

Sequence to Sequence Models

Timothy Chou, Andrew Ding, Albert Ge

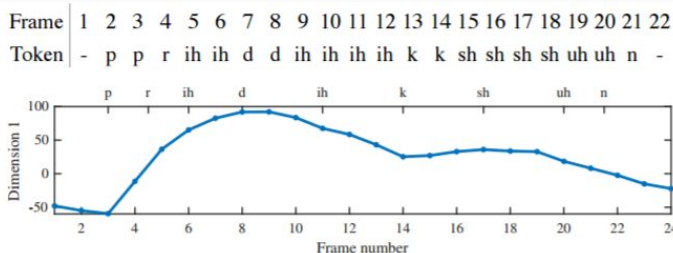
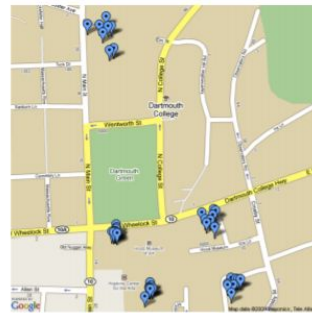
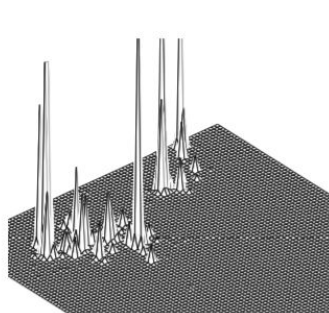
A Decision Tree Framework for Spatiotemporal Sequence Prediction

Taehwan Kim, Yisong Yue, Sarah Taylor, Iain Matthews



Spatiotemporal Sequences (1)

- spatiotemporal sequence prediction
- input sequences generally continuous signals (audio or video)
- output sequences are generally continuous and very high-dimensional
 - exs: face movements, speech, human motion, precipitation nowcasting, crowd flow, user location and movement
- clearly intractable to consider all possible outputs

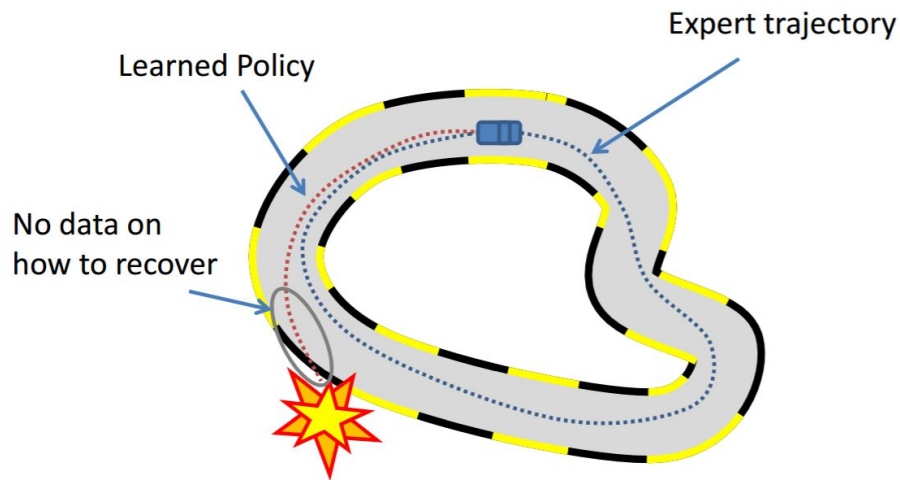


Spatiotemporal Sequences (2)

- some challenges associated with spatiotemporal sequences:
- continuous variation of the outputs
 - sequence prediction often uses discrete output labels (ex: translation, POS tagging)
- often difficult to decompose the input into features
 - prevents standard linear models from working
- variable input length
- possible corruption (missing values, misalignment)
- patterns are generally longer than one timestep
 - prohibits i.i.d assumptions

DAgger (1)

- a prediction approach that attempts to prevent compounding error
- compounding error occurs when we have no data after a slight deviation from the “expert” policy and thus cannot recover
- solution: iteratively augment the dataset



DAgger (2)

Initialize $\mathcal{D} \leftarrow \emptyset$.

Initialize $\hat{\pi}_1$ to any policy in Π .

for $i = 1$ **to** N **do**

Let $\pi_i = \beta_i \pi^* + (1 - \beta_i) \hat{\pi}_i$.

Sample T -step trajectories using π_i .

Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert.

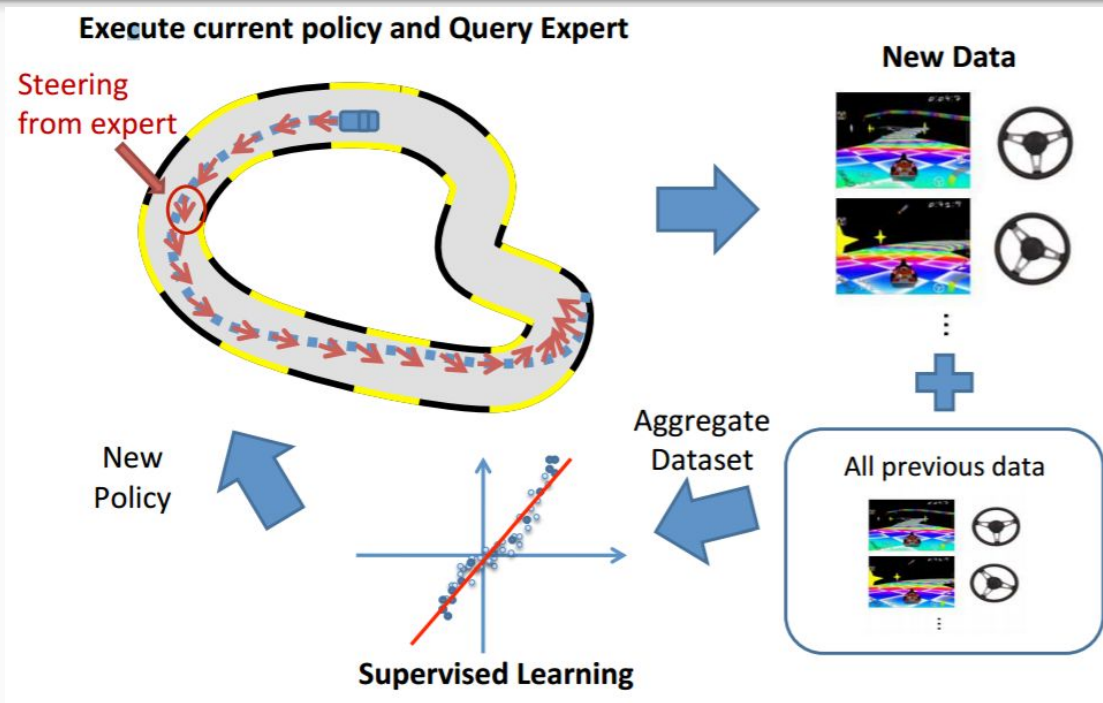
Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$.

Train classifier $\hat{\pi}_{i+1}$ on \mathcal{D} (or use online learner to get $\hat{\pi}_{i+1}$ given new data \mathcal{D}_i).

end for

Return best $\hat{\pi}_i$ on validation.

DAgger (3)



SEARN

- “search and learn”
- roughly similar principles as DAgger, but reformulates sequence prediction as a graph search problem
- does not augment the dataset, assumes that information from policies $1, 2, \dots, i-1$ are captured in policy i

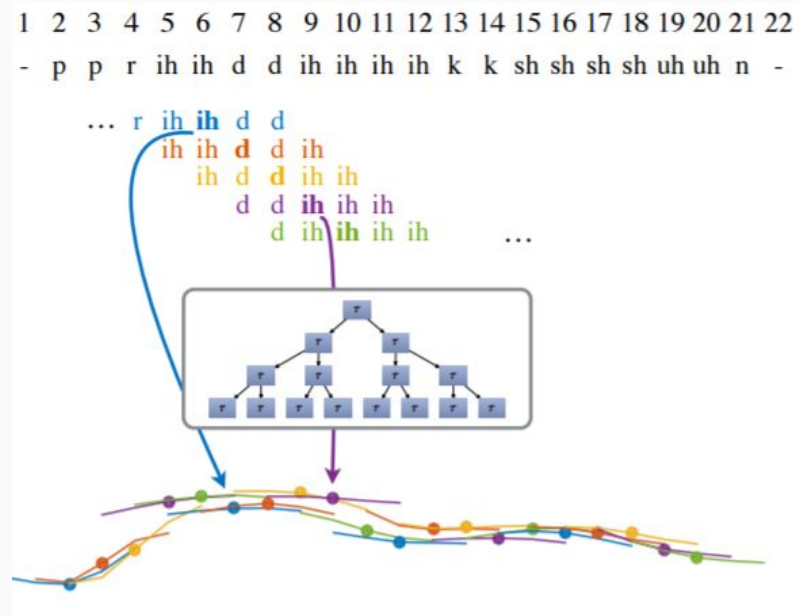
Basic Framework (1)

- at the core, use decision trees
 - only requires: ability to systematically check for splits, and (possibly approximate) measurement of entropy / purity
- in order to convert spatiotemporal sequences to a DT-friendly format, only apply the DT to sliding windows of fixed time
- in visual speech animation: ~11 frames at 30 Hz as input, 150 variables
- new problem: given an input window, predict a 150-variate output
- in general, window must be large enough to capture necessary information, but small enough for tractability

Basic Framework (2)

- The sliding window approach is computationally quick and enables the use of complex base models.
- Important assumptions:
 - Output sequence “preserves order” from the input sequence
 - Output at time t can be successfully predicted with only local information from the input sequence
- Good assumptions for speech animation, automatic camera control, and movement prediction
- Not suitable for tasks like natural language translation

Basic Framework (3)



input

windows

decision tree

output (1 feature)

Basic Framework (4)

Why DTs?

- very interpretable and transparent
- significantly easier to deal with corrupted data
- capable of predicting arbitrary nonlinear outputs from inputs, even when the inputs cannot be interpreted as numbers

Even so, it is not unthinkable that the DT base model could be replaced by something else.

Decision Tree Training (1)

- loss function: L2 error
- property of this loss function: given a leaf node, the leaf prediction should be the mean of all predictions at the leaf
 - this minimizes the loss function

$$L_S(h) = \sum_{(x,y) \in S} \|h(x) - y\|^2$$

Decision Tree Training (2)

- in a discrete-output DT, we try to minimize entropy / impurity for our splitting criterion
 - attempt to make all leaf nodes output the same prediction
- natural extension in this case: use *variance* as splitting criterion; want leaves with variance 0
 - convenient because the variance is precisely the loss at any leaf node

$$\begin{aligned} L^2(\hat{\mathbf{Y}}) &= |\hat{\mathbf{Y}}| \text{variance}(\hat{\mathbf{Y}}) \\ &= |\hat{\mathbf{Y}}| \sum_{t=1}^{K_y} \sum_{d=1}^D \text{variance} \left(\left\{ \hat{y}_{i,t}^{(d)} \mid \hat{\mathbf{y}}_i \in \hat{\mathbf{Y}} \right\} \right) \end{aligned}$$

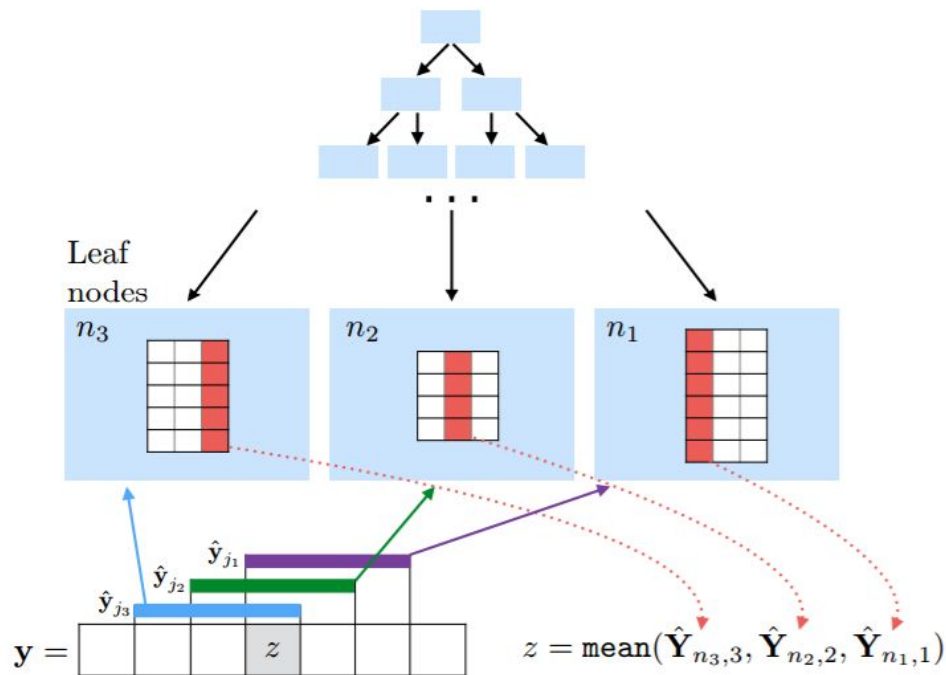
Correcting Missing Values (1)

- Some of the outputs might be missing.
- Suppose that in sequence i , the t th frame in the d th dimension is missing. Call this missing value $z_{i,t}^{(d)}$.
- The modified training set consists only of windows of k timesteps. (In speech animation, $k = 5$.)
- Therefore, $z_{i,t}^{(d)}$ appears in k windows, as frame number $1, 2, \dots, k$.

Correcting Missing Values (2)

- In training the decision tree, these windows can appear in up to k different leaf nodes.
- Infer value of $y_{i,t}^{(d)}$ as the value that minimizes the loss function at those leaf nodes.
- Because loss function is variance, this is conveniently the mean over all labels in all leaf nodes containing $z_{i,t}^{(d)}$.
- DT split criterion is only based on x ; tree is unchanged.

Correcting Missing Values (3)



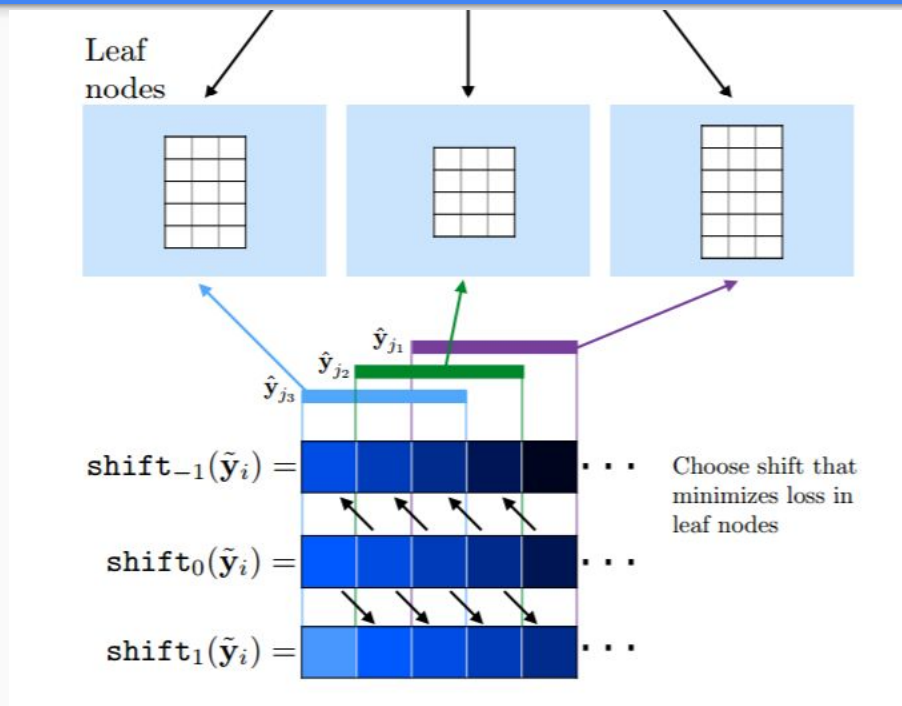
Correcting Missing Values (4)

- During training labels with missing value are weighed less.
- Weight is set to 0 during first training to train a preliminary DT without using any corrupted labels.
- Afterwards, missing values are inferred and the DT is retrained. Repeat as necessary.

Correcting Misalignments (1)

- Some sequences y_i might be misaligned. (Note that y_i is a full training sequence, not just a window.)
- Consider shifts of $[-a, a]$ for hyperparameter a .
- Pick the shift that minimizes average variance at the leaf nodes that include some window of y_i .
- As before, since DT split is based on x_i only, this does not affect the tree structure.

Correcting Misalignments (2)



Extension to Random Forests

- Decision trees have low bias (since they can become extremely complicated predictors), but high variance.
- To reduce variance and increase accuracy, we may consider ensemble methods. Most obvious: random forests.
- Train a number of trees, where each tree only considers a subset of possible splits at each node--allows stochasticity and decreases training time.
- Average results of all trees for final predictions.
- Could be done here: window DTs replaced with window RFs.

Experiment: Visual Speech Animation (1)

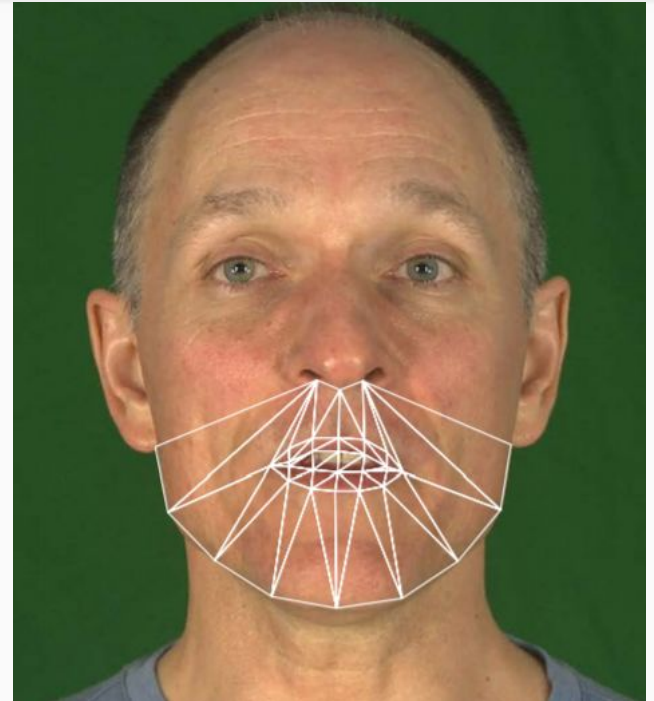


- dataset of an actor speaking ~2500 sentences, with face videos
- inputs: 30 Hz phoneme sequences



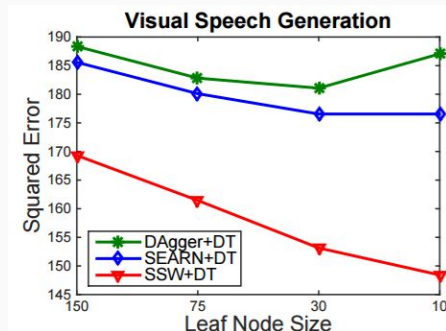
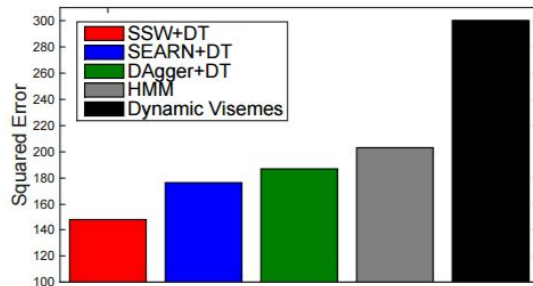
Experiment: Visual Speech Animation (2)

- parametrize outputs (facial expressions) into 30 dimensions
- label some vertex locations, then use PCA to find a good feature representation
- facial expressions decomposed into 30-dimensional vectors



Experiment: Visual Speech Animation (3)

- use sliding window of 11 input frames, 5 output frames
- i.e. given 11 phonemes, predict a sequence of 5 facial expressions (30 dimensions each) that correspond to it
- after transforming the training set to windows, about 200,000 data points
- compared vs SEARN and DAgger



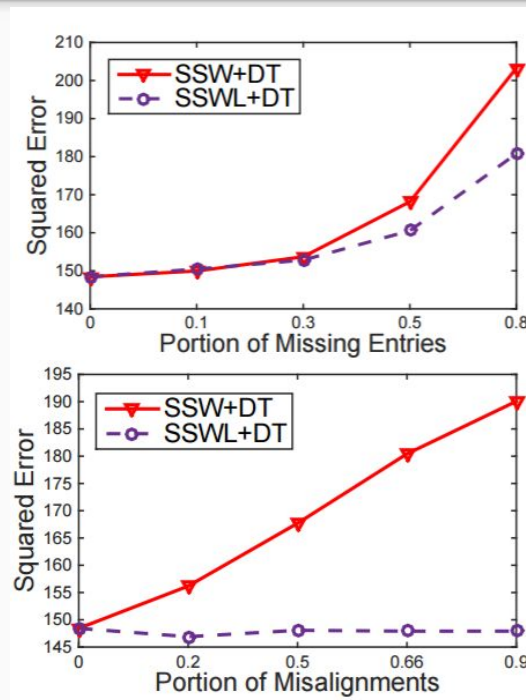
Experiment: Visual Speech Animation (4)

- user study: have humans pick the more “realistic” animation
- sliding window + DTs beats other methods easily, but not ground truth

COMPARISON	WIN/LOSS	VOTE DIFF
SSW+DT vs Dynamic Visemes [33]	47 / 3	3.32
SSW+DT vs HMM [40]	50 / 0	3.76
SSW+DT vs SEARN+DT [7]	44 / 6	1.96
SSW+DT vs DAgger+DT [25]	45 / 5	2.12
SSW+DT vs Ground Truth	10 / 40	-1.68

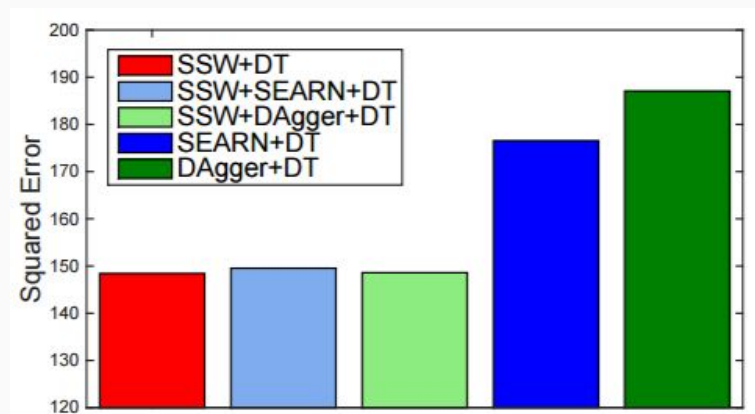
Experiment: Visual Speech Animation (5)

- missing values dataset created by removing values randomly
- misalignment dataset created randomly select sentences to misalign in $[-3,3]$
- latent variable approach is effective at reducing error even through corruption



Experiment: Visual Speech Animation (6)

- Could imagine integrating SEARN and DAgger with the SSW+DT approach
- decision tree instead predicts a length-5 sequence based on the last 5 predictions and the input x
- does not improve performance



Sequence to Sequence Learning with Neural Networks (2014)

Ilya Sutskever, Oriol Vinyals, Quoc V. Le

Background

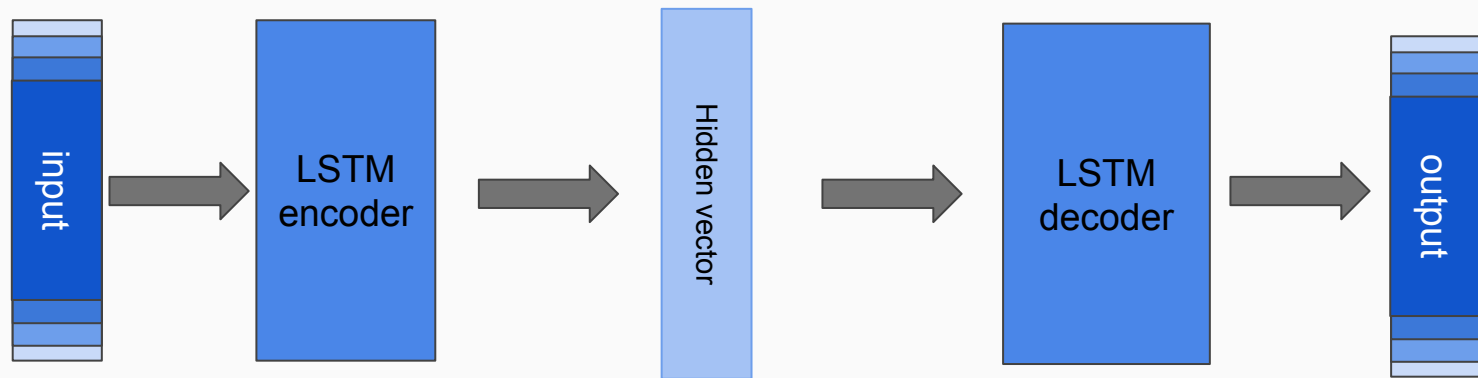
DNNs have input and target vectors with fixed dimensionality

Limitations: some problems have sequence within unknown length a-priori

- Speech recognition, machine translation
- **Q&A: the length of a question is not related to the length of the answer**

Idea

Use a LSTM encoder-decoder network



Model: Formal representation

The LSTM-LM attempts to estimate the *conditional probability*

$$p(y_1, \dots, y_{T'} | x_1, \dots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \dots, y_{t-1})$$

And optimizes according to the objective function

$$\frac{1}{|\mathcal{S}|} \sum_{(T,S) \in \mathcal{S}} \log p(T|S)$$

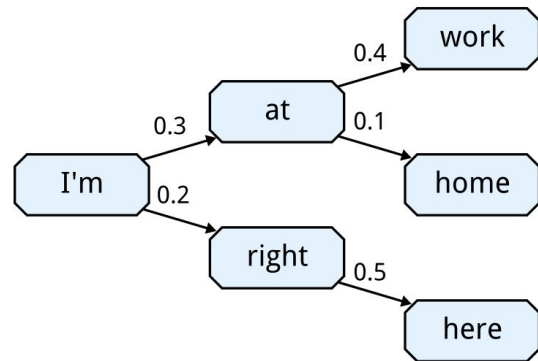
Model: Computing most likely hypothesis

It's intractable to sift through all hypotheses that the network outputs

Use a heuristics-guided algorithm called left-right *beam search decoder*

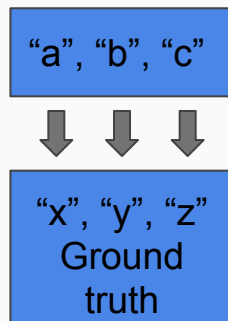
Beam size: what depth to consider

Example of a beam search tree



Model

New strategy: *reverse* the input sentence



Model

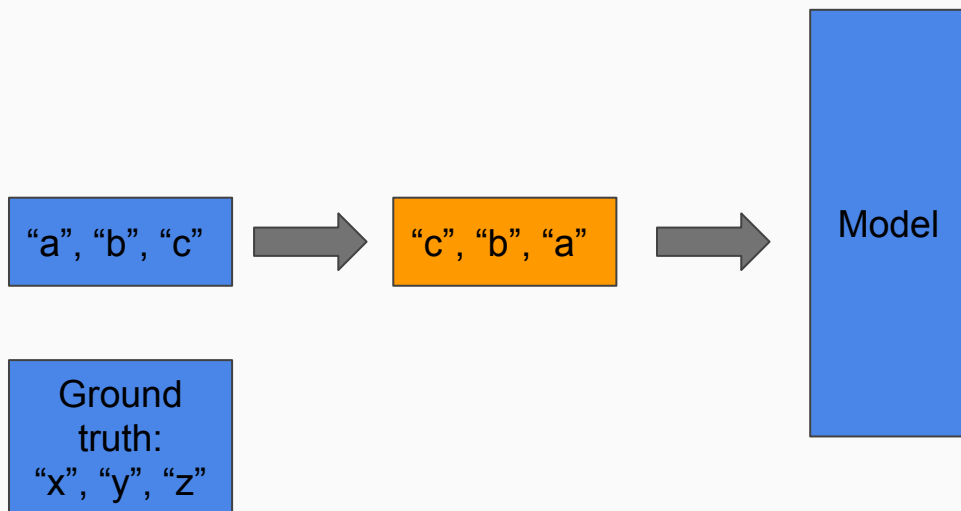
New strategy: *reverse* the input sentence



Ground
truth:
"x", "y", "z"

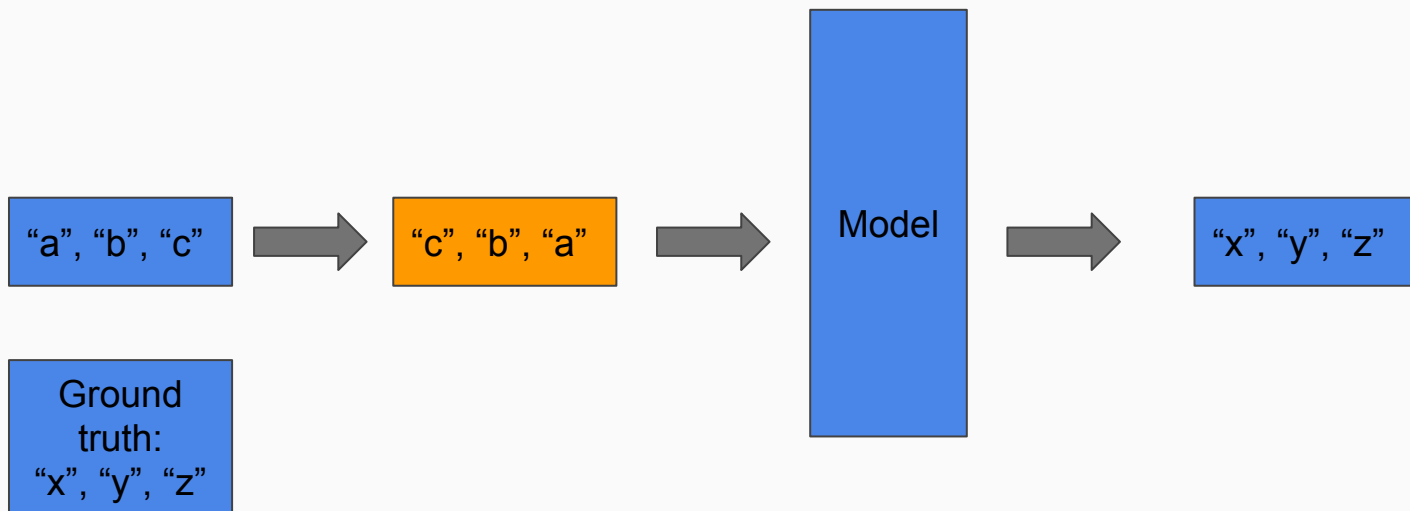
Model

New strategy: *reverse* the input sentence

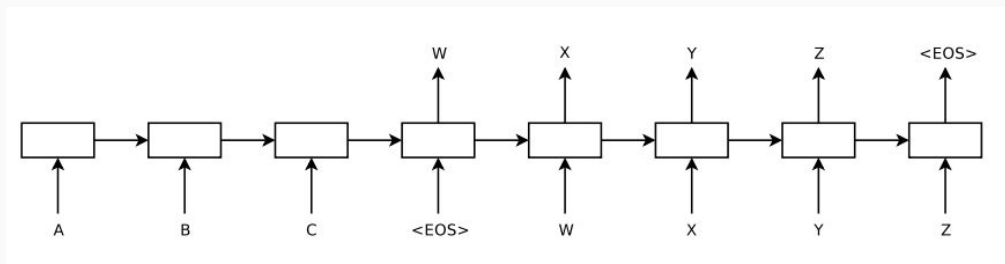


Model

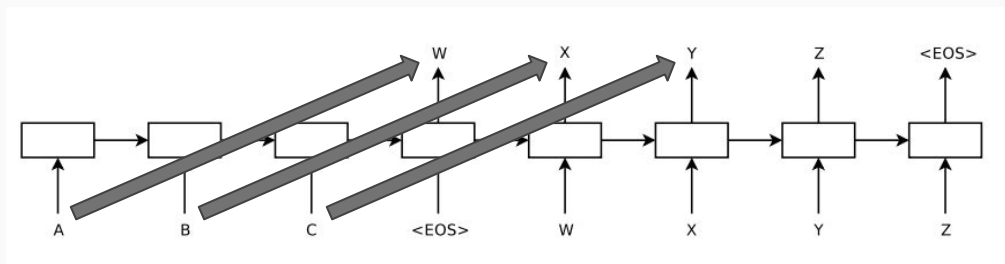
New strategy: *reverse* the input sentence



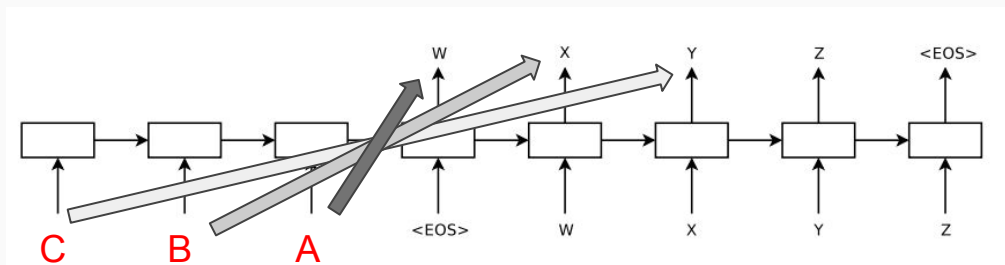
Why do this?



Why do this?



Why do this?



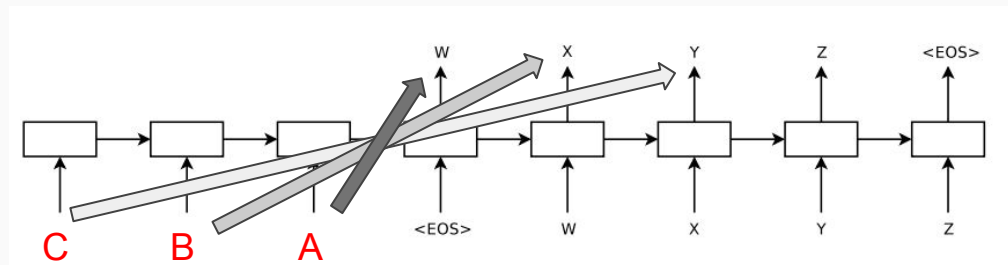
Why do this?

Theory:

many short-term dependencies
in the dataset

Decreases *minimal time lag*

Backpropagation can better
establish connections



Target Application - WMT '14 ENG-FR translation task

Sportsman Jhonathan Florez jumped from a helicopter above Bogota, the capital of Colombia, on Thursday.



Le sportif Jhonathan Florez a sauté jeudi d'un hélicoptère au-dessus de Bogota, la capitale colombienne.

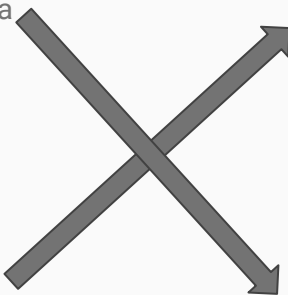
Target Application - WMT '14 ENG-FR translation task

Nous sommes heureux que la FAA reconnait qu'apas; une experience agreable passager n'apas; est pas incompatible avec la securite et la securite "; , dit Roger Dow , PDG de l'apas; US Travel Association .

Nous sommes heureux de la FAA reconnait qu'apas; une experience agreable passager n'apas; est pas incompatible avec la securite et la surete "; , a dit Roger Dow , PDG de l'apas; US Travel Association .

Nous sommes heureux de la FAA reconnait qu'apas; une experience agreable passager n'apas; est pas incompatible avec la securite et la surete "; , a dit Roger Dow , PDG de l'apas; US Travel Association .

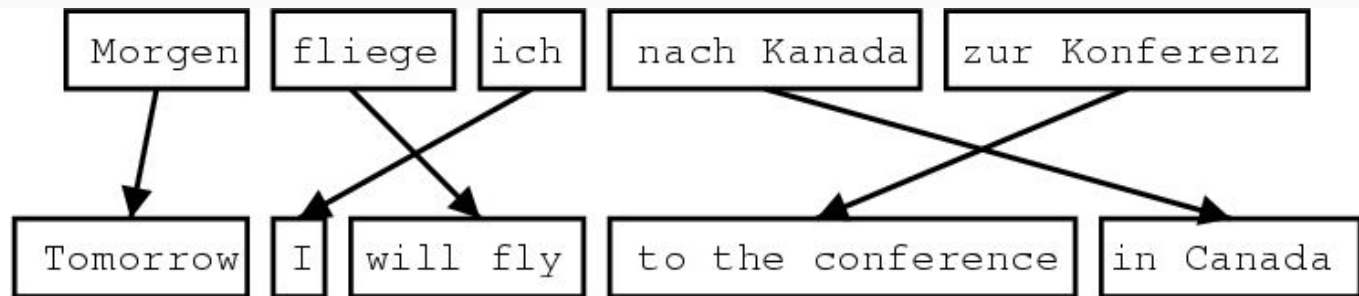
Nous sommes heureux que la FAA reconnait qu'apas; une experience agreable passager n'apas; est pas incompatible avec la securite et la securite "; , dit Roger Dow , PDG de l'apas; US Travel Association .



Phrase Based SMT

Used in both state of the art*, and baseline

Organizes sentences into phrase groups



*Edinburgh's Phrase-based Machine Translation Systems for WMT-14 (Durrant et. al, 2014).

BLEU Score (Bilingual Evaluation Understudy)

Inexpensive, correlated with human judgement*

Algorithm:

1. Take a weighted geometric mean of modified n-gram precisions, up to length N
2. Compute the brevity penalty
3. Compute BLEU score

*BLEU: a Method for Automatic Evaluation of Machine Translation (Papineni et. al, 2002)

BLEU Score (Bilingual Evaluation Understudy)

Algorithm:

1. Take a **weighted geometric mean of modified n-gram precisions, up to length N**
2. Compute the brevity penalty
3. Compute BLEU score

Modified n-gram precision:

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-gram} \in C} Count_{clip}(n\text{-gram})}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-gram}' \in C'} Count(n\text{-gram}')}.$$

BLEU Score (Bilingual Evaluation Understudy)

Algorithm:

1. Take a **weighted geometric mean of modified n-gram precisions, up to length N**
2. Compute the brevity penalty
3. Compute BLEU score

Example: (n=1)

Candidate (machine translation):

the the the the the the the.

Reference 1 (human translation):

The cat is on the mat.

Reference 2:

There is a cat on the mat.

BLEU Score (Bilingual Evaluation Understudy)

Algorithm:

1. Take a **weighted geometric mean of modified n-gram precisions, up to length N**
2. Compute the brevity penalty
3. Compute BLEU score

Example: (n=1)

Candidate (machine translation):

the the the the the the the.

Reference 1 (human translation):

The cat is on the mat.

Reference 2:

There is a cat on the mat.

BLEU Score (Bilingual Evaluation Understudy)

Algorithm:

1. Take a **weighted geometric mean of modified n-gram precisions, up to length N**
2. Compute the brevity penalty
3. Compute BLEU score

Example: (n=1)

Candidate (machine translation):

the the the the the the the.

Reference 1 (human translation):

The cat is on the mat.

Reference 2:

There is a cat on the mat.

BLEU Score (Bilingual Evaluation Understudy)

Algorithm:

1. Take a **weighted geometric mean of modified n-gram precisions, up to length N**
2. Compute the brevity penalty
3. Compute BLEU score

Example: (n=1)

Candidate (machine translation):

the the the the the the the.

Reference 1 (human translation):

The cat is on the mat.

Reference 2:

There is a cat on the mat.

Modified unigram precision: 2/7

BLEU Score (Bilingual Evaluation Understudy)

Algorithm:

1. Take a weighted geometric mean of modified n-gram precisions, up to length N
2. **Compute the brevity penalty**
3. Compute BLEU score

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} .$$

c is length of candidate translation, and r is effective reference corpus length

BLEU Score (Bilingual Evaluation Understudy)

Algorithm:

1. Take a weighted geometric mean of modified n-gram precisions, up to length N
2. Compute the brevity penalty
3. **Compute BLEU score**

$$\text{BLEU} = \text{BP} \cdot \exp \left(\sum_{n=1}^N w_n \log p_n \right).$$

$$\log \text{BLEU} = \min \left(1 - \frac{r}{c}, 0 \right) + \sum_{n=1}^N w_n \log p_n.$$

Notable Implementation Details

Used a 4-layer LSTM, with 1000 cells at each layer, for both encoder/decoder networks

All sentences within a minibatch (size 128) roughly the same length

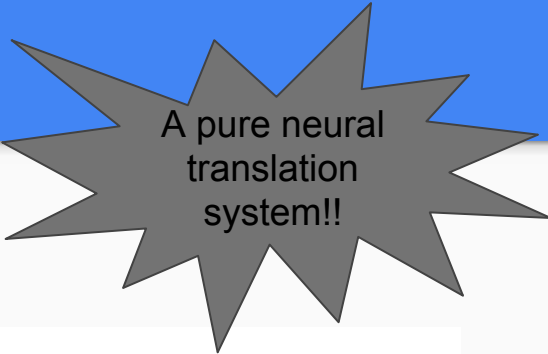
Model Performance

ENG-FR translation task

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	34.81

Model Performance

ENG-FR translation task



A pure neural
translation
system!!

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	<u>33.30</u>
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	<u>34.81</u>

Model Performance

Baseline rescoring task

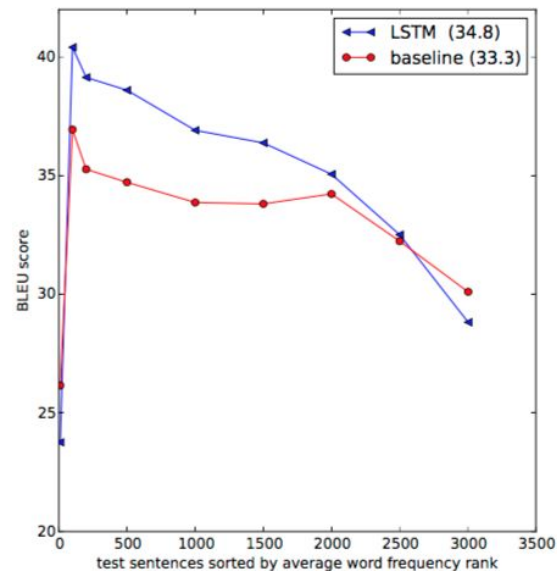
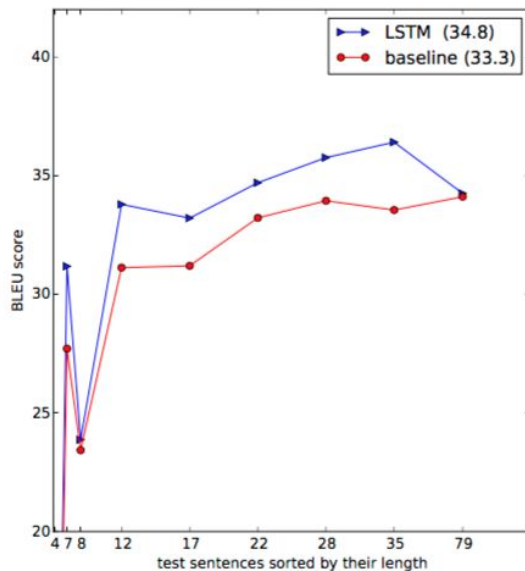
Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
State of the art [9]	37.0
Rescoring the baseline 1000-best with a single forward LSTM	35.61
Rescoring the baseline 1000-best with a single reversed LSTM	35.85
Rescoring the baseline 1000-best with an ensemble of 5 reversed LSTMs	36.5
Oracle Rescoring of the Baseline 1000-best lists	~45

Model Performance

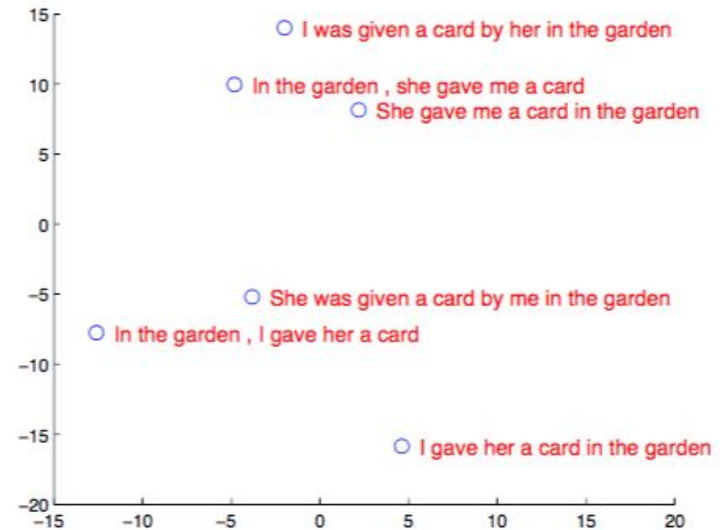
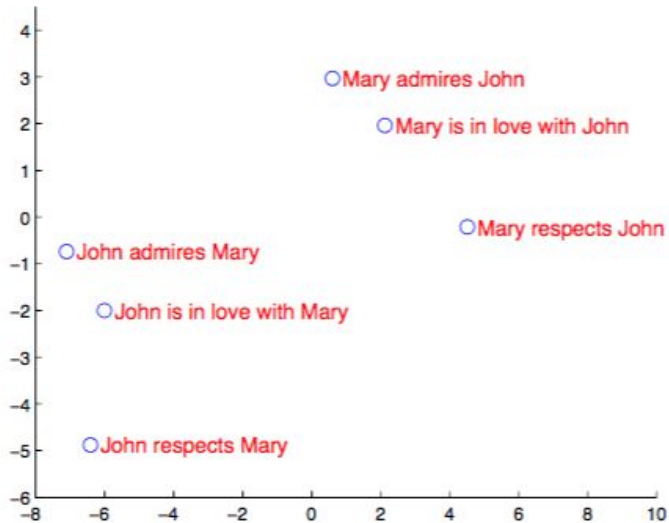
Baseline rescoring task

Method	test BLEU score (ntst14)
Baseline System [29]	<u>33.30</u>
Cho et al. [5]	34.54
State of the art [9]	37.0
Rescoring the baseline 1000-best with a single forward LSTM	35.61
Rescoring the baseline 1000-best with a single reversed LSTM	35.85
Rescoring the baseline 1000-best with an ensemble of 5 reversed LSTMs	<u>36.5</u>
Oracle Rescoring of the Baseline 1000-best lists	~45

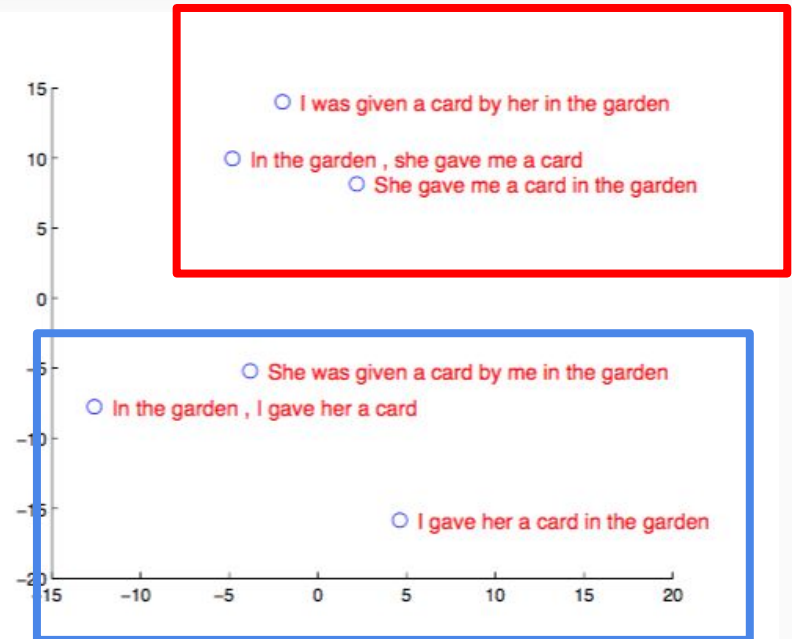
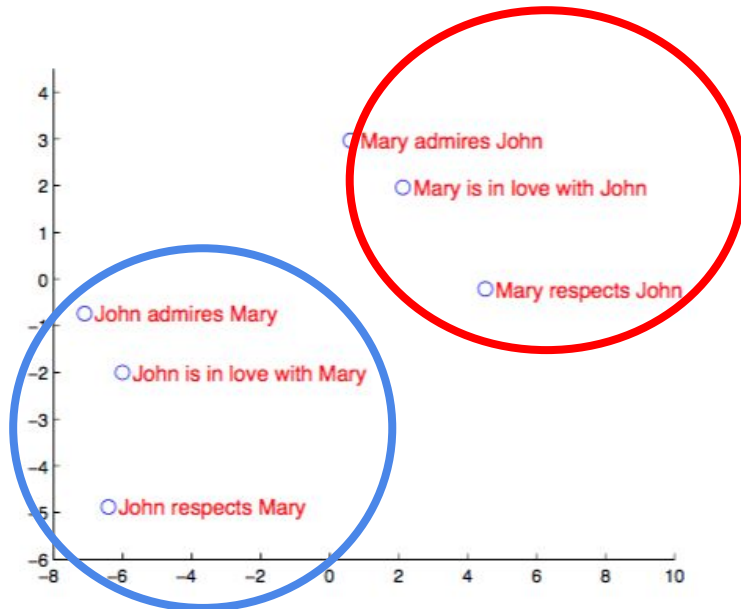
Model Performance



Model Analysis



Model Analysis



Where to go from here?

Vinyals et. al (2015b) also used the reversed-input strategy

- Marginal improvement of 0.5% on F1 score (parsing task)

Vinyals, Bengio, Keveman (2016) - order matters in seq2seq models

- Discuss how to order unstructured inputs (e.g., sets)

Bahdanau et. al (2015) - Alignment based model

Approach

- Encoder-Decoder translation performance deteriorates as sentences get longer
 - Fixed-length encoding bottlenecks amount of data
- Softly align parts of the input sentence that are relevant to predicting the next word

Goal

Encoder-decoder model

$$p(y_t \mid \{y_1, \dots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c),$$

Alignment model

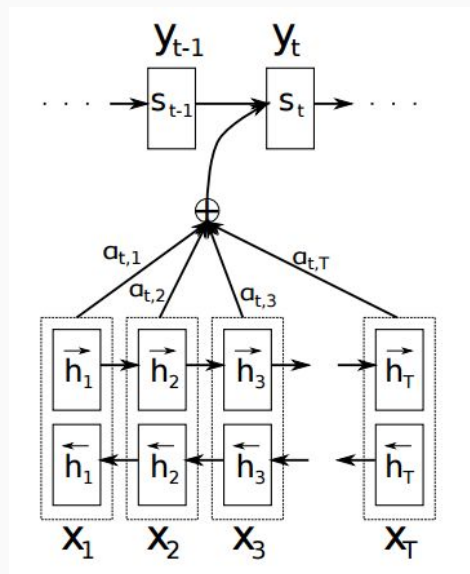
$$p(y_i \mid y_1, \dots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$

Context Vector

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

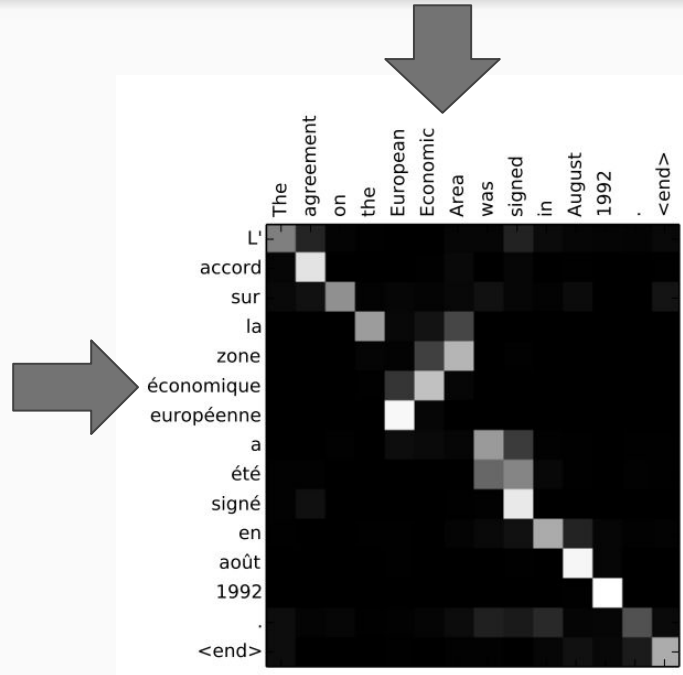
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \quad e_{ij} = a(s_{i-1}, h_j)$$

Annotation

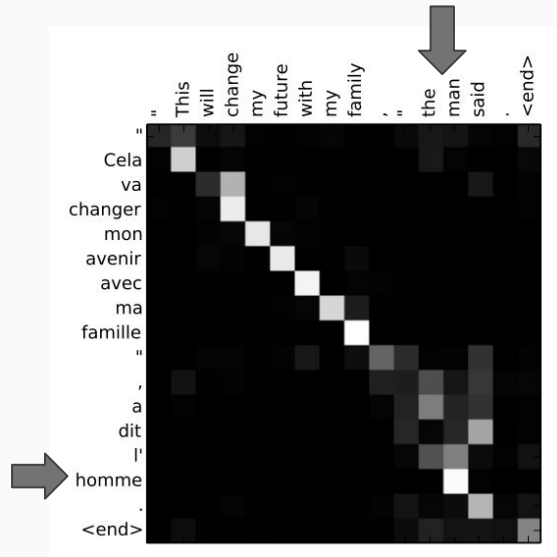


$$h_j = \left[\vec{h}_j^\top; \overleftarrow{h}_j^\top \right]^\top$$

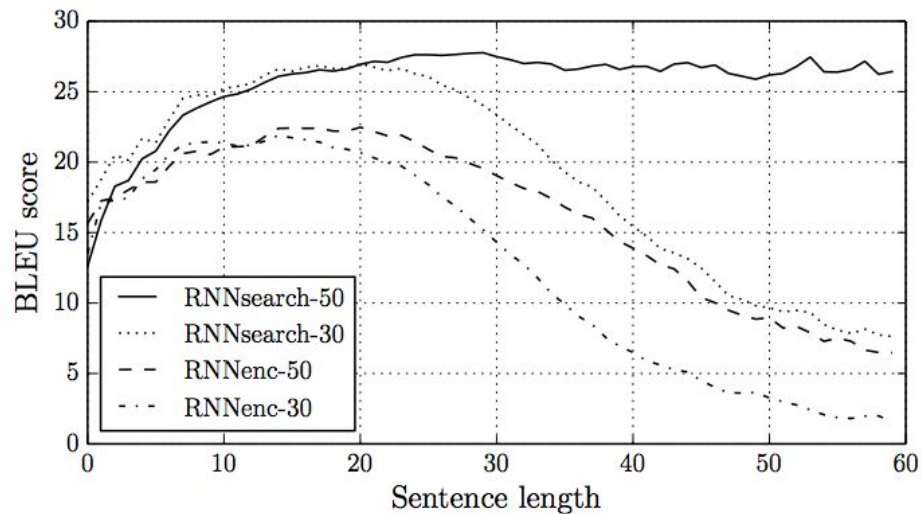
Analysis of Alignment



Analysis of Alignment



Performance



Performance Deterioration

- An admitting privilege is the right of a doctor to admit a patient to a hospital or a medical centre to carry out a diagnosis or a procedure, based on his status as a health care worker at a hospital
- Un privilege d admission est le droit d'un un medecin de reconaitre un patient a l'hopital ou un centre medical d'un diagnostic ou de prendre un diagnostic en fonction de son etat de sante
 - Translates to "based on his state of health"
- Un privilege d' admission est le droit d'un medecin d'admettre un patient a un hopital ou un centre medical pour effectuer un diagnostic ou une procedure, selon son statut de travailleur des soins de sante a l' hopital

Performance Cont'd

Model	All	No UNK ^o
RNNencdec-30	13.93	24.19
RNNsearch-30	21.50	31.44
RNNencdec-50	17.82	26.71
RNNsearch-50	26.75	34.16
RNNsearch-50*	28.45	36.15
Moses	33.30	35.63