Toward a Neural Aggregated Search Model for Semi-structured Documents

F. Z. Bessai-Mechmache
Research Centre on Scientific and Technical Information, CERIST, Ben Aknoun, 16306, Algiers, Algeria

Keywords: Neural Networks, Self-organizing Maps, Aggregated Search, XML Information Retrieval, XML Document, Aggregate, Classification of XML Elements, Learning.

Abstract: One of the main issues in aggregated search for XML documents is to select the relevant elements for information need. Our objective is to gather in same aggregate relevant elements that can belong to different parts of XML document and that are semantically related. To do this, we propose a neural aggregated search model using Kohonen self-organizing maps. Kohonen self-organizing map lets classification of XML elements producing density map that form the foundations of our model.

1 INTRODUCTION

XML information retrieval recovers information from different granularities such as document or parts of document (Kamps et al., 2003; Fuhr et al., 2004) and with different types such as images, text, videos, news…etc. However, assembly or grouping these elements in a suitable form to provide a complete view of the information available should be considered.

An ordered list of elements is the answer returned by the majority of current XML information retrieval systems (Sigurbjornsson et al., 2003; Ogilvie and Callan, 2003; Lalmas and Vannoorenberghe, 2004; Piwowarski et al., 2002). The result as an ordered liste requires the user to browse sequentially and examine the results one by one to find the appropriate content. The aggregated search comes to relieve this problem by assembling and combining elements to construct answers including all relevant information with respect to user's query. To do this, se suggest a neural aggregated search model using Kohonen self-organizing maps. Self-organizing map allows an automatic classification of XML elements producing density map to which we apply a genetic algorithm to generate aggregates.

This paper is structured as follows. We initially introduce related work concerning XML aggregated search. We describe our model in section 3. In section 4 we conclude the paper.

2 STATE OF THE ART

Few works in literature deal with aggregate search for semi structured documents as XML. The proposed models that cover this issue are limited to Web documents (Clarke et al., 2008; Agrawal et al., 2009; Kopliku, 2009; Arguello et al., 2011; Kopliku et al., 2011). So, Polyzotis and Huang were the first to have proposed a solution to aggregated search in XML documents. Indeed, XCLUSTERs (Polyzotis and Garofalakis, 2006) is a model representing XML summaries. It regroups some XML elements and uses a small space to store their abstarcts. eXtract (Huang et al., 2008) is an information retrieval system that generates results as XML fragments. An XML fragment is qualified like result if it is autonomous (understanding by the user), distinct (different from the other fragments), representative (of the themes of the query) and succinct. The Possibilistic aggregated search model (Bessai-Mechmache and Alimazighi, 2011; Bessai-Mechmache and Alimazighi, 2012) is an aggregated search model based on possibilistic networks that provide a natural representation of links between a document, its elements and its content, and allow an automatic selection of relevant aggregates. The neural aggregated search model we suggest in this paper uses Kohonen self-organizing maps (Kohonen, 1990; Haykin, 1999). These self-organizing maps allow an automatic classification of XML elements producing density map that form the foundations of aggregated search. Indeed, in order to

Bessai-Mechmache F..
Toward a Neural Aggregated Search Model for Semi-structured Documents.
DOI: 10.5220/0004538400910095
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval and the International Conference on Knowledge Management and Information Sharing* (KDIR-2013), pages 91-95
ISBN: 978-989-8565-75-4

identify the relevant aggregate which answers the query we apply a genetic algorithm (Ahmed et al., 2008; Bangorn and Quen, 2005) to the density map.

3 AGGREGATED SEARCH MODEL

3.1 Model Architecture

The architecture of the proposed model is a Kohonen neural network that consists of a grid (map) of neurons and an input stimulus, as illustrated in Figure 1 below.

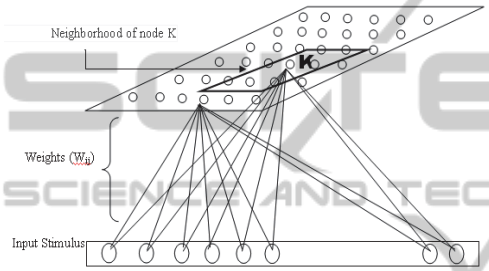

Figure 1: Model Architecture.

3.1.1 Map Description

Each neuron of the map represents an element of the XML document. Only leaves elements of the hierarchical structure of XML document are classified on the map. The non leaves elements will be classified by relevance propagation at search time.

The map is divided into subset of XML elements (areas or subregions) having the same characteristics, i.e. they deal with the same thematic. Therefore, elements belonging to the same subregion are potential candidates to appear in the same aggregate answering a given query.

Input stimulus is an N-dimensional vector of neurons that represents an element to be classified on the map. Each neuron in the input stimulus is an indexing term of the element.

Connection that join every neuron (term) of the stimulus to the map reflect the importance (W_{ij}) of term 't_i' in element 'e_j'. This importance (or weight) is calculated by the following equation (Trotman, 2005).

$$W_{ij} = tf_{ij} * ief * idf \quad (1)$$

With:
- tf_{ij} : term frequency
- ief: inverse frequency of element 'e_j' for term 't_i'.
- idf: inverse frequency

3.1.2 Connections between Elements

The Weight S_{ij} is used to model the semantic link between two elements 'e_i' and 'e_j' of the map. Sij is calculated based on the following three factors:

- The Number of Common indexing Terms (nct) (terms that the two elements have in Common); this factor determines whether the two elements deal with the same topic. It belongs to interval] 0, 1[and is calculated as follows.

$$nct(e_i, e_j) = \frac{Number\ of\ common\ terms\ between\ e_i\ and\ e_j}{Number\ of\ terms\ of\ e_i + Number\ of\ terms\ of\ e_j} \quad (2)$$

-The co-occurrence of query terms in the elements (R_{ij}).

$$R_{ij}(Q) = \frac{\sum_{t_k, t_l \in Q} \min(cooccurrences_{e_i}(t_k, t_l), cooccurrences_{e_j}(t_k, t_l))}{\sum_{t_k \in Q} occurrences_{e_i}(t_k) + \sum_{t_k \in Q} occurrences_{e_j}(t_k)} \quad (3)$$

- The Longest Common Prefix (lcp) between the Hierarchical Identifiers (HI) of the two elements (HIe_i, HIe_j). The lcp(HIe_i, HIe_j)=0 if the two elements e_i and e_j don't belong to the same document. Example: If HIe_i=**5.2**.1.4 and HIe_j= **5.2**.4.2 then lcp(HIe_i, HIe_j)= 2.

The weight S_{ij} is calculated as follows:

$$S_{ij} = nct(e_i, e_j) + lcp(HI_{e_i}, HI_{e_j}) + R_{ij}(Q) \quad (4)$$

3.1.3 Learning Algorithm

Let X(t) = { X_1(t), X2(t),…, Xn(t)} be a learning pattern at instant t. Let $W^k(t)$ = { W_1^k (t), W_2^k(t) ,…, W_n^k(t)} be a neuron at instant t.

Firstly, the map must be initialized randomly. For each input pattern:

1- Calculate the distance between the pattern and all neurones ($||X(t)-W^k(t)||$). The chosen distance measure is the Euclidean distance.

2- Select the nearest neurone as winner W^s ($||X(t)-W^s(t)|| = \min ||X(t)-W^k(t)||$)

3- Update each neurone according to the rule:
$W^k_i(t+1) = W^k_i(t) + \alpha(t).h_{(W^s, W^k)}(t).||X_i(t) - W^k_i(t)||$ with $1 \leq i \leq n$.

Let $0 \leq \alpha \leq 1$ be the learning rate, and $h_{(W^s, W^k)}$ be the neighbourhood function. This function assumes values in [0, 1] and is high for neurones that are close in the neighbourhood, and small (or 0) for neurones far away.

4- Repeat the process until a certain stopping criterion is met. Usually, the stopping criterion is a fixed number of iterations.

To guarantee convergence and stability of the map, the learning rate and neighbourhood radius are decreased in each iteration, thus converging to zero.

3.2 Query Processing

A query consists of a set of terms, Q = (t_1, t_2,..., t_m).

In our approach, we consider the query as a vector of terms ready to be classified on the Kohonen map.

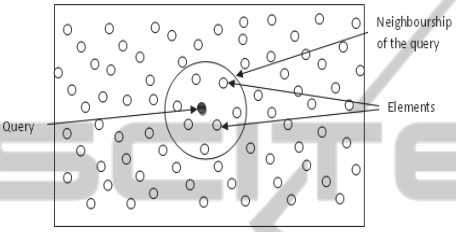

Figure 2: Classification of the query on Kohonen map.

Once the query classified on the map, the elements of its neighbourship are selected as illustrated in Figure 2 above. The set of selected elements is used to generate aggregates in response to the query. This set, denoted **SE**(Selected Element), will serve as initial population for the genetic algorithm we define in next section. **SE** = {le_1, le_2,…, le_d} with '**le**': **l**eaf **e**lement.

3.3 Aggregates Generation

The aim of our aggregate serach model is to return a list of aggregates in response to a user query. Each aggregate is composed of a set of coherent and non-redundant elements, conveying relevant information in relation to user's need. For this, we propose a specific genetic algorithm for the generation of these aggregates from the list of elements selected thanks to our self-organizing map.

Our genetic algorithm works on two populations, the population of elements 'P_e' and the population of aggregates 'P_{ag}'. Clearly, we define different fitness functions (or relevance function) to evaluate the two populations.

3.3.1 Relevance of Leaves Elements

The relevance of a leaf element with respect to the query (RSV(Q, le)) is equal to the sum of the weights of query terms with regard to this element.

$$RSV(Q, le) = \sum_{i=1}^{M} W_i^{le} \quad (5)$$

With:
- W_i^{le}: the weight of term 't_i' in leaf element 'le'.
- M: the number of query terms.

3.3.2 Relevance of Non-Leaf Element

The fitness of a non-leaf element corresponds to the relevance of this element in relation to the query. It is calculated by the following relevance propagation formula (Sauvagnat et al., 2006):

$$F_e = RSV(Q,le) = |E_{le}| * \sum_{le \in F_n} RSV(Q,le) * \alpha^{dist(e,le)-1} \quad (6)$$

With:
- $|E_{le}|$ is the number of leaves elements that are also child elements of the element 'e'.
- le: leaf element.
- $\alpha \in]0..1]$: is a parameter allowing to quantify the importance of the distance separating leaves elements from their ascendant element 'e'.
- dist (e, le) is the distance between the leaf element 'le' and its ascendant element 'e' in accordance with the hierarchical structure of the XML document.

3.3.3 Aggregate Relevance

The fitness of an aggregate 'ag', denoted 'F_{ag}', is the relevance of this aggregate in relation to the query. This relevance is calculated according to the relevance of elements composing the aggregate and according to the semantic link connecting these elements. It is formulated as follows:

$$F_{ag} = RSV(Q,ag) = \frac{1}{|E_{ag}|} * \sum_{e_i \in E_{ag}} F_{e_i} + \frac{1}{(|E_{ag}|-1)!} \sum_{e_i, e_j \in E_{ag}} S_{ij} \quad (7)$$

With:
- $|E_{ag}|$: Number of elements of the aggregate.
- E_{ag}: Set of elements those constitute the aggregate.
- F_{ei}: Relevance of element e_i with regard to the query.
- S_{ij}: Semantic link degree between two elements e_i and e_j.

3.3.4 Genetic Algorithm

To generate the aggregate result we use the following genetic algorithm:

a. Initialization

The $\mathbf{P_e}$ population is initialized by the set **SE**, so $\mathbf{P_e}$ = {le_1, le_2, …, le_d}. The $\mathbf{P_{ag}}$ Population is initialized with aggregates formed by considering all possible combinations of elements of $\mathbf{P_e}$ population. $\mathbf{P_{ag}}$= {ag_1, ag_2,…, ag_m}.

b. Assessment of population of aggregates $\mathbf{P_{ag}}$ with normalized fitness function $\mathbf{F_{ag}}$ then selection of K top individuals and their assignment to $\mathbf{P_{ag}}$ population: $\mathbf{P_{ag}}$ ← K top aggregates.

c. As long as the stopping criterion is not reached do:

- Go back up one level in the hierarchical tree of XML documents (that contain these elements) by applying the propagation of relevance.
- Add elements obtained by propagation to $\mathbf{P_e}$ population. Evaluate the new population of elements with the normalized fitness function $\mathbf{F_e}$.
- Selection of L top elements and their assignment to $\mathbf{P_e}$ population. The L top elements are potentially useful elements to generate relevant aggregates: $\mathbf{P_e}$ ← L top elements.
- Apply parameters of hybridization and mutation to $\mathbf{P_{ag}}$ population. This is to regenerate new aggregates from the new population of elements $\mathbf{P_e}$ taking into account various cases of overlap that may exist.
- Add new aggregates to $\mathbf{P_{ag}}$ population. Evaluate the new population of aggregates with the normalized fitness function $\mathbf{F_{ag}}$ and select K top aggregates: $\mathbf{P_{ag}}$ ← K top aggregates.
- Go to step c.

4 CONCLUSIONS

Our neural aggregated search model thanks to Kohonen self-organizing map assembles elements from different parts of XML documents to build aggregates including all relevant information for the query.

Future work will concern the evaluation of our approach on a data set.

REFERENCES

Ahmed, A., A., R., Bahgat, A., Abdel Latef, Abdel Mgeid, A., A., and Osman, A. S., 2008. *Using Genetic Algorithm to Improve Information Retrieval Systems*. World Academy of Science, Engineering and Technology 17.

Agrawal, R., Gollapudi, S., Halverson, A., 2009. *Diversifying Search Results*, ACM Int. Conference on WSDM.

Arguello, J., Diaz, F., Callan, J., 2011. *Learning to aggregate vertical results into web search results.* In Proceedings of the 20th ACM Conference on Information and Knowledge Management,CIKM'11, Glasgow, United Kingdom.

Bangorn, K., and Quen, P., 2005. *Applied genetic algorithms in information retrieval*. Proceedings of International Journal of Production Research (King Mongnut's Institute of Technology, Ladkrabang, Bangkok), 43, p.4083-4101.

Bessai-Mechmache F. Z., and Alimazighi, Z. 2012. Possibilistic Model for Aggregated Search in XML Documents**.** *International Journal of Intelligent Information and Database Systems*, IJIIDS, Vol. 6, No 4, pp 381-404

Bessai-Mechmache, F. Z., and Alimazighi, Z. 2011. Possibilistic Networks for Aggregated Search in XML Documents. *In proceedings of International Conference on Information & Communication Systems*, ICICS'2011, Irbid, Jordan, pp 67-72.

Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., 2008. *Novelty and diversity in information retrieval evaluation*. SIGIR'08, p.659-666.

Haykin, S., 1999. *Neural Networks: A Comprehensive Foundation*, Prentice Hall, ISBN 0-13-273350-1

Huang, Y., Liu, Z., Chen, y., 2008. *Query biased snippet generation in XML search*. ACM SIGMOD, p.315-326.

Kamps, J., Marx, M., De Rijke, M., Sigurbjörnsson, B., 2003. *XML Retrieval: What to retrieve?* ACM SIGIR Conference on Research and Development in Information Retrieval, p.409-410.

Kopliku, A., 2009. *Aggregated Search: From information nuggets to aggregated documents*. CORIA, p.507-514.

Kopliku, A., BOUGHANEM, M., PINEL-SAUVAGNAT, K., 2011. *Searching within Web pages: Retrieving attributes from HTML tables in the Web*. In Conference on Information and Knowledge Management, CIKM'11, Glasgow, United Kingdom.

Lalmas, M., Vannoorenberghe, P., 2004. *Indexation et recherche de documents XML par les fonctions de croyance*. CORIA'2004, p.143-160.

Fuhr, N., Lalmas, M., Malik, S., Szlavik, Z., 2004. *Advances in XML Information Retrieval: INEX 2004*. Dagstuhl Castle, Germany, December 6-8.

Ogilvie, P., Callan, J., 2003. *Using language models for flat text queries in XML retrieval*. In Proceedings of INEX 2003 Workshop, Dagstuhl, Germany, p.12-18.

Piwowarski, B., Faure, G.E., Gallinari, P., 2002. *Bayesian Networks and INEX. In INEX 2002 Workshop Proceedings*, p.149-153, Germany.

Polyzotis, N., Garofalakis, M. N., 2006. *XCluster Synopses for Structured XML Content*. ICDE.

Sauvagnat, K., Boughanem, M., Chrisment, C., 2006. *Answering content-and-structure-based queries on XML documents using relevance propagation.*

Information Systems, Special Issue SPIRE 2004, vol. 31, p.621-635, Elsevier.

Sigurbjornsson, B., Kamps, J., de Rijke, M., 2003. *An element-based approach to XML retrieval*. INEX 2003 workshop, Dagstuhl, Germany.

Kohonen, T., 1990. *Self-organizing map*. Proceedings of the IEEE, Vol. 78, n°9.

Trotman, A., 2005. *Choosing document structure weights*. Information Processing and Management, vol. 41, n°2, p.243-264.