MaSiMe: A Customized Similarity Measure and Its Application for Tag Cloud Refactoring

David Urdiales-Nieto, Jorge Martinez-Gil, and José F. Aldana-Montes

University of Málaga, Department of Computer Languages and Computing Sciences
Boulevard Louis Pasteur 35, 29071 Málaga, Spain
{durdiales,jorgemar,jfam}@lcc.uma.es
http://khaos.uma.es/

Abstract. Nowadays the popularity of tag clouds in websites is increased notably, but its generation is criticized because its lack of control causes it to be more likely to produce inconsistent and redundant results. It is well known that if tags are freely chosen (instead of taken from a given set of terms), synonyms (multiple tags for the same meaning), normalization of words and even, heterogeneity of users are likely to arise, lowering the efficiency of content indexing and searching contents. To solve this problem, we have designed the Maximum Similarity Measure (MaSiMe) a dynamic and flexible similarity measure that is able to take into account and optimize several considerations of the user who wishes to obtain a free-of-redundancies tag cloud. Moreover, we include an algorithm to effectively compute the measure and a parametric study to determine the best configuration for this algorithm.

Keywords: social tagging systems, social network analysis, Web 2.0.

1 Introduction

Web 2.0 is a paradigm about the proliferation of interactivity and informal annotation of contents. This informal annotation is performed by using tags. Tags are personally chosen keywords assigned to resources. So instead of putting a bookmark into a folder, users might assign it tags. The main aspect is that tagging creates an annotation to the existing content. If users share these with others, everybody benefits by discovering new sites and getting better matches for their searches.

Tag clouds represent a whole collection of tags as weighted lists. The more often a tag has been used, the larger it will be displayed in the list. This can be used to both characterize users, websites, as well as groups of users.

To date, tag clouds have been applied to just a few kinds of focuses (links, photos, albums, blog posts are the more recognizable). In the future, expect to see specialized tag cloud implementations emerge for a tremendous variety of fields and focuses: cars, properties or homes for sale, hotels and travel destinations, products, sports teams, media of all types, political campaigns, financial markets, brands, etc [1].

R. Meersman, P. Herrero, and T. Dillon (Eds.): OTM 2009 Workshops, LNCS 5872, pp. 937–946, 2009.

On the other hand, although automatic matching between tags is perhaps the most appropriate way to solve this kind of problems, it has the disadvantage but when dealing with natural language often it leads a significant error rate, so researchers try to find *customized similarity functions* (CSF) [2] in order to obtain the best solution for each situation. We are following this line. Therefore, the main contributions of this work are:

- The introduction of a new CSF called Maximum Similarity Measure (MaSiMe) to solve the lack of terminological control in tag clouds.
- An algorithm for computing the measure automatically and efficiently and a statistical study to choose the most appropriate parameters.
- An empirical evaluation of the measure and discussion about the advantages of its application in real situations.

The remainder of this article is organized as follows. Section 2 describes the problem statement related to the lack of terminological control in tag clouds. Section 3 describes the preliminary definitions and properties that are necessary for our proposal. Section 4 discusses our Customized Similarity Measure and a way to effectively compute it. Section 5 shows the empirical data that we have obtained from some experiments, including a comparison with other tools. Section 6 compares our work with other approaches qualitatively. And finally, in Section 7 the conclusions are discussed and future work presented.

2 Problem Statement

Tags clouds offer an easy method to organize information in the Web 2.0. This fact and their collaborative features have derived in an extensive involvement in many Social Web projects. However they present important drawbacks regarding their limited exploring and searching capabilities, in contrast with other methods as taxonomies, thesauruses and ontologies. One of these drawbacks is an effect of its flexibility for tagging, producing frequently multiple semantic variations of a same tag. As tag clouds become larger, more problems appear regarding the use of tag variations at different language levels [3]. All these problems make more and more difficult the exploration and retrieval of information decreasing the quality of tag clouds.

We wish to obtain a free-of-redundancies tag cloud as Fig. 1 shows, where tags with similar means have been grouped. The most significant tag can be visible and the rest of similar tags could be hidden, for example. Only, when a user may click on a significant tag, other less important tags would be showed.

On the other hand, we need a mechanism to detect similarity in tag clouds. In this way, functions for calculating relatedness among terms can be divided into similarity measures and distance measures.

- A similarity measure is a function that associates a numeric value with a pair of objects, with the idea that a higher value indicates greater similarity.

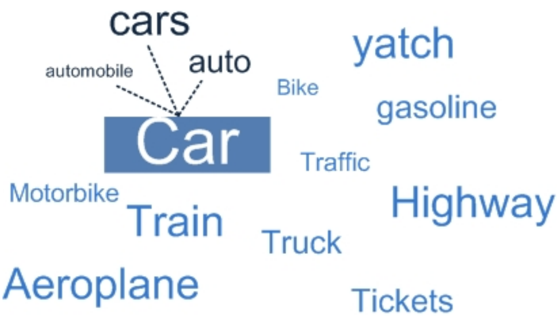

Fig. 1. Refactored tag cloud. Tags with similar means have been grouped.

- A distance measure is a function that associates a non-negative numeric value with a pair of objects, with the idea that a short distance means greater similarity. Distance measures usually satisfy the mathematical axioms of a metric.

Frequently, there are long-standing psychological objections to the axioms used to define a distance metric. For example, a metric will always give the same distance from a to b as from b to a, but in practice we are more likely to say that a child resembles their parent than to say that a parent resembles their child [4]. Similarity measures give us an idea about the probability of compared objects being the same, but without falling into the psychological objections of a metric. So from our point of view, working with similarity measures is more appropriate for detecting relatedness between different tags with a similar meaning.

3 Technical Preliminaries

In this section, we are going to explain the technical details which are necessary to follow our proposal.

Definition 1 (Similarity Measure). *A similarity measure sm is a function* $sm : \mu_1 \times \mu_2 \mapsto R$ *that associates the similarity of two input solution mappings* μ_1 *and* μ_2 *to a similarity score* $sc \in \Re$ *in the range [0, 1].*

A similarity score of 0 stands for complete inequality and 1 for equality of the input solution mappings μ_1 and μ_2.

Definition 2 (Granularity). *Given a weight vector* $\boldsymbol{w} = (i, j, k, ..., t)$ *we define granularity as the Maximum Common Divisor from the components of the vector.*

Its purpose is to reduce the infinite number of candidates in the solutions space to a finite number.

4 MaSiMe: Maximum Similarity Measure

In this section, we are going to explain MaSiMe and its associated properties. Then, we propose an efficient algorithm to compute MaSiMe and finally, we present a statistical study to determine the most appropriate configuration for the algorithm.

4.1 Maximum Similarity Measure

An initial approach for an ideal Customized Similarity Measure which would be defined in the following way:

Let $\boldsymbol{A}$ be a vector of matching algorithms in the form of a similarity measure and $\boldsymbol{w}$ a weight vector then:

$$MaSiMe(c1, c2) = x \in [0,1] \in \Re \rightarrow \exists \left\langle \boldsymbol{A}, \boldsymbol{w} \right\rangle, x = max(\textstyle\sum_{i=1}^{i=n} A_i \cdot w_i)$$
$$\textit{with the following restriction } \textstyle\sum_{i=1}^{i=n} w_i \leq 1$$

But from the point of view of engineering, this measure leads to an optimization problem for calculating the weight vector, because the number of candidates from the solution space is infinite. For this reason, we present MaSiMe, which uses the notion of granularity for setting a finite number of candidates in that solution space. This solution means that the problem of computing the similarity can be solved in a polynomial time.

Definition 3. Maximum Similarity Measure (MaSiMe)
Let $\boldsymbol{A}$ be a vector of matching algorithms in the form of a similarity measure, let $\boldsymbol{w}$ be a weight vector and let g the granularity then:

$$MaSiMe(c1, c2) = x \in [0,1] \in \Re \rightarrow \exists \left\langle \boldsymbol{A}, \boldsymbol{w}, g \right\rangle, x = max(\textstyle\sum_{i=1}^{i=n} A_i \cdot w_i)$$
$$\textit{with the following restrictions } \textstyle\sum_{i=1}^{i=n} w_i \leq 1 \wedge \forall w_i \in \boldsymbol{w}, w_i \in \{\dot{g}\}$$
$$\textit{where } \{\dot{g}\} \textit{ denotes the set of multiples of } g.$$

Example 1. Given an arbitrary set of algorithms and a granularity of 0.05, calculate MaSiMe for the pair $(author, name\_author)$.

$$MaSiMe(author, name\_author) = .542 \in [0,1] \rightarrow$$
$$\exists \left\langle A = (L,B,M,Q), w = (0.8,0,0,0.2), g = 0.05 \right\rangle, 0.542 = max(\textstyle\sum_{i=1}^{i=4} A_i \cdot w_i)$$

Where L = Levhenstein [5], B = BlockDistance [6], M = MatchingCoefficient [6] , Q = QGramsDistance [7]

There are several properties for this definition:

Property 1 (Continuous Uniform Distribution). *A priori, MaSiMe presents a continuous uniform distribution in the interval* $[0,1]$*, that is to say, its probability density function is characterized by*

$$\forall\ a,b \in [0,1] \rightarrow f(x) = \frac{1}{b-a}\ for\ a \leq x \leq b$$

Property 2 (Maximality). *If one of the algorithms belonging to the set of matching algorithms returns a similarity of 1, then the value of MaSiMe is 1.*

$$\exists A_i \in \boldsymbol{A},\ A_i(c1, c2) = 1 \rightarrow MaSiMe(c1, c2) = 1$$

Moreover, the reciprocal is true

$$MaSiMe(c1, c2) = 1 \rightarrow \exists A_i \in \boldsymbol{A},\ A_i(c1, c2) = 1$$

Property 3 (Monotonicity). *Let S be a set of matching algorithms, and let S' be a superset of S. If MaSiMe has a specific value for S, then the value for S' is either equal to or greater than this value.*

$$\forall S' \supset S, MaSiMe_s = x \rightarrow MaSiMe_{s'} \geq x$$

4.2 Computing the Weight Vector

Once the problem is clear and the parameters $\boldsymbol{A}$ and g are known, it is necessary to effectively compute the weight vector. At this point, we leave the field of similarity measures to move into the field of engineering.

It is possible to compute MaSiMe in several ways, for this work, we have designed a greedy mechanism that seems to be effective and efficient. In the next paragraphs, we firstly describe this mechanism and then we discuss its associated complexity. We are going to solve this using a greedy strategy, thus a strategy which consists of making the locally optimum choice at each stage with the hope of finding the global optimum.

Theorem 1 (About Computing MaSiMe). *Let S be the set of all the matching algorithms, let A be the subset of S, thus, the set of matching algorithms that we want to use, let g be the granularity, let Q the set of positive Rational Numbers, let $i, j, k, ..., t$ be indexes belonging to the set of multiples for the granularity (denoted $\{\dot{g}\}$) then, a set of rational vectors r exists where each element r_i is result of the scalar product between A and the index pattern $(i, j - i, k - j, ..., 1 - t)$. All of this subject to $j \geq i \wedge k \geq j \wedge 1 \geq k$. Moreover, the final result, called R, is the maximum of the elements r_i and is always less or equal than 1.*

And in mathematical form:

$$\exists A \subset S, \exists g \in [0, 1] \in Q+, \forall i, j, k, ..., t \in \{\dot{g}\} \rightarrow \exists \boldsymbol{r}, r_i = \boldsymbol{A} \cdot (i, j - i, k - j, ..., 1 - t)$$
$$\textit{with the followings restrictions } j \geq i \wedge k \geq j \wedge 1 \geq k$$
$$R = max\ (r_i) \leq 1$$

Proof 1. *r_i is by definition the scalar product between a vector of matching algorithms that implements similarity measures and the pattern $(i, j - i, k - j, ..., 1 - t)$. In this case, a similarity measure cannot be greater than 1 by Definition 1 and the sum of the pattern indexes cannot be greater than 1 by restriction $(i, j - i, k - j, ..., 1 - t)$, so scalar product of such factors cannot be greater than 1.*

Now, we are going to show how to implement the computation of MaSiMe by using an imperative programming language. Algorithm 1 shows the pseudocode implementation for this theorem.

```
Input: tag cloud: $TC$
Input: algorithm vector: $A$
Input: granularity: $g$
Output: $MaSiMe$
foreach pair $(c1, c2)$ of terms in $TC$ do
    foreach index $i, j, k, ..., t \in \kappa \times g$ do
        $result = A_1(c1, c2) \cdot i +$
        $A_2(c1, c2) \cdot j - i +$
        $A_3(c1, c2) \cdot k - j +$
        $A_4(c1, c2) \cdot t - k +$
        ...
        $A_n(c1, c2) \cdot 1 - t$ ;
        if $result > MaSiMe$ then
            $MaSiMe = result$;
        end
        if $MaSiMe = 1$ then
            stop;
        end
    end
    if $MaSiMe > threshold$ then
        merge $(MostWeigthedTerm(c1, c2), LightTerm(c1, c2))$;
    end
end
```

Algorithm 1. The greedy algorithm to compute MaSiMe

The algorithm can be stopped when it obtains a partial result equal to 1, because this is the maximum value than we can hope for.

Complexity. The strategy seems to be brute force, but it is not (n-1 loops are needed to obtain n parameters). Have into account that the input data size is, but the computational complexity for the algorithm according to big O notation [8] is

$$O(n^{length\ of\ A-1})$$

In this way, the total complexity (TC) for MaSiMe is:

$$TC(MaSiMe_A) = O(max(max(O(A_i)), O(strategy)))$$

and therefore for MaSiMe using the greedy strategy

$$TC(MaSiMe_A) = O(max(max(O(A_i)), O(n^{length\ of\ A-1})))$$

4.3 Statistical Study to Determine the Granularity

We have designed the proposed algorithm, but in order to provide a specific value for its granularity we have performed a parametric study. In this study, we have tried to discover the value that maximizes the value for the granularity by means of an experimental study. In Fig. 2, it can be seen that for several independent experiments the most suitable value is in the range between 0.1 and 0.13.

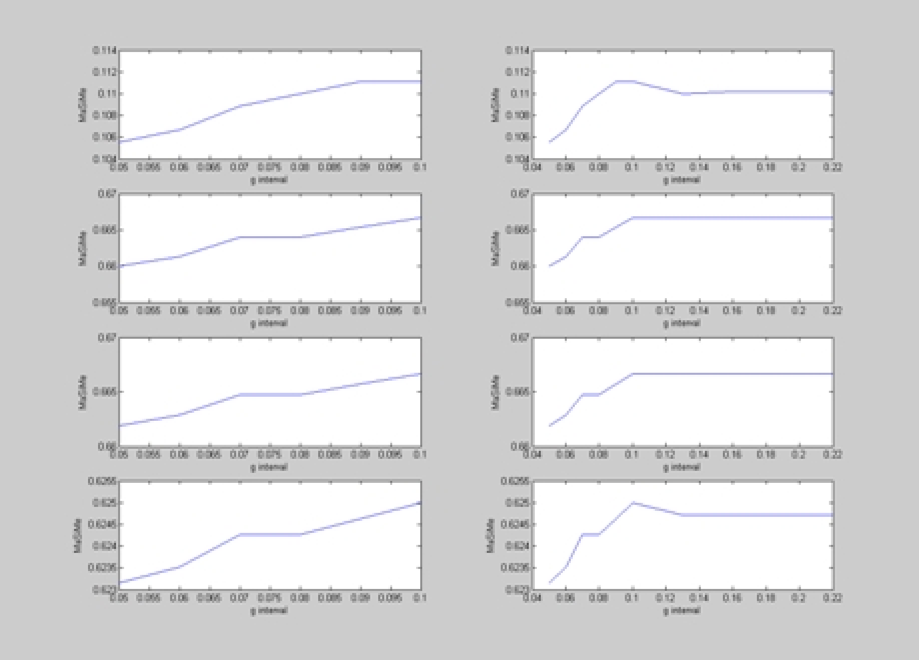

Fig. 2. Statistical study which shows that the most suitable value for granularity is in the range between 0.1 and 0.13. Cases analyzed present an increasing value of MaSiMe for low values of granularity, and MaSiMe presents the highest value between 0.1 and 0.13. MaSiMe is a constant value for higher values of granularity.

Table 1. The statistical study shows the most suitable value for granularity is 0.10 because it provides the best results in all cases

	Granularity value	**No-adding function**	**Adding function**
Experiment 1	0.10	0.11	1.00
	0.13	0.11	1.00
Experiment 2	0.10	0.67	0.67
	0.13	0.67	0.61
Experiment 3	0.10	0.63	0.63
	0.13	0.63	0.57
Experiment 4	0.10	0.67	0.67
	0.13	0.67	0.61

Once we have obtained the granularity range with which is obtained the best MaSiMe value, a new statistical study is made with the same concepts to obtain the best MaSiMe value between 0.1 and 0.13. The function used by Google to take similarity distances [9] is introduced in MaSiMe showing better MaSiMe values using a granularity value of 0.1. Adding this function and using a granularity of 0.13 MaSiMe values are lower than without adding this function. Then, we can conclude that the suitable granularity value is 0.1. Table 1 shows a comparative study with and without this new function.

5 Empirical Evaluation

We have tested an implementation of MaSiMe. We have used MaSiMe in the following way: For the matching algorithms vector, we have chosen a set of well known algorithms $A = \{$*Levhenstein [5], Stoilos [10], Google [9], Q-Gram [7]* $\}$ and for granularity, g = 0.1 (as we have determined in the previous section).

We show an example (Table 2) of mappings that MaSiMe has been able to discover from [11] and [12]. We have compared the results with two of the most outstanding tools: FOAM [13] and RiMOM [14].

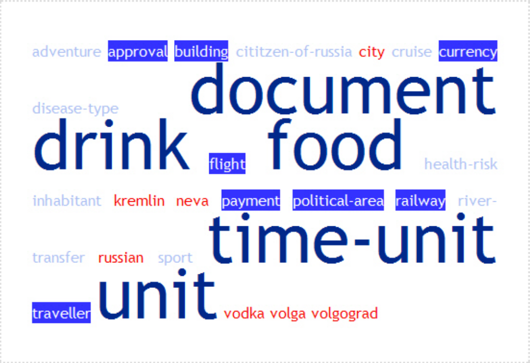

Fig. 3. Refactorized tag cloud. Similar tags have been added to the scope of their corresponding and most significant tag. As consequence, we obtain a free-of-redundancies tag cloud where new terms can be included.

Table 2. Comparison of several mappings from several tools

Russia1	Russia2	FOAM	RiMOM	MaSiMe
food	food	1.00	0.50	1.00
drink	drink	1.00	0.71	1.00
traveler	normal_traveler	0	0	0.90
health_risk	disease_type	0	0.17	0.17
document	document	1.00	0.99	1.00
approval	certificate	0	0.21	0.24
monetary_unit	currency	0	0	0.29
inhabitant	citizen_of_russia	0	0.11	0.12
adventure	sport	0	0.01	0.11
building	public_building	0.80	0.60	0.53
flight	air_travel	0	≈ 0	1.00
river_transfer	cruise	0	0.21	0.21
political_area	political_region	0	0.40	0.69

Moreover, in Fig. 3 we show the appearance from the experiment where we have obtained a free-of-redundancies tag cloud. Moreover, the refactoring process allows us to obtain a nicer tag cloud where new terms can be included. To obtain better results in the test, it is only necessary to expand the vector $\boldsymbol{A}$ with algorithms to have into account aspects to compare among the tags.

6 Related Work

A first approach to solve the problem could consist of systems employing an optional authority control of keywords or names and resource titles, by connecting the system to established authority control databases or controlled vocabularies using some kind of techniques, but we think that it is a very restrictive technique in relation to ours.

Other approach consists of the utilization of approximate string matching techniques to identify syntactic variations of tags [3]. But the weakness of this proposal is that it has been designed to work at syntactical level only. In this way, only misspelled or denormalized tags can be merged with the relevant ones.

On the other hand, there are tag clustering approaches. Most significant work following this paradigm is presented in [15], where a technique for pre-filtering tags before of applying an algorithm for tag clustering is proposed. Authors try to perform a statistical analysis of the tag space in order to identify groups, or clusters, of possibly related tags. Clustering is based on the similarity among tags given by their co-occurrence when describing a resource. But the goal of this work is substantially different from ours, because it tries to find relationships within tags in order to integrate folksonomies with ontologies.

7 Conclusions

We have presented MaSiMe, a new similarity measure and its application to tag cloud refactoring as part of a novel computational approach for flexible and accurate automatic matching that generalizes and extends previous proposals for exploiting an ensemble of matchers.

Using MaSiMe to compare semantic similarities between tags needs to the user for choosing the appropriate algorithms for comparing such aspects it could be corrected (i.e. misspellings or typos, plurals, synonyms, informal words and, so on). As the results show, MaSiMe seems to be an accurate, and flexible similarity measure for detecting semantic relatedness between tags in a tag cloud and its application has been satisfactory. Moreover, we should not forget that MaSiMe is easy to implement in an efficient way.

Acknowledgements

This work has been funded: *ICARIA: From Semantic Web to Systems Biology* (TIN2008-04844) and *Pilot Project for Training and Developing Applied Systems Biology* (P07-TIC-02978).

References

1. Marinchev, I.: Practical Semantic Web Tagging and Tag Clouds. Cybernetics and Information Technologies 6(3), 33–39 (2006)
2. Kiefer, C., Bernstein, A., Stocker, M.: The Fundamentals of iSPARQL: A Virtual Triple Approach for Similarity-Based Semantic Web Tasks. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 295–309. Springer, Heidelberg (2007)
3. Echarte, F., Astrain, J.J., Córdoba, A., Villadangos, J.: Pattern Matching Techniques to Identify Syntactic Variations of Tags in Folksonomies. In: Lytras, M.D., Carroll, J.M., Damiani, E., Tennyson, R.D. (eds.) WSKS 2008. LNCS (LNAI), vol. 5288, pp. 557–564. Springer, Heidelberg (2008)
4. Widdows, D.: Geometry and Meaning. The University of Chicago Press (2004)
5. Levenshtein, V.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics-Doklady 10, 707–710 (1966)
6. Ziegler, P., Kiefer, C., Sturm, C., Dittrich, K.R., Bernstein, A.: Detecting Similarities in Ontologies with the SOQA-SimPack Toolkit. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 59–76. Springer, Heidelberg (2006)
7. Ukkonen, E.: Approximate String Matching with q-grams and Maximal Matches. Theor. Comput. Sci. 92(1), 191–211 (1992)
8. Knuth, D.: The Art of Computer Programming. Fundamental Algorithms, 3rd edn., vol. 1. Addison-Wesley, Reading (1997)
9. Cilibrasi, R., Vitányi, P.M.B.: The Google Similarity Distance. IEEE Trans. Knowl. Data Eng. 19(3), 370–383 (2007)
10. Stoilos, G., Stamou, G.B., Kollias, S.D.: A String Metric for Ontology Alignment. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 624–637. Springer, Heidelberg (2005)
11. http://www.aifb.uni-karlsruhe.de/WBS/meh/foam/ontologies/russia1.owl (last visit: February 3, 2009)
12. http://www.aifb.uni-karlsruhe.de/WBS/meh/foam/ontologies/russia2.owl (last visit: February 3, 2009)
13. Ehrig, M., Sure, Y.: FOAM - Framework for Ontology Alignment and Mapping - Results of the Ontology Alignment Evaluation Initiative. Integrating Ontologies (2005)
14. Li, Y., Li, J., Zhang, D., Tang, J.: Result of Ontology Alignment with RiMOM at OAEI 2006. In: International Workshop on Ontology Matching collocated with the 5th International Semantic Web Conference (2006)
15. Specia, L., Motta, E.: Integrating Folksonomies with the Semantic Web. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 624–639. Springer, Heidelberg (2007)