Searching and Browsing Tag Spaces Using the Semantic Tag Clustering Search Framework

Jan-Willem van Dam, Damir Vandic, Frederik Hogenboom, and Flavius Frasincar
Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, the Netherlands
Email: {jwvdam, damir3004}@gmail.com, {fhogenboom, frasincar}@ese.eur.nl

***Abstract*—Many of the existing cloud tagging systems are unable to cope with the syntactic and semantic tag variations during user search and browse activities. As a solution to this problem, we propose the Semantic Tag Clustering Search, a framework which is able to cope with these needs. The framework consists of two parts: removing syntactic variations and creating semantic clusters. For removing syntactic variations, we use the normalized Levenshtein distance and the cosine similarity measure based on tag co-occurrences. For creating semantic clusters, we improve an existing non-hierarchical clustering technique. Using our framework, we are able to find more clusters and achieve a higher precision than the original method.**

***Keywords*-tag spaces; semantic clustering; syntactic variations; Flickr**

I. Introduction

Nowadays, many Web services enable users to label content on the Web by means of tags. Flickr [1] and Delicious [2] (also known as del.icio.us) are two well-known applications which make use of tags. In this paper we focus on the Flickr Service, but our results can be easily applied to other social tagging systems. Users that are registered on the Flickr Web site can upload photographs and assign tags to them. As with most tagging systems the user has no restrictions on the tags that can be used, the user can use any tag he or she wants.

Even though tags are a flexible way of categorizing data, they have their limitations. Tags are prone to typographical errors or syntactic variations due to the amount of freedom users have. This results in having different tags with the same meaning. An example of a typographical mistake is the usage of the keyword 'waterfal' in tags, instead of 'waterfall'. A query for the correctly spelled tag 'waterfall' on Flickr would return $1,158,957$ results, whereas 'waterfal' would only return a small fraction of these results, i.e., $1,388$ items. This implies that potentially $1,157,569$ results are lost due to a typographical mistake. These syntactic variations of tags are very common and are therefore very important aspects to consider when designing a search engine. Google, for example, implements an algorithm for automatic syntactic variation detection in their search service.

A similar problem is related to the use of synonyms (i.e., semantically related terms) in tags. For instance, for a picture which shows the interior of a house, in many cases, users would use the tag 'interior', whereas others might use a tag like 'inside' or 'furniture'. The tags 'interior', 'inside', and 'furniture' are therefore semantically related. Hence, when a user performs a search for 'furniture', (s)he is likely to be also interested in pictures that are tagged with 'interior' and/or 'inside'.

Furthermore, homonyms can also occur, e.g., the tag 'Apple' refers to pictures related to the brand as well as pictures related to an apple growing on a tree. The Flickr search engine cannot distinguish the multiple meanings the tag 'Apple' can have. As there is no structure, hierarchy, or classification available in most tagging systems, a lot of applications could benefit from the availability of such information. Marketing companies for instance often need pictures in their daily activities and these companies would certainly benefit from more structured tagging systems, where tags can be grouped in clusters and identifying the different meanings they have. Searching, browsing, and retrieving pictures would benefit from a structured approach for tag representation [3]. Therefore, we aim to improve the search and exploration of tag spaces by coping with syntactic variations, typographical mistakes, synonyms, homonyms, and related tags.

As a solution to the previously introduced problems related to tagging, we define the Semantic Tag Clustering Search (STCS) framework, which consists of two parts. The first part of the framework deals with syntactic variations, whereas the second part is concerned with deriving semantic clusters. We consider non-hierarchical clusters, where we select the method proposed by [4], as differently than other methods, this algorithm allows tags to appear in multiple clusters, enabling easy detection of different contexts for tags. We adjust this method in order to improve the clustering results.

This paper continues with discussing related work in Sect. II. In Sect. III, we elaborate on our framework and initial experimental results. Our conclusions are given in Sect. IV.

II. Related Work

This section gives a brief overview of related work on the two main tasks of the STCS framework. Firstly, Subsect. II-A discusses literature on syntactic variation, and secondly, Subsect. II-B elaborates on semantic clustering from tags.

A. Syntactic Variations

Syntactic variations between tags form a widely studied research subject, as they represent a well-known symptom in tagging systems. In [5], the authors analyze the performance of the Levenshtein distance [6] and the Hamming distance [7].

An example of an implementation of the Levenshtein distance measure is recent work by Specia and Motta [4]. The authors employ the Levenshtein similarity metric to group morphologically similar tags. They use a high threshold to determine both similar words and misspellings. Within each group of similar tags, one tag is selected to be the representative of the group, and the occurrences of tags in that group are replaced by their representative. As existing algorithms only aim for longer tags, in this paper we provide a solution for this problem by also considering the short tags.

B. Semantic Clustering

In previous approaches, the semantic symptoms are dealt with by either using a clustering technique which results in non-hierarchical clusters of tags, or a hierarchical graph of either tags or clusters of tags. There is an extensive body of literature available on tag clustering. Several measures that cluster related tags are based on co-occurrence data, e.g., Specia and Motta [4] use the cosine similarity. The authors present a complete framework where they address the syntactic variations in a tagging system, create clusters of semantically related tags, and within each cluster, identify the relationship between each tag pair. Their semantic clustering algorithm distinguishes itself from other approaches, by allowing tags to occur in multiple clusters.

Specia and Motta also experiment with different metrics to calculate the similarity between pairs of vectors of co-occurrence data, including the Euclidian and Manhattan distance. However, the cosine similarity measure is reported to yield the best results. Absolute distance metrics such as the Euclidian and Manhattan distance are inappropriate, as they are more sensitive to significant variations in a few elements than little variations in a large number of elements. In the case of Flickr, we deal with a data set with little variations in a large number of elements, and thus for our research we opt for the cosine similarity.

Similar to Specia and Motta, Begelman et al. [3] also create semantic clusters of tags by using co-occurrence data. The authors conclude that clustering techniques can and should be used in combination with tagging. They also argue that these techniques can improve the search and exploration in tag spaces in general.

In this paper we focus on non-hierarchical clustering, as the antithesis – hierarchical clustering – is more complex and thus more time consuming, because it first needs to build the tag hierarchy from which subsequently the clusters are deduced [8]. The amount of data that we are dealing with asks for fast clustering procedures. Current non-hierarchical clustering approaches, e.g., the algorithm proposed by Specia and Motta [4], suffer from merging issues, i.e., larger clusters merge too quickly and smaller clusters merge too slowly. In this paper, we provide a solution to this problem.

III. Framework

This section introduces the Semantic Tag Clustering Search (STCS), a framework for building and utilizing clusters for the browse and search activities in social tagging systems. The input data set for the Semantic Tag Clustering Search (STCS) framework is defined as a tuple $D = \{U, T, P, r\}$, where U, T, and P are the finite sets of users, tag IDs, and pictures, respectively, and r is the ternary relationship $r \subseteq U \times T \times P$, defining the initial annotations of the users. This section continues with an elaboration on our technique for removing syntactic variations in Subsect. III-A. Subsequently, Subsect. III-B gives the procedures for finding semantically related tags, and finally, Subsect. III-C presents initial experimental results.

A. Syntactic Variations

Syntactic variations detection (and removal) is done by creating a set $T' \subset \mathcal{P}(T)$, where $\mathcal{P}(T)$ represents the power set of T. Each element of T' represents a cluster of tags where each tag occurs only in one element (cluster), i.e., if $X, Y \in T'$, $X \neq Y$, and $a \in X$ and $b \in Y$, then this implies $a \neq b$. Then we denote by m' the bijective function that indicates a label for each $x \in T'$, $m' : T' \to L$. Furthermore, for each $l \in L$ there is a $x \in T'$ such that $l \in x$, i.e., l is one of the tags in cluster x. In this context, we employ the normalized Levenshtein similarity $\widetilde{lev}_{ij}$ between tags i and j, which is defined as

$$\widetilde{lev}_{ij} = \frac{lev_{ij}}{\max\left(\text{length}\left(t_i\right), \text{length}\left(t_j\right)\right)} \ . \quad (1)$$

The normalized Levenshtein distances addresses the string lengths, in contrast to absolute Levenshtein distances. For instance, in case of a string length of 24, an absolute Levenshtein distance of 3 is relatively little. However, taking into consideration two strings of length 6, this distance is quite large. By normalizing, we account for this problem.

The algorithm for the syntactic variation clustering uses an undirected graph $G = (T, E)$ as input. The set T contains elements which represent a tag id, and E is the set of weighted edges (triples (t_i, t_j, w_{ij})) representing the similarities between tags. To calculate the weight w_{ij} we propose to use a weighted average based on the normalized Levenshtein distance $\widetilde{lev}_{ij}$ and the cosine similarity between tags i and j based on co-occurrence vectors, $\cos\left(\text{vector}\left(i\right), \text{vector}\left(j\right)\right)$, i.e.,

$$\begin{aligned} w_{ij} \quad = \quad & z_{ij} \times (1 - \widetilde{lev}_{ij}) + \\ & (1 - z_{ij}) \times \cos\left(\text{vector}\left(i\right), \text{vector}\left(j\right)\right) \;, \end{aligned} \tag{2}$$

where

$$z_{ij} = \frac{\max\left(\text{length}\left(t_i\right), \text{length}\left(t_j\right)\right)}{\text{length}\left(t_k\right)} \in (0, 1] \;, \tag{3}$$

with $t_k \in T, \text{length}\left(t_k\right) \geq \text{length}\left(t\right) \forall t \in T$ and $t_i, t_j \in T$. Normalized Levenshtein values are not representative for short tags, which is why the cosine value gets more weight as the maximum tag length gets shorter. The algorithm then proceeds by cutting edges that have a weight lower than a threshold β. The syntactic clusters are computed by determining the connected components in the resulting graph.

B. Semantic Clustering

Semantically related tags are to be clustered based on their meaning. Thus, we create a set T'', with $T'' \subset \mathcal{P}(L)$, which represents clusters of elements from $l \in L$. By allowing tags to belong to multiple clusters, we can identify the different contexts of these tags. To measure the semantic relatedness between tags, we use the cosine similarity based on co-occurrence vectors. Cosine similarity between co-occurrence vectors a and b (where $a, b \in \mathbb{R}^m$, with m representing the number of tags) is denoted as $\cos\left(a, b\right)$, and is defined in the range of $[-1, 1]$. In case that vectors $a, b \in {\mathbb{N}_0}^m$, the range equals $[0, 1]$. Note that $\cos\left(a, b\right) = 0$ can be interpreted as not semantically related, and $\cos\left(a, b\right) = 1$ can be interpreted as fully semantically related.

The algorithm used for finding semantically related tags has originally been proposed by [4]. The algorithm is different from a classical clustering algorithm, as instead of using the centroid, all tags are used to calculate the distance between two clusters. This has the advantage that all the elements within a cluster must be similar amongst each other, instead of being similar just to the centroid. We improve the algorithm by replacing a heuristic for merging similar clusters by a disjunction of two new heuristics. We now continue with elaborating on the algorithm.

At the beginning, each tag is initialized as a cluster. Tags are added to an arbitrary cluster if they are sufficiently similar to that cluster, i.e., when the average cosine of a tag with respect to all elements in the cluster is larger than a threshold χ. Because many tags are similar to each other, the set of initial clusters can contain many duplicate or nearly duplicate clusters. Thus, there is a need for cluster merging. In [4], two heuristics taken in disjunction are proposed for this purpose. The first heuristic merges two clusters if one cluster K contains the other cluster L and is denoted as $K \subseteq L$.

The second heuristic checks whether clusters differ within a small margin, and if so, it adds the distinct words from the smaller cluster to the larger cluster, while removing the smaller cluster. A limitation of the latter heuristic is that it uses a constant threshold for merging clusters, which makes it hard to choose a correct value, i.e., a value where the larger clusters do not merge too quickly and the smaller clusters do merge too slowly. To address this issue, we propose a dynamic threshold, resulting in two new heuristics to be used in a disjunction. The first one considers the semantic relatedness of the difference between two clusters, whereas the second one considers the size of the difference between two clusters in combination with a dynamic threshold.

The first adapted heuristic uses the semantic relatedness of the difference between two clusters. We merge two clusters K and L, where $|K| \geq |L|$, when the average cosine $\overline{\cos}\left(K, L\right)$ is above a certain threshold δ, and thus $\overline{\cos}\left(K, L\right) > \delta$, where the average cosine is defined as

$$\overline{\cos}\left(K, L\right) = \sum_{l \in L-K} \frac{\sum\limits_{k \in K} \frac{\cos\left(\text{vector}\left(k\right), \text{vector}\left(l\right)\right)}{|K|}}{|L - K|} \;. \tag{4}$$

The second adapted heuristic uses the size of the difference between two clusters in combination with a dynamic threshold. We merge the clusters when the normalized difference $\eta\left(K, L\right)$ between the clusters K and L is smaller than a dynamic threshold ε, and thus $\eta\left(K, L\right) < \varepsilon$, where the normalized difference is defined as

$$\eta\left(K, L\right) = \frac{|L - K|}{|L|} \;, \tag{5}$$

and the threshold ε is defined using a parameter ϕ as

$$\varepsilon = \frac{\phi}{\sqrt{|L|}} \;. \tag{6}$$

Thus, we can calculate the maximum number of different elements for the small set to be merged with the big set using

$$f\left(|L|\right) = \lfloor \varepsilon \cdot |L| \rfloor = \left\lfloor \phi \cdot \sqrt{|L|} \right\rfloor \;. \tag{7}$$

Now, we are able to tune the distribution of the maximum allowed difference between clusters by means of the parameter ϕ. Thus, we have created a function that suits the clustering process better, as it is less sensitive to the size of the smaller cluster. In [4], the maximum number of different elements was proportional to the size of the smaller set.

C. Evaluation

We now present the experimental results of the STCS framework by analyzing sets of tag combinations and clusters. First, we discuss the results of removing syntactic variations, and subsequently those of semantic clustering.

Table I
PERFORMANCE OF NON-HIERARCHICAL SEMANTIC CLUSTERING METHODS

Technique	Error rate	Num. clusters	Avg. size clusters	Min. size clusters	Max. size clusters
STCS framework	9.6%	739	4.4	2	67
Specia and Motta	13.1%	421	4.6	2	63

In order to analyze the performance of the system in terms of syntactic variations detection, we define a test set S that contains 200 randomly chosen tag combinations ($S \subset T \times T$) that have been classified as syntactic variations of each other by the STCS framework. The distributions of the tag length for the test set and the original data set of $27,401$ tags are approximately the same. In our experiments, we apply a threshold value β of 0.62 for cutting edges, which is determined by result evaluation using a hill climbing procedure. After manually checking these tags on correctness, we identify 10 mistakes that are produced by the framework, resulting in a syntactic error rate of 5%.

In order to analyze the semantic clustering process, we create a test set which contains 100 randomly chosen clusters, of which the size distribution is similar to that of all clusters. Using a hill climbing procedure, we determine the thresholds χ, δ, and ε. The χ threshold determines whether or not a tag is added to a cluster during the initial cluster creation, and is set to 0.8. The second threshold, i.e., δ, defines the minimum average cosine similarity when merging two sets of which the smaller set has elements that the larger set does not contain, and is set to 0.7. Finally, setting parameter ϕ in the dynamic threshold function ε to 0.8 yields optimal results in our conducted experiments.

For the 100 random clusters (which have 458 tags in total), the error rate is 9.6% (44 misplaced tags). Most of the misplaced tags are part of clusters containing over 20 tags. In general, the algorithm finds many relevant clusters, such as {'rainy', 'Rain', 'wet', 'raining'} and {'iPod', 'iphone', 'mac'}. Furthermore, a lot of clusters are found that contain tags that are translations of concepts in different languages, e.g., {'springtime', 'primavera'}.

When compared with the algorithm proposed by Specia and Motta [4], our method shows better results. For the constant threshold ε in the original algorithm, we determine an optimal value of 0.2 (through a hill climbing procedure), which results in an error rate of 13.1%. Another observation is that our algorithm discovers more clusters (739 against 421) and thus more relationships between tags. The distribution of the cluster sizes is approximately the same for both methods. A summary of the results of the experiments elaborated on in this section is given in Table I.

IV. CONCLUSIONS

In this paper, we have proposed the Semantic Tag Clustering Search (STCS) framework for building and utilizing semantic clusters based on information retrieved from the Flickr social tagging system. The framework consists of two core tasks: removing syntactic variations and creating semantic clusters. For syntactic variation removal, we proposed a measure that uses the normalized Levenshtein value in combination with the cosine value based on co-occurrence vectors. Evaluation results indicate that the framework obtains a syntactic error rate of 5%. For semantic clustering, we compared an existing non-hierarchical clustering method to an adapted version that implements improved cluster merging heuristics. The STCS non-hierarchical clustering algorithm has a lower error rate than the original algorithm and produces finer-grained results.

Future work is aimed at improving the syntactic variation process, e.g., by taking into account abbreviations. Furthermore, it might be worthwhile to investigate fuzzy approaches to the clustering process.

REFERENCES

[1] C. Fake and S. Butterfield, "Flickr - Online Photo Sharing Service," 2010, from: http://www.flickr.com/.

[2] J. Schachter, "Delicious - Social Bookmarking," 2010, from: http://www.delicious.com/.

[3] G. Begelman, P. Keller, and F. Smadja, "Automated Tag Clustering: Improving Search and Exploration in the Tag Space," in *15th World Wide Web Conference (WWW 2006)*, L. A. Carr, D. C. D. Roure, A. Iyengar, C. A. Goble, and M. Dahlin, Eds. ACM Press, 2006, pp. 22–26.

[4] L. Specia and E. Motta, "Integrating Folksonomies with the Semantic Web," in *4th European Semantic Web Conference (ESWC 2007)*, ser. Lecture Notes in Computer Science, E. Franconi, M. Kifer, and W. May, Eds., vol. 4519. Springer, 2007, pp. 503–517.

[5] F. Echarte, J. J. Astrain, A. Córdoba, and J. Villadangos, "Pattern Matching Techniques to Identify Syntactic Variations of Tags in Folksonomies," in *1st World Summit on The Knowledge Society (WSKS 2008)*, ser. Lecture Notes in Computer Science, M. D. Lytras, J. M. Carroll, E. Damiani, and R. D. Tennyson, Eds., vol. 5288. Springer, 2008, pp. 557–564.

[6] V. I. Levenshtein, "Binary Codes Capable of Correction Deletions, Insertions, and Reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[7] R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell System Technical Journal*, vol. 26, no. 2, pp. 147–160, 1950.

[8] P. Schmitz, "Inducing Ontology from Flickr Tags," in *15th World Wide Web Conference (WWW 2006)*, L. A. Carr, D. C. D. Roure, A. Iyengar, C. A. Goble, and M. Dahlin, Eds. ACM Press, 2006, pp. 206–209.