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, marco.viviani, rchbeir, kokou}@u-bourgogne.fr

Abstract. Merging related RSS news (coming from one or different sources) is beneficial for end-users with different backgrounds (journalists, economists, etc.), particularly those accessing similar information. In this paper, we provide a practical approach to both: measure the relatedness, and identify relationships between RSS elements. Our approach is based on the concepts of semantic neighborhood and vector space model, and considers the content and structure of RSS news items.

Keywords: RSS Relatedness, Similarity, Relationships, Neighbourhood

1. Introduction

Really Simple Syndication (RSS) [16] is an XML-based family of web feed formats, proposed to facilitate the aggregation of information from multiple web sources. Merging related RSS news items would allow clients to efficiently access content originating from different providers, rather than roaming a set of news providers and often accessing related or identical news more than once (as existing RSS aggregators[1] do not provide facilities for identifying and handling such items).

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:

- A news might be totally included in another news (*inclusion*).
  Example 1. The title content of *CNN1* "U.N. chief launches $613M Gaza aid appeal" includes the title content of *BBC1* "UN launches $613m appeal for Gaza"[3] (cf. Figures 1 and 2).
- 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 …." and description content of *BBC2* "US carmaker Ford reports the biggest full-year loss in its history ...." are related and very similar, they share some words/expressions ('Ford', 'report', 'loss', 'US') and semantically related concepts ('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

[1] Newsgator, google-reader, etc., allows 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

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

'call' and 'cancel call'.

<item><title>U.N. chief launches $613M 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*
<description>The international youth forum cancels the call for stop-war demonstration due to security reason</description></item>	*CNN3*

Fig. 1. RSS news extracted from CNN

<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*

Fig. 2. RSS news extracted from BBC

Identifying these relationships is beneficial while defining merging rules and making merging decisions. For instance, (i) merging identical or including news might be reduced to keeping the including news (keeping $title_{cnn1}$ in example 1), (ii) merging intersecting news might refer to keeping the common parts and adding the differences ($title_{cnn2}$ and $title_{bbc2}$) (iii) merging opposite news might be done by keeping both news (i.e. merging *CNN3* and *BBC3*). 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 identify (*i*) RSS items that are related enough and (*ii*) the relationship that can occur between two RSS news 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 works. Section 3 defines basic concepts to be used in our measure. In Section 4, we detail how the relatedness and relationships between text values are computed. Section 5 details our RSS relatedness and relationship measures. Finally, Section 6 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 such as merging [9]. A lot of research has been done to determine XML document similarity, which we roughly categorize into *structure-based*, *content* and *hybrid* approaches. Most *structure-based* similarity approaches use tree edit distance [1]. Chawathe [3], Nireman and Jagadish [13] consider the minimum number of edit operations: insert node (insert tree), delete node (delete tree) and update node operations to transform one XML tree into another. Also, the use of Fast Fourier Transform [4] has been proposed to compute similarity between XML documents. With *content based* XML similarity measures [5], similarity is computed based on element contents, disregarding (to a certain extent) the document tags and structural information. Semantic similarity between concepts is estimated either by the distance between nodes [18] or the content of the most specific common ancestor of those nodes involved in the comparison [11] and is evaluated according to some predefined knowledge base(s). In Information Retrieval (IR) [12], 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 [7], the authors used a Vector Space having TF-IDF as a weight factor in XML retrieval. More recently, there are *hybrid* approaches that attempted to address XML comparison. In a recent work [17], the authors combine an IR semantic similarity technique with a structural-based algorithm based on tree edit distance. However, semantic similarity evaluation is limited only to tag names. In [8], *xSim* computes the matching between XML documents as an average of the elements' matching similarity value. Similarity between two elements is computed as the average of textual content, element name and path (i.e. sequence of node names starting from the root) 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. In [10], Ho-Lam *et al*. stress on the importance of considering relationships (equality, overlap, disjointness and containment or inclusion) between data sources while merging XML documents, without however addressing the issue. Ian Gracia *et al*. [5] used a correlation based approach (applied only to content) to identify relationship among RSS news articles: redundant (equal and inclusion), non-related (disjoint) and related (intersection) relationships. In paper work, we identify equality, inclusion, disjoint, intersection and opposite relationships in addition to measuring relatedness while considering tag name similarity.

3. PRELIMINARIES

An RSS[4] document is a well-formed XML document represented as a rooted ordered labeled tree following the Document Object Model (DOM) [19]. Each node or element of an RSS Tree is a pair having $e = \langle \eta, \varsigma \rangle$ where $e.\eta$ is the element name and $e.\eta$ its content. The content of an element can be another element (complex element) or a text value (simple element). Notice that, different versions of RSS items consistently follow the same overall structure with minor additions and removals (e.g. source is part of RSS 0.9x and guid in RSS 2.0).

3.1. Knowledge Base

A *Knowledge Base* [15] (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 assist relatedness and is formally defined as $KB = (C, E, R, f)$ where C 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, $R = \{\equiv, \prec, \succ, <<, >>, \Omega\}$ [5], f is a function designating the nature of edges in E, $f : E \to R$.

We introduced two types of knowledge bases to assist simple element relatedness: (*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 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).

---

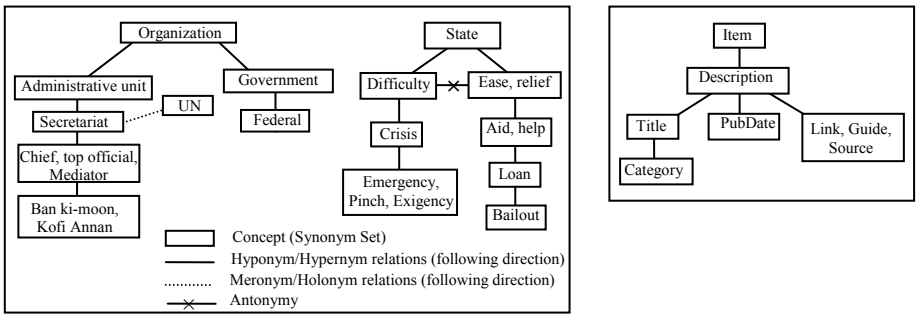
[4] RSS refers to one of the following standards: Rich Site Summary (RSS 0.91, RSS 0.92), RDF Site Summary (RSS 0.9 and 1.0), and Really Simple Syndication (RSS 2.0).

[5] The symbols in R underline 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].

3.2. 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 [6], is exploited in identifying the relationships between text (i.e., RSS element labels and/or textual contents) and consequently RSS elements/items.

Definition 1 [Semantic Neighborhood]: The *semantic neighborhood* of a concept C_i ($N^R_{KB}(C_i)$) 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.



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 2 [Global Semantic Neighborhood]: The *global semantic neighborhood* of a concept C_i ($\overline{N_{KB}}(C_i)$) is the union of each semantic neighborhood w.r.t. all synonymy ($\equiv$), hyponymy ($\prec$) and meronymy ($<<$) relations altogether.

Definition 3 [Antonym Neighborhood]: The antonym neighborhood of a concept C_i ($N^{\Omega}_{KB}(C_i)$ **)** 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 ($\succ$).

4. TEXT RELATEDNESS

4.1. Text Representation

A text is represented following the vector space model used in information retrieval [12]. Each text t_i is represented in an n-dimensional vector space such as: $[\langle C_1, w_1\rangle, \ldots, \langle C_n, w_n\rangle]$ where C_i is a *concept set*, w_i is its weight and n is the number of distinct concepts in both texts. The *concept set* of text t, denoted as *CS*, is a set of concepts $\{C_1, \ldots, C_m\}$, where each C_i (obtained after several textual pre-processing operations such as stop-words[6] removal, stemming[7], and mapping to knowledge base**)** represents the meaning of a group of terms in $\{k_1, \ldots, k_n\}$, where m is the total number of concepts describing t. The weight of a concept C_i reflects its existence in vector V_i (1 if it exists) or the maximum enclosure similarity it has with a concept C_j in V_j.

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

[6]Stop-words identify words/expressions which are filtered out (e.g., *yet, an, but, the, …*)

[7]Stemming is the process for reducing inflected (or sometimes derived) words to their stem, i.e., base.

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. Consider the *description* element of RSS items *CNN2* and *BBC2* (Figures 1, 2). The partial corresponding vector representations V_1 and V_2 are shown in Figure 4.

	Ford	*report*	*loss*	*...*	*Federal*	*Bailout*	*Big*	*say*	*government*	*loan*
V_1	1	1	1	…	1	1	**1**	**1**	**1**	**1**
V_2	1	1	1	…	**0.67**	**0.86**	1	1	1	1

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

For each concept C in V_1 and V_2 its weight is 1 if it exists, otherwise it is updated with the maximum semantic enclosure similarity value. For instance, following the WordNet 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, *Enclosure_sim*(federal, government) = 1 but in V_2, *Enclosure_sim*(government , federal) = 0.67. Likewise, 'loan' is included in the global semantic neighborhood of 'bailout' i.e. *Enclosure_sim*(loan, bailout) = 1 and *Enclosure_sim*(bailout, loan) = 0.86.

4.2. Text Relatedness and Relations

Given two texts t_1 and t_2, the *Textual Relatedness* (*TR*) algorithm returns a pair containing the semantic relatedness *SemRel* value and the relationship between the texts being compared. The *SemRel* value is computed using a vector based similarity method (e.g. cosine [12]) after building the corresponding text vector space. *SemRel* 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 using pre-defined/pre-computed similarity thresholds $T_{Disjointness}$ and $T_{Equality}$, as shown in Figure 5.

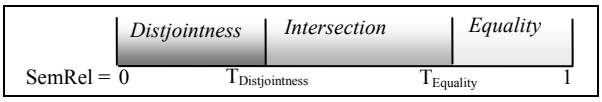


Fig. 5. Basic text relationships and corresponding thresholds.

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., $SemRel(t_1, t_2) \geq T_{Equality}$.

More intricate relations such as *inclusion* and *oppositeness* are identified 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 or vice-versa, 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* as no concept of t1 is included in antonymy neighborhood of concept of t2 and vice-versa. Hence, $TR(t_1,t_2)$ = <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 $TR(t_1,t_3)$ = <0.86, *Oppositeness*>.

5. RSS RELATEDNESS AND RELATIONS

Given two simple elements e_1, and e_2, the Element Relatedness (*ER*) algorithm returns a pair quantifying the semantic relatedness *SemRel* value and *Relation* based on corresponding *TR* label and content values. *SemRel* quantifies the relatedness value between elements, as the *weighted sum* value of label and value relatedness such as:

$$SemRel(e_1,e_2) = w_{Label} \times LB_{SemRel} + w_{Value} \times VR_{SemRel} \quad (2)$$

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

Relation exploits a rule-based method 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 contents are opposite. RSS label oppositeness is not relevant in identifying element oppositeness, especially w.r.t. RSS merging (cf. Example 3 and Figure 3b).

Given two RSS items I_1 and I_2, each made of a bunch of elements, the Item Relatedness (IR) Algorithm returns a pair containing *SemRel* and *Relation*. The *SemRel* is computed as the average of the relatedness values between corresponding element sets I_1 and I_2 as:

$$SemRel(i_1,i_2) = \frac{SemRel(e_i,e_j)}{|i_1| \times |i_2|} \quad \forall e_i \in I_1, \forall e_j \in I_2 \quad (3)$$

The *Relation* between I_1 and I_2 is identified by combining sub-element relationships using a rule-based method 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 $\{e_j\}$.
- Two items I_1 and I_2 are *opposite* if at least two of their respective elements are opposite.

Example 7. Let us consider RSS items *CNN2* and *BBC2* (Figures 1 and 2). Corresponding item relatedness is computed as follows. Notice that weight value of w_{label} = 0.5 and w_{value} = 0.5 and Thresholds $T_{Disjointness}$ = 0.2 and $T_{Equality}$ = 0.8 are used. Below, simple element relatedness values and relationship values are given.

ER	$title_{BBC2}$	$description_{BBC2}$
$title_{CNN2}$	<0.864, Equal>	<0.551, Intersect>
$description_{CNN2}$	<0.555, Intersect>	<0.799, Intersect>

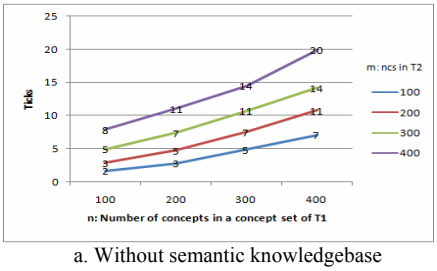
Using (c.f. 3) *SemRel*(*CNN2*, *BBC2*) = (0.864+ 0.551+ 0. 555+ 0.799) / 2 × 2 = 0.692, where $|I_1|$ and $|I_2|$ are equal to 2. *Relation*(*CNN2*, *BBC2*) = *Intersection* since a number of their elements intersect, i.e., *Relation*($title_{CNN2}$, $title_{BBC2}$) = *Relation*($description_{CNN2}$, $description_{BBC2}$) = *Intersection*.

6. EXPERIMENTS

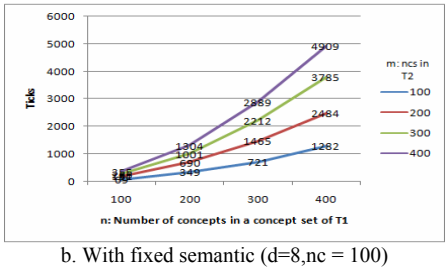
We have conducted a set of experiments in order to conform (*a*) the computational complexity and (*b*) the relevance of our relatedness measure. All the experiments were carried out on Intel Core Centrino Duo Processor machine (with processing speed of 1.73.0 GHz, 1GB of RAM).

6.1. Timing Analysis

Computational complexity of our item relatedness algorithm is polynomial and depends on the number of concept sets in input texts t_1 and t_2 (i.e. n and m), and the value knowledge base size (n_c - number of concepts and d - *depth*). Figure 6.a shows the timing result without considering knowledge base information while varying the size of the input texts. Timing increases in a liner fashion w.r.t. the number of concepts. Figure 6.b presents timing result considering fixed knowledge information (100 concepts within a depth of 8). The time needed to compute the relatedness between items increases drastically and in a quadratic fashion.



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)

6.2. Relevance of measure

In this set of tests, we used our *relationship-aware*[8] *level based single* link clustering algorithm (adapted from classical clustering approaches [7]; not detailed in this paper due to space limitation). The single link clusters at a clustering level l_i (between 1 and 0) produces all items I_i with pair-wise similarity values greater than or equal to l_i. Notice that clustering at level near to zero produces very big and less relevant grouping hence should be avoided. We used the popular information retrieval metrics *precision* (PR) and *recall* (R) [12] and an *f-score* value to check the quality of discovered cluster.

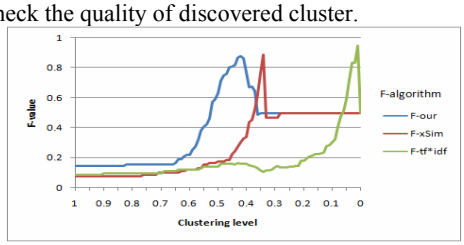


Fig. 7. *f*-score on real data set

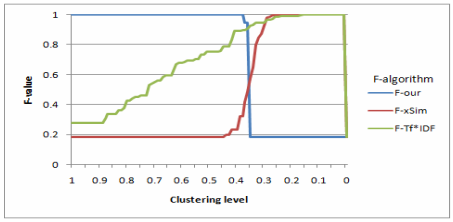


Fig. 8. *f*-score on synthetic dataset

Two data sets were used to conduct our experiments:

- Real data set: we used 158 RSS news items extracted from CNN, BBC, USAToday, L.A. Times and Reuters, clustered manually into 6 groups: *US Presidential elections 08*, *Middle-east*, *Mumbai-attacks*, *space-technology*, *oil*, and *football*. Figure 7 shows the *f-score*

[8] Classical clustering algorithms, do not consider the relationship between 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 values will belong to different clusters.

resulting graph. The average *f-value* computed over the entire clustering level conforms that our semantic relatedness measure provides relevant clustering results (clusters closer to the predefined ones, particularly between levels 1 and 0.37) compared to *xSim* and TF-IDF.

- Synthetic data set: we generated 100 synthetic RSS news items using our own random RSS item generator. The generated news belong to 10 disjoint clusters. Each cluster has 10 members and 9 of them are related via the inclusion relationship. Figure 8 shows the *f*-score graph corresponding to our clustering experiments. Our relationship aware clustering algorithm groups all items related with inclusion in the appropriate cluster (between clustering levels 1 and 0.4, achieving a maximum *f*-score=1), whereas *xSim* and TF-IDF generate incomplete clusters, disregarding the inclusion relationship.

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 RSS item relatedness computation, taking into account the different kinds of relationships that can occur among texts, elements and items. We have developed a prototype validating the complexity and effectiveness of our relatedness measure. Clustering experiments, conducted on both real and synthetic data show that our measure generates more relevant clusters compared to *xSim* and TF-IDF. In addition, we have shown the capability of our measure in identifying relationships between items. We are currently working on RSS merging, developing relation-aware merging rules. The next step would be extending our work to address XML multimedia merging (SVG, MPEG-7, etc.).

REFERENCES

[1] P. Bille. A survey on tree edit distance and related problems. *Theoretical CS*, 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 1999: 90-101, San Francisco, CA, USA. 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.
[5] I. Garcia, Y. Ng Eliminating Redundant and Less-Informative RSS News Articles Based on Word Similarity and a Fuzzy Equivalence Relation. ICTAI 2006: 465-473
[6] F. Getahun, J. Tekli, S. Atnafu, and R. Chbeir. Towards efficient horizontal multimedia database fragmentation using semantic-based predicates implication. SBBD 2007: 68-82.
[7] T. Grabs and H.-J. Schek. Generating Vector Spaces On-the-fly for Flexible XML Retrieval. ACM SIGIR Workshop on XML and Information Retrieval 2002: 4–13
[8] A. M. Kade, C. A. Heuser, Matching XML documents in highly dynamic applications. ACM symposium on Document engineering 2008: 191-198.
[9] R. La Fontaine. Merging XML files: A new approach providing intelligent merge of XML data sets. In Proceedings of XML, May 2002. Barcelona, Spain
[10] H. Lau, W. Ng. A Unifying Framework for Merging and Evaluating XML Information. DASFAA 2005: 81-94
[11] Lin D., An Information-Theoretic Definition of Similarity. In Proceedings of the 15th International Conference on Machine Learning, 1989: 296-304, Morgan Kaufmann Publishers
[12] M. J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.
[13] A. Nierman, H. V. Jagadish: Evaluating structural similarity in XML documents. WebDB 02: 61-66.
[14] Princeton University Cognitive Science Laboratory. WordNet: a lexical database for the English language. http://wordnet.princeton.edu/.
[15] R. Richardson and A. F. Smeaton. Using WorldNet in a knowledge-based approach to information retrieval. Technical Report CA-0395, Dublin, Ireland, 1995.
[16] RSS Advisory Board. RSS 2.0 Specification. http://www.rssboard.org/.
[17] J. Tekli, R. Chbeir, and K. Yetongnon. A hybrid approach for xml similarity. SOFSEM 07: 783-795.
[18] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, 133-138, Morristown, NJ, USA, 1994.
[19] WWW Consortium. The Document Object Model, http://www.w3.org/DOM.