) Estimating the popularity of the entity using Twitter Trends. Though it seems that it has solved the impeding problems of disambiguation, the efficiency of the approach is drastically questioned and the time computation of the approach is elapsing and lead to complication.

The major contribution of this paper [Fahad Alahmari, 2014] is that it has delineated about the problem of entity description i.e., providing users the necessary facts about the selected entity. They have taken three key parameters to estimate this task such as entity query, entity type and entity attributes. For the query ambiguity, they have identified two types of ambiguity, one is semantic

polysemy which deals about one entity is referring multiple real world entities. Second, semantic synonymy which deals about multiple entities is linking with one real world entity. Even though they have proposed this approach, it has the dearth of semantic orientation of entity linking and thus prompts to bad indication to entity disambiguation.

3. ENTITY LINKING MECHANISM

Entity Linking (EL) is the crucial process of identifying the extracted named entity mentions from the social media site (e.g. Twitter) and link them to the appropriate URI entry in the referenced knowledge base (e.g. DBpedia). Sometimes, during the process of entity mapping, there would be the chance of identifying the named entities which are all referring to the different entities in the knowledge base. In such abnormal cases, we need to cluster those similar named entity sets and apply the Vector Space Model (VSM) to proportionately rank the entity sets according to the similarity measures calculated. Through this, we had identified the unique entry for the ambiguous named entity into the knowledge base. These ambiguous problems had earlier been witnessed (Stefan Zwicklbauer et al, 2013) in the Information Retrieval in terms of Word Sense Disambiguation (WSD). Many studies (R. Bunescu et al, 2006 and S. Kulkarni et al, 2009) were carried out in Word Sense Disambiguation to eradicate the dissimilarity persist in the information. Now we are corresponding the WSD methods to the entity linking task for yielding higher preciseness and recall for the mapping of entity set.

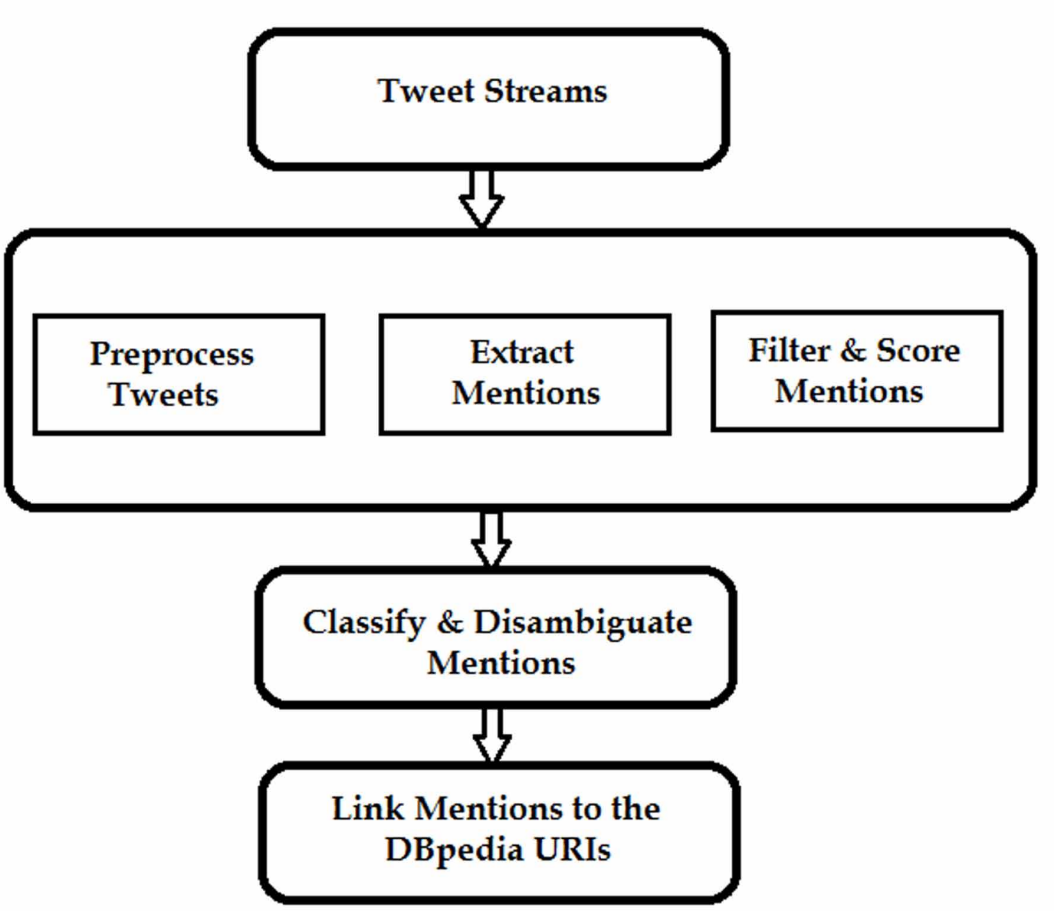
Generally, the disambiguation methods are classified into supervised methods, unsupervised methods and knowledge based methods. The supervised methods are basically a machine learning techniques which usually infer the results based on the data set available. It is a Decision List containing some set of proposed rules and classifies the samples through the If – then-else construct. The unsupervised methods are not inferring anything from the preoccupied datasets. Instead, it has absolutely relied with correlated words in the neighboring contexts. The co-referenced words in the text would form a cluster which points to the same subject or sometimes referring the same sense. In simpler terms, it clusters the words which are semantically similar. Unlike the above two methods, the knowledge based methods have used the knowledge resources (such as Dictionaries, Collocations, Thesaurus, Ontologies) to effectively disambiguate the content and yields better results when compared to other two supervised methods. Hence we have taken this last method to propose a solution to the task of entity linking from the tweets to the referring knowledge base.

3.1. Named Entity Extraction Procedure

Given the tweet streams of the specified event, the task is to extract the potential named entity phrases and filter the typed dependency relationships between the named entities and other related mentions in the tweets. Being consider this as an upheaval task, we have slightly modified the Standard POS Tagger algorithm (Ana Paula Silva et al, 2013) and proposed the new algorithm according to the requirement of our extraction task. In the Standard POS Tagger algorithm, the parser splits the sentences and collects the Noun Phrases, Verb Phrases etc and gives the appropriate labeling for all the segregated tokens. But there is a problem in the method that it does consider the category of every instance in the sentence. The objective of this research is to link the identified entities to the appropriate knowledge bases and give the machine a chance to comprehend the information flow of the content (see Figure 1). Hence, the very purpose of the research task has been failed if we just follow the standard POS tagger. Therefore, we have followed a heuristic to examine the sheer category of observed instances and augment its semantic similarity to the proximity.

The above algorithm has given the specific path of candidate entity selection and its relevant mention detection mechanism. When we deal with the process of entity-mention detection between the named entities from the tweets and the mention prevalence in the knowledge base like DBpedia(Valentina Presutti, et al, 2014 and D. Milne et al, 2008), the mention detection should be made easy and prevalent. In this regard, the DBpedia (Sebastian Hellmann et al, 2009) has promulgated

Figure 1. Proposed architecture for the system



the Infobox which holds the details of the entity and gives the appropriate URL link to other pages for some of the entities. Hence, to the context of semantic relatedness between the named entities from the tweets and mention detection from the DBpedia Infobox, we have defined it as:

$$S\left(e,m\right) = SR\left(\mathrm{e},\ \mathrm{C}_{\mathrm{Infobox}}\left(\mathrm{m}\right)\right)$$

The advantage of using the Infobox is that it gives the supplementary facts about the given entity (For instance, the entity Barack Obama would give the relational attributes like Occupation and Spouse that links to other entities in the DBpedia). Hence, it gives the search context more facilitating and robust by transforming the entity object into Linked Data. But the infobox has some inherent drawbacks such as born places, some canonical names have no specific links found on the DBpedia.

Algorithm 1

Input: Given the tweet streams $(T_1, T_2, \ldots..T_n)$ ***for the event*** (T_E)

Output: Identify the potential named entities $(N_1, N_2, \ldots..N_i)$ ***without any ambiguity***

Begin:

- Step 1: For every generated named entity (Ni) from the tweet (Tn), identify a URI that absolutely map the mention (m) in the knowledge base (KB).
- Step 2: If for the named entity (Ni), there would be more than one mention in the knowledge base referring, then in such cases, the link probability [17] for a mention m to the entity e would be calculated as:

$$F_{(e,m)} = \frac{\text{Count}(m, e)}{Count(m)}$$

  - Step 2.1: Then the targeted mention can be identified by using the minimum edit distance method.
  - Step 2.2: Then employ the classification task using the appropriate ontologies to disambiguate the mentions which have been ranked in minimum edit distance metric and select the entities that are fall in the same category of choices.
  - Step 2.3: Select the entity which have got high rank and assigned equal resemblance in the knowledge base.
- Step 3: Link that entity to the knowledge base.

End

4. PROPOSED WORK

In this section, we proposed the problem and computational methods to solve the impeding difficulties prevailing in the social media contents (e.g., Twitter) and make the posts more realistic and unambiguous. The task concerned here is to classify the set of tweets $T = (T_1, T_2, \ldots..T_n)$ whether they are related to the given specific event E. First, we assure that each tweet Ti is related to the event E, $\text{related}(T_i, E)$ if and only if the tweet is related or belongs to the category of the specified event.

The preliminary approach for the proposed work is consisting of three steps: Entity Detection, Feature Extractor & Entity Ranking and Category Similarity Score.

4.1. Entity Detection

Given the tweet $T = (T_1, T_2, \ldots\ldots T_n)$ for the specified event E, we need to preprocess the tweet in such a way that it removes the whitespaces, separators, emoticons, user IDs, urls, HTML tags etc and carry out the lightweight POS tagging to identify the proper nouns and noun phrases from the tweets. Then we need to classify them as potential named entities and identify the possible links in the DBpedia knowledgebase. For classification, we have applied Naïve Baiyes Classifier (Basave et al, 2013) to filter the extracted noun phrases (tokens) from the tweets and bridge them to the appropriate named entities. The Naïve Baiyes Classifier can extract the named entities based on the following facts:

A. If the selected candidate entity from the tweets found on the WordNet.
B. If the candidate entity shown its presence in DBpedia Knowledge Base.
C. If the candidate entity has given the path to the valid link in any web sites.

The entity detection and disambiguation is the major task of the information retrieval and we have given below (Table 1) the statistical performance of entity detection methods.

Given the amount of tweets T for the specified event or incident E, the task is to identify the events in the Tweets T which are fall in the category of the mentioned entity sets 'e' on the given time span. The candidate entities have been selected from the tweets T based on the relevance of the event or incidents. Hence to determine the function for the entity selection from the tweets, it has to set as related or not related. For every tweet t, we need to identify the entity sets e such that it would relate that entity to the specified event.

$$f\left(t\right)=T\rightarrow\left\{related|not\,related\right\}$$

That is,

$$f\left(t\right)=\begin{cases} related & if\,t\in Te \\ Not\,related & Otherwise \end{cases}$$

The confusion matrix for evaluation the entity relatedness and filtering is as follows (Table 2):

Hence we have followed the ARK POS tagger coupled with T-NER POS tagger for extracting the named entities from the tweets given and dissect the tweets into potential tokens (i.e. Named Entities). Besides, they were many ill-formed words present in the tweets and normalizing the ill-formed words is the challenging task (Abhishek Gattani et al, 2013). The normalization of ill-formed words is taken and chooses the relevant word based on the number of lexical similarity score.

4.2. Feature Extractor & Entity Linking

Once we detected the potential named entities from the tweets, we need to link them to the appropriate knowledge base like DBpedia to augment its context relatedness and bring in the proximity of comprehension. In order to perform this task, the entity occurrence should be checked against at DBpedia Infobox and information presented in the infobox should be mapped with named entity selection in the tweets. Hence, we define the task (Andrea Varga, et al, 2014) of named entity (e_i) to map the mention (m_j) in the infobox as:

Table 1. Performance of entity detection techniques

Entity Detection Technique	Accuracy
ARK POS TAGGER	77%
T-NER POS TAGGER	92%
ARK + T-NER (Merged)	98%

Table 2. Confusion matrix for entity relatedness

	Related	Not Related
Related	True Positive	False Positive
Not Related	False Negative	True Negative

$$\mathrm{F}\left(\mathrm{e},\mathrm{m}\right)=\left\{\begin{array}{c} 1 if\, e \varepsilon C_{info}\left(m\right) \\ 0 Otherwise \end{array}\right\}$$

During this mapping process, it had been noted that there were many mentions linking to the selected named entity (i.e., one – to – many cardinality). According to the DBpedia pages (R. Bunescu et al, 2006), the entities may have several candidate meanings and link to different DBpedia URIs. For instance, "Bank" can be linked to 'Reserve Bank' or 'Federal Bank' or mapped with 'Bank shores'. If we take the entity "Jaguar", it might be an animal or a car but it has different URI references in the DBpedia Spotlight. In such cases, we need to take the link probability (Anna Huang, 2008) for the mention (m) against the named entity (e) and define as:

$$\mathrm{F}_{\left(\mathrm{e},\mathrm{m}\right)}=\frac{\mathrm{Count}\left(\mathrm{m},\mathrm{e}\right)}{Count\left(m\right)}$$

The similarity score between the named entity and its associated concepts of the same ontology would not be implicated directly into the DBpedia category of links, because of the fact that the subcategory in DBpedia hasn't generated any hierarchy. Nevertheless, the idea of ancestors, predecessors, sub-categories can be still followed up for the mapping of updated query refinement. Here, we employed a designed ontology to categorize the mentions (m_j) for the given named entity (e) and estimate the similarity distance between a named entity (e) and the set of mentions (m_j) identified in the DBpedia knowledgebase. Now the underlying principle to estimate the distance factor for the named entity is relied with the selection of appropriate mention from the suggested set of mentions.

The similarity between the named entity and mention detection is formulated by the cosine similarity. The cosine similarity measure will be taken for all the link probabilities and noted down the similarity score differences in a separate way. The cosine similarity (Anna Huang, 2008) measure taken for the entity and mention can defined as:

$$CosSim\left(e,m\right)=\frac{\mathrm{Product}\left(\mathrm{e},\mathrm{m}\right)}{\left\|\mathrm{e}\right\|*\left\|\mathrm{m}\right\|}$$

Through this way, we give an indicator to map the candidate entity into the DBpedia referenced mention without any ambiguity. For each mention m identified to be disambiguated, we collected the possible list of candidate entities through the DBpedia Spotlight (Saira Gillani, et al, 2013) and establish the suitable function to link the suitable entity using the semantic similarity measures. In the algorithm given below, we have used the vectors of the entities and output the candidate entity which has the highest score. The similarity function for the algorithm for mention disambiguation is given as:

Then, we have used DBpedia Spotlight to fetch the URI reference for the entity which has assigned the highest score in the above algorithm 2and collect the JSON structure of entity along with type and resources. It can be implemented as:

4.3. Category Similarity Score

Once the preceding process of ambiguous problem gets solved, we create a binary relation for the named entity and DBpedia mention URI (See Table 3).

Algorithm 2

```
Input: Collect the possible list of ambiguous mention from DBpedia
Output: Yield the appropriate entity which has the highest rank
for the DBpedia URI
For each mention m ε M
Find the set of ambiguous entities ei of mention m
High ← MaxSim (vector(ei), vector(m))
Assign High → mention m
Return High
End for
```

Box 1

```
def filter(entity):
return JSON (DBpediaSpotlight.annotate(entity));
```

Table 3. Identifying the relation between named entity and candidate mention

Mention	NE Class	NE Link	DBpedia Ontology Class	Score
Barack Obama	Person	Dbpedia: Obama, USA	Dbpedia-owl: Person	3
Chennai	Location	Dbpedia: Chennai, India	Dbpedia-owl: Place	1
Cricket	Sports	Dbpedia: Cricket	Dbpedia-owl: Sports	2

Generally, all the entities in DBpedia have relevant name, label, type etc and to know the entity name given in the DBpedia for the specified URI, it can be queried through the SPAQRL query (Edgar Meij et al, 2012) as:

```
Select distinct *
where {
?URI rdf:label ?name
?URI dbpprop:iupacname ?name
filter(str(?name) = “Sachin Tendulkar”)
}
```

In order to get the category of the given entity from the DBpedia, we can give the SPARQL query as:

```
Select *
where
{
<http://dbpedia.org/resource/Vehicle>
<http://purl.org/dc/terms/subject>
?categories.
}
```

5. EVALUATION MEASURES

To validate the efficiency of the approach, we have used only four types of tweets (Current Events, Sports, Politics and Celebrities). The reason for selecting these categories as test cases for the proposed approach is that huge amounts of tweets have been posted in these three categories and we tested the same for the proposed approach. We had set the high priority to DBpedia Spotlight for referencing the entities from the tweets and link them to appropriate DBpedia URIs. That is, for each cases, we are here mapping the named entity (e) and the correct DBpedia URI(l) where the selected link is the valid DBpedia URI that give reference to the real world entity (e.g. http://dbpedia.org/resource/Narendra Modi). In some cases, there were chances that there would be no DBpedia URI references to the selected mentions, in such cases, we have link them to the non-DBpedia namespaces to avoid disambiguation. For every entity identified in the tweets, we have constructed the inverted index that would fetch the DBpedia URIs associated with the entity. Then give the entity to choose the best matching Dbpedia URI for the entity using the SPARQL query. To rank the DBpedia literals associated with the given entity, we have utilized the pre-existing mappings in DBpedia and constructed the coherent tree by <rdfs:subClassOf> relationships and to eliminate the duplicities, used <rdfs:equivalentClass>. The ranking of the DBpedia URIs list can be performed on the given context. Hence, we followed the context-aware approaches to find the co-occurrences of the entity e with other related entities in the same tweet context. This would be achieved by the appropriate SPARQL query as:

select ?x where { <e> <dbpedia-prop:wikilink> ?x. ?x <rdfs:type> <t_i>}

And to find the entity types linked to the given entity, we utilized the entity graph from the knowledge base <owl:sameAs> to discriminate the differences.

select ?x where {<e> <owl:sameAs> ?x . ?x <rdfs:type> <t_i> }

In order to test the performance and accuracy of the proposed method, we had taken the following real time named entities from recent ***Pathankot Attack*** and find the exact match of the DBpedia URI references of every potential named entity identified (See Table 4).

As we look for the exact-match of the pairs (e,l), it has to choose the correct resource (DBpedia URI) for the entity e. To make the process unambiguous, we have calculate the precision (P), recall (R) and F-measure (F) for the every candidate mentions against the selected entities and eventually filters the correct link for the entity.

The Precision (P) is defined as:

$$P = \frac{True\,Positive}{True\,Positive + False\,Positive}$$

The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.

The Recall(R) is defined as:

$$R = \frac{True\,Positive}{True\,Positive + False\,Negative}$$

The recall is intuitively the ability of the classifier to find all the positive samples.

Table 4. Find the exact match of DBpedia URI for every named entities

Tweets	Named Entities Identified	DBpedia Links to the Entity
Pakistan submits initial findings, ISI joins probe on #Pathankotattack	Pakistan	http://dbpedia.org/page/Pakistan_
	ISI	Not Found
China is also a victim of terrorist act. We share the anger of Indian people: Le Yucheng	China	http://dbpedia.org/page/China_
	Indian	http://dbpedia.org/page/Indian_
	Le Yucheng	Not Found
Kerala govt announces 50 lakh compensation for the family of Martyred Lt Col E K Niranjan	Kerala	http://dbpedia.org/page/Kerala_
	Martyred	http://dbpedia.org/page/Martyr_
	Lt Col E K Niranjan	Not Found
# **PathankotAttack** was not only a terror attack & attack on humanity.It was an attack on India- Anand Sharma, Congress	Terror attack	http://dbpedia.org/page/List_of_terrorist_incidents_
	India	http://dbpedia.org/page/India_
	Anand Sharma	http://dbpedia.org/page/Anand_Sharma_
It appears that there's lack of a coordination & should be immediately remedied:Yashwant Sinha, BJP	Yahwant Sinha	http://dbpedia.org/page/Yashwant_Sinha_
	BJP	http://dbpedia.org/page/BJP_

The F-Score is defined as:

$$R = 2X\frac{Precision\ .Recall}{Precision + Recall}$$

The F-measure is the harmonic mean of the precision and recall. Given below (Table 5) are the candidate mentions of the entities and calculated the F-Score of each against their ambiguity prevalence. We had witnessed the improvement of the accuracy rate of the precision and attained the satisfactory results over the proposed work.

6. CONCLUSION

In this paper, we have illustrated the working system of the named entity disambiguation methods and mapping the entities into the exact match of the knowledge base like DBpedia. Unlike other models of approaches, we have described the functional work of the model and presented the challenges in

Table 5. Sample accuracy score of the test

Entity	Precision	Recall	F-Score
Kamal Hassan	0.87	0.42	0.52
Sachin Tendulkar	0.9	0.65	0.67
Narendra Modi	0.9	0.44	0.51
Barack Obama	0.9	0.58	0.79

the selected named entity linking process such as dealing with the variations of the potential named entities, entity-mentions ambiguity, absence of entity in the DBpedia, entity mismatch etc. We have showed the working principles of these challenges and methods to overcome from all these technical glitches. Besides, we have proposed the method that strongly recommends when should not link the entities to the Knowledge Base even though it has high accuracy. This comprehensive work will further be enhanced with entity linking mechanism and clarify the impeding conundrums in various field of applications.

REFERENCES

Alahmari, F., & Thom, J.A. (2014). A Model for ranking entity attributes using DBpedia. *Journal of Information Management*.

Bansal, R. et al (2014, April). Linking entities in #Micropost". *Proceedings of #Microposts2014 Workshop.*

Basave, A.E.C., Varga, A., Rowe, M., Stankovic, M., & Dadzie, A.-S. (2013). Making sense of microposts (#MSM2013) concept extraction challenge. *Proceedings of the making sense of microposts concept extraction challenge*.

Bunescu, R., & Pasca, M. (2006). *Using Encyclopedic Knowledge for Named Entity Disambiguation*. EACL.

Derczynski, L. et al.. (2014, November). Analysis of named entity recognition and linking for tweets. *Information Processing & Management*.

Ganea, O.-E. et al. (2016). Probabilistic Bag of Hyperlinks Model for Entity Linking. *Proc. WWW2016*.

Gattani, A., Lamba, D. S., et al. (2013, August). Entity extraction, linking, classification, and tagging for social media: a wikipedia-based approach. *Journal Proceeding of the VLDB Endowment*. doi:10.14778/2536222.2536237

Gillani, S. et al. (2013). *Semantic Schema Matching Using DBpedia*. I.J. Intelligent Systems and Applications.

Hellmann, S. (2009). *Claus Stadler, et al, "DBpedia Live Extraction*. Springer OTM.

Hoffart, J. et al. (2011). *Robust Disambiguation of Named Entities in Text*. EMNLP.

Hogan, A., et al (2011, November). Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora. Web Semantics: Science, Services and Agents on the World Wide Web.

Holzmann, H., Tahmasebi, N., & Risse, T. (2013). BlogNEER: Applying Named Entity Evolution Recognition on the Blogosphere. *CEUR Workshop Proceedings (pp.* 28-39).

Huang, A. (2008, April). Similarity Measures for Text Document Clustering. In J. Holland, A. Nicholas, & D. Brignoli (Eds.), *Proceedings of the New Zealand Computer Science Research Student Conference*.

Kulkarni, S., Singh, A., Ramakrishnan, G., & Chakrabarti, S. (2009). *Collective Annotation of Wikipedia Entities in Web Text*. KDD. doi:10.1145/1557019.1557073

Marrero, M. (2009). *Evaluation of named entity extraction systems*. In Advances in Computational Linguistics, Research in Computing Science.

Meij, E., Weerkamp, W., & de Rijke, M. (2012). Adding Semantics to Microblog Posts.*Proceedings of the fifth ACM international conference on Web search and data mining*. doi:10.1145/2124295.2124364

Milne, D., & Witten, I. H. (2008). *Learning to Link with Wikipedia*. CIKM. doi:10.1145/1458082.1458150

Presutti, V., et al (2014, July). Uncovering the semantics of Wikipedia pagelinks. *Semantic Web – Interoperability, Usability, Applicability*.

Wei Shen, Jianyong Wang, et al (2015, February). Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *IEEE Transactions on Knowledge and Data Engineering*.

Silva, A.P. et al (2013, September). A New Approach to the POS Tagging Problem Using Evolutionary Computation. *Proceedings of Recent Advances in Natural Language Processing* (pp. 619–625).

Varga, A., et al (2014, April). Linked knowledge sources for topic classification of microposts: A semantic graph-based approach. Web Semantics: Science, Services and Agents on the World Wide Web.

Zwicklbauer, S., Seifert, C., & Granitzer, M. (2013, September). Do We Need Entity-Centric Knowledge Bases for Entity Disambiguation? *Proceedings of i-Know '13*. ACM.