Methodological Review

Unsupervised grammar induction and similarity retrieval in medical language processing using the Deterministic Dynamic Associative Memory (DDAM) model

Stefan V. Pantazi *

Health Informatics, School of Health & Life Sciences and Community Services, Conestoga College Institute of Technology and Advanced Learning, 299 Doon Valley Drive, Kitchener, ON, Canada N2G 4M4

ABSTRACT

This paper is an overview of unsupervised grammar induction and similarity retrieval, two fundamental information processing functions of importance to medical language processing applications and to the construction of intelligent medical information systems. Existing literature with a focus on text segmentation tasks is reviewed. The review includes a comparison of existing approaches and reveals the long-standing interest in these traditionally distinct topics despite the significant computational challenges that characterizes them.

Further, a unifying approach to unsupervised representation and processing of sequential data, the Deterministic Dynamic Associative Memory (DDAM) model, is introduced and described theoretically from both structural and functional perspectives. The theoretical descriptions of the model are complemented by a selection and discussion of interesting experimental results in the tasks of unsupervised grammar induction and similarity retrieval with applications to medical language processing. Notwithstanding the challenges associated with the evaluation of unsupervised information-processing models, it is concluded that the DDAM model demonstrates interesting properties that encourage further investigations in both theoretical and applied contexts.

1. Introduction

In [1] we have postulated the need for intelligent approaches that can address the high complexity and sensitivity of Medical Informatics (MI) applications through the axiom that "medical information systems must be, at the same time, usable and useful". In [2,3] it was also shown that addressing this problem requires advancements in representing and processing information on conceptual principles in an unsupervised, human-like manner. The relevance of Information Retrieval (IR), Case-based Reasoning (CBR) and Natural Language Processing (NLP) applications and approaches in this context was also noted and the unification of these traditionally distinct areas of research was proposed as the general goal of managing associative concept representation spaces. This unifying goal was addressed by work on a new memory model, the Deterministic Dynamic Associative Memory (DDAM) [3]. The main purpose of this paper is to review research on unsupervised grammar induction and similarity retrieval, two fundamental information processing functions with applications to medical language processing. The paper begins with a background section outlining the rationale of this research and underlining the importance of the algorithmic information theory for informatics. Further, a definition of unsupervised grammar induction is offered followed by a review of literature on this topic, with a focus on text segmentation tasks. A theoretical description of the DDAM model follows, from both a structural and functional perspective but with an emphasis on the latter. The presentation makes use of examples and visual aids and attempts assume as little as possible about the readers’ knowledge of the formal concepts discussed. The theoretical section is complemented by descriptions of selected experiments and results with relevance to unsupervised medical language processing, followed by conclusions and future work.

1.1. Background

A detailed discussion of the theoretical implications of this research, including a proposal for redefining the notion of clinical evidence to include individual case data, is available elsewhere [4]. Here I only reiterate the proposal that MI is essentially a case-oriented discipline [2] where explaining, predicting and managing individual clinical situations has the highest importance and equates to a “context-dependent collection, processing and
communication of medical information in order to support decision-making processes” [5].

The attempts to explain at a more fundamental level why, in naturalistic environments (e.g. clinical medicine), medical decision-making is biased and often departs from the objectiveness of normative decision models, has identified Algorithmic Information Theory (AIT) [6] as a relevant area of inquiry [1]. It is said [7] that rather than focusing on ensembles and probability distributions as in classical statistics and information theory [8], AIT focuses on the algorithmic properties of individual objects. Essentially, AIT is a pattern processing (discovery and recognition) perspective on Information Theory where the important notion of randomness hinges on the ability (or inability) of detecting patterns in data [9]. Quantitatively, AIT employs a very simple theoretical measure (i.e., algorithmic complexity) in form of the length, expressed in bits, of the shortest computer program that computes/represents one's data. As a consequence, hypothetical, random data that contains no regularities whatsoever has a high algorithmic complexity which is equal or very close to the length (in bits) of the data itself: data compression is not possible in this case. Conversely, data containing many regularities and patterns is said to have a low algorithmic complexity. The tenets of AIT (e.g., algorithmic complexity, randomness, data compression, minimum description length (MDL)) (also see [6] for an introduction) can be used to define the notion of algorithmic significance as the reoccurrence (i.e., at least twice) of a sufficiently long sequence of observations (i.e., a significant pattern). The concept of algorithmic significance may allow us to explain some of the characteristics of clinical reasoning that are based on the natural ability of recognizing patterns or regularities in data and to further the relevance of memory-based approaches, analogical reasoning, IR and CBR to clinical medicine. Pattern recognition is an undisputed feature of human cognitive abilities and a research area in its own right. It is also the fundamental mechanism behind associative or similarity-based retrieval which, in the context of computerized information retrieval, is referred to as retrieval on secondary keys [10]. In terms of memory models, this view may allow us to explain the need for information-processing models where computational complexity tradeoff aims at minimizing the read/retrieval complexity at the expense of more complex write/updates. This perspective on information processing is in agreement with the definition of “statistically rare but algorithmically significant patterns” (or algorithmic significance) where the algorithmic properties (e.g., description lengths) are more important than their counts (i.e., statistics). This also links back to the proposal to redefine the notion of clinical evidence to include individual case data whose algorithmic properties are more significant than their statistical properties (e.g., frequency of occurrence) and complements well what is currently accepted as evidence in Evidence-based Medicine, where counting (or frequency) is still considered the best criterion for judging significance (i.e., statistical significance).

2. Unsupervised grammar induction

A grammar is the specification, instructions or set or rules of how to construct something from a finite set of smaller components. Grammar induction is the often challenging process of determining a set of rules or patterns from a set of language examples (or strings). Intuitively, grammar induction is a process akin to what children do while acquiring a new language or to what scientists do when deriving a theory from their observations and data. For natural languages, unsupervised grammar induction is a very difficult problem, especially if one expects results to resemble the syntactic analysis derived by a linguist [11]. Even if one does not aim specifically at linguistically correct structures, there are theoretical proofs preventing results that satisfy optimality criteria [12]. For example, the problem of deriving the smallest grammar or the most concise theory that explains one's observations in an attempt to attain a minimum description length (MDL) — in perfect agreement to Occam's razor precept and with obvious relevance to applications in data compression — is known to be NP complete. As a consequence, grammar induction approaches generally aim at approximate results, are guided by purpose (e.g., syntactic parsing, chunking, semantic disambiguation, etc.) and typically try to make use of any available apriori knowledge in order to improve results (i.e., supervised approaches). The process of unsupervised grammar induction, a fundamental information processing task, can also be regarded as a compositional approach to dimensionality reduction and is the basis for creating a more meaningful, potentially compressed packaging that changes the properties (e.g., dimensionality, sparseness) of representation spaces. Despite its difficulties, the general problem of grammar induction is an intensively studied topic in various contexts but predominantly in language acquisition domains [12], hierarchical chunking [13], syntactic parsing [14,15] grammatical inference [16], unsupervised language acquisition [17].

2.1. Review of literature

The literature on grammar induction is too rich to allow a review of all approaches in detail. The focus will be on a certain selected models that are particularly relevant to DDAM. A look at the chronology of the publications reveals the longstanding interest in grammar induction starting with the work of Z. Harris in 1955 [18]. The literature also reveals the predominance of n-gram models and minimum description length (MDL) approaches. The few exceptions that do not explicitly advocate MDL as a guiding principle and are particularly relevant to DDAM, are those where segmentation is based on self-organizing maps [19] and those falling under the sequence alignment paradigm [20] and [17].

G. Wolff had a long track record and interest in the problems of text segmentation, grammar induction and was one of the early developers of chunking (or segmentation) algorithms (i.e., MK10 and SNPR models). These were improvements of previously published methods such as that of Olivier [21]. Wolff has used MDL as a guiding principle [22] and has recently proposed a unifying view in form of his ICMAUS framework (information compression by multiple alignment, unification and search). SP theory and models [23], where the sequence alignment paradigm is essential.

In his dissertation [16] J. Hutchens has proposed a chunking model based on information-theoretic principles. Without any prior knowledge about separators in a text, by monitoring the entropy level in an n-gram model, the algorithm was able to pick up chunks (e.g., words) based on the increase in entropy that occurred naturally around text separators such as blanks. Later Hutchens realized the similarities of his models with those of Wolff.

In his dissertation [13], C. Nevill-Manning has proposed an algorithm named SEQUITUR, that was successfully applied to a diversity of tasks but which did not include similarity-based retrieval. Though it shared similarities with Wolff's MK10 model, SEQUITUR was incremental and had a linear time and space complexity that made it an efficient and elegant approach that was an improvement over MK10.

By the same time as C. Nevill-Manning, C. deMarcken explored in his dissertation [24], linguistically plausible mechanisms for grammar induction and text segmentation models based on MDL principles. The fact that his algorithms required multiple passes through data is a major difference from SEQUITUR and DDAM whose processing is local to the current input. However, the approach of deMarcken is extremely relevant to DDAM from the perspective of its aim of finding increasingly larger patterns in a
bottom-up fashion and of creating hierarchical representations of strings. With respect to performance, deMarcken reported segmentation precisions of over 95% [24], a truly remarkable result that seemed to have settled the unsupervised text segmentation problem. However, on closer inspection of the evaluation methodology, it has been observed ([25,26]) that there was a complete lack of commitment to particular segmentations derived from the hierarchical representations. This has caused a re-estimation of the segmentation precision to about 17%, based on the probability that a particular segmentation out of several ones is the correct one.

Starting with the work on the DR (Distributional Regularity) [27] and later MBDP (Model Based Dynamic Programming) [26], M. Brent has set the state of the art in unsupervised text segmentation models ([25,28–30]) all based on the very same principles: MDL, n-grams and Viterbi search (dynamic programming). Nearly all of these models have been successfully used to approach a very specific task, namely modeling speech segmentation and child language acquisition for which the estimated average word segmentation precision and recall ranged between 65% and 80%, respectively.

To attain these results, MBDP models have been specifically optimized (e.g., MBDP “makes very good use of sentence boundaries […]” [31]) to work best on phonetic transcriptions of spoken language such as those in the CHILDES corpus [32] containing very short utterances with many repetitions. However, this high specificity has rendered MBDP models less applicable to other types of data that contains longer utterances with limited repetition [25]. A collection of published results of existing approaches (Table 1) demonstrated that DDAM (in two slightly different variations, 2 and 2.1) can yield comparable results in the English text segmentation task at the word level, on some specific corpora such as Alice in Wonderland (AIW) and CHILDES.

Despite their success at modeling language acquisition in children, the fact that MBDP models are all based on the explicit, top-down search in the solution space of the representations that satisfy the MDL principle, makes MBDP family of algorithms implausible biologically when compared to self-organizing map approaches or to models that construct a bottom-up, hierarchical representation of data (e.g., deMarken’s model, SEQUITUR and DDAM).

Van Zaanen explored in his dissertation [20] the acquisition of structure from sequences. His work was done in paradigm of alignment based learning (ABL), a relatively new approach, highly relevant to bioinformatics and natural language processing. In the natural language processing realm, the principles of ABL can be found in various equivalent formulations and the structures relevant to sequence alignment are referred to in various ways such as, for example, sausage graphs, word lattices, lexical chains, etc. [34,35]. The DDAM model can also be reformulated in the sequence alignment paradigm.

One other relevant model of inducing structure from unstructured data is ADIOS (Automatic Distillation of Structure) [15,17]. Besides general similarities, the relevance of ADIOS to DDAM is also significant in the specific aim at creating hierarchical structures in a bottom-up fashion and in the overall similarity of learning with sequence alignment procedures.

2.2. Text segmentation

Text segmentation can be considered a form of grammar induction that aims at the decomposition of a text into a series of compositional building blocks that may be morphemes, words or phrases. Therefore, this general task includes what in literature is referred to as morphosemantic decomposition and word segmentation. Text segmentation also includes situations where the text may be artificially created from nonsense syllables and words. It also covers the case where separators (e.g., blanks, commas, periods, brackets, etc.) – normally used in order to separate lexical items in many languages – are removed in order to eliminate the importance of separators in unsupervised lexical acquisition evaluation tasks.

2.2.1. Word segmentation

Word segmentation is a form of text segmentation where unsegmented text (i.e., text without any separators or punctuation) is segmented in words. Though unusual for languages such as English, word segmentation is of high interest and forms the object of computational linguistics approaches for the Asian languages whose writing systems do not include the use of separators (e.g., Chinese) [24,25,28,31,36].

For English language, much of the interest in word segmentation has stemmed from research on hierarchical chunking and from the study of language acquisition [21,24,26,27,29,30,37] that aims at explaining how children acquire language words, given the little or total lack of feedback information they receive with regard to the word boundaries. Other word segmentation tasks have been reported in the context of optical character recognition (OCR) systems [36] that may result in streams of text where word boundaries are occasionally suppressed.

Unsegmented text has variable levels of ambiguity and difficulty of parsing that occasionally may pose difficulties even to human language processors, especially for uncommon words. However, when facing this somewhat unusual task, humans usually employ multiple strategies and make use of any useful piece of information that helps them in the process, including high-level semantic knowledge about the concepts present in a text. Performing such a task, especially on a “difficult text,” may provide insights into the levels of processing and levels of information needed for text understanding. A human may need to read the text multiple

---

Table 1

<table>
<thead>
<tr>
<th>Model</th>
<th>Corpus</th>
<th>Phonetic transcription</th>
<th>Per word precision (%)</th>
<th>Per word recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQUITUR [13]</td>
<td>AIW</td>
<td>No</td>
<td>35</td>
<td>28</td>
</tr>
<tr>
<td>DDAM-2</td>
<td>AIW</td>
<td>No</td>
<td>45</td>
<td>43</td>
</tr>
<tr>
<td>SCHONE01 [25]</td>
<td>TREC-IR</td>
<td>No</td>
<td>12</td>
<td>25</td>
</tr>
<tr>
<td>DEMARCKEN96 [24]</td>
<td>CHILDES</td>
<td>Yes</td>
<td>17</td>
<td>–</td>
</tr>
<tr>
<td>SOM [19]</td>
<td>CHILDES</td>
<td>Yes</td>
<td>18.9</td>
<td>36.6</td>
</tr>
<tr>
<td>DR [27]</td>
<td>CHILDES</td>
<td>Yes</td>
<td>41</td>
<td>47</td>
</tr>
<tr>
<td>SCHONE01 [25]</td>
<td>Switchboard</td>
<td>Yes</td>
<td>54</td>
<td>56</td>
</tr>
<tr>
<td>DDAM-2.1</td>
<td>CHILDES</td>
<td>Yes</td>
<td>61</td>
<td>65</td>
</tr>
<tr>
<td>Bootlex [30]</td>
<td>CHILDES</td>
<td>Yes</td>
<td>67.2</td>
<td>68.2</td>
</tr>
<tr>
<td>MBDP [26]</td>
<td>CHILDES</td>
<td>Yes</td>
<td>80 (71% avg)</td>
<td>80 (72% avg)</td>
</tr>
<tr>
<td>MBDP variant [29]</td>
<td>CHILDES</td>
<td>Yes</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>DLG[33]</td>
<td>CHILDES</td>
<td>No</td>
<td>75</td>
<td>71</td>
</tr>
</tbody>
</table>
times and make extensive use of his/her knowledge because of the increased ambiguity. The use of artificially generated text that comprises nonsense words only exacerbates these processes.

Artificial data comprising nonsensical lexical items could be thought of as being able to bring human processors closer to a more primitive information-processing model by removing some of our powerful semantic processing capabilities. This may provide additional insights into processing mechanisms and could help with the development of pattern discovery and recognition algorithms suitable for lexical acquisition and associative information retrieval, such as the DDAM model. The German psychologist Hermann Ebbinghaus was the first to empirically investigate associationist memory mechanisms in this manner. In his attempts to objectively measure the association power of human memory, Ebbinghaus has made use of nonsense syllables (e.g., NUH, VEG, KUR, etc.) as they "have the property of removing certain cross-associations that manifest and are variable from person to person" [38]. The fact that statistical properties of artificial texts are easy to control and the experimental results of such tasks are relatively easy to quantify, compare and discuss, makes artificial text useful for the evaluation of segmentation models and algorithms [39,40]. The DDAM model was extensively tested and evaluated on such artificial sequences [3]. In fact, this methodological approach, which allows for the complete control of the information-theoretic properties of artificial datasets, was crucial to the development and testing of the DDAM composition algorithm, given the large number of iterations and software prototypes involved.

2.2.2. Morphosemantic decomposition

Morphosemantic decomposition is a form of text segmentation that aims at the decomposition of complex lexical items – in particular of compound words from technical, professional discourses – into their semantic compositional building blocks (i.e., morphosemantemes).

The earliest account for the unsupervised learning of general natural language morphology [18] is presented in detail together with an excellent review of other existing approaches by Goldsmith [41] many of which fall under the MDL framework, including the adaptive string composition algorithm, is outside the scope of this article and is available elsewhere [3]. In this following section, the structural and functional principles behind the DDAM model will be reviewed and complemented by references to relevant literature.

3. The DDAM model

A complete, formal description of the underlying principles of the DDAM model (e.g., the generalization of combinatorial compositions), including the adaptive string composition algorithm, is available elsewhere [3]. In this following section, the structural and functional principles behind the DDAM model will be reviewed and complemented by references to relevant literature.

3.1. Structural description

At a fundamental level, DDAM is a computational model for string representation that minimizes retrieval time complexity. As such, DDAM was shown to be a generalization of the trie memory model, an associative memory model first proposed by E. Fredkin in 1960 and named trie because of importance in information retrieval. Since then, the trie model has been improved in various ways by other researchers such as R. Rivest, R. de la Briandais and D. Morrison who proposed the generalized trie, the linked list trie and the PATRICIA (Practical Algorithm To Retrieve Information Coded in Alphanumeric) trie, respectively. A good overview of trie models and algorithms is available in [10]. One major advantage of trie models [51] is that they encode content information and allow for efficient similarity-based retrieval of all sequences with a given prefix or suffix. In addition, the computational complexity of retrieval from tries, in some implementations, depends only on the algorithmic properties of the stored sequences (e.g., length) but not on their counts (statistics). This is an important property given the algorithmic perspective on information processing introduced earlier.

For example, in Fig. 1 it can be seen how the DDAM model generalizes the prefix and suffix trie memories that would represent a language L comprising the strings {January, February, March, April, May, June, July, August, September, October, November, December}. The model does possess additional representational elements that capture diafix similarities such as the pattern “ar” in March, January and February. However, in order to reduce the complexity of the depiction, only the unique patterns that make the final representation of the language are shown.

As in any prefix trie memory model, the 12 strings are deterministically retrievable from the memory after providing a prefix query (i.e., a start pattern to match). An important feature of the DDAM structure is that it allows the implementation of various search algorithms that can capture and exploit the information-theoretic properties of the representations. For example, in a retrieval context, the amount of information required (in a query) to make a decision between n equiprobable choices (in the retrieval set) is known to amount to $\log_2(n)$ bits of information. For the example in Fig. 1, a non-ambiguous query that results in only one, unique match, may contain a minimum of $\log_2(8) = 3$ bit and a maximum of $\log_2(8) + \log_2(2) + \log_2(2) = 5$ bits of prefix information. The first term (i.e., 3 bits) of the calculation stands for the amount of information needed in a query.

---

1. The choices are considered equiprobable in order to keep calculations simple, however, more complex computations, based on character and patterns frequencies, are also possible and can reduce even further the amount of information needed in a query.
tion required by a choice between the 8 equiprobable characters {S, D, O, A, J, M, N, F} which form the set of first characters of all strings in the example language. Depending on the initial choice, this may need to be followed by an additional 1 bit choice between the patterns {Au, Ap}, {Ju, Ja} or {May, Mar}. This, in turn, may be followed by another 1 bit choice between patterns {Jun, Jul}. The same logic and calculations can be employed for the suffix trie model information.

Most importantly, the DDAM model is a directed graph that is also capable of representing diafix patterns that can be matched to diafix queries. This capability of recalling all strings containing a certain pattern, anywhere in the string goes beyond the prefix and suffix trie models which are just trees. A diafix query may need to contain a roughly estimated\(^2\) amount of information between \(\log_2(18) = 4.17\) bit (i.e., for a one-character non-ambiguous query such as “v”) and \(\log_2(18) + 5\log_2(18)\), where 5 is the length of the most ambiguous substring in the language (i.e., “ember”) shown in Fig. 1 and 18 is the length of the character subset \{a, b, c, e, g, h, i, l, m, n, o, p, r, s, t, u, v, y\} of the language alphabet \(\Sigma\) which includes the additional 8 capital letters at the beginning of each string. In case of long strings, the amount of information in a non-ambiguous query may be considerably less than that provided by typing the entire string of length \(n\) and that can be estimated to be \(n \log_2(|\Sigma|)\) bits. For example, the query “v” is such a non-ambiguous query that results in the unequivocal recall of “November” and which is the equivalent to providing “only” \(\log_2(26) = 4.7\) bits of input information.

Other information-theoretic measures can be constructed in similar ways to calculate the distance (in bits) between existing DDAM representations, for similarity retrieval purposes. For the example language in Fig. 1, one possible distance calculation between “January” and “February” would be close to 0.0 bits given their high similarity involving the pattern “uary)” in this language context. By the same token, the distance between “September”, “December” and “November” is expected to have a value higher than 0.0 bits given that more than two strings share the common pattern “ember”.

\(^2\) The calculations are conservative (i.e., overestimating) the amount of information needed for a query of length \(n\) requiring \(n\) equiprobable choices between the characters in an alphabet; the main point of these calculations is to suggest that they can be done easily and naturally in a DDAM structure.

Fig. 1. Depictions of DDAM compositions of the strings in a small language (the month names); the model does possess additional representational elements however, in the figure only the unique pattern elements that make the final representations are shown in order to reduce the complexity of the image.

![Fig. 1](image1.png)

Fig. 2. Illustration of a Dyck path of semilength 3 and its corresponding Dyck word encoding (110100).
### 3.2. Functional description

Functionally, DDAM is a deterministic memory model that has been employed in tasks such as sequence alignment, unsupervised grammar induction, automated indexing and similarity-based retrieval. In addition, DDAM has been shown [3] to be equivalent to variable-order Markov and n-gram models and to possess similarities with Self-Organizing Maps (SOM), Latent Semantic Indexing (LSI) [52] and Formal Concept Analysis (FCA) [53] models as well as with connectionist models such as Kanerva’s Sparse Distributed Memory model [54].

The DDAM string compositions are built on hierarchical patterns detected in a language. Formally, DDAM representations correspond to existing combinatorial objects named Dyck paths which allow the characterization of the computational complexity of the model. A Dyck path [55] of semilength \( n \) (where \( n \) is a positive integer larger than 0) is a walk in a 2-dimensional plane that extends from a point \( A \) situated at coordinates \((0,0)\) to a point \( B \) located at \((2n,0)\). The path consists of North-East moves called rises and South-East moves called falls (see Fig. 2).

A Dyck word is a binary encoding of a Dyck path. A Dyck word has a length of \( 2n \) symbols and uses a pair of two symbols (e.g., \( u \) and \( d \), \( 1 \) and \( 0 \)), one for the rise step and the other symbol for the fall step. For example, \( ududud \) or \( [][[]] \) are equivalent Dyck words, which encode the first Dyck path of semilength 3 listed in Table 2 and which corresponds to the trivial composition of any string of length 3. Dyck words based on the \( 1 \) and \( 0 \) symbols always start with a \( 1 \) and end in a \( 0 \) and contain the same number of \( 1 \)-s and \( 0 \)-s.

In a Dyck path, a peak is the occurrence of the sequence 10 in the corresponding Dyck word, while a valley is the occurrence of sequence 01. An ascent of a Dyck path is a maximal string of 1’s together with its length while a descent is a maximal string of 0’s together with its length. Most importantly, the direct correspondence between DDAM representations and well known combinatorial objects such as Dyck paths and words allows estimating the computational complexity and the representational power of the approach by counting all possible representations. In terms of string representation and grammar induction, this allows the precise determination of the number of possible representations that a string of length \( n \) can have in DDAM. The total number of Dyck paths of semilength \( n \) is in fact given by a well known combinatorial mathematics number, the Catalan number,\(^3\) commonly denoted \( C_n \). The values of the first fifteen Catalan numbers are 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, 742900, 2674440, 9694845 [56]. Therefore, for the example in Table 2, there are \( C_3 = 5 \) possible Dyck paths of semilength 3.

The other way to regard DDAM representations and to estimate the computational complexity of grammar induction algorithms is as combinatorial compositions. This view will make clearer the connection of DDAM with text segmentation and grammar induction.

A simple combinatorial composition is an integer partition (i.e., one way of writing an integer as a sum of other positive integers) where the order of the elements matters (i.e., \( 1 + 2 \) is different than \( 2 + 1 \)). For an integer \( n \), there are \( 2^{n-1} \) such combinatorial compositions. For example in Table 2, where \( n = 3 \) there are \( 2^2 = 4 \) combinatorial compositions (i.e., \( 1 + 1 + 1, 1 + 2, 2 + 1 \) and \( 3 \)). For \( n = 4 \) there are \( 2^3 = 8 \) combinatorial compositions (i.e., \( 1 + 1 + 1 + 1, 1 + 1 + 2, 1 + 2 + 1, 2 + 1 + 1, 1 + 3, 3 + 1, 2 + 2 \) and \( 4 \)), that is, a string of length 4 can be segmented in 8 possible ways. To generalize, any simple combinatorial composition of \( n \) can be regarded as a possible segmentation of a string of length \( n \).

Given a string of length \( n \), the number of possible Dyck paths (i.e., \( C_n \)) is always equal to or higher than the number of simple combinatorial compositions (i.e., \( 2^{n-1} \)). This is clearly visible in Table 2 where for each of the four simple combinatorial compositions of 3 there exists a generalized combinatorial composition that includes negative terms and that can be easily resolved (i.e., by canceling opposite sign terms and removing zero terms) into the corresponding simple combinatorial composition. The sequence

---

\(^3\) [http://en.wikipedia.org/wiki/Catalan_number](http://en.wikipedia.org/wiki/Catalan_number) is a good starting point for a definition and overview of Catalan numbers.

---

<table>
<thead>
<tr>
<th>Dyck path and word</th>
<th>DDAM representations of abc, generalized combinatorial composition of 3 and Dyck word of semilength 3</th>
<th>Text segmentation of abc and simple combinatorial composition of 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \begin{array}{l</td>
<td>l</td>
</tr>
<tr>
<td>2</td>
<td>( \begin{array}{l</td>
<td>l</td>
</tr>
<tr>
<td>3</td>
<td>( \begin{array}{l</td>
<td>l</td>
</tr>
<tr>
<td>4</td>
<td>( \begin{array}{l</td>
<td>l</td>
</tr>
<tr>
<td>5</td>
<td>( \begin{array}{l</td>
<td>l</td>
</tr>
</tbody>
</table>
of the signs (i.e., plus or minus) of each term of the generalized compositions create the Dyck word of that composition. Intuitively, the terms of the generalized combinatorial compositions represent the length of the substring pattern they correspond to or, alternatively, the “altitude” or $Y$ coordinate value of a corresponding point on the Dyck path.

The connection to text segmentation and the grammar induction functionality of DDAM hinges on these formal concepts and amounts to determining and committing to the appropriate segmentations (i.e., simple combinatorial compositions) of the strings in a language. More precisely, for any string of length $n$, the DDAM model starts with the trivial composition which corresponds to the 1010...10 Dyck word of semilength $n$. This trivial representation is transformed, through the adaptive composition algorithm documented in [3] into a generalized combinatorial composition which aims at minimizing the overall ambiguity the representations of all the strings in a given language. Grammar induction is therefore the inverse process by which the general representations available in DDAM are resolved into simple combinatorial compositions, through an adaptive decomposition algorithm. The composition and decomposition operations are possible because all DDAM representations of the same string are completely aligned and can easily be transformed into one another through simple operations. These operations have been named flip-up and flip-down as they intuitively correspond to flipping up a Dyck path valley into a peak or flipping down a peak into a valley. In terms of Dyck word encodings, flip-up and flip-down operations correspond to a change of a 01 into a 10 and viceversa (Fig. 3).

In the example in Table 2 it can be seen that all five DDAM representations are perfectly aligned. Upon reading the input string abc, the DDAM composition algorithm can easily derive its trivial composition $|a|b|c|$ (Dyck word 101010) by including the empty string in the representation. Further, the algorithm will transform the trivial composition $|a|b|c|$ into either $|a|ab\ b|c|$ (Dyck word 110010) or $|a|b\ b|c\ e|$ (Dyck word 101100) by flipping up (i.e., turning into 10) either one of the two valleys (01) present in the Dyck path of the trivial composition (Fig. 3). If the adaptive component of the algorithm requires it, either composition (i.e., 110010 or 101100) can be transformed into $|a\ ab\ b\ bc\ b\ b|$ (Dyck word 110100) through a similar flip-up operation involving the respective valleys (underlined). Finally, if required by the algorithm, the composition with the word 110100 can be transformed through a flip-up into $|a\ ab\ abc\ bc\ b\ b|$ (Dyck word 111000), a composition whose ascent/descent are maximal and whose spatial computational complexity is therefore the highest.

The inverse algorithm [3] aims at the decomposition of generalized DDAM representations into grammars. The decomposition is a recursive function, guided by an adaptive component dependent on the information-theoretic properties of the DDAM representations of all strings in a given language. Its aim is to determine, though flip-down operations, the simple combinatorial representations that correspond to appropriate (i.e., satisfying a certain criterion or objective function) segmentations (or grammar) of the strings in a language. For the more complex example language comprising the month names, one such grammar (Table 3) was derived in precisely this way from the generalized DDAM compositions of the 12 strings.

The analysis of the composition/decomposition algorithms available in [3] has determined an upper bound as the number of flip operations necessary to transform a trivial composition of a string into a maximal ascent/descent one and vice versa. This upper bound is important in that it allowed an estimation of the time complexity of the adaptive composition/decomposition algorithms which, in the worst case, must perform a high number of flip operations that is a quadratic function in the length of a given string. On the positive side however, this worst-case scenario is unlikely to occur in case of natural, long sequences such as text with many regularities. Since this bound is a function of only the length of a string, implementations can be made very little dependent of the total number of strings in a language. This is in perfect agreement with the general knowledge that the complexity of content addressable memories and secondary key retrieval models and algorithms, in general, does not depend on the number of items stored in such models but mostly on the length of the items. The length of representations is also kept relatively small by the compression achieved through the hierarchical structure of DDAM.

Finally, because nothing precludes the grammar induction process to be performed in the DDAM structure in a dynamic fashion (e.g., immediately after the addition of a new string such as one that would name a hypothetical 13th month), one can infer that the model may have a propensity for dynamic grammar induction. Dynamic grammar induction is, in fact, a form of on-line learning that allows a grammar to be reevaluated as soon as new input data is available.

To our knowledge, the DDAM is the only memory model with a formal description based on generalized combinatorial compositions and Dyck paths. This allows a clear characterization of its structure, function and computational complexity. DDAM also appears to be the only model that unifies a memory-based approach to representation, unsupervised dynamic grammar induction (e.g., text segmentation) and similarity-based retrieval of sequential data. Its biological plausibility rates high and along the similar lines as self-organizing maps and other bottom-up approaches, while its processing performance remains comparable to non-biologically plausible, top-down search methods such as dynamic programming and Viterbi search. As it will be seen in the following section, the model is capable of a complete unsupervised processing of natural language strings while
completely circumventing the need for lemmatization or other preprocessing procedures.

4. Experimental results

In this section, a selection of experimental results will be presented in order to illustrate some of the information processing capabilities of the DDAM model. All experiments have been conducted by the author and documented in [3]. In addition, the experiments on similarity retrieval are particularly significant in that they demonstrate a level of functionality unlike any of the competing information-processing models reviewed in the literature section.

4.1. Unsupervised grammar induction

The data used in some of the following DDAM experiments was the MedTest corpus, a collection of 75 queries by 2344 documents used for evaluation of information retrieval systems [57]. Each MedTest document contains an abstract and metadata (title, authors, journal, and MeSH index terms). The collection was originally created for the evaluation of the MEDLINE system in clinical settings, and was later adapted for the evaluation of retrieval systems in biomedicine.

The evaluation of unsupervised models for information processing in general and for grammar induction in particular is notoriously difficult due to the lack of a gold standard that defines what a good segmentation or a good grammar must look like. Some criteria for the evaluation of the unsupervised medical language processing capabilities of the DDAM model have only been possible in the case of the segmentation of compound biomedical terms and in the case of the text segmentation of biomedical literature abstracts where separators have been artificially removed.

4.2. Unsupervised morpho-segmentation of compound medical terms

Due to the need for high accuracy, most morpho-segmentation approaches for medical language processing are supervised and...
4.3. Unsupervised medical text segmentation

The unsupervised segmentation of medical text works on similar principles as the text segmentation for general language. The MedTest collection was preprocessed so that all separators (blanks, punctuation, parentheses, etc.) were completely removed. Out of the 2344 documents, one document was selected for evaluating the per-segment precision and retrieval.

The results in Table 4 are comparable to that of general text segmentation in terms of precision but show a lower recall due to the existence of many algorithmic regularities (i.e., patterns), characteristic to professional discourses and that have not been segmented (e.g., compared with, efficacious, patients receiving, advantageous, asside effects, randomized trial, the dose, plessthan0005). The precision and recall evaluation of word level segments penalizes the discovery of such regularities. However, the potential significance to automated indexing of such with low frequency patterns could allow them to be employed in IR measures based on inverse frequency (i.e., the rarer a term the higher its relevance to a document).

4.4. Unsupervised lexical equivalence set induction

The unsupervised induction of lexical equivalence sets is another example of unsupervised text segmentation and grammar induction. The evaluation approach is based solely the qualitative assessment of the interestingness of the results. The induction process is based on contextual (prefix, suffix, diafix) similarities and results in equivalence sets that may correspond to lexicosemantic classes. The analysis of the MedTest collection has yielded a grammar with over 85,000 rules. Metadata such as author names, journal names and MESH index terms were not processed. Though some equivalence sets are of limited use due to their abstract nature or to multiple spurious recalls, others may be useful collections of lexical items such as a set of pathological entities (Table 5).

Other lexical equivalence sets bring together words that simply share prefix contexts such as blood_−_ (e.g., bank, centers, clots, components, flow, gas, gases, loss, pressure, pressures, products, supply, transfusion, transfusions, vessels, volume) or in some cases, lexico-semantic properties. A good examples for the latter is the lexical equivalence set of various age groups (i.e., boy, female, girl, male, man, woman) that was induced based on the detection of their common context as the prefix year−old−. Other interesting induced sets of interest collect various types of medical care (e.g., coronary, critical, extended, health, intensive, medical, patient, primary, supportive), types of meningitis (e.g., viral, bacterial, chronic, tuberculous, with listeria monocytogenes, purulent), glomerulonephritis (e.g., proliferative, membranous, necrotizing, proliferative), cancer (e.g., breast, cervical, lung, metastatic, national institute, ovarian) or various types of therapies and chemotherapies (e.g., non-cross-resistant, adjuvant, cancer, combination, initial, maintenance) that are present in the MedTest collection.

While this exercise may appear trivial and being an unsupervised task may lack clear evaluation methodologies, the capability to discover regularities in data in a completely unsupervised fashion is interesting. Its importance is also demonstrated by the acquisition of an equivalence set whose algorithmic significance has exceeded all expectations. One of the common contexts for this equivalence set was the entire paragraph “patients with short duration of disease were especially prone to be antibody
negative in serum but positive in cfs. significant rise in serum antibody titers was seldom demonstrated in patients treated with". The sheer length of this regularity implies that its frequency to appear by chance in two distinct documents is extremely low. Its occurrence was therefore an extraordinary event, and highly indicative of one, virtually unequivocal scenario: the text must have been copied from one document into the other in some way. A simple search on this now known regularity in the MedTest collection turned out two distinct abstracts with the identification number 803 and 1972, respectively. The inspection of the metadata revealed that the documents have been written by the exact same authors (i.e., Stiernstedt, G., Granstrom M., Hederstedt, B., Skoldenberg, B.) but published in different years (i.e., 1985 and 1987) in two different journals (i.e., J Clin Microbiol, 21(5):819–25 and Zentralbl Bakteriol Mikrobiol Hyg; 263(3):420–4). Though this particular discovery is of limited usefulness in hindsight, the conclusions and analogies that can be drawn from this experiment are important and extend beyond the obvious application to discovering such regularities in document collections. Firstly, achieving the same feat with a trivial approach that searches the entire MedTest collection for all substrings up to length \( n \) from the collection itself would have taken very long time compared to the couple of minutes in the DDAM model. Secondly, the potential to discover significant (i.e., long) regularities, in various circumstances, may be a useful tool for decision-making, particularly in time constrained conditions, where filtering quickly what is significant from what is not, is of importance. Though human capabilities in this area would also work at higher semantic levels (e.g., paraphrasing) and patterns may possibly be multi-sensorial, the basic pattern discovery principles could be similar to those in this experiment.

### 4.6. Morphological similarity retrieval

A dataset of 1700 compound medical terms gleaned from the ICD10 and MedTest data sources was represented in the DDAM model. The model was subsequently queried with the term *tachycardia* which was among the terms in the original dataset. The retrieved strings, shown in Table 6, are placed in columns corresponding to concentric hyperspheres with bit radii ranging from 0.0 bits (a very close match) to 2.0 bits (more dissimilar matches shown). The remaining terms (up to 1700) may still possess similarities to the query but of a lower significance that makes them retrievable at bit radii higher than the 2.0 bits threshold.

*Given that the query is the center of the concentric hyperspheres, the terms that begin with tachy- and end in -cardia are expected within a low bit radius. However, at higher bit radii, the associative properties of DDAM led to recalls such as those that begin with brachy- and brady- due to the strong association with bradycardia. The associative properties of DDAM and the sparseness of the representation space are also responsible for spurious recalls such as psittacosis, which has clear similarities with anthracosis. Further, at 1.6 bits, entries containing the patterns -cardial, -cardiac, and -carpal are retrieved, followed by more distant ones such as*
Table 8
The results of the DDAM associative recall on the query “shigella” on the complete collection of about 30,000 ICD10 strings, using an increasingly large bit radius, from 0.0 bit to 3.6 bit.

<table>
<thead>
<tr>
<th>Token radius (bits)</th>
<th>String radius (bits)</th>
<th>Category, ICD10 code</th>
<th>ICD10 string</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>shigella</td>
<td>A03.0</td>
<td>shigellosis due to <em>shigella</em> dysenteriae</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.2</td>
<td>shigellosis due to <em>shigella</em> boydii</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.3</td>
<td>shigellosis due to <em>shigella</em> sonnei</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.1</td>
<td>shigellosis due to <em>shigella</em> flexneri</td>
</tr>
<tr>
<td>0.0</td>
<td>shigellosis</td>
<td>A03.0</td>
<td>group a <em>shigellosis</em> [shiga-kruse dysentery]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.9</td>
<td><em>shigellosis</em> unspecified</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.1</td>
<td><em>shigellosis</em> due to <em>shigella</em> flexneri</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.0</td>
<td><em>shigellosis</em> due to <em>shigella</em> dysenteriae</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.3</td>
<td><em>shigellosis</em> due to <em>shigella</em> sonnei</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.2</td>
<td><em>shigellosis</em> due to <em>shigella</em> boydii</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.1</td>
<td>group b <em>shigellosis</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.3</td>
<td>group d <em>shigellosis</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.8</td>
<td>other <em>shigellosis</em></td>
</tr>
<tr>
<td>1.0</td>
<td>shiga</td>
<td>A03.0</td>
<td>group a <em>shigellosis</em> [shiga-kruse dysentery]</td>
</tr>
<tr>
<td>1.0</td>
<td>kruse</td>
<td>A03.0</td>
<td>group a <em>shigellosis</em> [shiga-kruse dysentery]</td>
</tr>
<tr>
<td>1.0</td>
<td>dysentery</td>
<td>A03.0</td>
<td>group a <em>shigellosis</em> [shiga-kruse dysentery]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.9</td>
<td>bacillary <em>dysentery</em> nos</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A06.0</td>
<td>acute amoebic <em>dysentery</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A07.0</td>
<td>balantidial <em>dysentery</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A07.9</td>
<td>protozoal <em>dysentery</em></td>
</tr>
<tr>
<td>1.0</td>
<td>group</td>
<td>A03.0</td>
<td>group a <em>shigellosis</em> [shiga-kruse dysentery]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T80.3</td>
<td>reaction to blood-group incompatibility in infusion and transfusion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B95.0</td>
<td>streptococcus, group a, as the cause of diseases classified to other chapters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E78.0</td>
<td>hyperlipidaemia, group a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.2</td>
<td>group c <em>shigellosis</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.1</td>
<td>group b <em>shigellosis</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A03.3</td>
<td>group d <em>shigellosis</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>F91.2</td>
<td><em>group</em> delinquency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z63.9</td>
<td>problem related to primary support <em>group</em>, unspecified</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z63.8</td>
<td>other specified problems related to primary support <em>group</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z63</td>
<td>other problems related to primary support <em>group</em>, including family circumstances</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A40.0</td>
<td>septicemia due to streptococcus, group a</td>
</tr>
<tr>
<td>2.0</td>
<td>dysenteriae</td>
<td>A03.0</td>
<td>shigellosis due to <em>shigella</em> <em>dysenteriae</em></td>
</tr>
<tr>
<td>2.0</td>
<td>boydii</td>
<td>A03.2</td>
<td>shigellosis due to <em>shigella</em> <em>boydii</em></td>
</tr>
<tr>
<td>2.0</td>
<td>sonnei</td>
<td>B48.2</td>
<td>infection due to pseudallescheria <em>boydii</em></td>
</tr>
<tr>
<td>2.0</td>
<td>flexneri</td>
<td>A03.1</td>
<td>shigellosis due to <em>shigella</em> <em>flexneri</em></td>
</tr>
<tr>
<td>2.6</td>
<td>reaction</td>
<td>T80.3</td>
<td>reaction to blood-group incompatibility in infusion and transfusion</td>
</tr>
<tr>
<td>2.6</td>
<td></td>
<td>T80.4</td>
<td>reaction due to rh factor in infusion and transfusion</td>
</tr>
<tr>
<td>2.6</td>
<td></td>
<td>D50.0</td>
<td>iron deficiency anemia secondary to blood loss (chronic)</td>
</tr>
<tr>
<td>3.0</td>
<td>pseudallescheria</td>
<td>B48.2</td>
<td>infection due to <em>pseudallescheria</em> <em>boydii</em></td>
</tr>
<tr>
<td>3.3</td>
<td>delinquency</td>
<td>F91.2</td>
<td><em>group</em> delinquency</td>
</tr>
<tr>
<td>3.3</td>
<td>balantidial</td>
<td>A07.0</td>
<td><em>balantidial</em> <em>dysentery</em></td>
</tr>
<tr>
<td>3.3</td>
<td>bacillary</td>
<td>A03.9</td>
<td><em>bacillary</em> <em>dysentery</em> nos</td>
</tr>
<tr>
<td>3.6</td>
<td>custody</td>
<td>Z65.3</td>
<td>child <em>custody</em> or support proceedings</td>
</tr>
</tbody>
</table>
metacholine, hyperdense and lordosis, the latter most likely being the result of the association with anthracosis and psitacosis. At 2.0 bits, though some entries share similarities with the original query, a substantial number of them are the results of indirect associations (e.g., those that end in -sis). Though similarity-based retrieval is an interesting functionality, it has also quickly became clear that this approach is bound to yield results sets that contain spurious hits caused by the combination of associative recall and the potentially significant sparseness of the pattern space. However, it was determined that the validity of a model that is “bound to make mistakes” is also defendable [3]. The main argument was based on a demonstration in subsequent experiments proving that the addition of relevant additional information into DDAM causes the problematic associations and spurious results to be pushed at higher retrieval radii and therefore effectively eliminated from the recall results at lower bit radii. This demonstrated the possibility that similarity retrieval can be improved incrementally by continuously adding relevant information into the associative memory. This behavior, essentially a form of dynamic (or on-line) unsupervised learning, is in perfect agreement with the functional principles of CBR and of significant importance to building intelligent information processing systems capable of learning and that are both usable and useful.

A second line of reasoning in favor of this kind of similarity retrieval is based on the commonsense observation that associative recall never to happen in a vacuum but usually in a rich context. The full integration of contextual sources at phonological, morphological, lexical, syntactic, semantic and pragmatic levels, could provide the basis of a robust mechanism for eliminating spurious retrieval hits. Though such integration is unattainable currently and its effectiveness is far more difficult to prove than in the previous example, it is very tempting to conjecture that integrating information from different levels (i.e., increasing the amount of context) is only a matter of scale that leaves processing principles and mechanisms largely the same. Empirical evidence for this second argument could be built by stepping up a linguistic level and attempting similarity-based search and retrieval based not only on morphological similarity but also on lexical similarities between textual elements. A glimpse of this functionality is offered in the discussion of the next experiment.

4.7. Morpholexical similarity retrieval

For this experiment, the entire collection of about 30,000 strings in the International Classification of Diseases, revision 10 (ICD10) [59] was represented in a DDAM model with two layers: a morphological layer represents the tokens and a layer that represents the entire string collection. The usual rule-based linguistic normalization steps (e.g., lemmatization) are unnecessary in this model-free, unsupervised approach to representation. The query (i.e., the center of the pattern space) was set to the string “shigella” which is the name of a micro-organism responsible for a group of human infectious diseases. In ICD10, the entries that refer to these diseases fall under the code A03, Shigellosis, and are shown in Table 7.

Upon querying the DDAM model with the string “shigella” a series of tokens was retrieved within a 3.6 bit radius. In turn, the tokens have caused the retrieval of various ICD10 strings within bit radii ranging from 1.0 to 3.6 bits. A natural way to display the retrieval is as a multiple hierarchy where tokens form the first level and the ICD10 strings form the second level (Table 8) of the hierarchy. Because the hierarchy is multiple, the ICD10 strings may repeat at any level. Expectedly, all entries in Table 7 can be found among those retrieved within 0.0 bit radius in Table 8. Interestingly, this associative approach to similarity retrieval has also yielded some additional strings that do not contain the original query as such.

One example is “bacillary dysentery NOS”, which is retrieved within a bit radius of 2.6 bits, through an indirect association (i.e., through dysentery/dysenteriae) with the original query. Other strings, also retrieved by indirect association, denote additional types of dysentery (i.e., acute amoebic, balantidial, protozoal) and were retrieved within a 3.3 bit radius. Another indirect association example is the entry “B48.2, infection due to pseudallescheria boydii” retrieved due to its similarity to “A03.2, shigellosis due to shigella boydii”. However, retrieval sets at a bit radius higher than 3.3 bits contain some spurious results such as those containing the token group that is associated with words such as “support” and “reaction”. The sparseness in the dataset causing such spurious associations is therefore the main reason why entries such as “Z65.3, child custody or support proceedings”, “F91.2, group delinquency”, “T80.3, reaction to blood-group incompatibility in infusion and transfusion”, etc. are part of the result set. Preventing their retrieval would probably require the representation of additional data (e.g., relevant to the concept of “group”) or the integration of additional knowledge sources. It is conceivable that additional context in form of semantic and pragmatic knowledge could provide a mechanism to determine the lack of relevance of the entries about child support and group delinquency to a set of items whose common theme is infectious diseases.

The results of this similarity retrieval experiment can also be displayed using a graphical paradigm. Though individual items may not be as clearly visible as in the multiple hierarchy in Table 8, the graphical display of results, in conjunction with force-directed automatic layout optimization algorithms, appears to attain the functional and structural equivalence of clustering approaches such as self-organizing maps. For example, in the resulting maps in Fig. 4, three distinct clusters are visible.

The three clusters correspond clearly to three “themes” that characterize the items in the retrieval set:

- A middle cluster corresponding to a main theme that has to do with various micro-organisms and infectious diseases; this cluster shows some additional partial strings such as “due to streptococcus group a”
- A top cluster which seems to be centered on the phrase support group, and
- A bottom cluster containing tokens such as reaction, anemia and which appears to be responsible for ICD10 entries such as reaction to blood-group incompatibility in infusion and transfusion, reaction due to rh factor in infusion and transfusion, and iron deficiency anemia secondary to blood loss (chronic) in the results set.

The fact that the clusters are very distinct suggests that additional sophistications of the algorithm to improve the relevance of the similarity-based retrieval are possible, in addition to simply restricting the recall bit radius or to increasing the amount of data represented in the DDAM model.

5. Conclusions and future work

The deterministic dynamic associative memory (DDAM) is a new memory model with a relatively simple formal description that employs existing mathematical concepts such as the Dyck path and the combinatorial integer composition. The model also incorporates algorithms for string composition and decomposition and possesses capabilities for grammar induction with applications to general sequence processing and text segmentation. Sometimes, evaluating unsupervised information-processing models, particularly in tasks involving natural data (e.g., text), is not always straightforward due to a lack of clear evaluation criteria and to difficulties related to creating comparable experimental
setups for existing models. However, the element of surprise in results associated with the discovery of new patterns and significant regularities allows for an evaluation based on interestingness of results.

The DDAM model is applicable to a broad range of unsupervised information processing tasks. When possible, the direct comparison with existing unsupervised sequence processing models in various experiments and with various types of sequences (e.g., artificial, natural) has demonstrated good results consistently [3]. However, DDAM also had instances where it performed worse than search algorithms based on dynamic programming in the case of per-word segmentation tasks on some specific datasets such as the CHILDES corpus. Additional improvements of DDAM followed by comparisons with such dynamic programming algorithms (e.g., Viterbi search) are therefore desirable.

While still subject to ongoing research, DDAM also offers some solutions to the difficult task of similarity retrieval for which no other models reviewed have provided a viable basis for comparison (retrieval was not among their purposes). This is an indication of the degree of generality of DDAM whose virtually unchanged information-processing model has demonstrated consistent performance across a wide range of experimental setups, data sources and processing tasks, unlike other existing approaches considered in this research and which appeared to focus only on certain types of information processing tasks (e.g., word segmentation, morphosemantic segmentation, data compression).

Future theoretical work on the DDAM model will focus on the development of the theoretical foundations to include a generalized definition of the concept of ambiguity that takes into account the statistics of the patterns discovered and represented in DDAM. This will be complemented by an extension of the mathematical framework to include the processing and representation of multidimensional data signals such as 2- and 3-dimensional objects. This important development was already envisioned to employ the formal definition of at least one completely novel combinatorial object, the “Dyck surface.” The notion of a 3-dimensional “Dyck surface” that could be used to represent higher dimensional objects (e.g., images) has, in effect, suggested itself as the natural extension of the 2-dimensional Dyck paths used for DDAM representations of 1-dimensional sequential data.

On the applied research front, future work will aim at the exploration of DDAM capabilities for data compression, in the MDL sense, followed by an evaluation in data compression benchmarks. While it is not expected that DDAM yields the highest compression ratios, it is hoped that the results will be comparable to those of other data compression algorithms. In the same vein, currently there is a high interest in improving the DDAM implementation in order to achieve a parallel and distributed prototype that can leverage recent advances in grid computing and improve the scalability of the information processing applications beyond the representation of text collections with only a few thousands documents.

Past work on DDAM has determined a rich potential for applied and theoretical research on Case-based Reasoning (CBR) [3]. Of particular interest is the development of a robust algorithmic similarity measure and retrieval algorithm followed by applications to medical CBR. Of future interest in this area is also the investigation of a possible computational approach to “creativity” or “inventiveness” that was already identified conceptually in [3] and which may have applications to the difficult task of case adaptation in CBR.

Fig. 4. Image (detail left, zoom-out on the right) of the results of the DDAM associative recall on the query “shigella” on the complete collection of about 30,000 ICD10 strings showing three distinct clusters; the image shows all nodes residing within the 3.6 bit recall radius from the query and was created using a force-directed automatic layout optimization technique that renders it similar to the concept of a self-organizing map.
Acknowledgment
I acknowledge Dr. Jochen Moehr for his continuous support, contribution and mentorship that have made this research possible.

References