Named Entity Disambiguation

Hai Nguyen[*]
Marshall University
1 John Marshall
Huntington, West Virginia 25755
nguyen124@marshall.edu

Viriyothai Pattamon[†]
Marshall University
1 John Marshall
Huntington, West Viriginia 25755
viriyothai@marshall.edu

ABSTRACT

Twitter is used for connecting group of people with the same interest together. The number of tweets increases more than 400 millions tweets a day. As the length of tweets from Twitter is limited to no more than 140 characters, that makes the named entities in tweets are ambiguous. So, how people map the entity in tweets with the corresponding entity in their life is a new challenge. To work on named entity disambiguation for tweets, named entity recognition and named entity linking methods are applied for finding and linking mentioned entities in tweets. This paper describes the using of Natural Language Processing concepts with named entity recognition, and compares the methods and algorithms for named entity linking task. To work on the challenge, this paper also describes about the knowledge base such as YAGO or DBPedia which could be applied with this disambiguation task.

Keywords

Tweet, Twitter, Named Entity Disambiguation, Named Entity recognition, Named Entity Linking, Natural Language Processing, YAGO, DBPedia, mentioned entity

1. INTRODUCTION

Twitter is a free social networking micro-blogging platform with more than 500 million users. The posts in Twitter called tweets are updated 400 million tweets per day. Twitter users can tweet or comment the tweets with hashtag or hyperlink by using several kinds of device such as cellphone or computer.

The major reasons why people always use Twitter in everyday life are using it as a news feed tool and socializing with friends and group of people in the same interest. To find correct group of interest, ambiguity of named entity in tweets is a huge challenge. Therefore, the mining task here is to extract named entity from tweets and link this named entity with the correct meaning entity in knowledge base.

In many recent years, researchers face with the problem of extracting named entities. They found that extracting named entities from tweets is a hard task for two reasons. First, tweets contain plenty of special entity types such as Movies, Musics, Companies, etc. Second, due to the limit of the length of tweets, tweets do not have enough information for identify a correct type of the words.

Named Entity Recognition (NER) is one of main steps applied to this task. NER is to catagorize every words in sentences using Natural Language Processing concept. For entity disambiguous in tweets, main purpose is to extract named entities from tweets of same user. These named entities refer to proper nouns (Person, Location, Organization and more). From figure 1, the graph shows the extraction of words in tweet. The named entities NER extracts, Messi, Manchester city and LaLiga, are used for the NEL step.

The second step is Named Entity Linking (NEL). NEL is to link true mapping of the named entities getting from NER and entities getting from knowledge base. From figure 1, Messi has two candidate entities; Leonel Messi and Messi (Film). Manchester City also has two candidate; Manchester City (Football club) and Manchester City (City).

Many methods have been invented to link entities mentioned in Web resources to knowledge base. These method are usually based on context similarity between text nearby mentioned entities and the document that contain the mentioned entities. However, with the short and informal language tweet, these methods are not working so well. Along with comparing context similarity, many previous methods use other measurements like topical coherence, entity dependency. But the limitation of these method still appear because many short tweets do not contain multiple entities. Therefore, we can not leverage these measurement with a single simple tweet. To solve the problem of lacking information, many methods try to collect the tweets into the same document firsts. Unluckily, this way violate the rule that entities are topically related [3]. For all above reasons, we use another approach by leveraging 2 criterion to mine the tweets: Intra-tweet information and Inter-Tweet information.

Intra-tweet information. Intra-tweet information is information that can directly infer from the tweets' content or context themselves. Any candidate of mentioned entity is considered potential if it qualify these 3 properties: (1) the prior probability is high (2) the contexts around entity and candidate are highly similar (3) the candidate of this men-

[*]
[†]

WOODSTOCK '97 El Paso, Texas USA

DOI: 10.475/123_4

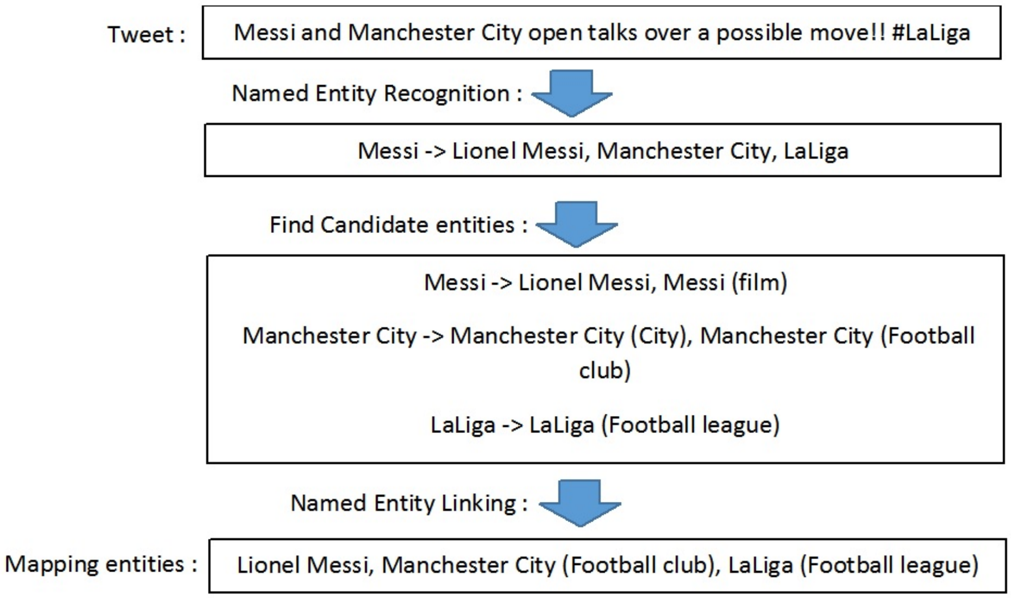

Tweet :
Messi and Manchester City open talks over a possible move!! #LaLiga
Named Entity Recognition :
Messi -> Lionel Messi, Manchester City, LaLiga
Find Candidate entities :
Messi -> Lionel Messi, Messi (film)
Manchester City -> Manchester City (City), Manchester City (Football club)
LaLiga -> LaLiga (Football league)
Named Entity Linking :
Mapping entities :
Lionel Messi, Manchester City (Football club), LaLiga (Football league)

Figure 1: Illustration of the Named entity disambiguation task. It shows two main steps of named entity disambiguation in tweets: Named entity recognition and Named entity linking.

tioned entity is highly topically coherent with other mentioned entity in the same tweet. From figure 2, McNealy and Sun are mentioned entities from intra-tweet.

Inter-tweet information. Sometime the tweet is too short to be understood fully. Therefore, we have to assume that each user has an underlying interest. That mean if users are interested in given entity and there are other entities that topically related to that entity then we can infer that users have a possibility of being interested in related entities too. From figure 2, McNealy and Scott are mentioned entities from inter-tweet.

2. DEFINITION AND NOTATIONS

Twitter special symbols Twitter users always use ” symbol to mention the other user's name in Tweet.

Collection of tweets In this paper, we will use notation T as a collection of tweets and |T| denotes for number of tweets in T.

Mentioned entity Let i is the index of tweets in T and 1≤ i ≤ |T|; each tweet t_i has number of mentioned entities denoted by M^i. We use j is the index of mentions entities in M^i and 1≤ j ≤ $|M^i|$. Let m^i_j is the mentioned entity in $|M^i|$.

Candidate entity Each mentioned entity m^i_j will have a set of possible candidates called R^i_j. Let q is the number of candidates in R^j_i with 1≤q≤$|R^i_j|$, the $r^i_{j,q}$ stands for the qth candidate in R^j_i .

True mapping entity Let e^i_j be the true mapping entity corresponding to mentioned entity m^i_j. If there is not an existence of e^i_j like that, NULL result is returned [3]

Dictionary To get candidate $\mathrm{r}^i_{j,q}$, the dictionary D [6] (figure 3) which is a list of key and value from five structures of Wikipedia: Entity, Redirect, Disambiguation, Hyperlink page is built. The key in entity page is the entity name and the corresponding value is the unique wikipedia link to this entity. The key in Redirect page is list words that have the same value which is the Wikipedia entity; for example the words ”MS Corporation”, ”Microsoft Company” will be redirected to the Wikipedia entity about ”Microsoft Cooperation”. The key in Disambiguation page is list of ambiguous words that refer to value which is the Wikipedia entity; for example the words ”N.Y” can refer to ”New York city” or ”New York times”. The key in Hyperlink page is the Wikipedia entities that are contained inside a Wikipedia entity.

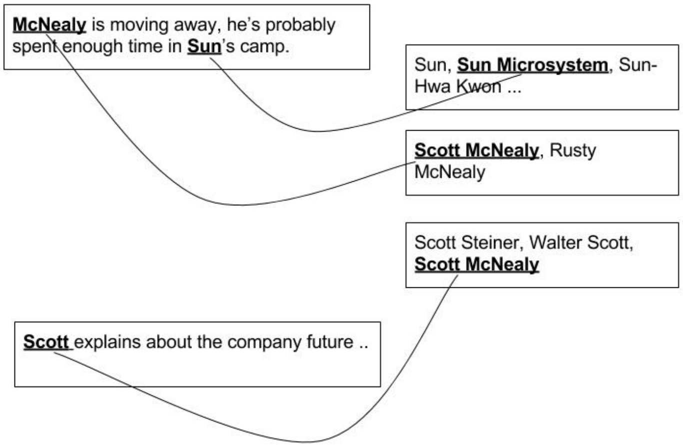
Graph definition Given graph G = (V,A,W) where V is a set of nodes which are candidates of entities; A is the set of edge connecting each pair of nodes and function W: A → R^+ is the weight assigning function which assigns the positive value to an edge in A. For candidates nodes that are belong to the same entity, there is no any edge connecting them.

3. KNOWLEDGE BASE

In this paper, we presents the knowledge base as dictionary used for finding mapping entity in Named Entity Linking. We will compare two knowledge bases: DBPedia and YAGO.

3.1 DBPedia

DBPedia [Bizer et al. 2011] is an open resource that contain structured information from Wikipedia. It allows users to query information about Wikipedia resources such as wiki page, hyperlinks, disambiguation, redirect, categories etc. For example, in Wikipedia, we have an article talk about the Sun which has unique wiki page link https://en.wikipedia.org/wiki/Sun. Inside the article Sun, we will have many hyperlinks such as https://en.wikipedia.org/wiki/Solar_System, https://en.wikipedia.org/wiki/Oxygen etc. When people mention about Sun in the tweet, there will be many other disambiguations such as Sun Microsystem, Brandon_Sun, Ari-



McNealy is moving away, he's probably spent enough time in **Sun**'s camp.

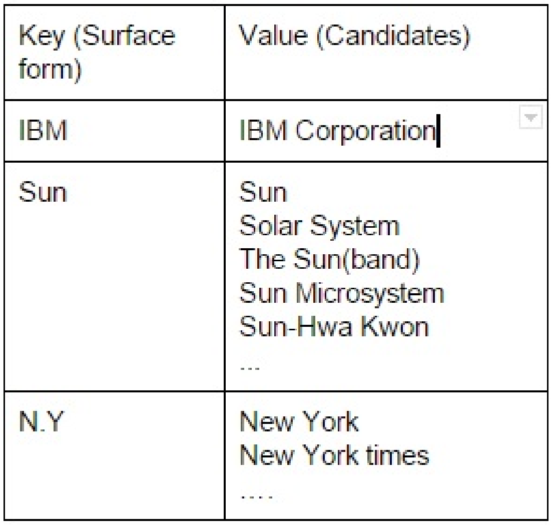
Sun, **Sun Microsystem**, Sun-Hwa Kwon ...

Scott McNealy, Rusty McNealy

Scott Steiner, Walter Scott, **Scott McNealy**

Scott explains about the company future ..

Figure 2: Illustration of the linking task. Left sentences is tweet and right words is candidates mapping entities.



Key (Surface form)	Value (Candidates)
IBM	IBM Corporation
Sun	Sun Solar System The Sun(band) Sun Microsystem Sun-Hwa Kwon ...
N.Y	New York New York times ....

Figure 3: Dictionary D

zona_Daily_Sun etc. Meanwhile, other entity that can be redirect to the Sun such as Solar, Center of Solar System, Helium Star etc. Currently, DBPedia contain datasets on 125 languages.

3.2 YAGO

YAGO [Fabian et al. 2012] is a knowledge base extracting information from Wikipedia as well as DBPedia. It contains a knowledge of more than 10 millions entities and 120 millions facts. YAGO3 can work on 10 languages and different scripts. While DBPedia extracts information from Wikipedia page, YAGO includes the DBPedia ontology with its datasets. Therefore, YAGO datasets is more diverse than DBPedia datasets.

4. NAMED ENTITY RECOGNITION

This paper compares Two models for finding the named entity in tweets.

4.1 HMM-Based Chunk Tagger

HMM-Based Chuck tagger [GuoDong and Jian 2002] works on 3 parts; Boundary Category, Entity Category and Word Feature.

Boundary Category is a set of number used to determine the start position and the end position of words in tweets. For example, 0, 1, 2, 3 is boundary category option. If the word in tweets has option 0, those words are a whole entity. While 0 means a whole entity, the options 1, 2 and 3 mean the words are at the beginning, in the middle and at the end of an entity respectively.

Entity Category is used to express the class of the named entity.

Word Feature is used to consider the other options of the words in tweets (One digit number, capital period, lowercase, uppercase, etc.). It determines three kinds of internal word features: 1) simple deterministic internal features of the words (capitalization, digitalization and more); 2) internal semantic features of important triggers; 3) internal gazetteer features. For external word features, it determines external macro context feature.

However, the problem of limit of length of tweets that is 140 characters makes this method might not accurate.

4.2 T-NER

T-NER [Ritter et al. 2011] is a model for named entity recognition in tweets.

Part of Speech Tagging After collect tweets from Twitter, each tweet will be processed to assign Part of Speech of each word. T-POS presents new Part of Speech Tagging which has higher performance than the other standard NLP tools. To find Part of Speech, n-grams model is annotated with Brown corpus and Penn Treebank tag set. It uses Conditional Random Fields for finding words identity such as prefixs and suffixs. Comparing to Standford NER systems which is a popular NER system in many researches, the error of T-POS is lower than Standford NER systems by 41%

error.

Shallow Parsing or chunking considers the phrases (noun phrases, verb phrase and more) in tweets by using Conditional Random Fields for learning.

Capitalization T-NER will predict the Capitalize words for named entity recognition because tweets may not come with capitalize words.

T-NER also processes Segmentation and Classification algorithms for predicting and classifying the named entity. This model works on tweets efficiently, by comparing to Standford NER.

5. NAMED ENTITY LINKING

We presents 4 models of named entity linking.

5.1 Model 1

This model presents the algorithm from LINDEN model [Wang et al. 2012b] and another paper [Milne and Ian 2008]. They work on finding context similarity in text.

5.1.1 Context Similarity

To calculate context similarity, the text in the tweet of mentioned entity will be compared with the text of the wikipedia searching snippet of candidate by using Cosine Similarity. The example of similarity between snippet and tweet is in figure (4).

The vector of words between snippet and tweet will be weighted by the number of those words appear in each snippet or tweet. To improve the exactness of Cosine Similarity, the words must not be Stop Words which is a list of meaningless words. The Cosine Similarity is describe in the following formula:

$$cos(A,B) = \frac{A_{i1} * B_{i1} + A_{i2} * B_{i2} + ... + A_{in} * B_{in}}{\sqrt{A_{i1}^2 + A_{i2}^2 + .. + A_{in}^2} * \sqrt{B_{i1}^2 + B_{i2}^2 + .. + B_{in}^2}} \quad (1)$$

From the formula above, the context similarity of mentioned entity m with candidate entity r_1 is higher than the context similarity of mentioned entity m with candidate entity r_2 if the r_1 is more similar than r_2.

5.2 Model 2

This model also processes the context similarity. The interesting point of this model is Graph reduction [Hoffart et al. 2011].

5.2.1 Graph Reduction Algorithm

After process initial interest score, the numbers of graph edges are huge numbers. The graph reduction algorithm will be applied to find new set of candidate entities. By comparing initial interest score, the algorithm will keep the closest 5 entities, and drop the others. After that, these new candidate entities set will be calculated the interest propagation score.

5.3 Model 3

This model analyzes from LIEGE model [Wang et al. 2012a] and AIDA light model [Nguyen et al. 2014]. This model combines context similarity from previous model, and processes Linking quality from Prior probability, context similarity and coherence.

5.3.1 Prior Probability

The prior probability of each candidate $r^i_j \in R^i_j$ corresponding to entity m^i_j means the possibility of being mentioned in the tweet of that candidate. The formula to calculate it is [3]:

$$Pp(r^i_{j,q}) = \frac{count(r^i_{j,q})}{\sum\limits_{c=1}^{|R^i_j|} count(r^i_{j,c})} \quad (2)$$

Where count($r^i_{j,q}$) is number of links that link to candidate $r^i_{j,q}$.

5.3.2 Topical Coherence

Topical Coherence $Coh(r^i_{j,q})$ for is calculated based on relatedness between candidate $r^i_{j,q}$ and the true mapping entity e^i_c of mentioned entity m^i_c as following formula [3]:

$$Coh(r^i_{j,q}) = \frac{1}{|M^i| - 1} \sum_{c=1, c \neq j}^{n} TR(r^i_{j,q}, e^i_c) \quad (3)$$

Where $|M^i|$ is the number of mentioned entity in tweet t_i. However we do not know the true mapping entity e^i_c yet so we have to find the most likely true entity to assign for this e^i_c. For that reason, we need an algorithm that can guess the best candidate to be e^i_c. The Iterative Substitution Algorithm [5] can help us to solve this.

5.3.3 Linking Quality

The initial interest score is the sum of prior probability, context similarity and topical coherence.

$$Coh(r^i_{j,q}) = \alpha * Pp(r^i_j) + \beta * Sim(r^i_j) + \gamma * Coh(r^i_j) \quad (4)$$

where

$$\alpha + \beta + \gamma = 1$$

are coefficients that are adjusted for the priority of each criteria [3].

The algorithm from LIEGE model is an algorithm to predict the true mapping entity. With recursive functions, this algorithm processes the mapping entity by improvement linking quality.

5.4 Model 4

This model presents KAURI model [Wang et al. 2013]. KAURI is a Graph based framework. To work on this graph based framework, the scenarios of Twitter's users are analyzed and concluded as 3 assumptions:

1) Every Twitter User has interest in something and the interest is represented through the interest score.

2) If users mention about given entity, we can infer that they are likely have interest in it.

3) If any entity that has high relatedness with users' favorite entities, we can assume that users may have interest in that entity too.

5.4.1 Topical Relatedness

Topical Relatedness relates to the different entities in real world which are candidate entities of different mentioned entities of tweets. The topical relatedness concept is shown in figure 4.

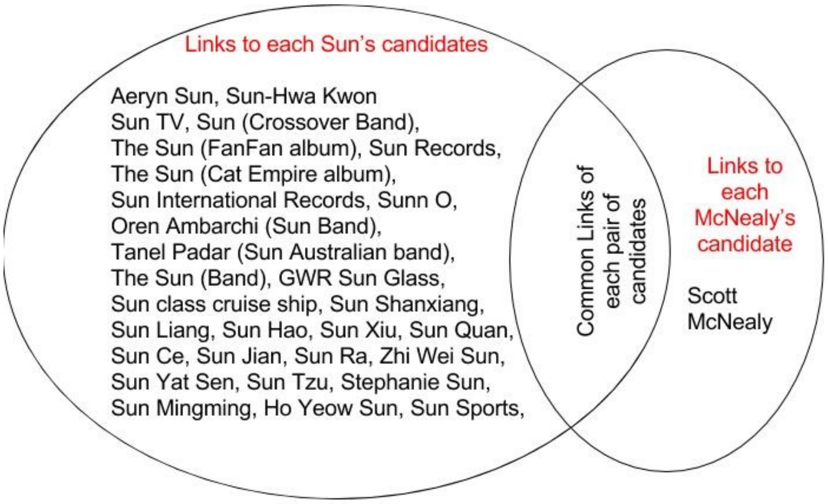

Links to each Sun's candidates
Aeryn Sun, Sun-Hwa Kwon
Sun TV, Sun (Crossover Band),
The Sun (FanFan album), Sun Records,
The Sun (Cat Empire album),
Sun International Records, Sunn O,
Oren Ambarchi (Sun Band),
Tanel Padar (Sun Australian band),
The Sun (Band), GWR Sun Glass,
Sun class cruise ship, Sun Shanxiang,
Sun Liang, Sun Hao, Sun Xiu, Sun Quan,
Sun Ce, Sun Jian, Sun Ra, Zhi Wei Sun,
Sun Yat Sen, Sun Tzu, Stephanie Sun,
Sun Mingming, Ho Yeow Sun, Sun Sports,
Common Links of each pair of candidates
Links to each McNealy's candidate
Scott McNealy

Figure 4: Candidates relatedness

In the previous research, the topical relatedness is calculated using Wikipedia documents for both entity candidates to check if either document has a link to the other, and assign value 1 if two candidates refer to each other. However, in this model each edge in the graph is assigned a weighted number called Topical Relatedness weight. Let M is the set of links relate to node v$_m$, N is the set of links relate to node v$_n$. The relatedness between these 2 nodes is calculated as:

$$TR(v_m, v_n) = \frac{log(max(|M|, |N|)) - log(|M \cap N|)}{log(|WP|) - log(min(|M|, |N|))}. \quad (5)$$

Where |WP| is the number of all Wikipedia entities [4].

KAURI combines the linking quality from LIEGE model as initial interest score. Due to initial interest score is not accurate for finding named entity in tweets. KAURI processes interst propagation algorithm to find correct mapping entity.

5.4.2 *Interest Propagation algorithm*

In this section, we will talk about how to propagate the interest score and topical relatedness weight to find the true mapping entity.

Normalized Initial Interest Score For each node v_k in graph G mentioned previously in Graph Definition, we have the initial interest score p_k as computed in formula 2. Let |V| is the number of nodes in graph G and the index of nodes 1≤k≤|V|. The normalized initial interest score np_k of *kth* node v_k is calculated as [3]:

$$np_k = \frac{p_k}{\sum_{c=1}^{|V|} p_c} \quad (6)$$

Normalized Topical Relatedness Weight A weight of the edge between 2 nodes v_m and v_n denoted as W(v_m,v_n). To normalized the weight W(v_m,v_n) of the edge E(v_m,v_n), we have to normalized the weight from node v_m to v_n as following formula [3]:

$$NW(v_m, v_n) = \frac{W(v_m, v_n)}{\sum_{v_c \in V_{v_m}} W(v_m, v_c)} \quad (7)$$

as well as from v_n to v_m using formula:

$$NW(v_n, v_m) = \frac{W(v_n, v_m)}{\sum_{v_c \in V_{v_n}} W(v_n, v_c)} \quad (8)$$

Because each edge has 2 ways of normalizing so we need a matrix B = |V|x|V| to represent the normalized weight where $b_{m,n}$ = NW($v_{m,n}$) and $b_{n,m}$ = NW($v_{n,m}$). Next we assign a vector $\overrightarrow{s}$=(s_1,..,s_k,..,$s_{|v|}$) where each s_k is the final interest score of node v_k and vector $\overrightarrow{p}$=(np_1,..,np_k,..,$np_{|v|}$) where each np_k is the normalized initial score for node v$_k$. We have the following formula [3]:

$$\vec{s} = \lambda\vec{p} + (1 - \lambda)B\vec{s} \quad (9)$$

Where 0≤ λ ≤ 1. As we we formula (9) is recursive so we need to initialize $\overrightarrow{s}$ equal to $\overrightarrow{p}$. The recursive formula (9) will stop when $\overrightarrow{s}$ reach the stable state in given threshold. The candidate $r^i_{j,q}$ corresponding to entity e^i_j will become the true mapping entity if its interest score in vector $\overrightarrow{p}$ is the highest among others' candidates that belong to the same entity e^i_j [3].

$$e^i_j = \max_{r^i_{j,q} \in R^i_j} s^i_{j,q} \quad (10)$$

5.5 Comparison of four models

From above three models, LINDEN model works on context around the text. However, it is not appropriate for tweets. Due to the limit of length of tweets, it makes context around tweets is also limit. Model 3 works on calculating linking quality and use recursive algorithm to find true mapping. This model is quite effective, but KAURI model proves that linking quality is not enough for determining the named entity in tweets. It may return false mapping entity. Therefore, KAURI model includes all features of these two models, and improve the linking quality by normalizing it.

The experimental result of KAURI comparing to LINDEN shows that KAURI is the most effective model in these three models. However, it would be more effective if KAURI combines the graph reduction algorithm in model 2.

6. CONCLUSIONS

This paper compares a method for named entity disambiguation. Other than presenting two knowledge base, we separate the task into two main taks: Named entity recognition and Named entity linking. For the named entity recognition task, two named entity recognition models are presented: HMM-Based Chunk Tagger and T-NER. Because T-NER is developed for handle the limit of tweets, it is more effective for the named entity recognition in tweets. For the named entity linking task, we compare three models: LINDEN, LEIGE and KAURI model. The result shows that KAURI is more appropriate for named entity linking in tweets bacause it combines all features from the other three models and improve them.

7. REFERENCES

[1] Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2011. *Named Entity Recognition in Tweets: An Experimental Study.* Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2011.

[2] C. Bizer, G. Kobilarov, J. Lehmann, S. Auer, and Z. Ives. *Dbpedia: A nucleus for a web of open data.* In ISWCâĂŹ07.

[3] Dat Ba Nguyen, Gerhard Weikum, Johannes Hoffart, and Martin Theobald. 2014. *AIDA-light: High-Throughput Named-Entity Disambiguation.* Proceeding of Linked Data on the Web, WWW 2014, Seoul, South Korea, 2014.

[4] David Milne and Witten Ian. *An effective, low-cost measure of semantic relatedness obtained from Wikipedia links.* Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA. 2008.

[5] GuoDong Zhou and Jian Su. 2002. *Named Entity Recognition using an HMM-based Chunk Tagger.* Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, USA. 2002.

[6] Jianyong Wang, Min Wang, Ping Luo, and Wei Shen. 2012. *LIEGE:: link entities in web lists with knowledge base.* In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '12). ACM, New York, NY, USA, 1424-1432. DOI=http://dx.doi.org/10.1145/2339530.2339753

[7] Jianyong Wang, Min Wang, Ping Luo, and Wei Shen. 2012. *LINDEN: linking named entities with knowledge base via semantic knowledge.* In Proceedings of the 21st international conference on World Wide Web (WWW '12). ACM, New York, NY, USA, 449-458. DOI=http://dx.doi.org/10.1145/2187836.2187898

[8] Jianyong Wang, Min Wang, Ping Luo, and Wei Shen. 2013. *Linking named entities in Tweets with knowledge base via user interest modeling.* In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '13), Inderjit S. Dhillon, Yehuda Koren, Rayid Ghani, Ted E. Senator, Paul Bradley, Rajesh Parekh, Jingrui He, Robert L. Grossman, and Ramasamy Uthurusamy (Eds.). ACM, New York, NY, USA, 68-76. DOI=http://dx.doi.org/10.1145/2487575.2487686.

[9] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Furstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. *Robust Disambiguation of Named Entities in Text.* Proceedings of the Conference on Empirical Methods in Natural Language Processing.

[10] Fabian M. Suchanek, Farzaneh Mahdisoltani, and Joanna Biega. 2012. *YAGO3: A Knowledge Base from Multilingual Wikipedias.* In Proceedings of the 21st international conference on World Wide Web (WWW '12). ACM, New York, NY, USA, 449-458. DOI=http://dx.doi.org/10.1145/2187836.2187898