Neural Associative Memory with Finite State Technology

D. Gnanambigai†, P. Dinadayalan † and R. Vasantha Kumari ††

† Department of Computer Science, K. M. centre for P.G. Studies, Puducherry, India.
††Principal, Perunthalaivar Kamarajar Government Arts College, Puducherry, India.

Summary
Morphological learning approaches have been successfully applied to morphological tasks in computational linguistics including morphological analysis and generation. We take a new look at the fundamental properties of associative memory along with the power of turing machine and show how it can be adopted for natural language processing. The ability to store and recall stored patterns based on associations make these memories so potentially valuable for natural language processing. A neural associative memory recollects word pattern based on an input that resembles the input word pattern, which is sufficient to retrieve the complete pattern. Morphological parser and Turing lexicon are used to process regular inflectional morphology. The rules of unrestricted grammar are extracted from the morphological parser. Turing Lexicon processes input word using the unrestricted grammar. The associative learning algorithm is a classification method based on Euclidean minimum distance for stem or attribute. The stems and attributes are indexed in the associated memory for quick retrieval. The associative memory is used to recall correct and incorrect word patterns. A machine is designed to implement the morphological process as a special case by achieving both concatenative and non-concatenative phenomena of natural languages. The experimental results shows that the proposed approach has attained good performance in terms of speed and efficiency.

Keywords:
Artificial Neural Networks, Associative Memory, Morphology, Morpheme, Patterns, Turing machine, lexicon, parser.

1. Introduction

Neural associative memory with finite state technology is a new method which combines neural associative memory and Turing machine for languages processing. An Artificial Neural Network (ANN) [2][10], often just called a Neural Network(NN), is a mathematical model or computational model based on Biological neural network. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. They can be used to model complex relationship between inputs and outputs or to find patterns in data. In neural network, mapping capabilities is that they can map input patterns to their associated output patterns. Learning [2] methods in neural network can be broadly classified into two basic types: supervised and unsupervised. In supervised learning every input pattern that is used to train the network is associated with an output pattern, which is the target or the desired pattern. In unsupervised learning, the target output is not present in the network, the system learns of its own by discovering and adapting to structural features in the input pattern. The information provided helps the network in its learning process.

Artificial neural networks are a type of massively parallel computing architecture, which is based on a very simplified brain like information encoding and processing model. Neural network can and do exhibit brain-like behaviours such as association, recognition, generalization, classification, feature extraction and optimization. These behaviours fall under the general categories of searching, representation and learning which are closely related to the associative property and self-organizing capability of the brain. Association is one of the fundamental characteristics of human brain. The human brain operates in the associated manner; that is a portion of the recollection can produce an associated stream of data from the memory. It is used to store and retrieve information from the database. We store and retrieve data based on association (i.e) search a target by association. The exitement in neural network started mainly due to the difficulties in dealing with problems in the field of speech and image, natural languages and decision making using known methods of pattern recognition and artificial neural network.

Automata theory [3] is one of the longest established areas in Computer Science. Over the past few years, Automata theory has not only developed in many different directions, but has also evolved in an exciting way at several levels: the exploration of specific new models and applications has at the same time stimulated a variety of deep mathematical theories. Standard applications of Automata theory include pattern matching, natural language processing and syntax analysis.

Morphology [2][6] is a description of a linguistic phenomenon. Knowledge representation is representing linguistic information in particular morphological information. Morphology is the name given to the study of the "minimal unit of grammatical analysis". Morphology is the field of linguistics that studies the internal structure of words.

(Words as units in the lexicon are the subject matter of

Manuscript received October 5, 2008
Manuscript revised October 20, 2008

lexicology.) While words are generally accepted as being the smallest units of syntax, it is clear that in most (if not all) languages, words can be related to other words by rules. In this way, morphology is the branch of linguistics that studies patterns of word-formation within and across languages, and attempts to formulate rules that model the knowledge of the speakers of those languages.

2. Related Work

We aim towards a new technique, namely, "Neural associative memory with finite state technology ". This technique comprises of Turing Machine, Associative memory and Morphology. Turing machine is powerful than any other finite state machines like finite state automata, pushdown automata and linear bounded automata. Turing machine is capable of moving in both directions. It has finite memory which is used to record necessary changes while performing morphological processing. Turing machine is used for concatenative and orthographic morphological processing. To process non-concatenative patterns we require large memory to efficiently retrieve the word patterns. Associative memory is used for this purpose. Neural associative memory implements pattern recognition and pattern association**.** A short introduction of neural associative memory, Turing machine and morphology of natural language is explained in this section.

In its simplest form, Associative Memory [5][13] is a memory system that allows one data item to be associated to another, so that access to one data item allows access, by association, to the other. In the neural network literature, Associative Memories are referred to as being auto-associative or hetero-associative. An auto-associative memory allows recall of the same item that is put in. A hetero-associative memory allows recall of an associated item that is different from the input query. Patterns stored in an associative memory are addressed by their contents. Associative memory is content-addressable memory as well as parallel search memory. The major advantage of Associative memory is its capability to performing parallel search and parallel comparison operations. Associative neural memories are a class of artificial neural networks which have gained substantial attention relative to other neural net paradigms. A number of important contributions have appeared in conference proceedings and technical journals addressing various issues of associative neural memories, including multiple-layer architectures, recording/storage algorithms, capacity, retrieval dynamics and, fault-tolerance. Currently, associative neural memories are among the most extensively studied and understood neural paradigms.

A model of computation [3] consisting of a set of states, a start state, an input alphabet, and a transition function that maps input symbols and current states to a next state. Computation begins in the start state with an input string. It changes to new states depending on the transition function. A Turing machine [3] is a simple mathematical model of modern digital computers having strong computational capability. The Turing machine is a very simple machine, but, logically speaking, has all the power of any digital computer. The Turing machine has the following features: it has an external memory which remembers arbitrarily long sequence of input and it has unlimited memory capacity.

Morphology is the study of how words are built up from smaller units called morphemes. The main morpheme is called the stem and the others are affixes. The stem gives the core meaning and the affixes changes their meaning and grammatical function. Natural language processing [6] deals with words whose grammatical parts are dealt with the help of morphology under the categories of concatenative morphology and non-concatenative morphology. Concatenative morphology glues together stem and affixes. Non-concatenative morphology is more complex. It is a form of word formation in which the root is modified other way than by stringing morpheme together. Non-concatenative morphology is known as base modification, a form in which part of the root undergoes a phonological change. It is also referred to as truncation, deletion, or subtraction. This process removes phonological material from the root. Inflectional morphology is the combination of a base form with the grammatical morpheme in order to express a grammatical feature such as singular/plural or past/present tense. Derivational morphology refers to the creation of new word with a different class. The kind of application that needs morphological processing are grammar, spelling checkers and information retrieval (i.e) sometimes it is useful to search documents**.**

In connection with the analysis we carried out towards the literature survey, we have started working on a new concept namely, "Neural associative memory with finite state technology ". We collected necessary materials, Journals, basic concepts and supporting evidence. From the literature survey it has been examined how the finite state machine is essential to model morphotactics. Morphotactics is a model of morpheme ordering. The beneficial of finite state automata in morphological analysis is acquired. The part played by finite state automata in the morphological study is realized. The significance of associative memory in morphological process is noted.

The limitations of Finite state automata have been reviewed. While processing language using finite state automata the time taken is more. When word is processed, the number of states taken by the finite state automata increases. When finite state automata is used to process morphology of a language, the FSA has to undergo more number of states. Therefore, it takes more traversing time to search a word. Finite state automaton is a read only machine. It reads the input only by forward traversing so it does not review the previous traversal. In addition the property of finite state automata states that these machines have no memory. Therefore, orthographic changes cannot be processed by finite state automaton. That is, since there is no writing capability in the finite state automaton modification is not possible.

3. Architecture of Associative Memory and Turing Machine

We take a new look at the fundamental properties of associative memory and show how it can be coupled with the power of Turing machine for natural language processing. The ability to store and recall patterns based on associations make these memories so potentially valuable for natural language processing. Turing machine with memory is considered for describing the morphological processes of natural languages. It is used for identifying concatenative and non-concatenative morphology. Traditional methods does not contain memory therefore the transition changes every time.

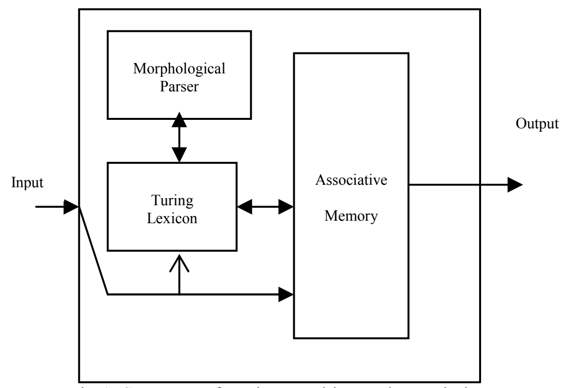

Fig.1 Structure of Turing machine and associative memory for Morphology

The new proposed machine remembers the finite number of symbols and this machine easily recalls whenever needed. The memory used is associative memory. This is used for pattern recognition and pattern association. The stems, irregular nouns and irregular verbs are present in associative memory in groups of different classes. Associative memory acts as a database (look-up-table). The system follows a modular architecture in which the input text is processed through a cascade of modules that handle different phases of translation process. In an over-view the model comprises of Turing lexicon, morphological parser and associative memory.

Turing lexicon splits the stem and affixes to find the basic class of the morpheme to associate with the morphological parser which describes the set rules. Morphological parser describes the set of rules. Associative memory comprises of associated morphological patterns. Turing lexicon is used to encode morphotactic knowledge. Turing machine acts as a lexicon. When a natural language query is given it splits it into smaller unit called morphemes. The Turing lexicon is a three tape Turing machine which has read/write heads. The symbols can be read or modified. $Tape_1$ holds the given input query. $Tape_2$ holds the grammar of natural language processing which is loaded from the morphological parser. $Tape_3$ consists of the results of the query.

Morphological parser has a set of rules for concatenative and orthography morphology. The morphological parser is like a file. It is useful for splitting the query into stems and by adding affixes to form new words. When the nature of query is concatentive or orthographical the Turing lexicon, identifies the corresponding rule from the morphological parser. The lexicon splits the query into tokens to form other related concatenative words which is stored in $Tape_3$.

The morpheme of the query is sent to the associative memory to find out the correctness of the query if the morpheme is a valid query then the result is produced as output from the $tape_3$ otherwise the query is rejected. For processing orthographic query, the input is slightly modified in $Tape_3$ based on the rules from morphological parser.(like consonant doubling, e-deletion, e-insertion, y-replacement, k-insertion etc).

Since the orthographic query processing is easily done it is inferred that the Turing machine is more powerful than the traditional approaches. Turing machine is capable of reading, writing, deleting, modifying and comparing facilities. The patterns that are not captured by grammar are considered to be non-concatenative. Therefore the query associates with the patterns in the associative memory to produce the required result.

Associative memory is like a database. It has the different classes of look-up table. When a word and its associated patterns are to be read from the associative memory the content of word is specified. The memory locates the associated words which match the specified content and marks them for reading. The associative memory is uniquely suited to do parallel searches by data association. The learning techniques used in the associative memory are supervised learning as the given input query is associated with the target pattern in the associative memory. To efficiently retrieve the associated data items we use the Euclidean distance method.

$$word_i = \min \| C_i - M \| \quad (1)$$

where $word_i$ - is the desired output,
C_i - closest patterns from the classes,
M - the key morpheme.

When a query is sent to the associative memory it searches for the associated minimum distance pattern from the associative memory. The searching is a parallel search mechanism. All the look-up-table are searched simultaneously. The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by its content. The distance between the given pattern and the target is determined. If there is no difference the exact content is located from the associative memory and all the associated patterns are produced. If the input query does not match the query is rejected (i.e) the input is invalid.

4. Simulation

The Turing lexicon and Morphological parser splits the input into morpheme and affixes and merges morphemes and affixes for concatenative morphology. The morphological parser has a set of rules for splitting and merging the input pattern. Some of the rules are given below.

(i) y-replacement rule:

<noun> → <word>y
<noun>→ <word><affix>
<affix>→ies

(ii) e-insertion rule:

<noun> → <word>
<noun>→ <word><affix>s
<affix>→ e

(ii) consonant doubling rule:

<verb> → <word>g
<verb>→ <word><affix>
<affix>→gg<con>
<con>→ ed

(iii) e-deletion rule:

<verb> → <word>e
<verb>→ <word><affix>
<affix>→ed

(iv) concatenation rule:

<noun> → <word>
<verb>→ <word><affix>
<affix>→s

The input query is placed on $tape_1$ of Turing lexicon. The input is surrounded by end markers $, for example $city$. The rule in $tape_2$ is derived from the morphological parser. Each rule stored in $tape_2$ is separated by two consecutives #. $Tape_3$ is a derivation of input pattern based on the morphological rules. For concatenative morphology, any move on the tape are to the beginning or ending of the stem on the $tape_3$ and if the last character of the stem needs changing this happens before the affix output is added.

If the input query is non-concatenative in nature, it is associated with associative memory and the input query searches simultaneously in all the lookup tables of the associative memory. The look-up stage in a natural language system can involve much more than simple retrieval. For example, mouse is the input query and the associative word mice is stored in the look-up table. Using the minimum Euclidean distance formula we find the distance between the input query and the patterns stored in the associated memory. If there is no difference, the correct pattern is located and the result is produced, otherwise it rejects the input query. That is, the given query is not a valid word of the language.

For simulation 200 samples of words are taken for the verification of the implemented system. The input pattern can be an inflected verb, infected noun, regular verbs, irregular verbs, regular nouns and irregular nouns. The sample patterns taken by the Turing Lexicon and the morphological parser are processed successfully to produce the desired output. For concatenative morphology 92% of the patterns are correctly recognized and recalled and 8% of the patterns leads to error. All the incorrect input patterns are rejected.

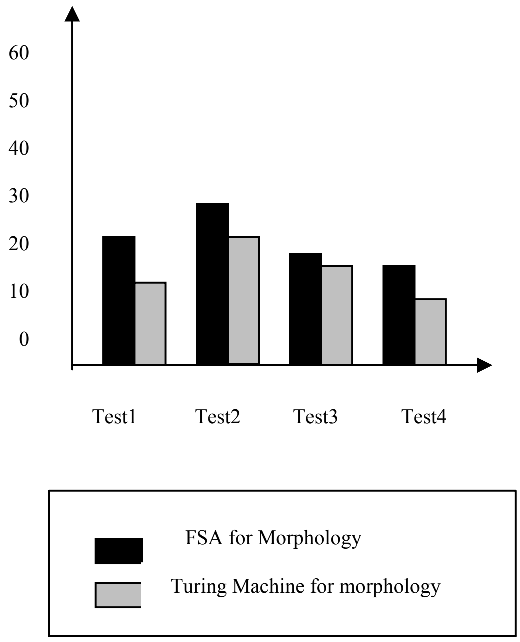

Fig.2 Comparison between FSA and TM

For non-concatenative morphology the given pattern is associated with the associative memory and successfully searched to produce the desired output. If the input pattern are present in the look-up table an associative search is made and the related output is produced. If the input pattern is not present in the look-up table then the pattern is rejected as it is not a word of English. For non-concatenative morphology all correct words are recognized by this structure. There is no error in the non-concatenative morphology. If input patterns are incorrect in non-concatenative, all incorrect patterns are rejected.

Neural associative memory with finite state technology is combination of Turing Lexicon and Associative memory. The searching time depends the sum of the searching time of Turing Lexicon and Associative memory.

Total_Search_Time=TL_Time+AM_Search_Time (2)

The Turing Lexicon searching time can be calculated in terms of number of states traversed. Associative memory search time is calculated based on the distance between actual pattern and target pattern.

Thus the structure is tested and implemented successfully. The model is implemented using Java and Jflap tool. To evaluate the performance of finite state automata and proposed model, a large number of sample patterns are taken for experiment. The first thing we did was to retrieve all the nouns and verbs from Wordnet database. As given in the graph, the performance is compared with the two models Finite State Automaton and Turing Machine. While using Finite State Automata for natural languages processing, often the generated networks become too large to be practical. In such cases Turing Machine can make the size manageable.

5. Conclusion

Thus a small efficient machine that encodes general morphological knowledge has been developed by implementing Turing machine and Associative memory. Word recognition is made to check whether a word is a member of a language. Morphological process has been achieved for both concatenative and non-concatenative phenomena of natural languages. We believe that the proposed structure will be the basis of a general solution for morphological analysis of language processing. Future work consists in looking for better measures to incorporate this work with Counter-Propogation Network for pattern completion (noisy data). Further work involves more efficient and better methods for representing parts-of-speech, adjectives (degrees of comparision) and multi-word recognition etc. It can also be developed for Crosslanguage generation or translation.

References

[1] Sergio Rao, Fernando Nino (2003) ‘Classification of natural language sentences using Neural Networks’, The Florida AI Research Society Conference, pp. 444-449.

[2] Goldsimith.J.A. (2002) ‘Unsupervised Learning of the morphology of a Natural Language’, Computational Linguistic,27:2 pp.153-198.

[3] John E. Hopcroft, Jeffery D. Ullman (1994), Introduction to Automata Theory, Languages, and Computation, NPH.

[4] Amengual. J.C and Benedi J.M, (2001) ‘EUTRANS – I speech translation system’, Machine Translation Journal.

[5] T. Gotez, H. Wunsch (2001) ‘An abstract Machine Approach to Finite State Transduction over Large Character Sets’, Finite State Methods in Natural Language Processing.

[6] Chun-Hsien Chen, Vasant Honavar (2000) ‘ A Neural Network Architecture for Syntax Analysis’, IEEE Transactions on Neural Networks, Vol. No.1.

[7] Van Noord. G, Gerdemann. D (2003) ‘An extensible regular expression compiler for Finite State approaches in Natural Language Processing’, Natural Language Engineering , Cambridge Journals, 9: 65-85.

[8] Cohen-Sygal. Y, Gerdemann. D, Wintner. S (2003) ‘Computational Implementation of non-concatenative morphology’, Proceedings of the workshop on Finite-State Methods in Natural Language Processing, pp.59-66.

[9] Yona. S, Wintner. S (2005) ‘A finite-State morphological grammar of Hebrew ‘, Proceedings of the workshop on Computational Approaches to Semitic Languages.

[10] Cohen Sygal. Y, Wintner . S (2006) ‘Finite-State Registered Automata for Non- Concatenative Morphology’, Computational Linguistics, pp.49-82.

[11] H. Kang (1994) ‘Multilayer associative neural networks storage capacity versus perfect recall’, IEEE Transactions on Neural Networks, vol.5, no.5, pp.812-822.
[12] P.K. Simpson (1990) ‘Higher-Ordered and interconnected bi-directional associative memories’, IEEE Transactions on Systems, Man, and Cybernetics, vol.20, no.3, pp.637-652.
[13] Ajit Narayanan & Lama Hashem (1993) ‘An abstract Machine Approach to Finite State’, In Proceedings of EACL'1993, pp.297-304.
[14] Yingquan Wu and Dimitris A. Pados (2000) ‘A Feedforward Bidirectional Associative Memory’, IEEE Transactions on Neural networks, vol. 11, No.4.
[15] H. Shi, Y. Zhao, and X. Zhuang (1998) ‘A general model for bidirectional associative memories’, IEEE Trans on Systems, Man, and Cybernetics, vol. 28, no. 4, pp. 511–519.
[16] Z. Wang (1996) ‘A bidirectional associative memory based on optimal linear associative memory’, IEEE Transactions on Computers, vol. 45, no. 10, pp. 1171--1179.
[17] H. Kang (1994) ‘Multilayer associative neural network (MANNs) Storage capacity versus perfect recall’, IEEE Transactions on Neural Networks, vol. 5, no. 5, pp. 812--822.
[18] Philip D. Wasserman (1989) Neural Computing – Theory and Practice, VNR.
[19] M. Anabda Rao (2003) Neural Networks – Algorithms and Applications, NPH.
[20] HOPF82 Hopfield, J. J. (1982) ‘Neural networks and physical systems with emergent collective computational abilities’, Proceedings of the National Academy of Sciences, Vol. 79, pp. 2,554-2,558.

Gnanambigai. D obtained her M.Sc. degree in Computer Science from Pondicherry University, India in 1996 and M.Phil. degree in Computer Science from Manonmaniam Sundaranar University, Tirunelveli, India in 2002. She is presently pursuing her Ph.D. in Computer Science from Vinayaka Missions University, Salem, India under the guidance of Dr. R. Vasanthakumari. She has published papers in national and international conferences in the area of Neural Networks and Automata theory. She is working as a Lecturer (from 2000) in the Department of Computer Science, Kanchi Mamunivar Centre for Post-graduate Studies, Puducherry, India.

Dinadayalan. P. obtained his M.C.A degree in the year 1996 and M.Tech degree in Computer Science and Engineering in the year 2000 from Pondicherry University, India and M.Phil. degree in Computer Science from Manonmaniam Sundaranar University, Tirunelveli, India in the year 2002. He is presently pursuing his Ph.D. in Computer Science from Vinayaka Missions University, Salem, India under the guidance of Dr. R. Vasanthakumari. He has published papers in national and international conferences in the area of Artificial Neural Networks and Theory of Computer Science. He is working as a Lecturer (from 1996) in the Department of Computer Science, Kanchi Mamunivar Centre for Post-graduate Studies, Puducherry, India.

Dr. R. Vasanthakumari obtained her Ph.D. degree from Pondicherry University, in 2005. She is working as a Principal in Government College, Puducherry, India. She has more than 27 years of teaching experience in P.G. and U.G colleges. She is guiding many research scholars and has published many papers in national and international conference and in many international journals.