DATA SCIENCE INTERVIEW PREPARATION

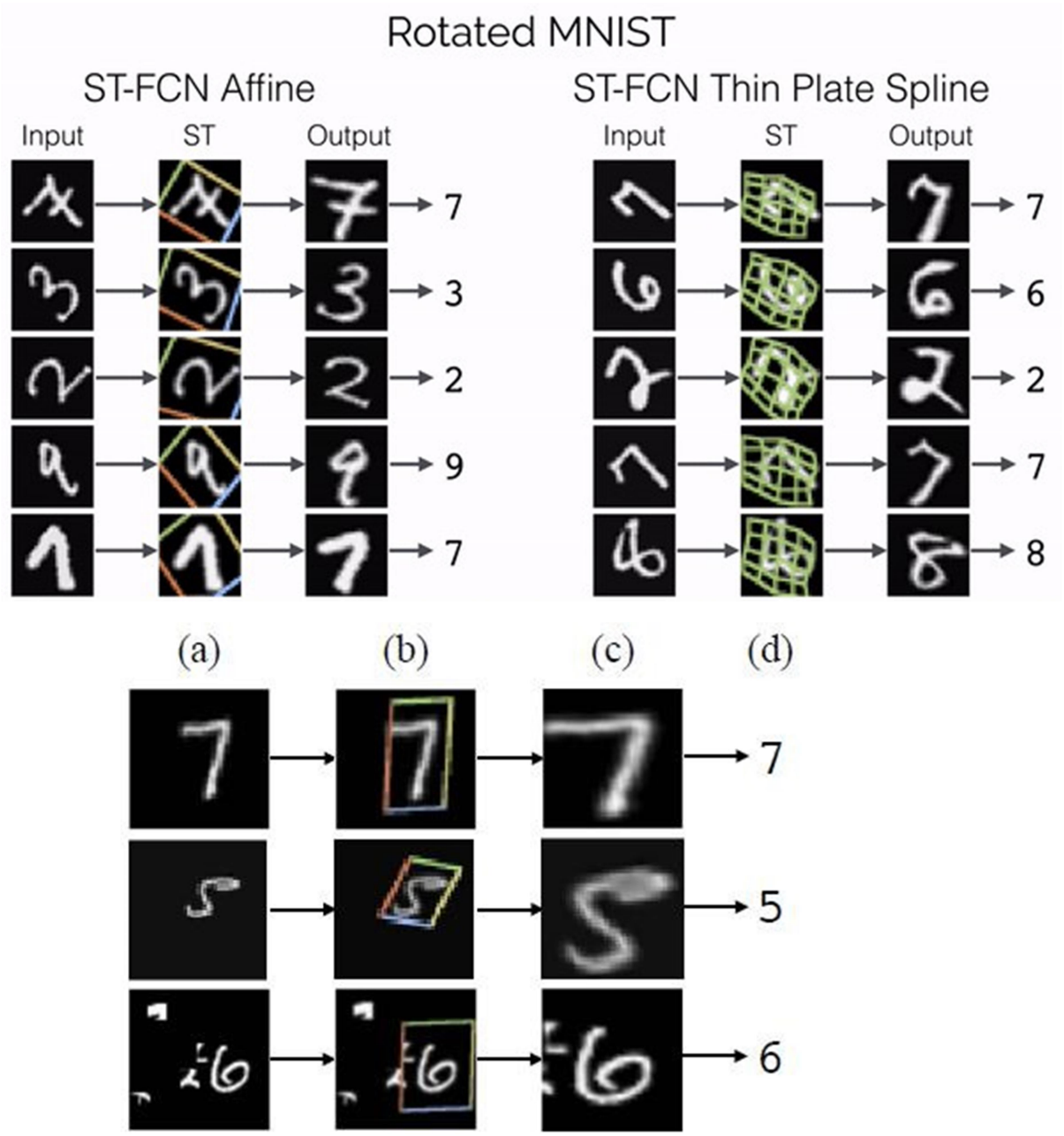
(30 Days of Interview Preparation)

Day24

Q1.What is STN?

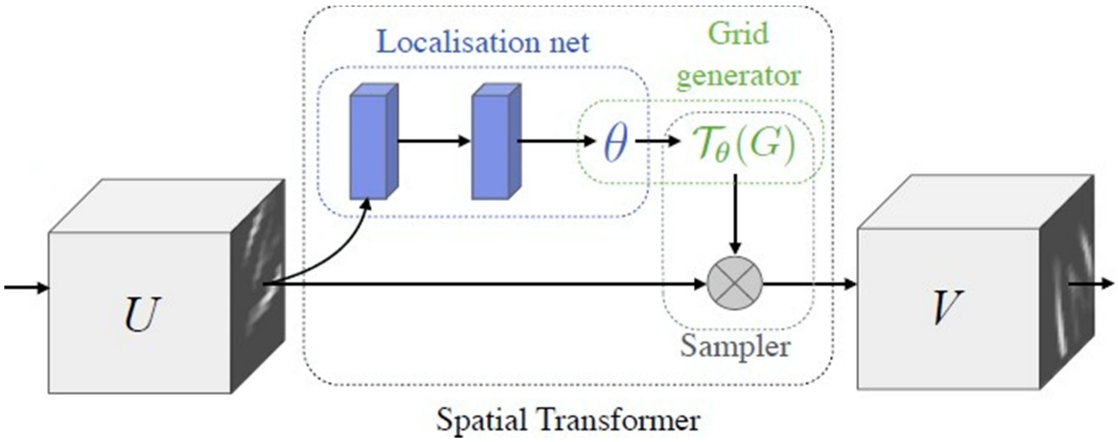
Answer:

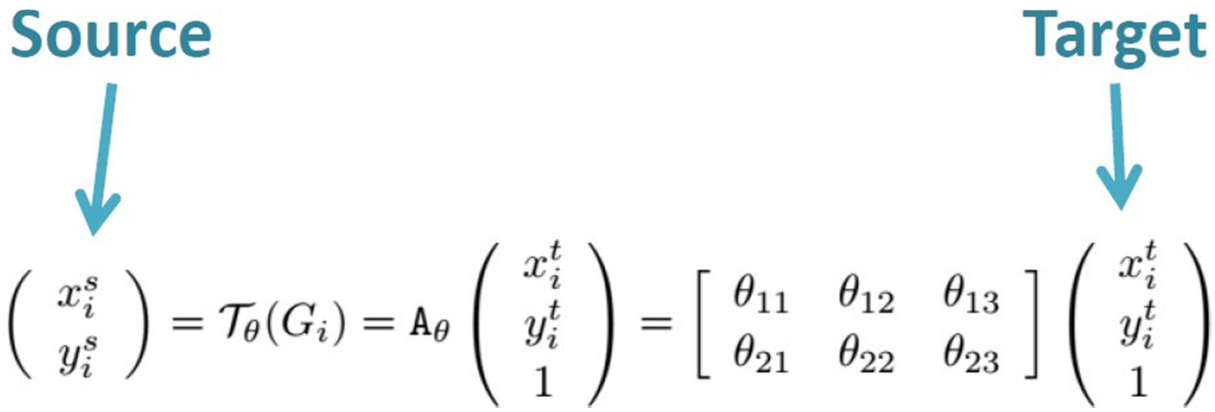
STN stands for Spatial Transformer Network for image classification. Google Deepmind briefly reviews it. STN helps to crop out and scale-normalizes appropriate region, which can simplify the subsequent classification task and lead to better classification performance as below:



(a) Input Image with Random Translation, Scale, Rotation, and Clutter, (b) STN Applied to Input Image, (c) Output of STN, (d) Classification Prediction

Spatial Transformer Network (STN)





- STN is composed of **Localisation Net**, **Grid Generator,** and **Sampler**.

Localization Net

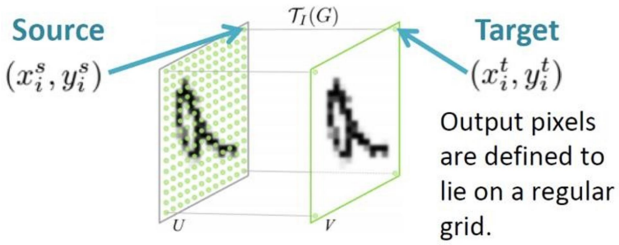
- With **input feature map *U***, with (width)*W*, (height) *H*, and *C* channels, **outputs are *θ***, parameters of transformation *Tθ*. It can be learned as affine transform as above. Or to be more constrained, such as the used for attention which only contains scaling and translation as below:

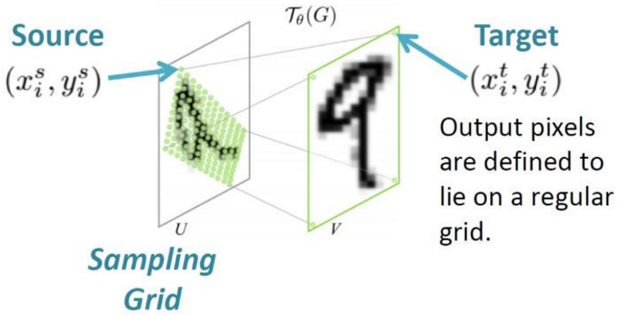
$$\mathbf{A}_\theta = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \end{bmatrix}$$

Grid Generator

- Suppose we have a regular grid G, this G is a set of points with **source coordinates (xs_i, ys_i)**, which act as **input**.
- Then we **apply transformation *Tθ* on G**, i.e., *Tθ*(*G*).
- After *Tθ*(*G*), a set of points with **destination coordinates (xt_i, yt_i) is outputted**. These points have been altered based on the transformation parameters. It can be Translation, Scale, Rotation or More Generic Warping depending on how we set *θ* as mentioned above.

Sampler





- **Based on the new set of coordinates (xt_i, yt_i), we generate a transformed output feature map *V***. This *V* is translated, scaled, rotated, warped, projective transformed or affined, whatever.
- It is noted that STN can be applied to not only input image but also intermediate feature maps.

Q2.What is decaNLP?

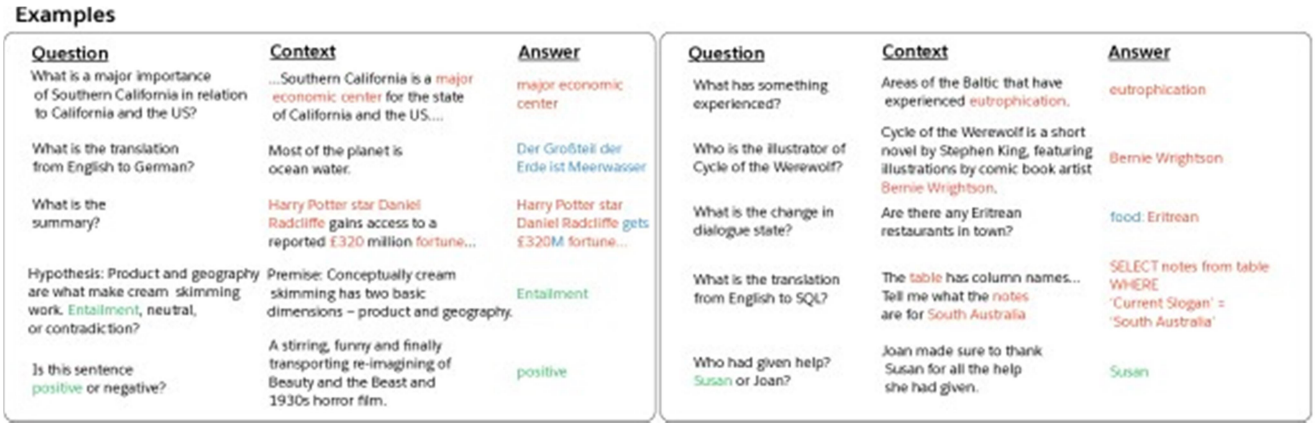
Answer:

We introduced the Natural Language Decathlon (decaNLP) to explore models that generalize to many different kinds of Natural Language Processing(NLP) tasks. decaNLP encourages single model to

simultaneously optimize for 10 tasks: question answering, machine translation, document summarization, semantic parsing, sentiment analysis, natural language inference(NLI), semantic role labeling, relation extraction, goal-oriented dialogue, and pronoun resolution.

We frame all the tasks as question answering [Kumar et al., 2016] by allowing task specification to take the form of a natural language question q: all inputs have a context, question, and answer (Fig. 1). Traditionally, NLP examples have inputs x and output y, and the underlying task t is provided through explicit modeling constraints. Meta-learning approaches include t as additional input. Our approach does not use the single representation for any t but instead uses natural language questions that describe underlying tasks. This allows single models to multitask effectively and makes them more suitable as pre-trained models for transfer learning and meta-learning: natural language questions allow a model to generalize to entirely new tasks through different but related task descriptions.

The MQAN (multitask question answering network) is designed for decaNLP and makes use of a novel dual attention and multi-pointer-generator decoder to multitask across all tasks in decaNLP. Our results represent that training the MQAN jointly on all tasks with the right anti-curriculum strategy can achieve performance comparable to that of ten separate MQANs, each trained separately. An MQAN pretrained on decaNLP shows improvements in transfer learning for machine translation and named entity recognition(NER), domain adaptation for sentiment analysis and natural language inference(NLI), and zero-shot capabilities for text classification. Though not explicitly designed for any one job, MQAN proves to be a robust model in a single-task setting as well, achieving state-of-the-art results on the semantic parsing component of decaNLP.



In the above figure: Overview of the decaNLP dataset with one example from each decaNLP task in the order presented in Section 2. They show how the datasets were pre-processed to become question

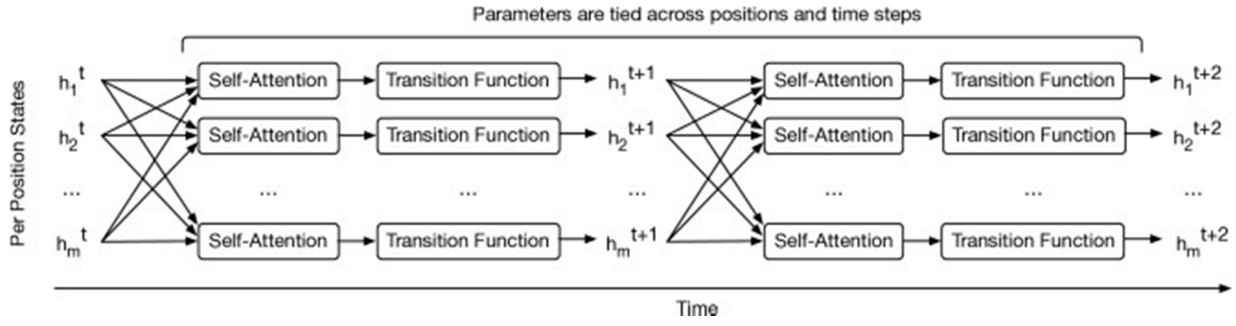
answering problems. Answer words in red are generated by pointing to the context, in green from the issue, and in blue if they are made from a classifier over the output vocabulary.

Q3.Universal Transformers

Answer:

Convolutional and fully-attentional feed-forward architectures such as the Transformer model have recently emerged as viable alternatives to RNNs(Recurrent neural networks) (for the range of sequence modeling tasks, notably machine translation (JonasFaceNet2017; transformer,). These architectures address the significant shortcoming of RNNs, namely their inherently sequential computation, which prevents parallelization across elements of input sequence while still addressing vanishing gradients problem (vanishing-exploding-gradient). Transformer model, in particular, achieves this by relying entirely on the self-attention mechanism (decomposableAttnModel; lin2017structured) to compute series of context-informed vector-space representations of symbols in its input and output, which are then used to predict distributions over subsequent symbols as the model predicts output sequence symbol-by-symbol. Not only in this mechanism straightforward to parallelize, but as each symbol's representation is also directly informed by all other symbols representations, this results in an active global receptive field. This stands, in contrast, to, e.g., convolutional architecture, which typically has limited receptive field.

Notably, however, Transformer foregoes the (Recurrent Neural Network)RNN's inductive bias towards learning recursive or iterative transformations. Our experiments indicate that this inductive bias may be important for several algorithmic and language understanding tasks of varying complexity: in contrast to models such as the Neural Turing Machine, the Neural GPU, or Stack RNNs, the Transformer does not generalize well to input lengths not encountered during training.



In this paper, we propose a *Universal Transformer*. It combines the parallelizability and global receptive field of a Transformer model with the recurrent inductive bias of RNNs, which seems to be better suited to range of algorithmic and natural language understanding(NLU) sequence-to-sequence problems. As the name implies, in contrast to standard Transformer, under certain assumptions, a Universal Transformer can be shown to be computationally universal.

In each step, the Universal Transformer iteratively refine its representations for all positions in sequence in parallel with self-attention mechanism decomposableAttnModel (); lin2017structured (), followed by the recurrent transformation consisting of a depth-wise separable convolution (xception2016) or a position-wise fully-connected layer (see above Fig). We also extended the Universal Transformer by employing an adaptive computation time mechanism at each position in sequence (graves2016adaptive), allowing model to choose the required number of refinement steps for each symbol dynamically.

When running for fixed number of steps, the Universal Transformer is equivalent to a multi-layer Transformer with a tied parameter across its layers. However, another, and possibly more informative, way of characterizing Universal Transformer is as recurrent function evolving per-symbol hidden states in parallel, based at each step on a sequence of the previous unknown state. In this way, it is similar to architectures such as Neural GPU and the Neural Turing Machine. The Universal Transformer thereby retains the attractive computational efficiency of original feed-forward Transformer model, but with an added recurrent inductive bias of RNNs. In its adaptive form, we show that the Universal Transformer can effectively interpolate between the feed-forward, fixed-depth Transformer, and a gated, recurrent architecture running for several steps depending on the input data.

Our experimental results show that its recurrence improve results in machine translation, where Universal Transformer outperforms the standard Transformer with a same no.of parameters. In experiments on several algorithmic tasks, Universal Transformer consistently improves significantly over LSTM(Long Short Term Memory) RNNs and the standard Transformer. Furthermore, on bAbI and LAMBADA text understanding data sets, the Universal Transformer achieves a new state of the art.

Q4. What is StarSpace in NLP?

Answer:

We introduce StarSpace, the neural embedding model that is general enough to solve a wide variety of problems:

- Other labeling tasks, or Text classification, e.g., sentiment classification.

- Ranking of the set of entities, e.g., a classification of web documents given a query.
- Collaborative filtering-based recommendation, e.g., recommending documents, videos or music.
- Content-based recommendation where content is defined with discrete features, e.g., words of documents.
- Embedding graphs, e.g., multi-relational graphs such as Freebase.
- Learning word, sentence, or document embeddings.

It can be viewed as a straight-forward and efficient strong baseline for any of these tasks. In experiment, it is shown to be on par with or outperforming several competing methods while being generally applicable to cases where many of that method are not.

The method works by learning entity embeddings with discrete feature representation from relations among collections of those entities directly for the task of ranking or classification of interest. In the general case, StarSpace embeds objects of different types into a vectorial embedding space; hence, the "star" ("*," meaning all types) and "space" in a name and in that familiar space compares them against each other. It learns to rank the set of entities, documents, or objects given a query entity, document, or object, where the query is not necessarily of the same type as the items in the set.

Q5. TransferTransfo in NLP

Answer:

Non-goal-oriented dialogue systems (chatbots) are interesting test-bed for interactive Natural Language Processing (NLP) systems and are also directly useful in wide range of applications ranging from technical support services to entertainment. However, building intelligent conversational agent remains an unsolved problem in artificial intelligence(AI) research. Recently, recurrent neural network(RNN) based models with sufficient capacity and access to large datasets attracted large interest when first attempted. It showed that they were capable of generating meaningful responses in some chit-chat settings. Still, further inquiries in the capabilities of these neural network

architectures and developments indicated that they were limited which made communicating with them a rather unsatisfying experience for human beings.

The main issues with these architectures can be summarized as:

- (i) the wildly inconsistent outputs and the lack of a consistent personality (Li and Jurafsky, 2016),
- (ii) the absence of long-term memory as these models have difficulties in taking into account more than the last dialogue utterance; and
- (iii) a tendency to produce consensual and generic responses that are vague and not engaging for humans (Li, Monroe, and Jurafsky, 2016).

In this work, we take a step toward more consistent and relevant data-driven conversational agents by proposing a model architecture, associated training and generation algorithms which are able to significantly improve over the traditional seq-2-seq and information-retrieval baselines in terms of (i) relevance of the answer (ii) coherence with a predefined personality and dialog history, and (iii) grammaticality and fluency as evaluated by auto.

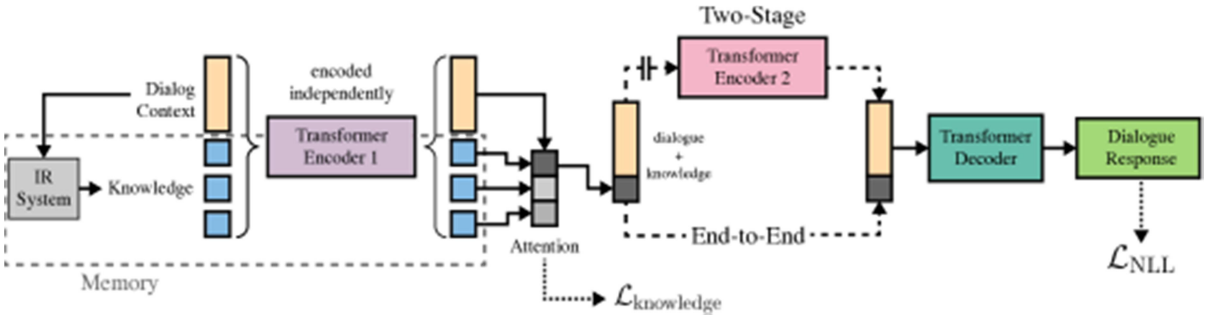
Q6. Wizard of Wikipedia: Knowledge-Powered Conversational Agents

Answer:

Arguably, one of the critical goals of AI and the ultimate goal of natural language research is for the human to be able to talk to the machine. In order to get close to this goal, machines must master the no. of skills: to be able to comprehend language, employ memory to retain and recall knowledge, to reason about these concept together, and finally output a response that both fulfills functional goals in conversation while simultaneously being captivating to their human speaking partner. The current state-of-the-art(SOTA) approaches, sequence to sequence(seq2SEQ) models of various kinds (Sutskever et al., 2014; Vinyals & Le, 2015; Serban et al., 2016; Vaswani et al., 2017) attempt to address some of these skills, but generally suffer from inability to bring memory and knowledge to bear; as indicated by their name, they involve encoding input sequence, providing limited reasoning by transforming their hidden state given input, and then decoding to the output. To converse intelligently on the given topic, the speaker needs knowledge of that subject, and it is our contention here that more direct knowledge memory mechanisms need to be employed. In this work, we consider setups where this can be naturally measured and built.

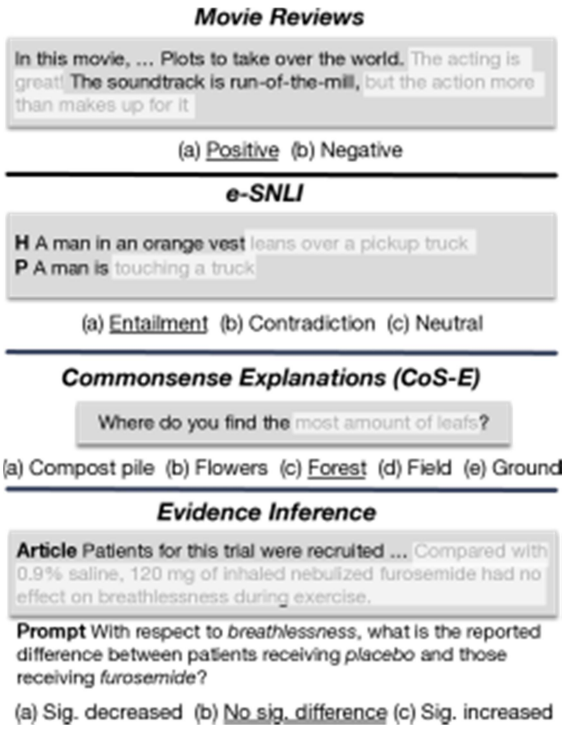
We consider the task of open-domain dialogue, where two speakers conduct open-ended chit-chat given an initial starting topic, and during the conversation, the topic can broaden or focus on related themes. During such conversations, an interlocutor can glean new information and personal points of view from

their speaking partner, while providing themselves similarly. This is a challenging task as it requires several components not found in many standard models. We design a set of architectures specifically for this goal that combine elements of Memory Network architectures (Sukhbaatar et al., 2015) to retrieve knowledge and read and condition on it, and Transformer architectures (Vaswani et al., 2017) to provide state-of-the-art text representations and sequence models for generating outputs, which we term Transformer Memory Networks.



Q7. ERASER: A Benchmark to Evaluate Rationalized NLP Models

Answer:



Interest has recently grown in interpretable (Natural Language Processing) NLP systems that can reveal **how** and **why** model make their predictions. But work in this direction has been conducted on the

different dataset with correspondingly different metrics, and inherent subjectivity in defining what constitute 'interpretability' has translated into researcher using different metrics to quantify performance. We aimed to facilitate measurable progress on designing interpretable NLP(Natural Language Processing) models by releasing the standardized benchmark of datasets — augmented and repurposed from pre-existing corpora, and spanning the range of NLP tasks — and associated metrics for measuring the quality of rationales. We refer to this as ERASER(Evaluating Rationales And Simple English Reasoning) benchmark.

In curating and releasing ERASER we take inspiration from stickiness of GLUE (Wang et al., 2019B) and SuperGLUE Wang et al. (2019A) benchmarks for evaluating progress in natural language understanding(NLU) tasks. These have enabled rapid growth in models for inclusive language representation learning. We believe still somewhat nascent subfield of interpretable NLP(Natural Language Processing) stands to similarly benefit from the analogous collection of standardized datasets or tasks and metric.

'Interpretability' is the broad topic with many possible realizations Doshi-Velez and Kim (2017); Lipton (2016). In ERASER, we focuses specifically on *rationales*, i.e., snippets of text from the source document that support a specific categorization. All datasets contained in ERASER include such rational, explicitly marked by annotators as supporting specific classifications. By definition, rationales should be *sufficient* to categorize document, but they may not be comprehensive. Therefore, for some dataset, we have collected *complete* rationales, i.e., in which *all* evidence supporting the classification has been marked.

How one measures 'quality' of extracted rationales will invariably depend on their intended use. With this in mind, we propose the suite of metrics to evaluate rationales that might be appropriate for different scenarios. Widely, this includes measures of agreement with human-provided rationales and assessment of *faithfulness*. The latter aim to capture extent to which rationales provided by the model, in fact, informed its prediction.

While we propose metrics that we think are reasonable, we view a problem of designing metrics for evaluating rationales-especially for capturing faithfulness — as a topic for further research that we hope that ERASER will help facilitate. We plan to revisit metrics proposed here in future iterations of benchmark, ideally with input from community. Notably, while we provide a 'leaderboard,' this is perhaps better viewed as the 'results board'; we do not privilege any particular metric. Instead, we hope that ERASER permits comparison between models that provide rationales wrt different criteria of interest.

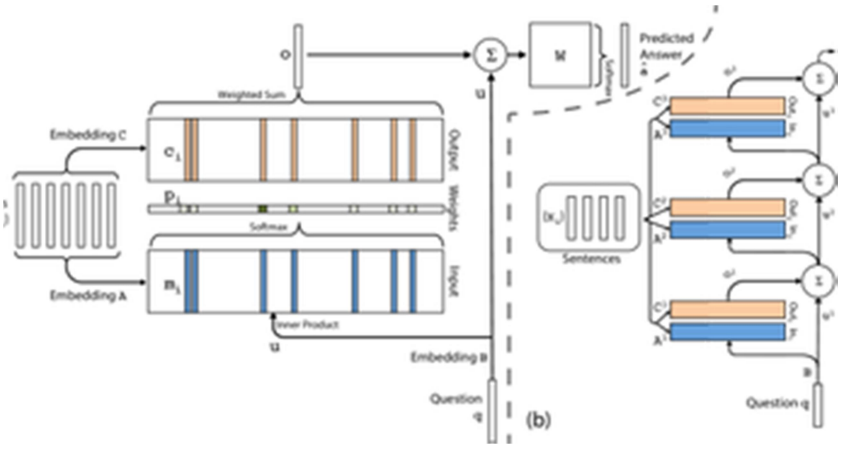
Q8. End to End memory networks

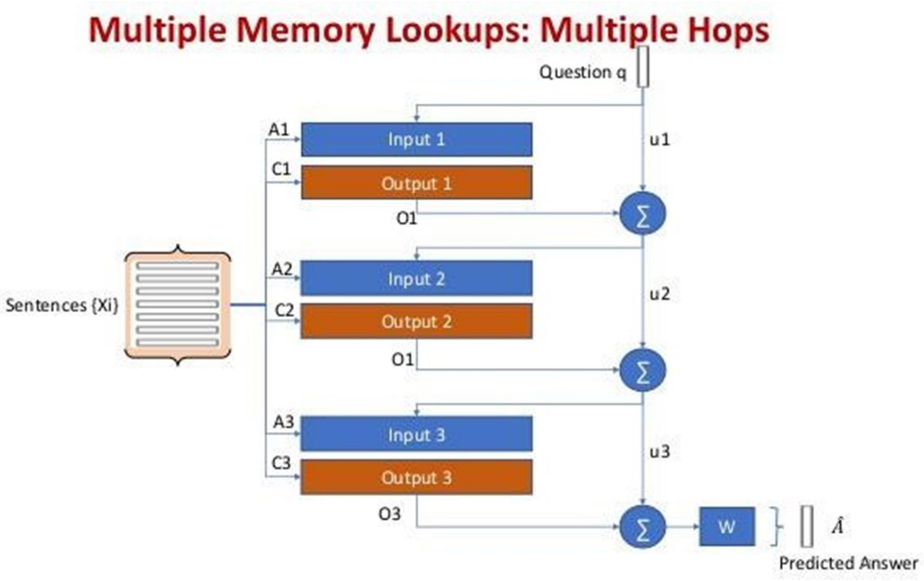
Answer:

Two grand challenges in artificial intelligence(AI) research have been to build a model that can make multiple computational step in the service of answering the question or completing the task, and models that can describe long term dependencies in sequential data.

Recently there has been the resurgence in models of computation using explicit storage and a notion of attention; manipulating such storage offers an approach to both of these challenges. In, the storage is endowed with continuous representation; reads from and writes to storage, as well as other processing steps, are modeled by actions of neural networks.

In this work, we present the new recurrent neural network (RNN) architecture where recurrence reads from possibly large external memory multiple times before outputting symbol. Our model can be considered the continuous form of the Memory Network implemented in. The model in that work was not easy to train via back-propagation and required supervision at each layer of a network. The continuity of model we present here means that it can be trained end-to-end from input-output pairs, and so applies to more tasks, i.e., tasks where such supervision is not available, like in language modeling or realistically supervised question answering tasks. Our model can also be seen as version of RNNsearch with multiple computational steps per output symbol. We will show experimentally that various hops over the long-term memory are crucial to excellent performance of our model on these tasks, and that training the memory representation can be integrated in a scalable manner into our end-to-end neural network model.





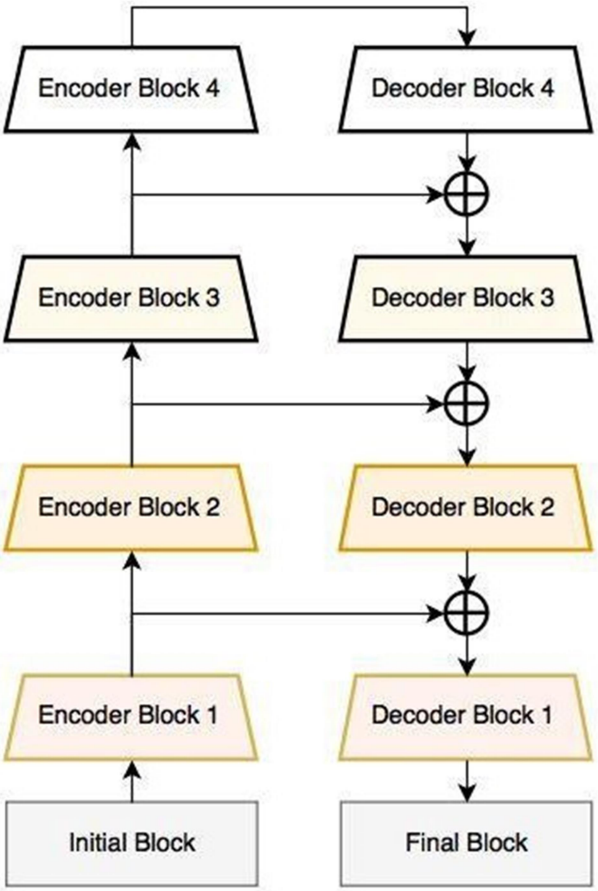
Q9. What is LinkNet?

Answer:

From my experience, LinkNet is lightning fast, which is one of the main improvements the authors site in their summary. LinkNet is a relatively lightweight network with around 11.5 million parameters; networks like VGG have more than 10x that amount.

The structure of LinkNet is to use a series of encoder and decoder blocks to break down the image and build it back up before passing it through a few final convolutional layers. The structure of the network was designed to minimize the number of parameters so that segmentation could be done in real-time.

I performed some tests of the LinkNet architecture but did not spend too much time iterating to improving the models.


Encoder Block 4
Decoder Block 4
Encoder Block 3
Decoder Block 3
Encoder Block 2
Decoder Block 2
Encoder Block 1
Decoder Block 1
Initial Block
Final Block