Sequential Adaptive Memory Model based English-Hindi Machine Translation System

Sandeep Saini, *Member, IEEE,* and Vineet Sahula, *Senior Member, IEEE*

***Abstract*—Language translation is certainly one of the tasks in which machine is lagging behind the cognitive powers of human beings. Cognitive architectures are developed to map the human cognitive processes with a computational framework. In this work, we propose a cognitive model based machine translation system for English to Hindi Machine Translation. Hierarchical Temporal Memory (HTM) is a cognitive architecture proposed by Jeff Hawkins et al. and its updated version CLA provides the computational framework for neocortex region of the brain, where speech and language related comprehension and production take place. In the present work, the original structure of CLA is substantially augmented, and a new model, Sequential Adaptive Memory (SAM) is proposed. This model works on the sequence to sequence learning approach which is better suited to translation task. This enables the creation of word pairs, dictionaries, and rules for translation. SAM results are compared with the results of traditional phrase-based SMT as well as of the state-of-the-art Neural Machine Translation approaches for machine translation. The results are comparable to those of the conventional approaches. We observe that SAM requires a quiet small amount of data size for training yet exhibits satisfactory translation for low resource languages as well.**

***Index Terms*—cognitive system and development, Cognitive linguistics, Machine Translation, biology-inspired architecture for development, Indian Languages, neural networks for development**

I. Introduction

Machine translation was one of the initial tasks taken by the computer scientists.The research in this field has been going on for more than 50 years. In these years, it is remarkable progress that linguists and computer engineers have worked together to achieve the current status of machine translation. In most of the statistical approaches for machine translation [1] and [2], the basic units of translation process are phrases and sentences. These phrases can be composed of one or more words. Most of the modern translation systems are based on Bayesian inferencing to predict and estimate translation probabilities for pairs of phrases. In these pairs, one phrase belongs to the source language and the other to the target language. In real time, these distinct phrase pairs may often share significant similarities. But linguistic or otherwise, they do not share statistical weight in the model's estimation of their translation probabilities. Thus even after ignoring the similarity of phrase pairs, the whole process leads to general sparsity issues. Since the probability of these phrases is very low, thus pairing and predicting the correct pair is very difficult in these systems. With the limitations of conventional Machine Translation Systems [3], there is a demand to look for alternate methods for machine translation.

In recent years, Google has also shifted its focus of translation research towards Neural Machine Translation (NMT). Sutskever, et al. [4] proposed a sequence to sequence learning mechanism using long and short term (LSTM) memory models. This neural network-based machine translation system had 8 layers of encoder and 8 layers of the decoder. Normally neural machine translation tends to require a lot of computing power which means that it is normally a great technique if only you have enough time or computing powers. The other issue with older NMT was inconsistency in handling rare words. Since these inputs were sparsely available in the network, the learning and inferencing were not efficient. By using LSTM models and having 8 layers of encoder and decoder, this system removes these errors to a large extent. The third major issue with NMT was that the system used to forget the words after a long. This issue is also resolved in 8 layer approach. After 2014, this work from Sutskever et. Al. have inspired a lot of researchers and NMT is developing as a good alternative to conventional machine translation techniques.

Considering the ability of human brain to acquire and translate multiple languages with better efficiency, researchers have started to focus on human brain-inspired machine translation systems. Linguists have proposed a lot of language acquisition theories and based on the acquisition of the second language. Our brain is capable of translating from one language to other. Recent research reports [5–8] have investigated the aspects of human translation process. Among these studies, authors [7] hav proposed a model which can replicate the human translation behavior. This model is based on User-Activity Data (UAD). UAD consists of the translator's recorded keystroke and eye-movement behavior. These recordings makes it possible to replay a translation session and to register the subjects' comments on their behavior during a retrospective interview. This model is not suitable for large document translation and requires special equipment for training and testing.

Cognitive architectures have been proposed for natural language comprehension and generation from past two decades. Few models are based on speech modeling. A set of such architectures are based on Adaptive Resonance Theory (ART).

Sandeep Saini is pursuing his Ph.D. from Malaviya National Institute of Technology, Jaipur, India.
Vineet Sahula is Professor at the department of Electronics and Communication Engineering at Malaviya National Institute of Technology, Jaipur, India

This theory claims that "in order to solve the stability-plasticity dilemma, only resonant states can drive new learning." In the year 1999, Stephen Grossberg, proposed the link between attention, intention, and consciousness with the help of cognitive architecture ARTSTREAM [9]. He improved the model in 2003 with name ARTPHONE [10] and ARTWORD [11] architectures. These architectures have been used to model perceptual processes involved in speech categorization, auditory streaming, source segregation and phonemic integration [12]. Most of the research in NLP is based on analysis of textual data. One of the real-time NLP cognitive architecture is NL-soar [13]. This architecture combines the semantic and syntactic knowledge of simple instructions for immediate reasoning task in block words. One of the most recent models, LUCIA [14], comprehends natural language and combines cognitive linguistics with known properties of human language processing. It is built on Embodied Construction Grammar (ECG) and the SOAR cognitive architecture. Most of the existing NLP systems are limited either in their domain of application or in terms of the syntactic structures they can understand. Recent architectures, such as DIARC [15], aim at supporting more naturally sounding requests.

In 2008, Dileep George [16] proposed a framework to explain the working of the human brain. In his work, he introduced a Hierarchical Temporal Memory (HTM) model to explain the structure and working of Neocortex region of the human brain. This model has been proved to be working similar to a human brain for applications like image processing. In 2009, Dan Robinson et.al. [17] applied HTM for spoken language identification. For more than two languages, the authors were able to get better results in language identification task using HTM.

A. Our contribution

In this work, we propose a human-inspired machine translation model for better English-Hindi translation. We have augmented Hierarchical Temporal Memory model to suitably match the requirements of machine translation process and have proposed Sequential Adaptive Memory (SAM) model for this task. We compare results of our approach with those of conventional SMT and recently reported NMT based machine translation systems.

Rest of the the paper is organized as follows. We have explained about the concept and process of language processing in human brain in section II. Neocortex region is briefly introduced and explained how it helps in the process of language comprehension and generation. We have reviewed Cortical Learning Algorithm (CLA) in section III, which is based on the architecture of Neocortex of the human brain. We have explained the features, functions and limitation of this algorithm. In section IV, we proposed our translation model based on Sequential Adaptive Memory (SAM). This model is discussed in detail in this section along with implementation details. We have compiled our results in section V and compared our system with existing SMT and NMT based systems. We have concluded the article in section VI.

II. Language Processing in Human Brain

According to the latest research in the field of human brain language comprehension [18], [19], neocortex region of the human brain is primarily responsible for language and speech processing inside the brain. This is one part of the human brain which sets us apart from other animals. This part of the brain is involved in sensory perception, generating motor commands, spatial reasoning, handling speech, and language. The neocortex is also the part, where our creativity emerges. The neocortex is divided into frontal parietal-occipital and temporal lobes, which perform different functions. For example, the occipital lobe contains the primary visual cortex in the temporal lobe contains the primary auditory cortex. From our complex language system to our capacity to understand the environments and create technology none of that would be possible without the crowning achievements of evolution called the neocortex.

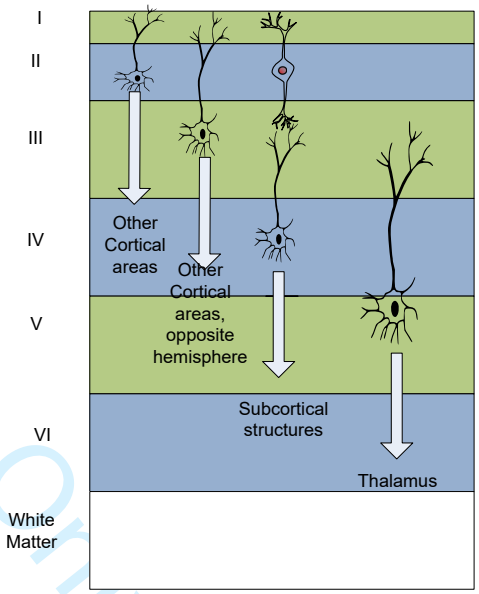

Fig. 1. Structure of the Neocortex region of the brain.

The neocortex is a 6 layered sheet covering the outer part of the brain. Figure 1 [20] shows a typical structure of the 6 layers present in the region. Anatomy of the neocortex is the most developed in the cerebral tissues. The neocortex consists of the gray matter, or neuronal cell bodies and unmyelinated fibers surrounding the deep white matter in the cerebrum. All human brains have the same structure for neocortex, but the internal neuron connections differ in detail from one person to another. The structure of the neocortex is relatively uniform consisting of 6 horizontal layers, segregated principally by cell type and neuronal connections. However, there are many exceptions to this uniformity. The initial layers are mainly responsible for receiving the sensory data from different neurons. These layers pass the information to the upper layers for more complex decisions and inferencing. The higher layers are inferring and predicting the behavior of motor for different organs. Thus there is a hierarchy in neocortex region which handles the information through billions of neurons sparsely placed inside. The main working principle of this region is the sparse

distribution of neurons carrying the data. Out of all possible neurons available in every layer, only 2-4 percent of the neurons are active at a given time.

Based on the structure and working of neocortex region, Jeff Hawkins proposed a prediction framework to model neocortex behavior using digital computers. This framework was based on the hierarchical temporal structure of memory elements. This model is known as Hierarchical Temporal Memory (HTM) [21]. In the next section, we explain HTM and its later version Cortical Learning Algorithm (CLA) and how these can be used to model a translation system.

III. Review of Cortical Learning Algorithm

In 2006, Jeff Hawkins and Sandra Blakeslee [21] proposed a cognitive model to frame the temporal memory called Hierarchical Temporal Memory (HTM). HTM is later upgraded to Cortical Learning Algorithm (CLA) by Numenta Inc. Cortical Learning Algorithm is based on neuroscience and the interactions and physiology of pyramidal structured neurons in the neocortex region of the human brain. The technology has been tested and implemented in software through example applications from their company Numenta.

The CLA can be viewed and structured as a neural network in its architecture. Thus CLA also has few of the properties exhibited by most of the neural networks like learning, inferring and predicting. CLA models neurons (which are called cells) in columns, layers, and hierarchy. CLA works on the principle of hierarchy of its different layers. CLA have most of the common properties of a typical cognitive architectures, and few of the properties are not implemented in this architecture.

A. Characteristics of CLA

1) Hierarchy: Hierarchy is the main working principle of an HTM network. A CLA network can have any number of layers arranged in the hierarchy, unlike real neocortex which have 6 fixed layers. A typical 4 layer structure of CLA network [21] is shown in Figure 2. It explains how the data is taken at the lowest level from sensory nodes and then communicated within the same layers and the inference of one layer is passed to the higher layers. If we have more than one source of sensory data, then HTMs can be combined in the same hierarchical manner. The pattern learned at lower levels are used in the higher levels, and thus hierarchy reduces the training time by a significant amount and resolves one of the most difficult issues in neural network architectures.

2) Regions: The neocortex is around 2 mm thick sheet covering the human brain. In this sheet, 6 layers are distributed one over another in a hierarchical manner. These layers in CLA models are called regions. Every region has millions of memory cells to store the data. Some regions will receive the data directly from the sensory nodes while the higher level regions would receive the inferred data from the lower regions. A typical region in CLA architecture is shown in Figure 3. In CLA, all the regions have the same structure and similar size.

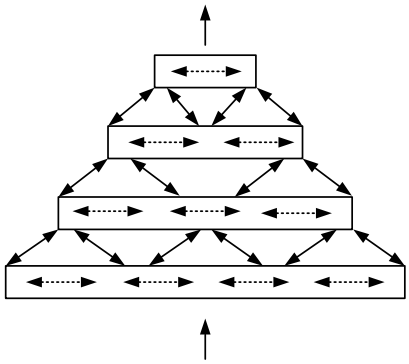

Fig. 2. Simplified diagram of four CLA regions arranged in a four-level hierarchy, communicating information within levels, between levels, and to/from outside the hierarchy.

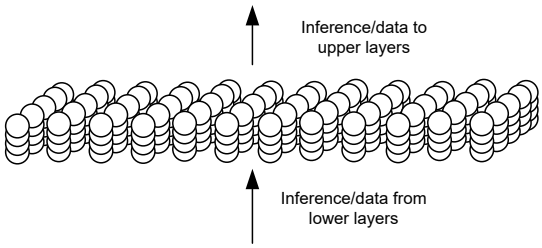

Fig. 3. A section of a CLA region. CLA regions are comprised of many cells. The cells are organized in a two-dimensional array of columns.

3) Sparse Distributed Representations: In neocortex region, the neurons are highly densely packed. But only a few percentages of these neurons are active at a time. Thus any information processing in our brain in done by a fraction of total neurons available in neocortex. In CLA the similar approach is taken to distribute the data as well in all the available cells. Available data is not uniformly distributed among all the cells of a region, but a sparse distribution approach is applied to share the data with few percents of the cells. This encoding scheme is defined as "Sparse Distributed Representations." Figure 4. Shows the highlighted cells (color filled cells in Figure 4) corresponds to a region having data distributed using Sparse Distributed Representations.

4) Temporal component: The third major property of CLA is the role of time in learning, inferring and predicting the outputs. Time plays an important role in all these three activities to be performed by CLA. For example, if we are solving a vision related problem, in which we are asked to identify the direction of movement of an object, then time

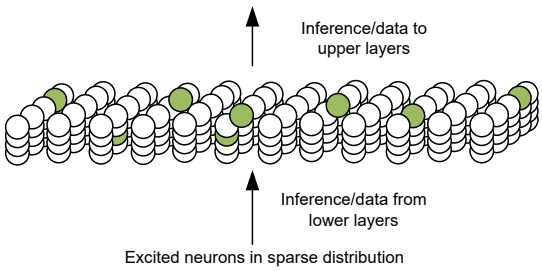

Fig. 4. Sparse Distributed Representations of data in the region of CLA.

frames are important to decide the relative movement of the object, which would help infering the direction of the movement. Similarly, when some words are spoken, those can be spoken by different people at different pitches of sound and have a different time duration. We are still able to recognize the words having such diversities because the syllable level connection is time-dependent.

B. Functions of CLA

CLA performs 4 different functions namely, learning, inference, prediction, and behavior. Every CLA region has 3 functions to perform while the behavior of the CLA may be explained by highest regions only.

1) Learning: Any CLA region learns about its world by finding some fixed patterns and then the sequences of such patterns received by the data from the sensor nodes or the previous layers. The learning power of a region is decided by its size as well. Thus every region has a limit on how much it can learn, and hence hierarchical structure becomes more important as the learning from one region can pass the inference to the higher regions. There are two kind learning processes in CLA; Temporal and spatial. Temporal learning is driven by both unpredicted columns and predicted columns that did not become active. Spatial learning updates the permanence of proximal synapses based on bottom-up prediction error [22].

2) Inference: Once the CLA layer is capable of learning the sequence of data from the input patterns, then it can perform the task of inference on novel inputs as well. Newly received inputs are matched with the previously learned temporal and spatial patterns. Successfully matching new inputs to previously stored sequences is the essence of inference and pattern matching. In inferencing process, time plays an important role. Lets consider an example of the spoken word ”resemble”. Different users will speak this word in different tones, with different pitches, giving different stress to syllables by taking a different amount of time. But we are always able to comprehend the word resemble. That is because neocortex learns the sequence of syllables in this word and whenever this sequence is received, we can infer the meaning of resemble. CLA models are more capable than conventional neural networks because of temporal integration in the learning and inferencing process.

3) Prediction: As a result of learning and inferencing, every region of a CLA stores sequences of patterns. These make the CLA capable of predicting about what inputs are likely to arrive next or what can be the output for rare/novel inputs. If we consider the similar example again, then whenever we come across a long, extended word which is never heard, we always try to break this into root words and find out the meaning. We might not hear the word ”resemblance” for a long time, but due to frequent hearing of resembling and words ending with ”ance,” we can predict the meaning of new words. In CLA as well, a CLA region will make different predictions based on the context that might stretch back far in time. The majority of memory in a CLA is dedicated to sequence memory or storing transitions between spatial patterns. Few of the key features of CLA predictions are:

1) predictions are always continuous and without being conscious of it.
2) prediction occurs in each layer of CLA. Lower layers pass their predictions to upper layers, and final layer produces the output.
3) predictions are context sensitive, and thus CLA is suitable for context-aware problems like language translation.
4) predictions lead to stability of the CLA network. The layers are not only predicting for one layer above it but, sometimes also predict for 2-3 layers above its hierarchy. That helps in cross-validation of the predicted outputs.
5) CLA can separate out a novel or rare inputs and try to predict the best outcomes based on sequential learning.

4) Behavior : Neocortex region receives inputs from sensory nodes and commands motor region to act accordingly. If we hear a sound from one direction, then our eyes are turned in that direction. When we receive any input from skin, our hands move to that part of our body. In the current implementations of CLA, motor part is not implemented, and thus it is difficult to change the behavior of sensory nodes from the inputs of other sensory data.

5) Mathematical Formalization of HTM: Jeff Hawkins et al. [23] have provided the mathematical formalization for HTM networks and how the features mentioned above are supported. To formally describe the activation and learning rules for an HTM sequence memory network, there are three basic aspects to the rules: initialization, computing cell states, and updating synapses on dendritic segments. The complete model has been explained by the authors [23].

C. Applications of CLA

Numenta and other researchers have used CLA, and it's modified versions for various real-time problems. One of the tasks taken by Jeff Hawkins et al. is online sequential learning [24]. This work shows that the model, built using HTM, can continuously learn a large number of the variable-order temporal sequence using an unsupervised Hebbian-like learning rule. The sparse temporal codes formed by the model can generate temporal sequences. The model designed using HTM has been found better than Long and Short Term Memory (LSTM) and extreme learning machine (ELM) [25]. Sabutai et. al. have worked on real-time anomaly detection [26] and [27] using HTMs. An HTM network doesn't provide the anomaly score directly. So the authors have used the internal layers' state vector $\pi(x_t)$ which represents a prediction for $a(x_{t+1})$. The HTM model's predictions are compared with the raw anomaly score and then anomaly likelihood. HTMs have also been used in Content-based image retrieval [28] and Multivariable time-series prediction [29]. The closest application of CLA/HTM for natural language processing was experimented by Dan Robinson et al. [17]. They applied HTM for Spoken Language Identification. The authors experimented with 2 and more languages at a time to identify the spoken languages. For more than two languages, the authors were able to get better results in language identification task using HTM.

D. Limitation of CLA for the translation task

Before we started working on the development of the cognitive architecture for English-Hindi Machine Translation, we had identified the missing features and capabilities of CLA for translation task. As mentioned in the CLA description as well, the current CLA version is not having motor part of neocortex implemented in its architecture. Thus, few of the language processing/generation tasks cannot be performed using the current version of CLA. Another major feature missing in CLA is feedback. There is no feedback from top layers to lower layers about their learning, inferencing, and predictions. Thus error correction using feedback is not possible in CLA. After identifying few of these key missing features in CLA, we propose a new algorithm, which is discussed in detail in next section.

IV. Proposed Machine Translation Model

Numenta Inc. in 2011, enhanced HTM as Cortical Learning Algorithm [30]. Numenta is a leading company working in the field of Machine Learning and helping open source development through its projects. Numenta publishes its work as white papers [1] and distribute the stable versions of its algorithms on GitHub [2]. CLA is not primarily developed for the task of Machine Translation as we discussed in the previous section. In this work, we have augmented CLA for English to Hindi Translation task. The reason behind selecting CLA was its resemblance to human brain's functionality. We have extended CLA for the translation task to design a machine translation system different than those based on statistical machine translation method. A typical SMT system model is illustrated in Figure 5. In such systems, foremost requirement is of the huge bilingual corpus for source and target language. The data is not available in required huge sizes for Indian languages. Thus, SMT base English-Hindi or any other Indian language systems are not performing up to the mark of European counterparts [2]. Cortical learning algorithm doesn't require huge data for its learning and inferencing mechanisms. Thus we have proposed a framework based on the structure of CLA for English-Hindi translation.

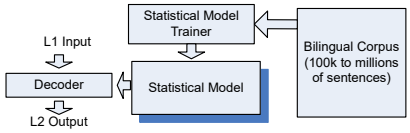

Fig. 5. A typical Statistical Machine Translation System

A. Sequential Adaptive Memory Model

Considering the availability of small bilingual corpus for Indian languages, we were exploring an algorithm which would support learning using minimal data. Ideally, this should be able to start the learning process with the empty dataset as well. The algorithm adapts the memory contents and inferences based on the sequence of learning, and thus it is named as Sequential Adaptive Memory (SAM). The SAM is based on CLA to fit the requirements of machine translation task.

1) Changes in CLA: To augment the CLA for the demands of translation task, following extensions were incorporated in the core architecture of CLA.

1) CLA has an inbuilt spatial pooling algorithm which converts the input data stream into an Invariant Sparse Distribution (ISD). CLA's pooler efficiently performs this task. But in the translation task, we need reverse mapping as well which is not possible in with this pooler, and hence a reversible algorithm is used to map syllables into sparse distribution [31].
2) CLA is developed by considering the same number of cells in each region. As described in Figure 6, there are a different number of elements in each region of the proposed model for the English language. The same is also valid for Hindi language structure. Thus, we have designed each layer of SAM with a different number of cells for optimized learning and inferencing. Since learning and predicting time is depending on the size of the region, thus keeping a constant size for each region would slow down the learning of smaller capacity regions.
3) We have to not only change the size of the complete region but also the number of cells in each column as well. Since syllables would occupy less number of cells compared to phrases and sentences, therefore different columns have a different number of cells.
4) CLA structure allows the inference/predictions from one layer to be passed to immediate next layer only. In the formulation of a translation problem, we realized that syllable could form words and words can directly form meaningful sentences without forming phrases. Thus the inference/prediction can directly transfer to higher layers as well. English and Hindi structures for CLA have been modified in such a way that information can be passed by skipping a layer as well.
5) Temporal pooling is one of the key features of CLA which makes it capable of time-dependent learning sequences. CLA temporal pooler distributes this learning of sequences in Invariant Sparse Distribution (ISD) form, and thus it is not comprehensible by other systems. Temporal Pooler is modified to map the predicted output from ISR to continuous format.
6) CLA prediction process is based on 1-step predictions. This implies that we can have only one prediction from last sequences. Since translation process require both long and short-term memories, we have kept the number of steps for prediction to be more than 1. We have opted for eight step predictions. In our experiments, eight is selected empirically.

2) Architecture of SAM: Sequential Adaptive Memory Models architecture is based on CLA architecture. We have proposed a similar layered structure having hierarchical and temporal properties. The lowest layer takes input from the sensory nodes (incoming data patterns) and passes the inference/predictions to the higher layers. Thus, we have considered

[1] http://numenta.com/papers/
[2] https://github.com/numenta

syllables as the basic unit of the language generation. It has been shown that language acquisition in children starts with the acquisition of syllables in the initial stages of his/her life and slowly the vocabulary is built. Once the child starts learning the second language, then he acquires the second language based on the knowledge of his/her first language and the relations between the two languages [32], [33] and [34]. English and Hindi both have an almost similar number of syllables at around 15,000 each. Thus the structure of English model has syllables at the least level of comprehension, followed by words, phrases and finally forming the meaningful sentences. This hierarchical ordering for English is shown in Figure 6. A similar structure is formed for Hindi as well. A single phrase or sometimes word as well can be a meaningful sentence, and thus words can directly send the inference to the sentence layer.

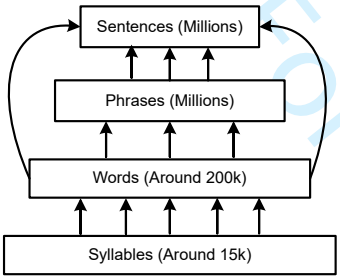

Fig. 6. The Hierarchical ordering of different building blocks for English. Syllables are at the least level of comprehension. By combining syllables, words are formed and thus resulting in phrases and meaningful sentences.

3) Data representation in SAM: Conventionally machine translation is performed at phrase level. In the English language, there are more than 200,000 words in the dictionary and which result in millions of possible phrases and a similar number of sentences. Thus, SMT would require a corpus of the sizes of millions of sentences for efficient training. In the present work, we have proposed mapping human brain's approach to the translation process. In the neocortex, information is stored in neurons which are scattered in the layers of the neocortex. If we assume that we have to represent 200,000 words of a particular language and a cell in a matrix would represent each word, then a square matrix of size 450x450 would be sufficient to store the data. This matrix size is less than the size of a typical digital image. However, in the neocortex, the data is not presented in condensed form, instead use Invariant Sparse Distribution (ISD). The knowledge is represented in bit matrices, which vary in size according to the length of the pattern (sentence in our case). For example, if we assume that the input pattern has a length ten, then the conventional CLA would require at least ten cells and each having the capacity to hold the complete data. We thus require ten square cells of size 16 each. In the proposed SAM, we use adaptive approach, and first input is stored in the minimum possible area, and thus a cell of size one is chosen. Next, three inputs are stored in three cells of 2x2 sized matrices. This way, the size of cells keep on increasing with an increase in the number of inputs. We have shown in Figure 7 the excited cells in conventional CLA model and in Figure 8 the excited cells in proposed SAM model.

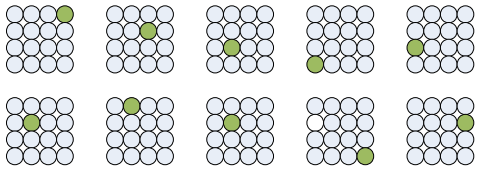

Fig. 7. Invariant Sparse Representation in conventional CLA

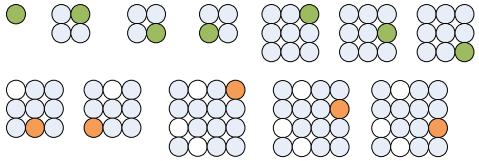

Fig. 8. Invariant Sparse Distribution in proposed SAM

4) Sequential Adaptive Learning: Data in SAM model is represented as bits in a bit matrix. Each of the bit in these bit matrices is related to a single SAM column. In CLA, the size of bit matrix is fixed, while we adapt to the length of the pattern and accordingly change the size of the matrix. The process starts with zero cells per column and every time a new pattern is entered in the region, corresponding to a layer in neocortex, the size of bit matrix is increased to fit the pattern size. We assume that each sentence starts with null. Lets assume a set of sentences in training dataset to be, "A round Orange.", "A Red Apple" and "Round apple." Then the corresponding bit matrix at word level would appear as shown in figure 9.

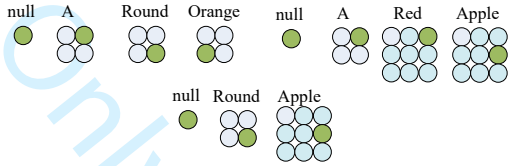

Fig. 9. Bit Matrix for three input sentences, "A round Orange.", "A Red Apple" and "Round apple."

SAM model can start the learning process with limited data, ideally starting with no data as well. If we consider the same two sentences as the only set of input sentences in our system, then SAM can learn from these as well. If we start the retrieval process now with the word 'A,' then SAM predicts Red and Round. If we provide the next input word as 'Round,' then the next prediction will be 'Orange,' whwreas the prediction will be 'Apple' if the next input is 'Red.' It is noted here that 'Round' will not predict 'Apple' here as 'A' already preceded it. This incremental learning process has been applied to both the languages and thus we created a region for both the languages for learning monolingual sequences. We also considered the dictionaries for both the languages. In our model, we consider English as L1 and Hindi to be L2. Each dictionary has an associated translation set for each of the words corresponding language. Monolingual learned phrases for each language are stored in the corresponding Learnt Monolingual Phrases (LMP)

block. We have considered phrase level translations as well for the efficient translation process. Thus, for each language, we have another region to store the phrase level translations to the other language. These regions, called Phrase Level Translations (PLT), are defined for both L1 and L2 in the model.

SAM is also able to predict for more than one step. We have considered the combinations of one step, 2 steps 4 steps and 8 steps learning processes in our experiments.

5) Translation Process: The translation process starts with the first word of the source language (L1). For each word, we retrieve the corresponding translation from the dictionary. Then, we check the PLT region of the target language whether a phrase is predicted with this word or not. We repeat this process for rest of the input words and keep on accepting the correctly predicted phrases present in PLT region. If predictions are not possible, then we go for 2, 4 or 8th step of prediction. This implies that PLT should be a resource-rich region to provide better and accurate translations. We are not using statistical methods to find the probabilities, but purely building the target sentence based on phrases present in PLT region of the target language.

B. Implementation of SAM model

SAM model is implemented by taking core algorithm from Numenta's CLA repository[3]. This repository is Python-based and requires a special package from Numenta called Nupic[4]. Changes proposed in Section 4.1 have been added to these open source codes for additional functional requirements.

The first requirement for English-Hindi Translation system is a parallel corpus for the languages. We have considered three different datasets for the experiments.

1) English-Hindi parallel corpus from Institute for Language, Cognition, and Computation, the University of Edinburgh [35].
2) Institute of Formal and Applied Linguistics (UFAL) at the Computer Science School, Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic [36].
3) Center for Indian Language Technology (CFILT), IIT Bombay. [37]

Data from ILCC, University of Edinburgh, contains translated sentences from Wikipedia. IIT Bombay and UFAL datasets contain the data from multiple disciplines. All these datasets are exhaustive with an abundant variety of words. Table I provides the information regarding the number of words and sentences in each of the dataset.

A complete training model for SAM is shown in Figure10. In this model, we obtain the data for both the languages and put them in two different files. Sentences obtained from these datasets are not directly usable for the designed system. Thus a series of preprocessing steps are required to make the data usable in the SAM algorithm. Initially, we need to remove

[3]http://numenta.org/implementations/
[4]https://github.com/numenta/nupic

TABLE I
DETAILS OF ILCC, UFAL AND CFILT HINDI DATASETS

	ILCC Dataset [35]	UFAL Dataset [36]	CFILT Dataset [37]
No of sentences	41,396	237,885	1,492,827
No of Words	245,675	1,048,297	20,601,012
No. Of Unique Words	18,342	82,673	125,619

unwanted symbols in the data and make all the letters small. Punctuation is removed from the sentences and sentences are ending with a line break. The extracted words are stored in the dictionary regions of respective languages.

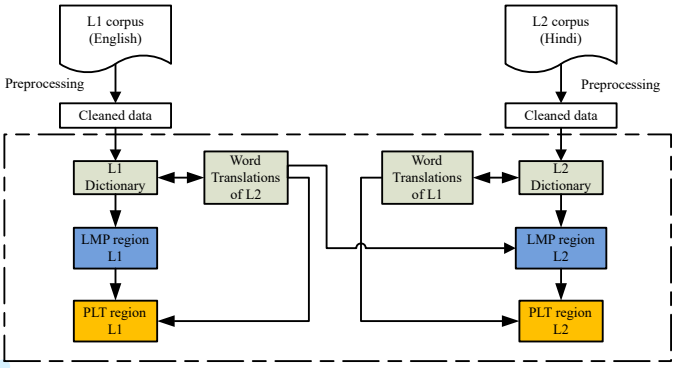

Fig. 10. Training model for proposed SAM algorithm

Next, after the preprocessing of th e word, we get streams of data for both the languages. As CLA doesn't support continuous storage of the data, these words have to be stored in Invariant Sparse Distribution (ISD) [38].

The total number of unique words present in the training region is around 125,000 for CFILT dataset. Thus, the size of this region is kept to be 400,000 cells, which is observed to be sufficient to hold all the words in the dataset. Typically a sentence consists of 10-12 words and thus at a given time only these number of cells in this region would be excited. Learning would be incremental in nature in each region, and the inference and predictions are passed to the next layer.

The proposed system has been implemented on an Nvidia GPU, consisting of NVIDIA Quadro K4200 graphics card. This GPU has 24 stacks with total 1344 CUDA cores. Every dataset is trained for 10 epochs. We have considered 1-step, 2-step, 4-step and 8-step learning processes to evaluate the quality of translations. Each cell of SAM requires 64x8 bits of data on a 64-bit processor. Thus we can store approximately 16 million unique sequences with 1 GB of RAM.

V. RESULTS AND DISCUSSION

Sequential Adaptive Learning (SAM) algorithm requires incremental steps to learn from data. Since we have used GPU for training, the training time was not huge, and we could train the complete model for CFILT dataset as well (which is having around 1.5 million sentence pairs).

TABLE II
BLEU SCORE BASED ON ADAPTIVE LEARNING IN SAM MODEL FOR DIFFERENT STEP LEARNING

No of learning steps	BLEU score for ILCC data	BLEU score for UFAL data	BLEU score for CFILT data
1	5.91	7.62	8.76
2	11.92	15.58	15.93
4	13.15	17.14	18.26
8	12.12	16.82	17.08

The adaptive learning results show interesting results. The BLEU score is low when we use only one step prediction in the training process. These results are related to the n-gram models of language models as well. As it has been shown over a period of time that language modeling and learning is better with 2-gram or 3-gram models as compared to 1-gram models. Our results also corporate the same with 2 and 4 steps of learning involved. We get slightly reduced BLEU score at 8 step learning as compared to 4 step learning. This reduction is mainly because the larger sequences are also not helpful in phrase construction and are not frequently available in the dataset. We also observed that even after a difference of around 6 times in the size of UFAL and CFIILT data set, the BLEU scores are quite comparable for both. This supports the hypothesis that CLA based systems can learn with a limited set of data as well. This is a major advantage as compared to the traditional SMT based systems, which requires huge datasets for better translation.

The results obtained from SAM based English-Hindi machine translation is comparable to the conventional statistical or phrase-based machine translation systems. One of the earliest SMT based system Anusaaraka [39] lacks the capability to handle complex sentences and doesn't perform at par with the latest MT systems such as Google and Bing Translator. Google had introduced recurrent neural network based Google Neural Machine Translator (GNMT) [40] in 2015 for French and German. This neural network based MT system have outclassed all SMT and PBMT systems for these two languages. The SAM-based system doesn't outperform GNMT (with a BLEU score of 38.20 for En-Fr), but it is showing comparable results when compared to Anusaaraka AnglaMT and Anglabharati [41]. In 2017, Singh et al. [42] have compared the performance on Neural Machine Translations on English to Hindi Machine Translation. These results showed that the best score of 22.44 was achieved by team XMUNLP in Hindi to English Translation Systems at WAT2017. Table III shows the BLEU score obtained for few of the contemporary machine translation systems.

TABLE III
BLEU SCORE COMPARISION FOR ENGLISH TO HINDI MT SYSTEMS WITH PROPOSED SAM MODEL

MT System	BLEU Score
Anusaaraka (PBMT)	21.18
Anusaaraka (Hierarchical)	21.10
AnglaMT	22.21
AnglaBharti	20.66
IBM English-Hindi SMT system, 2004 [43]	13.91
XMUNLP at WAT2017 [42]	22.44
SAM (Proposed work)	18.26

VI. CONCLUSION

In order to address the limitations faced by Statistical Phrase-based Machine translation systems, for demand of higher accuracy and large datasets, we have investigated into a new cognitive approach to translate from English to Hindi. The proposed approach is based on the structure of neocortex region of the human brain which processes speech and language. This hierarchical temporal memory based structure processes the information and passes the inference and predictions to the higher layers. In the proposed Sequential Adaptive Memory (SAM) based model, we have kept the words at the lowest level of the comprehension of language. Based on the inference from the lower layers, predictions are made for and stored in higher level layers. The interlayer relation between English and Hindi SAM structure helps in forming rules for translation between two languages. This approach shows significant improvement regarding data size required for the satisfactory translation process. Although for the highest level of accuracy, this approach would requires more layers in the networks and more sentences in the parallel corpus, yet this algorithm works fine for very little data set as well. The proposed approach SAM has been illustrated to perform at par with some of the conventional SMT and PBMT systems. We are currently working on fine-tuning the training of long and rare sentences using smaller data sets.

ACKNOWLEDGEMENT

This research is partially supported by our respective parent institutes, Malaviya National Institute of Technology (MNIT), Jaipur and The LNM Institute of Information Technology, Jaipur, India. We thank SMDP-C2S project, funded by MNIT Jaipur and also the LNMIIT's GPU services in simulations to obtain the results.

REFERENCES

[1] A. Lavie, S. Vogel, L. Levin, E. Peterson, K. Probst, A. F. Llitjós, R. Reynolds, J. Carbonell, and R. Cohen, "Experiments with a hindi-to-english transfer-based mt system under a miserly data scenario," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 2, no. 2, pp. 143–163, Jun. 2003. [Online]. Available: http://doi.acm.org/10.1145/974740.974747

[2] S. Saini and V. Sahula, "A survey of machine translation techniques and systems for indian languages," in *2015 IEEE International Conference on Computational Intelligence Communication Technology*, Feb 2015, pp. 676–681.

[3] S. Chand, "Empirical survey of machine translation tools," in *2016 Second International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, Sept 2016, pp. 181–185.

[4] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *CoRR*, vol. abs/1409.3215, 2014. [Online]. Available: http://arxiv.org/abs/1409.3215

[5] W. Yuan, J. Gao, and H. Suzuki, *An Empirical Study on Language Model Adaptation Using a Metric of Domain Similarity*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 957–968. [Online]. Available: http://dx.doi.org/10.1007/11562214_83

[6] V. MARIAN and M. SPIVEY, "Competing activation in bilingual language processing: Within- and between-language competition," *Bilingualism: Language and Cognition*, vol. 6, no. 2, p. 97115, 2003.
[7] M. Carl, A. L. Jakobsen, and K. T. Jensen, "Studying human translation behavior with user-activity data." in *NLPCS*, 2008, pp. 114–123.
[8] M. Simard and E. Macklovitch, "Studying the human translation process through the transsearch log-files," in *AAAI Symposium on Knowledge Collection from volunteer contributors*, Stanford, mar 2005, to appear Spring 2005.
[9] S. Grossberg, "The link between brain learning, attention, and consciousness," *Consciousness and cognition*, vol. 8, no. 1, pp. 1–44, 1999.
[10] ——, "Resonant neural dynamics of speech perception," *Journal of Phonetics*, vol. 31, no. 3-4, pp. 423–445, 2003.
[11] S. Grossberg and C. W. Myers, "The resonant dynamics of speech perception: Interword integration and duration-dependent backward effects." *Psychological review*, vol. 107, no. 4, p. 735, 2000.
[12] I. Kotseruba and J. K. Tsotsos, "A review of 40 years of cognitive architecture research: Core cognitive abilities and practical applications," *arXiv preprint arXiv:1610.08602*, 2016.
[13] R. Rubinoff and J. F. Lehman, "Real-time natural language generation in nl-soar," in *Proceedings of the Seventh International Workshop on Natural Language Generation*. Association for Computational Linguistics, 1994, pp. 199–206.
[14] P. Lindes and J. E. Laird, "Toward integrating cognitive linguistics and cognitive language processing," in *Proceedings of the 14th International Conference on Cognitive Modeling (ICCM)*, 2016.
[15] P. W. Schermerhorn, J. F. Kramer, C. Middendorff, and M. Scheutz, "Diarc: A testbed for natural human-robot interaction." in *AAAI*, 2006, pp. 1972–1973.
[16] D. George, *How the brain might work: A hierarchical and temporal model for learning and recognition*. Stanford University, 2008.
[17] D. Robinson, K. Leung, and X. Falco, "Spoken language identification with hierarchical temporal memories," 2009.
[18] L. C. Aiello and R. I. M. Dunbar, "Neocortex size, group size, and the evolution of language," *Current Anthropology*, vol. 34, no. 2, pp. 184–193, 1993. [Online]. Available: http://www.jstor.org/stable/2743982
[19] J. Panksepp, "Primal emotions and cultural evolution of language," *Emotion in Language: Theory–research–application*, vol. 10, p. 27, 2015.
[20] A. M. Thomson, "Neocortical layer 6, a review," *Frontiers in neuroanatomy*, vol. 4, no. 13, 2010.
[21] J. Hawkins and D. George, "Hierarchical temporal memory: Concepts, theory and terminology," Technical report, Numenta, Tech. Rep., 2006.
[22] R. McCall and S. Franklin, "Cortical learning algorithms with predictive coding for a systems-level cognitive architecture," in *Second annual conference on advances in cognitive systems poster collection*, 2013, pp. 149–66.
[23] J. Hawkins and S. Ahmad, "Why neurons have thousands of synapses, a theory of sequence memory in neocortex." *URL http://arxiv.org/abs/1511.00083*, Oct 2015.
[24] Y. Cui, C. Surpur, S. Ahmad, and J. Hawkins, "Continuous online sequence learning with an unsupervised neural network model," *CoRR*, vol. abs/1512.05463, 2015. [Online]. Available: http://arxiv.org/abs/1512.05463
[25] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 13, pp. 489 – 501, 2006, neural NetworksSelected Papers from the 7th Brazilian Symposium on Neural Networks (SBRN '04)7th Brazilian Symposium on Neural Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231206000385
[26] A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms - the numenta anomaly benchmark," *CoRR*, vol. abs/1510.03336, 2015. [Online]. Available: http://arxiv.org/abs/1510.03336
[27] S. Ahmad and S. Purdy, "Real-time anomaly detection for streaming analytics," *CoRR*, vol. abs/1607.02480, 2016. [Online]. Available: http://arxiv.org/abs/1607.02480
[28] B. A. Bobier and M. Wirth, "Content-based image retrieval using hierarchical temporal memory," in *Proceedings of the 16th ACM International Conference on Multimedia*, ser. MM '08. New York, NY, USA: ACM, 2008, pp. 925–928. [Online]. Available: http://doi.acm.org/10.1145/1459359.1459523
[29] D. Rozado, F. B. Rodriguez, and P. Varona, *Optimizing Hierarchical Temporal Memory for Multivariable Time Series*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 506–518. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15822-3_62
[30] J. Hawkins and D. George, "Numenta's htm community and open source project,https://numenta.org/resources/htmcorticallearningalgorithms.pdf," Technical report, Numenta, Tech. Rep., 2006.
[31] G. Brown and A. Sabry, "Reversible communicating processes," in *Proceedings Eighth International Workshop on Programming Language Approaches to Concurrency- and Communication-cEntric Software, PLACES 2015, London, UK, 18th April 2015.*, 2015, pp. 45–59. [Online]. Available: https://doi.org/10.4204/EPTCS.203.4
[32] L. Phillips and L. Pearl, "Bayesian inference as a cross-linguistic word segmentation strategy always learning useful things," in *Proceedings of the Computational and Cognitive Models of Language Acquisition and Language Processing Workshop*, 2014, pp. 9–13.
[33] L. Pearl and L. Phillips, "Utility-based evaluation metrics for models of language acquisition a look at speech segmentation," in *Proceedings of the 6th workshop on cognitive modeling and computational linguistics*, 2015, pp. 68–78.
[34] S. Saini, N. Gupta, S. Bhogal, S. Sharma, and V. Sahula, "Bayesian learner based language learnability analysis of hindi," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sept 2016, pp. 2089–2093.
[35] ILCC, "Indic multi-parallel corpus, http://homepages.inf.ed.ac.uk/miles/babel.html," Institute for Language, Cognition and Computation, University of Edinburgh,, Tech. Rep., 2011.
[36] O. Bojar, V. Diatka, P. Rychlỳ, P. Stranák, V. Suchomel, A. Tamchyna, and D. Zeman, "Hindencorp-hindi-english and hindi-only corpus for machine translation." in *LREC*, 2014, pp. 3550–3555.
[37] A. Kunchukuttan, P. Mehta, and P. Bhattacharyya, "The iit bombay english-hindi parallel corpus," *arXiv preprint arXiv:1710.02855*, 2017.
[38] J. Yang, K. Yu, and T. Huang, "Supervised translation-invariant sparse coding," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3517–3524.
[39] A. Bharati, V. Chaitanya, A. P. Kulkarni, and R. Sangal, "Anusaaraka: Machine translation in stages," *CoRR*, vol. cs.CL/0306130, 2003. [Online]. Available: http://arxiv.org/abs/cs.CL/0306130
[40] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144
[41] K. Sachdeva, R. Srivastava, S. Jain, and D. M. Sharma, "Hindi to english machine translation: Using effective selection in multi-model smt." in *LREC*, 2014, pp. 1807–1811.
[42] S. Singh, R. Panjwani, A. Kunchukuttan, and P. Bhattacharyya, "Comparing recurrent and convolutional architectures for english-hindi neural machine translation," in *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, 2017, pp. 167–170.
[43] R. Udupa and T. A. Faruquie, "An english-hindi statistical machine translation system," in *International Conference on Natural Language Processing*. Springer, 2004, pp. 254–262.

Sandeep Saini received his B.Tech. degree in Electronics and Communication Engineering from International Institute of Information Technology, Hyderabad, India in 2008. He completes his M.S. from the same institute in 2010. He is pursuing his Ph.D. from Malviya National Institute of Technology, Jaipur India.

He has been working at the LNM Institute of Information Technology, Jaipur as an Assistant Professor from 2011 onward. His research interests are in the areas of Natural Language Processing, cognitive modeling of language learning models. Sandeep is a member of IEEE from 2009 and active member of ACM as well.

Vineet Sahula obtained his Bachelors in Electronics (honors) from Malaviya National Institute of Technology, Jaipur, India in 1987 and Masters in Integrated Electronics; Circuits from the Indian Institute of Technology, Delhi in 1989, and the Ph.D. Degree from Department of Electrical engineering, Indian Institute of Technology, Delhi in 2001.

In 1990, he joined as faculty member at Malaviya National Institute of Technology, Jaipur, where he is currently Head of the Department of Electronics and Communications Engineering. He has 80+ research papers in reputed journals and conference proceedings to his credit. His research interests are into system level design, cognitive architectures, Cognitive aspects in Language Processing, modeling and synthesis for analog and digital systems and computer aided design of for VLSI and MEMS.

Dr. Sahula has served on the Technical programme committee of the VLSI Design and Test Symposium, India from 1998 to 2013. He has also served on organizing committee of Embedded Systems Week, Oct. 2014 Delhi and as fellowship-chair of 22 nd IEEE International Conference on VLSI Design, India in 2009. He is a senior member of IEEE, Life Fellow of IETE and IE, Life member of IMAPS and member of ACM SIGDA.