International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 03 | Mar-2018 www.irjet.net p-ISSN: 2395-0072

PYTHON BASED MACHINE LEARNING FOR PROFILE MATCHING

Arpitha D[1], Shrilakshmi Prasad[2] Prakruthi S[3], Raghuram A S[4]

[1,2,3,4] *Assistant Professor, ATME College of Engineering, Mysuru, arpithad86@gmail.com*

Abstract - *Information retrieval from different social sites has to be done for various Inter-social activities & functionalities for that matching user profiles has to be done. User creates same profiles under different sites.(for personal use, for work etc.,). The current work explores the different algorithms foe similarity measures for classifications, presenting new analysis. Different data sets are used to check the value and also classification. A model is built using Decision tree algorithm in order to check the similarity measures between two profiles. Experiments on real data sets are used to compare methods and their changes to the similarity-based learning.*

Key Words: *Machine Learning; Similarity Measures; Decision tree Algorithm;*

1. INTRODUCTION

Machine learning is subfield of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. By making use of Machine learning concepts, the computer programs that can teach themselves to grow and change when exposed to new data. A similarity measure is a method of fining the similar content between two objects. The values may be equal or unequal or may vary between these two values. In this paper we are considering different similarity algorithms to find similarity between two strings for profile matching[1]. We are using cosine similarity, jaro ratio, jaccard ratio, levenshtein distance, hamming distance, dice. The Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other.

Nowadays, social networking[2] has become an important part of the online activities on the web. Social sites gain popularity thanks to the diverse services provided ranging from collaborative tagging (e.g.,Flicker 1), blogging sites (e.g., Live journal 2), and mainly to social networking (e.g., Facebook 3, LinkedIn 4, MySpace 5) with a nonstop growing number of active users.

As you can see in Figure 1, Bob mainly uses two social sites: the first is Facebook (SN1) to stay connected with his friends, and the second is LinkedIn (SN2) to maintain professional contact with a group of software developers.

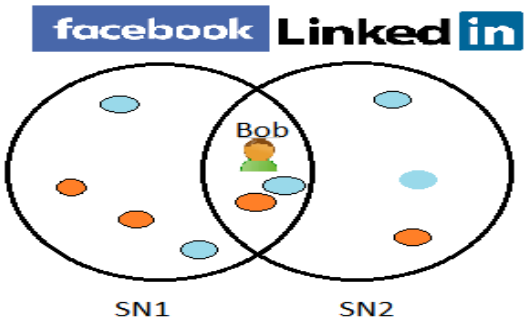

Fig1: Social Network of Bob within Facebook & LinkedIn

For different purposes he needs to identify, intersection between SN1 and SN2, Union between SN1 and SN2 and Difference between SN1 and SN2. Performing this kind of operations requires in one way or another the matching of users' profiles. In fact, the user profile matching consists of accurately linking records corresponding to the same entity in the same or different data sources. So it is necessary to perform Profile Matching to eliminate redundancy.

The paper is structured as follows: In Section 2 the general properties of similarity measures are discussed and the basis for most presented similarity measures is introduced. The related work is explored in section 2. Section 3 captures different similarity measures algorithms used in the proposed work. Finally, Section 6 concludes the paper.

2. RELATED WORK

For profile matching many recent technologies are used to measure the similarity between different profiles. Similarity learning comes under of supervised machine learning method in artificial intelligence. It is related to regression and classification. The objective of similarity is to consider the different data set and apply the methods for those sets; based on the data it provides the similarity value. A similarity measure or similarity function is a real valued function that verifies the similarity between two objects. The function deriving a similarity value of two strings has to fulfill the following general rules:

- The function is proportional, that is, the similarity measure of string A and B equals the similarity measure of string B and A;

- The values provided by the similarity measure are between zero (unrelated) and 1 equivalent;

Detecting user attributes based on user profiles. Many researches have proposed a different method to explore the similarity to check user's profile. And to check their work environment and communication. [1].

Flink, a system developed in [3], it is used to identify the person's multiple information across different source. It uses IFP comparison method. Or it checks the name attribute. If it checks with respect to names then only names from different source is considered and it finds the similarity value. But dissimilarity between last name is not considered. But for matching only name attribute is not sufficient. For better Other parameters should be considered. However, performing the matching by considering only the name is not enough, accuracy needs the use of more attributes for better matching results.

In [4], the authors consider that the single use of an IFP, such as the foaf:mbox sha1sum in FOAF, is not suitable. The work presents why the user creates same profile under different sites. They create different profiles using different email address. They provided some explanation showing that it is very common for a user to have two social network accounts with different email address. In their work, they cited the following reasons:

1) Change of email address.

2) More than one email address depending on the use of work.

3) Email addresses can act as proxies for more than one person.

To implement this they used Foaf-O-Matic to create an FOAF profiles. By making use of this each user is assigned with unique identifier and information is stored in their database. Initially the user has to add their information and then thy need to identify the friends. The registered user has to find the duplicate profiles of their friends. After noticing the report should be sent to FOAF profile

Then they presented an extended service called Foaf-O-Matic for the creation of FOAF profiles. This service is based on issuing a globally unique identifier for users and storing it in an infrastructure. In this work, the primary user has a manual task of adding and identifying each friend as well as determining by himself duplicated friends profiles. In their proposed application, they seek to propose a user-friendly way to include the identifier to the FOAF profile.

3. PROPOSED WORK

A similarity measure is a real-valued function that quantifies the similarity between two objects. Although no single definition of a similarity measure exists, usually such measures are in some sense the inverse of distance metrics: they take on large values for similar objects and either zero or a negative value for very dissimilar objects. Similarity measures are implemented using python to check similarity between two data sets.

sim(x,y) where, x and y can be strings, numbers, tuples, objects, images etc.

- sim(x,y) = 1 if for exact match
- sim(x,y) = 0 if x and y are different.
- 0 < sim(x,y) < 1 for some approximate similarity. Some maching between x and y

A. Different Similarity Measures:

We have developed unique methods for checking similarity between profiles which are present in different sources. There are many similarity measures such as Edit-based, Token-based, Domain – dependent, Phonetic, Hybrid. But we have implemented Edit-based and Token-based similarity measures such as Cosine similarity, Dice similarity, Jaccard similarity that comes under Token-based. And Levenshtein similarity, Jaro similarity, Jaro-winkler and Hamming which comes under Edit-based similarity measures.

The Cosine Similarity

The cosine similarity between two vectors (or two documents on the Vector Space) is a measure that calculates the cosine of the angle between them. The cosine rule checks the orientation but not the magnitude, it can be seen as a comparison between documents on a normalized space because we're not taking into the consideration only the magnitude of each word count (tf-idf) of each document, but the angle between the documents. What we have to do to build the cosine similarity equation is to solve the equation of the dot product for the

cos Θ:

$$\vec{a} \cdot \vec{b} = ||\vec{a}|| \; ||\vec{b}|| \cos\Theta$$

$$\cos\Theta = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \; ||\vec{b}||}$$

And that is it, this is the cosine similarity formula. Cosine Similarity will generate a metric that says how related are two documents by looking at the angle instead of magnitude. The Cosine Similarity values for different documents, 1 (same direction), 0 (90 deg.), -1 (opposite directions)

Levenshtein Distance

Minimum number of character insertions, deletions, and replacements necessary to transform s1 into s2.

1. Initialize matrix *M* of size *(|s1|+1) x (|s2|+1)*

2. Fill matrix: $M_{i,0}$ = i and $M_{j,0}$ = j

3. Recursion:

$$M_{i,j}=\begin{cases} Mi-1,j-1 & if x[i]=y[j] \\ 1+\min(Mi-1,j,Mi,j-1,Mi-1,j-1 & else \end{cases}$$

4. Distance: LevinshtienDist(x,y) = $M_{|x|,|y|}$

Levenshtein Similarity: $sim_{levinshtien}(x, y) = 1 - LevinshteinDist(x, y)/\ max\ (|x|, |y|)$

Jaro – Winkler similarity

The jaro winkler is a measure of similarity between two strings.

$$Sim_{jaro}=\frac{1}{3}\left(\frac{m}{|x|}+\frac{m}{|y|}+\frac{m-t}{m}\right)$$

Where, m: number of matching characters

t: number of transpositions

$Sim_{winkler}(x,y) = sim_{jaro}(x,y) + (1 - sim_{jaro}(x, y))\ p/\ 10$

Where, p is the length of the common prefix of x and y.

Jaccard coefficient:

The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets. It is token based similarity measures.

$$Sim_{jaccard}(x,y) = \frac{|tok(x)\cap tok(y)|}{tok(x)+tok(y)-|tok(x)\cap tok(y)|}$$

$$=\frac{|tok(x)\cap tok(y)|}{|tok(x)\cup tok(y)|}$$

Example: if A = { 5,7,1} and B = {4,1,8,2}

Then the Union A ∪ B = {1,2,4,5,7,8} and A ∩ B = {1}

$$Sim_{jaccard}(A,B) = \frac{|tok(x)\cap tok(y)|}{|tok(x)\cup tok(y)|}$$

= 1/6

= 0.15

Dice's coefficient

Dice's coefficient measures how similar a set and another set are. It can be used to measure how similar two strings are in terms of the number of common bigrams (a bigram is a pair of adjacent letters in the string).

$$Sim_{dice}(x,y) = \frac{|tok(x)\cap tok(y)|}{|tok(x)\cup tok(y)|}$$

where |*X*| and |*Y*| are the numbers of elements in the two samples.

Examples:

Let string 1 = night

string 2 = ncaht

The bigram in each word is calculated as below:

{ni, ig, gh, ht}

{nc, ca, ah, ht}

Set consisting of four elements, and the intersection of these two sets has one element i.e common element is "ht"

Inserting these numbers into the formula, we calculate, $s = (2 \cdot 1) / (4 + 4) = 0.25$.

Hamming

The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. In another way, it measures the minimum number of substitutions required to change one string into the other, or the minimum number of errors that could have transformed one string into the other.

Example: "karolin" and "kathrin" is 3.

B. Decision tree Algorithm:

Tree based algorithms used in supervised learning methods. It is used in classification method and also Regression. Initially it creates the root node. Based on it it uses the different kinds of entropy values. After calculation of entropy it selects the value which got maximum value. Methods like decision trees, random forest, gradient boosting are being popularly used in all kinds of data science problems. Here we are using Decision tree algorithm (tree based learning algorithm) to build a predictive model. The proposed algorithm finds the similarity and its between 0 and 1. The result is fed to the tree. It predicts the data based on the several input values. In case of similarity if it finds the matching then only it should be stated as 1.

Types of decision tree are based on the type of target variable we have. It can be of two types[5]:

1. Categorical Variable Decision Tree
2. Continuous Variable Decision Tree

Scilit-learn is Open source [5], commercially usable - BSD license. Simple and efficient tools for data mining and data analysis Accessible to everybody, and reusable in various contexts and it is built on NumPy, SciPy, and matplotlib. Once the input data to the decision tree is provided, which is obtained from similarity measures, the decision tree calculates the final result whether the two records are same or not. The output will be in the form 0 or 1. Figure 3.1 shows the result of cosine similarity. Cosine similarity is used to match the job title and name similarity. Fig 3.12 shows the result for jaccard and it is used to match the education and city and state match. For each attribute different algorithms is used. Jaro – winkler is used to match the sir-names of the given attributes. Levenshtein Distance is used to find the similarity between two names. There are different attributes that are considered they are : Name, City, Country, State, skill, email, jobtitle etc., that are need to be matched between two profiles that are present in two different sources to check whether those profiles are same or not.

Table 3.1. Attributes taken from Different sites

Different Sites	Data Fields
Proformative, Doximity, Oilpro.	Name, Education, City, State, Country, Skill, Email, Username, Job title

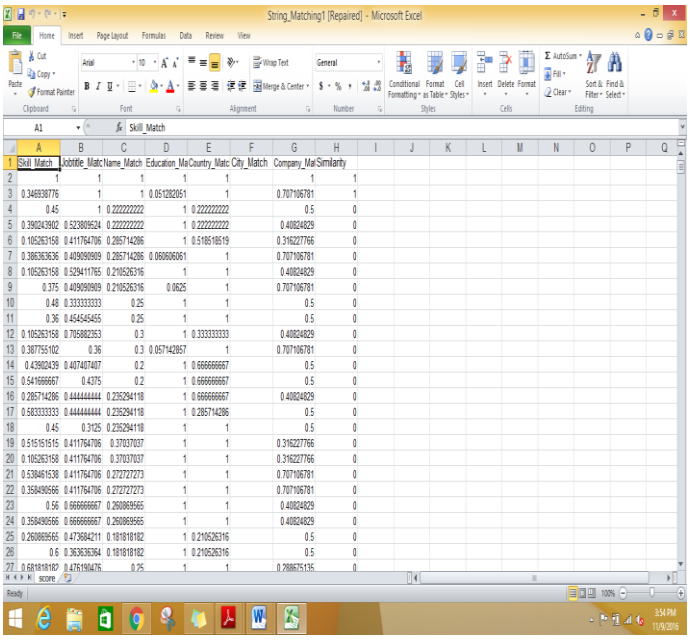

Fig 2. Final results of all similarity measures

4. CONCLUSIONS

All algorithms cannot give more weightage to the matching. In the proposed work we used different algorithms for different attributes like name, age, address and email. By making use of these algorithms duplicate profiles can be easily identified. All the algorithms give more probability to identify the correct match.

REFERENCES

[1] Marco Pennacchiotti and Ana-Maria Popescu, "A Machine Learning Approach to Twitter User Classification" in Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media.

[2] Elie Raad, Richard Chbeir, Albert Dipanda. User pro_le matching in social networks. Network- Based Information Systems (NBiS), Sep 2010, Japan. pp.297-304, 2014. <hal-00643509>.

[3] Benjamin Wassermann, Gottfried Zimmermann "User Profile Matching: A Statistical Approach" In the Proceedings of Fourth International Conference on Advance Human-oriented and Personalized Mechanisms, Technologies, and Services, pp. 60-642011.

[4] OanaGoga, Patrick Loiseau, Robin Sommer, "On the Reliability of Profile Matching Across Large Online Social Networks" In the Proceedings of International Conference.

[5] P. Mika, "Flink: Semantic web technology for the extraction and analysis of social networks," JOURNAL OF WEB SEMANTICS, vol. 3, pp. 211—223, 2005

[6] https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-pytho