# Cluster-Based Information Retrieval in Tag Spaces

Damir Vandic
vandic@ese.eur.nl

Jan-Willem van Dam
jwvdam@gmail.com

Flavius Frasincar
frasincar@ese.eur.nl

Frederik Hogenboom
fhogenboom@ese.eur.nl

Econometric Institute
Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, the Netherlands

## ABSTRACT

Many of the existing tagging systems fail to cope with syntactic and semantic tag variations during user search and browse activities. As a solution to this problem, we propose the Semantic Tag Clustering Search. The framework consists of three parts: removing syntactic variations, creating semantic clusters, and utilizing the obtained clusters to improve search and exploration of tag spaces. Using our framework, we are able to find relevant clusters and achieve a higher search precision by utilizing these clusters. The advantages of a cluster-based approach for searching and browsing through tag spaces have been exploited in XploreFlickr.com, the implementation of our framework.

## 1. INTRODUCTION

Today's Web offers many services that enable users to label content on the Web by means of tags. Flickr and Delicious (also known as del.icio.us) are two well-known applications utilizing tags. Registered Flickr users are allowed to upload and tag photographs. As with most tagging systems the user has no restrictions on the tags that can be used, i.e., the user can use any tag to his or her likings. 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 different tags with similar meanings, e.g., 'waterfal' and 'waterfall'. A query for 'waterfall' on Flickr returns $1,158,957$ results, whereas 'waterfal' returns $1,388$ results. This implies that potentially $1,157,569$ results are lost due to a typographical mistake. Users also describe pictures in different ways. For a picture which shows the interior of a house, most users would use the tag 'interior', where others would use a tag like 'inside' or 'furniture'. This is a problem for search engines which only implement keyword-based searching, as 'interior', 'inside', and 'furniture' are all semantically related.

As a solution to the previous problem, we define the Semantic Tag Clustering Search (STCS) framework, which consists of three parts. The first part deals with syntactic variations, whereas the second part is concerned with deriving semantic clusters. The last part of the framework consists of a part where search methods utilize these clusters to improve search for pictures. In the STCS framework, we consider non-hierarchical clusters, where we select the method proposed by [3]. Different from other methods, this algorithm allows tags to appear in multiple clusters, which enables easy detection of different contexts for tags. Each cluster is considered to be a context for a tag. Also, we propose an adaptation of this method that improves the clustering results. Finally, we devise a search method, of which the results are compared with a case without knowledge about the semantic clusters or syntactic variation clusters. We have made available an implementation of the STCS framework in the form of a Web application called XploreFlickr.com [4].

## 2. RELATED WORK

Syntactic variations between tags form a widely studied research subject, as they represent a well-known symptom in tagging systems. In [1], the authors analyze the performance of the Levenshtein distance [2] and the Hamming distance. The authors state that Levenshtein and Hamming distances provide similar results for some syntactic variation types, e.g., for typographic errors. In contrast, for variation identification based on the insertion or deletion of characters, the Levenshtein distance performs significantly better than the Hamming distance. This does not imply that the Levenshtein distance performs well enough, as it has problems with for instance identifying variations based on the transposition of adjacent characters, although results can be improved by ignoring candidate tags with less than four characters.

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 which create clusters of related tags are based on co-occurrence data, a commonly used similarity being the cosine similarity.

In this paper we focus on non-hierarchical clustering, as 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. The amount of data that we are dealing with asks for fast clustering procedures. Further, we observe that current non-

*DIR 2011* February 4, 2011, Amsterdam, The Netherlands.
Copyright 2011 by the author(s).

hierarchical clustering approaches, e.g., the algorithm proposed by Specia and Motta [3], suffer from merging issues, i.e., larger clusters merge too easily and smaller clusters merge too difficultly. In this paper, we provide a solution to this problem.

# 3. STCS FRAMEWORK

Due to space limitations, we only discuss the first and second part of the STCS framework in this version of the paper. An extended version of this paper also discusses the third part, i.e., how we use the clusters to improve the performance of tag search engines. This extended version of the paper is to be presented at the 26th ACM Symposium on Applied Computing (SAC 2011) [5].

## 3.1 Syntactic clustering

In the first part of the framework, the syntactic clustering algorithm uses an undirected graph $G = (T, E)$ as input. The set $T$ contains tags, and $E$ is the set of weighted edges (triples $(t_i, t_j, w_{ij})$) representing the similarities between tags. Weight $w_{ij}$ is calculated as a weighted average based on the normalized Levenshtein distance and the cosine similarity between tags $i$ and $j$ using the co-occurrence vectors. Normalized Levenshtein values are not representative for short tags, which is why we increase the weight for the cosine value as the length of the two tags decreases. The algorithm then proceeds by cutting edges that have a weight lower than a threshold $\beta$. The syntactic clusters are computed by determining the connected components in the resulting graph.

## 3.2 Semantic clustering

For semantic clustering, we propose a modified version of the algorithm that is proposed in [3]. The algorithms loops over all tags that are present in the data set and creates a new cluster which only includes the current tag. The algorithm then loops over all tags again and adds a tag to the cluster if it is sufficiently similar to the cluster. The tag is sufficiently similar when the average cosine of the tag with respect to all tags currently present in the cluster is larger than a threshold $\chi$. Because many tags are similar to each other, this procedure produces many duplicate or near duplicate clusters. Hence, there is a need for cluster merging.

The authors of [3] propose two heuristics for the semantic clusters merging process. The first heuristic merges two clusters if the one contains the other and the second heuristic merges clusters if the number of different elements between two clusters is below a certain threshold. We propose a merging heuristic with a dynamic threshold, depending on the cluster sizes. With a constant threshold the larger clusters often merge too easily and the smaller clusters merge too difficultly. The STCS heuristic fits the clustering process better, as it is less sensitive to the size of smaller clusters than the method proposed in [3].

# 4. STCS EVALUATION

In order to analyze the performance of the syntactic variations detection algorithm, we use a test set which contains 200 randomly chosen tag combinations. These tags are subject to the weighted average of the normalized Levenshtein value and the cosine similarity. In our experiments, the weighted average for all tag combinations is calculated with a threshold value $\beta$ 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%.

For the analysis of the semantic clustering process, we follow a similar procedure. For 100 random clusters, which contained 458 tags, the number of misplaced tags is counted, i.e., the tags that should have been placed in another cluster. We encounter 44 misplaced tags and thus the error rate is 9.6%. We report an error of 13.1% for the method of [3], which shows that the STCS method outperforms the original method on this data set. We observe that the STCS algorithm finds many relevant clusters, such as {rainy, Rain, wet, raining} and {iPod, iphone, mac}.

# 5. CONCLUSIONS

The Semantic Tag Clustering Search (STCS) framework is used for building and utilizing semantic clusters based on information retrieved from a social tagging system. The framework has three core tasks: removing syntactic variations, creating semantic clusters, and utilizing obtained clusters to improve search and exploration of tag spaces. For the syntactic clustering process we have proposed a measure based on the normalized Levenshtein value combined with the cosine value based on co-occurrence vectors. Results show that the framework obtains an error rate for syntactic clustering of 5% and 9.6% for semantic clustering. We compared the non-hierarchical clustering method proposed by Specia and Motta [3] to our adapted version and have found that the adapted version has a lower error rate than the original method.

As future work, we would like to improve the process of removing syntactic variations by using two ideas. First, we want to take into account abbreviations, as the Levenshtein distance does not address this issue. Second, we would like to experiment with variable cost Levenshtein distances, which associate different weights to edit operations depending on update characters and their location.

# 6. REFERENCES

[1] 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)*, volume 5288 of *LNCS*, pages 557–564. Springer, 2008.

[2] V. I. Levenshtein. Binary Codes Capable of Correction Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[3] L. Specia and E. Motta. Integrating Folksonomies with the Semantic Web. In *4th European Semantic Web Conference (ESWC 2007)*, volume 4519 of *LNCS*, pages 503–517. Springer, 2007.

[4] J. W. van Dam, D. Vandic, F. Hogenboom, and F. Frasincar. XploreFlickr.com, 2010. From: `http://www.xploreflickr.com/`.

[5] D. Vandic, J. W. van Dam, F. Hogenboom, and F. Frasincar. A Semantic Clustering-Based Approach for Searching and Browsing Tag Spaces. In *26th Symposium on Applied Computing (SAC 2011)*, pages 1698–1704. ACM, 2011.