# Condensed Memory Networks for Clinical Diagnostic Inferencing

**Aaditya Prakash**[*]
Brandeis University, MA
aprakash@brandeis.edu

**Siyuan Zhao**
Worcester Polytechnic Institute, MA
szhao@wpi.edu

**Sadid A. Hasan, Vivek Datla, Kathy Lee, Ashequl Qadir, Joey Liu, Oladimeji Farri**
[*]Artificial Intelligence Laboratory, Philips Research North America, Cambridge, MA
{firstname.lastname,kathy.lee_1,dimeji.farri}@philips.com

## Abstract

Diagnosis of a clinical condition is a challenging task, which often requires significant medical investigation. Previous work related to diagnostic inferencing problems mostly consider multivariate observational data (e.g. physiological signals, lab tests etc.). In contrast, we explore the problem using free-text medical notes recorded in an electronic health record (EHR). Complex tasks like these can benefit from structured knowledge bases, but those are not scalable. We instead exploit raw text from Wikipedia as a knowledge source. Memory networks have been demonstrated to be effective in tasks which require comprehension of free-form text. They use the final iteration of the learned representation to predict probable classes. We introduce condensed memory neural networks (C-MemNNs), a novel model with iterative condensation of memory representations that preserves the hierarchy of features in the memory. Experiments on the `MIMIC-III` dataset show that the proposed model outperforms other variants of memory networks to predict the most probable diagnoses given a complex clinical scenario.

## Introduction

Clinicians perform complex cognitive processes to infer the probable diagnosis after observing several variables such as the patient's past medical history, current condition, and various clinical measurements. The cognitive burden of dealing with complex patient situations could be reduced by having an automated assistant provide suggestions to physicians of the most probable diagnostic options for optimal clinical decision-making.

Some work has been done in building Artificial Intelligence (AI) systems that can support clinical decision making (Lipton et al. 2015; Choi, Bahadori, and Sun 2015; Choi et al. 2016). These works have primarily focused on the use of various biosignals as features. EHRs typically store such structured clinical data (e.g. physiological signals, vital signs, lab tests etc.) about the patients' clinical encounters in addition to unstructured textual notes that contain a complete picture of the associated clinical events. Structured clinical data generally contain raw signals without much interpretation, whereas unstructured free-text clinical notes contain detailed description of the overall clinical scenario.

In this paper, we explore the discriminatory capability of the unstructured free-text clinical notes to correctly infer the most probable diagnoses from a complex clinical scenario. In the Clinical Decision Support track[1] of the recent Text REtrieval Conference (TREC) series, clinical diagnostic inferencing from unstructured free texts has been shown to play a significant role in improving the accuracy of relevant biomedical article retrieval (Hasan et al. 2014; 2015; 2016). We also explore the use of an external knowledge source like Wikipedia from which the model can extract relevant information, such as signs and symptoms for various diseases. Our goal is to combine such an external clinical knowledge source with the free-text clinical notes and use the learning capability of memory networks to correctly infer the most probable diagnosis.

Memory Networks (MemNNs) (Sukhbaatar et al. 2015; Weston, Chopra, and Bordes 2014) are a class of models which contain an external memory and a controller to read from and write to the memory. Memory Networks read a given input source and a knowledge source several times (hops) while updating an internal memory state. The memory state is the representation of relevant information from the knowledge base optimized to solve the given task. This allows the network to remember useful features. The notion of neural networks with memory was introduced to solve AI tasks that require complex reasoning and inferencing. These models have been successfully applied in the Question Answering domain on datasets like *bAbi* (Weston et al. 2015), *MovieQA* (Tapaswi et al. 2015), and *WikiQA* (Sukhbaatar et al. 2015; Miller et al. 2016). Memory networks are harder to train than traditional networks and they do not scale easily to a large memory. End-to-End Memory Networks (Sukhbaatar et al. 2015) and Key-Value Memory Networks (KV-MemNNs) (Miller et al. 2016) try to solve these problems by training multiple hops over memory and compartmentalizing memory slots into hashes, respectively.

When the memory is large, hashing can be used to selectively retrieve only relevant information from the knowledge base, however not much work has been done to improve the information content of the memory state. If the network were trained for factoid question answering, the memory

[1]http://www.trec-cds.org/

state might be trained to represent relevant facts and relations from the underlying domain. However, for real world tasks, a large amount of memory is required to achieve state-of-the-art results. In this paper, we introduce Condensed Memory Networks (C-MemNNs), an approach to efficiently store condensed representations in memory, thereby maximizing the utility of limited memory slots. We show that a condensed form of memory state which contains some information from earlier hops learns efficient representation. We take inspiration from human memory for this model. Humans can learn new information and yet remember even very old memories as abstractions. We also experiment with a simpler form of knowledge retention from previous hops by taking a weighted **average** of memory states from all the hops (A-MemNN). Even this simpler alternative which does not add any extra parameter is able to outperform standard memory networks. Empirical results on the `MIMIC-III` dataset reveal that C-MemNN improves the accuracy of clinical diagnostic inferencing over other classes of memory networks. To the best of our knowledge, this is the first empirical study to classify diagnosis from EHR free-text clinical notes using memory networks.

## Related Work

### Memory Networks

Memory Networks (MemNNs) (Weston, Chopra, and Bordes 2014) and Neural Turing Machines (NTMs) (Graves, Wayne, and Danihelka 2014) are the two classes of neural network models with an external memory component. MemNN stores all information (e.g. knowledge base, background context) into the external memory, assigns a relevance probability to each memory slot using content-based addressing schemes, and reads contents from each memory slot by taking their weighted sum. End-to-End Memory Networks introduced multi-hop training (Sukhbaatar et al. 2015) and do not require strong supervision unlike MemNN. Key-Value Memory Networks (Miller et al. 2016) have a key-value paired memory and are built upon MemNN. The key-value paired structure is a generalized way of storing content in the memory. The contents in the key-memory are used to calculate the relevance probabilities whereas the contents in the value-memory are read into the model to help make the final prediction.

In contrast to MemNN, which uses a content-based mechanism to access the external memory, the NTM controller uses both content- and location-based mechanisms. The fundamental difference between these models is that MemNN does not have a mechanism for the content to be changed in the memory, while NTM can modify the content in each episode. This makes MemNN easier to train compared to NTM.

Recently, there has been an attempt to incorporate longer contextual memory into the basic Recurrent Neural Networks (RNNs) framework. Stack-Augmented RNN (Joulin and Mikolov 2015) proposes interconnecting RNN modules using a push-down stack in order to learn long-term dependencies. They are able to reproduce complicated sequence patterns. Chung, Ahn, and Bengio (2016) explore multi-scale RNN, which is able to learn a latent hierarchical structure by using temporal representation at different time-scales. These methods are well-suited for learning long-term temporal dependencies, but do not scale well to large memory. Hierarchical Memory Networks (Chandar et al. 2016) study the use of maximum inner product search (MIPS) to store memory slots in a hierarchy. Our aim is to improve the efficiency and knowledge density of the standard memory slots by exploring a hierarchy of internal memory representations over multiple hops.

Another related class of models are the attention-based neural networks. These models are trained to learn an attention mechanism so that they can focus on the important information on a given input. Applying an attention mechanism on the machine reading comprehension task (Hermann et al. 2015; Dhingra et al. 2016; Cui et al. 2016; Sordoni, Bachman, and Bengio 2016) has shown promising results. In tasks where inferencing is governed by the input source e.g. sentence-level machine translation (Bahdanau, Cho, and Bengio 2014), image caption generation (Xu et al. 2015), and visual question answering (Lu et al. 2016), the use of attention-based models has proven to be very effective. As attention is learned by the iterative finding of the highly-activated input regions, this is not feasible for a large-scale external memory; however, more research is required in order to achieve attention over memory.

### Neural Networks for Clinical Diagnosis

The application of neural networks to medical tasks dates back more than twenty years (Baxt 1990). The recent success of deep learning has drawn broader interest in building AI systems to support clinical decision making. Lipton et al. (2015) train Long Short-Term Memory Networks (LSTMs) to classify 128 diagnoses from 13 frequently but irregularly sampled clinical measurements extracted from the EHR. DoctorAI (Choi, Bahadori, and Sun 2015) and RETAIN (Choi et al. 2016) also use time series data for diagnosis classification. Similar to these works, we formulate the problem as multi-label classification, since each medical note might be associated with multiple diagnoses. However, there are two important differences between our work and the previous work. We only consider discharge notes for our experiments, which are unstructured free-texts and do not contain time series data, while they rely on time series datasets where each time series has a fixed number of clinical measurements. Moreover, we train our model in an end-to-end fashion and do not extract any hand-engineered features from the notes, while they resample all time series data to an hourly rate and fill in the gaps created by window-based resampling in clinical measurements. We propose the use of memory networks instead of LSTMs to classify the diagnoses since the memory component provides the flexibility to learn from an external knowledge source.

## Dataset

`MIMIC-III` (Multiparameter Intelligent Monitoring in Intensive Care) (Johnson et al. 2016) is a large freely-available clinical database. It contains physiological signals and vari-

**Medical Note** (partially shown)

Date of Birth: [**2606–2–28**] Sex: M
Service: Medicine
**Chief Complaint:**
Admitted from rehabilitation for hypotension (systolic blood pressure to the 70s) and decreased urine output. **History of present illness:**
The patient is a 76-year-old male who had been hospitalized at the [**Hospital1 3007**] from [**8–29**] through [**9–6**] of 2002 after undergoing a left femoral-AT bypass graft and was subsequently discharged to a rehabilitation facility.
On [**2682–9–7**], he presented again to the [**Hospital1 3087**] after being found to have a systolic blood pressure in the 70s and no urine output for 17 hours.

**Diagnosis**

Cardiorespiratory arrest.
Non-Q-wave myocardial infarction.
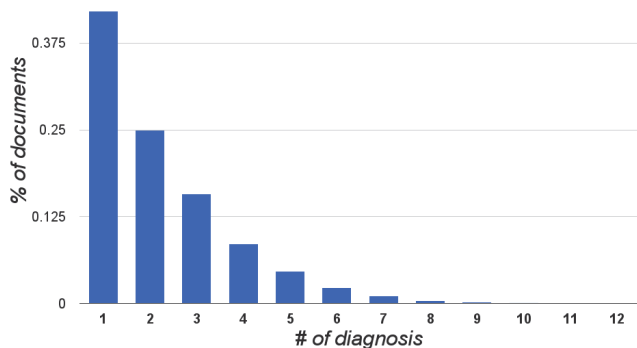Acute renal failure.

Table 1: An example `MIMIC-III` note.



Figure 1: Distribution of number of diagnosis in a note.

ous measurements captured from patient monitors, and comprehensive clinical data obtained from hospital medical information systems for over $58K$ Intensive Care Unit (ICU) patients. We use the *noteevents* table from `MIMIC-III`: v1.3, which contains the unstructured free-text clinical notes for patients. We use 'discharge summaries', instead of 'admission notes', as former contains actual ground truth and free-text. Since discharge summaries are written after diagnosisdecision, we sanitize the notes by removing any mention of class-labels in the text.

As shown in Table 1, medical notes contain several details about the patient but the sections are not uniform. We do not separate the sections other than the *DIAGNOSIS*, which is our label. There are multiple labels (diagnoses) for a given note, and a note can belong to multiple classes of diagnoses, thus we formulate our task as a multiclass-multilabel classification problem. The number of diagnoses per note is also not consistent and shows a long tail (Figure 1). We have taken measures to counteract these issues, which are discussed in the *Memory addressing* section.

**Cardiac arrest**

Cardiac arrest is a sudden stop in effective blood circulation due to the failure of the heart to contract effectively or at all[1]. A cardiac arrest is different from (but may be caused by) a myocardial infarction (also known as a heart attack), where blood flow to the muscle of the heart is impaired such that part or all of the heart tissue dies...
**Signs and symptoms**
Cardiac arrest is sometimes preceded by certain symptoms such as fainting, fatigue, blackouts, dizziness, chest pain, shortness of breath, weakness, and vomiting. The arrest may also occur with no warning ...

Table 2: Partially shown example of a relevant Wikipedia page.

Some diagnoses are less frequent in the data set. Without enough training instances, a model is not able to learn to recognize these diagnoses. Therefore, we experiment with a varying number of labels in this work (see details in the *Experiments* section).

## Knowledge Base

We use Wikipedia pages (see Table 2) corresponding to the diagnoses in the `MIMIC-III` notes as our external knowledge source. WikiProject Medicine is dedicated to improving the quality of medical articles on Wikipedia and the information presented in these pages are generally shown to be reliable (Trevena 2011). Since some diagnosis terms from `MIMIC-III` don't always match a Wikipedia title, we use the Wikipedia API with the diagnoses as the search terms to find the most relevant Wikipedia pages. In most cases we find an exact match while in the rest we pick the most relevant page. We use the first paragraph and the paragraphs corresponding to the *Signs and symptoms* sections for our experiments. In cases where such a section is not available, we use the second and third paragraphs of the page. This happens for the obscure diseases, which have a limited content.

## Condensed Memory Networks

The basic structure of our model is inspired by MemNN. Our model tries to learn memory representation from a given knowledge base. Memory is organized as some number of slots $m_1, \ldots m_t$. For the given input text i.e. medical notes $x_1, \ldots, x_n$, the external knowledge base (wiki pages, wiki titles) $(k_1, v_1), (k_2, v_2), \ldots, (k_m, v_m)$, and the diagnoses of those notes $y$, we aim to learn a model $\mathcal{F}$ such that

$$\mathcal{F}(x_n, (k_m, v_m)) = \hat{y} \rightarrow y \qquad (1)$$

We break down this function $\mathcal{F}$, in four parts *I, G, O, R* which are the standard components of Memory Networks.

- **I: Input memory representation** is the transformation of the input $x$ to some internal representation $u$ using learned weights $\boldsymbol{B}$. This is the internal state of the model and is similar to the hidden state of RNN-based models. In this
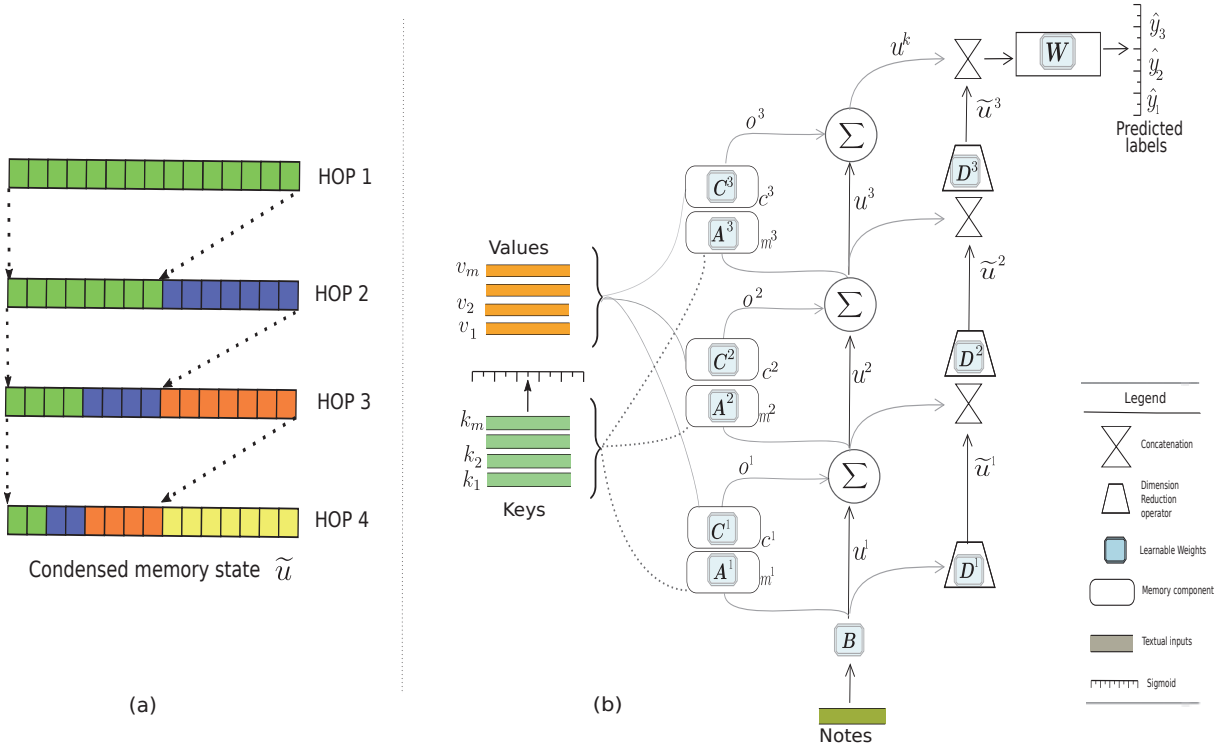
Figure 2: (a) Abstract view of transformation of memory representation over multiple hops. (b) Structural overview of end-to-end model for condensed memory networks.

paper, we propose the addition of a condensed memory state $\widetilde{u}$, which is obtained via the iterative concatenation of successively lower dimensional representations of the input memory state $u$.

- **G: Generalization** is the process of updating the memory. MemNN updates all slots in the memory, but this is not feasible when the size of the knowledge source is very large. Therefore, we organize the memory as key-value pairs as described in Miller et al. (2016). We use hashing to retrieve a small portion of keys for the given input.

- **O: Output memory representation** is the transformation of the knowledge $(k, v)$ to some internal representation $m$ and $c$. While End-to-End MemNN uses an embedding matrix to convert memories to learned feature space, our model uses a two-step process because we represent wiki-titles and wikipages as different learned spaces. We learn matrix $A$ to transform wikipages (keys) and $C$ to transform wiki-titles (values). Our choice of wikipages as keys and wiki-titles as values is deliberate – the input "medical notes" more closely match the text of the wikipages and the diagnoses more closely match the wiki-titles. This allows for a better mapping of features; our empirical results validate this idea.

Let $k$ represents the hop number. The output memory representation is obtained by:

$$o^k = \sum_i \text{Addressing}(u^k, m_i^k) \cdot c_i^k \qquad (2)$$

where *Addressing* is a function which takes the given input memory state $u$ and provides the relevant memory representation $m$.

- **R: Response** combines the internal state $u$, internal condensed state $\widetilde{u}$ and the output representation $o$ to provide the predicted label $\hat{y}$. We sum $u$ and $o$ and then take the dot product with another learned matrix $W$. We then concatenate this value with condensed memory state $\widetilde{u}$. This value is then passed through sigmoid to obtain the likelihood of each class. We use sigmoid instead of softmax in order to obtain multiple predicted labels, $\hat{y}_1, \ldots \hat{y}_r$ among possible $R$ labels. Both the memory states $u$ and $\widetilde{u}$ are computed as:

$$u^{k+1} = u^k + o^k \qquad (3)$$

$$\widetilde{u}^{k+1} = u^{k+1} \oplus D_1 \cdot \widetilde{u}^k \qquad (4)$$

where $\oplus$ denotes concatenation of vectors.

Our major contribution to memory networks is the use of condensed memory state $\widetilde{u}$ in combination with input memory state $u$ to do the inference. As shown in Figure 2(a), $\widetilde{u}$ is transformed to include the information of previous hops, but in lower dimensional feature space. This leads to a longer term memory representation, which is better able to represent hierarchy in memory. The class prediction is obtained using:

3277

$$\hat{y}_r = \arg\max_{r \epsilon R} \frac{1}{1 + e^{-1*(\widetilde{u}^{k+1} \cdot \boldsymbol{W})}} \qquad (5)$$

## Network Overview

Figure 2(b) shows the overview of the C-MemNN structure. The input $x$ is converted to internal state $u^1$ using the transformation matrix $\boldsymbol{B}$. This is combined with memory key slots $m^1$ using matrix $\boldsymbol{A}$. Memory addressing is used to retrieve the corresponding memory value $c^1$. This value is transformed using matrix $\boldsymbol{C}$ to output memory representation $o^1$. In parallel, memory state $u$ is condensed to half of its original dimension using the transformation matrix $\boldsymbol{D}$. If $u$ is of size $1 \times K$ then $\boldsymbol{D}$ is of size $K \times \frac{K}{2}$. We call this reduced representation of $u$ the condensed memory state, $\widetilde{u}$. This is the end of the first hop. This process is then repeated for a desired number of hops. After each hop, the condensed memory state $\widetilde{u}$ becomes the concatenation of its previous state and its current state, each reduced to half of its original dimension.

## Averaged Memory Networks

In C-MemNN, the transformation of $\widetilde{u}$ at every hop adds more parameters to the model, which is not always desirable. Thus, we also present a simpler alternative model, which we call A-MemNN, to capture hierarchy in memory representation without adding any learned parameters. In this alternative model, we compute the weighted average of $\widetilde{u}$ across multiple hops. Instead of concatenating previous $\widetilde{u}$ values, we simply maintain an exponential moving average from different hops:

$$\widetilde{u}^{k+1} = \widetilde{u}^k + \frac{\widetilde{u}^{k-1}}{2} + \frac{\widetilde{u}^{k-2}}{4} + \dots \qquad (6)$$

where, the starting condensed memory state is same as the input memory state $\widetilde{u}^1 = u^1$.

## Memory Addressing

Key-Value addressing as described in KV-MemNN uses softmax on the product of question embeddings and retrieved keys to learn a relevance probability distribution over memory slots. The representation obtained is then the sum of the output memory representation $o$, weighted by those probability values. KV-MemNN was designed to pick the single most relevant answer given a set of candidate answers. The use of softmax significantly decreases the estimated relevance of all but the most probable memory slot. This presents a problem for multi-label classification in which several memory slots may be relevant for different target labels. We experimented with changing softmax to sigmoid to alleviate this problem, but this was not sufficient to allow the incorporation of the condensed form of the internal state $u$ arising from earlier hops. Thus, we explore a novel alternate addressing scheme, which we call gated addressing. This addressing method uses a multi-layer feed-forward neural network (FNN) with a sigmoid output layer to determine the appropriate weights for each memory slot. The network calculates a weight value between 0 and 1 for each memory

slot, and a weighted sum of memory slots is obtained as before.

## Document Representation

There are a variety of models to represent knowledge in key-value memories, and the choice of a model can have an impact on the overall performance. We use a simple bag-of-words (BoW) model which transforms each word $w_{ij}$ in the document $d_i = w_{i1}, w_{i2}, w_{i3}, \dots, w_{in}$ to embeddings, and sums these together to obtain the vectors $\Phi(d_i) = \sum_j A w_{ij}$, with $A$ being the embedding matrix. Medical notes, memory keys and memory values are all represented in this way.

# Experiments

The distribution of diagnoses in our training data has a very long tail. There are 4,186 unique diagnoses in MIMIC-III discharge notes. However, many diagnoses (labels) occur in only a single note. This is not sufficient to efficiently train on these labels. The 50 most-common labels cover 97% of the notes and the 100 most-common labels cover 99.97%. Thus, we frame this task as multi-label classification for top-N labels. We present experiments for both the 50 most-common and 100-most common labels. For all experiments, we truncate both notes and wiki-pages to 600 words. We reduce the trained vocabulary to 20K after removing common stopwords. We use a common dimension of 500 for all embedding matrices. We use a memory slot of dimension 300. A smaller embedding of dimension 32 is used to represent the wiki-titles.

We present experiments for end-to-end memory networks (Sukhbaatar et al. 2015), Key-Value Memory Networks (KV-MemNNs) (Miller et al. 2016) and our models, Condensed Memory Networks (C-MemNN) and Averaged Memory Networks (A-MemNN). We separately train models for three, four and five hops. The strength of our model is the ability to make effective use of several memory hops, and so we do not present results for one or two hops. We did not consider standard text classification models like CNN, LSTM or Fasttext (Joulin et al. 2016) as they do not allow incorporating external memory, which is one of the main goals of this paper.

## Training

We use Adam (Kingma and Ba 2014) stochastic gradient descent for optimizing the learned parameters. The learning rate is set to 0.001 and batch size for each iteration to 100 for all models. For the final prediction layer, we use a fully connected layer on top of the output from equation 5 with a sigmoid activation function. The loss function is the sum of cross entropy from prediction labels and prediction memory slots using addressing schema. Complexity of the model was penalized by adding $L2$ regularization to the cross entropy loss function. We use dropout (Srivastava et al. 2014) with probability 0.5 on the output-to-decision sigmoid layer and limit the norm of the gradients to be below 20. Models are trained on $80\%$ of the data and validated on $10\%$. The remaining $10\%$ is used as test set which is evaluated only once across all experiments with different models.

|        |            | # classes = 50 | | | # classes = 100 | | |
|--------|------------|-----------------|---------------------------|-----------------|-----------------|---------------------------|-----------------|
| # Hops | Model      | AUC (macro) ↑ | Average Precision @5 ↑ | Hamming Loss ↓ | AUC (macro) ↑ | Average Precision @5 ↑ | Hamming Loss ↓ |
| 3 | End-to-End | 0.759 | 0.32 | 0.06 | 0.664 | 0.23 | 0.15 |
|   | KV MemNN   | 0.761 | 0.36 | **0.05** | 0.679 | 0.24 | 0.14 |
|   | A-MemNN    | 0.762 | 0.36 | 0.06 | 0.675 | 0.23 | 0.14 |
|   | C-MemNN    | **0.785** | **0.39** | **0.05** | **0.697** | **0.27** | **0.12** |
| 4 | End-to-End | 0.760 | 0.33 | 0.04 | 0.672 | 0.24 | 0.15 |
|   | KV MemNN   | 0.776 | 0.35 | 0.04 | 0.683 | 0.24 | 0.13 |
|   | A-MemNN    | 0.775 | 0.37 | 0.03 | 0.689 | 0.23 | 0.11 |
|   | C-MemNN    | **0.795** | **0.42** | **0.02** | **0.705** | **0.27** | **0.09** |
| 5 | End-to-End | 0.761 | 0.34 | 0.04 | 0.683 | 0.25 | 0.14 |
|   | KV MemNN   | 0.775 | 0.36 | 0.03 | 0.697 | 0.25 | 0.11 |
|   | A-MemNN    | 0.804 | 0.40 | 0.02 | 0.720 | 0.29 | 0.11 |
|   | C-MemNN    | **0.833** | **0.42** | **0.01** | **0.767** | **0.32** | **0.05** |

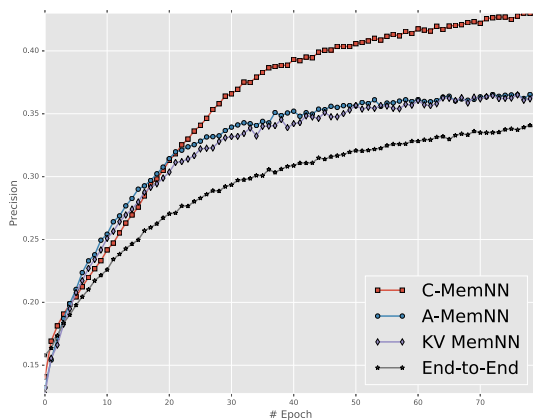Table 3: Evaluation results of various memory networks on `MIMIC-III` dataset.



Figure 3: Precision@5 plot for various models on validation data (at 4 hops).

## Results and Analysis

We present experiments in which performance is evaluated using three metrics: the area under the ROC curve (AUC), the average precision over the top five predictions, and the hamming loss. The AUC is calculated by taking unweighted mean of the AUC values for each label - this is also known as the macro AUC. Average precision over the top five predictions is reported because it is a relevant metric for real world applications. Hamming loss is reported instead of accuracy because it is a better measure for multi-label classification (Elisseeff and Weston 2001).

As shown in Table 3, C-MemNN is able to exceed the results of various other memory networks across all experiments. The improvement is more pronounced with a higher number of memory hops. This is because of the learning saturation of vanilla memory networks over multiple hops. While A-MemNN has better results for higher hops it does not improve upon KV-MemNN at lower hops. The strength of our model lies at higher hops, as the condensed memory state $\widetilde{u}$ after several hops contains more information than the same size input memory state $u$. Across all models, results improve as the number of hops increases, although with diminishing returns. The AUC value of C-MemNN with five memory hops for 100 labels is higher than the AUC value for End-to-End models trained only for three hops, which shows efficient training of higher hops produces good results.

Most documents do not have five labels (Figure 1) and thus precision obtained for five predictions is poor across all models. Hamming Loss correlates very well with other metrics along with the cross-entropy loss function, which was used for training.

Our model has 30% more parameters than standard Memory Networks ($D$ for every pair of $A$ and $C$). Figure 3 shows that adding more memory does not lead to overfitting. Training time of our model for GPU implementation is same as other memory networks, as condensed memory state ($\widetilde{u}$) is trained almost in parallel with standard memory state ($u$) for every hop.

## Conclusion and Future Work

Weston, Chopra, and Bordes (2014) discussed the possibility of a better memory representation for complex inferencing tasks. We achieved a better memory representation by condensing the previous hops in a novel way to obtain a hierarchical representation of the internal memory. We have shown the efficacy of the proposed memory representation for clinical diagnostic inferencing from raw textual data. We discussed the limitations of memory networks for multi-label classification and explored gated addressing to achieve a better mapping between the clinical notes and the memory

slots. We have shown that training multiple hops with condensed representation is helpful, but this is still computationally expensive. We plan to investigate asynchronous memory updating, which will allow for faster training of memory networks. In the future, we will explore other knowledge sources and the recently proposed word vectors for the biomedical domain (Chiu et al. 2016).

## Acknowledgments

## References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baxt, W. G. 1990. Use of an artificial neural network for data analysis in clinical decision-making: the diagnosis of acute coronary occlusion. *Neural computation* 2(4):480–489.

Chandar, S.; Ahn, S.; Larochelle, H.; Vincent, P.; Tesauro, G.; and Bengio, Y. 2016. Hierarchical memory networks. *arXiv preprint arXiv:1605.07427*.

Chiu, B.; Crichton, G.; Korhonen, A.; and Pyysalo, S. 2016. How to train good word embeddings for biomedical nlp. *ACL 2016* 166.

Choi, E.; Bahadori, M. T.; and Sun, J. 2015. Doctor ai: Predicting clinical events via recurrent neural networks. *arXiv preprint arXiv:1511.05942*.

Choi, E.; Bahadori, M. T.; Schuetz, A.; Stewart, W. F.; and Sun, J. 2016. Retain: Interpretable predictive model in healthcare using reverse time attention mechanism. *CoRR* abs/1608.05745.

Chung, J.; Ahn, S.; and Bengio, Y. 2016. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*.

Cui, Y.; Chen, Z.; Wei, S.; Wang, S.; Liu, T.; and Hu, G. 2016. Attention-over-attention neural networks for reading comprehension. *CoRR* abs/1607.04423.

Dhingra, B.; Liu, H.; Cohen, W. W.; and Salakhutdinov, R. 2016. Gated-attention readers for text comprehension. *CoRR* abs/1606.01549.

Elisseeff, A., and Weston, J. 2001. A kernel method for multi-labelled classification. In *NIPS*, 681–687.

Graves, A.; Wayne, G.; and Danihelka, I. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Hasan, S. A.; Zhu, X.; Dong, Y.; Liu, J.; and Farri, O. 2014. A hybrid approach to clinical question answering. In *TREC*.

Hasan, S. A.; Ling, Y.; Liu, J.; and Farri, O. 2015. Using neural embeddings for diagnostic inferencing in clinical question answering. In *TREC*.

Hasan, S. A.; Zhao, S.; Datla, V.; Liu, J.; Lee, K.; Qadir, A.; Prakash, A.; and Farri, O. 2016. Clinical question answering using key-value memory networks and knowledge graph. In *TREC*.

Hermann, K. M.; Kociský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In *NIPS*.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Lehman, L.-w. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data* 3.

Joulin, A., and Mikolov, T. 2015. Inferring algorithmic patterns with stack-augmented recurrent nets. In *NIPS*, 190–198.

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Lipton, Z. C.; Kale, D. C.; Elkan, C.; and Wetzel, R. C. 2015. Learning to diagnose with LSTM recurrent neural networks. *CoRR* abs/1511.03677.

Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2016. Hierarchical question-image co-attention for visual question answering. *arXiv preprint arXiv:1606.00061*.

Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.; Bordes, A.; and Weston, J. 2016. Key-value memory networks for directly reading documents. *CoRR* abs/1606.03126.

Sordoni, A.; Bachman, P.; and Bengio, Y. 2016. Iterative alternating neural attention for machine reading. *CoRR* abs/1606.02245.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Sukhbaatar, S.; szlam, a.; Weston, J.; and Fergus, R. 2015. End-to-end memory networks. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *NIPS*. Curran Associates, Inc. 2440–2448.

Tapaswi, M.; Zhu, Y.; Stiefelhagen, R.; Torralba, A.; Urtasun, R.; and Fidler, S. 2015. Movieqa: Understanding stories in movies through question-answering. *arXiv preprint arXiv:1512.02902*.

Trevena, L. 2011. Wikiproject medicine. *BMJ* 342:d3387.

Weston, J.; Bordes, A.; Chopra, S.; Rush, A. M.; van Merriënboer, B.; Joulin, A.; and Mikolov, T. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Weston, J.; Chopra, S.; and Bordes, A. 2014. Memory networks. *CoRR* abs/1410.3916.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R. S.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044* 2(3):5.