A *hashtags* dictionary from crowdsourced definitions

Mérième Ghenname
LT2C, Telecom Saint-Etienne
Université Jean Monnet
ghenname.merieme-@gmail.com

Julien Subercaze
LT2C, Telecom Saint-Etienne
Université Jean Monnet
julien.subercaze@univ-st-etienne.fr

Christophe Gravier
LT2C, Telecom Saint-Etienne
Université Jean Monnet
christophe.gravier@univ-st-etienne.fr

Frédérique Laforest
LT2C, Telecom Saint-Etienne
Université Jean Monnet
frederique.laforest@univ-st-etienne.fr

Mounia Abik
LeRMA, ENSIAS
Université Mohammed V Souissi
Rabat, Morocco
abmounia@gmail.com

Rachida Ajhoun
LeRMA, ENSIAS
Université Mohammed V Souissi
Rabat, Morocco
ajhoun@gmail.com

ABSTRACT
Hashtags are user-defined terms used on the Web to tag messages like microposts, as featured on Twitter. Because a hashtag is a textual word, its representation does not convey all the concepts it embodies. Several online dictionaries have been manually and collaboratively built to provide natural language definitions of hashtags. Unfortunately, these dictionaries in their rough form are inefficient for their inclusion in automatic text processing systems. As hashtags can be polysemic, dictionaries are also agnostic to collision of hashtags. This paper presents our approach for the automatic structuration of hashtags definitions into synonym rings. We present the output as a so-called *folksionary*, i.e. a single integrated dictionary built from everybody's definitions. For this purpose, we achieved a semantic-relatedness clustering to group definitions that share the same meaning.

Categories and Subject Descriptors
H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms
Algorithms, Clustering

Keywords
Hashtags, Social Network, Natural Language Processing, Clustering

1. PROBLEM DEFINITION
Users' writings and data on social networks are growing exponentially over time. They become hardly exploitable. In order to bind and easily find what they produce within this large mass of data, users label content using hashtags. Hashtags have become a lightweight solution to classify and search information on the Web 2.0 and 3.0.

Unfortunately, a hashtag is at best a composed word, and at worst a neologism. It is not a piece of information by itself. The primary information is the association (the *tagging* relation) that exists between a hashtag and a resource. It is however important to gain more knowledge on hashtags. The primary motivation is to enhance hashtag-based services. Examples include query expansion for hashtag queries using hashtag subsumption relationships, or hashtag recommender systems for boosting the tagging process. The associated learning process encompasses discovering hashtag-related concepts in an external knowledge base [8], and/or learning the relationships between them [11]. In the literature, this problem is usually addressed using the context involving a hashtag [15, 13]. For instance in a textual resource, this context is the terms window surrounding the hashtag. While this provides an information source to enrich hashtags with related bag-of-words, it suffers from several drawbacks. Firstly, the tag context is noisy, given that a tag may have been associated to hundreds of thousands or even millions resources. Moreover, the tremendous amount of data makes it impossible to retrieve all resources associated to a given hashtag. These two issues make the context of use of a hashtag incomplete and noisy. Enriching tags has not only become a knowledge discovery issue, but it is also a problem for the end-user. So, end-users feel the need to define the hashtags they use, for reusability and explanation purpose. For this, several web services are available so that any user can publish her own definition of a hashtag. One can cite *Tagdef.com* or *Hashtags.org.*

Our intuition is that such crowd-sourced services may turn out to be interesting sources of information to gain knowledge on hashtags. However their lack of structure compared to usual external dabatases used in IR or NLP (mainly Wordnet or DBPedia [1]) restricts the scope of possibilities offered by the service. We attempt to introduce a first structuration of crowd-sourced hashtags dictionaries using state of the art NLP and clustering techniques. If one refers to Wordnet [2], he could expect to introduce the same kind of relationships such as super-subordinate (hypernymy, meronymy,

[1] http://dbpedia.org/About [2] wordnet.princeton.edu

synonymy. . .). In this paper, we restrict the scope of our work to the automatic generation of different synsets for hashtags with several distinct meanings.

We tackle the following problem : given a user-generated hashtags dictionary, how to structure this list of definitions similarly to a usual dictionary? More precisely, for each hashtag of the dictionary how to group definitions with the same meaning? The resulting definitions partition can be serialized into a human readable document we call a *folksionary.* We coin the term folksionary as a porte-manteau from terms folks and dictionary.

This paper is organized as follows. Section 2 presents our approach for building a folksionary. Section 3 presents the prototype we have built and an evaluation of the results obtained. Section 4 concludes and discusses future works.

2. APPROACH FOR A *FOLKSIONARY*

In this section we present an approach that provides a clustering of user-generated definitions into different senses, over any dataset of words along with their user-generated definitions in natural English.

2.1 Formalisation

Let W be a set of words. For each $w \in W$, we define $D(w)$ the set of definitions for w and $S(w)$ the set of possible senses for w. We denote $d_{w,i}$ the i-th definition of the word w, where $i \in [1, |D(w)|]$.

We use the function *employed as* denoted $\mathcal{E}$ that relates each definition of a word to a sense of the same word as follows :

$$\begin{array}{ll} \mathcal{E}: & D(w) \longrightarrow S(w) \\ & \forall d \in D(w), \exists s \in S(w),\ \mathcal{E}(d) = s \end{array} \tag{1}$$

Definition 1. *Function $\mathcal{E}$ is a surjective function (c.f. Equation 1), therefore every sense of every word in the dictionay consists of at least one user-generated definition.*

Definition 2. *$S(w)$ is a partition of $D(w)$, such as every user-generated definition of a word w belongs to exactly one sense $s \in S(w)$, which means :*

$$\begin{cases} \bigcup s_{w,i} \in S(w) = D(w) \\ \forall s_1, s_2 \in S(w), s_1 \neq s_2 \Rightarrow s_1 \cap s_2 = \emptyset \end{cases} \tag{2}$$

We formalize the similarity matrix $Dist(w)$, with a normalized matrix which expresses the distances, taken pairwise, of a set of definitions for a given hashtag:

$$Dist(w) = (dist(d_{w,i}, d_{w,j}))_{1 \leq i,j \leq |D(w)|} \tag{3}$$

2.2 Process for building a folksionary

To build a folksionary, we perform a four-steps process. First, we crawl hashtags definitions from online services. Secondly, for each hashtag, we perform a pairwise comparison of its definitions by computing a distance between pairs of definitions. At third step, we apply a clustering algorithm for each hashtag in order to group its definitions into similar meaning clusters. Lastly, we export these results under the form of a human-readable document with a look very close to a standard dictionary. Figure 1 illustrates this approach. The following sections, detail these four steps.

2.2.1 Crawl hashtags definitions

This step populates W as well as $D(w)$, $\forall w \in W$.

Different sources of data on the Web contain users written definitions of hashtags in natural language. For instance, *Tagdef.com* or *Hashtags.org* are well-known online hashtags dictionaries. In first step, we crawl hashtags and their definitions from such sources. The scrapping process extracts definitions from each given page and, using a language classifier keeps only english definitions.

2.2.2 Distance between hashtags

The objective of this step is to populate $Dist(w)$, $\forall w \in W$.

User-generated definitions for a given hashtag can be redundant, i.e. some definitions can describe the same meaning. Our goal in this step is to measure the semantic-relatedness between definitions for the clustering phase (section 2.2.3).

In the literature, the traditional approach to compare two sentences relies on the co-occurence frequency of terms employed in the different natural language sentences [12, 5]. These approaches are limited to the strict co-occurence of the same terms in the definitions. But crowd-sourced hashtags definitions are populated by different users, using heterogeneous terms, neologisms and abbreviations.

We need an external knowledge base, to take into account proximity between terms in the metric between hashtags definitions. This issue is referred as semantic-relatedness for the word sense desambiguisation problem [10].

Among techniques involving an external knowledge base, the Extended Lesk algorithm has proven to be one of the most efficient [2]. Extended Lesk is an adaptation of the Lesk [9] using Wordnet [3] as an external knowledge base. Using the context of use (a term window) of a given target word, it selects the most plausible sense for this word from all the possible senses in Wordnet.

This algorithm is limited to the semantic-relatedness between two words. [14], propose a new approach for the semantic-relatedness between two sentences using Extended Lesk. We use Extended Lesk on each set $D(w)$ to provide the semantic-relatedness between definitions of the hashtag w under the form of a matrix. Each matrix represents the adjency matrix of a weighted graph where edges are the definitions of a hashtag, and the vertices are weighted by the distance between the two definitions.

2.2.3 Clustering of definitions

The objective of this step is to populate $S(w)$, $\forall w \in W$.

In the previous step we generate a graph providing $Dist(w)$, the distances between definitions of a hashtag. This graph

[3] http://wordnet.princeton.edu

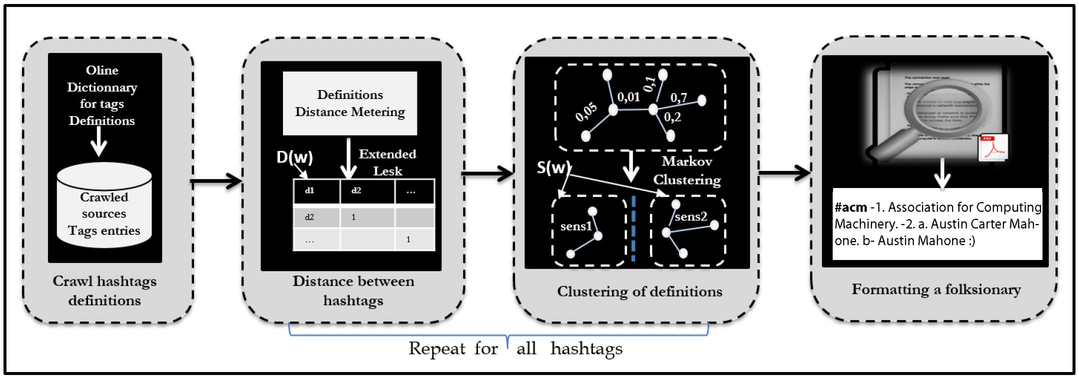

Figure 1: Approach for building a folksionary

is used to cluster hashtags based on their meanings.

In our approach we have no a priori information regarding the number of clusters. A comparative analysis has shown that the Markov Clustering algorithm (MCL) [6] is remarkably more robust than other clustering techniques [3]. It produces good clustering results mainly because the algorithm scales well with increasing graph size, it is robust against noise in graph data even if it cannot find overlapping clusters. Also we are not constrained to specify a number of clusters beforehand.

MCL interprets the matrix entries or graph edge weights as similarities. it simulates a random walk in the graph by changing iteratively the transition probabilities in adjency matrix with normalized value in [0;1]. In MCL two processes alternate: Expansion and Inflation. The expansion operator connects different regions of the graph, and the inflation operator is responsible for strengthens and weakens. Eventually, iterating expansion and inflation results in the separation of the graph into different segments. The collection of resulting segments is simply interpreted as a clustering.

Several parameters are available for tuning the mcl computing process. The most popular are inflation parameter setting for obtaining clusterings at different levels of granularity, the measure of idempotence and pruning, the maximum value considered zero for pruning operations and values for cycles.

During this step, for each hashtag, we group its definitions into units of meaning $S(w)$. We are then able to perceive to what extent each hashtag is polysemic with the cardinality of $S(w)$.

2.2.4 *Formatting a folksionary*

One of the objectives of a folksionary is to provide a new kind of dictionary to human users. Therefore we output the in-memory model of the folksionary in a format close to a traditional dictionary. This output is a PDF file that organizes hashtags entries in an alphabetic order. Each hashtag is presented with all its meanings, and we list in each meaning all the definitions that were clustered. For instance, consider the following definitions that were crawled for the hashtag *#acm* at step 1 (c.f. 2.2.1):

- "Austin Carter Mahone"
- "Association for Computing Machinery"
- "Austin Mahone :)"

We present it using a standard dictionary formatting :

#acm **-1.** Association for Computing Machinery **-2.** Austin Carter Mahone ❖ Austin Mahone :)

As shown in the previous example, two meanings were detected for the hashtag acm, one for Association for Computing Machinery (with one definition) and the other for the person named Austin Carter. This second meaning comes from two different definitions, which were grouped in the same cluster.

The different symbols are intended as the following :

- The different sense $s \in S(w)$ are separated by numbers. **-1.** $d$enotes the first meaning. **-2.** $d$enotes the second meaning, and so on.
- Definitions of the same sense $\forall d_i \in s$ with $s \in S(w)$ are separated by ❖ .

3. PROTOTYPE AND EVALUATION

This session is dedicated to our prototype and the characterization of results obtained on the folksionary. We also provide a qualitative analysis measuring the distance between the generated folksionary and a ground truth established manually.

3.1 Prototype implementation

To demonstrate our approach, we have constructed a dataset by crawling web sources. For this purpose, we have created dedicated Web scrappers using the `pjscrape` javascript library [4]. It performs a browser-like rendering, therefore we did not miss any AJAX-generated content. We used Apache Tika [7] for language filtering in order to select only english definitions. Then, we compute the distance using an inhouse developed Java version of Extended Lesk. And we

[4] `http://nrabinowitz.github.com/pjscrape/`

finally perform Markov Clustering using JavaML [1]. In this section we detail the characteristics of our folksionary and provide an evaluation.

3.2 Folksionary Characterization

We have built a folksionary by applying our approach on the aforementioned dataset. The folksionary PDF file containing all tags is available online at:`http://goo.gl/1b2Jp8`

This folksionary contains 22,738 hashtags, and a total of 28,191 definitions. Our approach identified 25,106 meanings in 28,191 definitions. Each hashtag has an average of $\sim 1,1$ meaning (SD : ∼0,45). In this folksionary, 1,731 hashtags out of 22,738 have several meanings.

Let us focus on the 1,731 tags that have been detected polysemic. Polysemic hashtags have on average ~ 2.37 meanings with a standard deviation of $\sim 0,94$. Figure 2 presents the number of tags grouped by number of meanings. For instance: 261 hashtags have three distinct meanings detected by our approach. 98,8% hashtags are polysemic with at maximum five different meanings. The last 1,2% tags with a degree of polysemy superior or equal to 6, represent a tiny portion of our folksionary and are considered as exceptions in this work. Those tags are hugely popular tags, such as *#justinbieber* where people express different, sometimes ironical definitions.

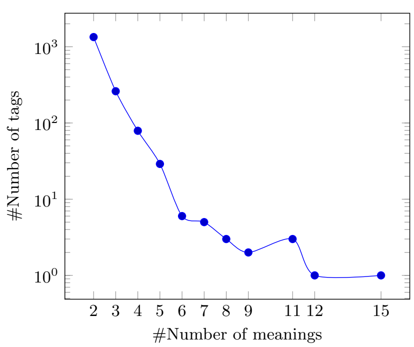

Figure 2: Number of tags grouped by number of meanings.

3.3 Evaluation

In order to complete the quantitative analysis of our folksionary, a qualitative analysis is needed. It consists in measuring the distance between the generated folksionary and the Ground Truth. The problem is the following : How to measure the effectiveness of clustering user-generated definitions into different senses ?

The primary issue lies in the lack of an evaluation framework for the clustering when the number of clusters in not known in advance. The second one is the lack of existing datasets with labelled instances, that could be used for competing with existing work. Both these limitations of the state-of-the-art lead us to build a Ground Truth dataset for evaluation, and then to develop an evaluation method, that relies on the measurement of approximate correlation[4].

3.3.1 Establishing Ground Truth

We have built an *ad hoc* Web application, and participants have manually built the ground truth by clustering hashtags' definitions into meanings. The number of definitions in the folksionary and the Ground Truth is the same, yet ordered differently. The number of meanings is chosen independently by each participant.

The web application eases enormously the manual work. To make a manual clustering, users group definitions that share the same meaning by adding a new meaning and sliding similar definitions on the same meaning.

3.3.2 Pairwise evaluation protocol

We want to evaluate the $\mathcal{E}$ function. In the following, we use the following notation:

- $\mathcal{E}_{DGT}$, the Dataset Ground Truth partitioning,
- $\mathcal{E}_{DP}$, the Dataset Prediction partitioning generated by our approach for the same dataset.

The evaluation objective is to measure how $\mathcal{E}_{DP}$ performs towards $\mathcal{E}_{DGT}$. We are using a *pairwise evaluation* for this. For all pairs of definitions (d_1, d_2) for a word w, we define the following observations:

- if d_1 and d_2 are in the same cluster both in the ground truth and in the prediction, it is a true positive (TP),
- if d_1 and d_2 are in different clusters both un the ground truth and in the prediction, it is a true negative (TN),
- if d_1 and d_2 are in the same cluster in the ground truth, but in different clusters in the prediction, it is a false negative (FN),
- if d_1 and d_2 are in different clusters in the ground truth but in the same cluster in the prediction, it is a false positive (FP).

A synoptic view on this process is as follows :

1. For each word $w \in \mathcal{W}$, enumerate all the pairs of user-generated definitions $(d_i, d_j) \in D(w) \times D(w)$ such as $i < j$,
2. Retrieve $C(s_i) = \mathcal{E}_{DGT}(d_i)$ and $C(s_j) = \mathcal{E}_{DGT}(d_j)$,
3. Retrieve $C(s'_i) = \mathcal{E}_{DP}(d_i)$ and $C(s'_j) = \mathcal{E}_{DP}(d_j)$,
4. Make observations (TP, TN, FP, or FN) depending on values of $C(s_i), C(s_j), C(s'_i), C(s'_j)$,
5. Compute a correlation measure for all words $w \in \mathcal{W}$.

We have conducted observations on the entire dataset in order to measure the distance between the Ground Truth partitioning and the automatic partitioning generated by our approach. For this purpose we have performed a straightforward evaluation using a metric adapted to our dataset.

The most classical metrics one can find in the literature are the $F_1 score$ and the Matthews Correlation Coefficient (MCC). But these coefficients can be undefined when the denominator value is zero, which happens quite often in our case. The chosen metric is then Average Conditional Probability (ACP) [4] that smoothly takes into account such a case.

ACP is defined as follows if all the sums are non-zero:

$$ACP = \frac{1}{4}\left[\frac{|TP|}{|TP|+|FN|} + \frac{|TP|}{|TP|+|FP|} + \frac{|TN|}{|TN|+|FP|} + \frac{|TN|}{|TN|+|FN|}\right] \quad (4)$$

Otherwise, ACP is the average over those conditional probabilities that are defined.

3.3.3 *Evaluation and interpretation*

Our study intends to carry out comparisons across the performance of calculated measurements, in order to interpret the clustering output, and its proximity to the Ground Truth. As outlined above, we chose a graph-based algorithms MCL for our clustering approach because it can be used for detecting clusters from different shapes without specifying the clusters number in advance. The values of some parameters must be specified by the user as input, which remains a real challenge.

In this section, experimental results on our Folksionary are presented, in order to generate the combination of parameters representing the best tuning for the algorithm. After this tuning, we conduct assessments to measure the quality of our clustering approach compared to the Ground Truth.

To achieve the first objective, several values of **gammaExp**(*inflation exponent for Gamma operator*), **maxResidual** (*maximum difference between row elements and row square sum, measure of idempotence*), and **maxZero**(*maximum value considered zero for pruning operations*) were tested (c.f. 2.2.3). We carried out our experiments for the range of the following values: maxZero $(10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7})$, maxResidual $(1, 0, 10^{-1}, 10^{-2}, 10^{-3})$, gammaExp(1.4, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20).

For each test, maxZero value is set and gammaExp value is varied with the maxResidual value in order to establish optimal values as said previously. Results substantially confirm that a good clustering requires a correct choice of parameters. The analysis clearly shows that ACP value keeps constant at **53.2%** for maxResidual = 1 and does not exceed **55.9%** for maxResidual = 0 regardless maxzero tested values. We also note that, for these maxResidual values $10^{-1}, 10^{-2}$ and 10^{-3}, the ACP value converges rapidly to very good values for small values of gammaExp while decreasing maxZero. For example with maxZero= 10^{-1} ACP remains constant at **89.2%** starting from gammaExp =6 and begins to increase at 8, 10, 14, 18, 20 for the values of maxZero $10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}$ respectively.

Results reported on Figure 3 represent the % of ACP by variant maxZero and setting maxResidual to 10^{-3}, and Figure 4 represent te % of ACP by varying maxResidual and setting maxZero to 10^{-1}.

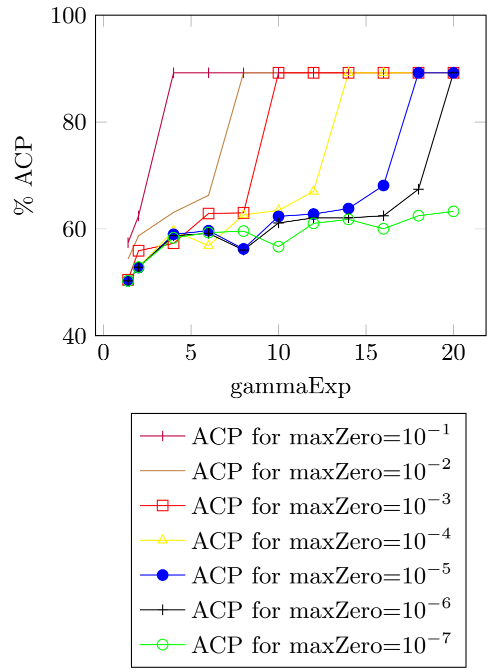

Figure 3: % ACP by variant maxZero and setting maxResidual

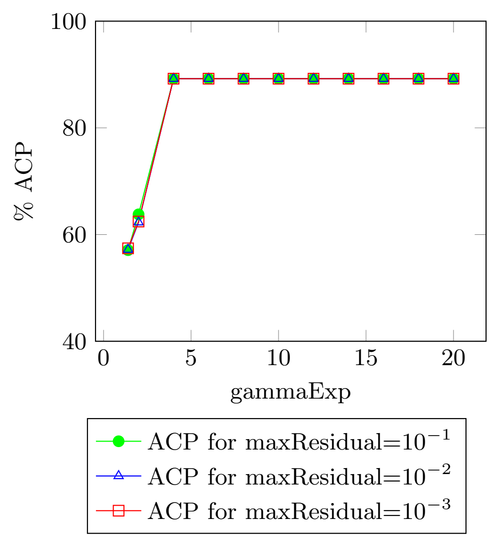

Figure 4: % ACP by variant maxResidual and setting maxZero to 10^{-1}

We conclude that the best combination of MCL parameters for our dataset is maxZero = 0.1 and maxResidual value in the interval $[10^{-1}, 10^{-3}]$, while gammaExp value can begin from 6. Results reported on Table 1 represent the ACP analysis for maxZero = 10^{-1}.

	maxZero=10^{-1}				
gammaExp	**r=1**	**r=0**	**r=10^{-1}**	**r=10^{-2}**	**r=10^{-3}**
20	53.20	50.98	89.21	89.21	89.21
18	53.20	50.24	89.21	89.21	89.21
16	53.19	50.80	89.21	89.21	89.21
14	53.20	50.09	89.21	89.21	89.21
12	53.20	51.51	89.21	89.21	89.21
10	53.20	49.43	89.21	89.21	89.21
8	53.19	48.05	89.21	89.21	89.21
6	53.20	51.20	89.21	89.21	89.21
4	53.20	60.47	63.78	62.30	62.44
2	52.95	49.94	89.21	89.21	89.21
1.4	53.20	54.79	57.07	57.18	57.41

Table 1: The ACP analysis for maxZero=10^{-1}

As shown in table 1 ACP value keeps constant at **89.2%** starting from gammaExp =6 and in the interval [10^{-1},10^{-3}] of maxResidual. Then to choose the best value for both parameters, we based on another criterion which is the temporal complexity. We opted for the combination which converges faster than the others. Table 2 summarizes the combinations for the different tests and their execution time.

	maxZero=10^{-1}		
gammaExp	**r=10^{-1}**	**r=10^{-2}**	**r=10^{-3}**
6	34 min 7 sec	32 min 6 sec	31 min 40 sec
20	33 min 0 sec	31 min 23 sec	30 min 53 sec

Table 2: Execution time for different combinations of gammaExp and maxResidual

We have pushed the value of gammaExp to 200 and 2000 and we noticed that the more its value grows, the more the execution time decreases. Then the best configuration of the MCL algorithm for us is : **maxZero=**10^{-1}, **maxResidual=**10^{-3} and **gammaExp=20**. As a conclusion, from the experimental analysis carried out we see that results generated by the Automatic Partitioning with the best tuning of the MCL algorithm are close to those derived from Ground Truth with ACP=89,2%, which proves that our approach for definition-sense clustering achieved good results. Finally it should be noted that evaluating the performance of our clustering approach was not trivial, as the construction of manual Gound Truth is not an easy task, there is always a large variability in the number of clusters that humans generate for the same dataset. That is why, this dataset was enhanced by the confrontation between diffierent manual partitionings made by different persons, so as to lower subjectivity and then have a good dataset for evaluation.

4. CONCLUSIONS AND PERSPECTIVES

In this paper we have introduced the concept of folksionary which consists in a dictionary that clusters each hashtag's definitions in meanings. We have also defined a four-steps process to build a folksionary. First we gather all definitions by crawling online services, we then apply a semantic distance measure between definitions for each hashtag. We perform a clustering that groups similar definitions into distinct meanings clusters. Clusters are finally presented under the form of a human-readable folksionary. We have conducted a validation of this process: we have developed a web application to build the Ground Truth where participants cluster the definitions. A pairwise evaluation of the results of our clustering process in comparison with the Ground Truth has been conducted. The Evaluation results show that our approach works not only in theory but also in practice: it performs well and produces good results for definition-sense clustering, by approaching Ground Truth with 89.2%. The very close next step concerns the development of techniques to discover other semantic relationships between tags: synonymy, hyperonymy, or part-of. In the long term our goal is to learn an ontology from the folksionary.

5. REFERENCES

[1] T. Abeel, Y. Van de Peer, and Y. Saeys. Java-ml: A machine learning library. *The Journal of Machine Learning Research*, 10:931–934, 2009.

[2] S. Banerjee and T. Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *CICLing '02*, pages 136–145, 2002.

[3] S. Brohée and J. van Helden. Evaluation of clustering algorithms for protein-protein interaction networks. *BBMC Bioinformatics*, 7:488, 2006.

[4] G. R. Burset M. Evaluation of gene structure prediction programs, 1996.

[5] B. Dorow and D. Widdows. Discovering corpus-specific word senses. In *AAAI*, pages 79–82, 2003.

[6] A. J. Enright, S. Van Dongen, and C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Researbch*, 30(7):1575–1584, Apr. 2002.

[7] E. Hatcher, O. Gospodnetic, and M. McCandless. Lucene in action, 2004.

[8] R. Jäschke, A. Hotho, C. Schmitz, B. Ganter, and G. Stumme. Discovering shared conceptualizations in folksonomies. *Web Semantics*, 6(1):38–53, 2008.

[9] M. Lesk. Information in data: using the oxford english dictionary on a computer. *SIGIR Forum*, 20(1-4):18–21, May 1986.

[10] S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *CICLing'03*, pages 241–257, Berlin, Heidelberg, 2003. Springer-Verlag.

[11] A. Plangprasopchok and K. Lerman. Constructing folksonomies from user-specified relations on flickr. In *WWW '09*, pages 781–790, New York, NY, USA, 2009. ACM.

[12] A. Purandare and T. Pedersen. Discriminating among word meanings by identifying similar contexts. In *AAAI*, pages 964–965, 2004.

[13] M. Strohmaier, C. KÃűrner, and R. Kern. Understanding why users tag: A survey of tagging motivation literature and results from an empirical study. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17(0), 2012.

[14] T. D. Troy Simpson. Wordnet-based semantic similarity measurement. October 2005.

[15] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *Proceedings of the Collaborative Web Tagging Workshop at the WWW 2006*, May 2006.