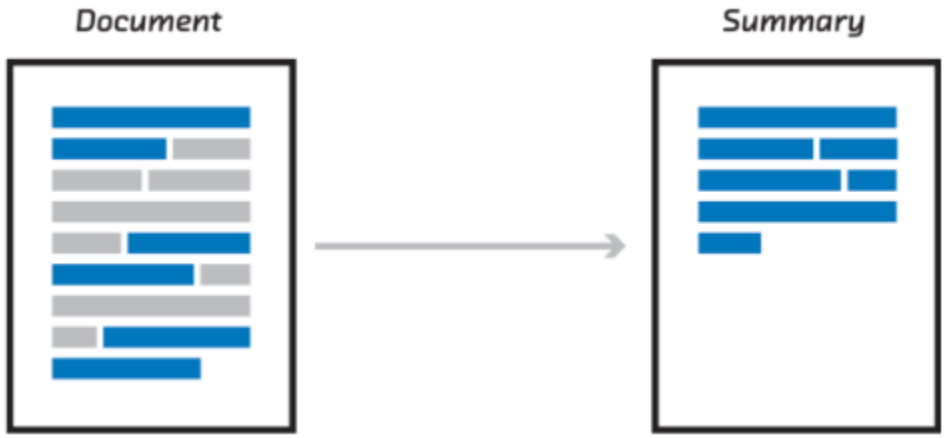


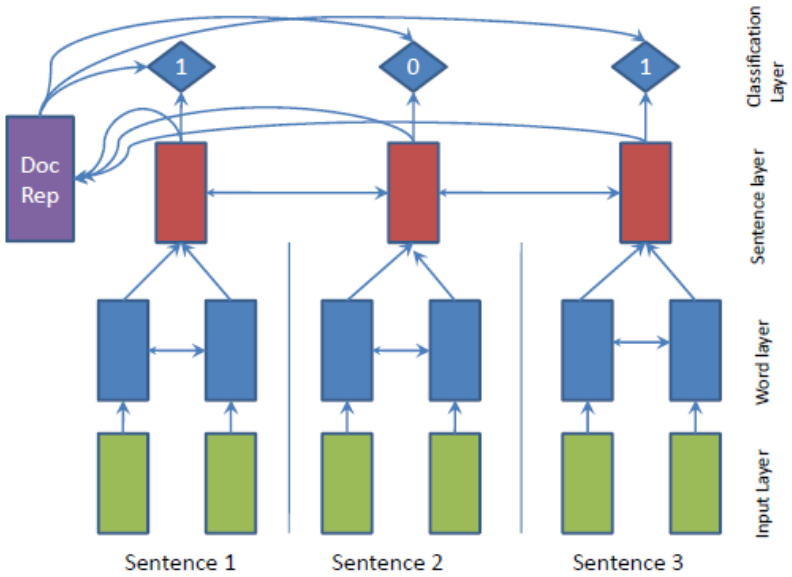
Neural Network-based Extractive Summarization

Pilsung Kang
School of Industrial Management Engineering
Korea University

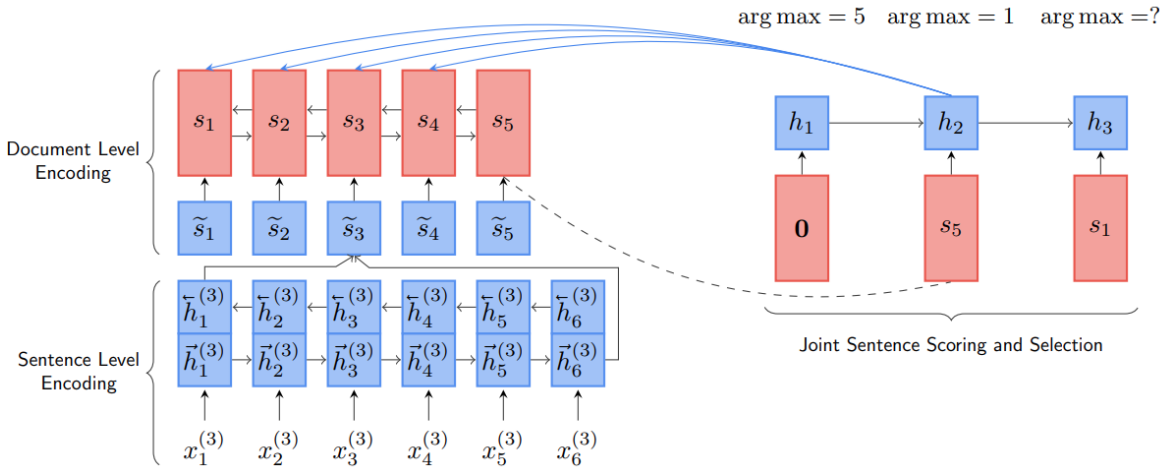
Neural Network-based Extractive Summarization

- Extractive Summarization
  - ✓ Formulated as a sequence classification problem
    - Each sentence is visited sequentially in the original document order and binary decision is made in terms of whether or not it should be included in the summary

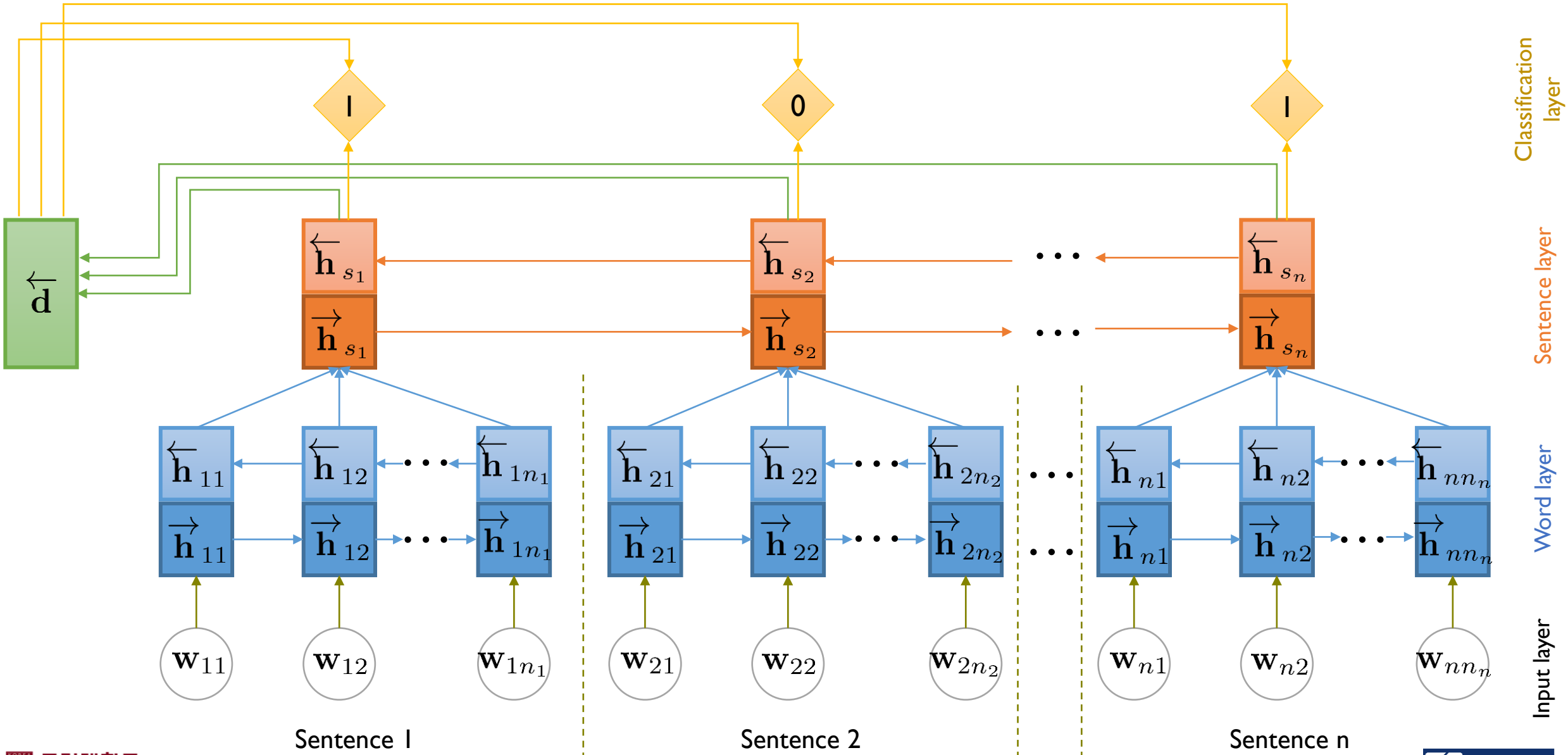
[SummaRuNNer]



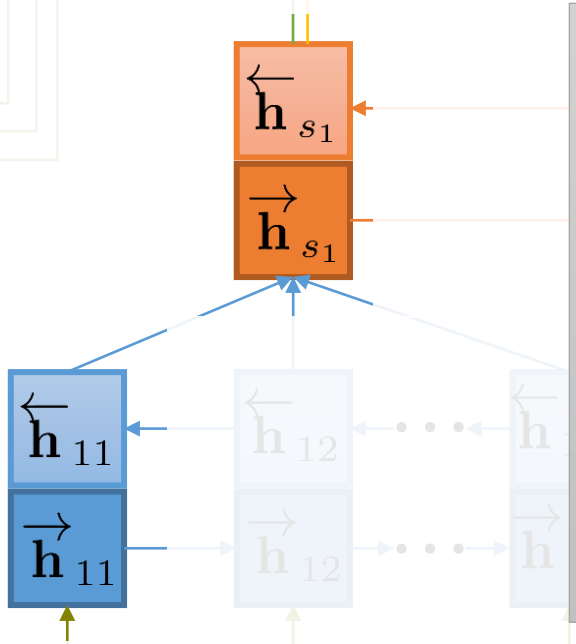
[NeuSum]



SummaRuNNer: Overall Structure



SummaRuNNer: Basic Building Block



Gated Recurrent Unit (GRU)

$$\mathbf{u}_j = \sigma(\mathbf{W}_{ux}\mathbf{x}_j + \mathbf{W}_{uh}\mathbf{h}_{j-1} + \mathbf{b}_u)$$

$$\mathbf{r}_j = \sigma(\mathbf{W}_{rx}\mathbf{x}_j + \mathbf{W}_{rh}\mathbf{h}_{j-1} + \mathbf{b}_r)$$

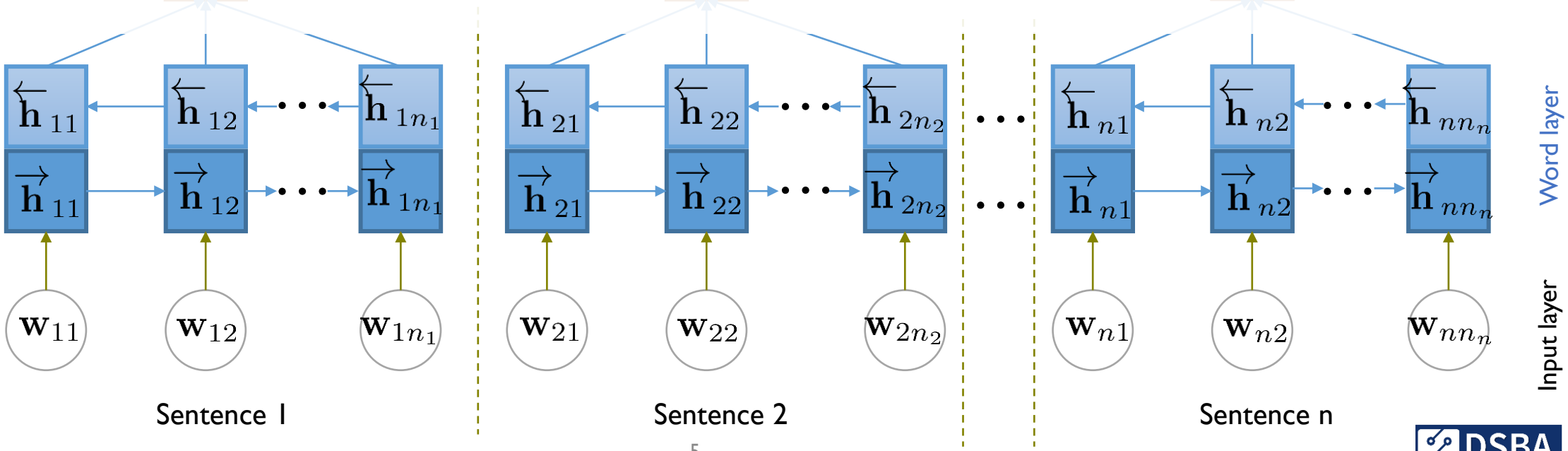
$$\mathbf{h}'_j = tanh(\mathbf{W}_{hx}\mathbf{x}_j + \mathbf{W}_{hh}(\mathbf{r}_j \odot \mathbf{h}_{j-1}) + \mathbf{b}_h)$$

$$\mathbf{h}_j = (1 - \mathbf{u}_j) \odot \mathbf{h}'_j + \mathbf{u}_j \odot \mathbf{h}_{j-1}$$

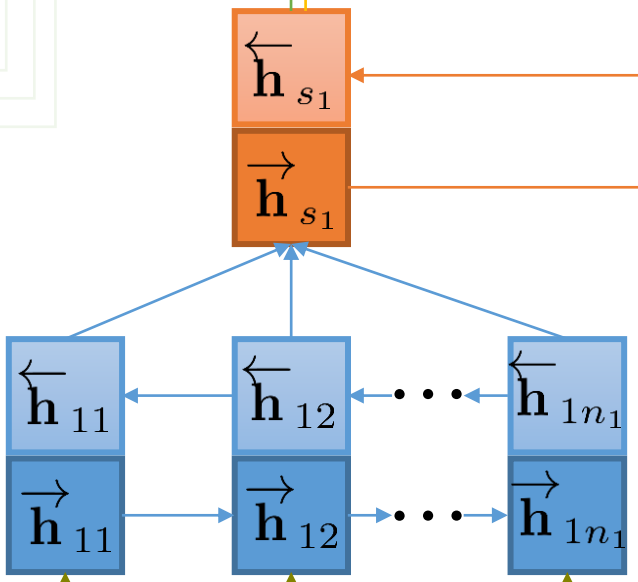
SummaRuNNer: Word-level Bi-directional GRU-RNN

Word-level two-layer bi-directional RNN

- Hidden nodes in the same sentence are connected
- Hidden nodes in different sentences are not connected



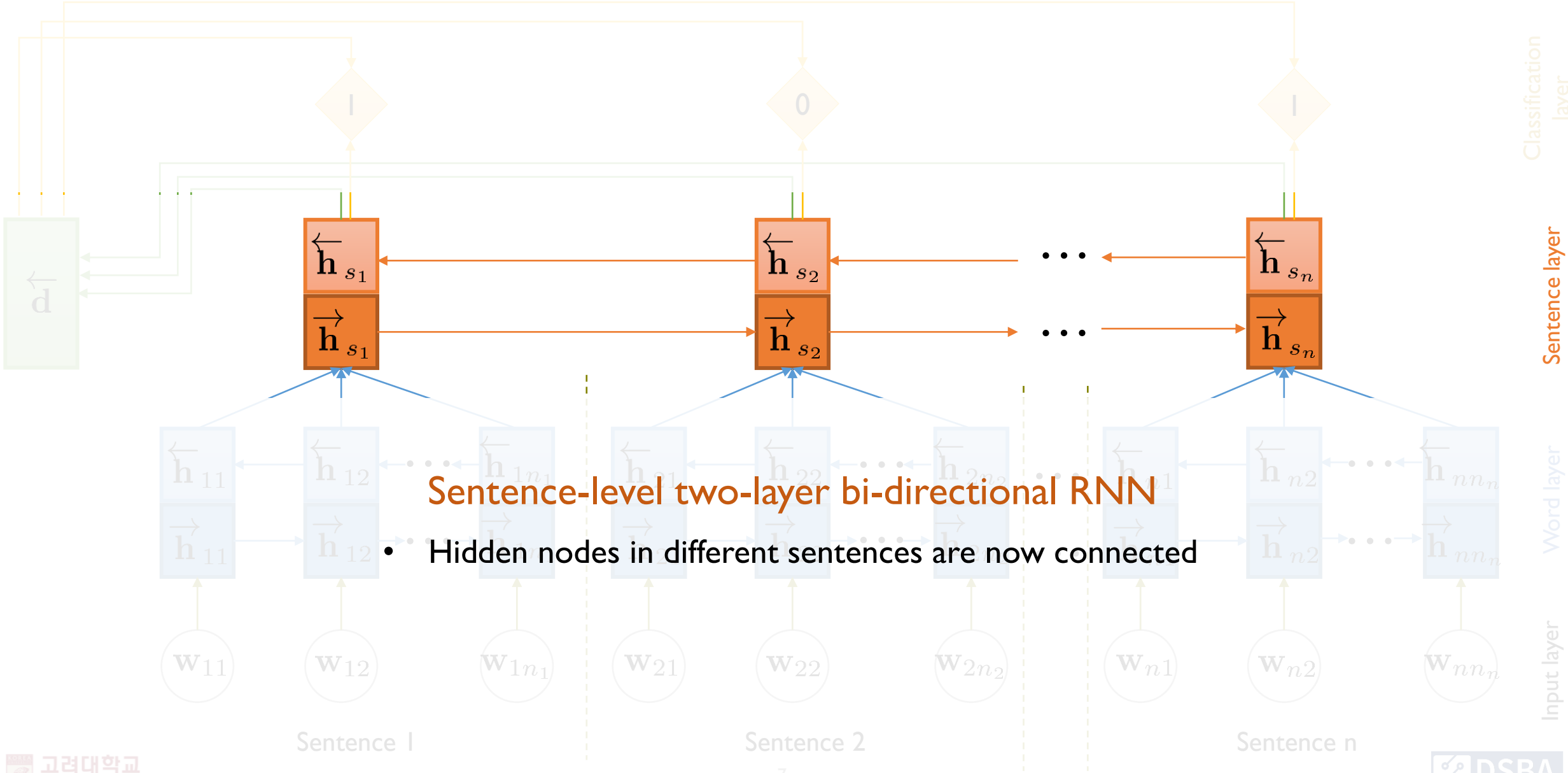
SummaRuNNer: Sentence Representation



Sentence vector representation:
average-pooled, concatenated hidden states

$$\text{Sentence Embedding} = \frac{1}{n_1}\sum_{i=1}^{n_1}\left[\overrightarrow{\mathbf{h}}_{1i}, \overleftarrow{\mathbf{h}}_{1i}\right]$$

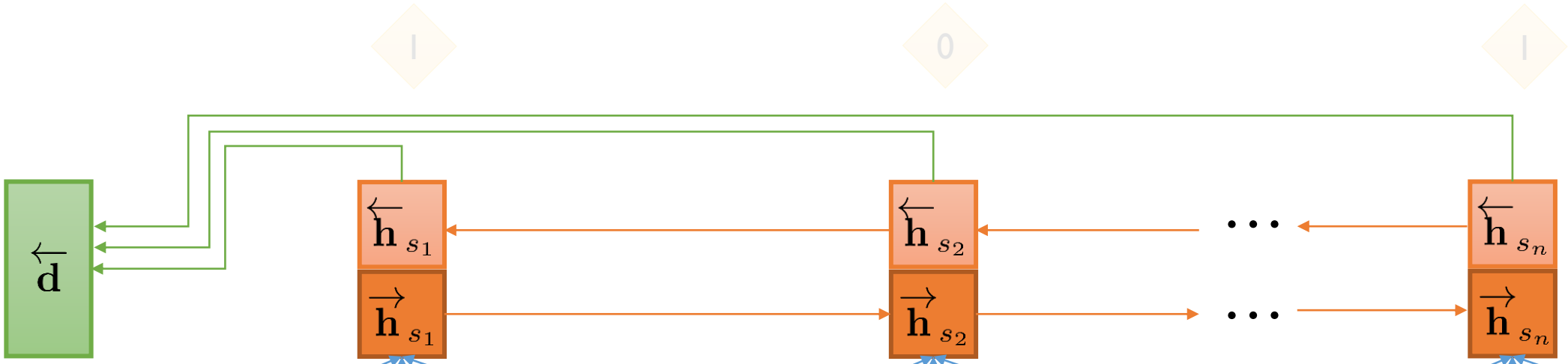
SummaRuNNer: Sentence-level Bi-directional GRU-RNN



Sentence-level two-layer bi-directional RNN

- Hidden nodes in different sentences are now connected

SummaRuNNer: Document Representation

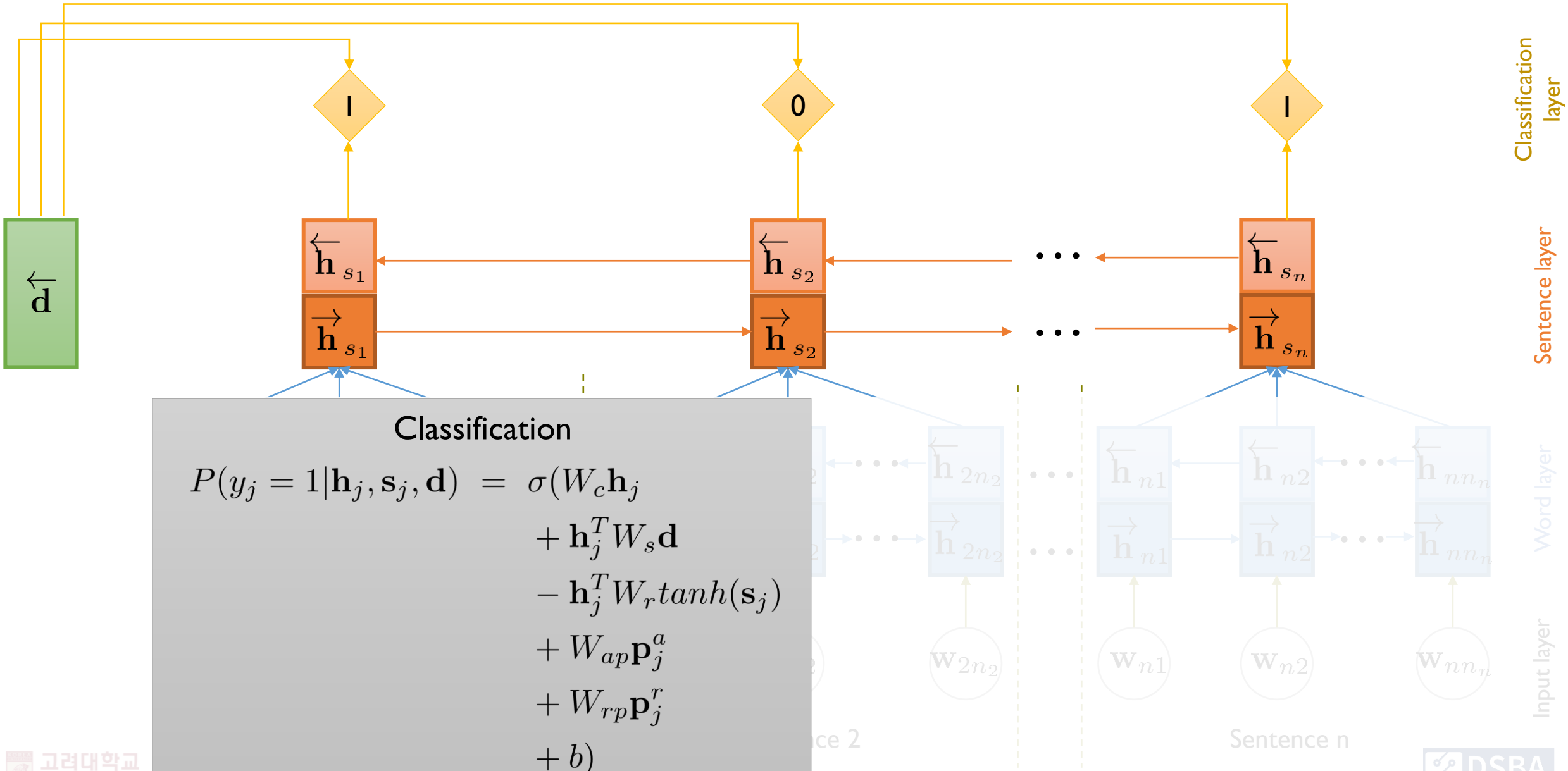


The representation of the entire document

- A non-linear transformation of the average pooling of the concatenated hidden states of the bi-directional sentence-level RNN

$$\mathbf{d} = tanh\left(W_d \frac{1}{n} \sum_{j=1}^{n} \left[\overrightarrow{\mathbf{h}}_{sj}, \overleftarrow{\mathbf{h}}_{sj}\right] + \mathbf{b}\right)$$

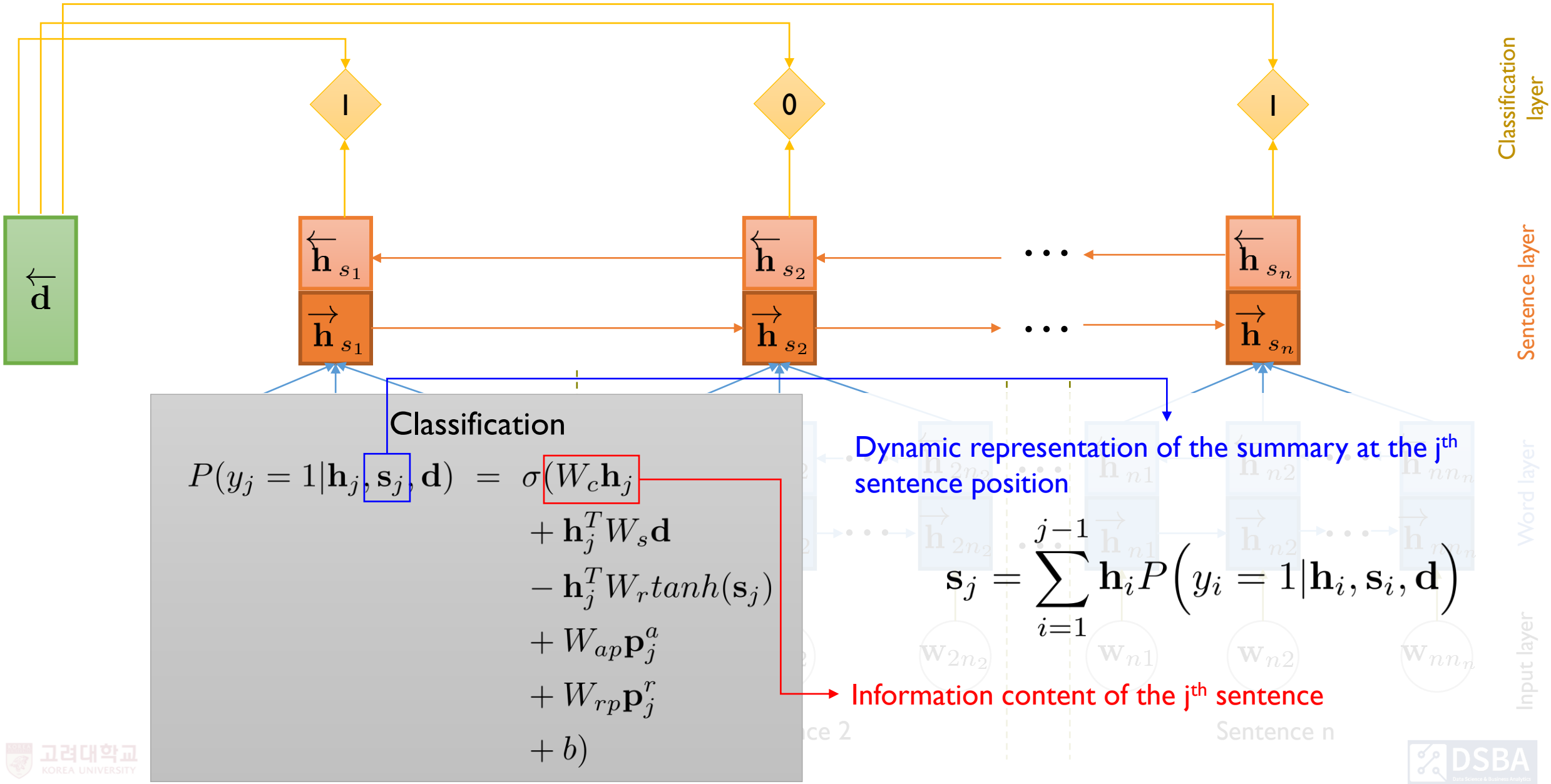
SummaRuNNer: Classification



Classification

$$P(y_j = 1|\mathbf{h}_j, \mathbf{s}_j, \mathbf{d}) = \sigma(W_c\mathbf{h}_j + \mathbf{h}_j^T W_s \mathbf{d} - \mathbf{h}_j^T W_r tanh(\mathbf{s}_j) + W_{ap}\mathbf{p}_j^a + W_{rp}\mathbf{p}_j^r + b)$$

SummaRuNNer: Classification



Classification

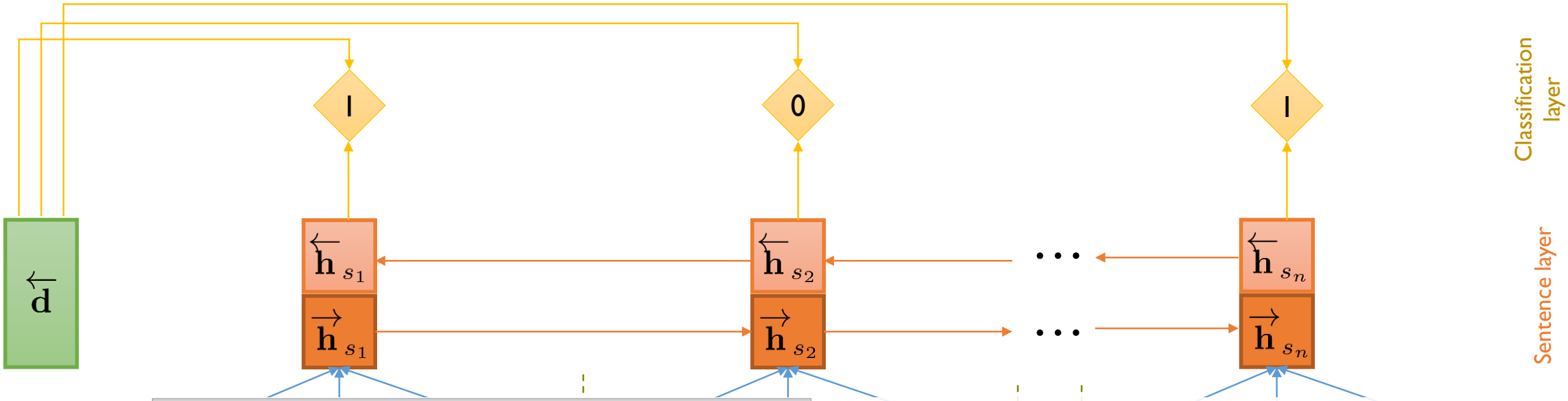
$$P(y_j = 1|\mathbf{h}_j, \mathbf{s}_j, \mathbf{d}) = \sigma(W_c\mathbf{h}_j + \mathbf{h}_j^T W_s \mathbf{d} - \mathbf{h}_j^T W_r tanh(\mathbf{s}_j) + W_{ap}\mathbf{p}_j^a + W_{rp}\mathbf{p}_j^r + b)$$

Dynamic representation of the summary at the j[th] sentence position

$$\mathbf{s}_j = \sum_{i=1}^{j-1} \mathbf{h}_i P\left(y_i = 1|\mathbf{h}_i, \mathbf{s}_i, \mathbf{d}\right)$$

Information content of the j[th] sentence

SummaRuNNer: Classification

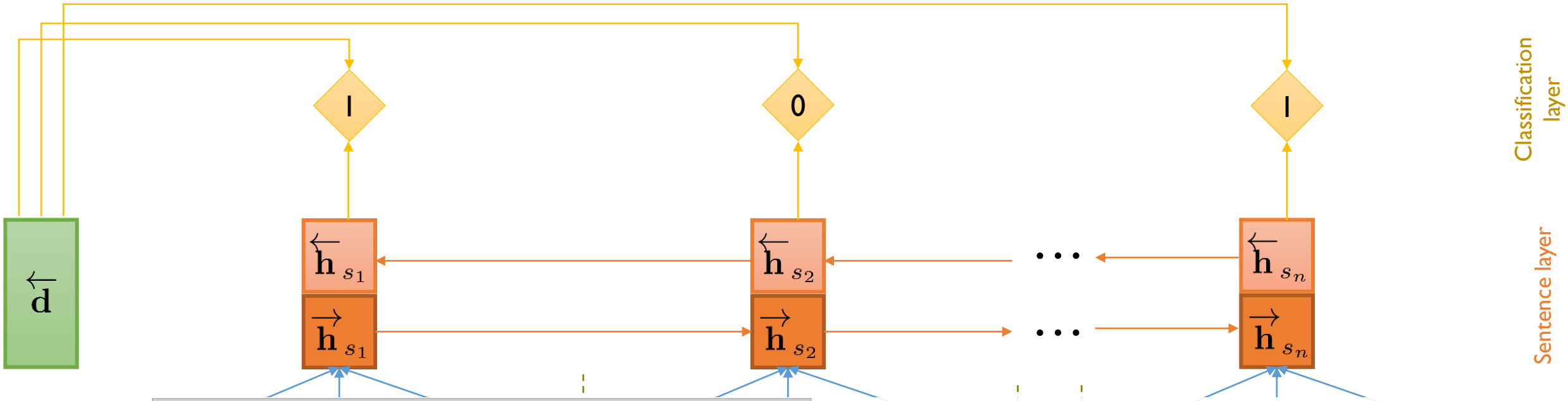


Classification

$$P(y_j = 1|\mathbf{h}_j, \mathbf{s}_j, \mathbf{d}) = \sigma(W_c\mathbf{h}_j + \mathbf{h}_j^T W_s \mathbf{d} - \mathbf{h}_j^T W_r tanh(\mathbf{s}_j) + W_{ap}\mathbf{p}_j^a + W_{rp}\mathbf{p}_j^r + b)$$

- $\mathbf{h}_j^T W_s \mathbf{d}$ → Salience of the sentence with respect to the document
- $\mathbf{h}_j^T W_r tanh(\mathbf{s}_j)$ → Redundancy of the sentence with respect to the current state of the summary
- $W_{ap}\mathbf{p}_j^a + W_{rp}\mathbf{p}_j^r$ → Absolute and relative positional embeddings

SummaRuNNer: Classification



Classification

$$P(y_j = 1|\mathbf{h}_j, \mathbf{s}_j, \mathbf{d}) = \sigma(W_c\mathbf{h}_j + \mathbf{h}_j^T W_s \mathbf{d} - \mathbf{h}_j^T W_r tanh(\mathbf{s}_j) + W_{ap}\mathbf{p}_j^a + W_{rp}\mathbf{p}_j^r + b)$$

```
# Parameters of Classification Layer
self.content = nn.Linear(2*H,1,bias=False)
self.salience = nn.Bilinear(2*H,2*H,1,bias=False)
self.novelty = nn.Bilinear(2*H,2*H,1,bias=False)
self.abs_pos = nn.Linear(P_D,1,bias=False)
self.rel_pos = nn.Linear(P_D,1,bias=False)
self.bias = nn.Parameter(torch.FloatTensor(1).uniform_(-0.1,0.1))
```

Sentence 2

Sentence n

SummaRuNNer: Training

- Objective function
  - ✓ Minimize the negative log-likelihood of the observed labels at training time

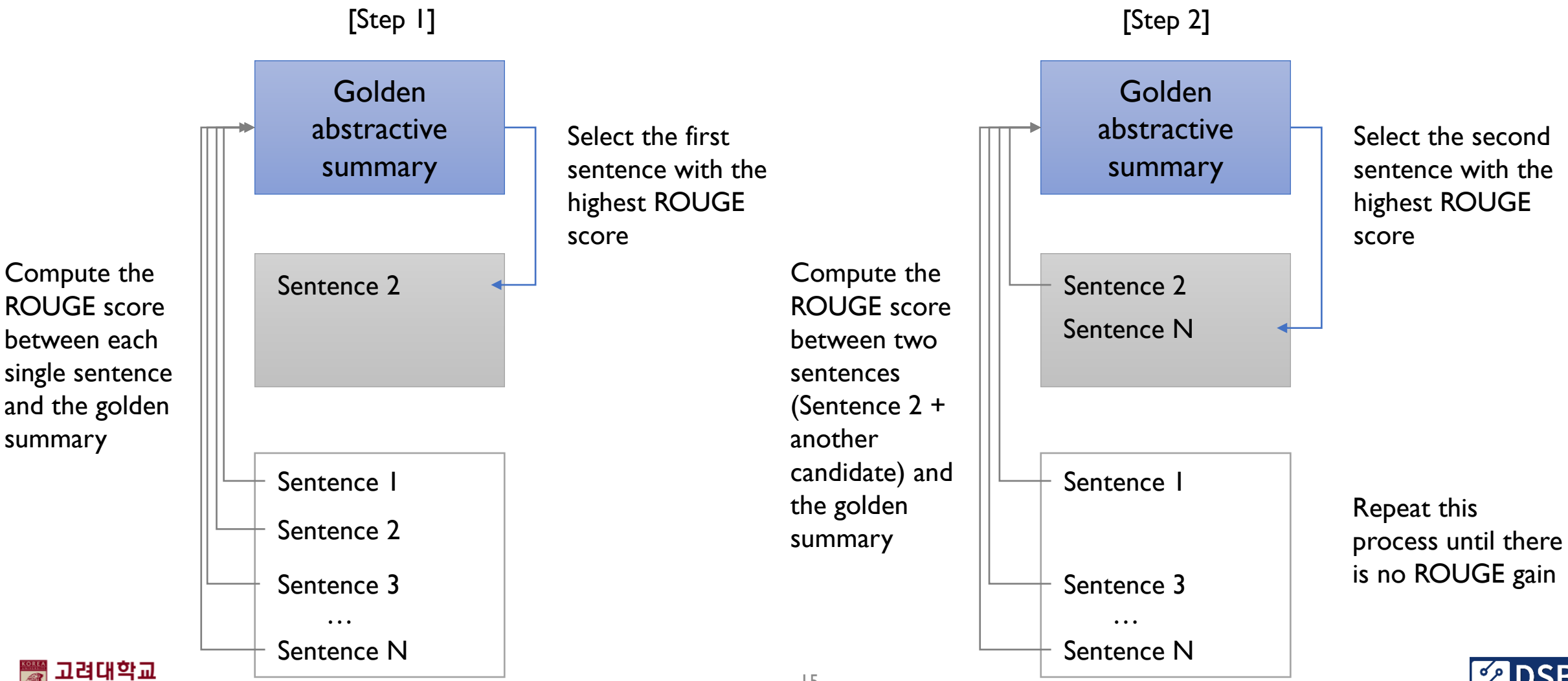
$$l(\mathbf{W}, \mathbf{b}) = -\sum_{d=1}^{N}\sum_{j=1}^{N_d}\Big(y_j^d \cdot \log P(y_j^d = 1|\mathbf{h}_j^d, \mathbf{s}_j^d, \mathbf{d}_d) + (1 - y_j^d) \cdot \log(1 - P(y_j^d = 1|\mathbf{h}_j^d, \mathbf{s}_j^d, \mathbf{d}_d))\Big)$$

SummaRuNNer: Training

- Extractive Training
  - ✓ One need ground truth in the form of sentence binary labels for each sentence representing their membership in the summary
  - ✓ Most summarization corpora only contain human written abstractive summaries as ground truth
  - ✓ Use an unsupervised approach to convert the abstractive summaries to extract labels
    - Selected sentences should maximize the ROUGE score with respect to gold summaries
  - ✓ Finding global optimal subset of sentences is computationally expensive, a greedy approach is used

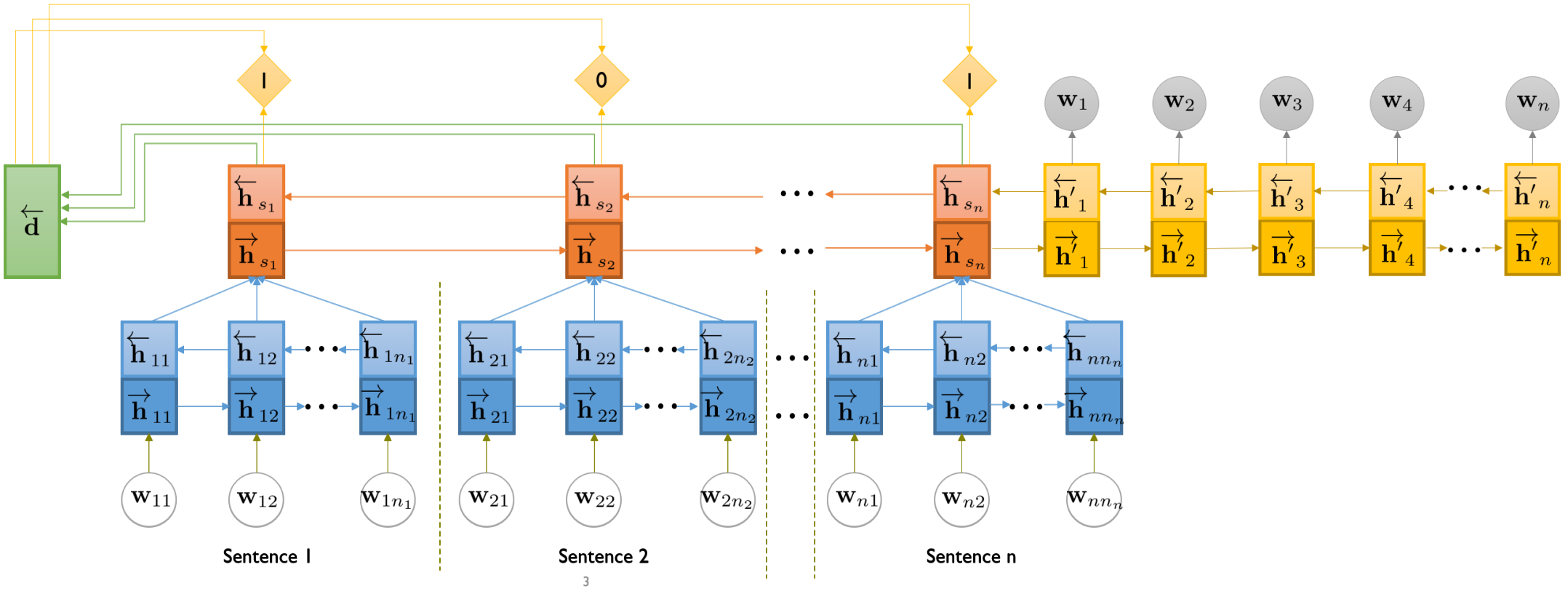
SummaRuNNer: Training

- Extractive Training: Label annotation (illustrative example)



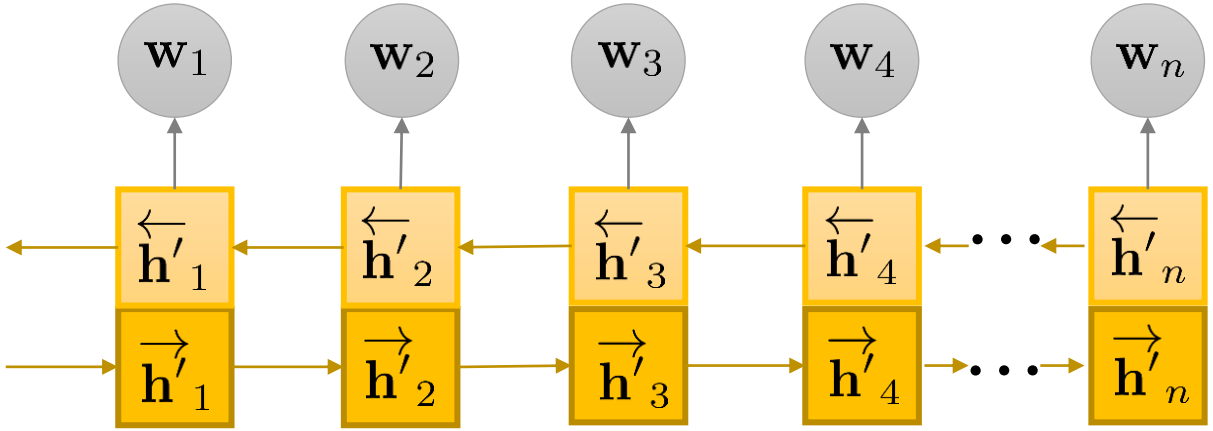
SummaRuNNer: Training

- Abstractive Training
  - ✓ Use an RNN decoder that models the generation of abstractive summaries at training time only



SummaRuNNer: Training

- Abstractive Training



✓ GRU in the decoder

$$\mathbf{u}_k = \sigma(\mathbf{W'}_{ux}\mathbf{x}_k + \mathbf{W'}_{uh}\mathbf{h}_{k-1} + \mathbf{W'}_{uc}\mathbf{s}_{-1} + \mathbf{b'}_u)$$

$$\mathbf{r}_j = \sigma(\mathbf{W'}_{rx}\mathbf{x}_k + \mathbf{W'}_{rh}\mathbf{h}_{k-1} + \mathbf{W'}_{rc}\mathbf{s}_{-1} + \mathbf{b'}_r)$$

$$\mathbf{h}'_j = tanh(\mathbf{W'}_{hx}\mathbf{x}_k + \mathbf{W'}_{hh}(\mathbf{r}_k \odot \mathbf{h}_{k-1}) + \mathbf{W'}_{hc}\mathbf{s}_{-1} + \mathbf{b'}_h)$$

- $\mathbf{s}_{-1}$: the summary representation as computed at the las sentence of the sentence-level bidirectional RNN of SummaRuNNer

SummaRuNNer: Training

- Abstractive Training
  - ✓ The decoder is equipped with a softmax layer to emit a word at each time-step.
  - ✓ The emission at each time-step is determined by a feed-forward layer *f* followed by a softmax layer that assign p_k.

$$\mathbf{f}_k = \tanh(\mathbf{W'}_{fh}\mathbf{h}_k + \mathbf{W'}_{fx}\mathbf{x}_k + \mathbf{W'}_{fc}\mathbf{s}_{-1} + \mathbf{b'}_f)$$

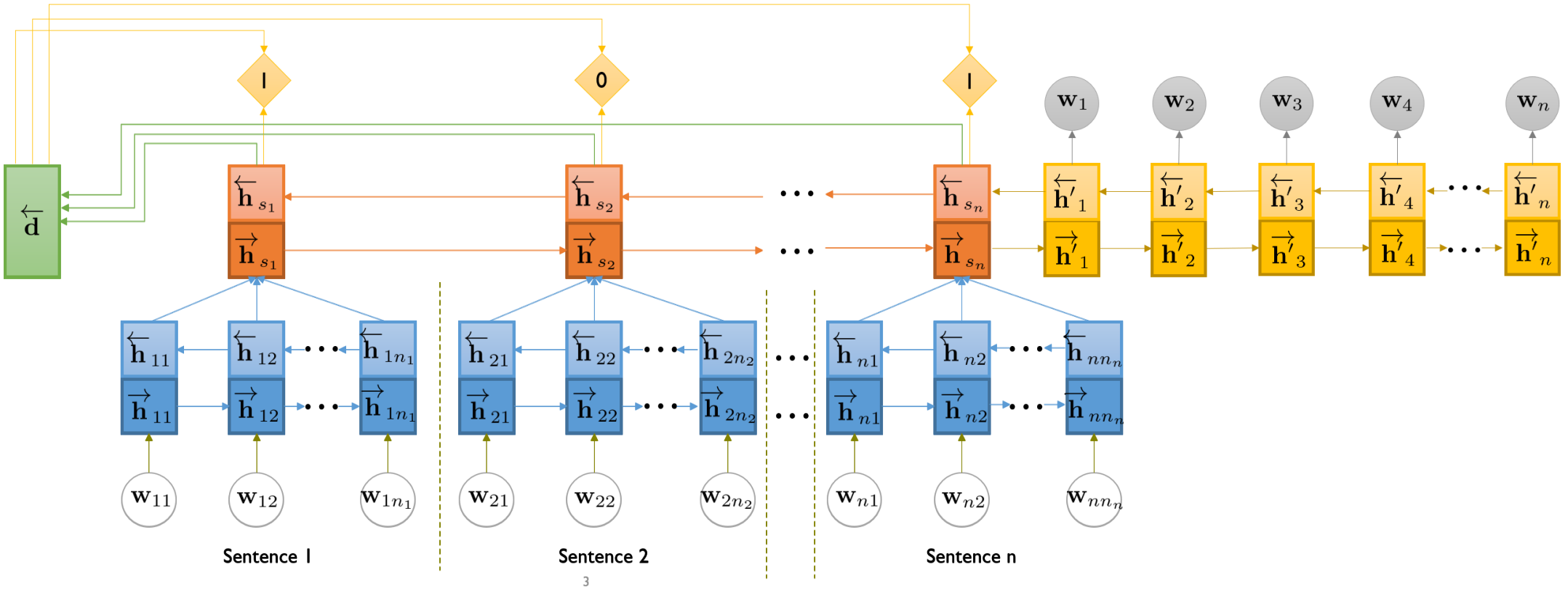
$$\mathbf{P_v}(\mathbf{w})_k = \text{softmax}(\mathbf{W'}_v\mathbf{f}_k + \mathbf{b'}_v)$$

  - ✓ Instead of optimizing the log-likelihood pf the extractive ground truth, the negative log-likelihood of the words in the reference summary is minimized:

$$l(\mathbf{W}, \mathbf{b}, \mathbf{W'}, \mathbf{b'}) = -\sum_{k=1}^{N_s} \log(\mathbf{P_v}(w_k))$$

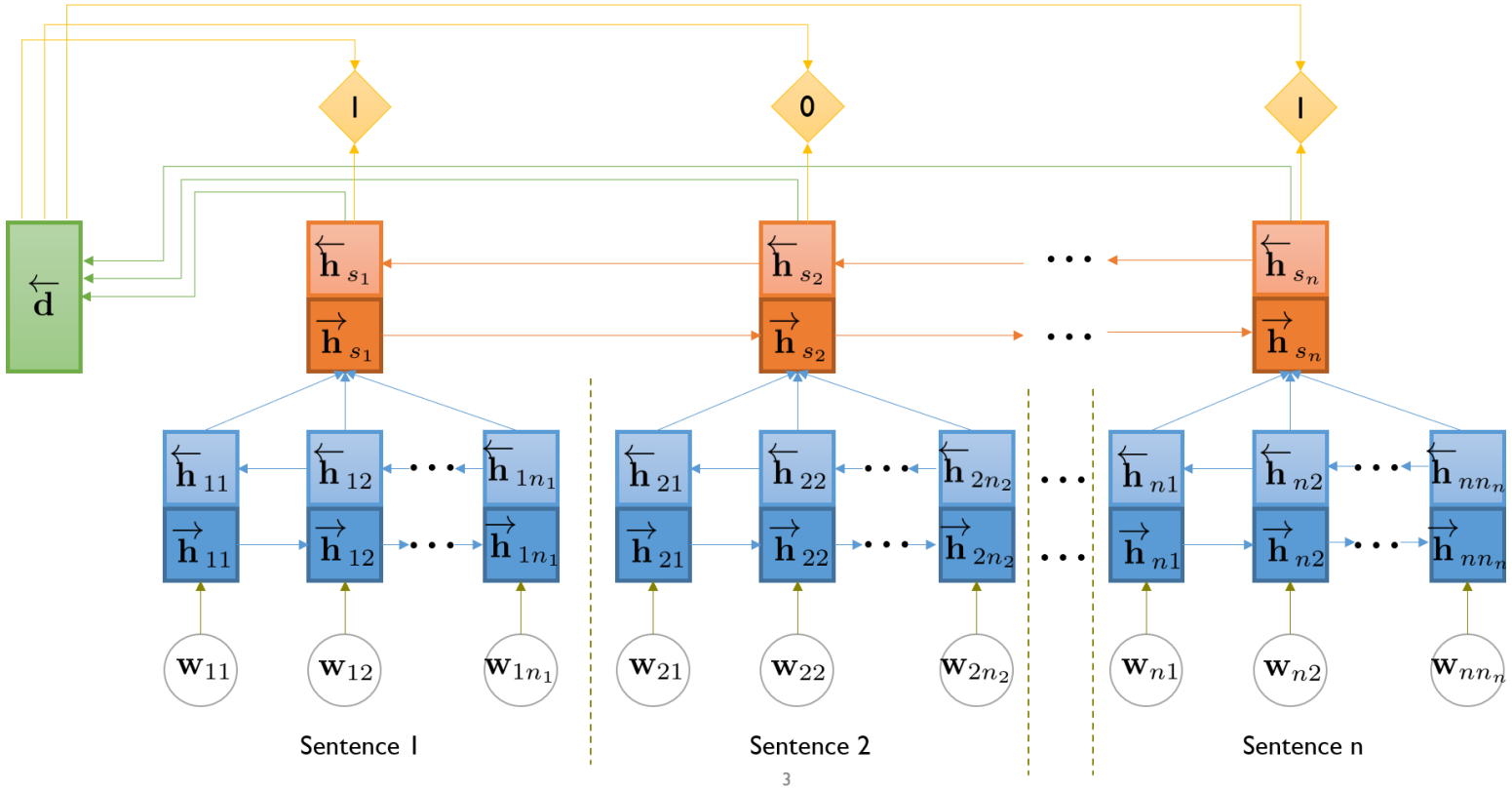
SummaRuNNer: Training

- Test after Abstractive Training
  - ✓ Uncouple the decoder from SummarRuNNer and emit only the sentence-level extractive probabilities $p(y_j)$



SummaRuNNer: Training

- Test after Abstractive Training
  - ✓ Uncouple the decoder from SummarRuNNer and emit only the sentence-level extractive probabilities p(y_j)
  - ✓ Q: How to optimize the weights for classification?



$$P(y_j = 1|\mathbf{h}_j, \mathbf{s}_j, \mathbf{d}) = \sigma(W_c\mathbf{h}_j + \mathbf{h}_j^T W_s \mathbf{d} - \mathbf{h}_j^T W_r tanh(\mathbf{s}_j) + W_{ap}\mathbf{p}_j^a + W_{rp}\mathbf{p}_j^r + b)$$

SummaRuNNer: Results

- Quantitative performances

	Rouge-1	Rouge-2	Rouge-L
Lead-3	21.9	7.2	11.6
LReg(500)	18.5	6.9	10.2
Cheng *et al* '16	22.7	8.5	12.5
SummaRuNNer-abs	23.8	9.6	13.3
SummaRuNNer	**26.2**±0.4*	**10.8**±0.3*	**14.4**±0.3*

Table 1: Performance of various models on the **entire Daily Mail test set** using the **limited length recall** variants of Rouge with respect to the abstractive ground truth at **75 bytes**. Entries with asterisk are statistically significant using 95% confidence interval with respect to the nearest model, as estimated by the Rouge script.

	Rouge-1	Rouge-2	Rouge-L
Lead-3	40.5	14.9	32.6
Cheng *et al* '16	**42.2**	**17.3**	**34.8***
SummaRuNNer-abs	40.4	15.5	32.0
SummaRuNNer	42.0 ±0.2	16.9 ±0.4	34.1 ±0.3

Table 2: Performance of various models on the **entire Daily Mail test set** using the **limited length recall** variants of Rouge at **275 bytes**. SummaRuNNer is statistically indistinguishable from the model of Cheng et al, '16 at 95% C.I. on Rouge-1 and Rouge-2.

	Rouge-1	Rouge-2	Rouge-L
Lead-3	39.2	15.7	**35.5**
(Nallapati et al. 2016)	35.4	13.3	32.6
SummaRuNNer-abs	37.5	14.5	33.4
SummaRuNNer	**39.6**±0.2*	**16.2**±0.2*	35.3±0.2

Table 3: Performance comparison of abstractive and extractive models on the entire CNN Daily Mail test set using **full-length F1** variants of Rouge. SummaRuNNer is able to significantly outperform the abstractive state-of-the-art as well as the Lead-3 baseline (on Rouge-1 and Rouge-2).

SummaRuNNer: Results

- Qualitative Analysis

Gold Summary: Redpath has ended his eight-year association with Sale Sharks. Redpath spent five years as a player and three as a coach at sale. He has thanked the owners, coaches and players for their support.	Salience	Content	Novelty	Position	Prob.
Bryan Redpath has left his coaching role at Sale Sharks with immediate effect.	0.1	0.1	0.9	0.1	0.3
The 43 - year - old Scot ends an eight-year association with the Aviva Premiership side, having spent five years with them as a player and three as a coach.	0.9	0.6	0.9	0.9	0.7
Redpath returned to Sale in June 2012 as director of rugby after starting a coaching career at Gloucester and progressing to the top job at Kingsholm .	0.8	0.5	0.5	0.9	0.6
Redpath spent five years with Sale Sharks as a player and a further three as a coach but with Sale Sharks struggling four months into Redpath's tenure, he was removed from the director of rugby role at the Salford-based side and has since been operating as head coach .	0.8	0.9	0.7	0.8	**0.9**
'I would like to thank the owners, coaches, players and staff for all their help and support since I returned to the club in 2012.	0.4	0.1	0.1	0.7	0.2
Also to the supporters who have been great with me both as a player and as a coach,' Redpath said.	0.6	0.0	0.2	0.3	0.2

NeuSum: Overview

- Two subtasks of text summarization
  - ✓ Sentence scoring aims to assign an importance score to each sentence.
  - ✓ Sentence selection adopts a particular strategy to choose content sentence by sentence.
- NeuSum
  - ✓ An end-to-end neural network model that learns to identify the relative importance of sentences
    - Relative importance: the gain over previously selected sentences.
  - ✓ Consists of two parts
    - Document encoder: a hierarchical architecture, which suits the compositionality of documents.
    - Sentence extractor: built with RNN providing two main functions (remembering the partial output summary by feeding the selected sentence into it, providing a sentence extraction state)

NeuSum: Problem Formulation

- Given a document $\mathcal{D} = (S_1, S_2, ..., S_L)$ containing L sentences, an extractive summarization system should select a subset of D to form the output summary $\mathcal{S} = \{\hat{S}_i | \hat{S}_i \in \mathcal{D}\}$
- During the training phase, the reference summary S* and the score of an output summary S under a given evaluation function $r(\mathcal{S}|\mathcal{S}^*)$ are available.
- Goal: to learn a scoring function $f(\mathcal{S})$ which can be used to find the best summary during testing:

$$\arg\max_{\mathcal{S}} f(\mathcal{S})$$

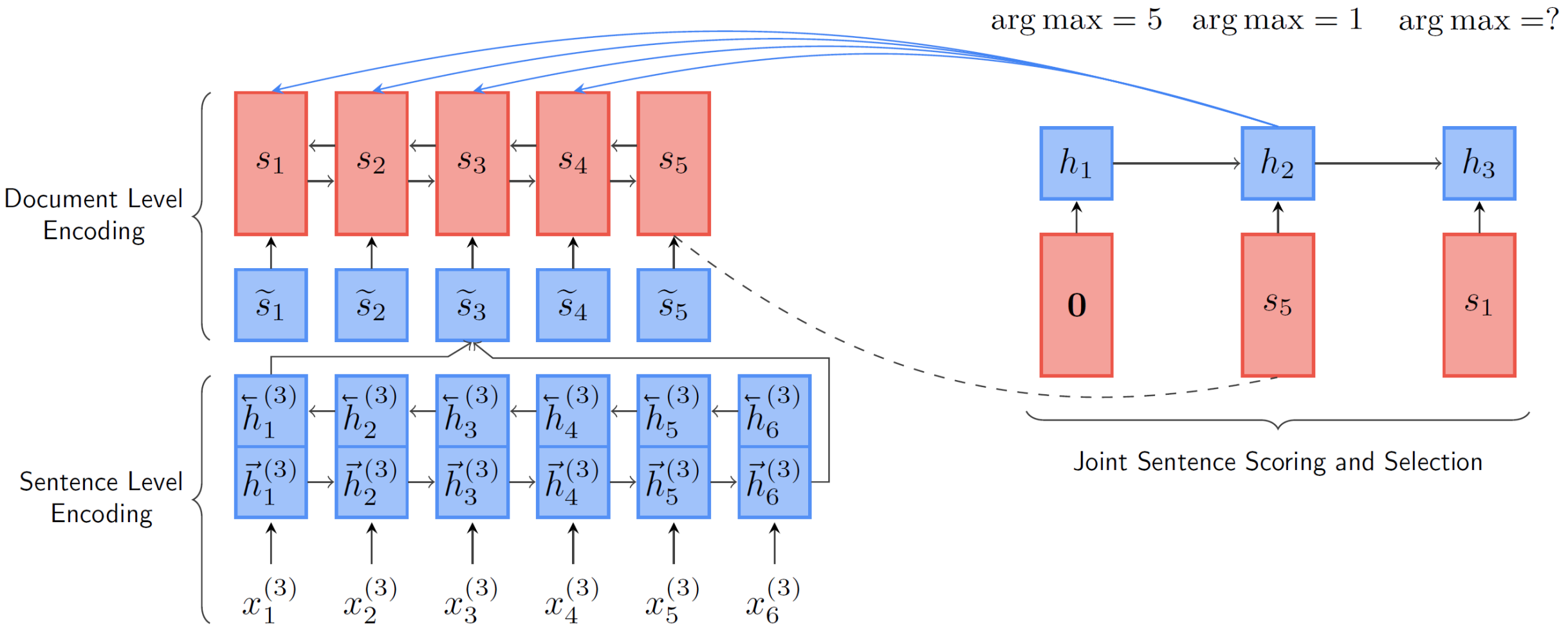
$$s.t. \quad \mathcal{S} = \{\hat{S}_i | \hat{s}_i \in \mathcal{D}\}, \quad |\mathcal{S}| \leq l.$$

✓ ROUGE F1 is used as the evaluation function $r(\cdot)$ in NeuSum

✓ NeuSum is trained to learn a scoring function $g(\cdot)$ of the ROUGE F1 gain:

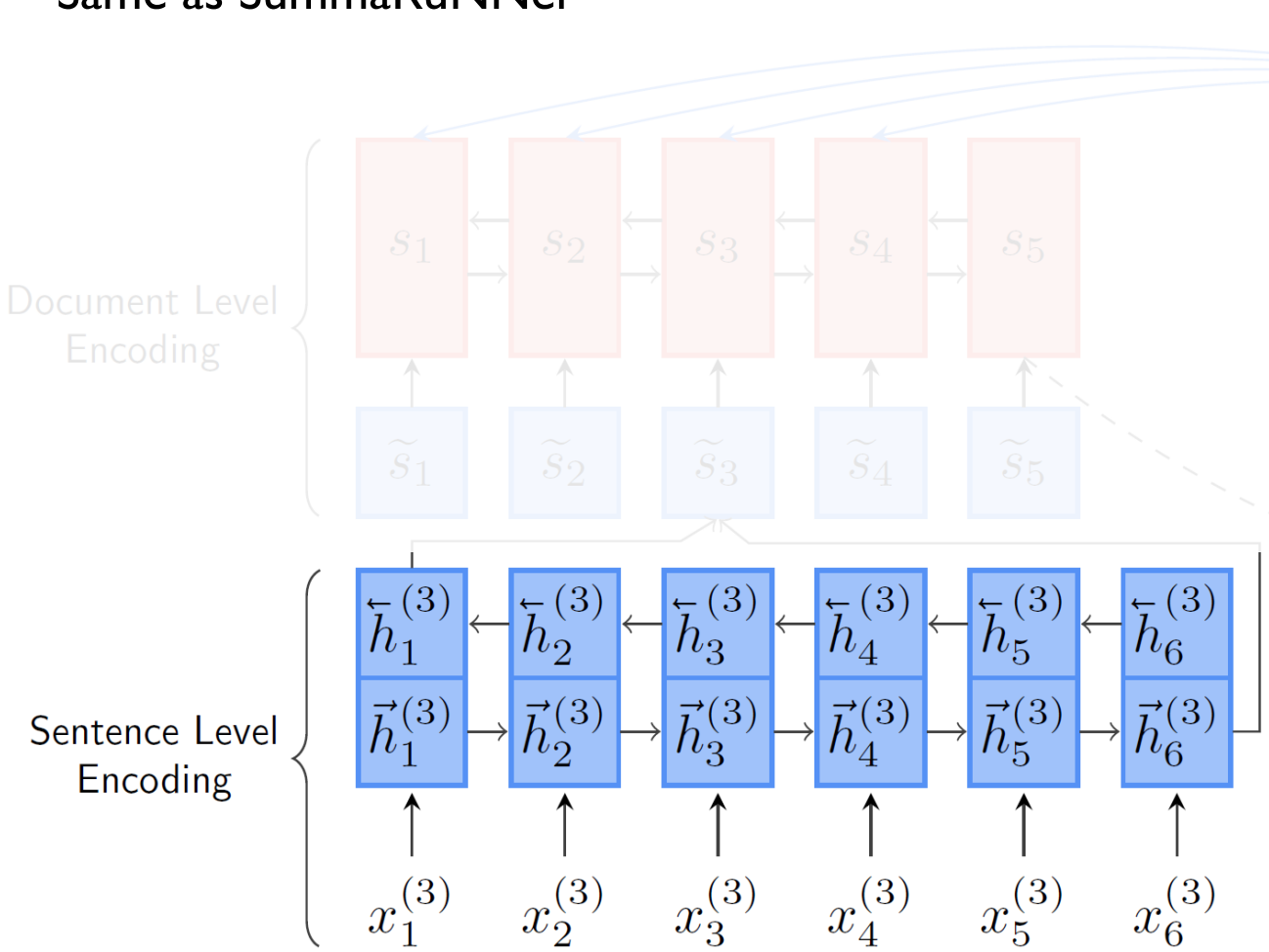
$$g(S_t | \mathbb{S}_{t-1}) = r(\mathbb{S}_{t-1} \cup \{S_t\}) - r(\mathbb{S}_{t-1})$$

NeuSum: Structure



NeuSum: Word-Level Bi-directional GRU

- Same as SummaRuNNer



$$z_i = \sigma(\mathbf{W}_z[x_i, h_{i-1}])$$
$$r_i = \sigma(\mathbf{W}_r[x_i, h_{i-1}])$$
$$\widetilde{h}_i = \tanh(\mathbf{W}_h[x_i, r_i \odot h_{i-1}])$$
$$h_i = (1 - z_i) \odot h_{i-1} + z_i \odot \widetilde{h}_i$$

NeuSum: Sentence Representation

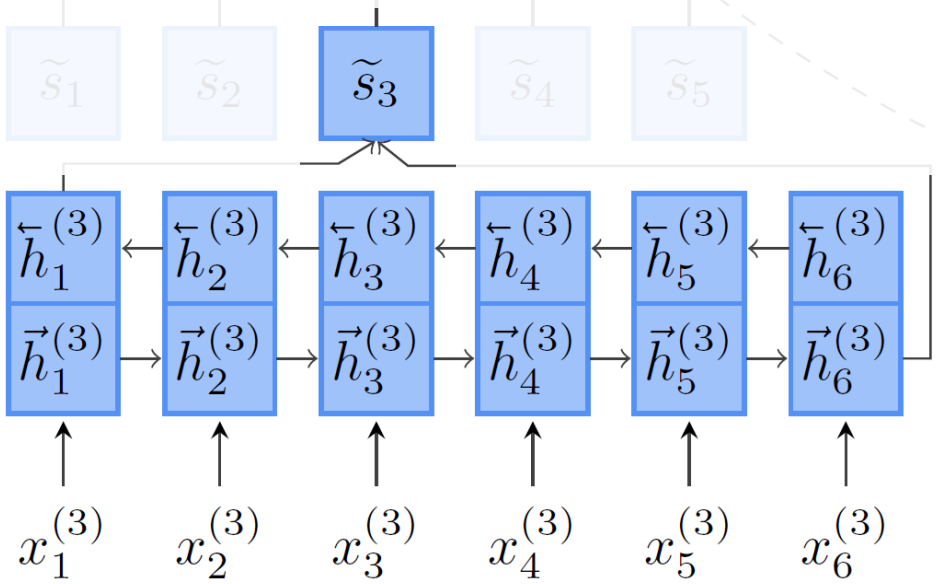
[SummarRuNNer]

$$\text{Sentence Embedding} = \frac{1}{n_1}\sum_{i=1}^{n_1}\left[\overrightarrow{\mathbf{h}}_{1i}, \overleftarrow{\mathbf{h}}_{1i}\right]$$

$$\widetilde{s}_j = \begin{bmatrix} \overleftarrow{h}_1^{(j)} \\ \overrightarrow{h}_{n_j}^{(j)} \end{bmatrix}$$

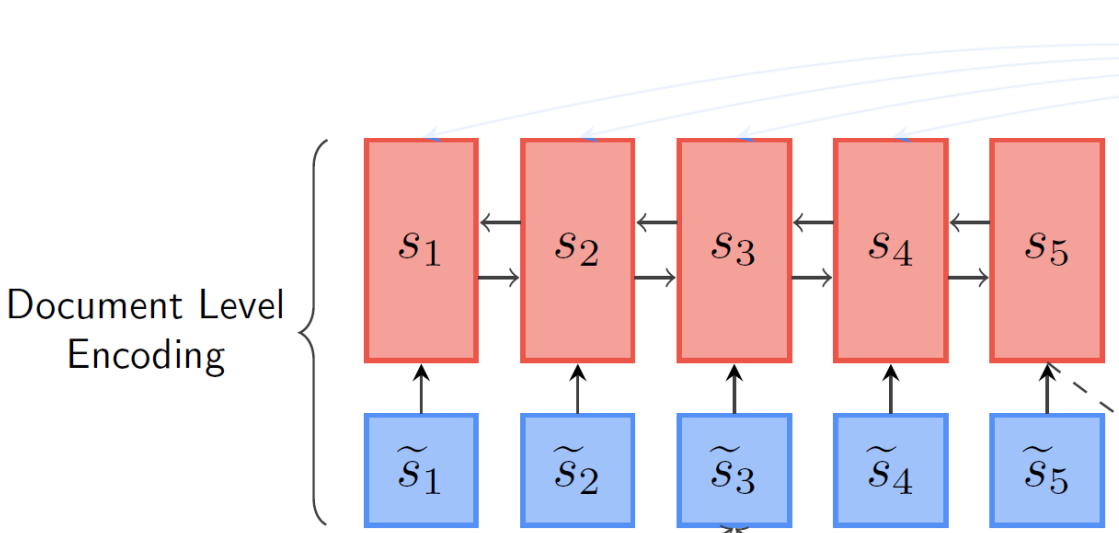
Document Level Encoding

Sentence Level Encoding



$$z_i = \sigma(\mathbf{W}_z[x_i, h_{i-1}])$$
$$r_i = \sigma(\mathbf{W}_r[x_i, h_{i-1}])$$
$$\widetilde{h}_i = \tanh(\mathbf{W}_h[x_i, r_i \odot h_{i-1}])$$
$$h_i = (1 - z_i) \odot h_{i-1} + z_i \odot \widetilde{h}_i$$

NeuSum: Structure

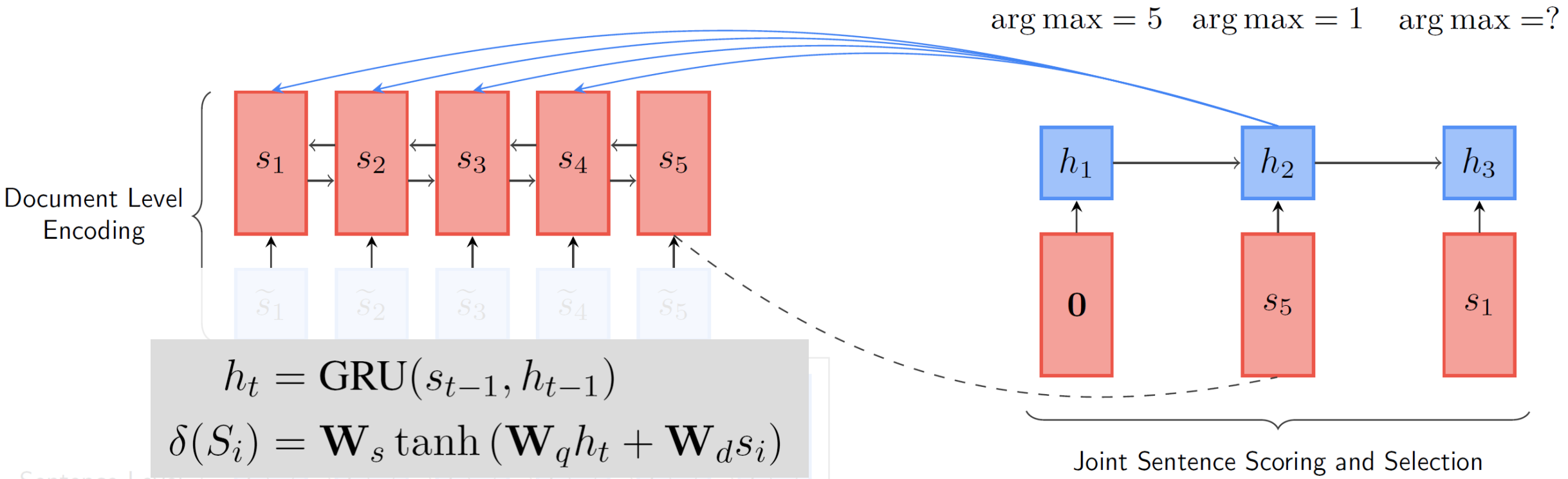


Sentence-level two-layer bi-directional RNN

- Hidden nodes in different sentences are now connected

NeuSum: Structure

[Joint Sentence Scoring and Selection]



$$h_t = \mathbf{GRU}(s_{t-1}, h_{t-1})$$
$$\delta(S_i) = \mathbf{W}_s \tanh\left(\mathbf{W}_q h_t + \mathbf{W}_d s_i\right)$$

$$h_0 = \tanh\left(\mathbf{W}_m \overleftarrow{s}_1 + b_m\right)$$
$$S_0 = \varnothing$$
$$s_0 = \mathbf{0}$$

$$\hat{S}_t = \underset{S_i \in \mathcal{D}}{\arg\max}\, \delta(S_i)$$

NeuSum: Training

- Objective Function
  - ✓ Kullback-Leibler (KL) divergence of the model prediction P and the labeled training data distribution Q is optimized
  - ✓ The predicted sentence score $\delta(S_i)$ is normalized with softmax function to get the model prediction distribution P:

$$P(\hat{S}_t = S_i) = \frac{\exp\left(\delta(S_i)\right)}{\sum_{k=1}^{L} \exp\left(\delta(S_k)\right)}$$

  - ✓ During training, the model is expected to learn the relative ROUGE F1 gain at time step t with previously selected sentences $\mathbb{S}_{t-1}$
  - ✓ Min-Max Normalization is used to compensate for the possible negative F1 gain

$$g(S_i) = r(\mathbb{S}_{t-1} \cup \{S_i\}) - r(\mathbb{S}_{t-1})$$

$$\widetilde{g}(S_i) = \frac{g(S_i) - \min\left(g(S)\right)}{\max\left(g(S)\right) - \min\left(g(S)\right)}$$

NeuSum: Training

- Objective Function (cont')
  - ✓ A softmax operation with temperature τ to produce the labeled data distribution Q as the training target
    - The temperature τ is applied as a smoothing factor to produce a smoothed label distribution Q:

$$Q(S_i) = \frac{\exp\left(\tau \widetilde{g}(S_i)\right)}{\sum_{k=1}^{L} \exp\left(\tau \widetilde{g}(S_k)\right)}$$

  - ✓ The KL loss function J becomes

$$J = D_{KL}(P \parallel Q)$$

NeuSum: Experimental Results

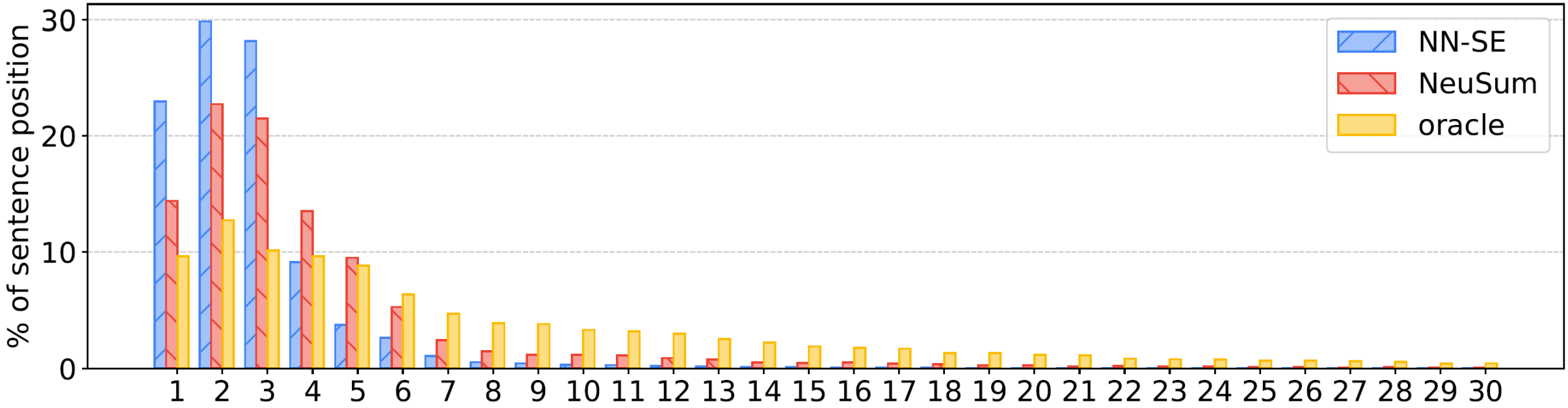
- Experimental results: ROUGE scores

CNN/Daily Mail	Training	Dev	Test
#(Document)	287,227	13,368	11,490
#(Ref / Document)	1	1	1
Doc Len (Sentence)	31.58	26.72	27.05
Doc Len (Word)	791.36	769.26	778.24
Ref Len (Sentence)	3.79	4.11	3.88
Ref Len (Word)	55.17	61.43	58.31

Models	ROUGE-1	ROUGE-2	ROUGE-L
LEAD3	40.24^{-}	17.70^{-}	36.45^{-}
TEXTRANK	40.20^{-}	17.56^{-}	36.44^{-}
CRSUM	40.52^{-}	18.08^{-}	36.81^{-}
NN-SE	41.13^{-}	18.59^{-}	37.40^{-}
PGN‡	39.53^{-}	17.28^{-}	36.38^{-}
LEAD3‡ *	39.2	15.7	35.5
SUMMARUNNER‡ *	39.6	16.2	35.3
NEUSUM	**41.59**	**19.01**	**37.98**

NeuSum: Experimental Results

- Experimental results: position distribution of selected sentences




ANY
questions?