Towards Semantic-based RSS Merging

F. Getahun, J. Tekli, M. Viviani, R. Chbeir, K. Yetongnon

Laboratoire Electronique, Informatique et Image (LE2I) – UMR-CNRS
Université de Bourgogne – Sciences et Techniques
Mirande, Aile de l'Ingénieur, 9 av. Savary – 21078 Dijon Cedex, France
{fekade-getahun.taddesse, joe.tekli, marco.viviani, rchbeir, kokou}@u-bourgogne.fr

Abstract Merging information can be of key importance in several XML-based applications. For instance, merging the RSS news from different sources and providers can be beneficial for end-users (journalists, economists, etc.) in various scenarios. In this work, we address this issue and mainly explore the relatedness relationships between RSS entities/elements. To validate our approach, we also provide a set of experimental tests showing satisfactory results.

1 Introduction

Really Simple Syndication (RSS) [17] is an XML-based family of web feed formats, proposed to facilitate the aggregation of information from multiple web sources. This way, clients can simultaneously access content originating from different providers rather than roaming a set of news providers, but they often have to read related (and even identical) news more than once as the existing RSS engines[1] do not provide facilities for merging related items.

Merging XML-based documents stands for (*i*) identifying semantically related elements between two documents, and (*ii*) generating a merged document that collapses these related elements preserving remaining source elements. In this work, we address the first problem, and particularly focus on measuring the *semantic relatedness*[2] [2] between RSS elements/items (labels and contents) and consecutively element semantic relationships w.r.t. the meaning of terms and not only their syntactic properties, as a necessary prerequisite to performing efficient RSS merging. To motivate our work, let us consider Figure 1 and Figure 2 showing a list of news (showing only their title and description) extracted from CNN and BBC's RSS feeds. Identifying (and merging) related news would enable the user to more easily and efficiently acquire information. XML news feeds (e.g., RSS items) can be related in different manners:

- The content of an element might be totally included in another (*inclusion*).

  Example 1. The title content of *CNN1* "Hong Kong cheers Olympic torch" includes the title content of *BBC1* "Torch cheered through Hong Kong"[3].
- Two news may refer to similar and related concepts (*intersection*).

  Example 2. The title content of *CNN2* "Bush wants $700 million more emergency food

[1] AmphetaDesk, MetaDot, Meerkat, Portal Software, PullRss, Radio UserLand, SlashCode/Slashdot, Weblog 2.0 aggregate, search, filter or display news in RSS format.

[2] Semantic relatedness is a more general concept than similarity. Dissimilar entities may also be semantically related by lexical relations such as meronymy and antonymy, or just by any kind of functional relation or frequent association.

[3] After text pre-processing such as stop-word removal, stemming, and semantic analysis

aid" and title content of *BBC2* "US president offers $700m for food crisis" are related and very similar, they share some words/expressions ('$700m', 'food') and semantically related concepts ('emergency' and 'crisis', 'US President' and 'Bush').

```
<CNN_RSS>
    <item>
        <title>Hong Kong cheers Olympic torch</title>
        <description>Hong Kong stages the Olympic torch relay, the first time the event is held on the soil of the host of this year's Summer Games, with thousands lining the route in support of the event and a peppering of protests.</description>
    </item>                                                                                                                                                                                                                    CNN1
    <item>
        <title>Bush wants $770 million more emergency food aid</title>
        <description>U.S. President George W. Bush urges Congress to approve $770 million in new global food aid to be made available beginning in October. The sum would be in addition to$200 million in emergency food aid announced two weeks ago.</description>
    </item>                                                                                                                                                                                                                    CNN2
</CNN_RSS>
```

Fig. 1. RSS news extracted from CNN

```
<BBC_RSS>
    <item>
        <title>Torch cheered through Hong Kong</title>
        <description>Cheering crowds and a few protesters turn out in Hong Kong to watch the Olympic torch parade.</description>
    </item>                                                                                                                                                                                                                    BBC1
    <item>
        <title>US president offers $770m for food crisis</title>
        <description>George W Bush offers $770m (£390m) in new international food aid to help ease the effects of surging food prices.</description>
    </item>                                                                                                                                                                                                                    BBC2
</BBC_RSS>
```

Fig. 2. RSS news extracted from BBC

Hence, the main objective of this study is to put forward a specialized XML relatedness measure, dedicated to the comparison of RSS items, able to (*i*) identify RSS items that are related enough to be merged and (*ii*) identify the relationships that can occur between two RSS items (i.e., *disjointness, intersection*, *inclusion*, and *equality*), to be exploited in the merging phase. Identifying common/different parts in the items to be merged would help decide on the merging rules to be executed in different application scenarios. Note that the merging phase itself (merging rules, merging process, …) is not developed in this paper. The remainder of this paper is organized as follows. In Section 2, we discuss background and related work. Section 3 defines basic concepts to be used in our measure. Section 4 details our RSS relatedness measure. Section 5 presents experimental results. Finally, Section 6 concludes this study and draws future research directions.

2 Related Work

Identifying correspondence or matching nodes is a known pre-condition in schema matching [4] and merging XML document [9]. In schema matching, corresponding nodes or elements are identified using the match operator. A lot of research has been done to determine similarity and are categorized into structure-based, semantic-based and hybrid-based approaches. It is to be noted that most of the proposed approaches in XML comparison are based on structural similarity using tree edit distance [1]. Chawathe [3], Nireman and Jagadish [12] consider the minimum number of edit operations: insert, delete and/or move to transform one XML tree to another. Also, the use of Fast Fourier Transform [5] has been proposed to compute similarity between XML documents.

The semantic similarity between concepts is estimated either by the distance between nodes [19] or the content of the most specific common ancestor of those nodes involved in the comparison [15][10] and defined according to some predefined knowledge base(s). Knowledge bases [14][16](thesauri, taxonomies and/or ontologies) provide a framework for organizing words (expressions) into a semantic space. In Information Retrieval (IR) [11], the content of a document is commonly modeled with set/ bag of words where each concept (and subsumed word(s)) is given a weight computed with Term Frequency (TF), Document Frequency (DT), Inverse Document Frequency (IDF), and the combination TF-IDF. In [7], the authors used a Vector Space having TF-IDF as weight factor in XML retrieval.

More recently, there are hybrid-based approaches that attempted to address XML comparison. In a recent work [18], the authors combined an IR semantic similarity technique with a structural-based algorithm based on edit distance. However, the semantic similarity is limited only to tag name. In [8], *xSim,* a structure and content aware XML comparison framework is presented. *xSim* computes the matching between XML documents as an average of matched list similarity values. The similarity value is computed as average of content, tag name and path similarity values without considering semantics.

However and to the best of our knowledge, none of the current techniques or measures identifies the semantic relationship between documents and semantic relatedness on content in general or items in particular and none of the approaches is RSS-focused.

3 Preliminaries

In the following, we define the basic concepts used in our approach and particularly detail RSS data model and hierarchal neighbourhood of a concept.

3.1 RSS data model

An RSS document comes down to a well-formed XML document (represented as a rooted ordered labeled tree following the Document Object Model (DOM) [20]) w.r.t. an RSS schema [17]. Note that different RSS schemas exist, corresponding to the different versions of RSS[4] available on the web. Nonetheless, analyzing different versions of RSS, we can see that RSS items consistently follow the same overall structure, adding or removing certain elements depending on the version at hand.

Definition 1 [Rooted Ordered Labeled Tree]
It is a rooted tree in which the nodes are labeled and ordered. We denote by R(T) the root of T.

Definition 2 [Element]
Each node of the rooted labeled tree T is called an *element* of T. Each element e is a pair $e = \langle \eta, \varsigma \rangle$ where $e.\eta$ refers to the element name and $e.\varsigma$ to its content. $e.\eta$ gen-

[4] RSS refers to one of the following standards: Rich Site Summary (RSS 0.91, RSS 0.92), RDF Site Summary (RSS 0.9 and 1.0), and Really Simple Syndication (RSS 2.0).

erally assumes an atomic text value (i.e., a single word/expression) whereas $e.\varsigma$ may assume either an atomic text value, a composite text value (sentence, i.e., a number of words/expressions), or other elements[5].

Definition 3. [Simple/Composite Element]
An element *e* is *simple* if $e.\varsigma$ assumes either an atomic or composite textual value[6]. In XML trees, simple elements come down to leaf nodes. Content and value of simple element are used interchangeably in this paper.
An element *e* is *composite* if $e.\varsigma$ assumes other elements. In XML trees, composite elements correspond to inner nodes.

Definition 4. [RSS Item Tree]
An *RSS item tree* is an XML tree *T* having one single composite element, the root node *r* (usually with $r.\eta$ = '*item*'), and *k* simple elements $\{n_1, \ldots, n_k\}$ describing the various RSS item components.

3.2 Knowledge Base

A *Knowledge Base* [16] (thesauri, taxonomy and/or ontology) provides a framework for organizing entities (words/expressions, generic concepts, web pages, etc.) into a semantic space. In our study, it is used to help computing relatedness and formally defined as $KB = (C, E, R, f)$ where *C* is the set of concepts (synonym sets as in WordNet [14]), *E* is the set of edges connecting the concepts, $E \subseteq C \times C$, *R* is the set of semantic relations, $R = \{\equiv, \prec, \succ, <<, >>, \Omega\}$ [7], the synonymous words/expressions being integrated in the concepts, *f* is a function designating the nature of edges in *E*, $f : E \rightarrow R$.

Following Definition 4, all elements of an RSS item (to the exception of the root) are simple, i.e., each composed of a label and a textual value (content). Hence, assessing the relatedness between (simple) RSS elements requires considering label as well as textual value relatedness. To that end, we introduce two knowledge bases: (*i*) *value-based*: to describe the textual content of RSS elements, and (*ii*) *label-based*: to organize RSS labels[8].

3.3 Neighborhood

In our approach, the *neighborhood* of a concept C_i underlines the set of concepts $\{C_j\}$, in the knowledge base, that are subsumed by C_i w.r.t. a given semantic relation. The

---

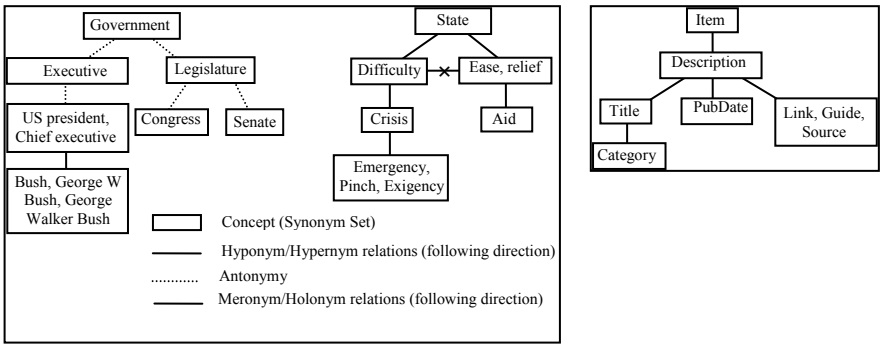
[5] We do not consider *attributes* in evaluating RSS item relatedness since they do not affect the semantic comparison process. Nonetheless, attributes will be considered in the merging phase.

[6] In this paper, we do not consider other types of data contents, e.g., numbers, dates, …

[7] R underlines respectively the synonym ($\equiv$), hyponym (Is-A or $\prec$), hypernym (Has-A or $\succ$), meronym (Part-Of or <<), holonym (Has-Part or >>) and Antonym (Ω) relations, as defined in [6].

[8] Note that one single knowledge base could have been used. However, since XML document labels in general, and RSS labels in particular, depend on the underlying document schema, an independent *label-based* knowledge base, provided by the user/administrator, seems more appropriate than a more generic one such as WordNet (treating generic textual content).

concept of neighborhood, introduced in [6], is exploited in identifying the relationships between text (i.e., RSS element labels and/or textual contents) and consequently RSS elements/items.



a. Two sample value KBs with multiple root concepts extracted from WordNet

b. Sample RSS label KB

Fig. 3. Sample value and label knowledge bases

Definition 5 [Semantic Neighborhood]

The *semantic neighborhood* of a concept C_i (i.e. $N_{KB}^{R}(C_i)$) is defined as the set of concepts $\{C_j\}$ (and consequently the set of words/expressions subsumed by the concepts) in a given knowledge base *KB*, related with C_i via the hyponymy or meronymy semantic relations, directly or via transitivity.

Definition 6 [Global Semantic Neighborhood]

The *global semantic neighborhood* $\overline{N_{KB}}(C_i)$ of a concept is the union of each semantic neighborhood w.r.t. all synonymy, hyponymy and meronymy relations altogether.

3.4 Text Representation

As illustrated previously, RSS (simple) element labels and contents underline basic text (cf. Definition 2). Thus, hereunder we define the idea of *concept set* to represent a piece of text. It will be exploited in representing (and consequently comparing) RSS element labels and contents.

Definition 7 [Concept Set]

Consider a textual value *t*, composed of a set of terms $\{k_1, \ldots, k_n\}$, where *n* is the total number of distinct terms in *t*, i.e., $|t|$. The *concept set* of *t*, denoted as *CS*, is a set of concepts $\{C_1, \ldots, C_m\}$, where each C_i represents the meaning of a group of terms in $\{k_1, \ldots, k_n\}$, where *m* is the total number of concepts describing *t*, i.e., $m = |CS_t|$, having $0 \leq |CS_t| \leq |t|$. Concept C_i is assumed to be obtained after several textual preprocessing operations such as stop-words removal[9], stemming[10], etc.

[9] Stop-words are very common words such as prepositions, demonstrative, articles, etc which do not provide useful information to distinguishing the content of the items (e.g. yet, an, but, …).

Definition 8 [Text Vector Space]
Let t_i be a text value described by concept set $CS_i = \{C_1, \ldots, C_n\}$. Following the vector space model used in information retrieval [11], we represent t_i as a vector V_i in an n-dimensional space such as: $V_i = [\langle C_1, w_1\rangle, \ldots, \langle C_n, w_n\rangle]$, where w_i represents the weight associated to dimension (concept) C_i. Given two texts t_1 and t_2, the vector space dimensions represent each a distinct concept $C_i \in CS_1 \cup CS_2$, such as $1 \le i \le n$ where $n = |CS_1 \cup CS_2|$ is the number of distinct concepts in both CS_1 and CS_2.

Definition 9 [Vector Weights]
Given a collection of texts T, a text $t_i \subset T$ and its corresponding vector V_i, the weight w_i associated to a concept C_i in V_i is calculated as $w_i = 1$ if the concept C_i is referenced in the vector V_i; otherwise, it is computed based on the maximum *enclosure similarity* it has with another concept C_j in its corresponding vector V_j.

$$enclosure\_sim(C_i, C_j) = \frac{|\overline{N_{KB}}(C_i) \cap \overline{N_{KB}}(C_j)|}{|\overline{N_{KB}}(C_j)|} \quad (1)$$

enclosure_sim(C_i, C_j) takes into account the global semantic neighborhood of each concept. This measure returns a value of 1 if C_i includes C_j.

Example 3. Let us consider titles of RSS items *CNN2* and *BBC2* (Figures 1, 2). The corresponding vector representations V_1 and V_2 are shown in Figure 4. For the sake of simplicity, we consider that only these two texts make up the new items.

	Bush	*want*	*700m*	*emergency*	*food*	*aid*	*US president*	*offer*	*crisis*
V_1	1	1	1	1	1	1	**1**	0	**1**
V_2	**0.56**	0	1	**0.4**	1	0	1	1	1

Fig. 4. Vectors obtained when comparing title texts of RSS items *CNN2* and *BBC2*

Vector weights are evaluated in two steps. First, for each concept C in V1 and V2, we assign value of 1 if C exists in the concept sets corresponding to the texts being compared. Second, we update the weight of those concepts having value of zero with maximum semantic enclosure similarity value.
Following the WordNet extract in Figure 3a, the concept 'US president' is included in the global semantic neighborhood of 'Bush', i.e., $US\ president \in \overline{N_{KB}}(Bush)$. Hence, *enclosure_sim*(*US president*, *Bush*) = 1. However, in V_2, *enclosure_sim*(*Bush*, *US president*) = 0.56. Likewise, 'Crisis' is included in the global semantic neighborhood of 'Emergency', i.e., $Crisis \in \overline{N_{KB}}(Emergency)$. Thus, *enclosure_sim*(*Crisis*, *Emergency*) = 1 but *enclosure_sim*(*Emergency*, *Crisis*) = 0.4.

4 RSS Relatedness Measure

As motivated in the beginning of the paper, a dedicated relatedness/similarity measure is needed as a prerequisite to merging RSS data. This section details the measures used for text, simple and complex element relatedness.

[10] Stemming the process for reducing inflected (or derived) words to their stem or base form (e.g., "*housing*", "*housed*" → "*house*")

4.1 Text Relatedness

Given two texts t_1 and t_2, *Textual Relatedness* (*TR*) algorithm returns a doublet, combining the semantic relatedness *SemRel* value and the relationship *Relation* between t_1 and t_2. Formally, it is denoted as:

$$TR(t_1, t_2) = \langle SemRel(t_1, t_2), Relation(t_1, t_2) \rangle \quad (2)$$

SemRel value is computed using vector based similarity measure (e.g. cosine [11]) applied to text vector having weights underlining concept existence and enclosure in the concept set of both text inputs (definition 9).

The relationship between two texts t_1 and t_2 is identified as follows:

- *Relation*(t_1, t_2) = *Disjointness*, i.e., $t_1 \rhd\lhd t_2$, if there is no relatedness whatsoever between t_1 and t_2 i.e., *SemRel*(t_1, t_2) = 0.
- *Relation*(t_1, t_2) = *Inclusion*, i.e., $t_1 \supset t_2$, if the product of the weights of vector V_1 (describing t_1) is equal to 1, i.e., $\Pi_{V_1}(w_p) = 1$. The weight product of V_1 underlines whether or not t_1 encompasses all concepts in t_2.
- *Relation*(t_1, t_2) = *Intersection*, i.e., $t_1 \cap t_2$, if t_1 and t_2 share some semantic relatedness, i.e., *SemRel*(t_1, t_2) > 0, and the product of the weights of both vectors V_1 and V_2 are equal to zero, i.e., $\Pi_{V_1}(w_p) \geq 0$ and $\Pi_{V_2}(w_q) \geq 0$.
- *Relation*(t_1, t_2) = *Equality*, i.e., $t_1 = t_2$, if corresponding vectors are identical, i.e., $SemRel(t_1, t_2) = 1$.

  Example 4. Considering Example 3, (t_1 of *CNN2* and t_2 of *BBC2*), *SemRel*(t_1, t_2) = 0.68 and *Relation*(t_1, t_2) = *Intersection*. Hence, *TR*(t_1 , t_2) = <*0.68, Intersection*>.

4.2 RSS Item Relatedness

As shown previously, quantifying the semantic relatedness and identifying the relationships between two RSS items amounts to comparing corresponding elements. This in turn comes down to comparing corresponding RSS (simple) element labels and values (contents), which simplify to basic pieces of text (cf. Definition 2). The relatedness between two simple elements is computed applying *TR* Algorithm for both text content and label. ER algorithm accepts two elements e_1 and e_2 as input and returns doublet quantifying the semantic relatedness *SemRel* and the relationships *Relation* between e_1 and e_2 based on corresponding label and value relatedness. *SemRel*(e_1, e_2) semantic the relatedness value between elements, is quantified as *weighted sum* value of label and value relatedness as:

$$SemRel(e_1, e_2) = w_{Label} \times LB_{Sem\,Rel} + w_{Value} \times VR_{Sem\,Rel} \quad (3)$$

where $w_{Label} + w_{Value} = 1$ and (w_{Label}, w_{Value}) ≥ 0.

The relation between elements is computed based on rule rule-based method that combines label and value relationships as follows:

- Elements e_1 and e_2 are *disjoint* if either their labels and values are disjoint
- Element e_1 *includes* e_2, if $e_1.\eta$ includes $e_2.\eta$ and $e_1.\varsigma$ includes $e_2.\varsigma$
- Two elements e_1 *and* e_2 *intersect if either th*eir *labels* o*r* v*a*lue$_s$ *i*nter*sect*
- Two elements e_1 and e_2 are *equal* if both their labels and values are equal.

Having identified the semantic relatedness and relationships between simple elements, Algorithm 1 evaluates RSS item relatedness. Given two RSS items I_1 and I_2, each made of a bunch of elements, *Item Relatedness* (*IR*) algorithm quantifies the semantic relatedness and identifies the relationships between I_1 and I_2 based on corresponding element relatedness (lines 7 – 12). Line 9 computes the relatedness between simple elements e_i and e_j and returns semantic relatedness eij_{SemRel}, and relationship $eij_{Relation}$. In line 10, semantic relatedness value eij_{SemRel} is accumulated to get grand total, and, in line 11, $eij_{Relation}$ is stored for later use. In line 13, the semantic relatedness value between I_1 and I_2 is computed as the average of the relatedness values between corresponding element sets I_1 and I_2.

Algorithm 1: ***IR Algorithm***	Line
Input: I_1, I_2: element // the two items (Complex elements)	1
Variable: eij_{SemRel}: double // semantic relatedness values e_i and e_j	
$eij_{Relation}$: string // relationship value between e_i and e_j	
$Eij_{Relation\_set}$: Set // would contain sub-elements relationship values	
Output: *SemRel*: double // relatedness value between I_1 and I_2	
Relation: String // relationship value between I_1 and I_2	
$SumRel$=0	
$Eij_{Relation\_set} = \varnothing$	
For each e_i In I_1	7
For each e_j In I_2	
$<eij_{SemRel}, eij_{Relation}> = ER(e_i, e_j)$	9
$Eij_{Relation\_set} = Eij_{Relation\_set} \cup eij_{Relation}$	
$SumRel = SumRel + eij_{SemRel}$	
Next	
Next	
$SemRel = SumRel / \|I_1\| \times \|I_2\|$	13
$Relation = I_{Relation}(\{Eij_{Relation\_set}\})$ // $\forall i \in [1,\|I_1\|], \quad \forall j \in [1,\|I_2\|]$	
Return <*SemRel, Relation*>	15

As for the relationships between two items, we develop a rule-based method $I_{Relation}$ (line 14) for combining sub-element relationships stored in $Eij_{Relation\_set}$ (which is the relationship between e_i and e_j) as follows:

- Items I_1 and I_2 are *disjoint* if all elements $\{e_i\}$ and $\{e_j\}$ are disjoint (elements are disjoint if there is no relatedness whatsoever between them, i.e., $SemRel(I_1, I_2) = 0$)
- Item I_1 *includes* I_2, if all elements in $\{e_i\}$ include all those in $\{e_j\}$
- Two items I_1 and I_2 *intersect* if at least two of their elements intersect
- Two items I_1 and I_2 are *equal* if all their elements in $\{e_i\}$ equal to all those in $\{e_j\}$.

Example 5. Let us consider RSS items *CNN2* and *BBC2* (Figure 1, 2). Corresponding item relatedness is computed as follows. Notice that $w_{Label} = 0.1$ *and* $w_{Value} = 0.9$ *is* used while computing simple element relatedness (cf. 3). Below, each cell represent doublet returned by simple element relatedness ER algorithm.

ER	$title_{BBC2}$	$description_{BBC2}$
$title_{CNN2}$	<0.700, intersection>	<0.526, intersection>

$description_{CNN2}$	<0.483, intersection>	<0.435, intersection>

Using (line 13), *SemRel*(*CNN2*, *BBC2*) = (0.700+0.526+0.483+0.538+0.435)/ 2 × 2 = 0.671. where $|I_1|$ and $|I_2|$ are equal to 2.

Relation(*CNN2*, *BBC2*) = *Intersection* since a number of their elements intersect.

Hence, *IR*(*CNN2*, *BBC2*) =<0.671, Intersection>

5 Experiments

We have conducted a set of experiments to conform the computational complexity and efficiency of our relatedness measure in comparison with current approaches. All the experiments were carried out on Intel Core Centrino Duo Processor machine (with processing speed of 1.73.0 GHz, 1GB of RAM). The experiments related to measure the efficiency of our relatedness measure won't be detailed due to the lack of space. We compared the efficiency of *xSim* [8], TF-IDF and our algorithm while identifying the relatedness of randomly generated synthetic news. The aim here was to compare our time processing with others. In all algorithms, relatedness identification is done without semantics as both *xSim* and TF-IDF do not consider semantics information. Figure 5 shows that our semantic relatedness method provides efficient result compared to *xSim* and it is less efficient compared to TF-IDF (which doesn't consider the structure of the RSS news item). Hence, our relatedness algorithm is efficient and also identifies relationship between text, element and items which is not the case in both *xSim* and TF-IDF.

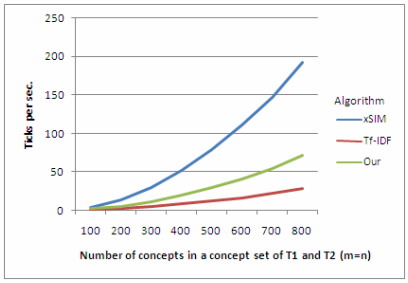


Fig 5 Timing result obtaioned using three algorithms: *xSim*, TF-IDF and our algorithm

6. Conclusions and Future Directions

In this paper, we have addressed the issue of measuring relatedness between RSS items, a pre-condition for merging. We have studied and provided a technique for texts, simple elements and items relatedness computation, taking into account different kinds of relationship among texts, elements and items. We have developed a prototype and compared the efficiency of our algorithm against *xSim* and TF-IDF. Our measure will help us in making decisions about merging rules to apply to clustered elements. Currently, we are investigating the relevance of the topological relationships in the construction of the merging rules. Later on, we are willing to extend our work so to address XML documents merging in multimedia scenarios (SVG, MPEG-7, etc.).

References

[1] P. Bille. A survey on tree edit distance and related problems. Theoretical Computer Science, 337(1-3):217-239, 2005.

[2] A. Budanitsky and G. Hirst. Evaluating WordNet-based measures of lexical semantic relatedness. Computational Linguistics, 32(1):13-47, 2006.

[3] S. S. Chawathe. Comparing hierarchical data in external memory. In VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases, pages 90-101, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[4] Erhard Rahm and Philip.A. Bernstein. A survey of approaches to automatic schema matching, VLDB Journal 10 (2001) (4), pp. 334–350

[5] S. Flesca, G. Manco, E. Masciari, and L. Pontieri. Fast detection of xml structural similarity. IEEE Transactions on Knowledge and Data Engineering, 17(2):160-175, 2005. Student Member-Andrea Pugliese.

[6] F. Getahun, J. Tekli, S. Atnafu, and R. Chbeir. Towards efficient horizontal multimedia database fragmentation using semantic-based predicates implication. In XXII Simposio Brasileiro de Banco de Dados (SBBD'07), 15-19 Oct., Joao Pessoa, Brazil, pp. 68-82, 2007.

[7] T. Grabs and H.-J. Schek. Generating Vector Spaces On-the-fly for Flexible XML Retrieval. In Proceedings of the ACM SIGIR Workshop on XML and Information Retrieval, Tampere, Finland, pages 4–13. ACM Press, 2002.

[8] A. M. Kade and C. A. Heuser Matching XML documents in highly dynamic applications. Proceeding of the eighth ACM symposium on Document engineering ISBN:978-1-60558-081-4, Sao Paulo, Brazil, Pages 191-198 (2008).

[9] R. La Fontaine. Merging XML files: A new approach providing intelligent merge of XML data sets. In Proceedings of XML Europe '02, 2002.

[10] Lin D. An Information-Theoretic Definition of Similarity. In Proceedings of the 15th International Conference on Machine Learning, 296-304, Morgan Kaufmann Publishers Inc., 1998

[11] M. J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.

[12] A. Nierman and H. V. Jagadish. Evaluating structural similarity in XML documents. In Proceedings of the Fifth International Workshop on the Web and Databases, WebDB 2002, pages 61-66. University of California, 2002.

[13] M. F. Porter. 1980. An algorithm for suffix stripping. Program, 14(3):130—137.

[14] Princeton University Cognitive Science Laboratory. WordNet: a lexical database for the English language. http://wordnet.princeton.edu/.

[15] P. Resnik. Semantic Similarity in a Taxonomy: An Information-based Measure and its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research, 11:95–130, 1999.

[16] R. Richardson and A. F. Smeaton. Using wordnet in a knowledge-based approach to information retrieval. Technical Report CA-0395, School of Computer Applications, Trinity College, Dublin, Ireland, 1995.

[17] RSS Advisory Board. RSS 2.0 Specification. http://www.rssboard.org/.

[18] J. Tekli, R. Chbeir, and K. Ytongnon. A hybrid approach for xml similarity. In J. van Leeuwen, G. F. Italiano, W. van der Hoek, C. Meinel, H. Sack, and F. Plasil, editors, SOFSEM '07, Proceedings, volume 4362 of Lecture Notes in Computer Science, pages 783-795. Springer, 2007.

[19] Z. Wu and M. Palmer. V Verbs semantics and lexical selection. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pages 133-138, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[20] WWW Consortium. The Document Object Model, http://www.w3.org/DOM.