Relating RSS News/Items

Fekade Getahun, Joe Tekli, Chbeir Richard, Marco Viviani, Kokou 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, rchbeir, marco.viviani,
kokou.yetongnon}@u-bourgogne.fr

Abstract. Identifying related RSS news (coming from one or different sources and providers) can be beneficial for end-users (journalists, economists, etc.) in various scenarios (merging, filtering, access control, etc.). In this paper, we provide a practical approach to both measure the relatedness/similarity and identify relationships between RSS entities/elements. Our approach is based on the concepts of semantic neighborhood and vector space and is able to consider the content and the structure of RSS. Here, we also show the set of experimental tests conducted to validate our approach.

Keywords: RSS Relatedness, Similarity, Relationships, Neighbourhood

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, often having to read related (and even identical) news more than once as the existing RSS engines[1] do not provide facilities for identifying and handling such items.

Computing relatedness between XML documents in general and RSS in particular is beneficial in different scenarios such as merging, filtering, access control, etc. In this work, we address *semantic relatedness*[2] [2] between RSS elements/items (labels and contents) and consecutively element semantic relationships with respect to (w.r.t.) the meaning of terms and not only their syntactic properties. To motivate our work, let us consider Figure 1 and Figure 2 showing a list of news extracted from CNN and BBC's RSS feeds. Identifying related news would enable the user to more easily and efficiently acquire and/or merge information. XML news feeds (e.g., RSS items) can be related in different manners:

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

  Example 1. The title content of *CNN1* "U.N. chief launches $600M Gaza aid appeal" includes the title

[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.

content of *BBC1* "UN launches $613m appeal for Gaza"[3].

<CNN_RSS>	
<item> <title>U.N. chief launches $600M Gaza aid appeal</title> <description> United Nations Secretary-General Ban Ki-moon on Thursday launched a humanitarian appeal to provide emergency aid to the people of Gaza in the aftermath of Israel's military offensive in the region.</description> </item>	*CNN1*
<item> <title>Ford reports $5.9 billion loss in the fourth-quarter </title> <description>Ford Motor reported that its ongoing losses soared in the fourth quarter, but the company reiterated it still does not need the federal bailout already received by its two U.S. rivals.</description> </item>	*CNN2*
<item> <title>The youth forum cancels scheduled demonstration</title> <description>The international youth forum cancels the call for stop-war demonstration due to security reason</description> </item>	*CNN3*
</CNN_RSS>	

Fig. 1. RSS news extracted from CNN

<BBC_RSS>	
<item> <title> UN launches $613m appeal for Gaza </title> <description> The UN will launch an appeal for $613m to help people affected by Israel's military offensive in Gaza, the body's top official says </description> </item>	*BBC1*
<item> <title> Ford reports record yearly loss </title> <description> US carmaker Ford reports the biggest full-year loss in its history, but says it still does not need government loans.</description> </item>	*BBC2*
<item> <title>Youth's form call for demonstration</title> <description> International youth forum call demonstration as part of stop the war </description> </item>	*BBC3*
</BBC_RSS>	

Fig. 2. RSS news extracted from BBC

- Two news may refer to similar and related concepts (*intersection*).

  Example 2. The description content of *CNN2* "Ford Motor reported that its ongoing losses soared in the fourth quarter, but the company reiterated it still does not need the federal bailout already received by its two U.S. rivals." and description content of *BBC2* "US carmaker Ford reports the biggest full-year loss in its history, but says it still does not need government loans." are related and very similar, they share some words/expressions ('Ford', 'report', 'loss', 'US') and semantically related concept ('fourth quarter', 'year'), ('biggest', 'soar'), ('reiterate', 'say'), ('federal bailout'), and ('government loan').

- News might be opposite but refer to the same issue (*oppositeness*).

  Example 3. "The international youth forum cancel call for stop-war demonstration due to security reason" (description of CNN3) and "International youth forum call demonstration as part of stop the war" (description of BBC3) can be considered as opposite because of the use of antonym expressions 'call' and 'cancel call'.

Hence, the main objective of this study is to put forward a specialized XML relatedness

[3] After a pre-process of stop word removal, stemming, ignoring non textual values and semantic analysis.

measure, dedicated to the comparison of RSS items, able to identify (*i*) RSS items that are related enough and (*ii*) the relationship that can occur between two RSS items (i.e., *disjointness, intersection*, *inclusion*, *antonomy* and *equality*).

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. In Section 4, we detail how the relatedness and relationship between text values are computed. Section 5 details our RSS elements relatedness and relationship measures. Section 6 presents experimental results. Finally, Section 7 concludes this study and draws some future research directions.

2 RELATED WORK

Identifying correspondence or matching nodes in hieratically organized data such as XML is a pre-condition in different scenarios [9]. A lot of research has been done to determine similarity and can be 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 (tree), delete (tree) and update node to transform one XML tree to another. Also, the use of Fast Fourier Transform [4] has been proposed to compute similarity between XML documents.

The semantic similarity between concepts is estimated either by the distance between nodes [20] or the content of the most specific common ancestor of those nodes involved in the comparison [15] [10] and is 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 sets/bags words where each word (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 [6], 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 names. 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.

The relationships between objects such as equality, inclusion, intersection, disjointness, etc. have been used in different applications such as spatial data retrieval, access control and text mining. However and to the best of our knowledge, none of the current techniques or measures identifies the semantic relationships between documents (and RSS news) or identifies the semantic relatedness on content in general.

3 PRELIMINARIES

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) [19]) w.r.t. an RSS schema

[17]. Note that different RSS schemas exist, corresponding to the different versions of RSS available on the web (RSS 0.9x[4], 1.0[5], and 2.0). 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. $e.\eta$ generally 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[6].

Definition 3. [Simple/Composite Element]
An element *e* is *simple* if $e.\varsigma$ assumes either an atomic or composite textual value[7]. In XML trees, simple elements come down to leaf nodes.
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.η* = '*item*'), and *k* simple elements {n_1, …, 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 is formally defined as *KB* = (*C*, *E*, *R*, *f*) where *C*[8] is the set of concepts (synonym sets of words/expressions 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, $R = \{\equiv, \prec \succ <<, >>, \Omega\}$[9], the synonymous words/expressions being integrated in the concepts, *f* is a function designating the nature of edges in *E*, $f : E \rightarrow R$.

As assessing the relatedness between (simple) RSS elements requires considering label as well as textual value relatedness. We introduced two knowledge bases: (*i*) *value-based*: to describe the textual content of RSS elements, and (*ii*) *label-based*: to organize RSS labels. Note that one single knowledge base could have been used. However, since XML document labels in

[4] RSS 0.92 is upward compatible to RSS 0.91. http://backend.userland.com/rss09x
[5] RSS 1.0 (also called RDF Site Summary) conforms to the W3C's RDF Specification and is extensible via XML-namespace and/or RDF based modularization. http://web.resource.org/rss/1.0/spec
[6] We do not consider *attributes* in evaluating RSS item relatedness since they do not affect the semantic comparison process.
[7] Here, we do not consider other types of data contents, e.g., numbers, dates, …
[8] A concept $C_i \in C$ in KB has a depth *d* representing the length of path from the root of the KB.
[9] The symbols in R underline respectively the synonym (≡), hyponym (Is-A or ≺), hypernym (Has-A or ≻), meronym (Part-Of or <<), holonym (Has-Part or >>) and Antonym (Ω) relations, as defined in [5].

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).

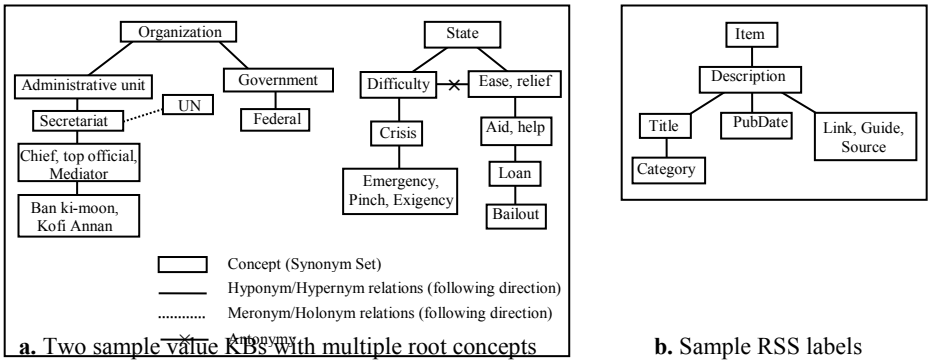
3.3 Neighborhood

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 concept of neighborhood, introduced in [5], is exploited in identifying the relationships between text (i.e., RSS element labels and/or textual contents) and consequently RSS elements/items.

Definition 5 [Semantic Neighborhood]
The *semantic neighborhood* of a concept 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 ($\prec$) or meronymy ($<<$) semantic relations, directly or via transitivity. It is formally defined as:

$$N_{KB}^{R}(C_i) = \{C_j / C_i R C_j \wedge R \in \{\equiv, \prec, <<\}\} \quad (1)$$

Note that the neighborhood of a concept w.r.t. the synonymy relation ($\equiv$) is the concept itself, i.e., the set of synonymous words/expressions subsumed by the concept.



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

b. Sample RSS labels

Fig. 3. Sample value and label knowledge bases

Definition 6 [Global Semantic Neighborhood]
The *global semantic neighborhood* of a concept is the union of each semantic neighborhood w.r.t. all synonymy ($\equiv$), hyponymy ($\prec$) and meronymy ($<<$) relations altogether. Formally:

$$\overline{N_{KB}}(C_i) = \bigcup \qquad \in \{\equiv, \prec \quad (2)$$

Definition 7 [Antonym Neighborhood]
The antonym neighborhood of a concept C_i is defined as the set of concepts $\{C_i\}$, in a given knowledge base KB, related with C_i via the antonymy relation (Ω), directly or transitively via synonymy($\equiv$), hyponymy ($\prec$) or hypernym ($\succ$). Formally:

$$N_{KB}^{\Omega}(C_i) = \{C_j, C_k \; / \; C_i \; R_1 \; C_m \; \Omega \; C_j \wedge C_j \; R_2 \; C_k$$
$$having \; R_l \in \{\equiv, \prec \qquad \succ \quad (3)$$

4 TEXT RELATEDNESS AND RELATIONS

4.1 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 8 [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 pre-processing operations such as stop-words removal[10], stemming[11], etc.

Definition 9 [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 $1 \leq i \leq n$ where $n = |CS_1 \cup CS_2|$ is the number of distinct concepts in both CS_1 and CS_2.

Definition 10 [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. Formally, it is defined as:

$$w_i = \begin{cases} 1 & \text{if } freq(C_i) > 0 \\ \max(Enclosure\_sim(C_i, C_j)) & \text{otherwise} \end{cases} \quad (4)$$

$$Enclosure\_sim(C_i, C_j) = \frac{|\overline{N_{KB}}(C_i) \cap \overline{N_{KB}}(C_j)|}{|\overline{N_{KB}}(C_j)|} \quad (5)$$

Enclosure_sim(C_i, C_j) takes into account the global semantic neighborhood of each concept. It is asymmetric, allows the detection of the various kinds of relationships between RSS items, and returns a value equal to 1 if C_i includes C_j.

Example 4. Let us consider description of RSS items *CNN2* and *BBC2* (Figures 1, 2). The partial 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.

[10] Stop-words identify words/expressions which are filtered out prior to, or after processing of natural language text (e.g., *yet, an, but, the,* …) which is done using stop list.

[11] Stemming is the process for reducing inflected (or sometimes derived) words to their stem, i.e., base or root (e.g., "*housing*", "*housed*" → "*house*").

	Ford	*report*	*loss*	…	*reiterate*	*Fourth-auarter*	*Federal*	*Bailout*	*Big*	*Full-year*	*say*	*governme nt*	*loan*
V_1	1	1	1	…	1	1	1	1	**1**	**1**	**1**	**1**	**1**
V_2	1	1	1	**…**	**1**	**0**	**0.67**	**0.86**	1	1	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 V_1 and V_2, we check the existence of C in each of 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 subset extract in Figure 3a, the concept 'Government' is included in the global semantic neighborhood of 'Federal', i.e., $government \in \overline{N_{KB}}(federal)$. Hence, it has the maximum enclosure similarity with 'federal', i.e., *Enclosure_sim*(federal, government) = 1. However, in V2, *Enclosure_sim*(government, federal) = 0.67.
Likewise, 'loan' is included in the global semantic neighborhood of 'bailout', i.e., $loan \in \overline{N_{KB}}(bailout)$. This way *Enclosure_sim*(loan,bailout) = 1 and *Enclosure_sim*(bailout, loan) = 0.86.

4.2 Relatedness and Relations

Given two texts t_1 and t_2, *Textual Relatedness* (*TR*) algorithm shown in Algorithm 1 returns a doublet, combining the semantic relatedness *SemRel* value and the relationship *Relation* between t_1 and t_2. Formally, it is denoted as:

$$TR(t_1, t_2) = \langle SemRel(t_1, t_2), Relation(t_1, t_2) \rangle \quad (6)$$

Algorithm 1: ***TR Algorithm***	Line
Input: t_1, t_2: String // two input texts	1
Variable: V_1: vector // vector for t_1	
V_2: vector //vector for t_2	
CS_1: Set //concept set of t_1	
CS_2: Set // concept set of t_2	
Output: *SemRel*: Double//relatedness value between t_1,t_2	
Rel: string //topological relationships between t_1, t_2	
$CS_1 = f(t_1)$	8
$CS_2 = f(t_2)$	
$C = CS_1 \cup CS_2$	
$V_2 = V_1 = Vector\_Space\_Generator\ (C)$ // generate vector space having C as concepts	11
For each C_i in C	
$V_1[C_i] = w_i$	
$V_2[C_i] = w_i$	
Next	
$SemRel = Vector\_Similarity\_Measure(V_1, V_2)$	16
$Rel = Relation(V_1, V_2)$	
Return <*SemRel*, *Rel*>	18

The algorithm accepts two texts t_1 and t_2 as input (line 1) and corresponding concept sets CS_1 and CS_2 is generated using a function f (lines 8 – 9) representing Natural Language Processing (NLP) or mapping process and returning the concept sets of text. In lines 11 – 15, texts t_1 and t_2 are represented as a vector V (V_1 and V_2 respectively) with weights underlining concept

existence, and inclusion in both CS_1 and CS_2 (following Definition 10). In line 16, the semantic relatedness between two texts is quantified using a measure of similarity between vectors. In this study, we use the *cosine measure:*

$$SemRel(t_1, t_2) = cos(V_1, V_2) = \frac{V_1 \bullet V_2}{|V_1| \times |V_2|} \in [0,1] \quad (7)$$

Semantic relatedness is consequently exploited in identifying basic relations (i.e., *disjointness, intersection* and equality) between texts. Our method for identifying basic relationships is based on a *fuzzy logic* model to overcome the often imprecise descriptions of texts. For instance, texts (likewise RSS items) that describe the same issue are seldom exactly identical. They might contain some different concepts, detailing certain specific aspects of the information being described, despite having the same overall meaning and information substance (cf. Section 1, Example 2). Thus, we address the fuzzy nature of textual content in identifying relations by providing pre-defined/pre-computed similarity thresholds $T_{Disjointness}$ and $T_{Equality}$, as shown in Figure 5.

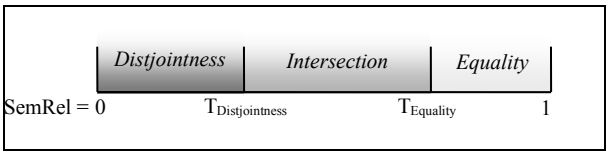


Fig. 5. Basic text relationships and corresponding thresholds.

Thus, we identify the relationships between two texts t_1 and t_2 as follows:

- ***Relation*(t_1, t_2) = *Disjointness*,** i.e., $t_1 \rhd\lhd t_2$, if there is a minimum relatedness between t_1 and t_2 i.e., $SemRel(t_1, t_2) \leq T_{Disjointness}$.
- ***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., $T_{Disjointness} < SemRel(t_1, t_2) < T_{Equality}$.
- ***Relation*(t_1, t_2) = *Equality***, i.e., $t_1 = t_2$, if t_1 and t_2 share a maximum amount of relatedness, i.e., $SemRel(t_1, t_2) \geq T_{Equality}$.

While the basic *disjointness*, *intersection* and *equality* relations can be defined based on semantic relatedness (in the context of fuzzy relations), this is not the case for more intricate relations such as *inclusion* and *oppositeness* that we defined as follows:

- ***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) = *Oppositeness***, i.e., $t_1 \Omega\, t_2$, if they intersect ($t_1 \cap t_2$) having at least one concept C_i of CS_1 included in the antonym neighborhood of a concept C_k in CS_2 (i.e., a concept of CS_2 included in the antonym neighborhood of a concept in CS_1), and such as neither CS_1 nor CS_2 encompass themselves concepts that are antonym to C_i and C_k respectively (we call this last condition *inner antonymy*), considering the antonym neighborhood.

Example 5. Considering Example 2, (t_1 of *CNN2* and t_2 of *BBC2*), and thresholds $T_{Disjointness}$=0.1 and $T_{Equality}$=0.9, $SemRel(t_1, t_2)$ = 0.86 and $Relation(t_1, t_2)$ = *Intersection*. Hence, TR(t_1 of *CNN2* and t_2 of *BBC2)* = *<0.86, Intersection>*.

Example 6. Considering Example 3, (t_1 of *CNN3* and t_3 of *BBC3*), and thresholds $T_{Disjointness}$=0.1 and $T_{Equality}$=0.9, $SemRel(t_1, t_2)$ = 0.612 and $t_1 \cap t_2$ (intersection) and as 'Call' and 'Cancel call' are related with antonymy. $Relation(t_1, t_2)$ = *Oppositeness.* Hence TR(t_1 of *CNN3* and t_3 of *BBC3)* =

<0.86, Oppositeness>.

5 RSS RELATEDNESS AND RELATIONS

This section details the measures used for simple and complex element relatedness.

5.1 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 simplifies to basic pieces of text (cf. Definition 2). The relatedness between two simple elements is computed using Algorithm 2. It accepts two elements e_1 and e_2 as input (line 1) 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. In lines 6 – 7, label and value relatedness are computed respectively using the *TR* algorithm. In line 8, the method E_{SemRel} quantifies the relatedness value between elements, as *weighted sum* value of label and value relatedness such as:

$$SemRel(e_1, e_2) = w_{Label} \times LB_{Sem\,Rel} + w_{Value} \times VR_{Sem\,Rel} \tag{8}$$

where $w_{Label} + w_{Value} = 1$ and $(w_{Label}, w_{Value}) \geq 0$.
Note that several methods for combining label and value relatedness results could have been used, among which the maximum, minimum, average and weighted sum functions. Nonetheless, this latter provides flexibility in performing the match operation, adapting the process w.r.t. the user's perception of element relatedness. The w_{Label} and w_{Value} are computed automatically using the depth/level of concepts in the label knowledge base. Formally:

$$w_{Label}(e_1, e_2) = \begin{cases} 0.5 & \text{if } e_1.\eta \equiv e_2.\eta \\ \dfrac{1}{1 + \max(e_1.\eta.d, e_2.\eta.d)} & \text{otherwise} \end{cases} \tag{9}$$

where $e_i.\eta.d$ represents the depth of the label of e_i and $w_{Value} = 1 - w_{Label}$.

Algorithm 2: ER Algorithm	**Line**
Input: e_1, e_2: element // two simple elements	1
Variable: LB_{SemRel}, VR_{SemRel} : Double // label and value semantic relatedness values $LB_{Relation}$, $TR_{Relation}$: String // Label and value relationship values	
Output: *SemRel*: Double // relatedness value between e_1 and e_2 *Relation*: String // relationship value between e_1 and e_2	
$<LB_{SemRel}, LB_{Relation}> = TR(e_1.\eta, e_2.\eta)$	6
$<VR_{SemRel}, VR_{Relation}> = TR(e_1.\varsigma, e_2.\varsigma)$	
$SemRel = E_{semRel}(LB_{SemRel}, VR_{SemRel})$	8
$Relation = E_{Relation}(LB_{Relation}, VR_{Relation})$	
Return < *SemRel, Relation*>	10

The rule-based method $E_{Relation}$ in line 9 is used for combining label and value relationships as follows:

- Elements e_1 and e_2 are *disjoint* if either their labels or 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 their labels or values intersect.
- Two elements e_1 and e_2 are *equal* if both their labels and values are equal.
- Two elements e_1 and e_2 are *opposite* if both their texts are opposite. RSS label oppositeness is not relevant in identifying element oppositeness, especially w.r.t. RSS merging (cf. Example 4 and Figure 4b).

Having identified the semantic relatedness and relationships between simple elements, Algorithm 3 evaluates RSS item relatedness and relationships. 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 relationship 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 3: IR Algorithm	**Line**
Input: I_1, I_2: element // two input 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
$Relation = I_{Relation}(\{Eij_{Relation\_set}\})$ // $\forall i \in [1,\|I_1\|]$, $\forall j \in [1,\|I_2\|]$	
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 $\{e_j\}$. $Relation(I_1, I_2) = Equality$ if $i = j$ AND $\forall e_2 \in \{e_j\}$, $\exists e_1 \in \{e_i\}$ / $Relation(e_1, e_2) = Equality$.
- Two items I_1 and I_2 are *opposite* if at least two of their respective elements are opposite.

Example 6. Let us consider RSS items *CNN2* and *BBC2* (Figures 1 and 2). Corresponding item relatedness is computed as follows. Notice that different weighting factors are used for label and text values based on the level of the concepts in the knowledge base while computing simple

element relatedness (c.f. 15). Thresholds $T_{Disjointness}$=0.1 and $T_{Equality}$=0.9 is used in getting the relation value. Below, simple element relatedness values and relationship values are given.

ER	$title_{BBC2}$	$guide_{BBC2}$	$link_{BBC2}$	$description_{BBC2}$
$title_{CNN2}$	0.864, x	0.165, x	0.165, x	0.551, x
$guide_{CNN2}$	0.288, x	0.5, x	0.33, x	0.247, x
$link_{CNN2}$	0.288, x	0.247, x	0.5, x	0.247, x
$description_{CNN2}$	0.555, x	0.799, x	0.368, x	0.799, x

where x represents the intersection relationship existing between elements. Using (line 13) *SemRel*(*CNN2*, *BBC2*) = (0.864+ 0.165+ 0.165+ 0.551+ 0.288+ 0.50 + 0.33 + 0.247+ 0. 288+ 0. 247+ 0.50 + 0. 247+ 0. 555+ 0.799+ 0.368+ 0.799) / 4 × 4 = 0.407, where $|I_1|$ and $|I_2|$ are equal to 4. *Relation*(*CNN2*, *BBC2*) = *Intersection* since a number of their elements intersect, i.e., *Relation*($title_{CNN2}$, $title_{BBC2}$) = *Relation*($description_{CNN2}$, $description_{BBC2}$) = *Intersection*.

5.2 Computational Complexity

The computational complexity of our relatedness algorithm is estimated on the basis of the worst case scenario. Suppose I_1 and I_2 are two items (elements), and n_{e_1}, n_{e_2} be number of sub-elements, t_1 and t_2 be the corresponding content of sub-elements, n and m represents the number of concept sets in the vector spaces of V_1 and V_2, d is the depth of the knowledge base, and n_c is the number of words/concepts in synset. Item/XML document relatedness is computed with polynomial time complexity of $O(n \times m \times d \times n_c)$.

Text relatedness (*TR*) is computed with time complexity dependent on complexity of (*i*) building the vector space – that depends on the size of the knowledge base and of the concept set, i.e., $O(n \times m \times n_c \times d + m + n)$, (*ii*) detecting the relationships done with $O(n + m + d)$ which is time needed to detect inclusion. Hence $O(TR) = (n \times m \times n_c \times d)$ The complexity of simple element relatedness is dependent mainly on $O(TR)$.

The complexity of a complex element (item) relatedness is dependent on the number of sub-elements and on simple element relatedness. Hence $O(n_{e_1} \times n_{e_2} \times O(\mathrm{TR}))$, i.e., $O(n_{e_1} \times n_{e_2} \times n \times m \times n_c \times d)$.

Subsequently, identifying relatedness between elements is done with time complexity of $O(n \times m \times n_C \times d)$ as the number of elements n_{e_1} and n_{e_2} in RSS in particular and XML in general is fixed or less determinant compared to number of key terms (n and m), and knowledge based information n_c and d.

6 EXPERIMENTS

To validate our approach, we have implemented a C# prototype entitled *R^3 (RSS Relatedness and Relationships)* encompassing:

- A KB component: stores reference text value and label value knowledge bases via MySql DBMS. The value knowledge base can be modified based on the application considered.
- RSS Input component: allows users to register existing RSS news addresses and also accepts parameters to be used in generating synthetic news.

- Containers for generated and/or extracted news.

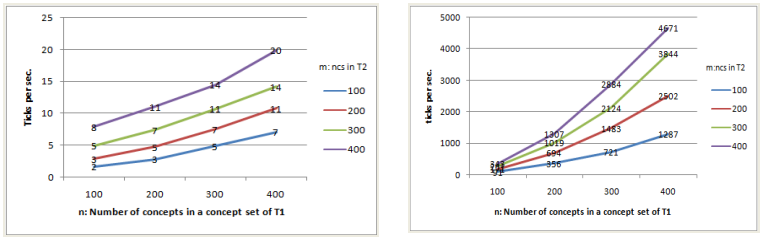
The prototype accepts RSS news items and Boolean flag determining usage of semantic information or not. It measures relatedness between news items automatically after (*i*) stemming text values using Porters' algorithm [13], (*ii*) generating vectors for each text, (*iii*) computing relatedness and relationship in different level of granularity, i.e., text, label, simple element, and item.

We have conducted a set of experiments. The aim of the experimentation is to conform (*a*) the computational complexity and efficiency, (*b*) the relevance of our relatedness measure, and (c) the relevance of topological relationships. All the experiments were carried out on Intel Core Centrino Duo Processor machine (with processing speed of 1.73.0 GHz, 1GB of RAM).

6.1 Timing Analysis and Efficiency

To demonstrate the polynomiality of our approach (as shown in Section 4.3), we experimentally tested the complexity of our relatedness algorithm (relationship computing is not included here as its impact is minimal on timing) following the sizes of two input texts t_1 and t_2 (n and m), and value knowledge base information (n_c and d).

On one hand, we can quickly observe the polynomial nature of the timing result shown in Figure 6 demonstrating experimentally the polynomial dependency of the complexity on input size and knowledge base information. The x axis represents the number of concepts in a concept set and the y axis shows the consumed number of ticks per second in order to get relatedness value.



a. Without semantic knowledgebase b. With fixed semantic (d=8,nc = 100)

Fig. 6. Timing analysis text concept set in t_1, t_2 (n, m)

In Figure 6, we also show the effect of varying number of concepts in concept sets. Figure 6a shows the timing result without considering knowledge base information while varying the size of the input texts. Increasing the number of concept increases the timing in a liner fashion. Figure 6b represents timing result considering fixed semantic information (knowledge base having 100 concepts within a depth of 8). The time needed to compute the relatedness between items increases drastically (compared to the result shown in Figure 6a) and in quadratic fashion.

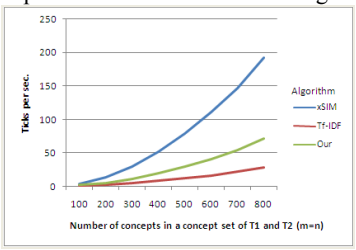


Fig. 7. Timing result obtained using three algorithms: *xSim*, TF-IDF and our algorithm

On the other hand, we wanted to compare the efficiency of our algorithm with similar

existing ones. This is why *xSim* [8] and TF-IDF were selected for this comparison. In all algorithms, relatedness computing between randomly generated synthetic news is done without semantics (as both *xSim* and TF-IDF do not consider semantics information). Figure 7 shows that our approach provides better result than *xSim* but worst than TF-IDF. However, this is due to the fact that TD-IDF does not consider the structure of the RSS news item.

6.2 Relevance of measure

In this set of tests, we used clustering to measure the relevance of our approach by putting together related/similar news. Checking the clustering quality involves (*i*) computation of metrics on pre-defined knowledge of which document belongs to which clusters (*ii*) mapping the discovered clusters to original clusters. Then, we used the popular information retrieval metrics *precision* (PR) and *recall* (R) [11] to check the relevance of the discovered clusters. In addition, an *f-score* value is used to compare the accuracy of different clustering results based on the combined values of PR and R, computed as:

$$f\text{-score} = \frac{2 \times \mathrm{PR} \times \mathrm{R}}{(\mathrm{PR} + \mathrm{R})} \tag{10}$$

To achieve this, we adapted classical clustering approaches [6] and proposed a *relationship-aware*[12] *level based*[13] *single* clustering algorithm (not detailed in this paper due to space limitation).

Using our clustering algorithm, we compared (*i*) our semantic relatedness algorithm, (*ii*) TF-IDF and (*iii*) *xSim* on both real and synthetic dataset, with and/or without semantic information, calculating PR, R, and *f*-score values. Precision and recall graphs exhibit two basic properties independent of the similarity measure used: (*i*) the precision around the clustering level of 1 is maximum (i.e. PR = 1 and the clusters are smaller, and disjoint) whereas recall value is very low (it means that there are lot of mis-matching clusters), (*ii*) precision around the level of 0 is very low (results in bigger and bigger clusters) whereas recall value is higher as mis-clustering is lower. Hence, actual clustering of dataset should end before clustering level is nearer to zero.

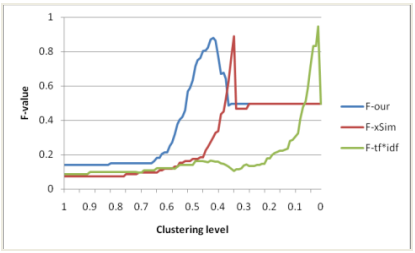


Fig. 8. *f*-score on real data set

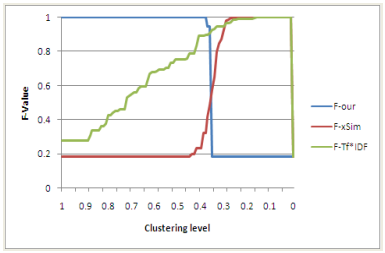


Fig. 9. *f*-score on synthetic dataset

Two data sets were used to conduct our experiments:

- Real data set: we used 158 RSS news items extracted from the known news providers (CNN, BBC, USAToday, L.A. Times and Reuters) clustered manually into 6 clusters:

[12] Classical clustering algorithms, do not consider the relationship between RSS news items, so they may produce clusters having highly related members with lots of intersections which are less relevant during merging. As a result items related with inclusion and having less relatedness value will belongs to different clusters.

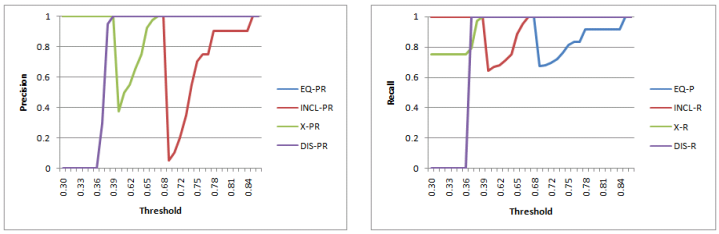
[13] The algorithm generates clusters by varying clustering level between 1 and 0. The difference between two clustering levels is fixed (in our tests, it is 0.1).

US Presidential elections 08, *Middle-east*, *Mumbai-attacks*, *space-technology*, *oil*, and *football*. Figure 8 shows the *f-score* resulting graph. Our semantic relatedness measure provides relevant clustering result for levels between 1 and 0.37 compared to *xSim* and TF-IDF. It generates single cluster at level of 0.35 while *xSim* does at 0.28 and TF-IDF at level of 0.

- Synthetic data set: we generated 100 synthetic RSS news items using our own random RSS item generator. The generated news belongs to 10 disjoint clusters. Each cluster has 10 members and 9 of them have inclusion relationship. Figure 9 shows the *f*-score graph. Clustering using our measure provides the maximum score (*f*-score =1) for levels between 1 and 0.4. This is due to the topological relationships between items that are incorporated at clustering level of 1 (using our proposed *relationship* aware clustering algorithm) and the bigger cluster is generated about level 0.3.

6.3 Relevance of relationship

In this set of tests, we aimed to study to which extent our measures identify the equality, inclusion, intersection or disjointness relationships between items. We generated 100 synthetic news items with various different distributions.



a. Precision graph **b. Recall graph**

Fig. 10. Relevance of relationships on synthetic data

Figure 10 shows the Recall and Precision graph generated on on distribution having (20 equal, 20 include, 40 intersection, and 20 disjoint news) by varying similarity threshold. The graph shows that our measure identifies equality and inclusion relationships all the time. However, the measure misclassifies disjoint news and considers them as intersected due to element label relatedness (without threshold and/or with $T_{Disjointness}$ less than 0.30). Using $T_{Equality}$ of 0.39 allows identifying all disjoint news items and hence provides optimal recall and precision). However, intersection relation recall is lower as the news are considered as equal. A falling down of the precision value for intersection relationship (*x*-PR) around a threshold of 0.6 is observed as the news are considered as equal using equality threshold between 0.61 and 0.68. Similarly for equality threshold between 0.68 and 0.84 where included and intersecting news are considered as equal.

We can conclude here that a correlation can be identified between the threshold values and the distribution of news relationships. This can be inferred using learning and mining techniques. This issue needs to be studied further in the future.

7 CONCLUSIONS and PERSPECTIVES

In this paper, we have addressed the issue of measuring relatedness between RSS items. 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 validating the complexity of our relatedness measure.

The resulting *f*-score value computed on both real and synthetic data shows that our measure generates relevant clusters compared to *xSim* and TF-IDF. In addition, we have shown the capability of our measure in identifying relationships between items. Several future directions will be considered. First, we plan to use our approach so to merge RSS items. Later on, we will extend our work to address XML documents in multimedia scenarios (SVG, MPEG-7, etc.).

8 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] 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.

[5] 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, 15-19 de Outubro, Joao Pessoa, Para ba, Brasil, Anais, Proceedings, pages 68-82, 2007.

[6] 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.

[7] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. Applied Statistics, 28(1):100-108, 1979.

[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] WWW Consortium. The Document Object Model, http://www.w3.org/DOM.
[20] Z. Wu and M. Palmer. 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.