

Week 6: TruthFinder algorithm

Understanding, Pseudo Code and Release Plan for OKB-R

<i>Name</i>	<i>Description</i>
M	Number of web sites
N	Number of facts
w	A web site
$t(w)$	The trustworthiness of w
$\tau(w)$	The trustworthiness score of w
$F(w)$	The set of facts provided by w
f	A fact
$s(f)$	The confidence of f
$\sigma(f)$	The confidence score of f
$\sigma^*(f)$	The adjusted confidence score of f
$W(f)$	The set of web sites providing f
$o(f)$	The object that f is about
$imp(f_j \rightarrow f_k)$	Implication from f_j to f_k
ρ	Weight of objects about the same object
γ	Dampening factor
δ	Max difference between two iterations

Table 2: Variables and Parameters of TRUTHFINDER

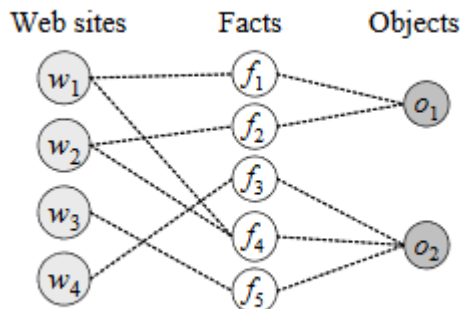


Figure 1: Input of TRUTHFINDER

Trustworthiness: (release 1)

In start we assign some trust value to all sources

We iterate through the process to reach stable state

Until

We measure stability with cosine similarity between the old vector and new vector which are trustworthiness of all websites

Implication (similarity or influence between the facts)

Fact 1 and fact 2 influence each other

If implication is very high between two the other fact also should be influenced by confidence score of other one (release 1)

If Implication is low (negative) there is a conflict between facts (release 1)

Handling Additional Subtlety (release 2)

- We assume websites are different and data is not copied but original
- If fact conflicts with trusted sites it is assumed wrong

We need to define implication according to our scenario (release 1)

Confidence

The confidence score of a fact is just the sum of the trustworthiness scores of websites providing the fact.

The Trustworthiness of website and fact provided are interdependent.

Pseudo Code

Input:

1. Set of Web Sources (W)
2. Set of facts(F)
3. Relationship of inference (Inf) = { (F * W, t)} where t is trustworthiness of facts provided by Source.

Output:

The set of website trust worthiness and confidence of facts

t = {set of trustworthiness}

t1 = {trustworthiness score of website}

s = {set of confidence of facts}

cs = {set of confidence score of facts}

W = Set of sources

F = Set of facts

Initialise matrix A and B:

Sparse matrix A is defined for trustworthiness.

```
function matrixA (W,F)
    for ( int i 0 to size(W)-1)
        for(it(int j 0 to size(F) - 1)
            A[ i ][ j ] = sum( all facts provided by w[i])
```

Sparse Matrix B is defined as

```
function matrixB (A , F)
    for( int i 0 to size( rows in A))
        for (in j 0 to size( column in A))
            if ( A [ i ] [ j ] != 0 and website do not provide another fact)
                B[i][j] = -1
            if website provide other facts
                B[i][j] = p*imp( Fk -> Fj)
            following unit matrices are defined
```

/** we use matrix A and B for each website to find its confidence score

```
using
t = As
cs =Bt1 */
```

Calculating Trustworthiness and confidence

A = matrixA(W, F)

B = matrix(A, F)

for each w1 in W

 initialise t[w1= to //A set value

$$t1(w) = -\ln(1-t(w1))$$

do

$$cs = Bt1$$

compute s using $s(f) = 1 - e(-cs(f))$ // Confidence for the current fact

$$t' = t$$

$$t = As$$

$$\text{compute } t1(w) = -\ln(1-t(w1))$$

while cosine similarity(t, t') > 1- α

Release Plan

1st Release

- 1) Implication definition for our scenario
- 2) Trustworthiness Algorithm implementation

2nd Release

- 1) Handling Additional Subtlety in Trustworthiness
- 2) We will introduce some human interaction in 2nd release for new sources entered into the system

3rd Release

- 1) We mix trustworthiness with other algorithm or techniques to improve
- 2) We analyze the outcomes and push results to WikiData