Operation on Text Entities

Ion IVAN, Daniel MILODIN, Marius POPA
Economic Informatics Department, Academy of Economic Studies, Bucharest

There are defined level of orthogonality for text entities. There are built orthogonal entities. There are identified operations on orthogonal entities and for each operation there are specified the proprieties and the signification from applicability point of view. There is described software use to implement operations with structured entities.
Keywords: *text entity, orthogonality, operation.*

Text entities and structured text entities

There is considerate the A alphabet, where A = {a_1, a_2, …, a_n}, formed with *n* symbols a_1, a_2, ..., a_n. Using the alphabets symbols are built the words c_1, c_2, ..., c_{nA}, where *nA* represents the number of words for V(A) vocabulary, build with the symbols of A alphabet. The words c_i and c_j are considered to be orthogonal if have no common symbol. If it is considered the alphabet A formed with symbols *h, s, x* and *w*, where A = { h, s, x, w }, use to built words:

c_1 = <*hxx*>
c_2 = <*sssw*>
c_3 = <*hxs*>
c_4 = <*hxhx*>

which are forming the vocabulary V(A) = {c_1, c_2,c_3,c_4 }

The orthogonality indicator $H(c_i,c_j)$ for words c_i, c_j, is calculated using the formula:

$$H(c_i,c_j)=1-\frac{\sum_{k=1}^{n}\min(f_{ik},f_{jk})}{\max\{Lg(c_i),Lg(c_j)\}}$$

where:

f_{ik} – shows the number of apparition of a_k symbol from alphabet A in word c_i

f_{jk} – shows the number of apparition of a_k symbol from alphabet A in word c_j

$Lg(c_i)$ – shows the number of characters composing word c_i

n – shows the number of symbols used for build the two words

The length for each word is given in table 1.

Table 1 – Words length

Paragraph	**Lg(P_i)**
c_1	3
c_2	4
c_3	3
c_4	4

$H(c_i,c_j)$ is the orthogonality indicator and measures how different are the words c_i and c_j.

$H(c_i,c_j) \in [0;1]$. If $H(c_i,c_j) = 0$ then c_i and c_j are identically. If $H(c_i,c_j) = 1$ then c_i and c_j are totally different.

By convention, based on experience and validated experiments, if $H(c_i,c_j) < 0.78$ results that the elements c_i and c_j are not orthogonal. If $H(c_i,c_j) \in [0.78;0.92)$ then the elements c_i and c_j are sufficiently different to be distinguish one from another and to choose one of them based on his own performance.

If $H(c_i,c_j) \in [0.92;1]$, it means that c_i and c_j have very few common characteristics or are totally different and it is impossible to be confused.

For the words of the considerate vocabulary, there is build the table 2, on which base will be determined the orthogonality:

Table 2 – The orthogonality of the c_1,c_2,c_3,c_4 words

f()	The alphabet symbols				$\sum_{k=1}^{n}\min(f_{ik},f_{jk})$	$H(c_i,c_j)$
	h	*s*	*x*	*w*		
$f(c_1)$	1	0	2	0	-	-
$f(c_2)$	0	3	0	1	-	-
$f(c_3)$	1	1	1	0	-	-

$f(c_4)$	2	0	2	0	-	-
$f(c_1)*f(c_2)$	0	0	0	0	0	1
$f(c_1)*f(c_3)$	1	0	2	0	2	0.33
$f(c_1)*f(c_4)$	2	0	4	0	3	0.25
$f(c_2)*f(c_3)$	0	3	0	0	1	0.75
$f(c_2)*f(c_4)$	0	0	0	0	0	1
$f(c_3)*(c_4)$	2	0	2	0	2	0.5

After finishing the calculation, results that only words c_1 and c_2, and respectively c_2 and c_4 are orthogonal. Using words from vocabulary, there are built paragraphs P_1, P_2, ..., P_m, which are words separated with a symbol, called separator.

If it is considered the alphabet AA = {x,y,z,u,w,+}, where the symbol + is a separator, which is used to build the vocabulary V(AA) = {xy, xzy, zzyx, uuu, uzx, wuz, wxy, wwz, wyx} and the paragraphs:

P_1 = <*xy+xzy+zzyx*>
P_2 = <*uuu+uzx+wuz*>
P_3 = <*wwz+wyx*>
P_4 = <*xy+wyx+wwz*>
P_5 = <*xy+uuu+wuz+wyx*>.

A paragraph P_i has $Lg(P_i)$ length, given by the number of words which are composing it. The P_1, P_2, P_3, P_4 and P_5 paragraphs' length are given in table 3.

Table 3 – Paragraph's length

Paragraph	**$Lg(P_i)$**
P_1	3
P_2	3
P_3	2
P_4	3
P_5	4

In order to determine the paragraphs' orthogonality it is used the same way as for studying the words' orthogonality, being used the words from vocabulary.

The orthogonality is calculated using the formula:

$$H(P_i, P_j) = 1 - \frac{\sum_{k=1}^{m} \min(P_i, P_j)}{Lg(P_i) + Lg(P_j)}$$

where the numerator shows the number of common words from the two paragraphs.

The final results are given in table 4.

Table 4 – The paragraphs' orthogonality

f()	The alphabets' symbols									$\sum_{k=1}^{n} \min(f_{ik}, f_{jk})$	$H(P_i,P_j)$
	xy	*xzy*	*zzyx*	*uuu*	*uzx*	*wuz*	*wxy*	*wwz*	*wyx*		
$f(P_1)$	1	1	1	0	0	0	0	0	0	-	-
$f(P_2)$	0	0	0	1	1	1	0	0	0	-	-
$f(P_3)$	0	0	0	0	0	0	0	1	1	-	-
$f(P_4)$	1	0	0	0	0	0	0	1	1	-	-
$f(P_5)$	1	0	0	1	0	1	0	0	1	-	-
$f(P_1)*f(P_2)$	0	0	0	0	0	0	0	0	0	0	1
$f(P_1)*f(P_3)$	0	0	0	0	0	0	0	0	0	0	1
$f(P_1)*f(P_4)$	1	0	0	0	0	0	0	0	0	1	0.83
$f(P_1)*f(P_5)$	1	0	0	0	0	0	0	0	0	1	0.85
$f(P_2)*f(P_3)$	0	0	0	0	0	0	0	0	0	0	1
$f(P_2)*(P_4)$	0	0	0	0	0	0	0	0	0	0	1
$f(P_2)*(P_5)$	0	0	0	1	0	1	0	0	0	2	0.71
$f(P_3)*(P_4)$	0	0	0	0	0	0	0	1	1	2	0.6
$f(P_3)*(P_5)$	0	0	0	0	0	0	0	0	1	1	0.83
$f(P_4)*(P_5)$	0	0	0	0	0	0	0	0	1	2	0.71

A chapter *k* is formed with a succession of paragraphs. The chapter *k* orthogonality depends on each paragraphs orthogonality. It is defined the *k* chapter orthogonality with the formula: $H(K) = \frac{A}{B}$, where:

A – the number of different paragraphs
B – the total number of paragraphs
The orthogonality is studied composing paragraphs from chapter, two with two, resulting a number of $\frac{n(n-1)}{2}$ paragraphs composing, so the orthogonality of a chapter is given by the formula: $H(K)=\frac{2*A}{n(n-1)}$, where A represents the number of different paragraphs. The orthogonality of an entity as a series of chapter is given by the formula: $H(E)=\frac{\alpha}{\beta}$, where:

α - number of orthogonal chapter; β - number of articles which orthogonality is tested.
Orthogonality must be seeing on levels of approach, for an easier administration of a entity structure. For example, if it is considered an encyclopedia about butterfly, there are incorporated: names; the aria of dispersion; the biologic characteristics; pictures.
There is calculated the orthogonality for all pairs of butterflies, which leads to global orthogonality.

Creating a list of orthogonal entities

Text entities are entities created using word strings that are based certain rules, by which they define a real world context. The rules imposed to the text entities built using word strings are connected with:
- the clear definition of the tackled context, using the terms of the respective context such that the text entities are specialized in the context terminology;
- the identification of the key notions in the respective field;
- the compliance with the existing rules in the respective field and their interpretation in the text entities framework;
- the knowledge of the organization in the studied field, so that the text entities building modality based on the studied terminology is as clear, concise and, especially, easy to read as possible;
- the identification of existing similarities between the subfields of the study field, the establishment of a clear structure when building entities, basing on the link between the subfields.

The entities' structuring is about building a text assembly based on certain relations imposed to the constituent text, so that the new configuration defines, in an organized manner, a subfield.
The difference between text entities and structured text entities derives out of the fact that the latter entities category refers to a certain component of the former category, on which it operates using a series of rules having the role of contributing to a very clear association between its components, in order to create an organizational structure.
Generally, text entities are characterized by length, represented by the number of component words, the degree of detailing of the tackled field, the connection between the text entities constituent elements, the identification of the existing structure within the field framework and its application in the entities' context.
By structuring, entities become more efficient to use, the whole information in a certain field being transferred to the entities by using a pattern. The used pattern contributes to the information uniformity, and also to its arrangement in an organized manner, based on sorting criteria.
The entities' orthogonality measures the degree of differentiation between two entities. Two entities are orthogonal when they are completely different. Two entities are different if the values on similar positions are distinct. Entities are comparable because they respect the same modality of representation and the same pattern of structuring. In other words, before determining the level of orthogonality of two entities, it must be checked whether they are compatible from the structure point of view or not.
In a manner similar to the notion of orthogonal entities, the notion of non-orthogonal entities is defined. Non-orthogonal are called the entities that have identical values on similar positions.
The orthogonality of the text entities is studied from the point of view of the entity itself, and also from the point of view of its being a part of an entities' list, resulting the next forms of entities' orthogonality:

- internal orthogonality;
- external orthogonality.

The internal orthogonality identifies the degree of similitude between the components of an entity. Whenever the entity implements a structural pattern based on texts, are checked the component substrings and the differences and similarities between them.

In picture 1 is being presented the content of the E entity, content that is subject to checking, in order to determine the internal orthogonality of the entity.

Figure 1 – The entity E

Still, whenever is compared the content of two or more entities, is checked their external orthogonality, too. In order to determine the external orthogonality, the entities must be compatible, meaning they must be implemented using the same working structure.

In picture 2 are being identified the positions with similar values, corresponding to entities A and B.

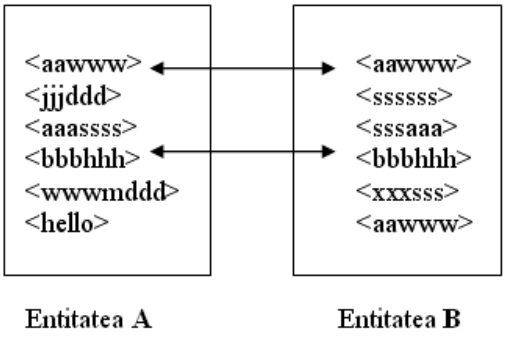

Figure 2 – The Orthogonality of the entities A and B

As it can be noticed in picture 2, even if the value <aawww> is found in the first entity on the first position, and in the second entity both on the first and last position, the value <aawww> corresponds, from the orthogonality point of view, only for the first position.

It considers the entity E_1 as base entity. It builds the entity list $L=\{E_1\}$. The list is completed with the entity if and only if the two entities have acceptable internal orthogonality, greater than θ : $H(E_i) \geq \theta$.

The insertion of a new entity into list supposes the inclusion in the entity list of a new entity E_2 that respects both the belonging to the rest of the entities, and criteria about its arrangement in existing entity list.

Based on the analysis of the entities E_1 and E_2, it results $H(E_1,E_2) > \Sigma$, accepted level in order to include the entity E_2 in the list.

After orthogonality checking of the two entities, the sorting criterion applied in entity list is verified. Depending on this, the entity that respects the orthogonality criterion will be arranged in entity list.

If the list $L = \{E_1, E_2, ..., E_n\}$ contains entities with internal orthogonality greater than θ and $H(E_i,E_j) > \Sigma$, $\forall i,j \in \{1,2,...,n\}$, with $i \neq j$, to add an entity E_{n+1} to the list consists of:

- to verify if $H(E_i,E_{n+1}) > \Sigma$, i=1, 2, ..., n;
- to verify if $H(E_i) \geq \theta$.

The resulted list is an orthogonal construction, having the orthogonality average:

$$H(L)=\left(\prod_{i=1}^{n-1}\prod_{k=i+1}^{n} H(E_i,E_k)\right)^{\frac{2}{n(n-1)}}, \text{ with } n \geq 2.$$

The orthogonal level of the entity list is given by:

$$\min \{H(E_i,E_{i-1})\} \leq H(L) \leq \max \{H(E_i,E_{i+1})\}$$

It considers three entities E_1, E_2, E3 that respect the following:

$$H(E_1) = 0.7 > 0.6$$
$$H(E_2) = 0.8$$
$$H(E_3) = 0.89$$
$$H(E_1,E_2) = 0.6 > 0.5$$
$$H(E_1,E_3) = 0.8 > 0.5$$
$$H(E_2,E_3) = 0.9$$
$$L\{E_1,E_2,E_3\}$$
$$0.6 \leq H(L) \leq 0.9$$

For above example, $H(L) = (H(E_1,E_2)* H(E_1,E_3)* H(E_2,E_3))^{1/3} = (0.6*0.8*0.9)^{1/3} = 0.432^{1/3} = 0.755$

Sorting the entities from entity list

It supposes the existence of a word or word stream called key that define an entity.

For an explicative dictionary, the word that it is the definition object represents the key.

The arrangement of the words in alphabetical order leads to dictionary building.

The sorting of the entities from the list L don't modify the orthogonality level: $H(E_i,E_j) = H(E_j,E_i)$

The entity sorting supposes the identification of criteria for the sorting. The criteria depends of representative field in which the entities act.

For text entities, the sorting criterion is the alphabetic one. An important role is played by text or compared word lengths.

For instance, it considers the words c_1 = {aaabb} and c_2 = {aaab}. Even the words look similar, they differ through the last letter, the difference being made by the length. $Lg(c_1) > Lg(c_2)$ and it results that in the alphabetical sorting process the word c_1 will be placed after the word c_2.

The sorting of two entities supposes that these ones are put in correspondence in order to identify their similitude and difference and to establish their positions in entity list taking into account the criterion.

The sorting operation is made to optimize the access to asked information. The sorting criterion is the one that emphasis the sorting need of the information. The aim of the sorting is to arrange the text information contained by entities such as they respect an certain rule.

The sorting advantages:

- the retrieval time for asked information;
- the work way with the information;
- information storage such as for new values it respects the sorting criterion.

The disadvantages regard the data loading. The data must be compared with the existed data to determine their position in entity.

To extenuate this disadvantage, some algorithms for information sorting in a minimum steps were built:

- counting;
- insertion;
- Shell method;
- Bubble;
- Quicksort;
- selection;
- interchange.

The need for information sorting is given by decreasing of retrieval time of the information. Within organizations, the decisions must be done in a shortest time and taking into account all factors of the information storage efficiently.

Entity concatenation

The concatenation of two entities consists in unifying the content of the entities to obtain a new entity that defines the domain more accurate.

In figure no. 3, the concatenation of two entities is depicted:

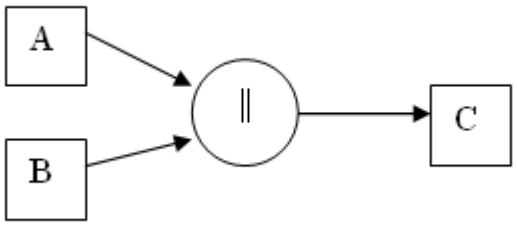

Figure 3 – Concatenation of two entities

The length of the new entity is determined as it follows: $Lg(A||B) = Lg(A)+Lg(B)$, where:

Lg(A||B) –the length of the new entity

Lg(A), Lg(B) – the lengths of the two source entities

In figure no. 4, the length of the new entity is presented:

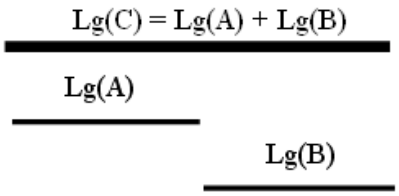

Figure 4 – The length of the new entity

In the same way with sorting operation, the concatenation is made using entities with the same structure.

In order to obtain a new entity, the entity concatenation is made by adding to the end of the first entity the content of the two entity.

The entities E_i and E_j become the entity E_{ij} through concatenation.
$H(E_i,E_j)$ represents the orthogonality between entities. $H(E_i||E_j)$ represents the orthogonality intra-entities.
The concept that helps to study the text entity orthogonality is the frequency. This concept measures the number of appearances of a text in an entity.
It considers the words:

$$C_1 = <xyz>$$
$$C_2 = <uvw>$$
$$H(C_1,C_2) = 1$$
$$C_1||C_2 = <xzyuvw>$$
$$H(C_1||C_2) = 1$$

$H(C_1,C_2)$ represents the orthogonality of the vocabulary and $H(C_1||C_2)$ is the orthogonality of the alphabet.
If

$$c_1 = <xyzu>$$
$$c_2 = <xzyw>$$
$$c_3 = <xzzt>$$
$$c_4 = <xxyq>$$

$$E_1 = <c_1,c_2>$$
$$E_2 = <c_3,c_4>$$

$H(E_1,E_2) = 1$, because the entity E_1 is not included in E_2. Below, it presents the analytical expression that leads to this result.
$H(E_1|||E_2)$ impose to analyze the orthogonality in $E_1|||E_2$ = <xyzu+xzyw+ xzzt+xxyq>, the words being separated by the separator "+".
When the all words that form the new entity have the appearance frequency equalt to 1, then the intra-entities orthogonality is 1, each words being most one time in the new entity. If there is a word that has the frequency greater than , then the orthogonality coefficient of the new entity starts to decrease.
The intra-entities orthogonality is determined using the expression: $H(E_1||E_2)=\frac{nrc}{\sum_{i=1}^{nrc} f_i}$, where:

nrc – the number of words in a concatenated text entity;
f_i – the appearance frequency of each word in concatenated text entity.
In case in which the appearance frequency of each word has the value 1, the orthogonality is also 1, and if the appearance frequency of the words is increasing, then the orthogonality indicator level decreases. There is an inverse proportional relationship between the orthogonality level of the text entities and appearance frequency of the words that form the text entity.
For the considered example, the indicator $H(E_1|||E_2)$ is 1 because the word frequencies are also 1.
The orthogonality between entities is determined using the below formula:

$$H(E_1,E_2)=1-\frac{\sum_{k=1}^{nrc}\min(E_1,E_2)}{Lg(E_1)+Lg(E_2)}$$

where:
min(E1,E2) – the minimum number of common appearances for the common words for the two entities; when the two entities have not any common word then the value of the numerator has the value 0, and the orthogonality indicator is 1;
nrc – the word number used to build the two entities.
The entity concatenation is used from more reasons:
- to combine values contained by more entities with the goal to obtain new entities that will include all these values;
- to have continuity in the presentation of the information;
- to perform the operations of searching, sorting, intersection in an efficient way;
- to group the information on different criteria; thus, the entities approach very much to the concept of class.

An aspect that it must keep in mind when it works with concatenated text entities aims a good information management such as to be respected the conditions regarding the structure and the content.
Through the entity concatenations it is built an efficient way to work with the entities that contain information in the same interest domain.

Conclusions

The structured text entities are a concept that combines the work with the texts in an environment built on the base of a model. The use of the concept contributes to assurance of

a high level of othogonality through similar entity identification. The structured entities can be applied in a large scale of domains, assuring the originality character of the components.

The implemented operations on text entities have utility in a more efficient work with the stored information and improved access to the information in an organization.

Implementation of techniques for structured entity orthogonality study implies firstly a very good knowledge of the domain, and secondly the knowledge of algorithms and techniques for orthogonality study.

The implications of the concept belong to specializing the reference domains with some structures care re-define the notions used at macro level into notions at micro levels implemented with structured entities.

The paper described the operations of building, sorting and concatenation made on text entities, the effects of these ones on the text entity content, and ways to improve the work way with the texts.

The paper will be extended for other operation presentations that are made on structured text entities, and to observe the impact of these ones on orthogonality for the entities resulted from their applying.

Bibliography

[IVMP05], Ion IVAN, Daniel MILODIN, Marius POPA, Cosmin LUGOJI – *Based Entities*, in „Economia – seria Management", year VIII, no. 1, 2005

[POPM05], Marius POPA – *The Text Entity Quality Evaluation – Theory and Practice*, ASE Printing House, Bucharest, 2005

[IVAM05], Ion IVAN, Daniel MILODIN, Marius POPA – *The Alphabet Orthogonality*, in „Revista Română de Informatică şi Automatică", vol. 15, no. 3, 2005

[IVAP05], Ion IVAN, Marius POPA, Cătălin BOJA, Cristian TOMA – *Metrics for Text Entity Similarity*, in „Studii şi Cercetări de Calcul Economic şi Cibernetică Economică", vol. 39, no. 4, 2005, pp. 43 – 57

[POPA05], Marius POPA – *The Data Quality Characteristic System,* ASE, Bucharest, 2005, PhD stage paper

[IVAN04a], Ion IVAN, Marius POPA, Alexandru POPESCU – *The Aggregation of the Text Entities*, în „Economic Computation and Economic Cybernetics Studies and Research", vol. 38, no. 1-4, 2004, pp. 37 – 50

[IVAN04b], Ion IVAN, Marius POPA, Cristian TOMA, Iulian RĂDULESCU – *The Aggregation of the Data Orthogonality Metrics*, Proceedings of „The 35th International Scientific Symposium of METRA", vol. 1, Bucharest, 27th and 28th of May, 2004, pp. 590 – 595

[IVAN04c], Ion IVAN, Marius POPA, Iulian RĂDULESCU – *Techniques and Methods for Aggregation of Structured Texts*, in the 9th Scientific Session of Papers with International Participation „Ştiinţa şi învăţământul – fundamente ale secolului al XXI-lea", Academia Forţelor Terestre Sibiu, 25th and 26th of November, 2004,

[IVAN04d], Ion IVAN, Marius POPA, Roland DRĂGOI – *Validation of the Orthogonality for Companies and Economic Organization Sigles*, in „Studii şi Cercetări de Calcul Economic şi Cibernetică Economică", vol. 38, no. 3, 2004, pp. 17 – 24

[IVAN03], Ion IVAN, Marius POPA, Sergiu CAPISIZU, Lukacs BREDA, Bogdan FLORESCU – *Information Cloning*, ASE Printing House, Bucharest, 2003

[IVAN99], Ion IVAN, Gheorghe NOŞCA, Sebastian TCACIUC, Otilia PÂRLOG, Răzvan CĂCIULĂ – *Data Quality*, INFOREC Printing House, Bucharest, 1999