On the Uncertainty of Information Retrieval in Associative Memories

Eitan Yaakobi[*†] and **Jehoshua Bruck**[*]
[*]Electrical Engineering Department, California Institute of Technology, Pasadena, CA 91125, U.S.A
[†]Electrical and Computer Engineering Department, University of California San Diego, La Jolla, CA 92093, U.S.A.
{*yaakobi, bruck*}*@caltech.edu*

***Abstract*—We (people) are memory machines. Our decision processes, emotions and interactions with the world around us are based on and driven by associations to our memories. This natural association paradigm will become critical in future memory systems, namely, the key question will not be "How do I store more information?" but rather, "Do I have the relevant information? How do I retrieve it?"**

The focus of this paper is to make a first step in this direction. We define and solve a very basic problem in associative retrieval. Given a word W, the words in the memory that are t-associated with W are the words in the ball of radius t around W. In general, given a set of words, say W, X and Y, the words that are t-associated with $\{W, X, Y\}$ are those in the memory that are within distance t *from all the three words.* Our main goal is to study the maximum size of the t-associated set as a function of the number of input words and the minimum distance of the words in memory - we call this value *the uncertainty of an associative memory*. We derive the uncertainty of the associative memory that consists of all the binary vectors with an arbitrary number of input words. In addition, we study the retrieval problem, namely, how do we get the t-associated set given the inputs? We note that this paradigm is a generalization of the sequences reconstruction problem that was proposed by Levenshtein (2001). In this model, a word is transmitted over multiple channels. A decoder receives all the channel outputs and decodes the transmitted word. Levenshtein computed the minimum number of channels that guarantee a successful decoder - this value happens to be the uncertainty of an associative memory with two input words.

I. Introduction

One of the interpretations of the term *association*, especially in the context of psychology, is the connection between two or more concepts. Throughout our life, we remember and store an enormous amount of information. However, while we are not aware of the method this information is stored, amazingly, it can be accessed and retrieved with relative ease. The way we think and process information is mainly performed by associations. Our memory content retrieval process is done by stimulating it with an external or internal inputs. That is, the knowledge of some information gives rise to related information that was stored earlier. Mathematically speaking, assume the memory is a set of words $\mathcal{M} = \{m_1, \ldots, m_N\}$. Then, given an arbitrary word $\boldsymbol{x}$ as an input to the memory M, its output is another word or a set of words from the memory M that are related or close to the input word $\boldsymbol{x}$. Here, the term "close" can be interpreted as using any distance metric between words, for example the Hamming distance. A word is associated with another word or words which they again can be associated with more words and so on, resulting in a sequence of associations.

From the information theory perspective, we say that the words associated with an input word $\boldsymbol{x}$ are those in distance at most t (a prescribed value) from $\boldsymbol{x}$. This set comprises a ball of radius t, see Fig. 1(a). We generalize this paradigm and consider a *set of words* that are presented as an input to the memory. For example, for two input words $\boldsymbol{x}, \boldsymbol{y}$, their set of associated words are the ones that their distance from both $\boldsymbol{x}$ and $\boldsymbol{y}$ is at most t, see Fig. 1(b). Clearly, this set of associations is strictly smaller then the ball of radius t. For more than two input words, the set of associated words is getting even smaller, and in general, the larger the set of input words is, the smaller the set of associated words is, see Fig. 1(c). For example, assume the input word is "tall", then many words can be associated with it, such as "tree", "mountain", "tower", "ladder", etc. But if the input words are "tall" and "fruit", then out of these four associated words, only the word "tree" will be associated with "tall" and "fruit".

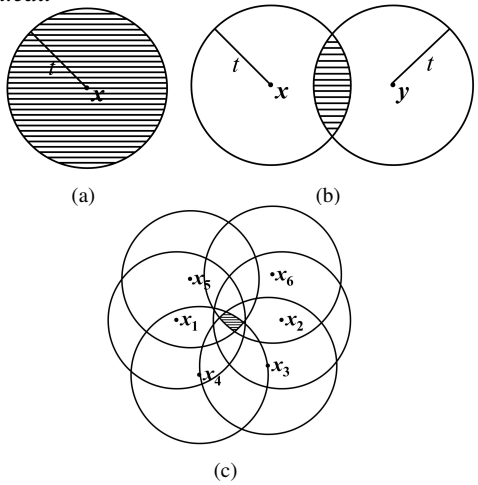

Fig. 1. Three example of associated words: (a) associated words with a single word $\boldsymbol{x}$, (b) associated words with two words $\boldsymbol{x}$ and $\boldsymbol{y}$, (c) associated words with the six words $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_6$.

Assume that the memory is the set of all binary vectors, $\mathcal{M} = \{0,1\}^n$. For any input word $\boldsymbol{x}$, it is immediate to see that if its set of associated words are the ones of distance at most t, then there are $\sum_{i=0}^{t} \binom{n}{i}$ such words. However, for two input words, $\boldsymbol{x}, \boldsymbol{y}$, the problem of finding the set of associated words of distance at most t from both $\boldsymbol{x}$ and $\boldsymbol{y}$ becomes more complex. In fact, this problem was proposed and solved by Levenshtein in [10], [11]. The motivation came from a completely different scenario in the context of the *sequences reconstruction problem*. In this model, a codeword $\boldsymbol{x}$ is transmitted through multiple channels. Then, a decoder receives all channel outputs and generates an estimation on the transmitted word, while it is guaranteed that all channel outputs are different from each other, see Fig. 2. If $\boldsymbol{x}$ belongs to a code $\mathcal{C}$ with minimum distance d and in every channel there can be at most $t > \lfloor \frac{d-1}{2} \rfloor$ errors, then Levenshtein studied the minimum number of channels N that guarantees the existence of

978-1-4673-2579-0/12/$31.00 ©2012 IEEE

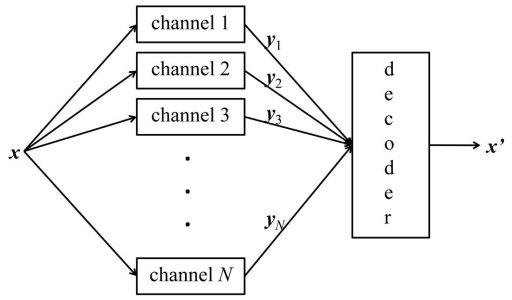

Fig. 2. Channel model of the sequences reconstruction problem.

a successful decoder. This number has to be greater than

$$N = \max_{\boldsymbol{x}_1,\boldsymbol{x}_2\in\mathcal{C},\boldsymbol{x}_1\neq\boldsymbol{x}_2} |B_t(\boldsymbol{x}_1)\cap B_t(\boldsymbol{x}_2)|, \quad (1)$$

where $B_t(\boldsymbol{x})$ is the ball of radius t surrounding $\boldsymbol{x}$. To see that, notice that if the intersection of the radius-t balls of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ contains N words and the channel outputs are these N words, then a decoder cannot determine what the transmitted word is. However, if the number of channel outputs is greater than the maximum size of the intersection of two balls, then there is only one codeword of distance at most t from all received channel outputs.

The motivation to the model studied by Levenshtein came from fields such as chemistry and biology, where the redundancy in the codewords is not sufficient to construct a successful decoder. Thus, the only way to combat errors is by repeatedly transmitting the same codeword. Recently, this model was shown to be also relevant in storage technologies [3], [16]. Due to the high capacity and density of today's and future's storage medium, it is no longer possible to read individual memory elements, but, rather, only a multiple of them at once. Hence, every memory element is read multiple times, which is translated into multiple estimations of the same information stored in the memory.

Finding the maximum intersection problem in (1) was studied in [11] with respect to the Hamming distance and other metric distances. In [7]–[9], it was analyzed over permutations, and in [13], [14] for error graphs. In [12], the equivalent problem for insertions and deletions was studied and reconstruction algorithms for this model were given in [2], [5], [15]. The case of deletions only was studied in the context of trace reconstruction in [4].

Returning to our original problem, the set of associated words with $\boldsymbol{x}$ and $\boldsymbol{y}$ is $B_t(\boldsymbol{x})\cap B_t(\boldsymbol{y})$ and the maximum intersection is the value N in (1). The generalized problem of finding the maximum size of associated words of $m \geqslant 2$ input words with mutual distance d is expressed as

$$N_t(m,d) = \max_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m,d_H(\boldsymbol{x}_i,\boldsymbol{x}_j)\geqslant d} \{|\cap_{i=1}^m B_t(\boldsymbol{x}_i)|\}. \quad (2)$$

The main goal in this paper it to analyze the value of $N_t(m,d)$ with respect to the Hamming distance for different values of t,m,d. In particular, we show that if $A(D)$ is the size of a maximal anitcode of diameter D, that is, the largest set of words with maximum distance D, then $N_t(A(D),1) = A(2t-D)$.

The rest of the paper is organized as follows. In Section II, we define the concept of associative memories and describe the connection to the sequences reconstruction problem. In Section III, we solve the problem stated in (2) for the case $d = 1$. Extensions for arbitrary d are given in Section IV. In Section V, we give efficient decoders to the reconstruction problem studied by Levenshtein. Finally, Section VI concludes the paper.

II. Definitions and Basic Properties

In this work, the words are binary vectors of length n. The Hamming distance between two words $\boldsymbol{x}$ and $\boldsymbol{y}$ is denoted by $d_H(\boldsymbol{x},\boldsymbol{y})$ and the Hamming weight of a word $\boldsymbol{x}$ is denoted by $w_H(\boldsymbol{x})$. For a word $\boldsymbol{x}\in\{0,1\}^n$, $B_t^n(\boldsymbol{x})$ is its surrounding ball of radius t, $B_t^n(\boldsymbol{x}) = \{\boldsymbol{y}\in\{0,1\}^n \ : \ d_H(\boldsymbol{x},\boldsymbol{y})\leqslant t\}$. The size of $B_t^n(\boldsymbol{x})$, comprising of length-n words, is $b_{t,n} = \sum_{i=0}^t \binom{n}{i}$. If the length of the words is clear from the context, we use the notation $B_t(\boldsymbol{x})$. For $1\leqslant i\leqslant n$, $\boldsymbol{e}_i$ is the unit vector where only its i-th bit is one, and $\mathbf{0}$ is the all-zero vector. Two words $\boldsymbol{x},\boldsymbol{y}\in\{0,1\}^n$ are called *t-associated* if $d_H(\boldsymbol{x},\boldsymbol{y})\leqslant t$.

Definition. *The **t-associated set** of the words $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m$ is denoted by $S_t(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\})$ and is defined to be the set of all words $\boldsymbol{y}$ that are t-associated with $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m$,*

$$S_t\big(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}\big) = \{\boldsymbol{y} : d_H(\boldsymbol{y},\boldsymbol{x}_i)\leqslant t, 1\leqslant i\leqslant m\} = \bigcap_{i=1}^m B_t(\boldsymbol{x}_i).$$

Note that for a single word $\boldsymbol{x}$, we have $S_t\big(\{\boldsymbol{x}\}\big) = B_t(\boldsymbol{x})$. Given an associative memory $\mathcal{M}$, we define the maximum size of a t-associated set of any m words from the memory.

Definition. *Let $\mathcal{M}$ be an associative memory and m,t be two positive integers. The **uncertainty of the associative memory** $\mathcal{M}$ for m and t, denoted by $N_t(m,\mathcal{M})$, is the maximum size of a t-associated set of m different input words from $\mathcal{M}$. That is,*

$$N_t(m,\mathcal{M}) = \max_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\in\mathcal{M},\boldsymbol{x}_i\neq\boldsymbol{x}_j} \big\{\big|S_t\big(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}\big)\big|\big\}. \quad (3)$$

In case the associative memory $\mathcal{M}$ is a code with minimum distance d, we will use the notation $N_t(m,d)$ instead of $N_t(m,\mathcal{M})$. Then, the value in Equation (3) becomes

$$N_t(m,d) = \max_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m,d_H(\boldsymbol{x}_i,\boldsymbol{x}_j)\geqslant d} \big\{\big|S_t\big(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}\big)\big|\big\}. \quad (4)$$

For example, $N_t(m,1)$ refers to $N_t(m,\{0,1\}^n)$.

We now give the definitions that establish the connection with the channel model by Levenshtein [11]. Assume a codeword $\boldsymbol{x}$ is transmitted over N channels. The channel outputs, denoted by $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N$, are all different from each other (Fig. 2). A *list decoder* $\mathcal{D}_{\mathcal{L}}$ receives the N channel outputs and returns a list of at most $\mathcal{L}$ words $\widehat{\boldsymbol{x}}_1,\ldots,\widehat{\boldsymbol{x}}_\ell$, where $\ell\leqslant\mathcal{L}$. We call it an *$\mathcal{L}$-decoder* $\mathcal{D}_{\mathcal{L}}$. The $\mathcal{L}$-decoder $\mathcal{D}_{\mathcal{L}}$ is said to be *successful* if and only if the transmitted word $\boldsymbol{x}$ belongs to the decoded output list, i.e.,

$$\boldsymbol{x}\in\mathcal{D}_{\mathcal{L}}(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N) = \{\widehat{\boldsymbol{x}}_1,\ldots,\widehat{\boldsymbol{x}}_\ell\}.$$

In case $\mathcal{L}=1$ then the decoder output is a single word and this is the model studied by Levenshtein [11].

The next Lemma shows the connection between the value of $N_t(m,d)$ and the decoding success of an $\mathcal{L}$-decoder.

Lemma 1. *Assume the transmitted word $\boldsymbol{x}$ belongs to a code $\mathcal{C}$ of minimum distance d. Then, there exists a successful $\mathcal{L}$-decoder with N channels if and only if*

$$N\geqslant N_t(\mathcal{L}+1,d)+1.$$

Proof: Assume to the contrary that the number of channels is $N_t(\mathcal{L}+1,d)$ and let $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{\mathcal{L}+1}$ be $\mathcal{L}+1$ words

such that the set $S_t(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{\mathcal{L}+1}\})$ contains $N_t(\mathcal{L}+1,d)$ words. If one of these $\mathcal{L}+1$ words is the transmitted one and the received channel outputs are the $N_t(\mathcal{L}+1,d)$ words in $S_t(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{\mathcal{L}+1}\})$, then any of the $\mathcal{L}+1$ words $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{\mathcal{L}+1}$ could be the transmitted one and thus can belong to the decoder's output list. Hence, the transmitted word may not belong to the output list.

On the other hand, if there are $N_t(\mathcal{L}+1,d)+1$ channels, then for any transmitted word $\boldsymbol{x}$, there are at most $\mathcal{L}$ words in $\mathcal{C}$, all of distance at least d from each other, such that the $N_t(\mathcal{L}+1,d)+1$ channel outputs are located in the intersection of their radius-t balls. ∎

For $\mathcal{L}=1$, the value $N_t(2,d)$ was studied by Levenshtein [11] and was shown to be

$$N_t(2,d)=\sum_{i=0}^{t-\lceil\frac{d}{2}\rceil}\binom{n-d}{i}\sum_{k=d-t+i}^{t-i}\binom{d}{k}. \tag{5}$$

Let us first remind how this value was calculated. Assume $d_H(\boldsymbol{x},\boldsymbol{y})=d$ and the goal is to find the cardinality of the set

$$S_t(\{\boldsymbol{x},\boldsymbol{y}\})=\{\boldsymbol{z}\in\{0,1\}^n\ :\ d_H(\boldsymbol{z},\boldsymbol{x}),d_H(\boldsymbol{z},\boldsymbol{y})\leqslant t\}.$$

For any word $\boldsymbol{z}\in S_t(\{\boldsymbol{x},\boldsymbol{y}\})$, let $S_{0,0},S_{0,1},S_{1,0},S_{1,1}$ be the following four sets:

$$S_{0,0}=\{i:y_i=z_i=x_i\},\qquad S_{0,1}=\{i:y_i=x_i,z_i=\overline{x}_i\},$$
$$S_{1,0}=\{i:y_i=\overline{x}_i,z_i=x_i\},\quad S_{1,1}=\{i:y_i=z_i=\overline{x}_i\}.$$

Note that $|S_{0,0}|+|S_{0,1}|=n-d$ and $|S_{1,0}|+|S_{1,1}|=d$. Since $d_H(\boldsymbol{z},\boldsymbol{x})\leqslant t$ and $d_H(\boldsymbol{z},\boldsymbol{y})\leqslant t$ we get that

$$|S_{0,1}|+|S_{1,1}|\leqslant t,\quad |S_{0,1}|+|S_{1,0}|\leqslant t,$$

or

$$|S_{0,1}|+|S_{1,1}|\leqslant t,\quad |S_{0,1}|+d-|S_{1,1}|\leqslant t.$$

Denote $|S_{0,1}|=i$ and $|S_{1,1}|=k$ so we get

$$i+k\leqslant t,\quad i+d-k\leqslant t,$$

or

$$0\leqslant i\leqslant t-\lceil d/2\rceil,\quad i+d-t\leqslant k\leqslant t-i.$$

Therefore, the number of words in the intersection of these two spheres is given by

$$|S_t(\{\boldsymbol{x},\boldsymbol{y}\})|=N_t(2,d)=\sum_{i=0}^{t-\lceil d/2\rceil}\binom{n-d}{i}\sum_{k=i+d-t}^{t-i}\binom{d}{k}.$$

where $\binom{a}{b}=0$ if $b<0$ or $b>a$.

If we substitute the order of i,k in the last term, we get $0\leqslant k\leqslant\min\{d,t\}, 0\leqslant i\leqslant t-\max\{k,d-k\}$, and

$$|S_t(\{\boldsymbol{x},\boldsymbol{y}\})|=N_t(2,d)=\sum_{k=0}^{\min\{d,t\}}\binom{d}{k}\sum_{i=0}^{t-\max\{k,d-k\}}\binom{n-d}{i}. \tag{6}$$

This last representation of $N_t(2,d)$ will be helpful in showing the following property. Due to the lack of space, we omit the proof of the next lemma as well as the following one, which extends the solution of $N_t(2,d)$ for $m=3$.

Lemma 2. *Let t,d be two positive integers such that d is even, then*

$$N_t(2,d)=N_t(2,d-1).$$

Lemma 3. *Let t,d be such that $t>\lceil\frac{d-1}{2}\rceil$, and n large enough. The value of $N_t(3,d)$ is given by*

$$N_t(3,d)=\sum_{i_1,i_2,i_3,i_4}\binom{n-\lceil\frac{3d}{2}\rceil}{i_1}\binom{\lceil\frac{d}{2}\rceil}{i_2}\binom{\lceil\frac{d}{2}\rceil}{i_3}\binom{\lfloor\frac{d}{2}\rfloor}{i_4},$$

where i_1,i_2,i_3,i_4 satisfy the following constraints:

1) $0\leqslant i_1\leqslant t-\lceil\frac{d}{2}\rceil$,
2) $i_1+\lfloor\frac{d}{2}\rfloor-t\leqslant i_4\leqslant t-\lceil\frac{d}{2}\rceil-i_1$,
3) $d-t+i_1\leqslant i_3\leqslant t-(i_1+i_4)$,
4) $\max\{i_1-i_3-i_4+\lceil\frac{3d}{2}\rceil-t, i_1+i_3+i_4+\lceil\frac{d}{2}\rceil-t\}\leqslant i_2\leqslant t-(i_1+i_4+\lceil\frac{d}{2}\rceil-i_3)$.

III. The Case $d=1$

In this section, we analyze the value of $N_t(m,d)$ for $d=1$. A first observation on the value of $N_t(m,1)$ is stated in the next lemma.

Lemma 4. *For $m,t\geqslant 1$, if $N_t(m,1)\geqslant\ell$ and $N_t(m+1,1)<\ell$, then $N_t(\ell,1)=m$.*

Proof: Since $N_t(m,1)\geqslant\ell$, there exist m different words $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m$ such that

$$|S_t(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\})|=|B_t(\boldsymbol{x}_1)\cap\cdots\cap B_t(\boldsymbol{x}_m)|\geqslant\ell$$

and assume $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_\ell$ are ℓ words which belong to this intersection. Therefore, $d_H(\boldsymbol{x}_i,\boldsymbol{y}_j)\leqslant t$ for all $1\leqslant i\leqslant m$ and $1\leqslant j\leqslant\ell$, and thus

$$\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\}\subseteq S_t(\{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_\ell\})=B_t(\boldsymbol{y}_1)\cap\cdots\cap B_t(\boldsymbol{y}_\ell),$$

and hence $N_t(\ell,1)\geqslant m$.

Assume to the contrary that $N_t(\ell,1)\geqslant m+1$ and let $\boldsymbol{z}_1,\ldots,\boldsymbol{z}_\ell$ be ℓ words such that

$$|S_t(\{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_\ell\})|=|B_t(\boldsymbol{z}_1)\cap\cdots\cap B_t(\boldsymbol{z}_\ell)|\geqslant m+1.$$

As in the first part, we get that $N_t(m+1,1)\geqslant\ell$, which is a contradiction. Hence, $N_t(\ell,1)=m$. ∎

In general, for a given set of words $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m$, the closer the words are, the larger the size of the set $S_t(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m\})$ is. In case $d=1$, we look for a set of words that are all close to each other, or equivalently - the maximum distance between all pairs of words is minimized.

An *anticode* of *diameter* D is a set $A\subseteq\{0,1\}^n$ of words such that the maximum distance between every two words in A is at most D. That is, for all $x,y\in A$, $d_H(x,y)\leqslant D$. For $D\geqslant 1$, $A(D)$ is the size of the largest anticode of diameter D. It was shown in [6] that the value of $A(D)$ is given by

$$A(D)=\begin{cases}b_{\frac{D}{2},n} & \text{if } D \text{ is even,}\\ 2b_{\frac{D-1}{2},n-1} & \text{if } D \text{ is odd.}\end{cases}$$

Our next goal is to show that for all $D\geqslant 1$,

$$N_t(A(D),1)=A(2t-D).$$

That is, the t-associated set of a maximum anticode of diameter D is a maximum anticode of diameter $2t-D$.

Lemma 5. *For all $0\leqslant D\leqslant 2t\leqslant n$,*

$$N_t(A(D),1)\geqslant A(2t-D).$$

Proof: Assume that D is even. We take the $A(D)$ words in $B_{\frac{D}{2}}(\boldsymbol{0})$ and consider the set

$$S_t\left(B_{\frac{D}{2}}(\boldsymbol{0})\right)=\bigcap_{\boldsymbol{x}\in B_{\frac{D}{2}}(\boldsymbol{0})}B_t(\boldsymbol{x}).$$

Then, $B_{t-\frac{D}{2}}(\boldsymbol{0})\subseteq S_t\left(B_{\frac{D}{2}}(\boldsymbol{0})\right)$ and hence $N_t(A(D),1)\geqslant A(2t-D)$ for even D.

In case that D is odd, let $i=(D-1)/2$. Let us start with a maximal anticode of diameter $2i+1$. Let X be the set

$$X = B_i(\mathbf{0}) \cup B_i(\boldsymbol{e}_1) = \{aw : a \in \{0,1\}, w \in B_i^{n-1}(\mathbf{0})\},$$
and let
$$Y = B_{t-i-1}(\mathbf{0}) \cup B_{t-i-1}(\boldsymbol{e}_1) = \{bu : b \in \{0,1\}, u \in B_{t-i-1}^{n-1}(\mathbf{0})\}.$$
Then, for every $\boldsymbol{x} \in X, \boldsymbol{y} \in Y$, $d_H(\boldsymbol{x}, \boldsymbol{y}) \leqslant t$. Therefore, $N_t(A(D), 1) \geqslant A(2t - D)$ for odd D as well. ∎

The equivalent upper bound is proved in the next two lemmas.

Lemma 6. *For all $0 \leqslant D \leqslant 2t$ and $n \geqslant (t - \frac{D}{2})(2^{D+1} + 1)$, where D is even,*
$$N_t(A(D) + 1, 1) < A(2t - D).$$

Proof: Let $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{A(D)+1}\}$ be a set of $A(D) + 1$ words. Since the largest anticode with diameter D has size $A(D)$, there exist two words, say $\boldsymbol{x}_1, \boldsymbol{x}_2$, where $d_H(\boldsymbol{x}_1, \boldsymbol{x}_2) \geqslant D + 1$. Hence, the size of $S_t(X)$ is no greater than the size of $S_t(\{\boldsymbol{x}_1, \boldsymbol{x}_2\})$, which, according to (5), is at most
$$M = \sum_{j=0}^{t-(\frac{D}{2}+1)} \binom{n-D-1}{j} \sum_{k=D+1-t+j}^{t-j} \binom{D+1}{k}.$$
Note that
$$M < \sum_{j=0}^{t-(\frac{D}{2}+1)} \binom{n}{j} 2^{D+1}.$$
For $0 \leqslant j \leqslant t - (\frac{D}{2} + 1)$ and $(t - \frac{D}{2})(2^{D+1} + 1)$, we have $\binom{n}{j} 2^{D+1} \leqslant \binom{n}{j+1}$ and hence,
$$M < \sum_{j=0}^{t-(\frac{D}{2}+1)} \binom{n}{j+1} < \sum_{j=0}^{t-\frac{D}{2}} \binom{n}{j} = A(2t - D).$$
∎

An equivalent property can be shown for D odd. We skip its details due to its long proof and the lack of space.

Lemma 7. *For all $0 \leqslant D \leqslant 2t$, where D is odd, and n large enough,*
$$N_t(A(D) + 1, 1) < A(2t - D).$$

We summarize this result in the following corollary.

Corollary 8. *For all $0 \leqslant D \leqslant 2t$ and n large enough,*
$$N_t(A(D), 1) = A(2t - D).$$

Proof: From Lemma 5 we get that $N_t(A(D), 1) \geqslant A(2t - D)$ and from Lemma 6 and Lemma 7 $N_t(A(D) + 1, 1) < A(2t - D)$. The conditions of Lemma 4 hold and thus $N_t(A(2t - D), 1) = A(D)$, or $N_t(A(D), 1) = A(2t - D)$. ∎

We note that the result shown by Levenshtein for $d = 1$ is a special case of Corollary 8 for $D = 1$.

IV. Extensions for Arbitrary d

Our goal in this section is to use the results found in Section III in order to derive bounds on $N_t(m, d)$ for arbitrary d. First, we state a useful Theorem from [1].

Theorem 9. [1] *Let $\mathcal{C}$ be a code in the Hamming graph Γ with distances from $\mathcal{D} = \{d_1, \ldots, d_s\} \subseteq \{1, \ldots, n\}$. Further let $L_{\mathcal{D}}(B)$ be a maximal code in $B \subseteq \Gamma$ with distances from $\mathcal{D}$. Then, one has*
$$\frac{|\mathcal{C}|}{|\Gamma|} \leqslant \frac{|L_{\mathcal{D}}(B)|}{|B|}.$$

For all $d \geqslant 1$, we denote by ρ_d to be the maximal rate of a code $\mathcal{C}_d$ of minimum distance d and length n, that is,
$$\rho_d = \max_{\mathcal{C}_d} \left\{ \frac{|\mathcal{C}_d|}{2^n} \right\}.$$
Theorem 9 will serve us to prove the next lemma.

Lemma 10. *Let B be a set and let $L(B)$ be a maximal code in B with minimum distance d, then*
$$|L(B)| \geqslant \rho_d \cdot |B|.$$

Proof: We take the code $\mathcal{C}$ in Theorem 9 to be a code $\mathcal{C}_d$ of minimum distance d and maximal rate ρ_d. Then, we get
$$\frac{|L(B)|}{|B|} \geqslant \frac{|\mathcal{C}_d|}{2^n} = \rho_d.$$
∎

Now, we can derive a connection with $N_t(m, d)$.

Lemma 11. *For all m, t, d and n large enough,*
$$N_t(\lceil \rho_d m \rceil, d) \geqslant N_t(m, 1).$$

Proof: Assume that X is a set of m words such that $S_t(X)$ has size A. According to Lemma 10, let $L(X)$ be a code in X of minimum distance d and size $\lceil \rho_d m \rceil$. Then, $S_t(X) \subseteq S_t(L(X))$ and thus $N_t(\lceil \rho_d m \rceil, d) \geqslant N_t(m, 1)$. ∎

Finally, in case m, t, d are fixed we derive the following.

Lemma 12. *For any fixed m, t, d and n large enough, such that $m \geqslant 3$ and $t \geqslant \lceil \frac{d}{2} \rceil$, $N_t(m, d) = \Theta(n^{t - \lceil \frac{d}{2} \rceil})$.*

Proof: Since $N_t(m, d) \leqslant N_t(2, d)$ and $N_t(2, d) = \Theta(n^{t - \lceil \frac{d}{2} \rceil})$, then $N_t(m, d)$ is at most $O(n^{t - \lceil \frac{d}{2} \rceil})$.

To show the other direction of this equality, we show an example of a set X such that the cardinality of $S_t(X)$ is $O(n^{t - \lceil \frac{d}{2} \rceil})$. Let $\boldsymbol{e}_i^j$ be the vector which its ℓ-th bit is one if and only if $\ell \in \{i, \ldots, j\}$. For $1 \leqslant i \leqslant m$, let $i_0 = (i-1)\lceil \frac{d}{2} \rceil + 1$ and $i_1 = i\lceil \frac{d}{2} \rceil$, and $\boldsymbol{x}_i = \boldsymbol{e}_{i_0}^{i_1}$. Then, for all $i \neq j$, $d_H(\boldsymbol{x}_i, \boldsymbol{x}_j) = 2\lceil \frac{d}{2} \rceil \geqslant 2d$. For any vector $\boldsymbol{y}$ of weight at most $t - \lceil \frac{d}{2} \rceil$, such that its first $m\lceil \frac{d}{2} \rceil$ bits are zero, $\boldsymbol{y} \in S_t(X)$. Since there are $\sum_{\ell=0}^{t - \lceil \frac{d}{2} \rceil} \binom{n - m\lceil \frac{d}{2} \rceil}{\ell}$ such vectors we get that for n large enough, $N_t(m, d)$ is at least $O(n^{t - \lceil \frac{d}{2} \rceil})$. Together we conclude that $N_t(m, d) = \Theta(n^{t - \lceil \frac{d}{2} \rceil})$. ∎

V. Sequences Reconstruction Decoders

The main goal in [11] was to find the necessary and sufficient number of channels in order to have a successful decoder for a code with minimum distance d. This number was studied for different error models in [7]–[14], however the only decoder constructions, which we are aware of, were given in [2], [5], [15] for channels with insertion and deletions, and in [4] for deletions only. In this section, we show how to construct decoders for substitution errors, where the decoder has to output the transmitted word (and not a list of words).

The case $d = 1$ was solved in [11] where the majority algorithm on each bit successfully decodes the transmitted word. According to Lemma 2, this algorithm works for $d = 2$ as well since the number of channels has to be the same. However, if d is greater than two, then the majority algorithm on each bit no longer works. In general, according to Lemma 2, if d is even then the number of channels for a code with minimum distance d or $d - 1$ is the same. Hence, we only need to solve here the case of odd minimum distance.

For the rest of this section, we assume that the transmitted word belongs to a code $\mathcal{C}$ with odd minimum distance d, there are at most $t > \frac{d-1}{2}$ errors in every channel, and the number of channels is $N = N_t(2, d) + 1$. The N channel outputs are denoted by $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$. Furthermore, the code $\mathcal{C}$ has a decoder $\mathcal{D}_{\mathcal{C}}$, which can successfully correct $\frac{d-1}{2}$ errors.

A first observation in constructing a decoder is that we can always detect whether the output word is the transmitted one. This can simply be done by checking if the maximum distance from all channel outputs is at most t.

Lemma 13. *For any $\widehat{\boldsymbol{c}} \in \mathcal{C}$, $\widehat{\boldsymbol{c}} = \boldsymbol{c}$ if and only if*

$$\max_{1 \leqslant i \leqslant N} \{d_H(\widehat{\boldsymbol{c}}, \boldsymbol{y}_i)\} \leqslant t.$$

Proof: If $\widehat{\boldsymbol{c}} = \boldsymbol{c}$ then every channel suffers at most t errors and thus $\max_{1 \leqslant i \leqslant N}\{d_H(\widehat{\boldsymbol{c}}, \boldsymbol{y}_i)\} \leqslant t$. In case $\widehat{\boldsymbol{c}} \neq \boldsymbol{c}$, let us assume to the contrary that $\max_{1 \leqslant i \leqslant N}\{d_H(\widehat{\boldsymbol{c}}, \boldsymbol{y}_i)\} \leqslant t$. Then the set $S_t(\{\widehat{\boldsymbol{c}}, \boldsymbol{c}\})$ contains at least $N = N_t(2, d) + 1$ words in contradiction to the definition of $N_t(2, d)$. ∎

A naive algorithm can choose any of the channel outputs and add all error vectors of weight at most $t - \frac{d-1}{2}$. For one of these error vectors we will get a word with at most $\frac{d-1}{2}$ errors which can be decoded by the decoder of the code $\mathcal{C}$. We will show how to modify and improve the complexity of this algorithm. Assume for example that $t = \frac{d-1}{2} + 1$. Then, there are two channel outputs, say $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, that are different in at least one bit location. If we flip this bit in both $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, then in exactly one of them the number of errors reduces by one and thus is at most $\frac{d-1}{2}$, which can be decoded by $\mathcal{D}_{\mathcal{C}}$. We show how to generalize this idea for arbitrary t. We let $\rho = t - \frac{d-1}{2}$. First, we prove the following Lemma.

Lemma 14. *There exist two channel outputs $\boldsymbol{y}_i, \boldsymbol{y}_j$ such that $d_H(\boldsymbol{y}_i, \boldsymbol{y}_j) \geqslant 2\rho - 1$.*

Proof: Assume to the contrary that there do no exist such words. Then, the words $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N$ form an anticode of diameter $2\rho - 2$. According to [6], the maximum size of such an anticode is $b_{\rho-1,n} = \sum_{i=0}^{\rho-1} \binom{n}{i}$, while according to (5) the value of N satisfies

$$N > N_t(2, d) = \sum_{i=0}^{t-\frac{d+1}{2}} \binom{n-d}{i} \sum_{k=d-t+i}^{t-i} \binom{d}{k}$$
$$= \sum_{i=0}^{\rho-1} \binom{n-d}{i} \sum_{k=d-t+i}^{t-i} \binom{d}{k} > b_{\rho-1,n}.$$

∎

Algorithm 15. *The input to the decoder are the N words $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ and it returns an estimation $\widehat{\boldsymbol{c}}$ on $\boldsymbol{c}$.*

Step 1. Find two words $\boldsymbol{y}_i, \boldsymbol{y}_j$ such $d_H(\boldsymbol{y}_i, \boldsymbol{y}_j) \geqslant 2\rho - 1$, and let $i_1, i_2, \ldots, i_{2\rho-1}$ be $2\rho - 1$ different indices that the two vectors are different from each other.

Step 2. For all vectors $\boldsymbol{e}$ of weight ρ on these $2\rho - 1$ indices,

a) $\mathcal{D}(\boldsymbol{y}_1 + \boldsymbol{e}) = \widehat{\boldsymbol{c}}_1$, $\mathcal{D}(\boldsymbol{y}_2 + \boldsymbol{e}) = \widehat{\boldsymbol{c}}_2$.

b) If $\max_{1 \leqslant i \leqslant N}\{d_H(\widehat{\boldsymbol{c}}_1, \boldsymbol{y}_i)\} \leqslant t$, $\widehat{\boldsymbol{c}} = \widehat{\boldsymbol{c}}_1$.

c) If $\max_{1 \leqslant i \leqslant N}\{d_H(\widehat{\boldsymbol{c}}_2, \boldsymbol{y}_i)\} \leqslant t$, $\widehat{\boldsymbol{c}} = \widehat{\boldsymbol{c}}_2$.

Theorem 16. *The output of Algorithm 15 satisfies $\widehat{\boldsymbol{c}} = \boldsymbol{c}$.*

Proof: The success of Step 1 is guaranteed according to Lemma 14. For every index i_j, $1 \leqslant j \leqslant 2\rho - 1$, exactly one of the channel outputs $\boldsymbol{y}_1$ or $\boldsymbol{y}_2$ has an error. Therefore, either $\boldsymbol{y}_1$ or $\boldsymbol{y}_2$ has at least ρ errors on these indices. Without loss of generality assume it is $\boldsymbol{y}_1$ and let $I \subseteq \{i_1, \ldots, i_{2\rho-1}\}$ be a subset of its error locations, where $|I| = \rho$. In Step 2 we exhaustively search over all error vectors $\boldsymbol{e}$ of weight ρ on these $2\rho - 1$ indices. For every error vector $\boldsymbol{e}$ let $I_{\boldsymbol{e}} = \{i : e_i = 1\}$. Therefore, there exists an error vector $\boldsymbol{e}_1$ such that its the set of indices with value one is covered by the set I, i.e. $I_{\boldsymbol{e}_1} \subseteq I$. Hence, $d_H(\boldsymbol{c}, \boldsymbol{y}_1 + \boldsymbol{e}_1) \leqslant \frac{d-1}{2}$, so the decoder in Step 2.b succeeds. Hence the algorithm succeeds and $\widehat{\boldsymbol{c}} = \boldsymbol{c}$. ∎

The complexity of Algorithm 15 is significantly better than the naive approach. However, the larger the value of ρ is, the larger the algorithm's complexity is. We report on another algorithm with better complexity, $O(nN)$, for the case $d = 3$.

VI. Conclusion

This paper proposed a model of an associative memory: Two words are associated if their Hamming distance is no greater than some prescribed value t. Our main goal was to study the maximum size of the associative memory output as a function of the number of input words and their minimum distance. We observed that this problem is a generalization of the sequences reconstruction problem that was proposed by Levenshtein. Finally, we presented a decoding algorithm for the sequences reconstruction problem.

VII. Acknowledgement

The authors thank Tuvi Etzion for helpful discussions on anticodes. This research was supported in part by the ISEF Foundation, the Lester Deutsch Fellowship, and the NSF Expeditions in Computing Program under grant CCF-0832824.

References

[1] R. Ahlswede, H.K. Aydinian, and L.H. Khachatrian, "On perfect codes and related concepts," *Designs, Codes and Cryptography*, vol. 22, pp. 221–237, 2001.

[2] T. Batu, S. Kannan, S. Khanna, and A. McGregor, "Reconstructing strings from random traces," *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 903–911, 2004.

[3] Y. Cassuto and M. Blaum, "Codes for symbol-pair read channels," *IEEE Trans. on Information Theory*, vol. 57, no. 12, pp. 8011–8020, Dec. 2011.

[4] T. Holenstein, M. Mitzenmacher, R. Panigrahy, and U. Wieder, "Trace reconstruction with constant deletion probability and related results," *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 389-398, 2008.

[5] S. Kannan and A. McGregor, "More on reconstructing strings from random traces: insertions and deletions," *Proc. IEEE International Symposium on Information Theory*, pp. 297–301, Australia, Sep. 2005.

[6] D.J. Kleitman, "On a combinatorial conjecture of Erdős," *J. Combin. Theory*, vol. 1, pp. 209–214, 1966.

[7] E. Konstantinova, "On reconstruction of signed permutations distorted by reversal errors," *Discrete Mathematics*, vol. 308, pp. 974–984, 2008.

[8] E. Konstantinova, "Reconstruction of permutations distorted by single reversal errors," *Discrete Applied Math.*, vol. 155, pp. 2426–2434, 2007.

[9] E. Konstantinova, V.I. Levenshtein, and J. Siemons, "Reconstruction of permutations distorted by single transposition errors," arXiv:math/0702191v1, February 2007.

[10] V.I. Levenshtein, "Reconstructing objects from a minimal number of distorted patterns", (in Russian), *Dokl. Acad. Nauk 354* pp. 593-596; English translation, *Doklady Mathematics*, vol. 55 pp. 417–420, 1997.

[11] V.I. Levenshtein, "Efficient reconstruction of sequences," *IEEE Trans. on Information Theory*, vol. 47, no. 1, pp. 2–22, January 2001.

[12] V.I. Levenshtein, "Efficient reconstruction of sequences from their subsequences or supersequences", *Journal of Combin. Theory, Ser. A*, vol. 93, no. 2, pp. 310–332, 2001.

[13] V.I. Levenshtein, E. Konstantinova, E. Konstantinov, and S. Molodtsov, "Reconstruction of a graph from 2-vicinities of its vertices," *Discrete Applied Mathematics*, vol. 156, pp. 1399–1406, 2008.

[14] V.I. Levenshtein and J. Siemons, "Error graphs and the reconstruction of elements in groups," *Journal of Combin. Theory, Ser. A*, vol. 116, pp. 795–815, 2009.

[15] K. Viswanathan and R. Swaminathan, "Improved string reconstruction over insertion-deletion channels," *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 399–408, 2008.

[16] E. Yaakobi, J. Bruck, and P.H. Siegel, "Decoding of cyclic codes over symbol-pair read channels," *IEEE International Symposium on Information Theory*, Cambridge, MA, July 2012.