A Gentle introduction to Deep learning and NLP

Tal Perry
@thetalperry
tal@lighttag.io

LightTAG

Who am I

- Tal Perry
- Founder of LightTag
- Google Development Expert (ML)
- Data Scientist at Citi
- Built that tree

What We'll Cover

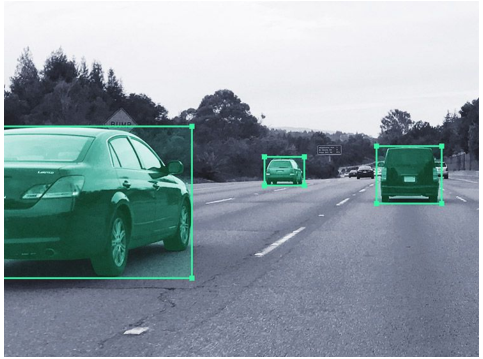
Deep Learning and the need for data

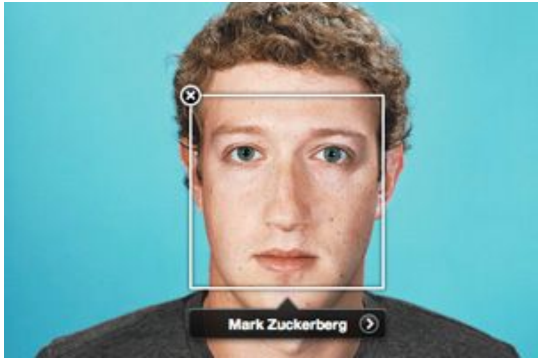
Deep Learning for NLP

LightTAG

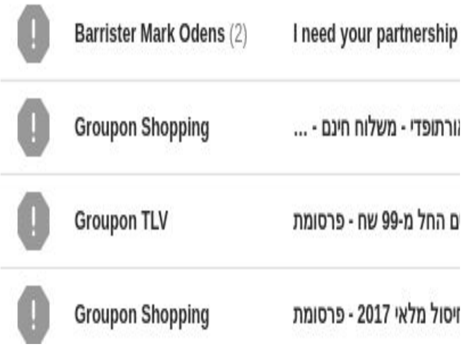
What we do (demo)

Why would anyone need that ?



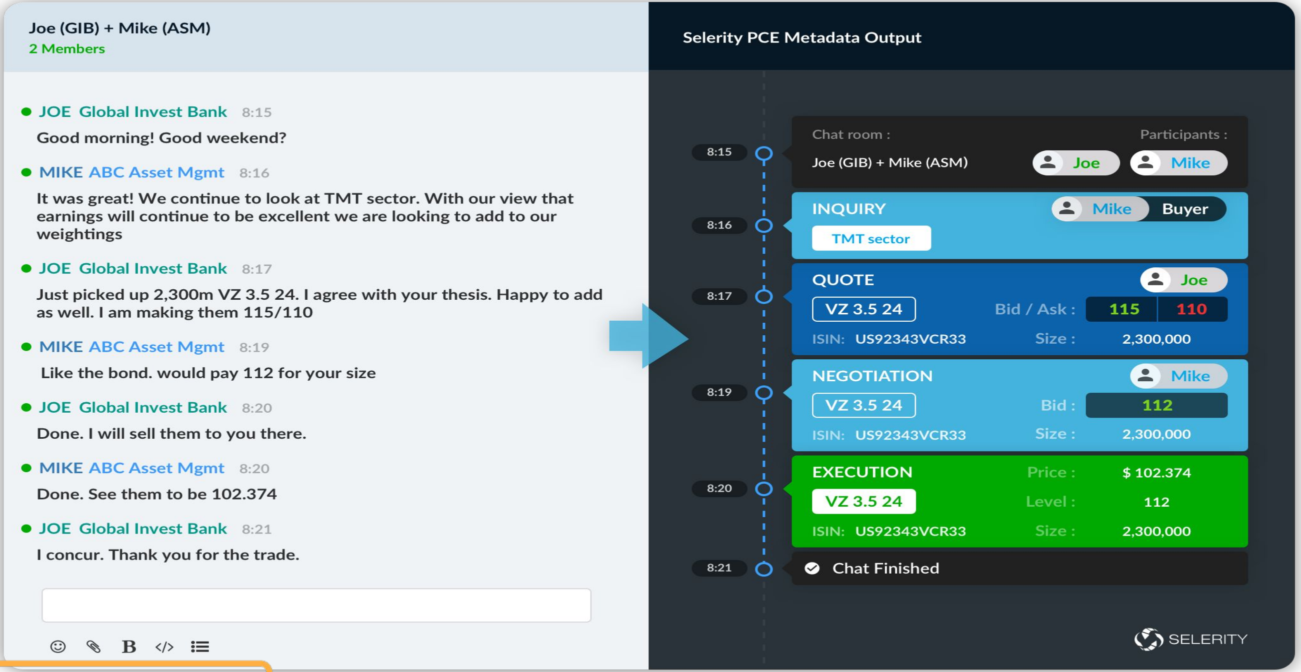


Mark Zuckerberg



Barrister Mark Odens (2)
I need your partnership
Groupon Shopping
Groupon TLV
Groupon Shopping

A “personal” example



SELERITY

LightTAG

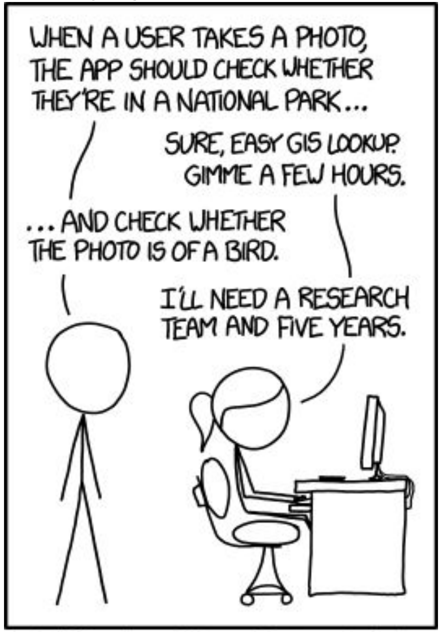
What does “deep learning” do

And why is it being adopted ?

LightTAG

What's changed?

~2013



IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

~2016

two birds sitting on top of a tree branch.

LightTAG

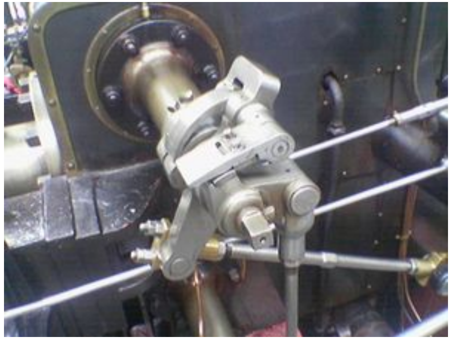
The Deep Learning revolution

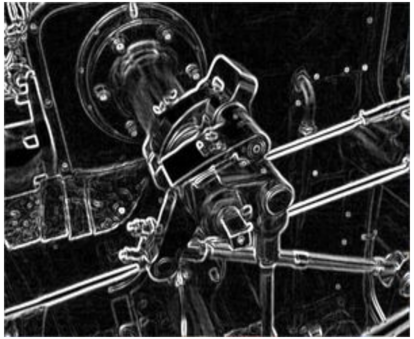
1. Deep learning eliminates types of work data professionals had to do
2. Open source software lets anyone do deep learning

LightTAG

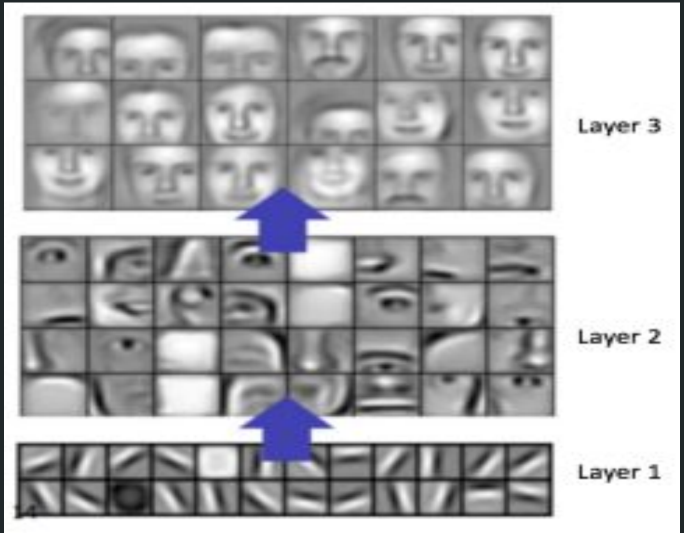
Feature Engineering

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A} \qquad \mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$



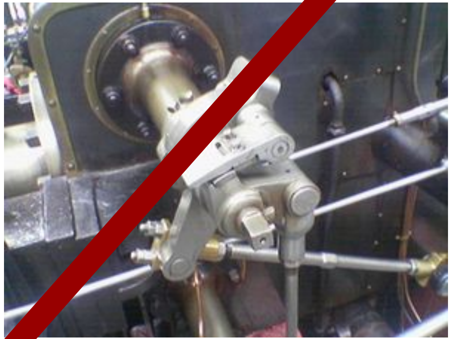


Deep Learning

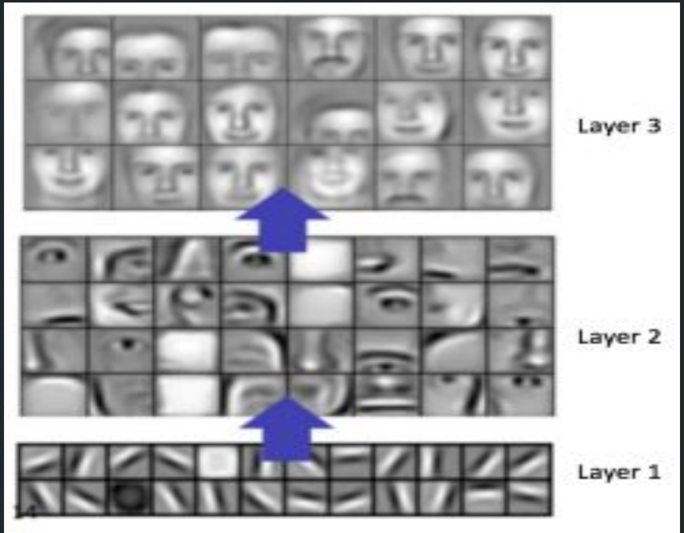


~~Feature Engineering~~

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A} \qquad \mathbf{G} = \sqrt{\mathbf{G}_x^{\,2} + \mathbf{G}_y^{\,2}}$$



Deep Learning



Deep Learning goes commodity

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$
$$h_t = o_t \circ \sigma_h(c_t)$$

```python
def forward(X, WLSTM, c0 = None, h0 = None):
  """
  X should be of shape (n,b,input_size), where n = length of sequence, b = batch size
  """
  n,b,input_size = X.shape
  d = WLSTM.shape[1]/4 # hidden size
  if c0 is None: c0 = np.zeros((b,d))
  if h0 is None: h0 = np.zeros((b,d))

  # Perform the LSTM forward pass with X as the input
  xphpb = WLSTM.shape[0] # x plus h plus bias, lol
  Hin = np.zeros((n, b, xphpb)) # input [1, xt, ht-1] to each tick of the LSTM
  Hout = np.zeros((n, b, d)) # hidden representation of the LSTM (gated cell content)
  IFOG = np.zeros((n, b, d * 4)) # input, forget, output, gate (IFOG)
  IFOGf = np.zeros((n, b, d * 4)) # after nonlinearity
  C = np.zeros((n, b, d)) # cell content
  Ct = np.zeros((n, b, d)) # tanh of cell content
  for t in xrange(n):
    # concat [x,h] as input to the LSTM
    prevh = Hout[t-1] if t > 0 else h0
    Hin[t,:,0] = 1 # bias
    Hin[t,:,1:input_size+1] = X[t]
    Hin[t,:,input_size+1:] = prevh
    # compute all gate activations. dots: (most work is this line)
    IFOG[t] = Hin[t].dot(WLSTM)
    # non-linearities
    IFOGf[t,:,:3*d] = 1.0/(1.0+np.exp(-IFOG[t,:,:3*d])) # sigmoids; these are the gates
    IFOGf[t,:,3*d:] = np.tanh(IFOG[t,:,3*d:]) # tanh
    # compute the cell activation
    prevc = C[t-1] if t > 0 else c0
    C[t] = IFOGf[t,:,:d] * IFOGf[t,:,3*d:] + IFOGf[t,:,d:2*d] * prevc
    Ct[t] = np.tanh(C[t])
    Hout[t] = IFOGf[t,:,2*d:3*d] * Ct[t]

  cache = {}
  cache['WLSTM'] = WLSTM
  cache['Hout'] = Hout
  cache['IFOGf'] = IFOGf
  cache['IFOG'] = IFOG
  cache['C'] = C
  cache['Ct'] = Ct
  cache['Hin'] = Hin
  cache['c0'] = c0
  cache['h0'] = h0
```

Deep Learning goes commodity

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$
$$h_t = o_t \circ \sigma_h(c_t)$$

```
def forward(X, WLSTM, c0 = None, h0 = None):
  """
  X should be of shape (n,b,input_size), where n = length of sequence, b = batch size
  """
  n,b,input_size = X.shape
  d = WLSTM.shape[1]/4 # hidden size
  if c0 is None: c0 = np.zeros((b,d))
  if h0 is None: h0 = np.zeros((b,d))

  # Perform the LSTM forward pass with X as the input
  xphpb = WLSTM.shape[0] # x plus h plus bias, lol
  Hin = np.zeros((n, b, xphpb)) # input [1, xt, ht-1] to each tick of the LSTM
  Hout = np.zeros((n, b, d)) # hidden representation of the LSTM (gated ce[illegible]t)
  IFOG = np.zeros((n, b, d * 4)) # input, forget, output, gate (IFO[illegible]
  IFOGf = np.zeros((n, b, d * 4)) # after nonlinearity
  C = np.zeros((n, b, d)) # cell content
  Ct = np.zeros((n, b, d)) # tanh of cell conten[illegible]
  for t in xrange(n):
    # concat [x,h] as input to the [illegible]
    prevh = Hout[t-1] if t > [illegible]
    [illegible][t,:,0] = 1 # b[illegible]
    [illegible]1] = X[t]
    Hin[t[illegible]ze+1:] = prevh
    [illegible]e all [illegible]ivations. dots: (most work is this line)
    [illegible]OG[t] = Hin[t].dot([illegible]
    # non-linearities
    IFOGf[t,:,:3*d] = 1.0/(1.0+np.exp([illegible]:,:3*d])) # sigmoids; these are the gates
    IFOGf[t,:,3*d:] = np.tanh(IFOG[t,:,3*d:[illegible]
    # compute the cell activation
    prevc = C[t-1] if t > 0 else c0
    C[t] = IFOGf[t,:,:d] * IFOGf[t,:,3*d:] + IFOGf[t,:,d:2*d][illegible]
    Ct[t] = np.tanh(C[t])
    Hout[t] = IFOGf[t,:,2*d:3*d] * Ct[t]

  cache = {}
  cache['WLSTM'] = WLSTM
  cache['Hout'] = Hout
  cache['IFOGf'] = IFOGf
  cache['IFOG'] = IFOG
  cache['C'] = C
  cache['Ct'] = Ct
  cache['Hin'] = Hin
  cache['c0'] = c0
  cache['h0'] = h0
```

Deep Learning goes commodity

```
import tensorflow as tf
lstm = tf.contrib.rnn.BasicLSTMCell(lstm_size)
```

LightTAG

So instead of 5 Phds for 2 years

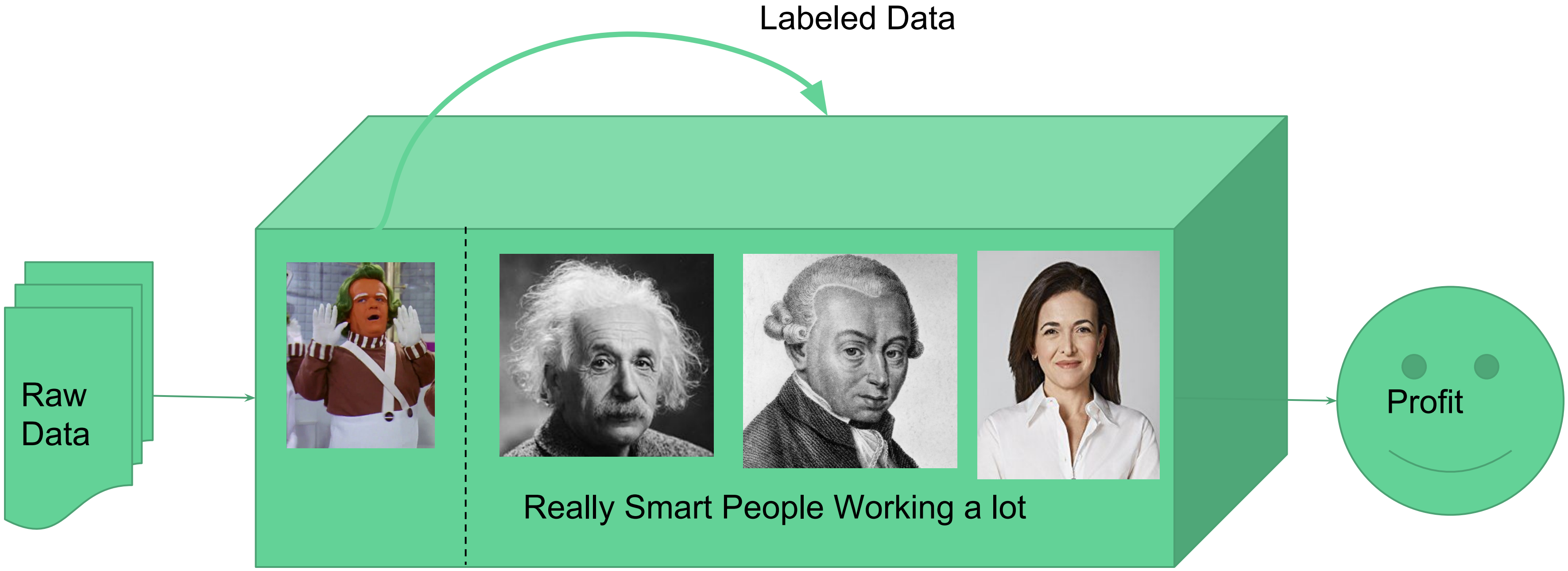
We need 1 engineer for 3 weeks

LightTAG

But their is no ML without labeled data

Data Science is a manufacturing Process

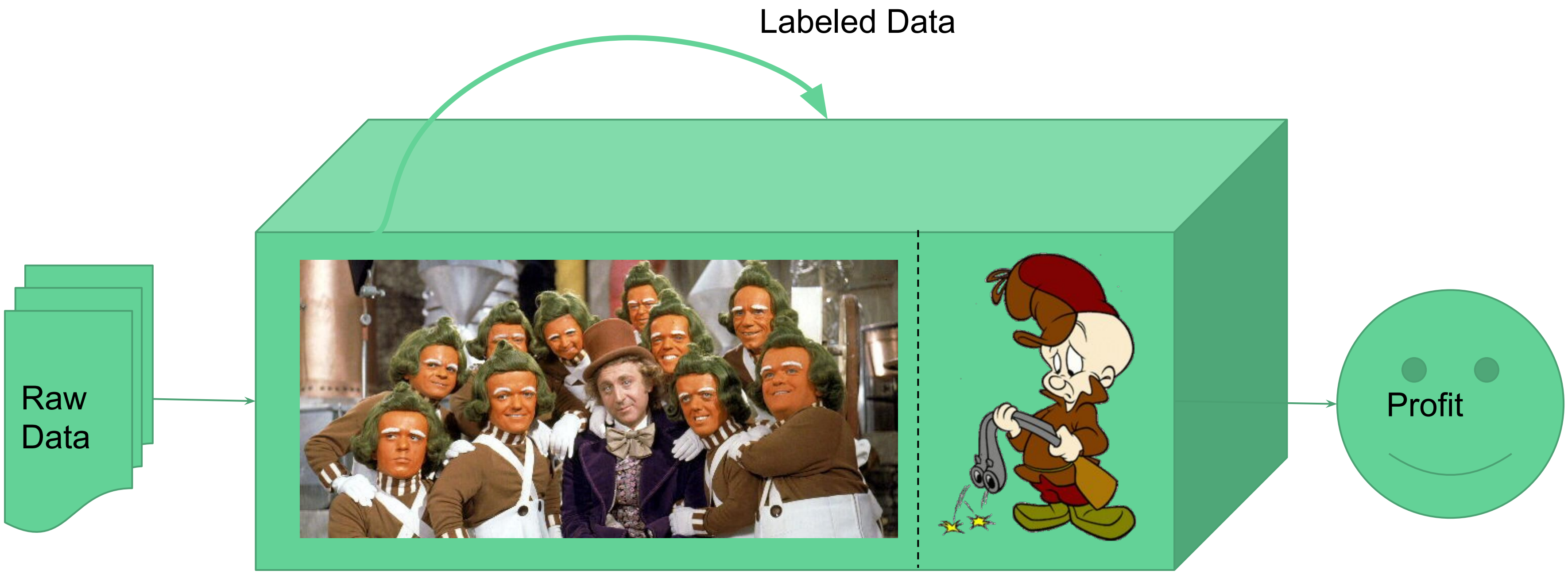
Data Science is a manufacturing Process



LightTAG

Data Science is a manufacturing Process

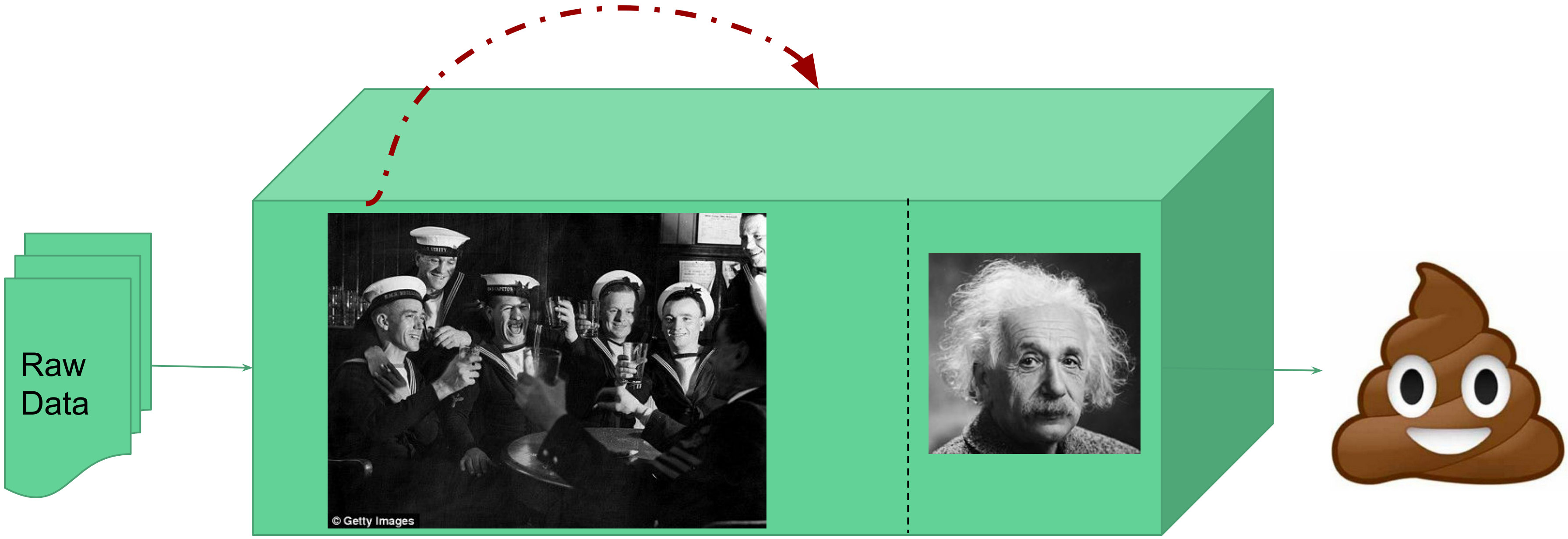
Data Science is a manufacturing Process



Garbage In = Garbage Out

Light TAG

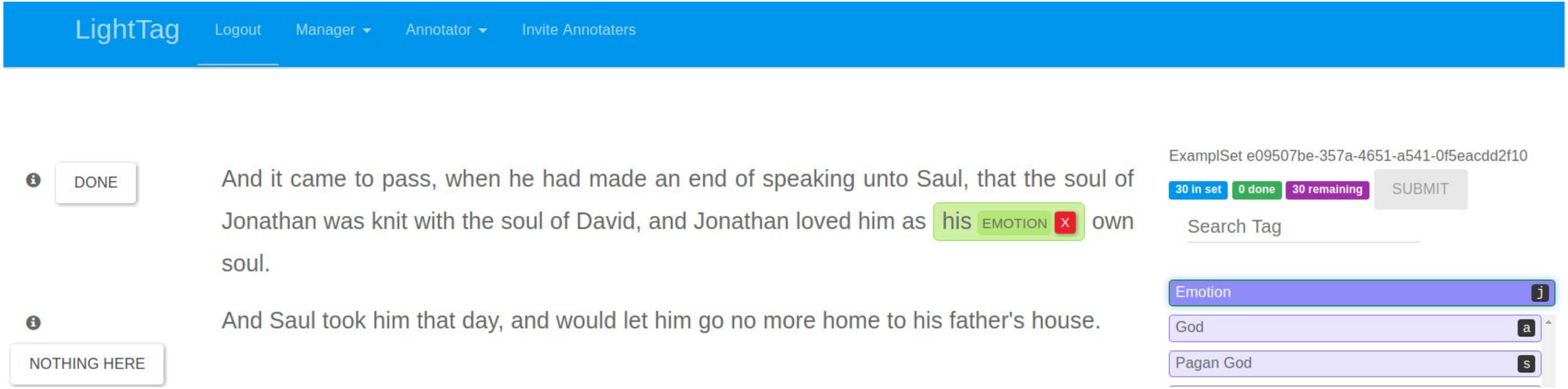
Garbage In = Garbage Out



The Biggest Challenge in Applied ML is getting Labeled Data

LightTAG

That’s why companies pay for LightTag

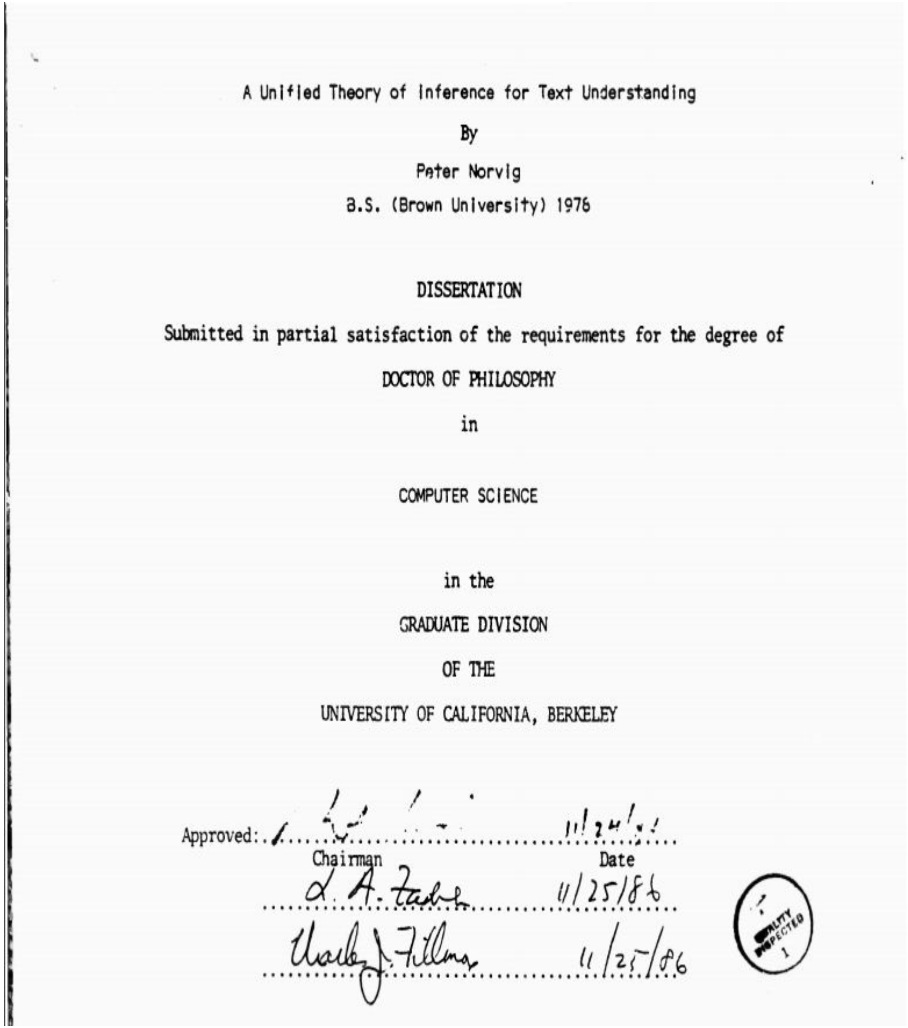


So how does DL apply to NLP?

A fun example on why language is hard

What has 4 letters, sometimes 9 letters, but never has 5 letters.

What we can’t do



A Unified Theory of Inference for Text Understanding

By

Peter Norvig

B.S. (Brown University) 1976

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

in the

GRADUATE DIVISION

OF THE

UNIVERSITY OF CALIFORNIA, BERKELEY

Approved: .... [illegible]
Chairman   Date

.... 11/25/86

.... 11/25/86

People are very good at interpreting texts and making inferences. They generally do not notice when the text is under-specified and they have to make inferences to resolve ambiguities, or to gain a fuller understanding of the text. As an example, consider the following text, excerpted from a book of fairy tales [9]. It will be referred to as text (1).

> In a poor fishing village built on an island not far from the coast of China, a young boy named Chang Lee lived with his widowed mother. Every day, little Chang bravely set off with his net, hoping to catch a few fish from the sea, which they could sell and have a little money to buy bread.
> . . .

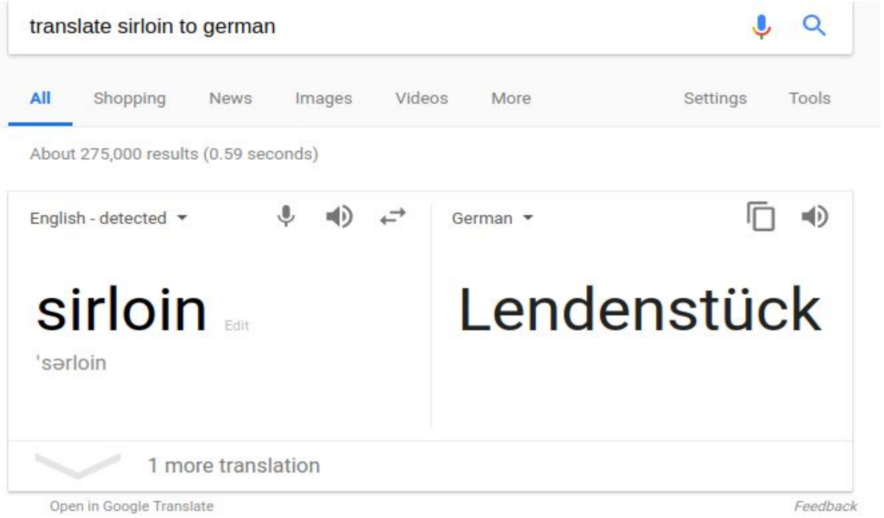
A reader of text (1) should be able to make inferences like these:

(2a) There is a sea which is used by the villagers for fishing, surrounds the island, and forms the coast of China.
(2b) Chang intends to trap fish in his net, which is a fishing net.
(2c) The word *which* in *which they could sell* refers to the fish.
(2d) The word *they* in *they could sell* refers to Chang and his mother.

What we can’t do

- We don’t know how to “infer” *obvious* information from text

LightTAG



What we can do

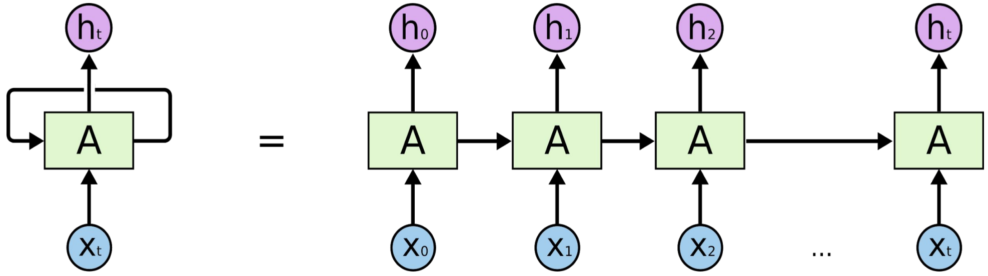
- Translation
- Entity Recognition
- Sentiment Analysis
- Imputation
- Dependency Parsing
- Why is this hard ?
    - "The rules" are ambiguous
    - People don't follow them
    - All data is dirty

Deep Learning for text gave us

- A way to handle arbitrary input/output lengths
- And capture dependencies at multiple time scales

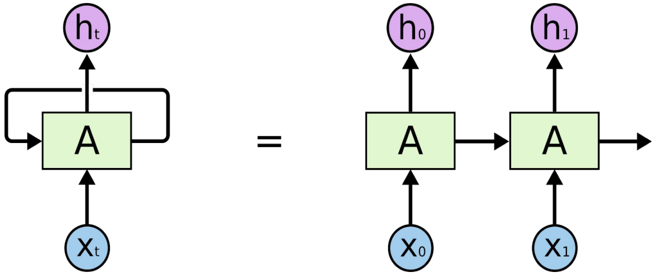
With Recurrent Neural Networks

```
class RNN():
    def __init__(self,hidden_size):
        self.W_hh = np.random.rand(hidden_size,hidden_size)
        self.W_xh = np.random.rand(hidden_size, hidden_size)
        self.W_hy = np.random.rand(hidden_size, hidden_size)
        self.h = np.zeros(hidden_size)
    def step(self,x):
        #update the hidden state
        self.h = np.tanh(np.dot(self.W_hh, self.h) + np.dot(self.W_xh, x))
        # compute the output vector
        y = np.dot(self.W_hy, self.h)
        return y
```



Generating Text

```
/*
 * Increment the size file of the new incorrect UI_FI
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
  rw->name = "Getjbbregs";
  bprm_self_clearl(&iv->version);
  regs->new = blocks[(BPF_STATS << info->historidac)]
  return segtable;
}
```



Unsupervised Sentiment

This is one of Crichton's best books. The characters of Karen Ross, Peter Elliot, Munro, and Amy are beautifully developed and their interactions are exciting, complex, and fast-paced throughout this impressive novel. And about 99.8 percent of that got lost in the film. Seriously, the screenplay AND the directing were horrendous and clearly done by people who could not fathom what was good about the novel. I can't fault the actors because frankly, they never had a chance to make this turkey live up to Crichton's original work. I know good novels, especially those with a science fiction edge, are hard to bring to the screen in a way that lives up to the original. But this may be the absolute worst disparity in quality between novel and screen adaptation ever. The book is really, really good. The movie is just dreadful.

Character Level Translation

(a) Spelling mistakes

DE ori	Warum sollten wir nicht Freunde sei ?
DE src	Warum solltne wir nich Freunde sei ?
EN ref	Why should not we be friends ?
bpe2char	Why are we to be friends ?
char2char	Why should we not be friends ?

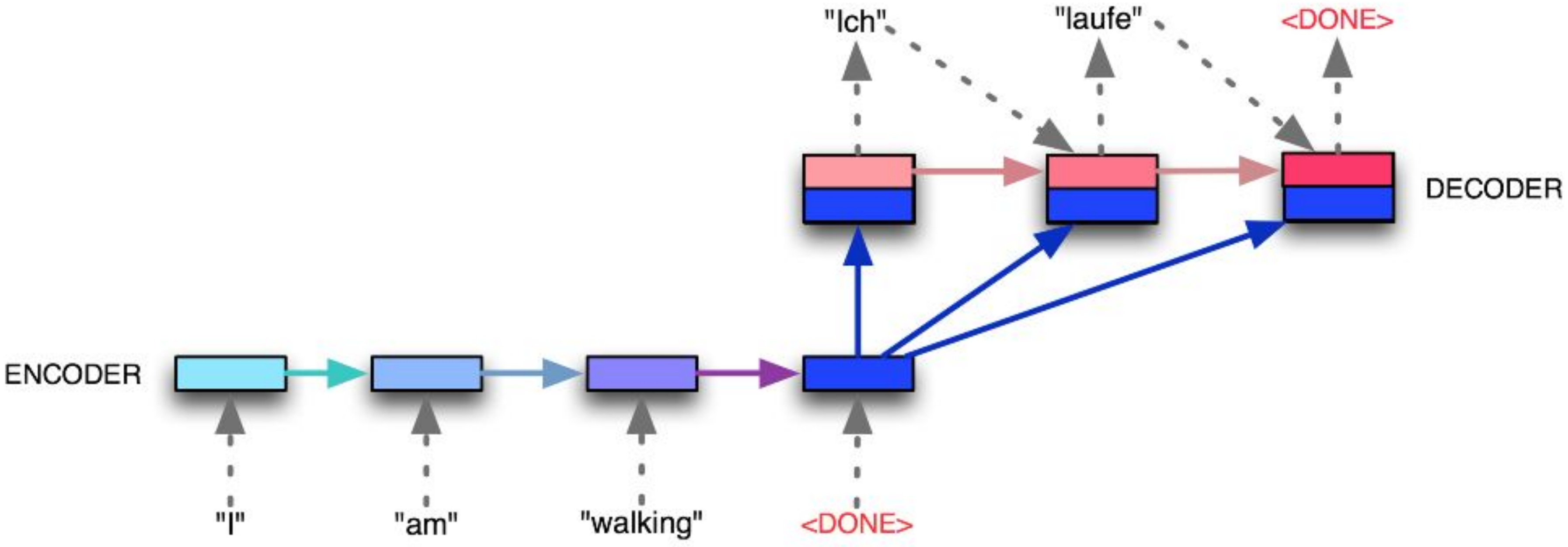
(b) Rare words

DE src	Siebentausendzweihundertvierundfünfzig .
EN ref	Seven thousand two hundred fifty four .
bpe2char	Fifty-five Decline of the Seventy .
char2char	Seven thousand hundred thousand fifties .

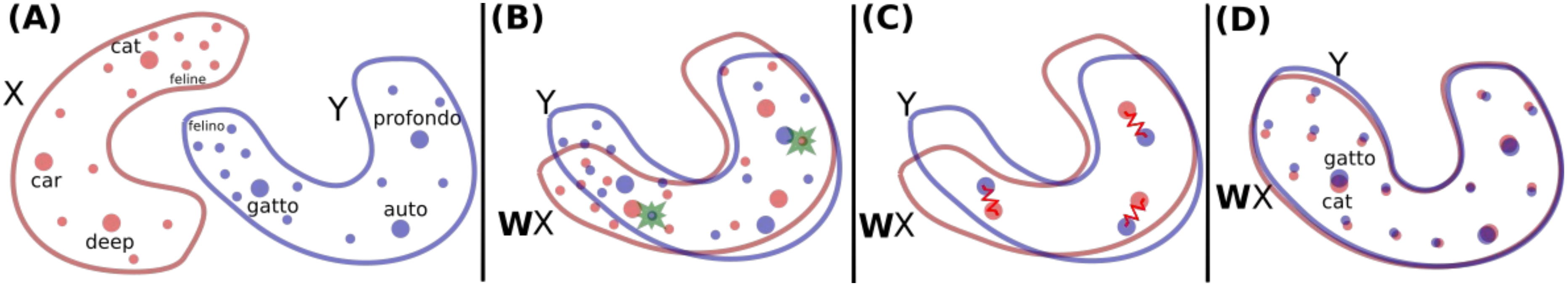
(c) Morphology

Advanced Deep Learning for NLP

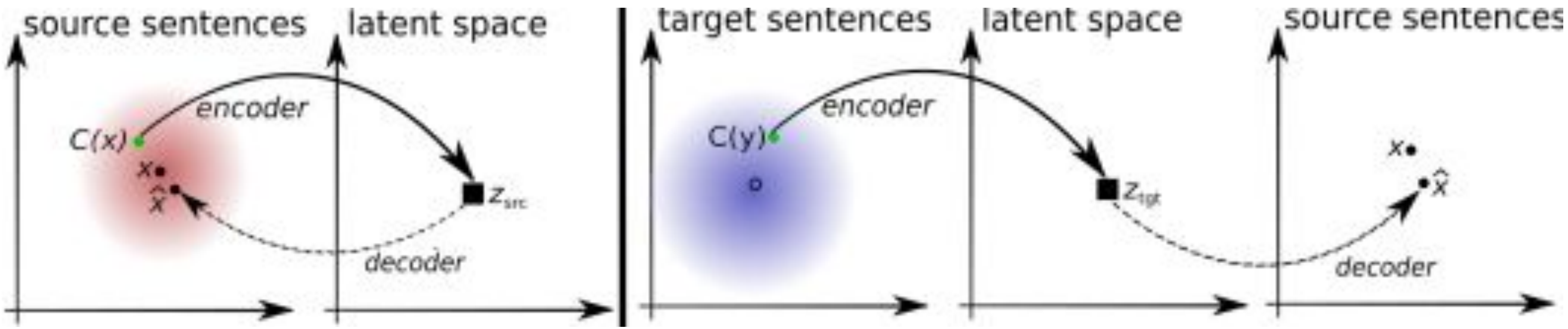
Encoder/Decoder (Seq2Seq) Models



Translating without Parallel Corpus



Translating without Parallel Corpus



Wrapping up

- Deep learning makes data science “easier”
- And open source makes deep learning easier
- But the price we pay is a need for more labeled data

- Language is difficult
- Deep Learning offers technical solutions to sequences
- And can learn clever things

Thank you

Tal Perry
tal@lighttag.io
@thetalperry

Questions ?