

SWAPPER – SELF-ORGANIZING AUTOMATIC CONTEXT VISUALIZATION

Thomas Riisgaard Hansen

Information Study and Computer Science, Aarhus University, thomasr@imv.au.dk

Abstract: The amount of unstructured texts has recently largely increased due to technologies like the Internet and large databases. Many data mining approaches have tried to organize documents according to different categories, but even narrow categories sometimes contain several thousand items.

This project is about trying to automatically generate a map visualizing the different contexts, which is represented in a collection of documents. This approach can be used to give an overview of a set of documents without having to go into all the details. At the same time this approach might yield some result in knowledge discovery, because if several different documents contain similar information, this information will be group together as a context.

The project uses the Kohonen Self-Organizing Map (SOM) to generate the map, but also introduces some new techniques like the notion of Dynamic Self-Organizing Map to handle large feature vectors and the notion of distance table as an alternative to using histograms in the attempt to build the feature vectors.

Key words: Text/data mining, Text visualization, Self-organizing Map, Dynamic Self-Organizing Map, Context representation

1. INTRODUCTION

With the upcoming of the information society, the amount of information produced and available to every person has greatly increased. The main part of this information is encoded as text, but because of the increasing amount, we need new ways to navigate, search and probe text. Different manual approaches have been tried out: Dividing the document into layers (Title, abstract and text), providing metadata (keyword, version, author etc) and categorizing the documents.

Many different algorithms have also been proposed in automating the task of categorizing a collection of documents (or images) [2][3][4][5] all using the method of self-organizing maps first presented by Kohonen[1]. The problem is that even relatively narrow categories sometimes contain several thousand different documents (For instance a search for the enzyme cyclooxygenase yield 21.200 matches on the search machine google).

In this project a collection of documents is processed with the purpose of generating a context map representing an abstract of the different contexts in the documents. This context map is not meant as a tool for categorization, but as a graphical way for humans to get an overview of a collection of documents and as a tool for knowledge discovery by merging different contexts.

The main algorithm used in the project is a special version of the Self-Organizing Map algorithm denoted Dynamic Self-Organizing Map, which is able to dynamically alter it's model vectors to provide a better fit with a large input space. (The term Dynamic Self-Organizing Maps is also used in other contexts [6]. In these cases however the focus is on the maps ability to grow and shrink called GSOM and has nothing to do with the vectors ability to change dynamically). The input vector to the map is encoded with the use of a distance table slightly resembling the technique used in the "self-organizing semantic map"-method [7].

2. THE MODEL

2.1 The idea

The idea behind this project is, that a written sentence uses common words (am, are, there, they etc) to connect or bring words together to form a context. The same is to some lesser degree true within a paragraph. It's the special words used in sentences and paragraphs, which form the context. In

this project these structures are used as input to a Self-Organizing map, which then creates a context map based on a given collection of texts.

2.2 Creation of the distance table

The inputs to the algorithm are web pages, which are first pre-processed. The pages are parsed and the text within a paragraph tag (<p>) is extracted. HTML-tags (, etc.) are removed and the paragraph is subdivided into sentences using ‘.’ as delimiter. Each of the sentences are again subdivided into words. Common English words like (they, are, I, how, before etc.) are removed and a simple *stemmer* is used to deal with different inflections. The remaining data structure, referred to as the **input data structure**, is then used to create a **distance table** consisting of a collection of **relationship vectors**.

A **relationship vector** is a vector with a **name** and a number of **features**. Each feature consists of a unique **name** and a **fraction** representing how often “the name of the feature” is in the same sentence or paragraph as the “name of the relationship vector”. If they always occur together this value is 1 and if they never occur together the value is 0.

Table 1. The creation of a relationship vector

It is summer and the children are playing. They are playing with balls and toys.

If these are the original two sentences in a paragraph the text is first filtered.

Summer, children, playing. Playing balls toys

The algorithm picks up the first word “Summer”. Children and playing are added to a new relationship vector called “Summer”, and given a value of 12 point because they are in the same sentence as summer. Playing, balls and toys are also added and given 5 point because they are in the same paragraph as summer, which means that playing ends up with 17 points (The points are parameters).

This is the relationship vector Summer after parsing the two sentence

Summer: Children 12, Playing 12+5, balls 5, toys 5

This process is repeated for every word.

Children: Summer 12, playing 12+5, balls 5, toys 5

Playing: Summer 12+5, children 12+5, balls 5+12, toys 5+12

Balls: Children 5, playing 5+12, toys 12

Toys: Children 5, playing 5+12, balls 12

The following example shows how the relationship vector **professor** is represented in a distance table after an analysis of a web log (Internet diary):

Professor,21?idea,21?read,13?education,12?students,12?papers,9?college,8?thoughts,8?theses,8?expounded,8?fell,8?ears,8?mainly,8?professor,8?things,7?retrospect,7?find,7?frustrating,7?traditional,7?virtual,7?traffic,7?between,7?teachers,7?writing,6?class,6?author,4?posts,4?online,4?today,4?thinking,4?lat

ely,4?English,4?major,4?wrote,4?spent,3?time,3?listening,3?works,3?both,3?classes,3?guests,3

It’s easy to see how the words relate even though no one has taught the algorithm how to define a professor. The list is formed purely from analysing one year of entries in an internet diary.

There is one relationship vector for each distinctive word in the input data structure. The number of features in a feature vector would ideally also be the number of distinctive words in the input data structure, because all word from the input data structure would have a distance to all other word in the structure. In the test cases however the number of distinctive words in the input data structure ranged between 1500 to 5000 words, which were way too many for practical use and besides, many words often occurred only once or never together. Therefore the size of the relationship vector in this project was limited to a small constant (50) and only the largest fraction was remembered. However with the use of different decay mechanisms [1] new words were allowed to enter the vector and words, which seldom occurred together with the “relationship vector name”, were removed thereby creating a dynamic relationship vector. The creation of the relationship vector is illustrated in table 1.

2.3 The map

The distance table in itself represents relationships between words and may be used as input to many different machine-learning algorithms. In this project the relationship vectors are used as input to a neural net algorithm to produce a Self-Organized Map (SOM). The idea behind a SOM is to map an input data space of n-dimension onto a two-dimensional map of **reference vectors**. The algorithm then uses a distance function to calculate the best match between the input data and the reference vector node and then updates the node and it's neighbourhood to resemble the input data (the idea behind the Self-Organizing map will not be further explained here [1]).

In this project the lattice type of the map is rectangular and the size of the map varied (normally around 15x10). A large size gave more detailed maps and a small size gave a more general map. The structure of the reference vector resembled the relationship vector and consisted of a number of features with unique names and a fraction value. The size of the reference vector was also limited to a constant (also 50) and using random chosen relationship vectors as start reference vectors the map was initialised.

2.4 The dynamic Self-Organizing Map

One of the problem concerning the use of Self-Organizing Map with natural language as input is the vast number of features. A single text typically contains several thousand distinct words and this is far too much for an effective SOM algorithm. Several different methods have been tried

out: multidimensional scaling [8], Latent semantic indexing[9], Random projected histograms[10], Histograms on the word category map[7].

In this project a technique referred to as dynamic Self-Organizing Map (dSOM) is being used. With a dynamic Self-Organizing Map the size of the model vector is fixed, but the actual features are dynamic and interchangeable. When the algorithm is done using dSOM, nodes next to each other will contain almost identical features in their reference vector only differing in their values, whereas notes in remote area of the map might contain notable different features in their reference vectors. This effect is obtained by modifying two of the central parts of the SOM-algorithm namely the find function and especially the update function.

2.5 The find function

The purpose of the find function is to find the closest match between the input relationship vector and the note with the reference vector resembling the input the most. Each time the algorithm iterated a random relationship vector was chosen. For each note in the map the distance $dist$ between the reference vector ref_i and the relationship vector was calculated.

$$dist_{rel,ref_i} = \sum_{k=1}^{size(rel)} \sqrt{dist2(rel(k), ref_i)}$$

If the feature $f1$, represented in the relationship vector at position k , was also represented as a feature $f2$ in the reference vector, $dist2$ was calculated

$$dist2 = \sqrt{f1_{value}^2 + f2_{value}^2}$$

as the Euclidian distance between the two feature values.

If the feature was not represented the $dist2$ was set to 1. This process was repeated for all notes i in the map and the best match was the reference vector at note i giving the smallest $dist$. This reference vector or node was used as center for the updating function.

2.6 The update function

The update function used the reference vector found by the find function as centre (p_x, p_y) and updated this point and the neighbourhood (x_i, y_i) in a radius r around this point according to this SOM-algorithm function:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)]$$

$$distxy = \sqrt{(x_i - p_x)^2 + (y_i - p_y)^2}$$

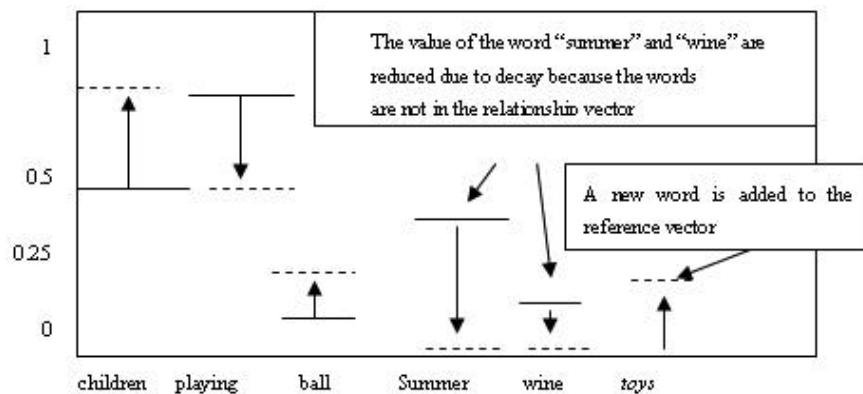
$$h_{ci} = \alpha(t) * e^{-\frac{distxy}{deviation(t)}}$$

$m_i(t)$ being the reference vector at time t , $x(t)$ the relationship vector and h_{ci} a function that determinates the strength of the update function depending on the neighbourhood distance.

For all features $f1$ in the relationship vector the reference vector was updating according to the following rules. If the feature $f1$ was represented as a feature $f2$ in the reference vector, the value of the feaure $f2$ was updated according to:

$$f2_{value}(t+1) = h_{ci} * (f1_{value} - f2_{value}(t))$$

If the feature $f1$ is not found the feature is added to the reference vector with a small fraction $c(t)$ and eventually pushing another feature out of the reference vector. Again there is a decay function enabling new features to enter and seldom-updated features to leave [I2]. This is what referred to as **dynamically updated features**. The update process is illustrated in figure 1.



Reference vector: children 0.5 Playing 0.75 Ball 0.20 Summer 0.45 Wine 0.27

Relationship vector: Children: 0.75 Playing 0.50 Ball 0.30 Toys 0.25

Figure 1. Updating proces

The radius of the neighbourhood, α and the deviation are reduced as a function of the time to ensure stabilization of the map. Different functions were tried out for α , radius and the deviation.

3. EXPERIMENTS AND RESULTS

3.1 Results

The experiments and testing were done on a collection of Internet pages. One of the test collections was the entries in a personal web log (Internet

diary) during an 18-month period [TC1]. Normally web logs are written as a stream of consciousness and without structure. The test was to see if the self-organising map was able to give an overview of the web log. Another test case was a collection of interviews and essays written by the philosopher Manuel de Landa[TC2] and the last test case was the first twenty web pages found on the enzyme cyclooxygenase[TC3].

Normally when the results of a self-organizing map are presented the map is divided into different groups. The map generated in this project had a lot of small contexts gradually changing and it was very hard to arrange the map in different groups. The main measurement was to see how much the shortest average distance (SAD) between the relationship vector and reference vector decreased as a function of time. Because the map has a much smaller dimension than the input space a decrease in the SAD would represent an economisation of the map. This is reflected in the reference vectors, which do not resemble one relationship vector, but is constituted by a collection of related relationship vectors and thereby reflecting a context. Hence a decrease in the SAD would at the same time represent a collection of relationship vectors being brought together in reference vectors to form contexts.

One of the great challenges was to find the parameters giving the best ordering and to speed the process up. With the large test cases (4000 word) the random map normally started out with a SAD around 12.5, and at the end ended up with a SAD around 6, with the more homogeneous enzyme test cases (2300 word) it started out at around 8 and ended at 3. The figure 2 shows the development of the SAD on TC1 as a function of the time and with different parameters.

The drop from 12 to around 7 happened at $t=12000$ (TCP1 and TCP2), at $t=9000$ (TCP3) and at $t=6000$ (TCP4 and TCP5). This correspond with the time where the radius was reduced to 1, which gave a more specialized map and a dramatically decrease in SAD. The different parameters, which were regulated, were the decay constant, the alpha function, the radius function, the deviation function and some other constants representing the strength of new features entering into the reference vector. Work still have to be done to clarify the influence by the different parameters.

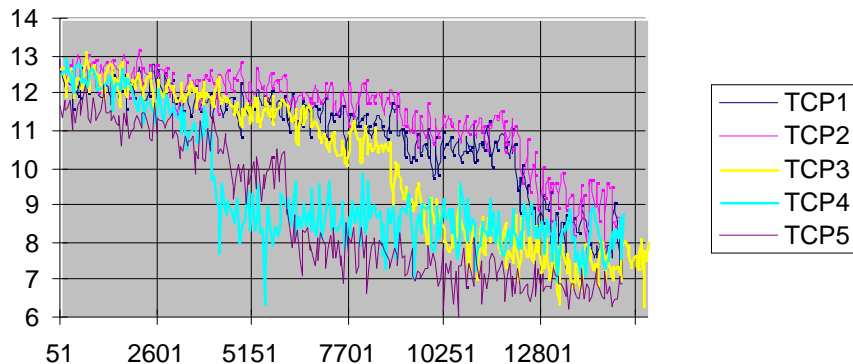


Figure 2. The development of shortest average distance as a function of the time

3.2 Visualization

The map is graphically visualised and it's possible with a mouse click to browse around and investigate different areas of the map. When you click on a field of the map the first 10 features are shown along with their name and fraction value. The first three features are coloured red, green and blue and the map is coloured according to these three features. If a point on the map (a reference vector) contains the first feature this point is coloured red*(a special computed fraction value of the reference vector feature in this specific point **fp**). If it contains the second feature it's coloured green***fp**, and if it contains the third feature it's coloured blue***fp**. This means that if a point contains all three features it gets a shade of grey/white etc.

This approach differ from for instance the approach suggested by Kohonen [1][2], where the map is coloured according to the internal distance between the different vectors in the map. With this project it proved difficult to use this approach because the variance in the internal distance between the vectors were very small. At the same time it was important in this project to visualize the different contexts. The approach chosen was to describe the context with a collection of words and visualize related contexts. This visualization approach contains a lot of information, is exploratory and interactive on the behalf of providing a fast overview.

One would expect the map to be coloured only in a small circle around the chosen point, but that was not the case. Instead the map contained a lot of small coloured areas as the following example shows. This is because different words occur in different contexts with totally different meanings. For example the word "event" might be used as in figure 3 to represent the

context around event driven software, computer interfaces and machine language, but in another part of the map, “event” might symbolise a surprise party, a natural disaster etc, which is not directly connected to event driven software. This is the reason why the map is not only coloured around the chosen point but instead in small colonies. At the same time most of the points on the map were easily understood without the original text and it was easy to find the originally part the map had used to generalize from as the following example shows.

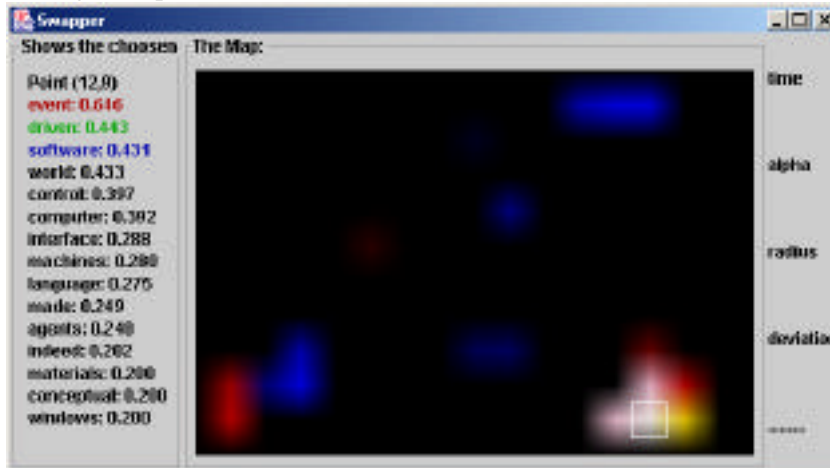


Figure 3. An example from a collection of essays TC2

3.3 An example

The following screen shot (figure 4) is taken from a collection of interviews and essays from the philosopher Manuel de Landa. The active part of the map is the point (0,0), which is the white rectangle in the upper left corner.

The ten first and most important features of the reference vector is: optimal, pressures, peacock, selection, females, different, feathers, given, populations, males, come, example, idea, under, dilemma. This point is generated from many parts of the text collection, but many of the features can be found in this paragraph from the input space (table 2).

If we move around the map we'll find different contexts from the input space arranged in a way that economize the map. Contexts, which are related, are situated close to each other. If you are travelling the map in the y-direction, the map contains subjects about natural selection and chemistry, and if you travelling the map in the x-direction it's about Greeks, black smith, metal, warfare etc.



Figure 4. An example from texts by Manuel de Landa

In the enzyme case certain areas of the map was also identified as cross-textual areas, where a small part of the word in the reference vector originated from only one or two texts (ishaemic, chromosome, colon, cacinogenesis). These words were brought together by the other half of the reference vector, which normally contained words that occurred in several texts (brain, genes, mRNA).

Table 2. Part of the original Manuel de Landa web page

<P>One may wonder just what has been achieved by switching from the concept of a "fittest mutant" to that of an "optimal" one, except perhaps, that the latter can be defined contextually as "optimal given existing constraints". However, the very idea that selection pressures are strong enough to pin populations down to "adaptive peaks" has itself come under intense criticism.

One line of argument says that any given population is subjected to many different pressures, some of them favoring different optimal results. For example, the beautiful feathers of a peacock are thought to arise due to the selection pressure exerted by "choosy" females, who will only mate with those males exhibiting the most attractive plumage. Yet, those same vivid colors which seduce the females also attract predators. Hence, the male peacock's feathers will come under conflicting selection pressures. In these circumstances, it is highly improbable that the peacock's solution will be optimal and much more likely that it will represent a compromise. Several such sub-optimal compromises may be possible, and thus the idea that the solution arrived at by the "searching device" is unique needs to be abandoned. 4 But if unique and optimal solutions are not the source of stability in biology, then what is?.

For more experimental result, complete maps and more screen shots in colours see (<http://imv.au.dk/~thomasr/Swapper/>)

4. DISCUSSION AND CONCLUSIONS

4.1 Conclusion

The project shows a still primitive way of how to generate an overview from an information space and presenting it in a graphical manner. It shows another way of extracting information about a text, not by generating a histogram of the word frequency, but by building a context table. From the map it's possible to get an overview of an information space, not by reading an abstract but by browsing around, and by simple search it's easy to find the different areas in the information space, which the map generalizes from. The project shows a way for knowledge discovery by merging information from different document into one context. Finally this project shows how to implement a dynamic self-organizing map, which is able to switch the actual features in the reference vector.

4.2 Future work

Some of the area left for future research is to further update the parameters and speed up the process. As well the project needs to be tested on large collection of related papers to further explore the patterns created by the project and the possible for knowledge discovery. Another area open for future research is to find a better way of measuring the quality of a self-organizing map, and work needs to be done on how to improve visualizing technique of a self-organizing map and to explore the possibilities available with context tables.

REFERENCES

- [1] Kohonen, T, *Self-Organizing Maps*, Spring, 1995
- [2] Kohonen T, Self-Organization of Very Large Document Collections: State of Art, Proceedings of ICANN98
- [3] Laaksonen, Jorma, Koskela, Markus, Laakso, Sami & Oja, Erkki, PicSOM – content-based image retrieval with self-organizing maps, Pattern Recognition Letters, 2000
- [4] Lin, Xia, Soergel, Dagobert & Marchionini, Gary, A Self-Organizing Semantic Map for Information Retrieval, ACM-library
- [5] Lee, Chung-Hong & Yang, Hsin-Chang, A Web Text Mining Approach Based on Self-Organizing Map, ACM-library
- [6] Alahakoon, Daminda, Halgamuge, Saman K., Srinivasan, Bala, *Dynamic Self-Organizing Maps with Controlled Growth for Knowledge Discovery*, IEEE Transactions on Neural Networks, Vol. 11, No. 3. May 2000
- [7] Ritter H, Kohonen T. *Self-organizing semantic maps*, Biol Cyb, 1989, 61:241-254

- [8] Y. Linde, A. Buzo, and R. M. Gray, *An algorithm for Vector Quantizer Design*, IEEE Transaction on Communications, vol. 28, pp 84-94, Jan 1980.
- [9] Deerwester s, Dumain S, Furnas G, Landauer K., *Indexing by latent semantic analysis*, J Am Soc Inform Sci, 1990; 41:391-407
- [10] Kaski S. *Data exploration using self-organizing maps*, Acta Polytechnica Scandinavica, Mathematic, Computing and Management in Engineering Series No 82, 1997. Dr Tech Thesis, Helsinki University of Technology, Finland.

IMPLEMENTATION NOTES

- [I1] The decay mechanisms were implemented by multiplying each feature value in the reference vector by a constant ($0 < c < 1$) after each update. The reference vector was order according to the decreasing feature value. Each time a new feature was added, the feature was added at the right place and low feature values were pushed out. After the distance table was created, reference vectors corresponding to words occurring only once or twice were removed to give a smaller table and making it more resistant to deviations.
- [I2] This decay mechanism was implemented slightly different by multiplying each word in the reference vector, which was not updated by the value:
 $1 - ((1 - \text{Constants.DECAY}()) * \text{hci}(t))$
 Constants.DECAY() being a constant and hci(t) being the value calculated by the update function.

TEST CASES

- [TC1] Web log
 Megnut, entries from January 2000 to June 2001, <http://www.megnut.com>
- [TC2] Manuel de Landa
 - An Interview with Manuel de Landa at VirtualFutures, Warwick 96
<http://www.t0.or.at/delanda/intdelanda.htm>
 - Economics, Computers and the War Machine by Manuel de Landa
<http://www.t0.or.at/delanda/netwar.htm>
 - Markets and antimarkets in the world economy by Manuel de Landa
<http://www.t0.or.at/delanda/a-market.htm>
 - The geology of morals by Manuel de Landa
<http://www.t0.or.at/delanda/geology.htm>
 - Uniformity and variability by Manuel de Landa
<http://www.t0.or.at/delanda/matterdl.htm>
 - Meshworks, hierarchies and interfaces by Manuel de Landa
<http://www.t0.or.at/delanda/meshwork.htm>
 - Virtual environments and the emergence of synthetic reason by Manuel de Landa
<http://www.t0.or.at/delanda/delanda.htm>
- [TC3] Enzyme cyclooxygenase:
 A complete list of web pages used in this test can be found on
<http://imv.au.dk/~thomasr/Swapper/>
- [Further tests] More test result and implementation details can be found on
<http://imv.au.dk/~thomasr/Swapper/>