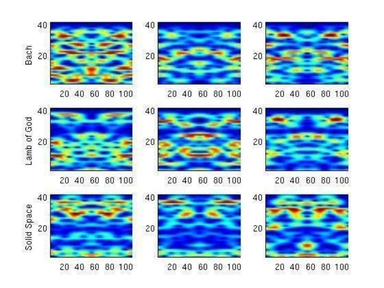
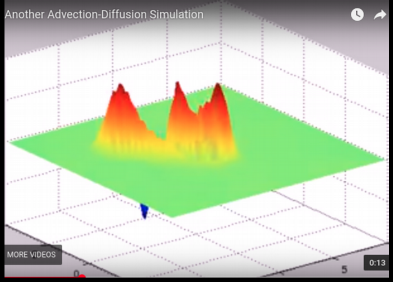
THOMAS WOOD



Who am I?



- BS Physics, 2007
- MS Applied Mathematics, 2013
- Former Research Scientist at Univ. of Wash.
- Former employee of Microsoft and Intel
- Co-founder of SynPon
- Lifelong Learner

Certifications

- **Underactuated Robotics**, *MITx*, Dec 2015
- **Initiating and Planning Projects**, *Coursera*, Jul 2015
- **Plasma Physics**, *EPFLx*, Jun 2015
- **Parallel Programming**, *Udacity*, Sep 2014
- **Data Wrangling with MongoDB**, *Udacity*, Jul 2014
- **Algorithms**, *Udacity*, Mar 2014
- **Machine Learning**, *Coursera*, Jan 2014

SYNPON

SYNPON

What is SynPon?

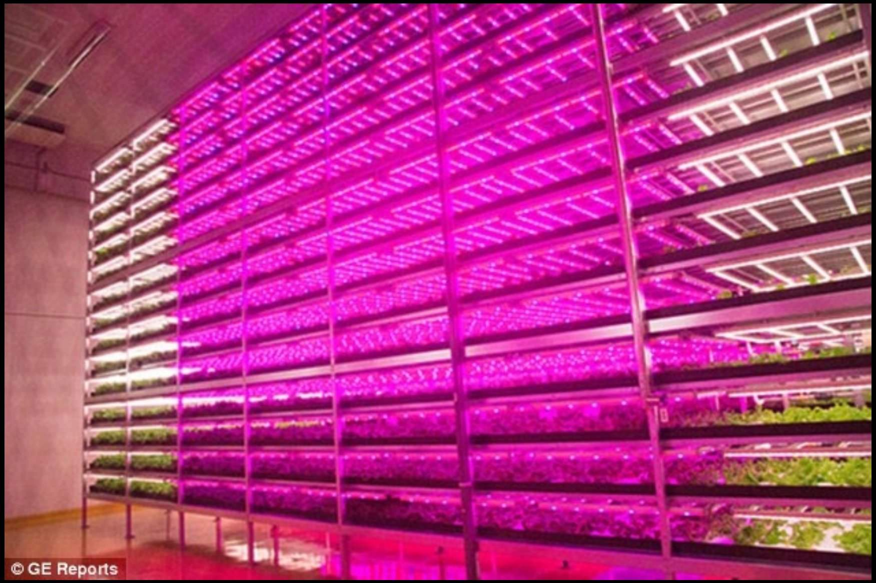
A small robotics laboratory focused on automation for vertical indoor gardens.

- Vertical Planting Towers
- Automated Plant Tending

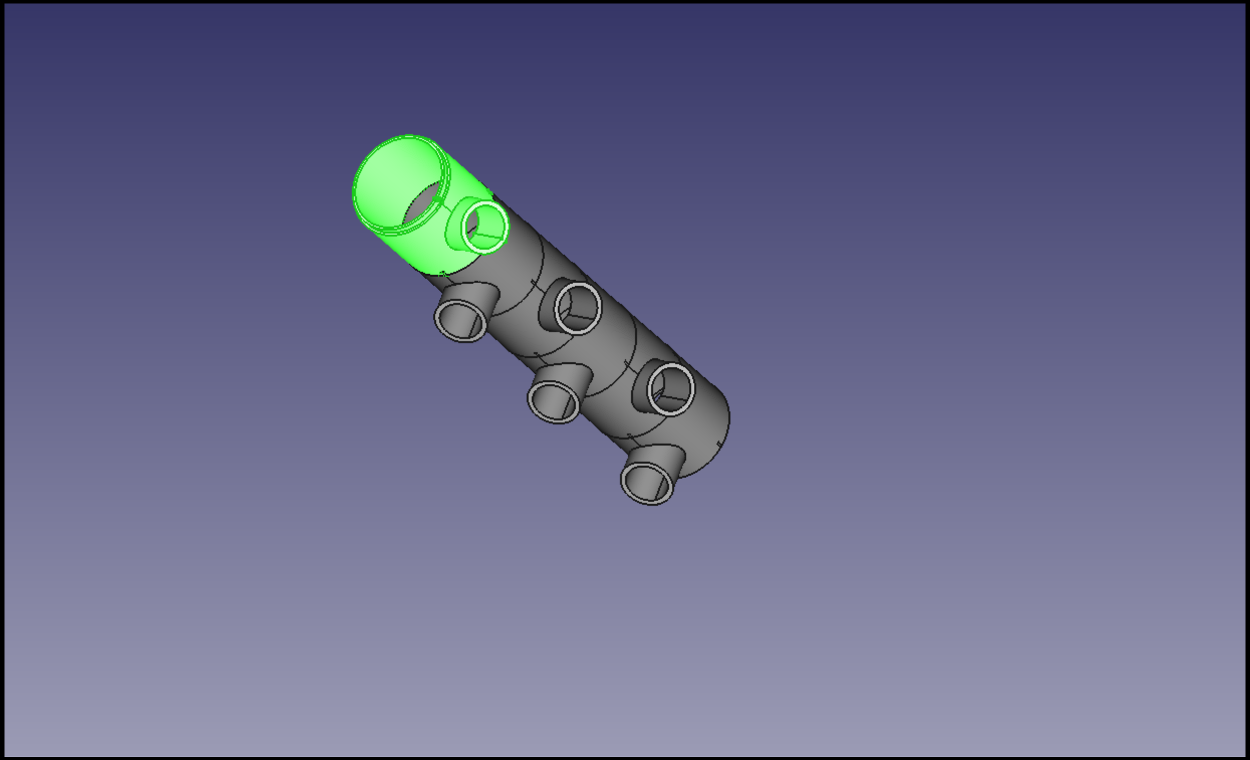
SYNPON

Why Vertical Farming?

- Indoor hydroponic horticulture uses **90%** less water than traditional farming and typically produces higher yields without the need for harmful pesticides.
- Vertical planting towers allow an indoor hydroponic operation to scale by the **volume** of their growing area as opposed to the **area** of the building, enabling even greater possible crop yields than traditional hydroponics.



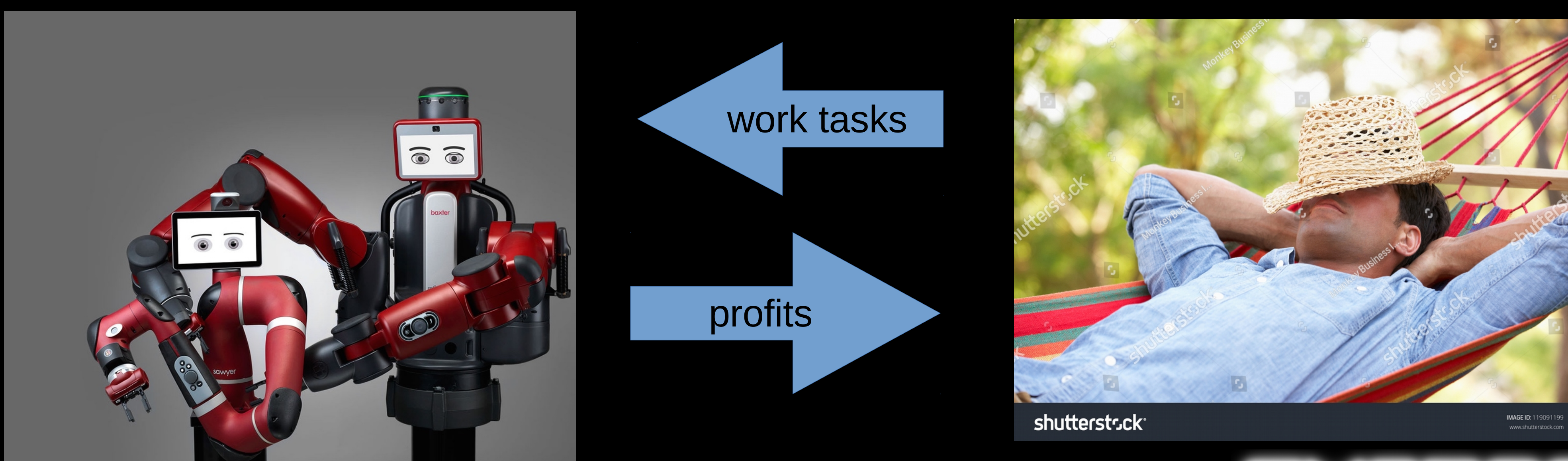
Vertical Planting Tower Design



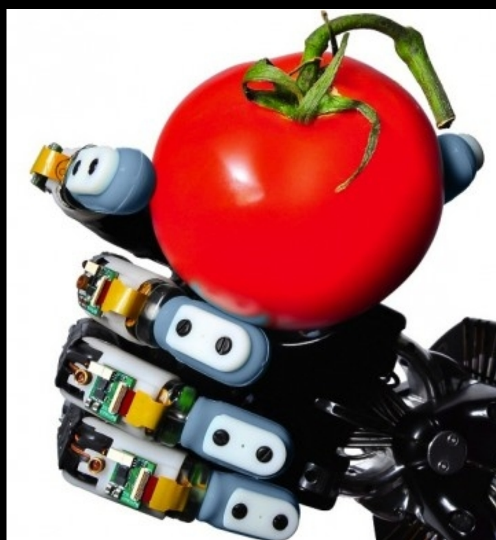
Vertical Planting Tower Manufacture

Why Automated Plant Tending?

- Labor costs account for around 40% of the price of certain fruits and vegetables. With automation we can drive down food prices.
- Sufficient levels of automation can turn a robotic indoor farm into a latch key business requiring little oversight.



Automated Plant Tending



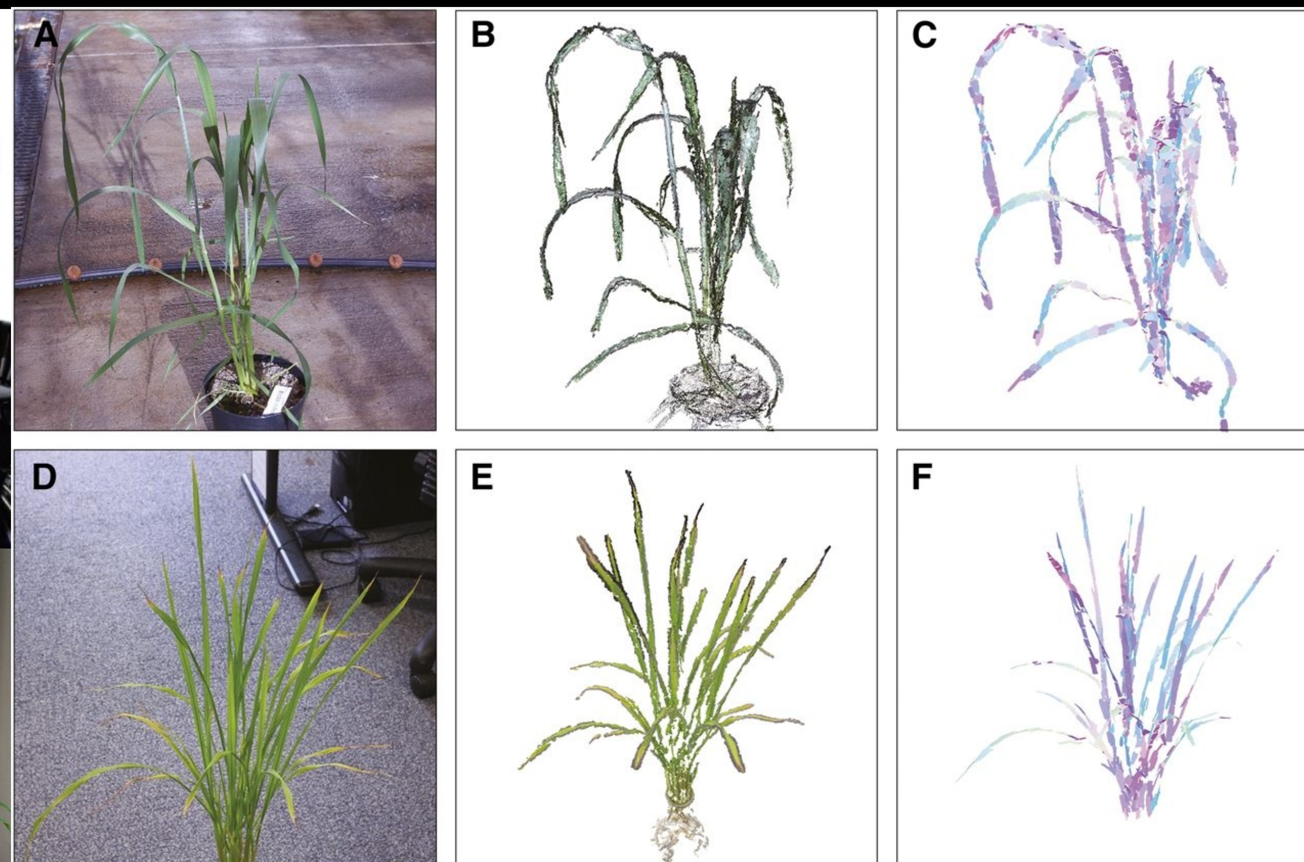


Image of digital plant reconstruction courtesy of:
Pound, Michael P., et al. “Automated Recovery of Three-Dimensional Models of Plant Shoots from Multiple Color Images.” Plant Physiology, American Society of Plant Biologists, 1 Dec. 2014, doi.org/10.1104/pp.114.248971.

Iron Ox

Well-funded California startup Iron Ox is also working on automated plant tending for indoor horticulture with a focus on lettuce.

Research

- Deep Reinforcement Learning for Robotics
- Multitask and Transfer Learning
- Open Source Robotic Manipulator Design
- Natural Language Understanding Systems†

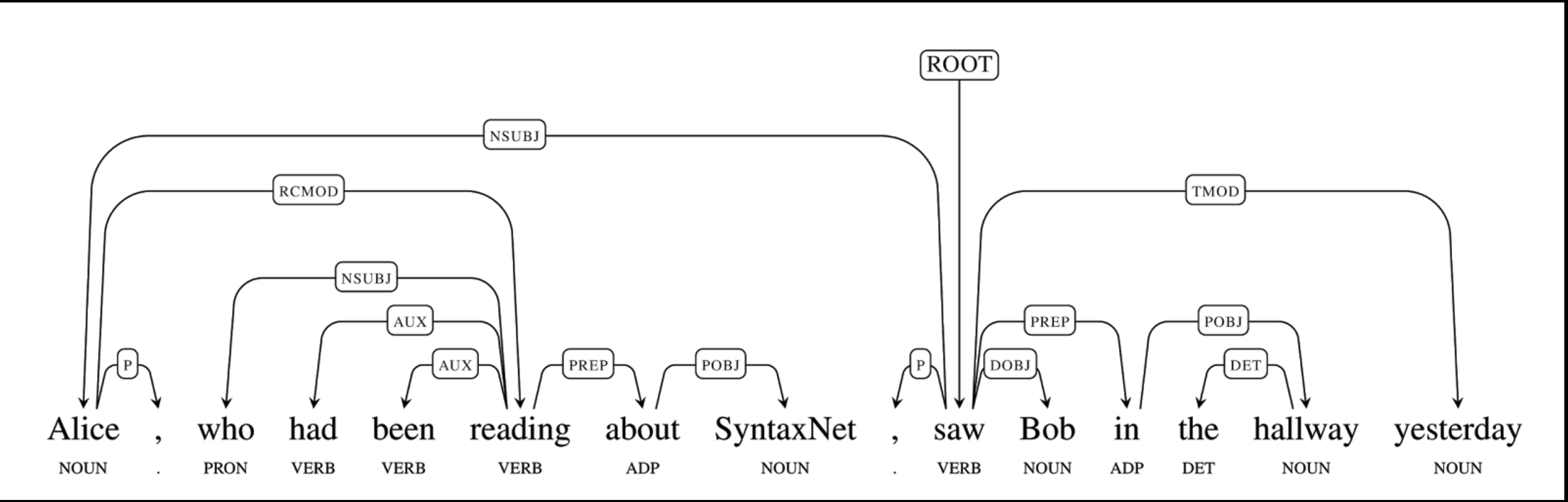
† We are going to talk about NLU from here on out.

NATURAL LANGUAGE UNDERSTANDING

What is Natural Language Understanding?

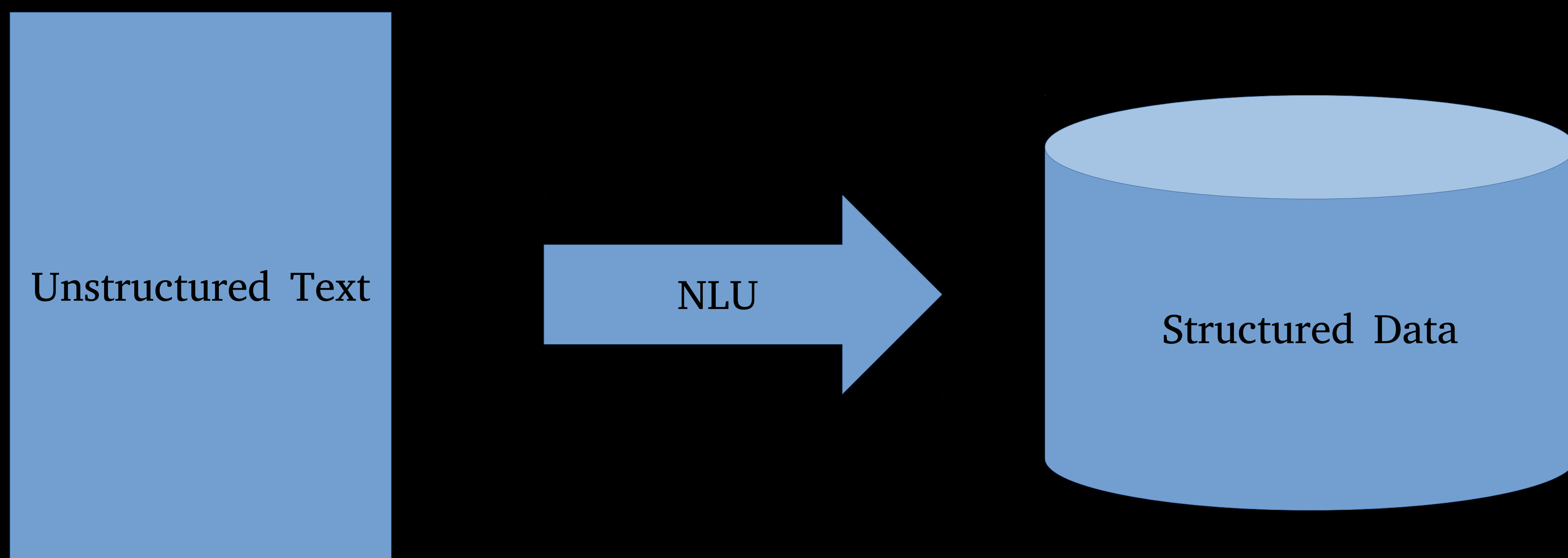
- Natural Language Understanding (**NLU**), or Machine Comprehension (**MC**), deals with the problem of disassembling and parsing input using both **semantic** and **syntactic** information contained in unstructured text.

SyntaxNet



Google released SyntaxNet, a powerful deep learning toolkit for NLU built on top of TensorFlow, in May 2016 along with the pre-trained English language syntax parser Parsey McParseface. The DRAGNN toolkit for Python was released in early 2017 and increases SyntaxNet's usability.

So What's the Big Deal?



It is estimated over 95% of information online is in Unstructured Text.

Structured text can be used to create Knowledge Graphs directly from syntax without a semantic database for reference.

Natural Language Processing in Enterprise

- Enterprise datasets are several orders of magnitude smaller than Commercial Datasets.
- Hugely successful Commercial NLP projects driven by Google, Yahoo, etc. depend upon large datasets which do not contain critical domain-specific **semantic** information needed for Enterprise.
- Natural Language Understanding tools are trained to process natural language based on **syntax** which can discern the domain-specific knowledge needed for Enterprise solutions.

So How Do We Get Structured Data from Unstructured Text?

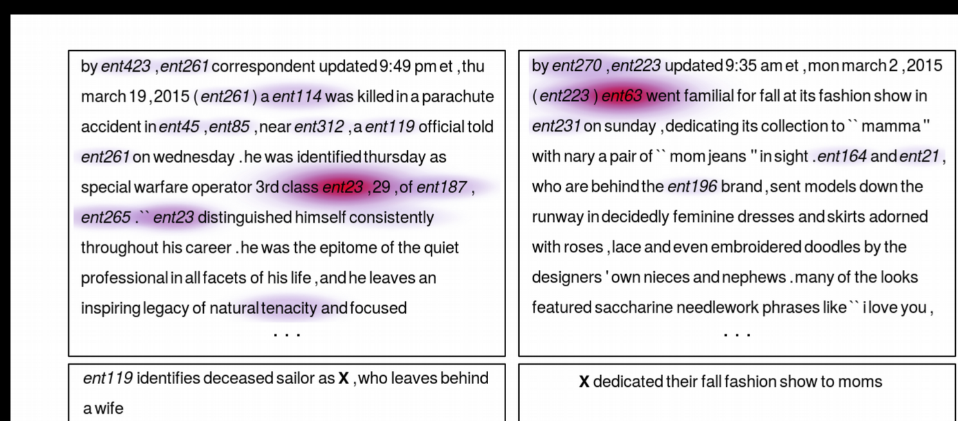
1. Create your knowledge graph directly from the tree output of a Syntactical Parser such as Parsey McParseface. Automatically discover relationships between entities and clean it up later. Generates a tree that can generally be viewed as a prioritization of the importance of the words in a sentence.

2. Have a pre-compiled list of relationships one wishes to explore from the unstructured text and use a Question-Answering System to answer the questions in which you're interested. These systems commonly feature attention models to prioritize word importance.

SYNAPON

Answering Questions From Text

- "Teaching Machines to Read and Comprehend" by Hermann et al. was published in 2015 and focused on using an attention mechanism in concert with a BiLSTM to learn how to answer questions from a dataset derived from CNN and Daily Mail articles.

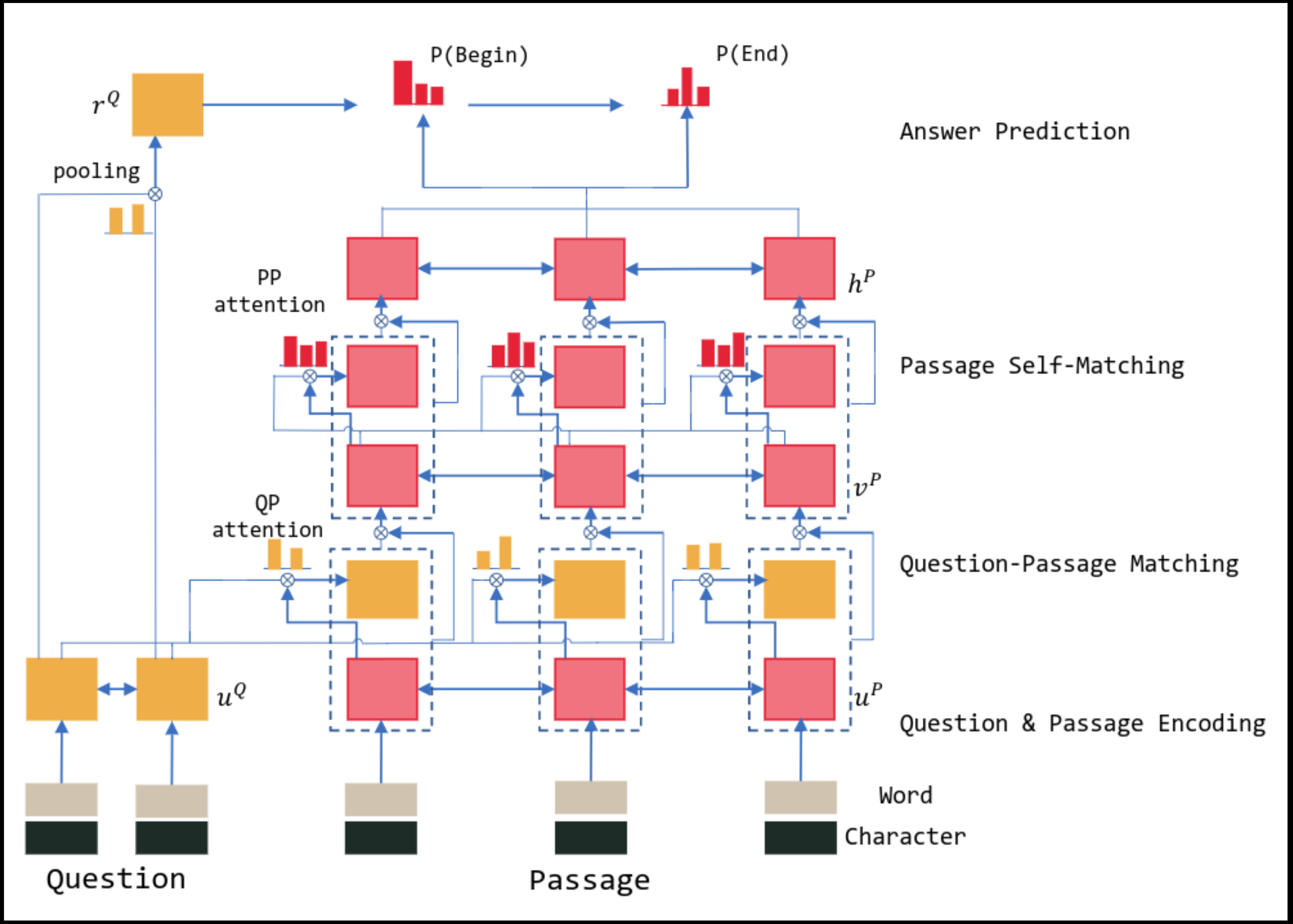


Stanford Question Answering Dataset (SQuAD)

- In 2016 a new dataset for Question Answering Systems generated from Wikipedia and an accompanying leader board was launched.
- The SquAD training set contains over 100,000 question-answer pairs and does not exclusively use anonymized entities as answers, unlike the CNN and Dailymail datasets.
- R-NET* from Microsoft Asia quickly established itself as the dominant method on the leaderboard.

*https://www.microsoft.com/en-us/research/publication/mrc/

R-NET: Machine Reading Comprehension With Self-Matching Networks



*https://www.microsoft.com/en-us/research/wp-content/uploads/2017/05/r-net.pdf

R-NET Overview

Takes the question and passage as inputs and outputs an interval of the passage which contains the answer.

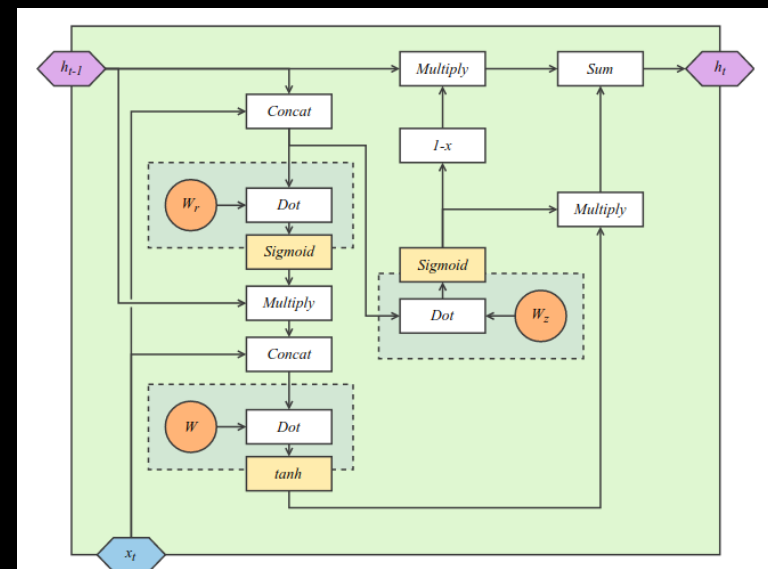
Steps of an R-NET:

1. Encode the question and the passage
2. Obtain question aware representation for the passage
3. Apply self-matching attention on the passage to get its final representation
4. Predict the interval which contains the answer.

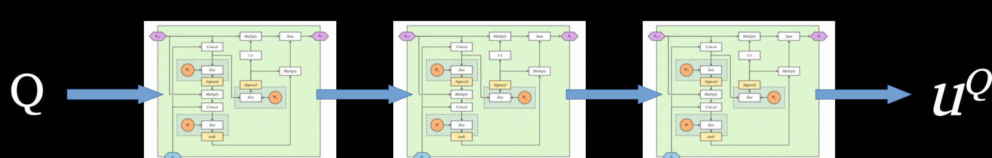
Encode the Question and Passage

1. Preprocess the question and passage with *gensim* to obtain GloVe embeddings for the words they contain.
2. Encode the question Q and passage P with three bidirectional GRU layers to obtain their respective representations u^Q and u^P.

GRU Cell



$$
\begin{aligned}
z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\
r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\
\tilde{h}_t &= tanh(W \cdot [r_t \circ h_{t-1}, x_t]) \\
h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t
\end{aligned}
$$



SYNPON

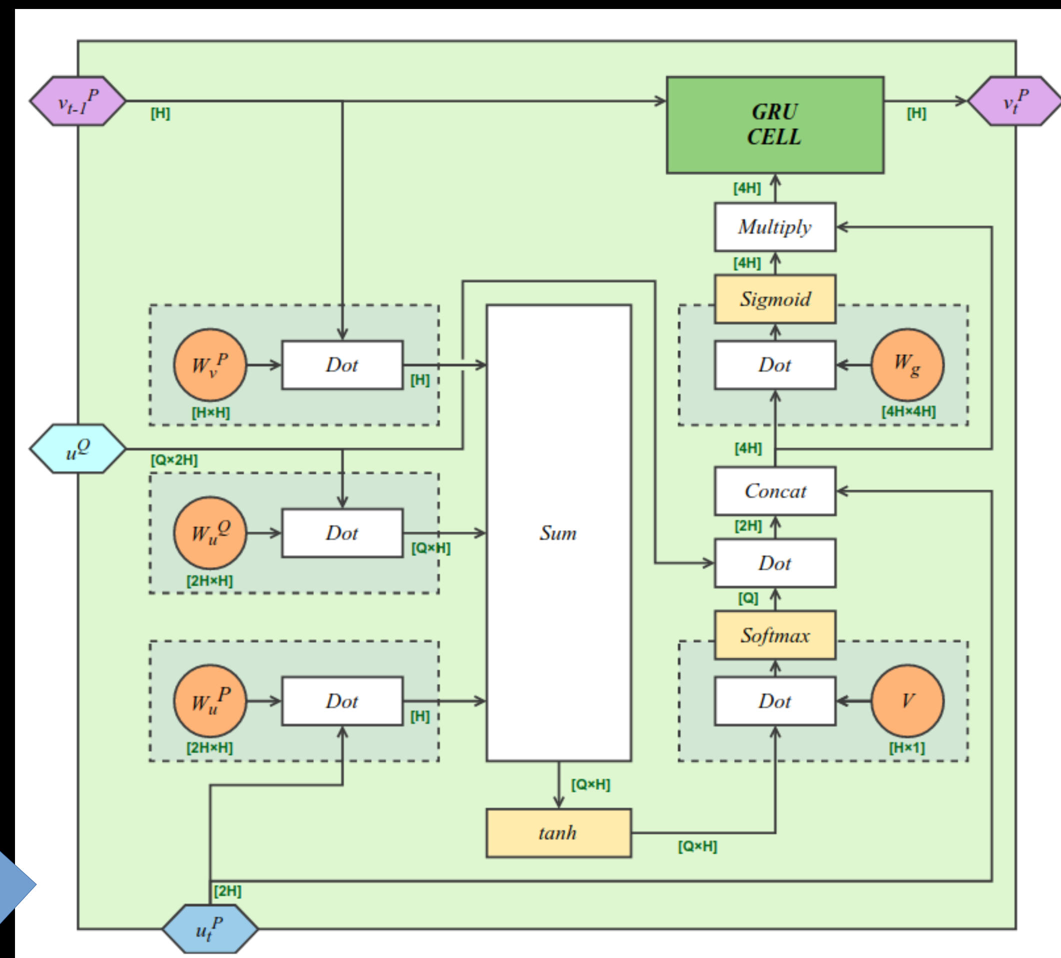
Obtain Question Aware Representation for the Passage

Combine three inputs:
1. Previous state of the GRU, v_{t-1}^P
2. Matrix representation of question u^Q
3. Vector representation of passage at the t-th word u_t^P

Perform operations, including broadcasting vectors into matrices, with the new weight matrices: W_v^P, W_u^Q, W_u^P, V, and W_g before sending the result through as input to a "vanilla" GRU.

The step is repeated T times, T being the length of the passage, to create an output matrix V^P out of the hidden states of the Question Attention GRU.

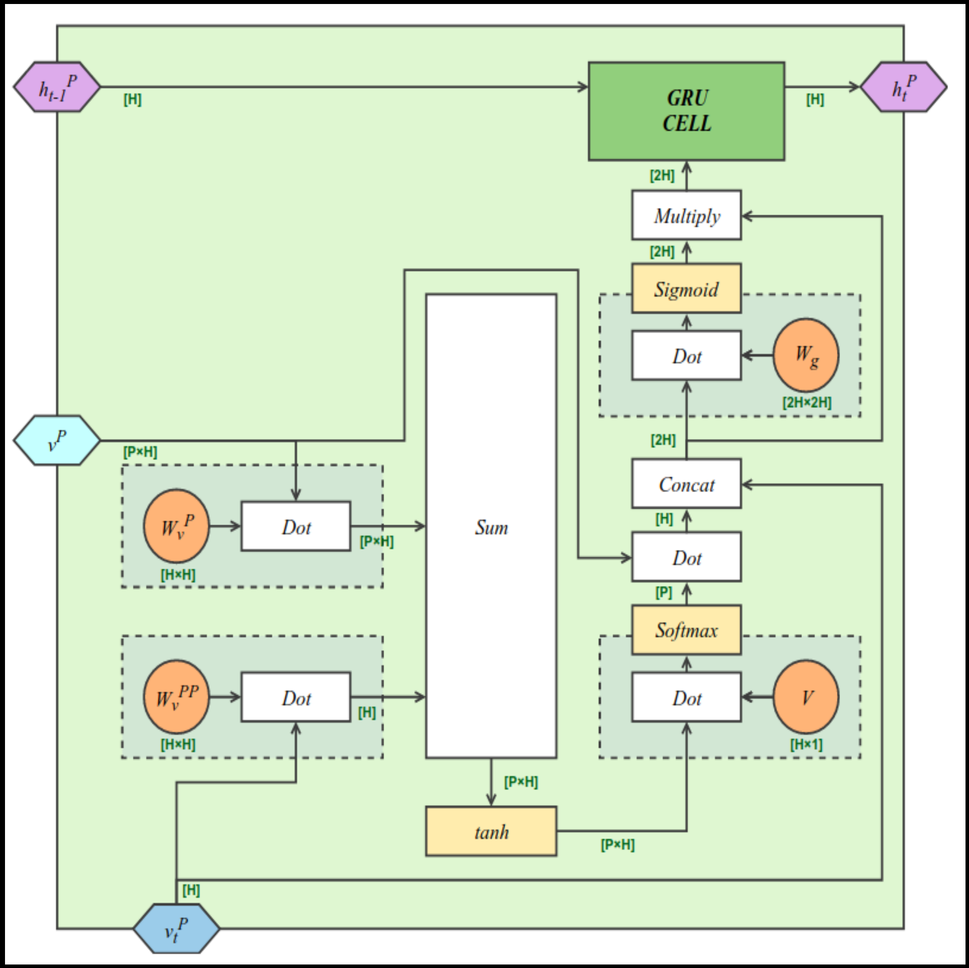
Question Attention GRU



Apply Self-Matching Attention on the Passage

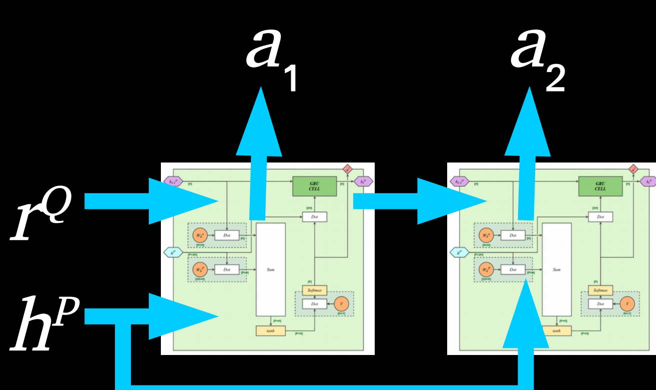
The output v^P of the previous step is used as input to the Self-Matching Attention Module, which is similar to the Question Attention GRU in that we do a bunch of operations using a new set of weight matrices, this time:
W_v^P, W_v^{PP}, V, and W_g,
before sending the result to a standard GRU to obtain the output h_t^P which is the hidden state of the module at timestep t. Each h_t^P compares the vector representation of each word in the passage, v_t^P, with the matrix representation of the passage as a whole, v^P.

Self-Matching Attention Module

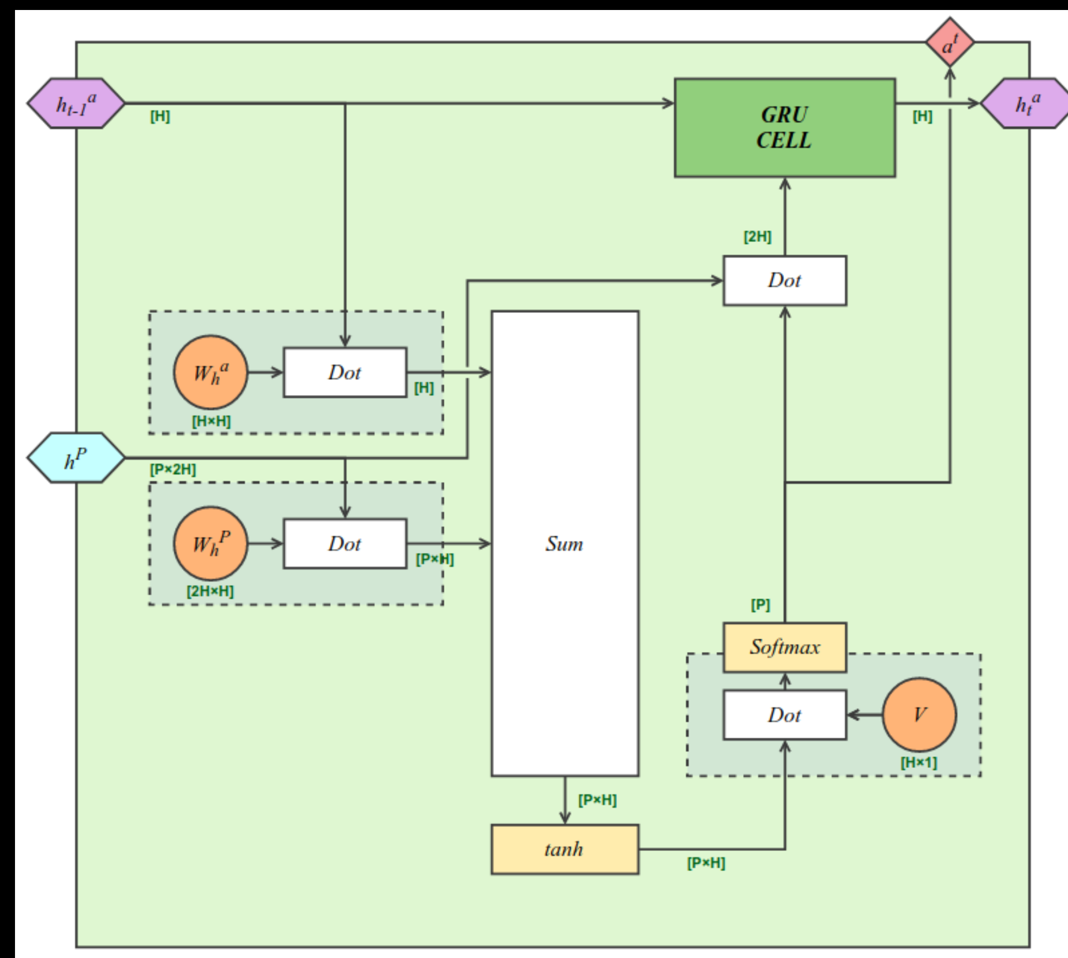


Predict the Interval Which Contains the Answer

Another attention mechanism called the Question Pooling Layer is used on the vector representation of the whole question u^Q to create a representation r^Q of the question which is used as the initial hidden state of the Pointer GRU.
After a single step, the output of the Pointer GRU, which is not its hidden state in the final layer, is a vector which represents the starting index for the answer. One last step of the Pointer GRU gives the stopping index vector.

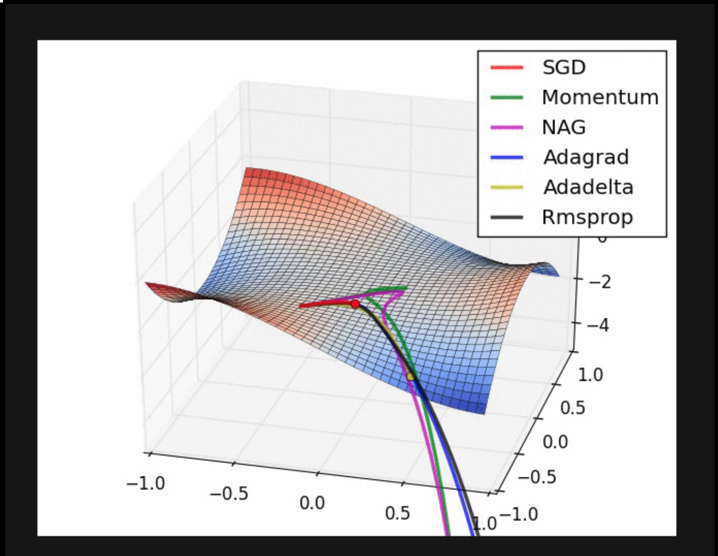


Pointer GRU



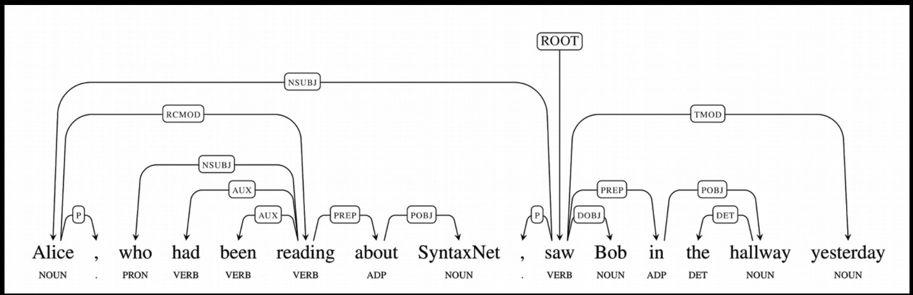
Training

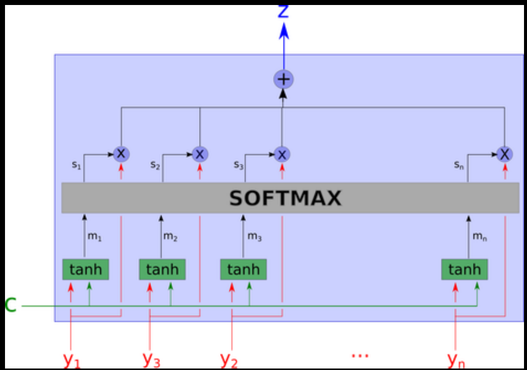
- The loss is defined by the categorical cross entropy function in Keras, which evaluates $H(p,q) = -\Sigma_x \, p(x) \cdot \log(q(x))$, where p is the target distribution (a one-hot encoding in our case), and q is the distribution predicted by the model, a_1 and a_2.
- AdaDelta optimization is used to train the network.



Next Directions

- Ensembles of R-NETs
- Adding in tree-graph syntactic information on the question and passage from independently trained parsers like Parsey McParseface in addition to attention models on sequences
- Training the syntactic parsers end-to-end with the R-NET so that even the initial embeddings which are currently provided by GloVe can be optimized for question answering.





Questions?

THOMAS WOOD