

# Sense-Level Semantic Clustering of Hashtags

Ali Javed and Byung Suk Lee

Department of Computer Science, University of Vermont, Burlington, VT, U.S.A.  
{ajaved, bslee}@uvm.edu

**Abstract.** We enhance the accuracy of the currently available semantic hashtag clustering method, which leverages hashtag semantics extracted from dictionaries such as Wordnet and Wikipedia. While immune to the uncontrolled and often sparse usage of hashtags, the current method distinguishes hashtag semantics only at the word-level. Unfortunately, a word can have multiple senses representing the exact semantics of a word, and, therefore, word-level semantic clustering fails to disambiguate the true sense-level semantics of hashtags and, as a result, may generate incorrect clusters. This paper shows how this problem can be overcome through sense-level clustering and demonstrates its impacts on clustering behavior and accuracy.

**Keywords:** sense-level, semantic clustering, hashtag

## 1 Introduction

Hashtag clustering has emerged as an interesting and important topic of study in online social media, which is arguably the best source of timely information. On Twitter alone, an average of 6,000 micro-messages are posted per second [14]. Thus, social media analysts use clusters of hashtags as the basis for more complex tasks on tweets [6] such as retrieving relevant tweets [6, 7], tweet ranking, sentiment analysis [17], data visualization [1], semantic information retrieval [11], and user characterization. Therefore, the accuracy of hashtag clustering is important to the quality of the resulting information in those tasks.

The popular approach to hashtag clustering has been to leverage the tweet texts accompanying hashtags [1] [2] [6] [8] [11, 12, 13] by identifying their “contextual” semantics [9]. There are two prominent problems with this approach, however. First, tweet texts are limited to only 140 characters in length and, moreover, a majority of hashtags are not used frequently enough to find sizable tweet texts accompanying them, thus causing a sparsity problem. Second, tweet texts are open-ended, with no control over their contents at all, and therefore often exhibit poor linguistic quality. (According to 2009 Pear Analytics Twitter Study, 40% of tweets are “pointless babble” [4].) These problems make text-based techniques ineffective for hashtag clustering. Hence, methods that utilize other means to identifying semantics of hashtags are needed.

In this regard, the focus of this paper is on leveraging *dictionary metadata* to identify the semantics of hashtags. We adopt the pioneering work done by Vicent and Moreno [15, 16]. Their approach identifies the “lexical” semantics of hashtags from external resources (e.g., Wordnet, Wikipedia) independent of the tweet messages themselves. To the best of our knowledge, their work is the only one that uses this metadata-based approach. This approach has the advantage of being immune to the sparsity and poor linguistic quality of tweet messages, and the results of their work demonstrate it.

On the other hand, their work has a major drawback, in that it makes clustering decisions at the *word* level while the correct decision can be made at the *sense* (or “concept”) level. Obviously the correct use of metadata is critical to the performance

of any metadata-based approach, and indeed clustering hashtags based on their word-level semantics has been shown to erroneously putting hashtags of different senses in the same cluster (more on this in Section 4).

In this paper, we devise a more accurate sense-level metadata-based semantic clustering algorithm. The critical area of improvement is in the construction of similarity matrix between pairs of hashtags, which then is input to a clustering algorithm. The immediate benefits are shown in the accuracy of resulting clusters, and we demonstrate it using a toy example. Experimental results using two gold standard tests showed gains of 26% (when hashtag semantics are not controlled) and 47% (when controlled), respectively, in terms of the weighted average pairwise maximum f-score (Equation 5), where the weight is the size of a ground truth cluster. Despite the gain in the clustering accuracy, we were able to keep the run-time and space overheads for similarity matrix construction within a constant factor (e.g., 5 to 10) through a careful implementation scheme.

This paper contains more rigorous experiments than the authors' conference paper [3], which was invited into this LNCS series with extended content.

The remainder of this paper is organized as follows. Section 2 provides some background knowledge. Section 3 describes the semantic hashtag clustering algorithm designed by Vicent and Moreno [16]. Section 4 discusses the proposed *sense-level* semantic enhancement to the clustering algorithm, and Section 5 presents its evaluation against the word-level semantic clustering. Section 6 presents other work related to the semantic hashtag clustering. Section 7 summarizes the paper and suggests future work.

## 2 Background

### 2.1 Wordnet – synset hierarchy and similarity measure

Wordnet groups English words into sets of synonyms called synsets. Synsets in Wordnet are interlinked by their semantics and lexical relationships, which results in a network of meaningful related words and concepts. The concepts are linked to each other using the semantic and lexical relationships mentioned. Given this network of relationships, we use the Wu-Palmer [18] similarity measure in order to stay consistent with the baseline algorithm by Vicent and Moreno [16]. Given concepts organized in a hierarchy, the Wu-Palmer similarity,  $\text{sim}_{WP}(C_1, C_2)$ , between two concepts  $C_1$  and  $C_2$  is defined as

$$\text{sim}_{WP}(C_1, C_2) = \frac{2 \cdot \text{depth}(\text{LCS}(C_1, C_2))}{\text{depth}(C_1) + \text{depth}(C_2)} \quad (1)$$

where  $\text{LCS}(C_1, C_2)$  is the least common subsumer of  $C_1$  and  $C_2$  in the hierarchy.

We use this Wordnet functionality to calculate the semantic similarity between hashtags, that is, by grounding hashtags to specific concepts (called “semantic grounding”) and calculating the similarity between the concepts.

### 2.2 Wikipedia – auxiliary categories

Wikipedia is the most popular crowd-sourced encyclopedia. Not all hashtags can be grounded semantically using Wordnet because many of them are simply not legitimate terms found in Wordnet (e.g. #Honda). This situation is where Wikipedia can be used to look up those hashtags. Wikipedia provides auxiliary categories for each article. For example, when Wikipedia is queried for categories related to the page titled “Honda”, it returns the auxiliary categories shown in Figure 1.

Auxiliary categories can be thought of as categories the page belongs to. In this example, if we are unable to look up the word “Honda” on Wordnet, then, through the help of these auxiliary categories, we can relate the term to Japan, Automotive, Company, etc. There are several open source Wikipedia APIs available to achieve this purpose – for example, the Python library “wikipedia”.

```
[Automotive companies of Japan',
Companies based in Tokyo',
Boat builders',
Truck manufacturers',
Vehicle manufacturing companies',
...
]
```

**Fig. 1.** Wikipedia auxiliary categories for “Honda”.

### 2.3 Hierarchical clustering

Hierarchical clustering is a viable approach to cluster analysis, and is particularly suitable for the purpose of hashtag clustering in this paper. There are two strategies for hierarchical clustering – bottom-up (or agglomerative) and top-down (or divisive) – and bottom-up strategy is used in our work because it is conceptually simpler than top-down [5]. Several distance measures are available to provide linkage criteria for building up a hierarchy of clusters. Among them, single-linkage method and unweighted pair group method with arithmetic mean (UPGMA) are used most commonly. Single-linkage method calculates the distance between two clusters  $C_u$  and  $C_v$  as

$$d(C_u, C_v) = \min_{u_i \in C_u \wedge v_j \in C_v} \text{dist}(u_i, v_j) \quad (2)$$

and UPGMA calculates the distance as

$$d(C_u, C_v) = \sum_{u_i \in C_u, v_j \in C_v} \frac{d(u_i, v_j)}{|C_u| \times |C_v|} \quad (3)$$

where  $|C_u|$  and  $|C_v|$  denote the number of elements in clusters  $C_u$  and  $C_v$ , respectively.

To generate output clusters, we extract “flat clusters” from the hierarchy using the *distance* criterion, which, given a distance measure, forms flat clusters from the hierarchy when items in each cluster are no farther than a distance threshold.

## 3 Semantic Hashtag Clustering

The semantic clustering approach proposed by Vicient and Moreno [16] uses Wordnet and Wikipedia as the metadata for identifying the lexical semantics of a hashtag. Source codes of their algorithms were not available, and so we implemented the approach described in Vicient’s PhD dissertation [15] to the best of our ability.

There are three major steps in their semantic clustering algorithm [16]: (a) semantic grounding, (b) similarity matrix construction, and (c) semantic clustering. Algorithm 1 summarizes the steps.

In the first stage (i.e., semantic grounding), each hashtag is looked up in Wordnet. If there is a direct match, that is, the hashtag is found in Wordnet, then it is added as a single candidate synset, and, accordingly, all the concepts (or senses) (see Section 2.1) belonging to the synset are saved in the form of a list of candidate concepts related to the hashtag. We call this list  $LC_h$ . If, on the other hand, the hashtag is not found in Wordnet, then the hashtag is split into multiple terms (using a word segmentation technique) and, then, the leftmost term is dropped sequentially until either a match is found in Wordnet or there is no more term left.

Input: list  $H$  of hashtags

Output: clusters

**Stage 1 (Semantic grounding):**

Step 1: For each hashtag  $h \in H$  perform Step 1a.

Step 1a: Look up  $h$  from Wordnet. If  $h$  is found then *append* the synset of  $h$  to a list ( $LC_h$ ). Otherwise segment  $h$  into multiple words and drop the leftmost word and then try Step 1a again using the reduced  $h$  until either a match is found from Wordnet or no more word is left in  $h$ .

Step 2: For each  $h \in H$  that has an empty list  $LC_h$ , look up  $h$  in Wikipedia. If an article matching  $h$  is found in Wikipedia, acquire the list of auxiliary categories for the article, extract main nouns from the auxiliary categories, and then, for each main noun extracted, go to Step 1a using the main noun as  $h$ .

**Stage 2 (Similarity matrix construction):** Discard any hashtag  $h$  that has an empty  $LC_h$ . Calculate the maximum pairwise similarity between each pair of lists  $LC_{h_i}$  and  $LC_{h_j}$  ( $i \neq j$ ) using any ontology-based similarity measure.

**Stage 3 (Clustering):** Perform clustering on the distance matrix (1's complement of the similarity matrix) resulting from Stage 2.

**Algorithm 1:** Semantic hashtag clustering [16].

For each hashtag that was not found from Wordnet in Step 1 (i.e., of which the  $LC_h$  is empty), it is looked up in Wikipedia. If a match is found in Wikipedia, the auxiliary categories (see Section 2.2) of the article are acquired. Main nouns from the auxiliary categories are then looked up in Wordnet, and if a match is found, we save the concepts by appending them to the list  $LC_h$ ; this step is repeated for each main noun.

In the second stage (i.e., similarity matrix construction), first, hashtags associated with an empty list of concepts are discarded; in other words, hashtags that did not match any Wordnet entry, either by themselves or by using word segmentation technique, and also had no entry found in Wikipedia are discarded. Then, using the remaining hashtags (each of whose  $LC_h$  contains at least one concept in it), semantic similarity is calculated between each pair of them. Any ontology-based measure can be used, and Wu-Palmer measure [18] (see Section 2.1) is used in our work to stay consistent with the original work by Vicient and Moreno [16].

Specifically, the similarity between two hashtags,  $h_i$  and  $h_j$ , is calculated as the maximum pairwise similarity (based on the Wu-Palmer measure) between one set of concepts in  $LC_{h_i}$  and another set of concepts in  $LC_{h_j}$ . Calculating the similarity this way is expected to find the correct sense of hashtag (among all the sense/concepts in  $LC_h$ ).

Finally, in the third stage (i.e., clustering), any clustering algorithm can be used to cluster hashtags based on the similarity matrix obtained in the second stage. As mentioned earlier, in this paper we use hierarchical clustering which was used in the original work by Vicient and Moreno [16].

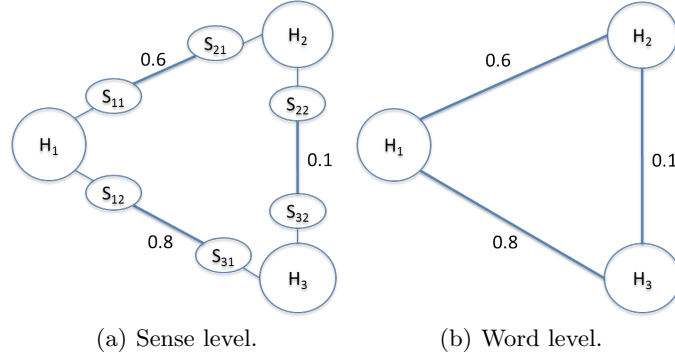
## 4 Sense-Level Semantic Hashtag Clustering

In this section, we describe the enhancement made to the word-level semantic hashtag clustering and showcase its positive impact using a toy example. Both Stage 1 (i.e., semantic grounding) and Stage 3 (i.e., clustering) of the sense-level semantic clustering algorithm are essentially the same as those in the word-level semantic clustering algorithm (see Algorithm 1 in Section 3). So, here, we discuss only Stage 2 (i.e., similarity matrix construction) of the algorithm, with a focus on the difference in the calculation of maximum pairwise similarity.

#### 4.1 Similarity matrix construction

**Word-level versus sense-level similarity matrix** As mentioned in Section 3, the similarity between two hashtags  $h_i$  and  $h_j$  is defined as the maximum pairwise similarity between one set of senses in  $LC_{h_i}$  and another set of senses in  $LC_{h_j}$ . (Recall that  $LC_h$  denotes a list of senses retrieved from Wordnet to semantically ground a hashtag  $h$ .) This maximum pairwise similarity is an effective choice for disambiguating the sense of a hashtag and was used to achieve a positive effect in the word-based approach by Vicent and Moreno [16].

However, we have observed many instances where a hashtag word is polysemic (i.e., has multiple senses) and it introduces an error in the clustering result. That is, the word-level algorithm does not distinguish among different senses of the same word when constructing a similarity matrix and, as a result, two hashtags are misjudged to be semantically similar (because they are similar to a third hashtag in two different senses) and are included in the same cluster. Moreover, a false triangle that violates the triangular inequality property may be formed at the word level. (Note this property is required of any distance metric like Wu-Palmer.) See Figure 2 for an illustration. As its side effect, we have observed that a cluster tends to be formed centered around a hashtag that takes on multiple senses.



(Edge weights denote similarity values (similarity =  $1 - \text{distance}$ ). Assume the minimum similarity threshold is 0.5. Then, at the sense level (a), *two* clusters ( $\{H_1, H_2\}, \{H_1, H_3\}$ ) should be formed because  $H_2$  and  $H_3$  are not similar (note  $0.1 < 0.5$ ), but, at the word level (b), *one* cluster  $\{H_1, H_2, H_3\}$  is formed because it appears as if  $H_2$  and  $H_3$  were similar via  $H_1$ . Moreover, the false triangle that appears to be formed at the word level violates the triangular inequality property because  $\text{dist}(H_1, H_2) + \text{dist}(H_1, H_3) < \text{dist}(H_2, H_3)$ .)

**Fig. 2.** An illustration of clustering at the word level versus sense level.

Thus, we chose to explicitly record the sense in which a hashtag is close to another hashtag when constructing a similarity matrix. This sense-level handling of hashtag semantic distance helps us ensure that the incorrect clustering problem of word-level clustering does not happen. Accordingly, it avoids the formation of clusters that are centered around a hashtag that has multiple senses.

**Word-level similarity matrix construction** Algorithm 2 outlines the steps of calculating maximum pairwise similarity between hashtags in the word-level algorithm. One maximum pairwise similarity value is calculated for each pair of hashtags semantically grounded in the previous stage (i.e., Stage 1) and is entered into the similarity matrix. The similarity matrix size is  $|H|^2$ , where  $|H|$  is the number of

hashtags that have at least one sense (i.e., nonempty  $LC_h$ ). Note that the pairwise similarity comparison is still done at the sense level, considering all senses of the hashtags that are compared.

Input: set  $H$  of hashtags  $h$  with nonempty  $LC_h$ .

Output: pairwise hashtag similarity matrix.

```

1 Initialize an empty similarity matrix  $\mathbf{M}[|H|, |H|]$ .
2 Initialize  $maxSim$  to 0.
3 for each pair  $(h_i, h_j)$  of hashtags in  $H$  do
4     // Calculate the maximum pairwise similarity between  $h_i$  and  $h_j$ .
5     for each  $s_p \in LC_{h_i}$  do
6         for each  $s_q \in LC_{h_j}$  do
7             Calculate the similarity  $sim$  between  $s_p$  and  $s_q$ .
8             if  $sim > maxSim$  then
9                 Update  $maxSim$  to  $sim$ .
10            end
11        end
12    end
13    Enter  $maxSim$  into  $\mathbf{M}[i, j]$ .
14 end

```

**Algorithm 2:** Word-level construction of semantic similarity matrix.

**Sense-level similarity matrix construction** Algorithm 3 outlines the steps of constructing a similarity matrix in the sense-level algorithm. Unlike the case of the word-level algorithm, entries in the similarity matrix are between senses that make maximum similarity pairs between a pair of hashtags. Since these senses are not known until the maximum pairwise similarity calculations are completed, the construction of the similarity matrix is deferred until then. In the first phase (Lines 2~16), for each pair of hashtags, the algorithm saves the pair of senses  $(h_i.s_p, h_j.s_q)$  in the maximum similarity pair and the maximum similarity value in the list  $LH_s$ . Then, in the second phase (Lines 18~22), for each triplet element  $(h_i.s_p, h_j.s_q, maxSim)$  in  $LH_s$ , the algorithm enters the maximum similarity value  $maxSim$  at the matrix index corresponding to the pair of senses  $(h_i.s_p, h_j.s_q)$ .

This two-phase construction of similarity matrix brings two advantages. First, it enables the algorithm to use exactly the needed number of matrix entries for those senses that are *distinct* among all senses that constitute pairwise maximum similarities between hashtags. The size of the matrix, therefore, is  $|\hat{S}|^2$ , where  $\hat{S}$  is the set of distinct senses in  $LH_s$  (see Lines 18~19). Second, it enables the algorithm to add exactly the needed number of entries, that is,  $|H|^2$  entries (i.e., one for each pair of hashtags (see Lines 20~22)) into a matrix of size  $|\hat{S}|^2$ , where  $|\hat{S}|^2 > |H|^2$ . (The remaining entries are initialized to 0 and remain 0, as they are for pairs of senses that do not represent maximum similarity pair between any hashtags.) Our observation is that the ratio  $|\hat{S}|/|H|$  is limited to the range of 5 to 10 for most individual hashtags, which is consistent with Vicent's statement [15] that, out of semantically-grounded 903 hashtags, almost 100 of them have only 2 senses and very few have more than 5 senses.

Since what is clustered are *hashtags*, although their similarities are measured at the sense level, a number of interesting points hold. First, we do not need to add similarities between all pairs of senses in the similarity matrix. Second, a hashtag

Input: set  $H$  of hashtags  $h$  with nonempty  $LC_h$ .

Output: pairwise hashtag similarity matrix.

```

1 Create an empty list  $LH_s$  of (hashtag sense pair, pairwise maximum similarity).
2 for each pair  $(h_i, h_j)$  of hashtags in  $H$  do
3   // Calculate the maximum pairwise similarity between  $h_i$  and  $h_j$ .
4   Initialize  $maxSim$  to 0.
5   Initialize  $maxSimPair$  to (null, null).
6   for each  $s_p \in LC_{h_i}$  do
7     for each  $s_q \in LC_{h_j}$  do
8       Calculate the similarity  $sim$  between  $s_p$  and  $s_q$ .
9       if  $sim > maxSim$  then
10        Update  $maxSim$  to  $sim$ .
11        Update  $maxSimPair$  to  $(h_i.s_p, h_j.s_q)$ .
12      end
13    end
14  end
15  Add  $(maxSimPair, maxSim)$  to  $LH_s$ .
16 end
17 // Construct the similarity matrix.
18 Count the number  $|\hat{S}|$  of distinct hashtag senses in  $LH_s$ .
19 Initialize a similarity matrix  $\mathbf{M}[|\hat{S}|, |\hat{S}|]$  as a  $\mathbf{0}$  matrix.
20 for each triplet  $(h_i.s_p, h_j.s_q, maxSim)$  in  $LH_s$  do
21   Update the  $\mathbf{M}[m, n]$  to  $maxSim$ , where  $(m, n)$  is the matrix index for  $(h_i.s_p,$ 
    $h_j.s_q)$ .
22 end

```

**Algorithm 3:** Sense-level construction of semantic similarity matrix.

may appear in multiple clusters, where each cluster is formed based on distinct senses of the hashtag, and therefore the resulting clusters are *overlapping*.

## 4.2 A toy example

To demonstrate the merit of clustering at the sense level as opposed to the word level, we made a toy set of hashtags and ran the metadata-based semantic clustering algorithm at both the word level and the sense level. The hashtags used are #date, #august, #tree, and #fruit. From Wordnet, we found that there were 3 senses associated with the word august, 13 senses with date, 5 senses with fruit, and 7 senses with tree.

Using the Wu-Palmer similarity measure (Section 2.1) at the word level, we obtained the distance matrix on the right. Then, to perform clustering using this distance matrix as the input, we used both the single-linkage and UP-GMA (Section 2.3) as the distance measure between newly formed clusters and set the distance threshold for extracting flat clusters from hierarchical clusters to 0.5.

Table 1 shows the clusters obtained using the word-level clustering. We see that #august, #date, and #fruit are included in the same cluster in both cases of the distance measure. This example demonstrates a case in which #date takes on multiple sense identities and

Hashtag	august	date	fruit	tree
august	0.000	0.200	0.500	0.667
date	0.200	0.000	0.100	0.400
fruit	0.500	0.100	0.00	0.556
tree	0.667	0.400	0.556	0.000

Hashtag	Cluster using single-linkage	Cluster using UPGMA
august	1	1
date	1	1
fruit	1	1
tree	1	2

**Table 1.** Cluster assignment at word level.

glues together #august and #fruit in the same cluster at the word level although these two are not similar at the sense level, as shown next.

Now, using the sense-level clustering, out of a total of 28 senses associated with the four hashtags, the algorithm picked 10 senses shown in Table 2. These 10 senses were picked as a result of maximum pairwise similarity calculations between two sets of senses belonging to each pair of hashtags. (With 4 hashtags, there are a maximum of 12 ( $= 2 \times 6$ ) senses that can be obtained for 6 ( $= C(4, 2)$ ) maximum similarity pairs, and in this example case, there were duplicate senses, consequently giving 10 distinct senses.) As mentioned earlier, each of these senses represents the semantics of the hashtag word it belongs to, and thus makes an entry into the similarity (or distance) matrix input to the hierarchical clustering algorithm.

Sense	Semantics
august.n.01	the month following July and preceding September
august.a.01	of or befitting a lord
corner.v.02	force a person or animal into a position from which he can not escape
date.n.02	a participant in a date
date.n.06	the particular day, month, or year (usually according to Gregorian calendar) that an even occurred
date.n.08	sweet edible fruit of the date palm with single long woody seed
fruit.n.01	the ripened reproductive body of a seed plant
fruit.v.01	cause to bear fruit
tree.n.01	a tall perennial woody plant having a main trunk and branches forming a distinct elevated crown; includes both gymnosperms and angiosperms
yield.n.03	an amount of product

(‘n’ stands for noun, ‘v’ for verb and ‘a’ for adjective.)

**Table 2.** Senses and their semantics (source: Wordnet).

The distance matrix obtained from the 10 senses is shown in Figure 3. The numbers in bold face are the maximum similarity values entered. Note that distance 1.000 means similarity 0.000.

Hashtag sense		august.n.01	august.a.01	corner.v.02	date.n.02	date.n.06	date.n.08	fruit.n.01	fruit.v.01	tree.n.01	yield.n.03
	Hashtag	august	august	tree	date	date	date	fruit	fruit	tree	fruit
august.n.01	august	0.000	1.000	1.000	1.000	<b>0.200</b>	1.000	1.000	1.000	1.000	1.000
august.a.01	august	1.000	0.000	<b>0.667</b>	1.000	1.000	1.000	1.000	<b>0.500</b>	1.000	1.000
corner.v.02	tree	1.000	<b>0.667</b>	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
date.n.02	date	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	<b>0.400</b>	1.000
date.n.06	date	<b>0.200</b>	1.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	1.000
date.n.08	date	1.000	1.000	1.000	1.000	1.000	0.000	<b>0.100</b>	1.000	1.000	1.000
fruit.n.01	fruit	1.000	1.000	1.000	1.000	1.000	<b>0.100</b>	0.000	1.000	1.000	1.000
fruit.v.01	fruit	1.000	<b>0.500</b>	1.000	1.000	1.000	1.000	1.000	0.000	1.000	1.000
tree.n.01	tree	1.000	1.000	1.000	<b>0.400</b>	1.000	1.000	1.000	1.000	0.000	<b>0.556</b>
yield.n.03	tree	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	<b>0.556</b>	0.000

**Fig. 3.** Distance matrix in the toy example.

Table 3 shows the resulting cluster assignments. (The outcome is the same for both distance measures, which we believe is coincidental.) We see that #august and #date are together in the same cluster and so are #date and #fruit but, unlike the word-level clustering result, the three of #august, #date, and #fruit are not

Hashtag	Hashtag sense	Cluster using single-linkage	Cluster using UPGMA
date	date.n.02	1	1
tree	tree.n.01	1	1
fruit	yield.n.03	2	2
fruit	fruit.v.01	3	3
august	august.a.01	3	3
tree	corner.v.02	4	4
fruit	fruit.n.01	5	5
date	date.n.08	5	5
august	august.n.01	6	6
date	date.n.06	6	6

**Table 3.** Cluster assignment at the sense level.



altogether in the same cluster. This separation is because, at the sense level, #date can no longer take on multiple identities as it did at the word level.

## 5 Evaluation

The focus of evaluating the sense-level clustering algorithm is on the accuracy gained from the finer granularity of semantics compared with the word-level clustering algorithm. To highlight this focus, we conducted two experiments distinguished by the choice of control on the semantics of hashtags used – the semantics are not controlled in one experiment and are controlled in the other.

All algorithms were implemented in Python and the experiments were performed on a computer with OS X operating system, 2.6 GHz Intel Core i5 processor, and 8 GB 1600 MHz DDR3 memory.

### 5.1 Experiment setup

**Performance metric** We use *f-score*, which is commonly used in conjunction with recall and precision to evaluate clusters in reference to ground truth clusters, as the accuracy metric. In our evaluation, the f-score is calculated for each pair of a cluster in the ground truth cluster set and a cluster in the evaluated algorithm’s output cluster set. Then, the final f-score resulting from the comparison of the two cluster sets is obtained in two different ways, depending on the purpose of the evaluation. For the purpose of evaluating individual output clusters, the pairwise maximum (i.e., “best match”) f-score, denoted as  $f^m$ -score, is used as the final score. Given a ground truth cluster  $G_i$  matched against an output cluster set  $\mathbf{C}$ , the  $f^m$ -score is obtained as

$$f^m\text{-score}(G_i, \mathbf{C}) = \max_{C_j \in \mathbf{C} \wedge f\text{-score}(G_i, C_j) > 0} f\text{-score}(G_i, C_j) \quad (4)$$

where the pairwise matching is one-to-one between  $\mathbf{G}$  and  $\mathbf{C}$ .

On the other hand, for comparing overall accuracy of the entire set of clusters, the weighted average of pairwise maximum f-scores, denoted as  $f^a$ -score, is used instead. Given a ground truth cluster set  $\mathbf{G}$  and an output cluster set  $\mathbf{C}$ , the  $f^a$ -score is calculated as

$$f^a\text{-score}(\mathbf{G}, \mathbf{C}) = \frac{\sum_{G_i \in \mathbf{G}} (f^m\text{-score}(G_i, \mathbf{C}) \times |G_i|)}{\sum_{G_i \in \mathbf{G}} |G_i|} \quad (5)$$

**Distance threshold** The distance threshold for determining flat clusters in hierarchical clustering was set using the “best result” approach. That is, we tried both distance measures (i.e., single-linkage and UPGMA) and different distance threshold values and picked the measure and value that produced the best result based on the weighted average f-score measure.

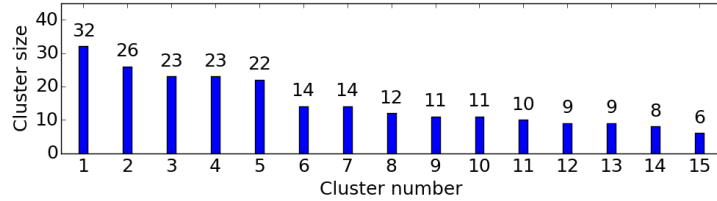
**Qualified output clusters** The clustering output shows a large number of small clusters, many of them including only one or two hashtags. Thus, for a given ground truth cluster, we consider the best matching output cluster only if it contains at least 3 hashtags and the  $f^m$ -score is greater than 0.1.

### 5.2 Experiment 1: uncontrolled hashtag semantics

In this experiment, hashtags are collected without any control over the semantics, i.e., “indiscriminately”, from tweet messages.

To build the ground truth, we manually gathered 2,910 tweets from the Symplur web site (www.symplur.com) – the same number of Symplur tweets was also used in

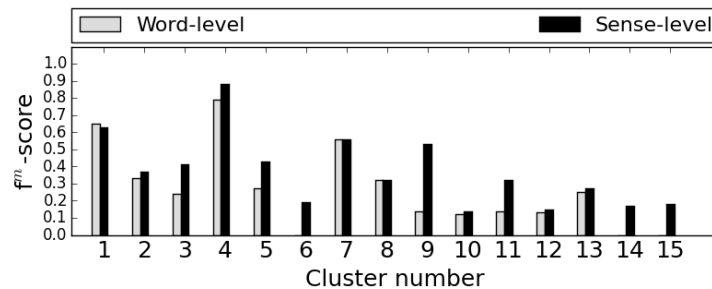
the evaluation of word-level clustering by Vicent and Moreno [16]. There were 1,010 unique hashtags in the 2,910 tweets. We then manually annotated the semantics of the 1,010 hashtags to choose 230 hashtags and classified them into 15 clusters. The remaining hashtags were classified as noise. Figure 4 shows the sizes of the resulting ground truth clusters.



**Fig. 4.** Sizes of ground truth clusters in the uncontrolled hashtag experiment.

In the hierarchical clustering, distance threshold value that gave the best result was 0.4 when the UPGMA measure was used for both sense-level and word-level.

Figure 5 shows the accuracies achieved by semantic clustering at the word-level and the sense-level using the uncontrolled hashtag from the Symplur dataset. Table 4 shows more details, including precision and recall for individual clusters. From the results we see that every sense-level cluster outperforms the word-level counterpart (except cluster 1 due to rounding-off difference). Particularly, the  $f^m$ -scores are zero for word-level clusters 6, 14, and 15, thus bringing the performance gain to “infinity”. (Word-level clustering did not generate any qualified matching cluster for the ground truth clusters 6, 14, and 15.) Further, when all 15 clusters are considered together, the weighted average of maximum pairwise f-scores,  $f^a$ -score, is 0.43 for sense-level clustering and 0.34 for word-level clustering – a 26% gain.



**Fig. 5.** Maximum pairwise f-scores of output clusters for word-level and sense-level in the uncontrolled hashtag experiment.

### 5.3 Experiment 2: controlled hashtag semantics

In this experiment, hashtags are collected with controlled semantics, that is, “discriminated” based on their semantics.

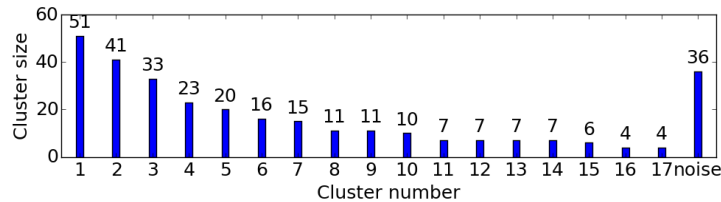
The ground truth clusters were prepared using a combination of two tweet datasets. One dataset has 2.5 million tweets collected through Twitter REST API. This dataset contained 708 hashtags that had 20 or more tweets associated with them. We selected from the 708 hashtags approximately 150 hashtags based on the topics they addressed. Then, we started with the 15 clusters in the ground truth from the Symplur dataset used in Experiment 1 (see Section 5.2) and randomly picked from them one cluster at a time, merging them, until the total number of hashtags in the selected clusters reached approximately 150. Then, we combined

Ground truth clusters		Sense-level clusters				Word-level clusters			
Id	Size	Recall	Precision	$f^m$ -score	Size	Recall	Precision	$f^m$ -score	Size
1	32	0.63	0.65	0.63	31	0.63	0.67	0.65	30
2	26	0.35	0.39	0.37	23	0.31	0.35	0.33	23
3	23	0.39	0.43	0.41	21	0.35	0.19	0.24	43
4	23	0.91	0.84	0.88	25	0.83	0.76	0.79	25
5	22	0.41	0.45	0.43	20	0.41	0.20	0.27	44
6	14	0.21	0.18	0.19	17	n/a	n/a	n/a	n/a
7	14	0.64	0.50	0.56	18	0.64	0.50	0.56	18
8	12	0.25	0.43	0.32	7	0.50	0.24	0.32	25
9	11	0.82	0.39	0.53	23	0.82	0.08	0.14	118
10	11	0.18	0.11	0.14	18	0.09	0.17	0.12	6
11	10	0.40	0.27	0.32	15	0.50	0.08	0.14	59
12	9	0.11	0.25	0.15	4	0.11	0.17	0.13	6
13	9	0.22	0.33	0.27	6	0.22	0.29	0.25	7
14	8	0.13	0.25	0.17	4	n/a	n/a	n/a	n/a
15	6	0.17	0.20	0.18	5	n/a	n/a	n/a	n/a

$f^a$ -score is 0.43 for sense-level clusters and 0.34 for word-level clusters.

**Table 4.** Details of gold standard test results in the uncontrolled hashtag experiment.

the two sets of approximately 150 hashtags each to form one set, which contained a total of 309 hashtags. Figure 6 shows the sizes of the resulting ground truth clusters.



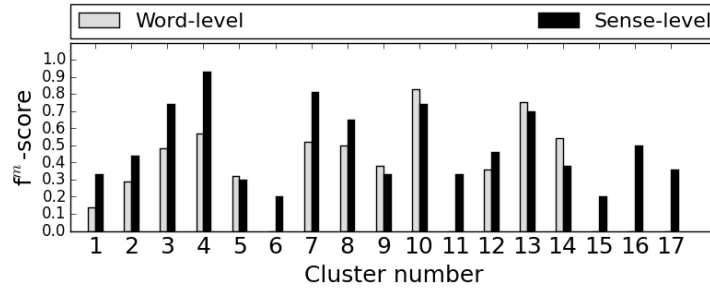
**Fig. 6.** Sizes of ground truth clusters in the controlled hashtag experiment.

For the hierarchical clustering, the distance threshold values that gave the best results were 0.3 at the word level and 0.4 at the sense level, respectively, both using the UPGMA measure.

Figure 7 shows the accuracies achieved by the semantic clustering at the word-level and the sense-level using the controlled hashtags. Table 5 shows more details. Sense-level clusters outperform word-level clusters in 12 of 17 clusters and are very close runner-up’s in the remaining 5 clusters. Compared with Experiment 1, the 309 hashtags with controlled semantics gave sense-level clustering 1,293 unique senses to work with to produce best match clusters with higher  $f^m$ -scores to a larger number of ground truth clusters. In contrast, word-level clustering did not generate any qualified matching cluster for the ground truth clusters 6, 11, 15, 16 and 17, and hence the accuracy suffered significantly. Consequently, when all 17 clusters are considered, the  $f^a$ -score is 0.50 for sense-level clustering and 0.34 for word-level – a gain of 47%. Notably, in this Experiment 2, the sense-level outperforms the word-level by a substantially larger margin than in Experiment 1 because hashtags were hand-picked deliberately based on their semantics, while the word-level performs the same as in Experiment 1 because word-level is “oblivious” to the exact (i.e., sense-level) semantics of hashtags.

## 6 Related Work

There are several works on semantic clustering of hashtags that focused on the contextual semantics of hashtags [6] [8] [10] [12, 13] by using the bag of words model to represent the texts accompanying a hashtag. Tsur et al. [12, 13] and Muntean et al. [6] appended tweets that belonged to each unique hashtag into a unique document called “virtual document”. These documents were then represented as vectors in the vector space model. Rosa et al. [8] used hashtag clusters to achieve



**Fig. 7.** Maximum pairwise f-scores of output clusters for word-level and sense-level in the controlled hashtag experiment.

Ground truth clusters		Sense-level clusters				Word-level clusters			
Id	Size	Recall	Precision	f <sup>m</sup> -score	Size	Recall	Precision	f <sup>m</sup> -score	Size
1	51	0.25	0.45	0.33	29	0.08	0.80	0.14	5
2	41	0.29	0.86	0.44	14	0.20	0.57	0.29	14
3	33	0.61	0.95	0.74	21	0.39	0.62	0.48	21
4	23	0.91	0.95	0.93	22	0.52	0.63	0.57	19
5	20	0.30	0.30	0.30	20	0.20	0.80	0.32	5
6	16	0.13	0.50	0.20	4	n/a	n/a	n/a	n/a
7	15	0.73	0.92	0.81	12	0.47	0.58	0.52	12
8	11	0.91	0.50	0.65	20	0.64	0.41	0.50	17
9	11	0.27	0.43	0.33	7	0.27	0.60	0.37	5
10	10	1.00	0.59	0.74	17	1.00	0.71	0.83	14
11	7	0.29	0.40	0.33	5	n/a	n/a	n/a	n/a
12	7	0.43	0.50	0.46	6	0.29	0.50	0.36	4
13	7	1.00	0.54	0.70	13	0.86	0.67	0.75	9
14	7	0.43	0.33	0.38	9	0.43	0.75	0.55	4
15	6	0.17	0.25	0.20	4	n/a	n/a	n/a	n/a
16	4	0.50	0.50	0.50	4	n/a	n/a	n/a	n/a
17	4	0.50	0.29	0.36	7	n/a	n/a	n/a	n/a

f<sup>a</sup>-score is 0.50 for sense-level clusters and 0.34 for word-level clusters.

**Table 5.** Details of gold standard test results in the controlled hashtag experiment.

topical clustering of tweets, where they compared the effects of expanding URLs found in tweets. Stilo and Paola [10] clustered hashtag “senses” based on their temporal co-occurrence with other hashtags. The term “sense” in their work is different from the lexical sense used in this paper.

Lacking the ability to form lexical semantic sense-level clusters of hashtag has been a major shortcoming of the current approaches. To the best of our knowledge, the work by Vicent and Moreno [16] is the only one that opened research in this direction. They used Wordnet and Wikipedia as the metadata source for clustering hashtags at the word level.

## 7 Conclusion

In this paper, we enhanced the current metadata-based semantic hashtag clustering algorithm by determining semantic similarity between hashtags at the *sense* level as opposed to the word level. This sense-level decision on clustering avoids incorrectly putting hashtags of different senses in the same cluster. The result was significantly higher accuracy of semantic clusters without increasing the complexities of the algorithm in practice. Gold standard tests showed an overall gain of 26% (in case of uncontrolled hashtag semantics) and 47% (in case of controlled hashtag semantics) in the weighted average of maximum pairwise f-scores.

For the future work, new metadata sources can be added to provide the metadata-based semantic hashtag clustering algorithm with more abilities. For example, a crowdsourced website like Urban Dictionary ([www.urbandictionary.com](http://www.urbandictionary.com)) that specializes in informal human communication can be a helpful metadata source for

decoding lexical semantics of hashtags. Internet search engines also provide rich information on the semantics of hashtags. Additionally, online translation service like Google Translate ([translate.google.com](http://translate.google.com)) can be a good source for understanding hashtags of a different language as well as correcting spelling errors and expanding abbreviations.

## References

1. Sandjai Bhulai et al. “Trend Visualization on Twitter: What’s Hot and What’s Not?” In: *1st International Conference on Data Analytics*. 2012, pp. 43–48.
2. Joanna Costa et al. “Defining Semantic Meta-Hashtags for Twitter Classification”. In: *LNCS 7824* (2013), pp. 226–235.
3. Ali Javed and Byung Suk Lee. “Sense-Level Semantic Clustering of Hashtags in Social Media”. In: *the 3rd Annual International Symposium on Information Management and Big Data*. Sept. 2016.
4. Ryan Kelly. *Twitter Study Reveals Interesting Results About Usage - 40% is Pointless Babble*. URL: <http://pearanalytics.com/blog/2009/twitter-study-reveals-interesting-results-40-percent-pointless-babble/> (visited on 05/10/2016).
5. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. Chap. 17.
6. Cristina Muntean, Gabriel Morar, and Darie Moldovan. “Exploring the Meaning behind Twitter Hashtags through Clustering”. In: *LNBIP 127* (2012), pp. 231–242.
7. Suzi Park and Hyopil Shin. “Identification of Implicit Topics in Twitter Data Not Containing Explicit Search Queries”. In: *25th International Conference on Computational Linguistics*. 2014, pp. 58–68.
8. Kevin Dela Rosa, Rushin Shah, and Bo Lin. “Topical Clustering of Tweets”. In: *3rd Workshop on Social Web Search and Mining*. July 2011, pp. 133–138.
9. Hassan Saif, Yulan He, and Harith Alani. “Semantic Sentiment Analysis of Twitter”. In: *11th International Conference on The Semantic Web - Volume Part I*. 2012, pp. 508–524.
10. Giovanni Stilo and Paola Velardi. “Temporal Semantics: Time-Varying Hashtag Sense Clustering”. In: *19th International Conference on Knowledge Engineering and Knowledge Management*. Nov. 2014, pp. 563–578.
11. Peter Teufl and Stefan Kraxberger. “Extracting Semantic Knowledge from Twitter”. In: *3rd IFIP WG 8.5 International Conference on Electronic Participation*. 2011, pp. 48–59.
12. Oren Tsur, Adi Littman, and Ari Rappoport. “Efficient Clustering of Short Messages into General Domains”. In: *7th International AAAI Conference on Weblogs and Social Media*. 2013.
13. Oren Tsur, Adi Littman, and Ari Rappoport. “Scalable Multi Stage Clustering of Tagged Micro-messages”. In: *International Conference on World Wide Web*. Apr. 2012, pp. 621–622.
14. *Usage Statistics*. URL: <http://www.internetlivestats.com/twitter-statistics/>.
15. Carlos Vicient. “Moving Towards The Semantic Web: Enabling New Technologies through the Semantic Annotation of Social Contents”. PhD thesis. Universitat Rovira I Virgili, Dec. 2014.
16. Carlos Vicient and Antonio Moreno. “Unsupervised Semantic Clustering of Twitter Hashtags”. In: *21st European Conference on Artificial Intelligence*. Aug. 2014, pp. 1119–1120.
17. Xiaolong Wang et al. “Topic Sentiment Analysis in Twitter: a Graph-based Hashtag Sentiment Classification Approach”. In: *20th ACM Conference on Information and Knowledge Management*. 2011, pp. 1031–1040.
18. Zhibiao Wu and Martha Palmer. “Verbs Semantics and Lexical Selection”. In: *32nd Annual Meeting on Association for Computational Linguistics*. 1994, pp. 133–138.