Hashtag Sense Disambiguation Based on Content and Temporal Proximities

Minxin Zou† Mizuho Iwaihara‡
†早稲田大学大学院情報生産システム研究科
〒808-0135 福岡県北九州市若松区ひびきの 1-15-2206
E-mail: †zoumx@toki.waseda.jp, ‡iwaihara@waseda.jp

Abstract A twitter hashtag is a label or tag which makes it easier for users to find a message on a specific topic. But in fact, free creation of hashtags leads to the situation such that a hashtag may have multiple senses. In this paper, we propose a method to disambiguate hashtag senses according to release time and tweet contents. Our assumption is that in a specific time period, a sense of a hashtag and context words around the sense becomes dominant. In addition, for one user, he/she may also use the same sense when creating a hashtag. Then we construct a co-occurrence graph and perform community detection over graphs of different time periods. In the co-occurrence graph, nodes are hashtags and words in tweets, and edges represent the relationship between two nodes. Edge weights are based on four types of relationships: two nodes co-occur in one tweet, two nodes released by the same user, two nodes which are retweeted, and two nodes co-occur in an external document. A set of nodes which can represent a sense is extracted as a community. We use Wikipedia disambiguation lists to filter out and merge minor senses. When doing final matching, we associate a hashtag to a specific community according to the release time of the tweet and its contents.

Keyword Twitter, Hashtag, Sense Disambiguation, Proximity, co-occurrence graph

1. Introduction

Twitter is a popular online social networking service that enables users to send and read short 140-character messages called "tweets". Quite a large number of people write tweets every day. From the official data, there are 320 million monthly active users and 1 billion unique visits monthly to sites with embedded tweets. Special words in tweets starting with "#" are called hashtags. People use the hashtag symbol # at their will before a relevant keyword or phrase in their tweets to categorize tweets and help them easily retrieved in Twitter Search. There is no restriction on how to create hashtags. One of the results is that one hashtag may have multiple senses. For example, "#Apple" may refer to the fruit, while it also may refer to the Apple Company.

The process of identifying the sense of a polysemic word is called Word Sense Disambiguation (WSD). There exist many different approaches to WSD [1], which can be classified into the following three broad categories:

Knowledge based approaches

This type of approach is based on information contained in structured knowledge bases. WordNet [2] is one of the most commonly used knowledge bases in this type of approach.

Machine-learning based approaches

This approach can be further classified into three categories: supervised, semi-supervised and unsupervised approaches. Supervised approaches usually require large amounts of annotated datasets, called training data. Then machine-learning techniques are applied on training data to identify the sense of a word. Unsupervised approaches do not require any training data; they rely on unannotated corpus instead. Semi-supervised approaches take use of bootstrapping techniques to enlarge the training data. They use a small amount of annotated data first, called seed data, then a classifier trained on this small annotated data annotates raw corpus and the new annotated data serves as new seed data on which the classifier is retrained. All of these

three categories have their own merits and demerits.

Hybrid approaches

This type of approach combines the previous method to find the most appropriate sense of a given word.

In this paper, we propose a method based on content and temporal proximities to disambiguate the senses for hashtags. In our previous work [5], we proposed co-occurrence graphs to disambiguate senses. In a co-occurrence graph, nodes are hashtags and words in tweets, and edges represent the co-occurrence relationship between two nodes. Edge weights of the graph are frequencies of co-occurrence in tweets in our previous work. In this paper, we discuss temporal changes of senses, and improve edge weighting based on four types of relationships: 1) two nodes co-occur in one tweet, 2) two nodes released by the same user, 3) two nodes which are retweeted, and 4) two nodes co-occur in an external document. By considering these four factors, we think we can better show the relationship between a hashtag and the words related to it in comparison with our previous version. The graph is partitioned according to a community detection algorithm based on modularity optimization [3]. For temporal proximity, we divide the time axis into certain periods, and then apply content proximity on each period, to detect temporal changes of hashtag senses. We are facing two challenges:

Due to no restriction on usage of hashtags and words, people often use irregular words in tweets. These words are often influenced by popular events. One of our observations is that words in some communities can not represent a sense.

Because of the fast development of word senses, there is no golden standard to create perfect time periods. Each hashtag in different time periods often has different popular senses.

In order to resolve these issues, first we consider transforming informal words to formal words, then we detect communities and map to a sense inventory. In this paper, we choose Wikipedia Disambiguation Lists as our sense inventory.

We choose a sense based on ranking of the weights of words occurring in the tweets. For evaluation, a human judge examines the mapped results and determines whether matches are correct or wrong. Finally, we use error rate to show the rate of the wrong matches. In our result, average error rate changes from 0.05 to 0.3 over time. In July, the error rate reaches the lowest value, while in January, the error rate gets the highest value. It could be explained as people use special words in January.

The rest of the paper is organized as follows: In Section 2, we review related work focus on word sense disambiguation briefly. In Sections 3 and 4, the construction of our method is described. In Section 5, we perform experiments and show results. In Section 6, we evaluate the results. Finally, a conclusion and future work are presented in Section 7.

2. Related work

Vincent D. Blondel [3] has proposed an algorithm to detect communities. This algorithm first assigns a different community to each node of the graph. Then merge each node with its neighbors which can maximize a gain. Second, the communities found during the first phase are nodes in the new graph. Repeat the two phases iteratively and find the communities finally.

Another method uses community detection on co-occurrence graphs to do word sense induction. It uses a community detection algorithm called Link Clustering, clustering edges, which is equivalent to grouping the word collocations to identify sense-specific contexts [4]. On the other hand, our approach reflects social-media aspects within tweets into co-occurrence graphs.

3. Hashtag sense disambiguation

3.1 The outline of our system

Our system uses tweets at different time periods as our input, and executes preprocessing to collect the release time and contents of these tweets. Then we divide these tweets into several time intervals according to their release time. In each time interval, we generate a word list for each tweet. Each word and hashtag in the tweets are assigned as labels to nodes of the co-occurrence

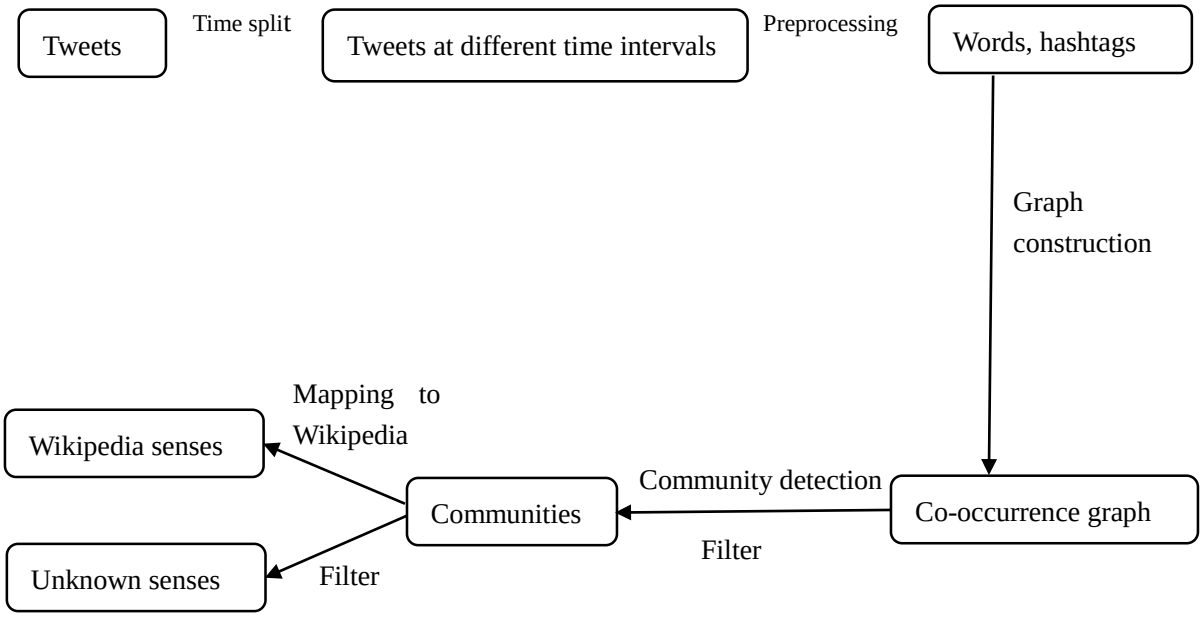


Fig. 1. System Overview

graph to be generated. Edges represent relationships between two nodes. Edge weights are based on four types of relationships: 1) two nodes co-occur in one tweet, 2) two nodes released by the same user, 3) two nodes which are retweeted, and 4) two nodes co-occur in an external document. The final weights are the linear weighting of them. Communities are detected based on the community detection algorithm proposed by Vincent D. Blondel in 2008 [3]. Each community is assigned with a list of words which can represent this community. The overview is shown in Fig. 1.

3.2 Time split

Hashtags have temporal features, meaning that hashtags are easily influenced by popular events. For example, the sense of "#orange" is different at different time periods. The left column belongs to interval "December", while the right column belongs to interval "November". Around 2015-12-31, many people may go to watch the sunset, so they talk about sky more often. But around 2015-11-15, although there are people talking about sunset, topic "fruit" and "juice" are also appearing in tweets (see Table 1). In order to improve the accuracy of our system, according to different release time, we divide these tweets into several time intervals by months. In each time interval, we perform graph construction.

Table 1. Words or hashtags around "#orange" at different time periods

t=2015-12-31	**t=2015-11-15**
#job	#job
#hiring	#sunset
#blue	#sky
bowl	love
#yellow	#color
#nature	health
sunset	#juice
#red	deal
#green	photo
close	#sun
careerarc	morning
#trees	#sunrise
cotton	#flower
opening	joseph
#pink	#fruit
#ca	syracuse
#brown	ale
love	#county
reflection	shop
#sky	#carrot

3.3 Data processing

Tweets are written freely by different people, resulting that words in tweets are often abbreviated or irregular. For example, some people use "sta" to represent the word "station". We apply the following steps to process tweets in each time interval:

1) The tweet contents are transformed from informal languages to formal languages according to a formal-informal list (constructed, samples are shown in Table 2) [6]. 2) The contents in tweets are filtered according to a stop word list. 3) All the words in tweets are transformed into their lowercase form. 4) POS (part-of-speech) tagging is applied to each word and hashtag, to extract noun words.

Table 2. Part of informal-formal list [6]

Informal	Formal
say sorry	apologize
go up	increase
go down	decrease
set up	establish
look at	examine
blow up	explode
find out	discover

3.4 Construction of co-occurrence graph

We define a co-occurrence graph of tweets as **G**=(**V**, **E**), where **V** is the union of a hashtag set $\mathbf{V_h}$ and word set $\mathbf{V_w}$. The nodes in $\mathbf{V_h}$ are assigned with larger weights than nodes in $\mathbf{V_w}$, because hashtags can be used to categorize tweets.

Edge weights are based on the following four types of relationships:

1. If two words or hashtags appear in one tweet, then create an edge between them and set the edge weight a factor $\mathbf{w_1}$ times the number of the tweets in which the two nodes co-occur.
2. If two words or hashtags in two tweets are released by an identical user, create an edge between their corresponding nodes. Set the edge weight as a factor $\mathbf{w_2}$ times the number of common users.
3. If two words or hashtags in two tweets are retweeted, create an edge between their corresponding nodes set the edge weight as a factor $\mathbf{w_3}$ times the number of the retweets between the two nodes.
4. After Step 3, if two nodes have an edge, then multiply their edge weight by $\mathbf{w_n}$, where $\mathbf{w_n}$ is the similarity of the two nodes calculated by WordNet. WordNet similarity can measure the semantic similarity or relatedness between two words [2].

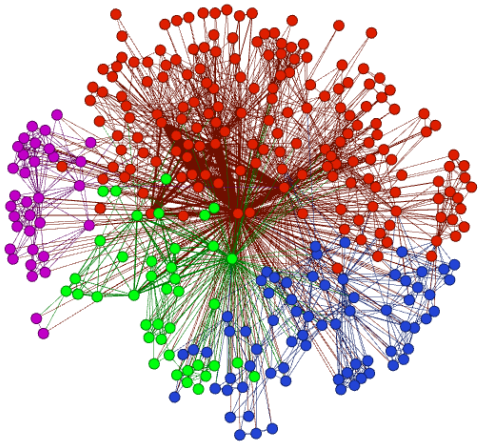


Fig. 2. Community detection for #cell

The final edge weights are the sum of the four relationships. For each edge, the edge weight is determined by formula (1). In the formula, $\mathbf{w_1}$ is set as 1, $\mathbf{w_2}$ is set as 0.8, $\mathbf{w_3}$ is set as 0.5, and $\mathbf{w_n}$ is determined by WordNet similarity value. Here, the constant factors on the four relationships are empirically determined.

$$w = w_1 + w_2 + w_3 + w_n \qquad (1)$$

After the co-occurrence graph is constructed, we remove all the nodes with a frequency in tweets below two, since nodes with such a low frequency cannot represent a sense.

3.5 Community detection

In the community detection algorithm of [3], first each node is assigned to a different community. Then, for each node **v**, **v** is moved to its neighbors **w** of **v** which the gain is maximum. This process is repeated until no further improvement can be achieved. In the second phase, the nodes are the communities obtained from the first phase. The weights of edges are given by the sum of the weight of the links between nodes in the corresponding two communities. Repeat the two phases on this smaller

graph until there is no more improvement and then a maximum of modularity is attained.

After we obtain communities, we first need to filter out communities whose sizes are small. The minimum size of a community is calculated by formula (2). In this formula, **n** is the number of words in this community, and **m** is the number of hashtags in this community. W_{w_i} is the weight of the word, and W_{h_i} is the weight of the hashtag.

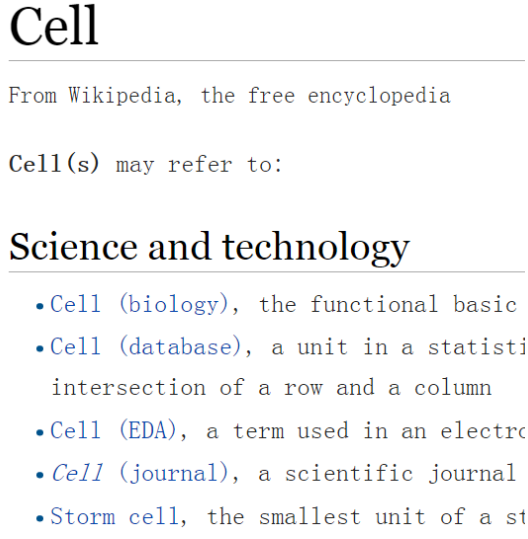
$$\text{Size} = \sum_{i=1}^{n} W_{w_i} + \sum_{i=1}^{m} W_{h_i} \quad (2)$$

If the size of a community is lower than the threshold, we delete it because it cannot represent a sense well.

In each community, we rank the words according to their weights, and the top 40 words are chosen as the context words of the community.

3.6 Mapping to Wikipedia Disambiguation Lists

Wikipedia has disambiguation lists for resolving ambiguous article titles. Figure 3 shows a part of the disambiguation list for "cell".


Cell

From Wikipedia, the free encyclopedia

Cell(s) may refer to:

Science and technology

- Cell (biology), the functional basic
- Cell (database), a unit in a statisti intersection of a row and a column
- Cell (EDA), a term used in an electro
- *Cell* (journal), a scientific journal
- Storm cell, the smallest unit of a st

Fig. 3. An example of Wikipedia disambiguation lists

Each Wikipedia sense on a Wikipedia disambiguation list corresponds to an article. We can map between hashtag context words and Wikipedia articles, by considering similarities between them. Here, tweets are often containing words that are casual, resulting in no corresponding sense in Wikipedia. So we try to resolve senses that are having Wikipedia entries, and the remaining communities as unknown or new senses. We do the mapping by the following steps:

1. Delete articles which are too short. Preprocess the remaining articles.
2. Calculate TF-IDF value for each word of all the articles.
3. Calculate cosine similarity in formula (3) for each community vector $\mathbf{d_1}$ and each Wikipedia word vector $\mathbf{d_2}$, where the values in $\mathbf{d_1}$ are the node weights when constructing co-occurrence graph, and the values of $\mathbf{d_2}$ are the TF-IDF value calculated in Step 2. As we mentioned in Section 3.5, the dimension of $\mathbf{d_1}$ is 40. Here we also choose the top 40 words in $\mathbf{d_2}$, so the dimension of $\mathbf{d_2}$ is 40.

$$\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)| \cdot |\vec{V}(d_2)|} \quad (3)$$

4. After calculating cosine similarities, we obtain a similarity score. The highest score is regarded as the most suitable match.
5. In the score obtained in Step 4, if all scores are lower than a threshold, then drop this sense.
6. If two groups match to the same Wikipedia article, merge them.
7. Repeat step 5 until there is no change.

Table 3. Mapping result of "#cell"

	Sense1	Sense2
Mobile_phone	0.5433	0.0927
Flash_memory	0.1849	0.1068
Cell_(biology)	0.0836	0.1954
Cell_(novel)	0.4841	0.0813
Cell_(microprocessor)	0.084	0.1068
Monasticism	0.4508	0.0645
Cellular_automaton	0.1154	0.0654
Clandestine_cell_system	0.1594	0.0924
Cellular_network	0.5298	0.1247
Asynchronous_transfer_mode	0.2796	0.075

4. Experiments

4.1 Dataset

We collected tweets via Twitter Search API, and parsed according to their release time. We chose 10 hashtags, including "#apple", "#banana", "#bank", "#book", "#cell", "#key", "#orange", "#rose", "#season" and "#train". For each hashtag, we use this hashtag as a search keyword to collect 3000 tweets in 2015. We conduct experiments for each hashtag at each time interval separately. In our experiment, we construct 12 time intervals. Each time interval starts from the beginning of each month and end at the end of each month. For example, 2015-1-1 to 2015-1-31 is a time interval, 2015-2-1 to 2015-2-28 is also a time interval. An example is shown in Table 4.

We found the following phenomena:

- Word preference on twitter. In Twitter, people prefer using funny, interesting, beautiful words than formal words. Because by using these words, tweets are more likely to be retweeted, the user him/herself is also more likely to be followed.
- Because Wikipedia disambiguation lists are not complete, certain communities cannot be mapped to a correct sense, even when it has a corresponding Wikipedia article.

Table 4. Example of the senses of #train at different time intervals

Time	Senses	Words
2015-11-1 to 2015-11-30	1	Train (noun) peace, gym, love, travel, workout, health, fit, photograph, gain, london, photogra
	2	Train_(wrestler) soccer, potty, child, videos, easily, dvd, mma, system, program, pro, save, wrestlin
2015-12-1 to 2015-12-31	1	Train (noun) brain, workout, gym, performance, fit, station, railway, new, ride, service, photograph
	2	Train (Training) follow, save, videos, body, radio, bag, walk, bike, door, mma

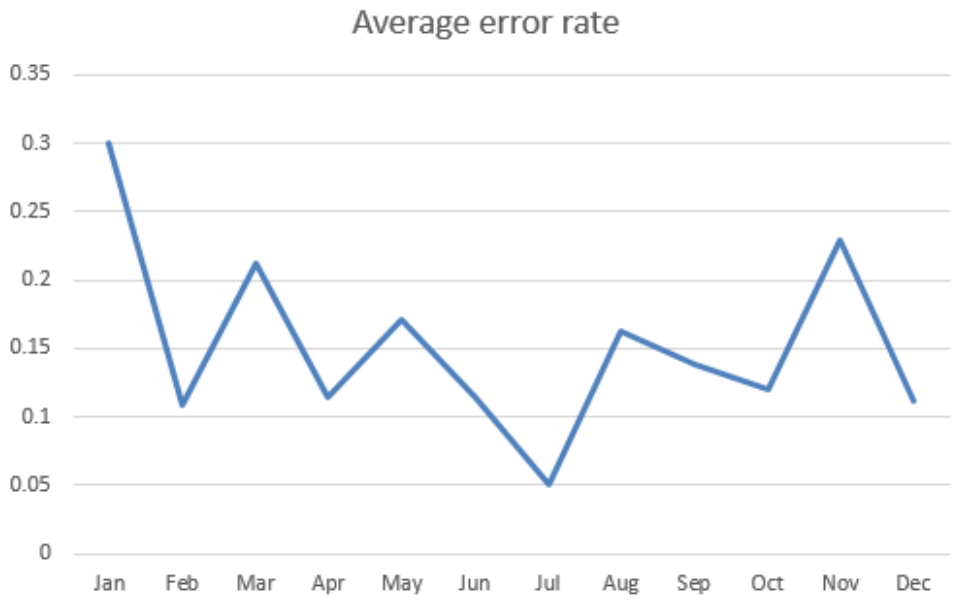


Fig. 4. Average error rate at each month

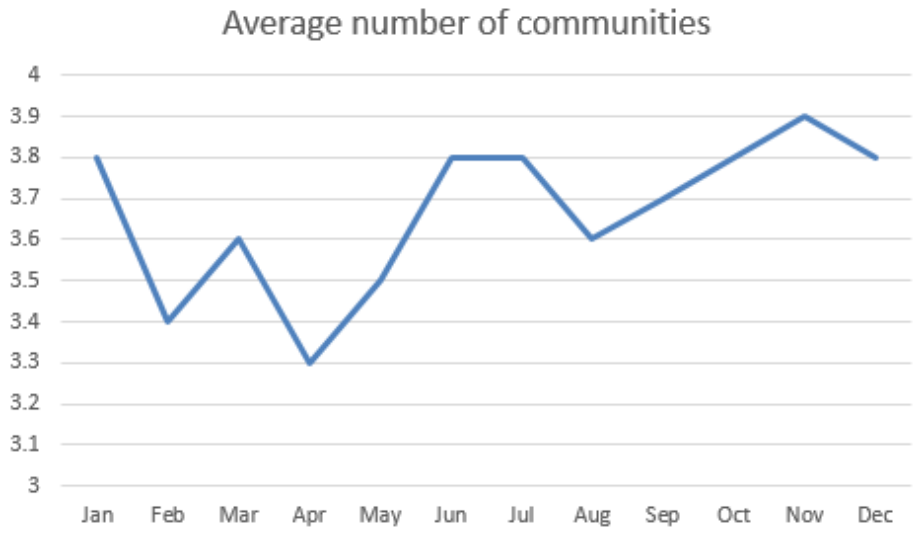


Fig. 5. Average number of communities in each month

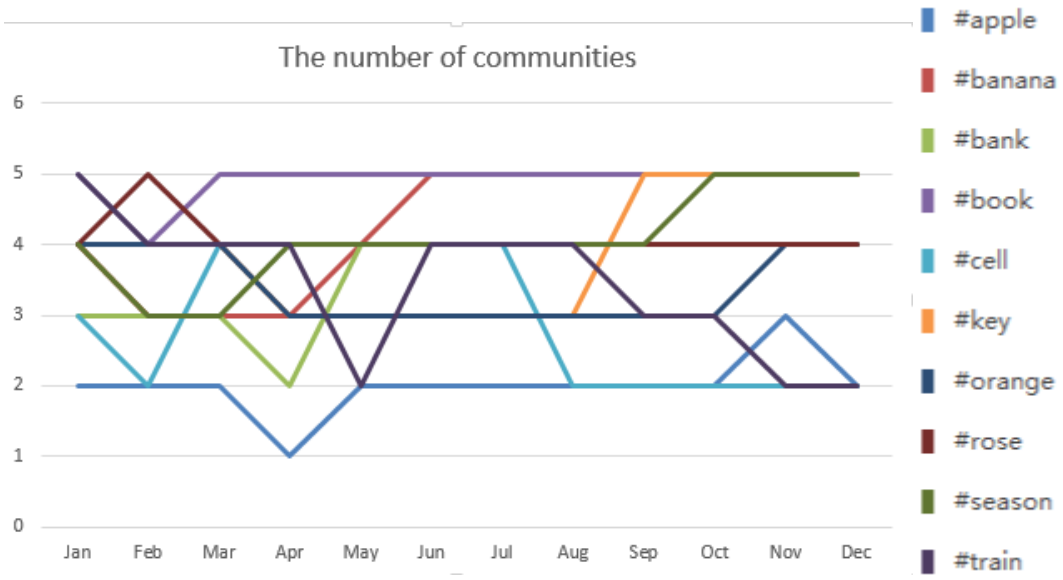


Fig. 6. The number of each community in each month

5. Evaluation

We evaluate our results based on the error rate. For each tweet, we first allocate it to a time interval according to its release time. Then as described in Section 3.5, we utilize context words to map graph communities to articles listed in disambiguation lists. Tweets are ranked by summing up all the weights of words occurring in the tweets, where the weight of each word is the sum of the edge weights of the graph community the word belongs to, as we mentioned in Section 3.4. Then a human judge examines the mapped results and determines whether matches are correct or wrong. In our framework, the Wikipedia article which is related to a given graph community most will be chosen. But communities and articles are not always corresponding each other. Therefore the human judge examines two stages of validity: 1) mapping between a hashtag and its community, and 2) mapping between a community and its article. An example is shown below:

Tweet Content: Cell #Phone Plans - NO Contracts, NO #Credit Checks, NO Bills, #Mobile -

http://CellPhone-Plans.net - Low #Cell Prices - #CellPhone 2015-2-15.

Sense1: 928.8 Sense2: 0.0

This is a tweet containing hashtag "#cell". The number after a sense Id is the edge weights of the mapped graph community. By doing the mapping we mentioned in Section 3.6, we can get a correspondence between Sense 1 and "Mobile Phone" in Wikipedia, while Sense 2 has a correspondence with "biology" in Wikipedia. Sense1 has a higher score than Sense 2, so our algorithm chooses Sense 1. The human judge examines correctness of the top-ranked mappings.

Here also has an error example:

Tweet Content: If you don't have a #KINDLE download the free app for #CELL #PC #TABLET http://amzn.to/1BtVqdO http://getBook.at/HTG100K #Howtouse #Twitter 2015-11-15

Sense1:0.0 Sense2: 0.0

In this example, the edge weights are all 0.0. But obviously, "#cell" in this tweet should also be mapped to "Mobile Phone". So we regard this tweet as a wrong example.

We evaluate by the error rate as shown in (4). If the ratio of correct mappings become higher, the error rate will be lower.

$$\text{Error rate} = \frac{The\ wrong\ matches}{All\ matches} \quad (4)$$

In evaluating error rates, we choose 100 tweets for each of the 10 benchmark hashtags. Figure 4 shows the average error rate of the evaluation, where the average is taken over hashtags collected in each period. In Figure 4, the error rate in July is lowest, while in January it becomes highest. It could be explained as people use special words in January, such as "#festival" and "#NewYear". As a result, words become difficult to be mapped to Wikipedia disambiguation lists. Figure 5 shows the average number of communities in each month, and Figure 6 shows the details of the Figure 5.

6. Conclusion and future work

In this paper, we proposed a method to disambiguate hashtag senses based on contents, and we observed temporal changes of these senses. The most challenge is that twitter users use words freely so that we have no golden standard to match them. In order to solve this problem, we transform informal words to formal words. This can improve the situation in some degrees. Our ultimate goal is to detect senses that are emerging and not covered by Wikipedia or other knowledge bases, but popular between twitter users.

In the future, we will focus on finding a better standard to match words in tweets. Also, for dividing tweets into different time intervals, we will find a better function to divide it.

7. Reference

[1] R. Pandit, S. Kumar Naskar. A Memory Based Approach to Word Sense Disambiguation in Bengali Using k-NN Method. 2015 IEEE, p 383 – 386.

[2] T. Pedersen, S. Patwardhan, J. Michelizzi: WordNet::Similarity - Measuring the Relatedness of Concepts. HLT-NAACL 2004.

[3] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre.: Fast unfolding of communities in large networks J. Stat. Mech. Theory Exp. (10), 2008, pp. 10008.

[4] D. Jurgens: Word Sense Induction by Community Detection. Graph-based Methods for Natural Language Processing, 2011, pages 24–28.

[5] M. Wang, M. Iwaihara: Hashtag Sense Induction Based on Co-Occurrence Graphs. Proc. 17th Asia-Pacific Web Conference (APWeb 2015), Lecture Note in Computer Science 93132, pp. 154-165, Sep. 2015.

[6] http://www.engvid.com/english-resource/formal-informal-english/; http://ielts-academic.com/2012/07/06/informal-formal-vocabulary-for-ielts/