Image Captioning with Semantic Attention

2020.07.03

박진혁

Data Mining
Quality Analytics

목차

1. Introduction

❖ 주제 선정 이유

➢ 폐 이미지에 해당하는 질병에 대한 문장 생성

<Input>

CNN + RNN

<Output>

in comparison with the study of DATE, the monitoring and support devices are in essentially unchanged position. there is again mild enlargement of the cardiac silhouette with pulmonary edema and bilateral layering pleural effusions.

Image

Sentence

1. Introduction

❖ 컴퓨터 비전(Computer Vision) 이란?

- 지도학습의 한 분야로 사진/동영상에 대한 정답(Label)이 존재
- 사진/동영상의 의미있는 정보를 추출하여 분석하는 연구 분야
- 연구 분야에는 이미지 분류 및 위치파악, 객체 탐지, 이미지 분할 등이 존재

Image Classification

Iron Man

Image Localization

Iron Man

1. Introduction

❖ 컴퓨터 비전(Computer Vision) 이란?

- 지도학습의 한 분야로 사진/동영상에 대한 정답(Label)이 존재
- 사진/동영상의 의미있는 정보를 추출하여 분석하는 연구 분야
- 연구 분야에는 이미지 분류 및 위치파악, 객체 탐지, 이미지 분할 등이 존재

Object Detection

Thor, Captain America, Car

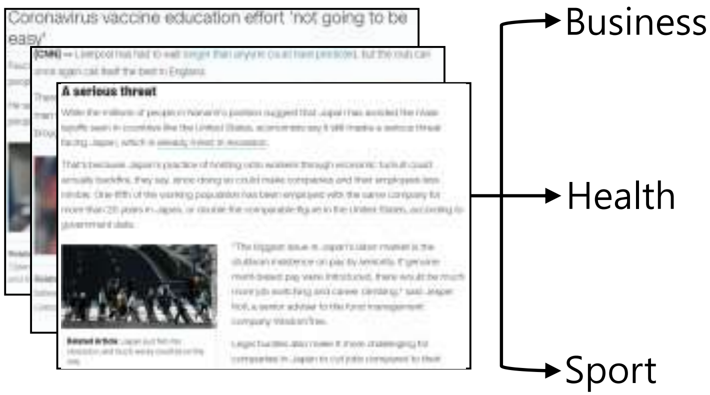
Image Segmentation

Thor, Captain America, Car

1. Introduction

❖ 자연어 처리(NLP) 란?

- 기계가 자연어를 이해하고 해석하여 처리하는 연구 분야
- 연구 분야에는 텍스트 분류, 감성 분석, 텍스트 요약 등이 존재



2. Image Captioning

❖ 이미지 캡셔닝(Image Captioning) 이란?

- 컴퓨터 비전과 자연어 처리를 연결하는 연구 분야
- 이미지를 설명하는 문장을 생성하는 알고리즘
- CNN과 RNN이 결합된 구조

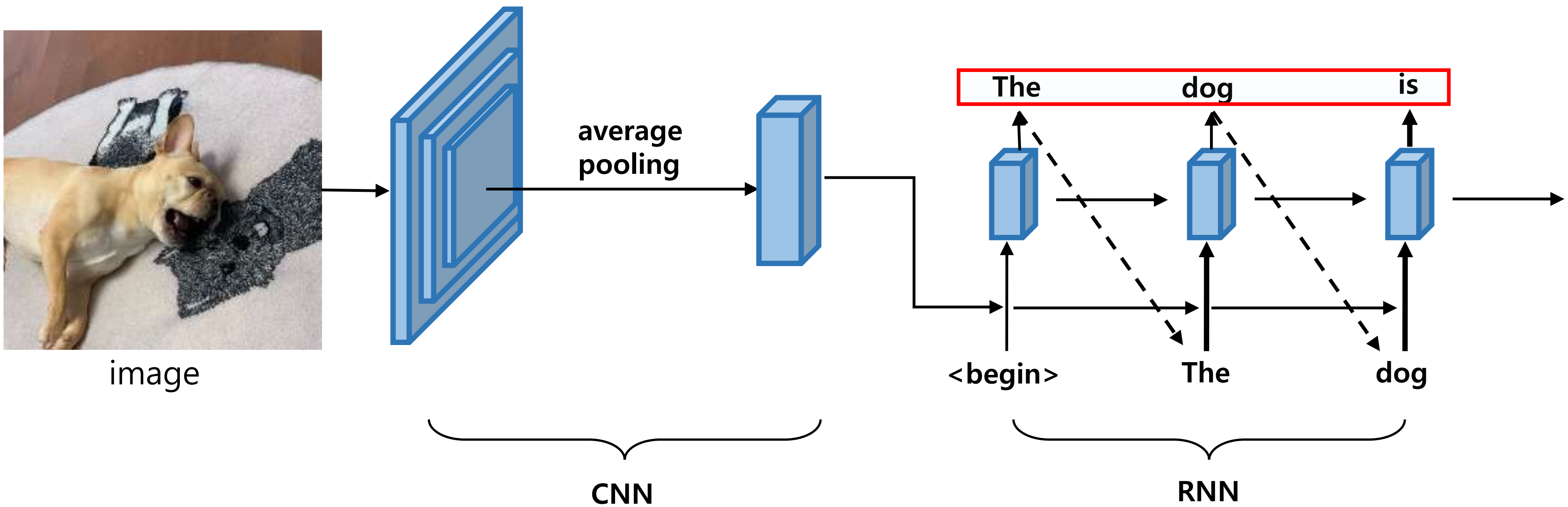
The dog is yawning.

The man with the umbrella
is walking down the street.

2. Image Captioning

❖ 이미지 캡셔닝 일반적인 구조

- Input : 이미지
- Output : 문장



3. Image Captioning with Semantic Attention

❖ Image Captioning with Semantic Attention

- 2016년 **C**omputer **V**ision and **P**attern **R**ecognition(**CVPR**)에서 발표된 논문
- 2020년 6월 30일 기준으로 **868**회 인용

Image Captioning with Semantic Attention

Quanzeng You[1], Hailin Jin[2], Zhaowen Wang[2], Chen Fang[2], and Jiebo Luo[1]

[1]Department of Computer Science, University of Rochester, Rochester NY 14627, USA
[2]Adobe Research, 345 Park Ave, San Jose CA 95110, USA

Abstract

Automatically generating a natural language description of an image has attracted interests recently both because of its importance in practical applications and because it connects two major artificial intelligence fields: computer vision and natural language processing. Existing approaches are either top-down, which start from a gist of an image and convert it into words, or bottom-up, which come up with words describing various aspects of an image and then combine them. In this paper, we propose a new algorithm that combines both approaches through a model of semantic attention. Our algorithm learns to selectively attend to semantic concept proposals and fuse them into hidden states and outputs of recurrent neural networks. The selection and fusion form a feedback connecting the top-down and bottom-up computation. We evaluate our algorithm on two public benchmarks: Microsoft COCO and

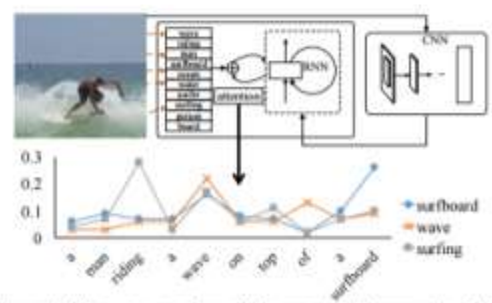


Figure 1. **Top**: an overview of the proposed framework. Given an image, we use a convolutional neural network to extract a top-down visual feature and at the same time detect visual concepts (regions, objects, attributes, etc.). We employ a semantic attention model to combine the visual feature with visual concepts in a recurrent neural network that generates the image caption. **Bottom**: We show the changes of the attention weights for several candidate concepts with respect to the recurrent neural network iterations.

3. Image Captioning with Semantic Attention

❖ Image Captioning with Semantic Attention

- Top-down approach, Bottom-up approach로 구성
- Top-down approach : 이미지의 전체적인 특징을 확인
- Bottom-up approach : 이미지의 자세한 부분을 확인

<Top-down>

CNN

<Bottom-up>

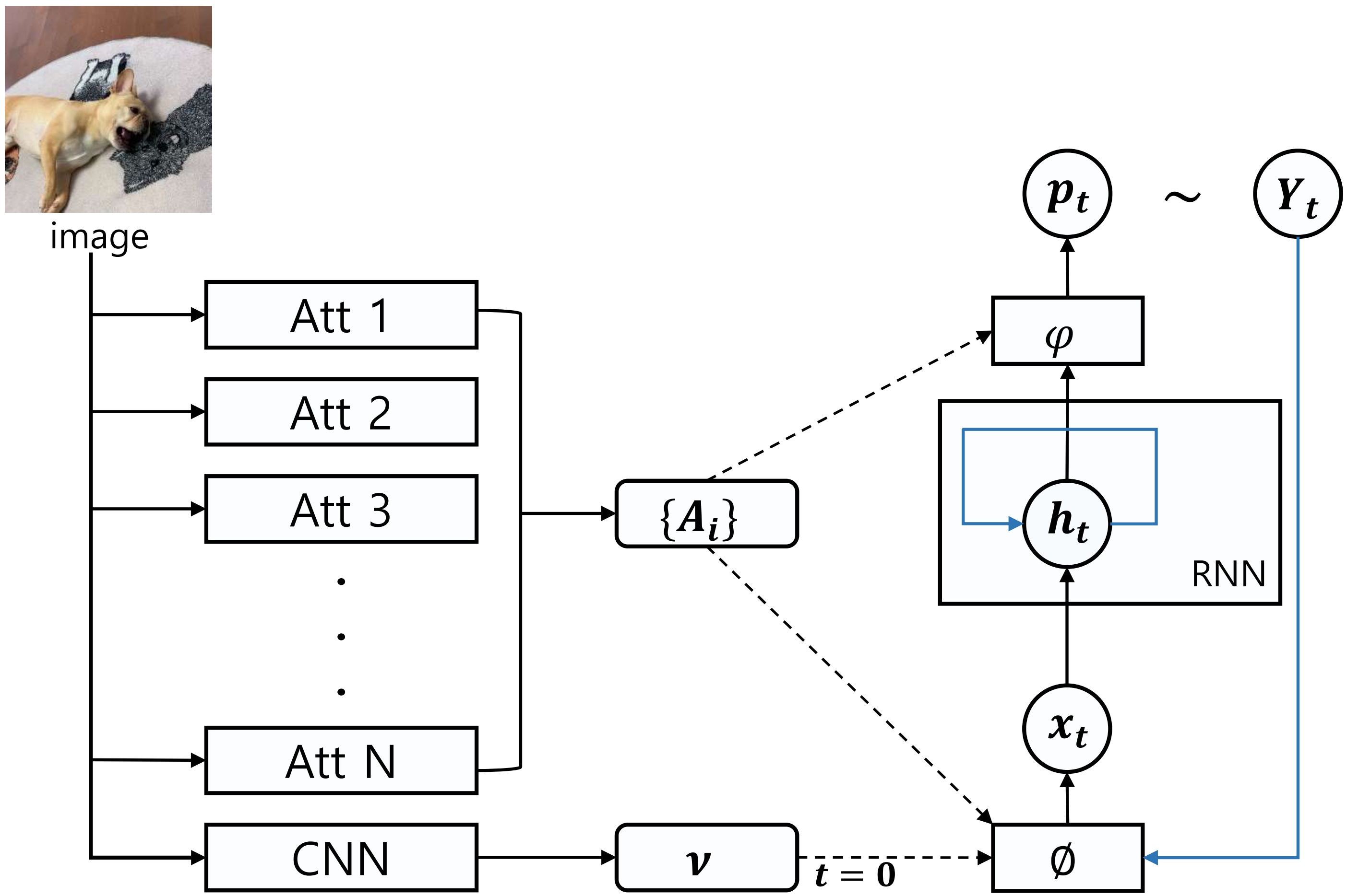
Street, man, umbrella

RNN

The man with the umbrella is walking down the street.

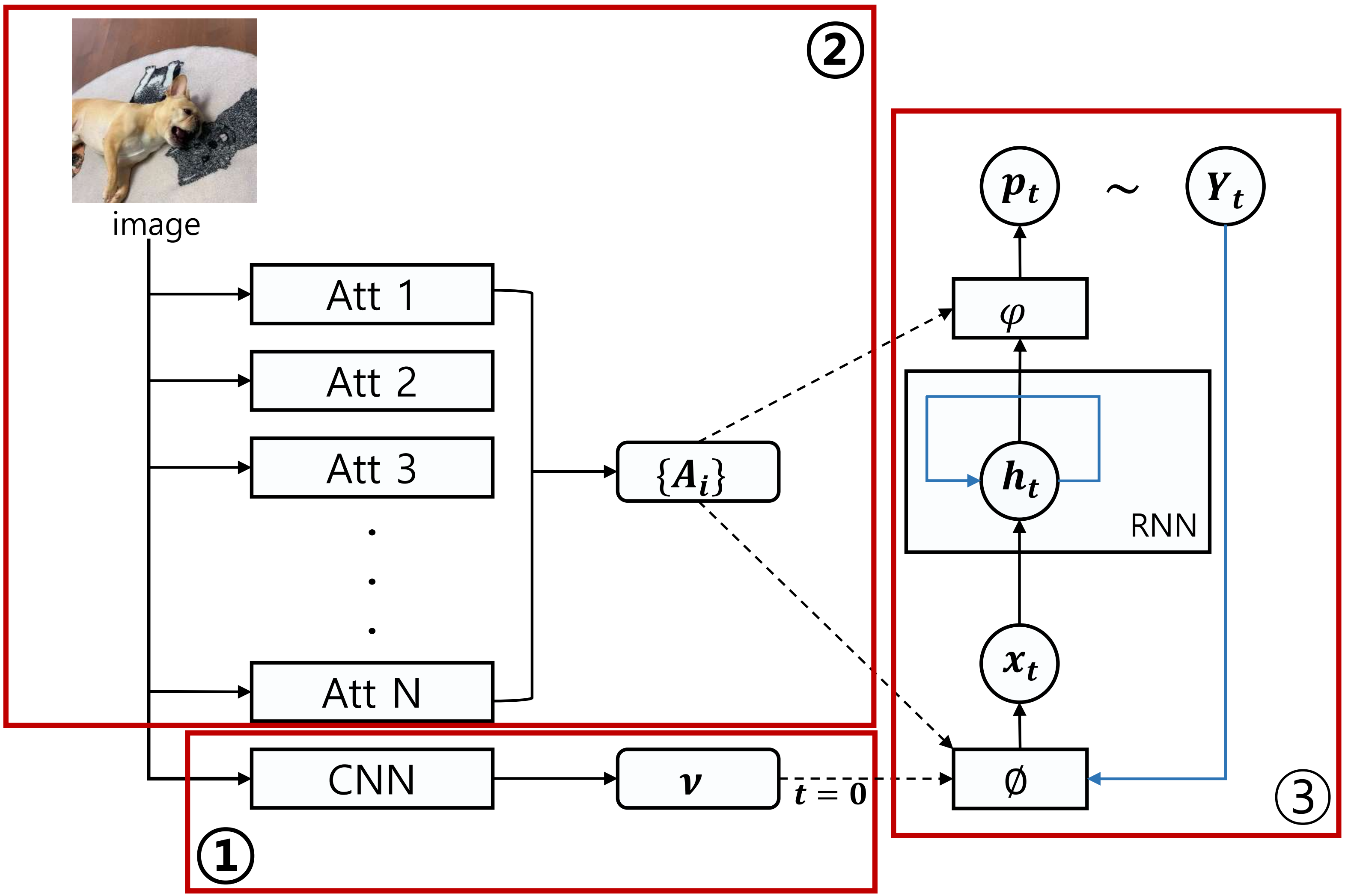
3. Image Captioning with Semantic Attention

❖ Image Captioning with Semantic Attention 구조



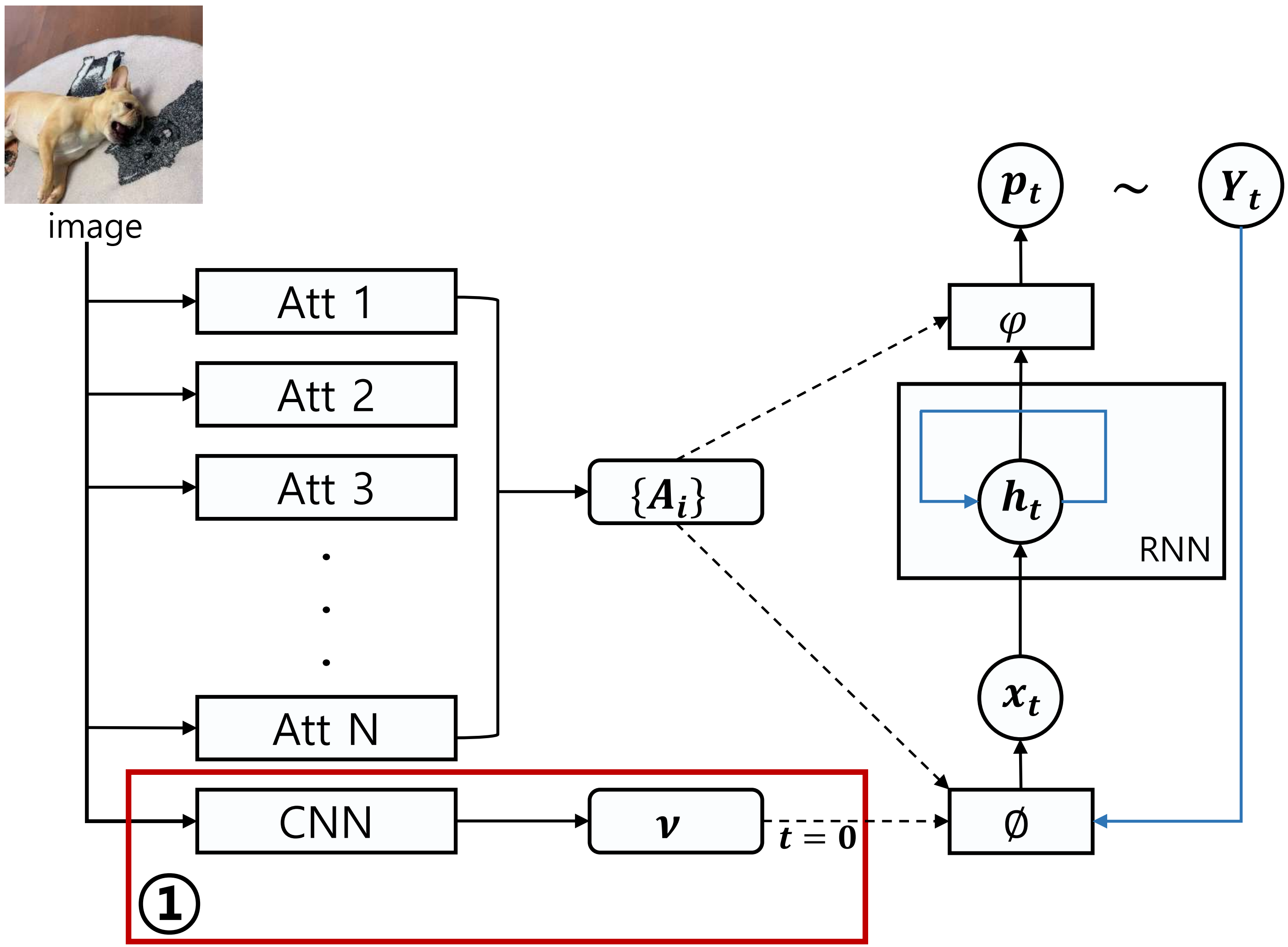
3. Image Captioning with Semantic Attention

❖ Image Captioning with Semantic Attention 구조

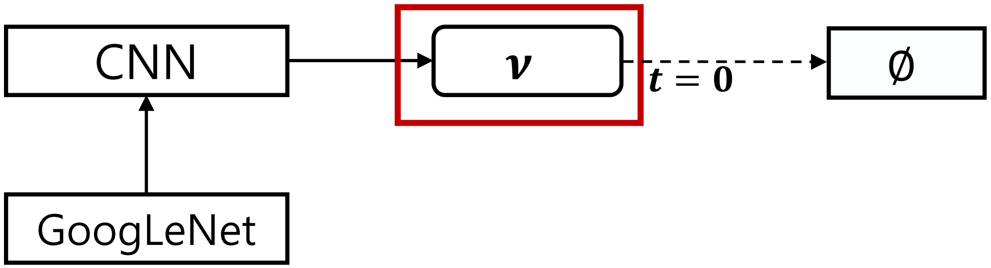


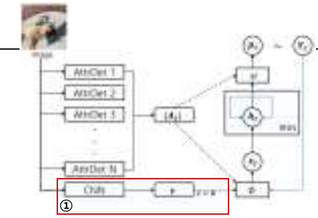
3. Image Captioning with Semantic Attention

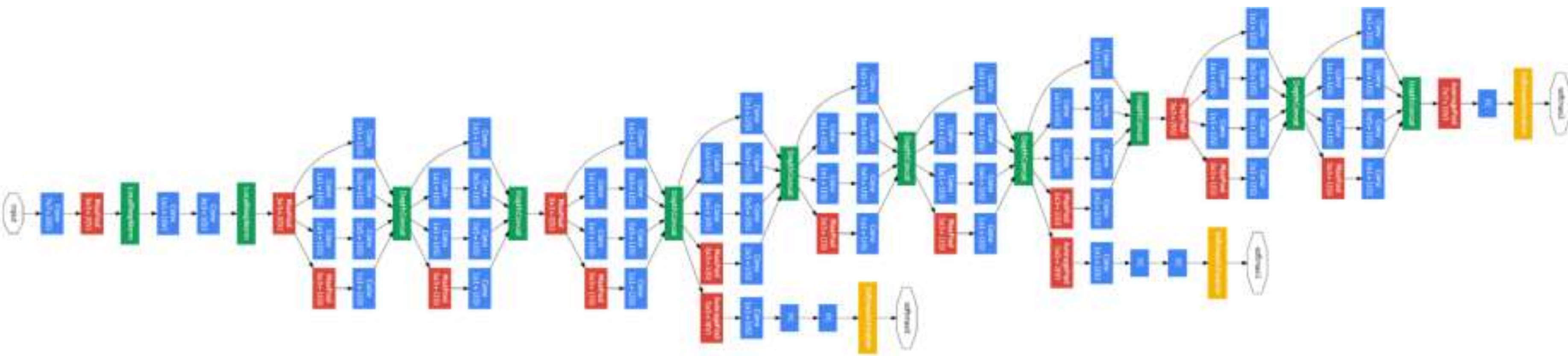
❖ Image Captioning with Semantic Attention 구조



3. Image Captioning with Semantic Attention

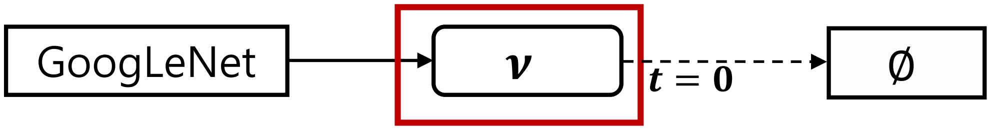


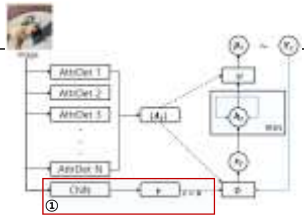




<GoogLeNet 구조>

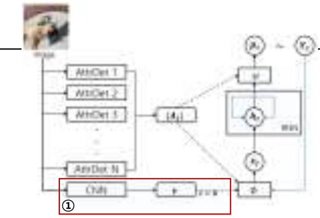
3. Image Captioning with Semantic Attention





Deep 하고 Width하게

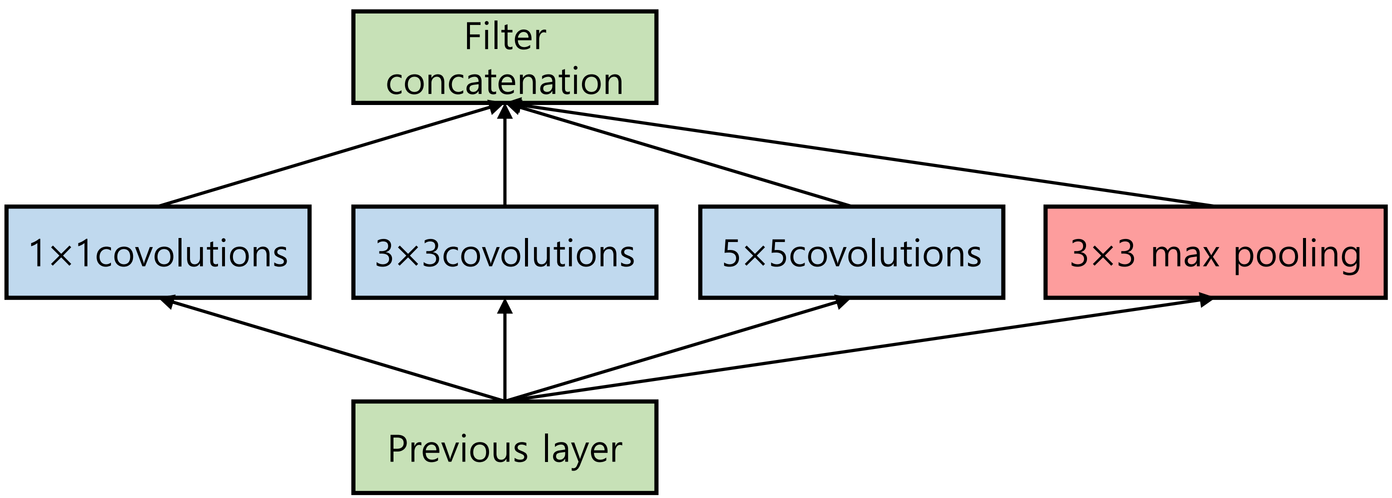
3. Image Captioning with Semantic Attention



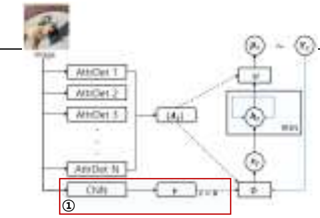
❖ GoogLeNet

1) Inception 모듈

- 여러 size의 filter를 병렬적으로 합쳐보자 !



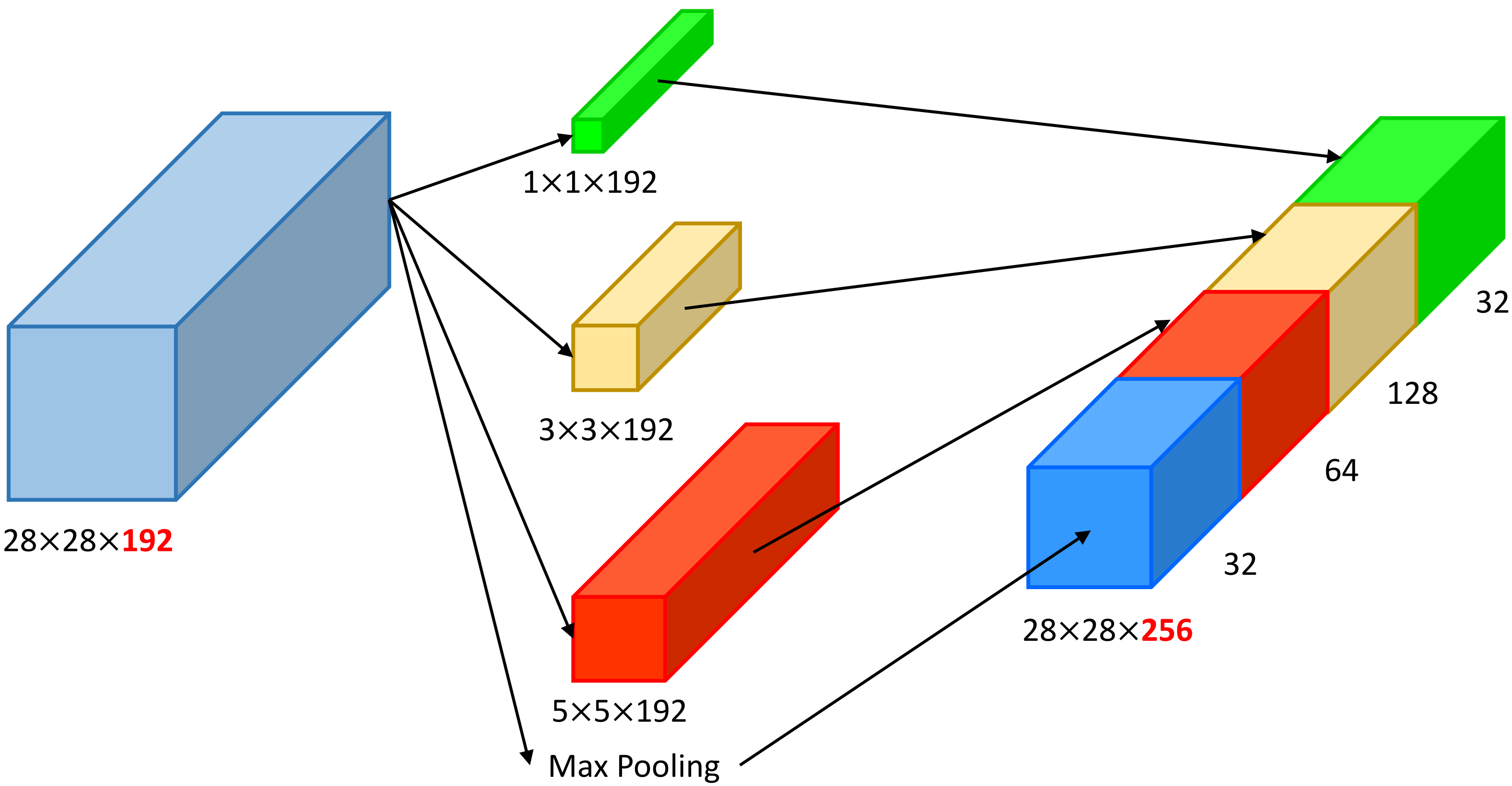
3. Image Captioning with Semantic Attention



❖ GoogLeNet

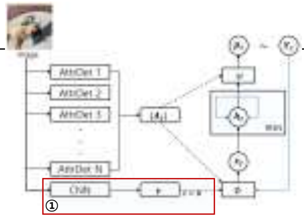
1) Inception 모듈

- 여러 size의 filter를 병렬적으로 합쳐보자 !



Data Mining Quality Analytics

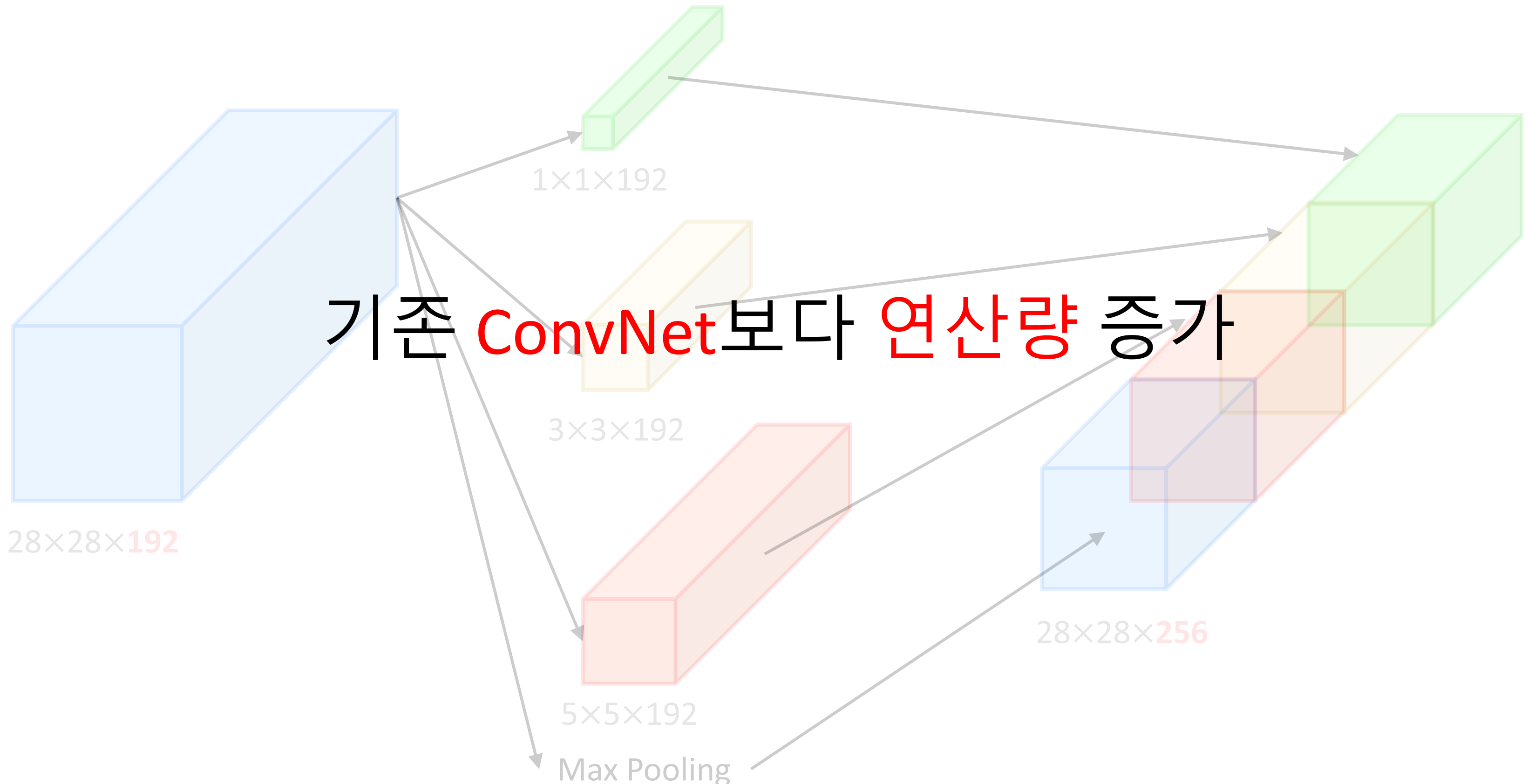
3. Image Captioning with Semantic Attention



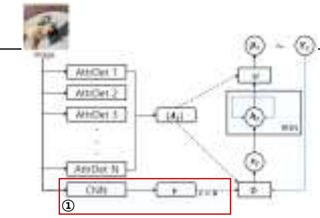
❖ GoogLeNet

1) Inception 모듈

- 여러 size의 filter를 병렬적으로 합쳐보자 !

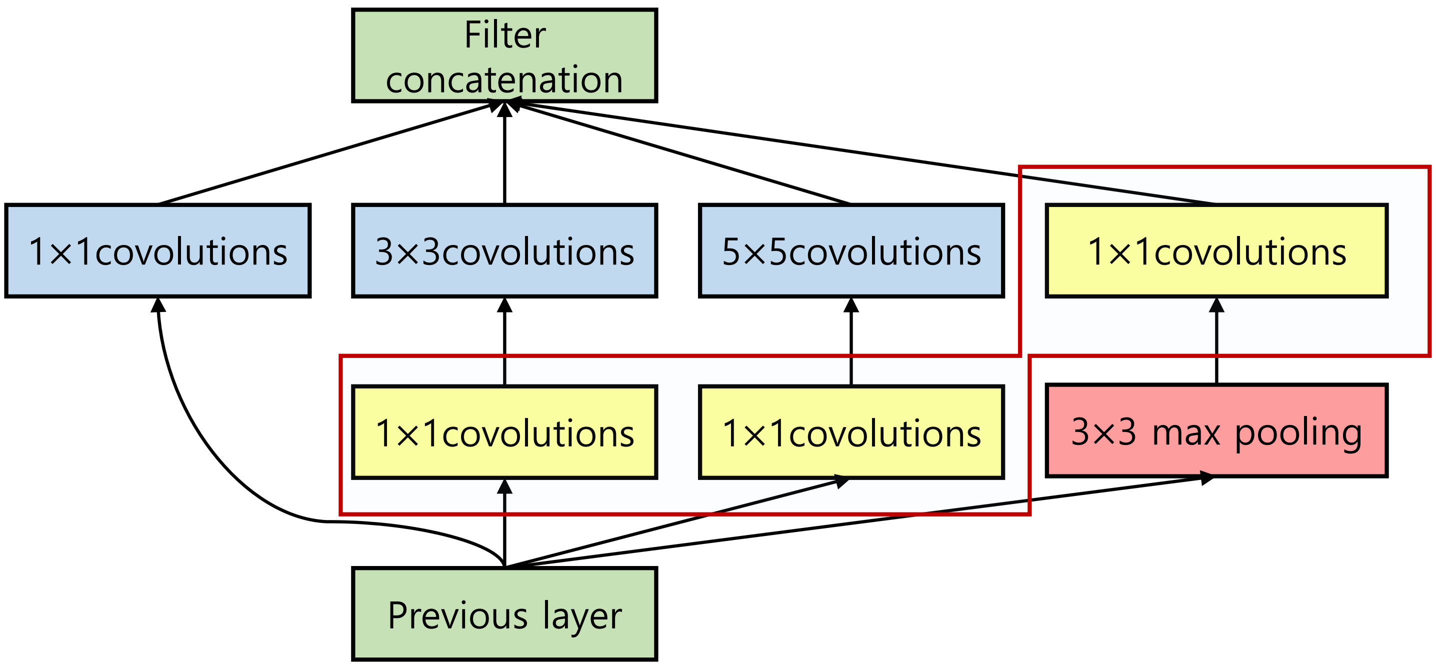


3. Image Captioning with Semantic Attention

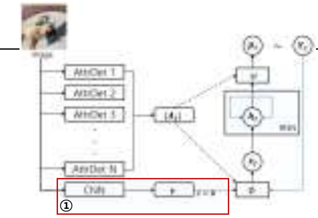


❖ GoogLeNet

1) Inception 모듈

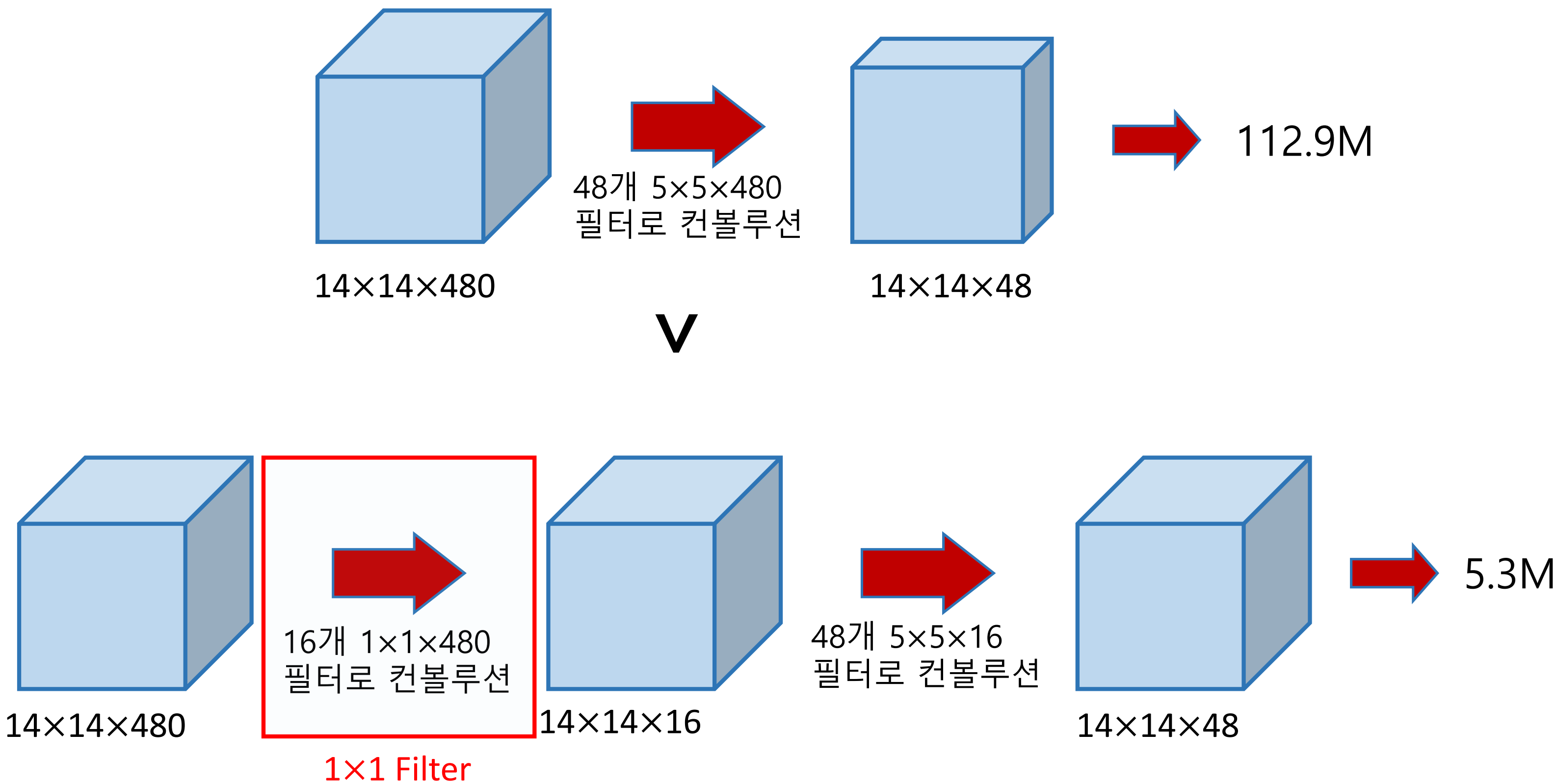


3. Image Captioning with Semantic Attention

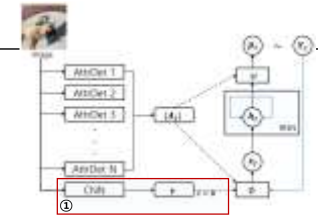


❖ GoogLeNet

2) 1×1 Filter

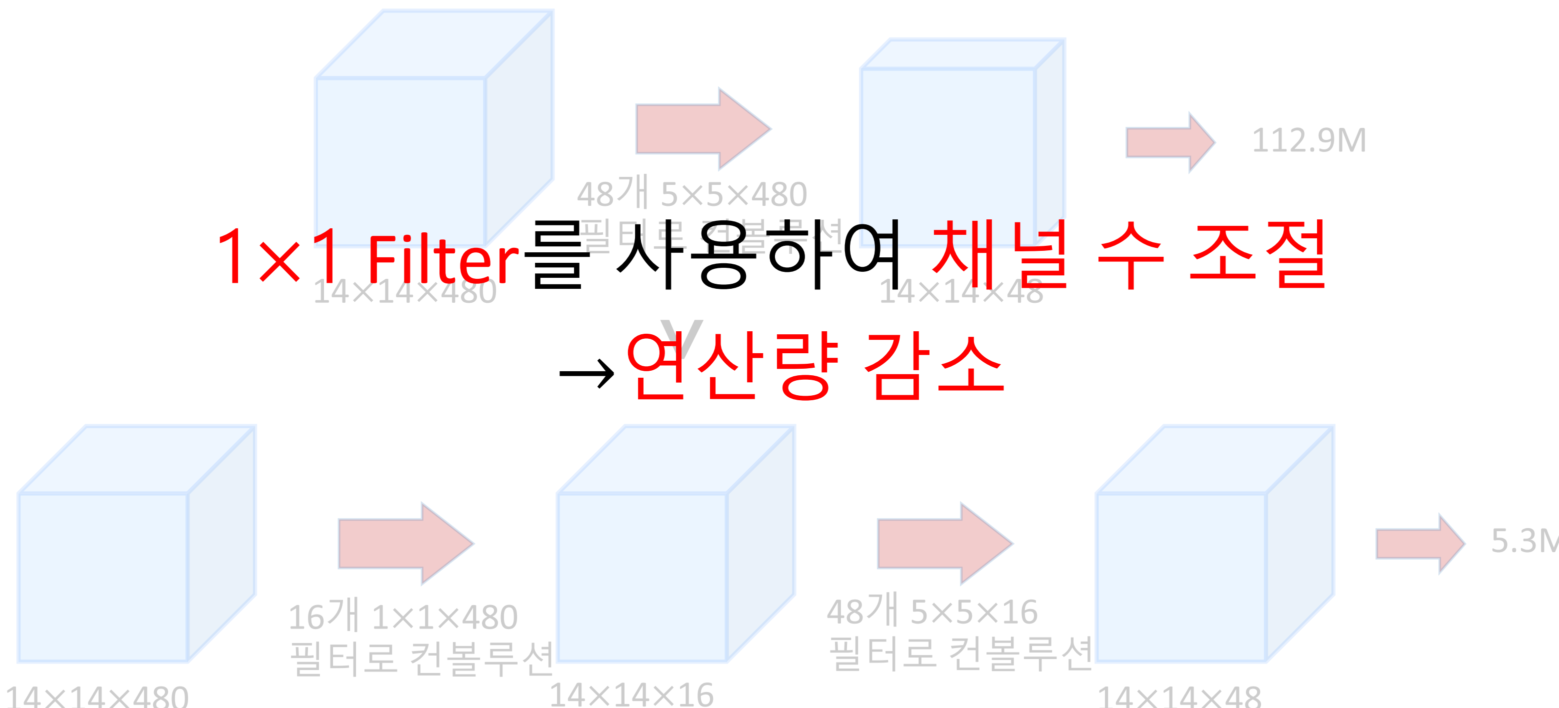


3. Image Captioning with Semantic Attention



❖ GoogLeNet

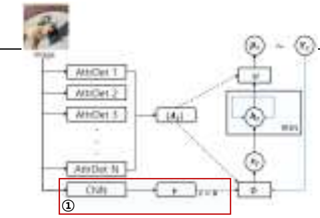
2) 1×1 Filter



1×1 Filter를 사용하여 채널 수 조절

→연산량 감소

3. Image Captioning with Semantic Attention

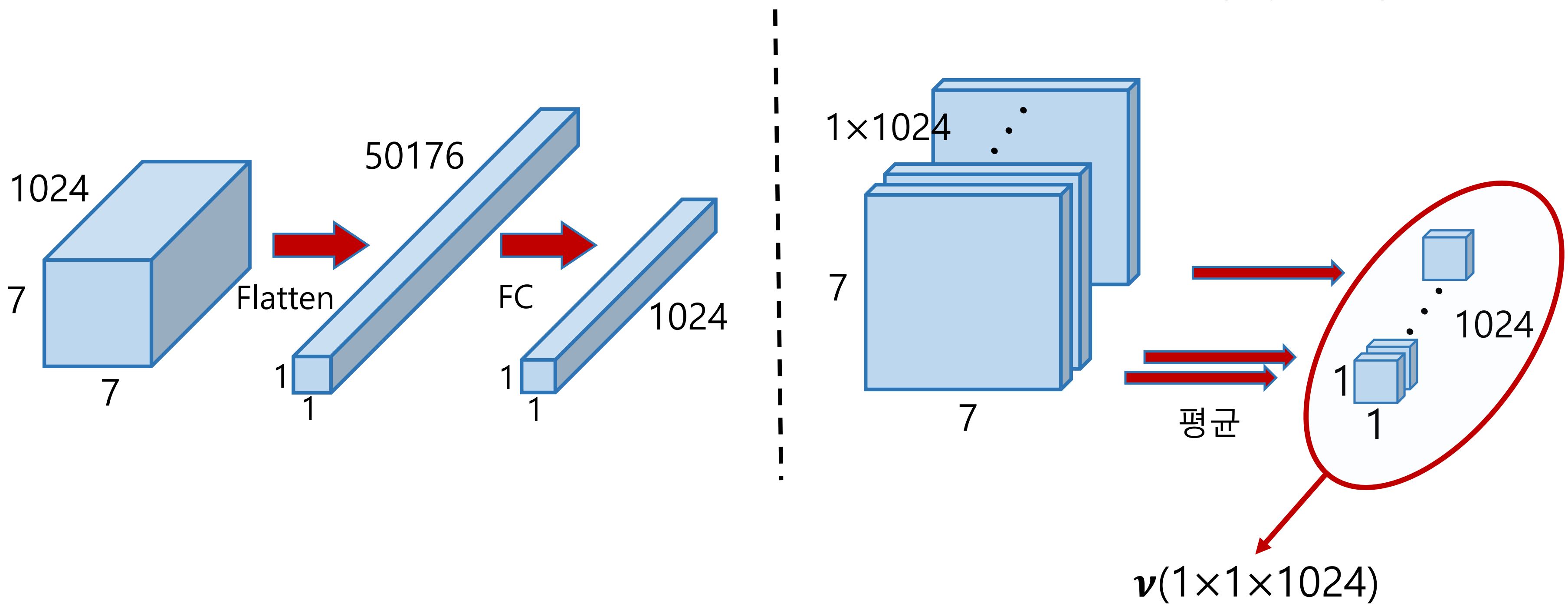


❖ GoogLeNet

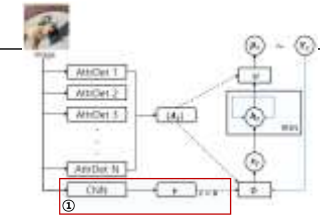
3) Global average pooling

<Fully Connected>

<Global average pooling>



3. Image Captioning with Semantic Attention



❖ GoogLeNet

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Table 1: GoogLeNet incarnation of the Inception architecture.

Image의 특성

GoogLeNet → $\boldsymbol{v}$ $\xrightarrow{t=0}$ ϕ → x_0

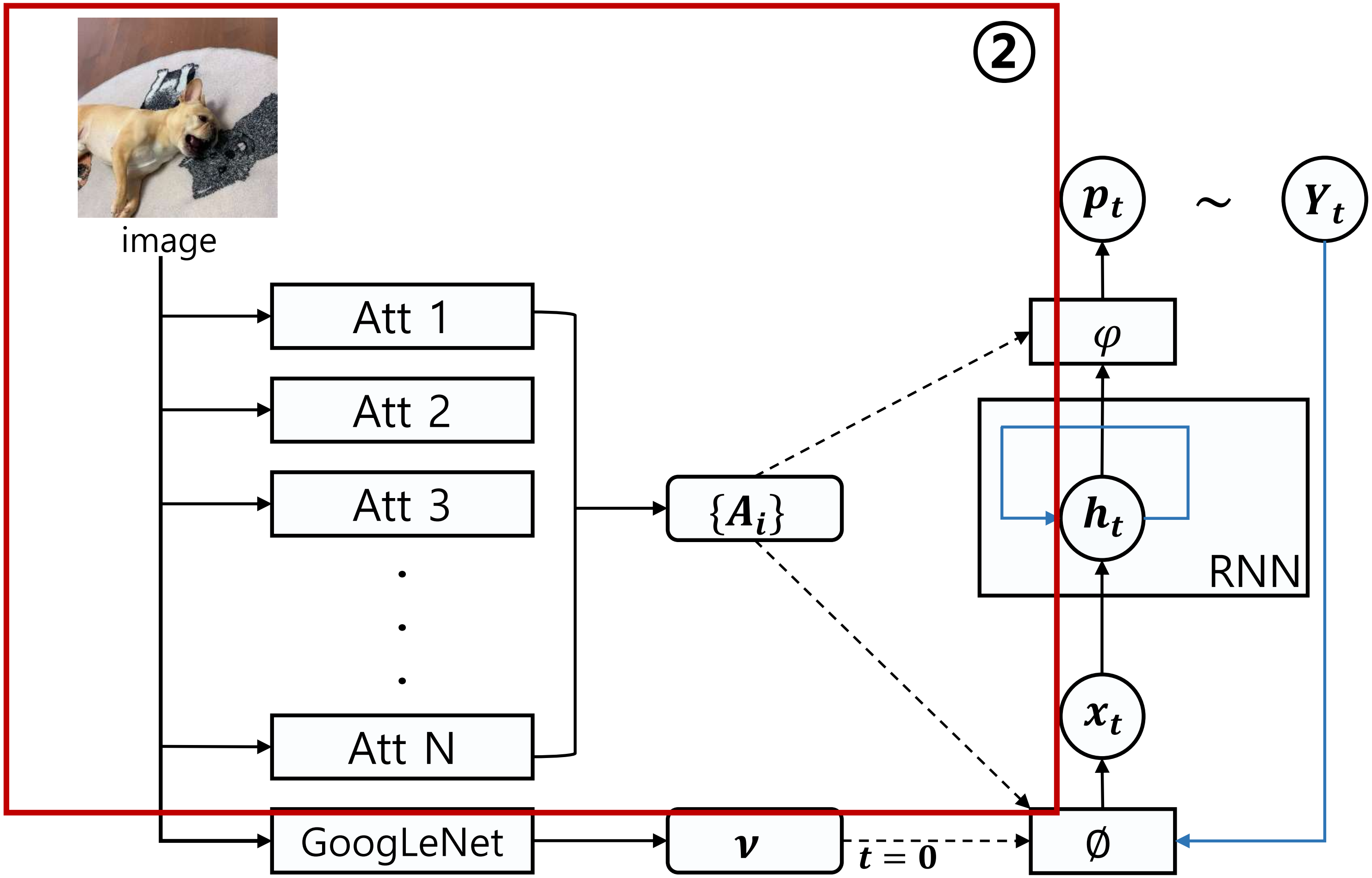
$\boldsymbol{v} = W^{x,v}\, v$

$\phi = \phi_0\,(v)$

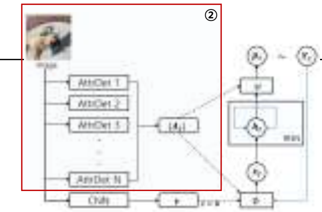
$x_0 = x_0$

3. Image Captioning with Semantic Attention

❖ Image Captioning with Semantic Attention 구조



3. Image Captioning with Semantic Attention



❖ Semantic Attention

- Visual Attribute
  - ➢ Training Data에서 가장 많이 등장한 단어 K개를 이용하여 Class 생성

The dog is yawning.

The man with the umbrella is walking down the street.

The dog is yawning.
The man with the umbrella is walking down the street.
A large bus sitting next to a very tall building.
Man in black shirt is playing guitar.
•
•
•

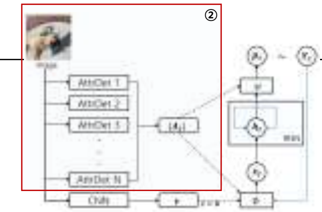
<Training Data>

The
A
Is
Man
Dog
•
•
•

K-Class

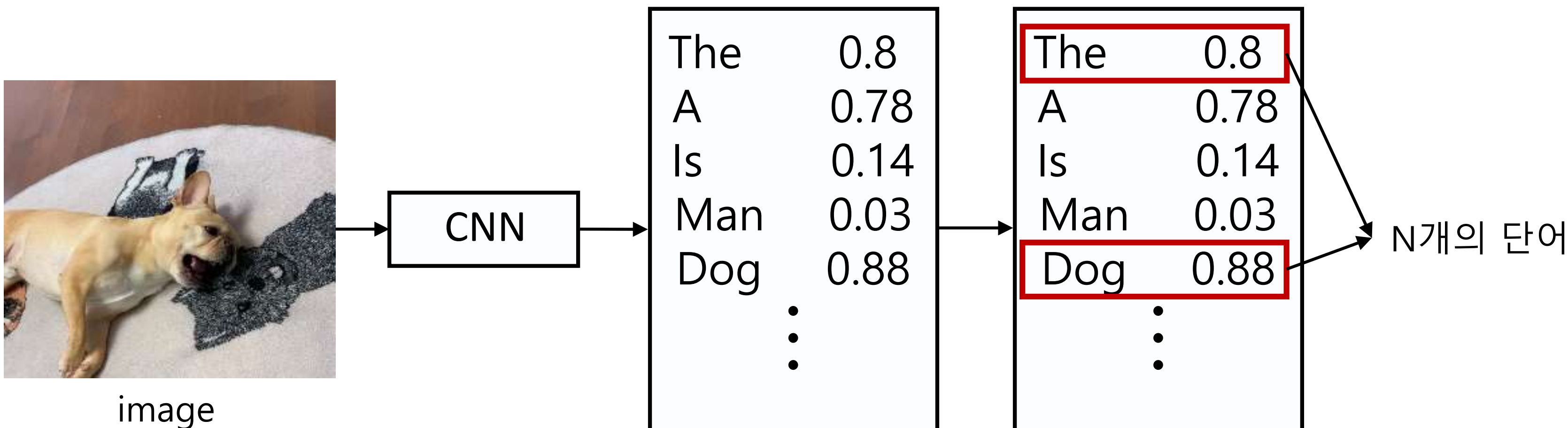
Data Mining Quality Analytics

3. Image Captioning with Semantic Attention



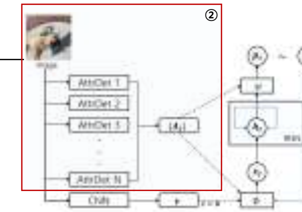
❖ Semantic Attention

- Training Data에서 가장 많이 등장한 단어 K개
- Class score가 가장 높은 단어 N개

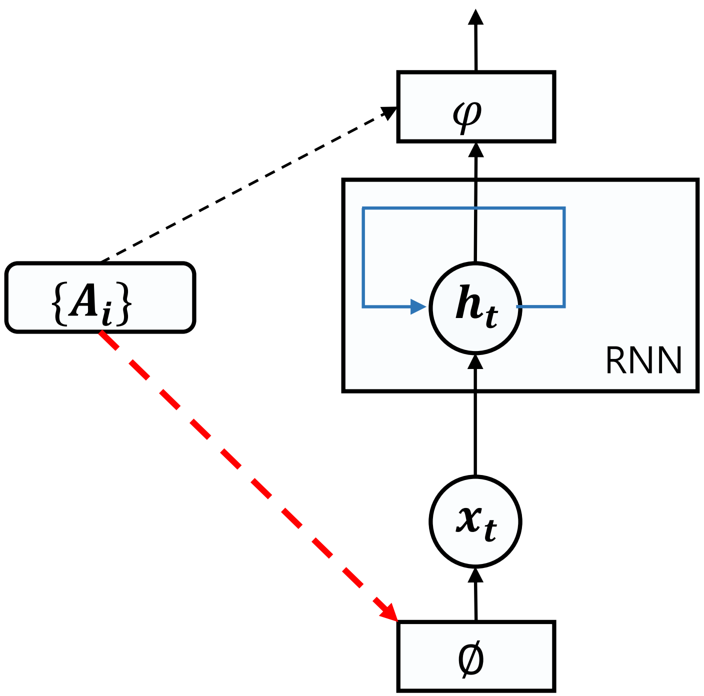


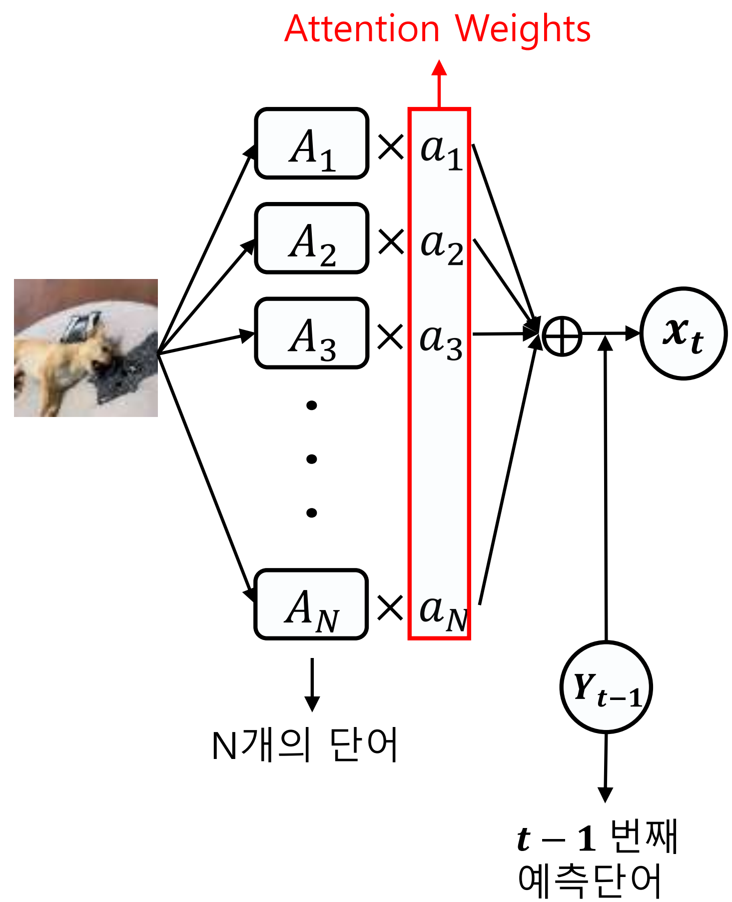
Data Mining Quality Analytics

3. Image Captioning with Semantic Attention

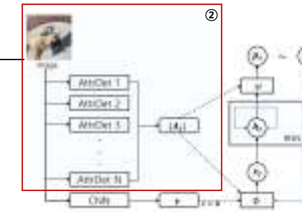


❖ Semantic Attention

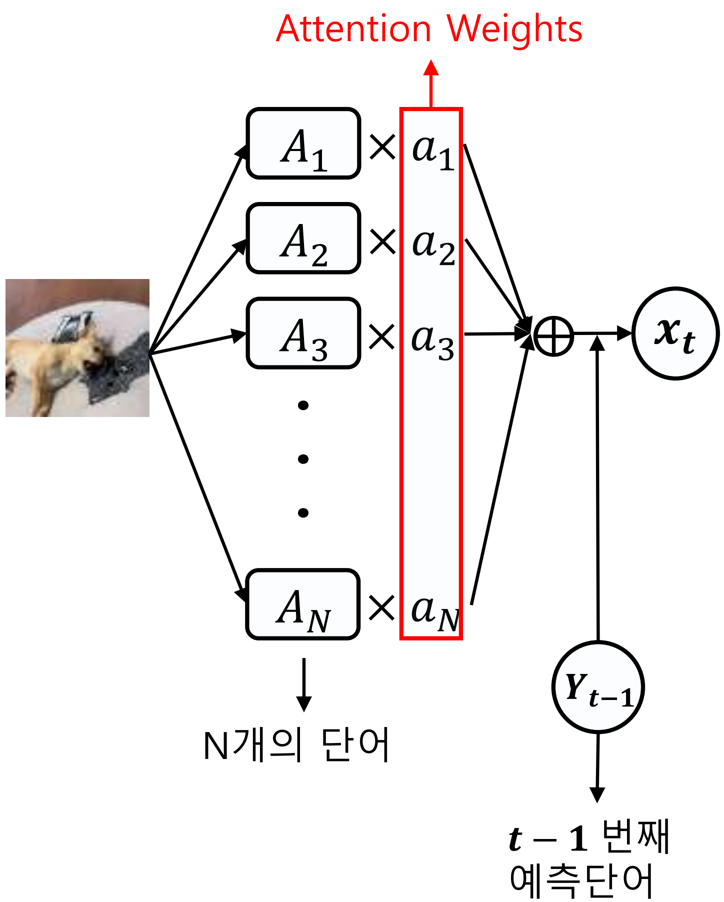




3. Image Captioning with Semantic Attention



❖ Semantic Attention



$A_i \rightarrow y^i$(One Hot Vector)

$Y_{t-1} \rightarrow y_{t-1}$(One Hot Vector)

고차원

$$\alpha_t^i \propto exp\left(y_{t-1}{}^T \tilde{U} y^i\right)$$

$A_i \rightarrow Ey^i$(Embedding Vector)

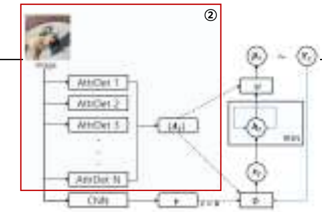
$Y_{t-1} \rightarrow Ey_{t-1}$(Embedding Vector)

저차원

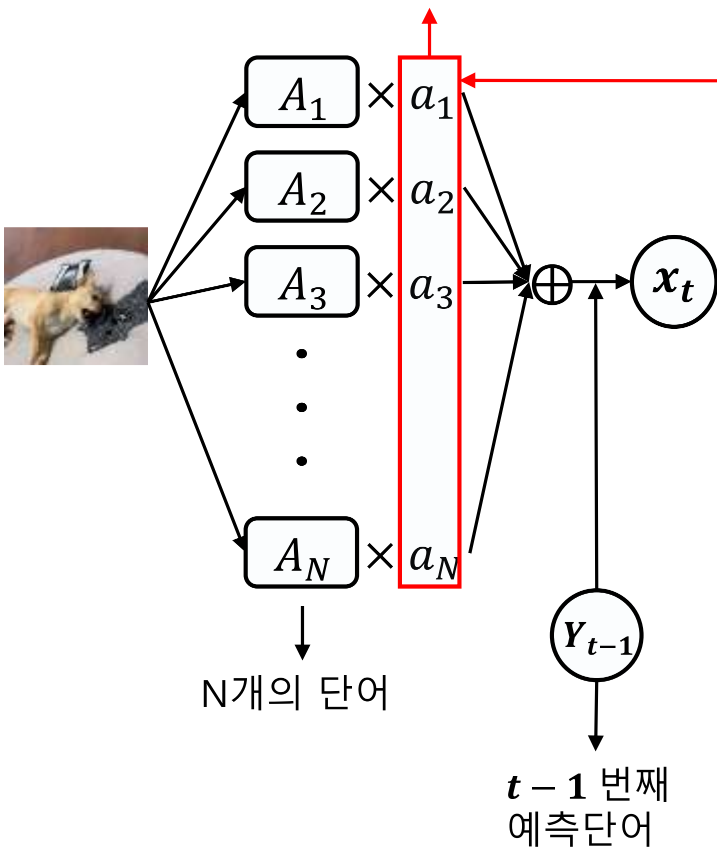
$$\alpha_t^i \propto exp\left((Ey_{t-1})^T U E y^i\right)$$

Data Mining Quality Analytics

3. Image Captioning with Semantic Attention



❖ Semantic Attention



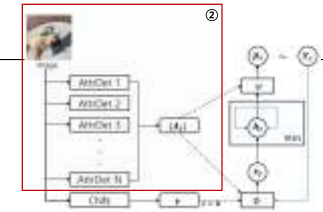
$$\alpha_t^i \propto exp\left((Ey_{t-1})^T UEy^i\right) = \exp(y_{t-1}^T E^T UEy^i)$$

$$x_t = W^{x,Y}(\boxed{Ey_{t-1}} + diag(w^{x,A})\boxed{\sum_i \alpha_t^i Ey^i})$$

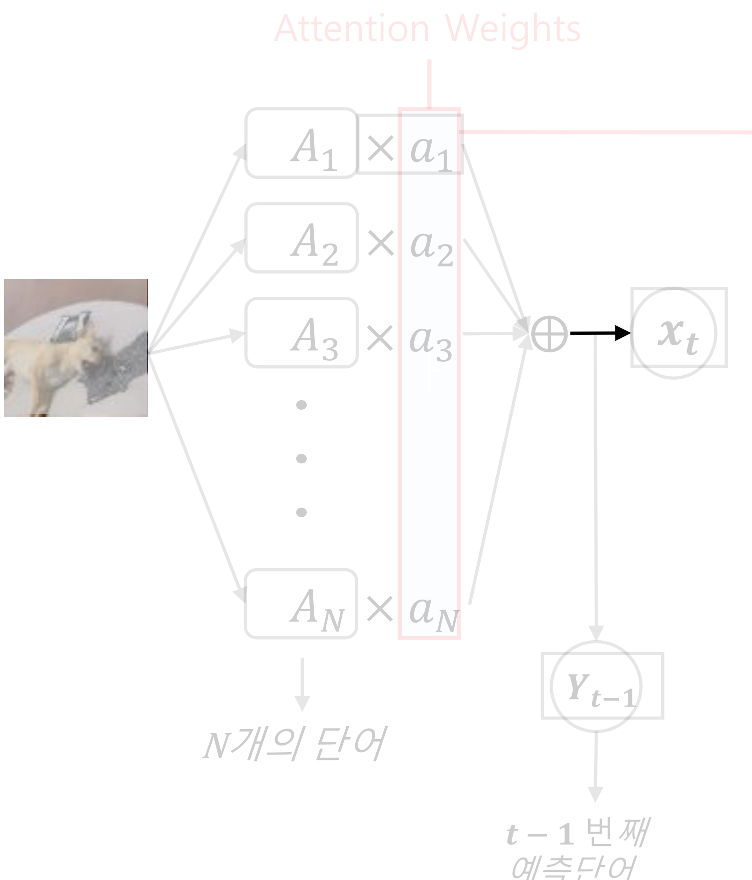
Embedding Vector

Attribute의 Weighted sum

3. Image Captioning with Semantic Attention



❖ Semantic Attention

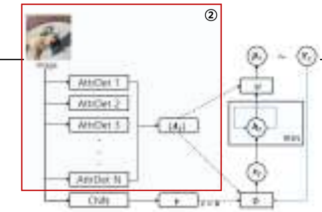


$$\alpha_t^i \propto exp\left((Ey_{t-1})^T UEy^i\right) = \exp(y_{t-1}^T E^T UEy^i)$$

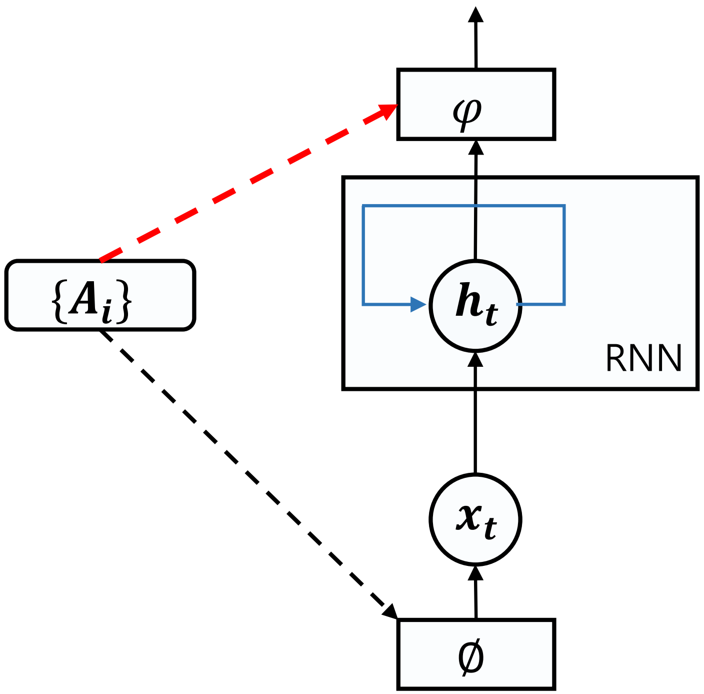
$$x_t = \emptyset(Y_{t-1}, \{A_i\})$$

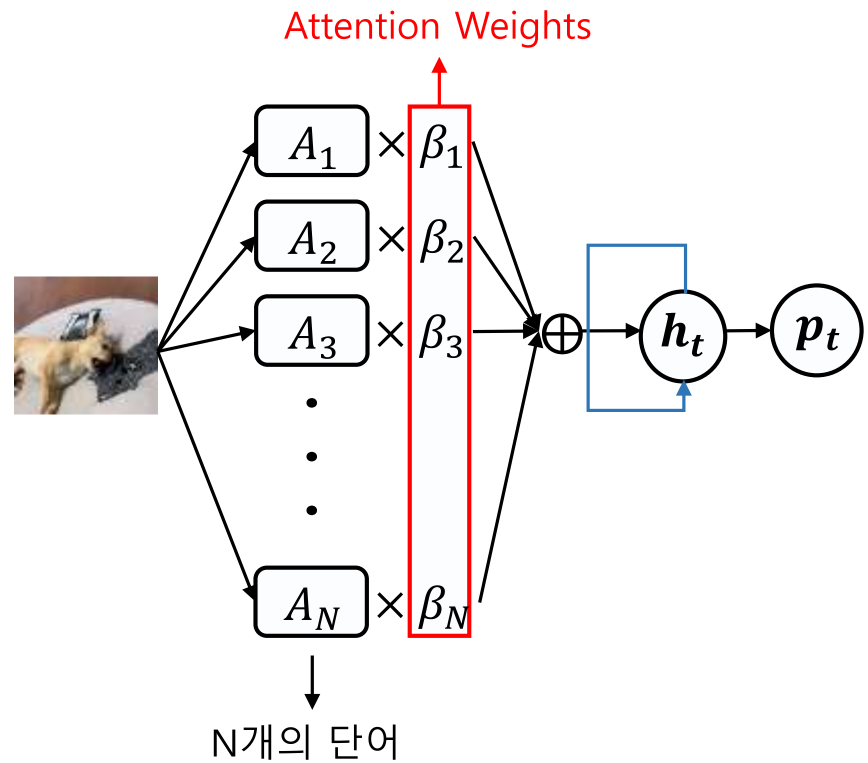
$$x_t = W^{x,Y}(\underbrace{Ey_{t-1}}_{\text{Embedding Vector}} + diag(w^{x,A}) \underbrace{\sum_i \alpha_t^i Ey^i}_{\text{Attribute의 Weighted sum}})$$

3. Image Captioning with Semantic Attention

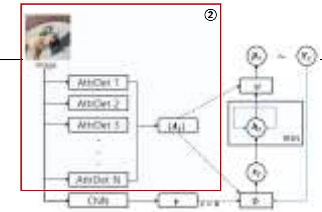


❖ Semantic Attention

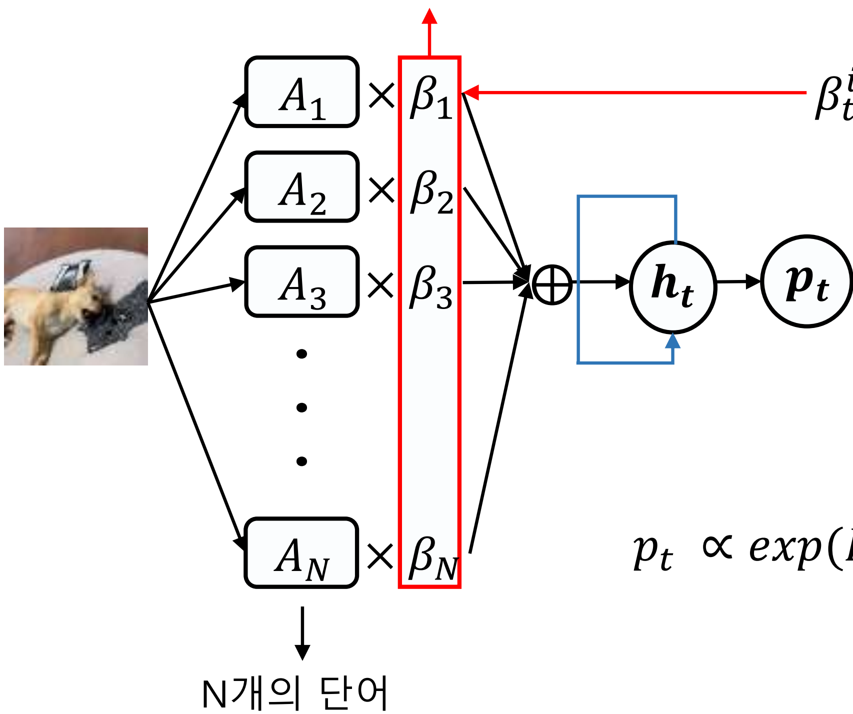




3. Image Captioning with Semantic Attention



❖ Semantic Attention

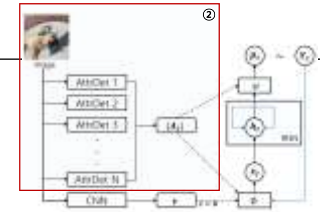


$$\beta_t^i \propto exp(h_t^i V \sigma(E y^i))$$

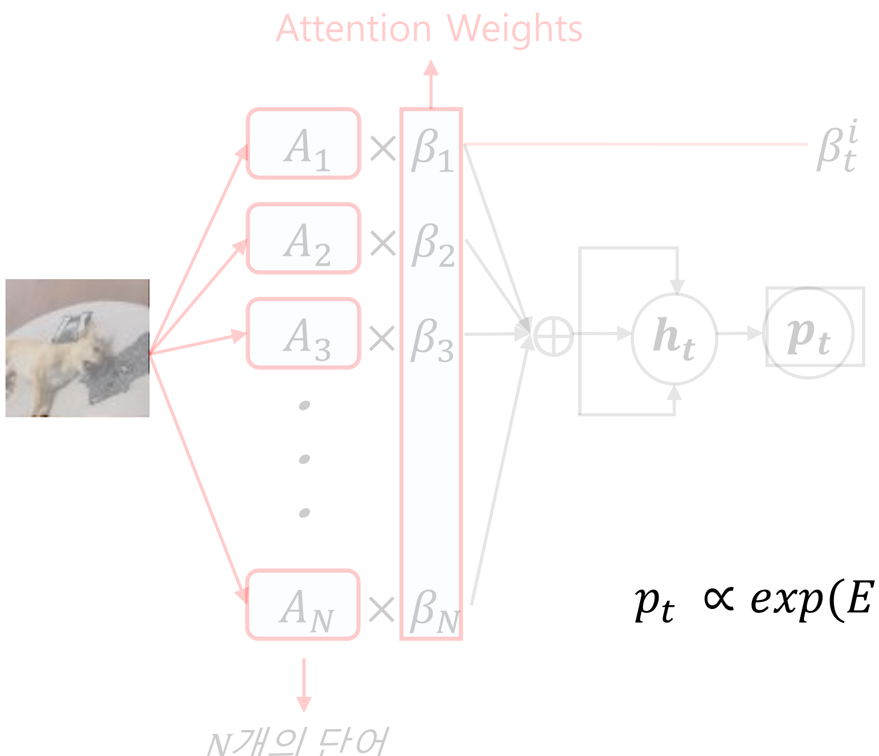
$$p_t \propto exp(E^T W^{Y,h}(h_t + diag(w^{Y,A}) \boxed{\sum_i \beta_t^i \sigma(E y^i)}))$$

Attribute의 Weighted sum

3. Image Captioning with Semantic Attention



❖ Semantic Attention



$$\beta_t^i \propto exp(h_t^i V \sigma(E y^i))$$

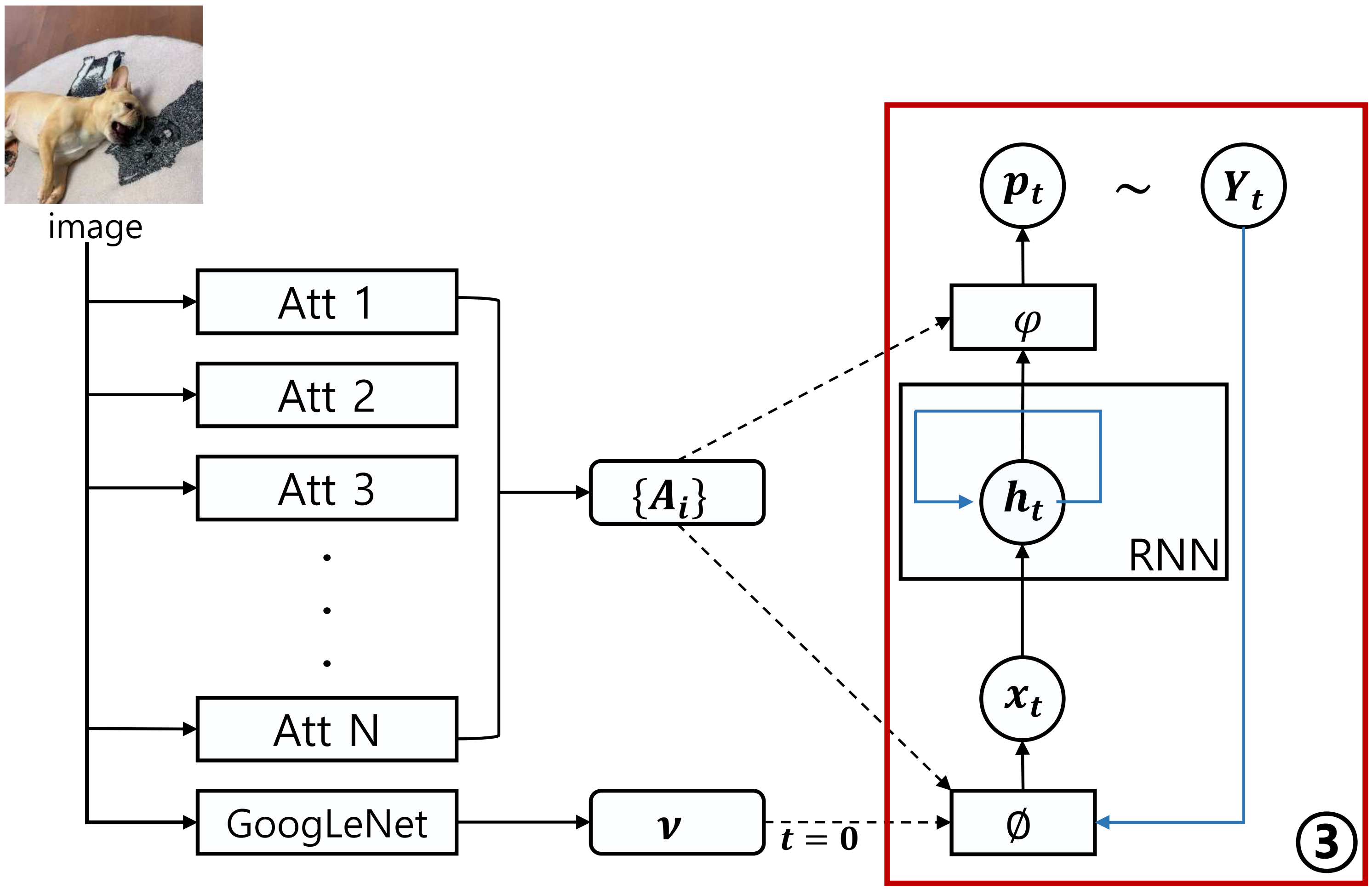
$$p_t = \varphi(h_t, \{A_i\})$$

$$p_t \propto exp(E^T W^{Y,h}(h_t + diag(w^{Y,A}) \boxed{\sum_i \beta_t^i \sigma(E y^i)}))$$

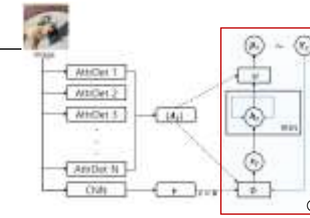
Attribute의 Weighted sum

3. Image Captioning with Semantic Attention

❖ Image Captioning with Semantic Attention 구조

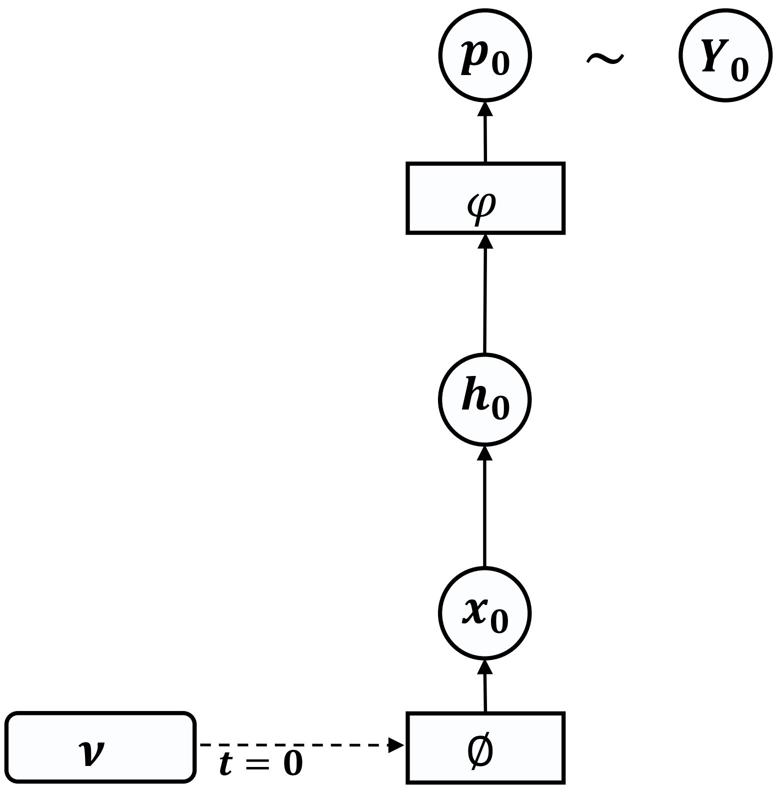


3. Image Captioning with Semantic Attention



❖ RNN

- t = 0인 경우



Input : $x_0 = \emptyset_0(v) = W^{x,v}v, \ t = 0$

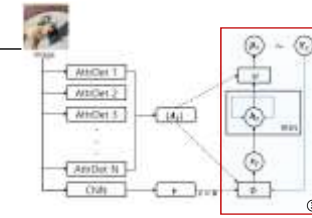
Calculation : $h_0 =$ 초기 hidden state vector

$Y_0 \sim p_0 = \varphi(h_0)$

Output : $p_0 \sim Y_0$

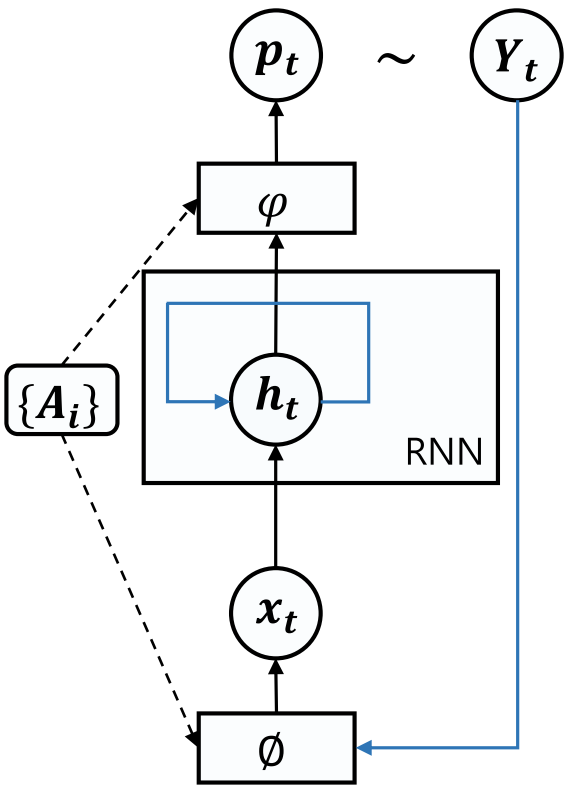
- p_t : 단어 Y_t 가 나올 확률 벡터
- A_i : i 번째 단어
- h_t : State t의 hidden state vector

3. Image Captioning with Semantic Attention



❖ RNN

- $t > 0$ 인 경우



Input : $x_t = \emptyset(Y_{t-1}, \{A_i\}),\ t > 0$

Calculation : $h_t = RNN(h_{t-1,} x_t)$

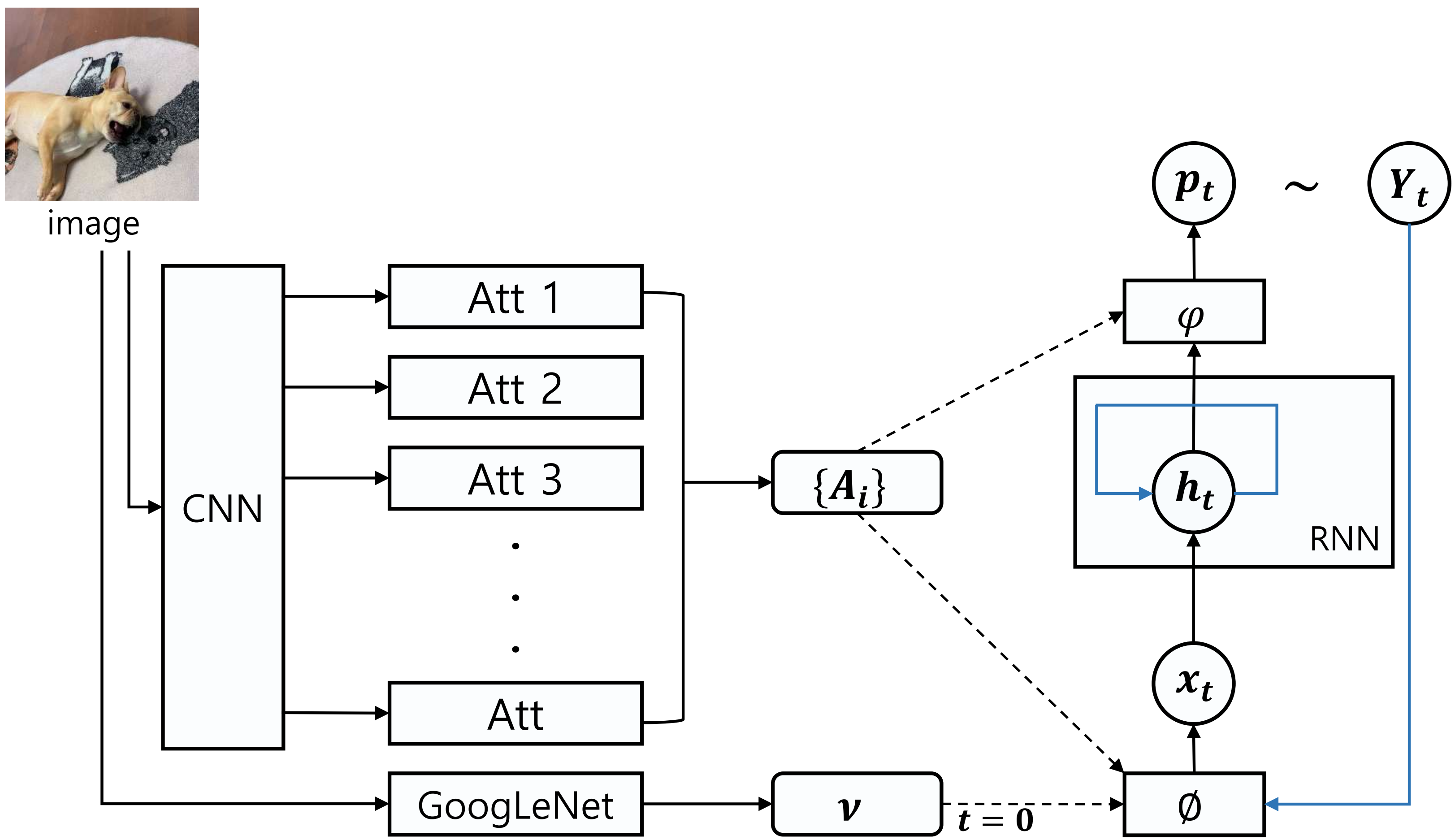
$$Y_t \sim p_t = \varphi(h_t, \{A_i\})$$

Output : p_t

- p_t : 단어 Y_t 가 나올 확률 벡터
- A_i : i 번째 단어
- h_t : State t의 hidden state vector

3. Image Captioning with Semantic Attention

❖ Image Captioning with Semantic Attention 구조



4. Result

❖ Image Captioning with Semantic Attention 결과 비교

	Flickr30k					MS-COCO				
Model	B-1	B-2	B-3	B-4	METEOR	B-1	B-2	B-3	B-4	METEOR
Google NIC [35]	0.663	0.423	0.277	0.183	–	0.666	0.451	0.304	0.203	–
m-RNN [26]	0.60	0.41	0.28	0.19	–	0.67	0.49	0.35	0.25	–
LRCN [8]	0.587	0.39	0.25	0.165	–	0.628	0.442	0.304	0.21	–
MSR/CMU [4]	–	–	–	0.126	0.164	–	–	–	0.19	0.204
Toronto [37]	**0.669**	0.439	0.296	0.199	0.185	**0.718**	0.504	0.357	0.250	0.230
Ours-CON-k-NN	0.619	0.426	0.291	0.197	0.179	0.675	0.503	0.373	0.279	0.227
Ours-CON-RK	0.623	0.432	0.295	0.200	0.179	0.647	0.472	0.338	0.237	0.204
Ours-CON-FCN	0.639	0.447	0.309	0.213	0.188	0.700	0.532	0.398	0.300	0.238
Ours-MAX-k-NN	0.622	0.426	0.287	0.193	0.178	0.673	0.501	0.371	0.279	0.227
Ours-MAX-RK	0.623	0.429	0.294	0.202	0.178	0.655	0.478	0.344	0.245	0.208
Ours-MAX-FCN	0.633	0.444	0.306	0.21	0.181	0.699	0.530	0.398	0.301	0.240
Ours-ATT-k-NN	0.618	0.428	0.290	0.195	0.172	0.676	0.505	0.375	0.281	0.227
Ours-ATT-RK	0.617	0.424	0.286	0.193	0.177	0.679	0.506	0.375	0.282	0.231
Ours-ATT-FCN	0.647	**0.460**	**0.324**	**0.230**	**0.189**	0.709	**0.537**	**0.402**	**0.304**	**0.243**

4. Result

❖ Image Captioning with Semantic Attention 결과 비교

- BLEU Score

$$BLEU = BP \times exp(\sum_{n=1}^{N} W_n log p_n)$$

- c : 생성 문장 길이
- r : 실제 문장 길이
- N : n-gram에서의 n의 최대 숫자(보통은 4까지)
- $p_n : \frac{Count_{clip}}{Count}$
- W_n : $Weight$
- $BP = \begin{cases} 1 & if\ c > r \\ e^{(1-\frac{r}{c})} & if\ c \le r \end{cases}$

4. Result

❖ Image Captioning with Semantic Attention 결과 비교

- BLEU Score

- c : 생성 문장 길이, r : 실제 문장 길이
- N : n-gram에서의 n의 최대 숫자
- $p_n : \frac{Count_{clip}}{Count}$, $W_n : Weight$
- $BP = \begin{cases} 1 & if\ c > r \\ e^{(1-\frac{r}{c})} & if\ c \le r \end{cases}$

1 2 3 4 5 6 7
생성 문장 : the dog the dog on the mat

1 2 3 4 5 6
실제 문장 : the dog is on the mat

c = 7, r = 6

4. Result

❖ Image Captioning with Semantic Attention 결과 비교

- BLEU Score

  ➢ N-gram : n개의 연속적인 단어 나열

- c : 생성 문장 길이, r : 실제 문장 길이
- N : n-gram에서의 n의 최대 숫자
- $p_n : \frac{Count_{clip}}{Count}$, $W_n : Weight$
- $BP = \begin{cases} 1 & if\ c > r \\ e^{(1-\frac{r}{c})} & if\ c \le r \end{cases}$

n = 2일 경우

생성 문장 : the dog the dog on the mat

1	2	3	4	5	6
the dog	dog the	the dog	dog on	on the	the mat

⇩

6

4. Result

- c : 생성 문장 길이, r : 실제 문장 길이
- N : n-gram에서의 n의 최대 숫자
- $p_n : \frac{Count_{clip}}{Count}$, W_n : $Weight$
- $BP = \begin{cases} 1 & if\ c > r \\ e^{(1-\frac{r}{c})} & if\ c \leq r \end{cases}$

❖ Image Captioning with Semantic Attention 결과 비교

- BLEU Score

생성 문장 : the dog the dog on the mat

실제 문장 : the dog is on the mat

$n = 2$	the dog	dog the	dog on	on the	the mat	SUM
$Count$	2	1	1	1	1	6
$Count_{clip}$	1	0	0	1	1	3

$Count$: 생성된 문장의 n-gram 수

$Count_{clip}$: 생성된 문장의 n-gram을 기준으로 등장하는 n-gram 수

4. Result

- c : 생성 문장 길이, r : 실제 문장 길이
- N : n-gram에서의 n의 최대 숫자
- $p_n : \frac{Count_{clip}}{Count}$, $W_n : Weight$
- $BP = \begin{cases} 1 & if\ c > r \\ e^{(1-\frac{r}{c})} & if\ c \leq r \end{cases}$

❖ Image Captioning with Semantic Attention 결과 비교

- BLEU Score
  - ➢ $c \leq r$인 경우

생성 문장 : the dog

실제 문장 : the dog is on the mat

$n = 2$	**the dog**	**SUM**
$Count$	1	1
$Count_{clip}$	1	1

$p_2 = \frac{1}{1} = 1$(확률은 높지만 제대로 된 문장이 아님) $\rightarrow$ $\boxed{BP = e^{(1-\frac{5}{2})}}$

Penalty

Data Mining Quality Analytics

4. Result

- c : 생성 문장 길이, r : 실제 문장 길이
- N : n-gram에서의 n의 최대 숫자
- $p_n : \frac{Count_{clip}}{Count}$, $W_n : Weight$
- $BP = \begin{cases} 1 & if\ c > r \\ e^{(1-\frac{r}{c})} & if\ c \le r \end{cases}$

❖ Image Captioning with Semantic Attention 결과 비교

- BLEU Score
  - ➢ $c > r$인 경우

생성 문장 : the dog the dog on the mat

실제 문장 : the dog is on the mat

$\boldsymbol{n = 2}$	**the dog**	**dog the**	**dog on**	**on the**	**the mat**	**SUM**
$Count$	2	1	1	1	1	6
$Count_{clip}$	1	0	0	1	1	3

생성 문장이 길어지면 분모의 count의 합이 커지게 됨

↓

p_2가 자연스럽게 낮아 짐

4. Result

❖ Image Captioning with Semantic Attention 결과 비교

- BLEU Score

- c : 생성 문장 길이
- r : 실제 문장 길이
- $p_n : \frac{Count_{clip}}{Count}$
- W_n : $Weight$
- $BP = \begin{cases} 1 & if\ c > r \\ e^{(1-\frac{r}{c})} & if\ c \le r \end{cases}$

$$BLEU = BP \times exp(\sum_{n=1}^{N} W_n log p_n)$$

생성 문장 : the dog the dog on the mat

실제 문장 : the dog is on the mat

$\boldsymbol{n = 2}$	**the dog**	**dog the**	**dog on**	**on the**	**the mat**	**SUM**
$Count$	2	1	1	1	1	6
$Count_{clip}$	1	0	0	1	1	3

$$p_2 = \frac{3}{6} = \frac{1}{2}\ ,\ BP = 1$$

4. Result

❖ Image Captioning with Semantic Attention 결과 비교

	Flickr30k					MS-COCO				
Model	B-1	B-2	B-3	B-4	METEOR	B-1	B-2	B-3	B-4	METEOR
Google NIC [35]	0.663	0.423	0.277	0.183	–	0.666	0.451	0.304	0.203	–
m-RNN [26]	0.60	0.41	0.28	0.19	–	0.67	0.49	0.35	0.25	–
LRCN [8]	0.587	0.39	0.25	0.165	–	0.628	0.442	0.304	0.21	–
MSR/CMU [4]	–	–	–	0.126	0.164	–	–	–	0.19	0.204
Toronto [37]	**0.669**	0.439	0.296	0.199	0.185	**0.718**	0.504	0.357	0.250	0.230
Ours-CON-k-NN	0.619	0.426	0.291	0.197	0.179	0.675	0.503	0.373	0.279	0.227
Ours-CON-RK	0.623	0.432	0.295	0.200	0.179	0.647	0.472	0.338	0.237	0.204
Ours-CON-FCN	0.639	0.447	0.309	0.213	0.188	0.700	0.532	0.398	0.300	0.238
Ours-MAX-k-NN	0.622	0.426	0.287	0.193	0.178	0.673	0.501	0.371	0.279	0.227
Ours-MAX-RK	0.623	0.429	0.294	0.202	0.178	0.655	0.478	0.344	0.245	0.208
Ours-MAX-FCN	0.633	0.444	0.306	0.21	0.181	0.699	0.530	0.398	0.301	0.240
Ours-ATT-k-NN	0.618	0.428	0.290	0.195	0.172	0.676	0.505	0.375	0.281	0.227
Ours-ATT-RK	0.617	0.424	0.286	0.193	0.177	0.679	0.506	0.375	0.282	0.231
Ours-ATT-FCN	0.647	**0.460**	**0.324**	**0.230**	**0.189**	0.709	**0.537**	**0.402**	**0.304**	**0.243**

4. Result

❖ Image Captioning with Semantic Attention 결과 비교

Google NIC	a white plate topped with a variety of food.	a baby is eating a piece of paper.	a close up of a plate of food on a table.	a teddy bear sitting on top of a chair .	a person is holding colorful umbrella.	a woman is holding a cell phone in her hand .
Top-5 visual attributes	plate broccoli fries food french	teeth brushing toothbrush holding baby	cake table plate sitting birthday	teddy cat bear stuffed white	umbrella beach water sitting boat	woman bathroom her scissors man
ATT-FCN	a plate with a sandwich and french fries.	a baby with a toothbrush in its mouth.	a table topped with a cake with candles on it.	a white teddy bear sitting next to a stuffed animal .	a black umbrella sitting on top of a sandy beach .	a woman holding a pair of scissors in her hands .

5. Conclusion

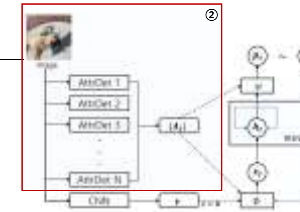
❖ 결론

- Image Captioning
  - ➢ CNN과 RNN으로 구성된 구조
  - ➢ 입력 변수 : 이미지
  - ➢ 출력 변수 : 문장

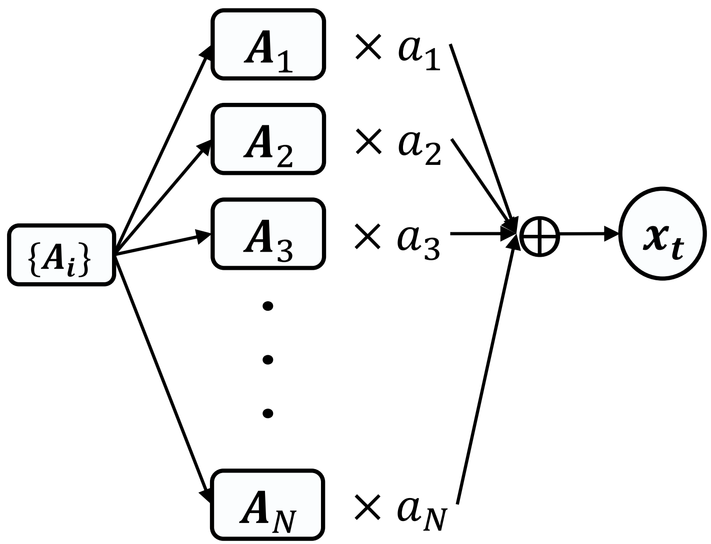
- Image Captioning with Semantic Attention
  - ➢ CNN과 RNN으로 구성된 구조의 RNN에 이미지의 특성을 반영
  - ➢ 입력 변수 : 이미지
  - ➢ 출력 변수 : 문장

감사합니다.

Appendix



❖ Semantic Attention



A_i : i 번째 해당하는 단어

y^i : A_i의 One Hot Vector

Y_{t-1} : $t-1$번째 단어

y_{t-1} : Y_{t-1}의 One Hot Vector

$|\mathcal{Y}|$: 현존하는 모든 단어 수

$\tilde{U} \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$

$E \in \mathbb{R}^{d \times |\mathcal{Y}|}$, Embedding matrix

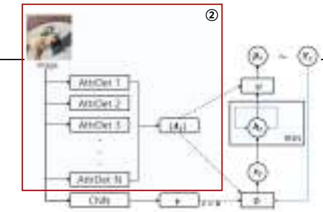
$$\alpha_t^i \propto exp\left(y_{t-1}{}^T \tilde{U} y^i\right) \longrightarrow \text{고차원}$$

Embedding

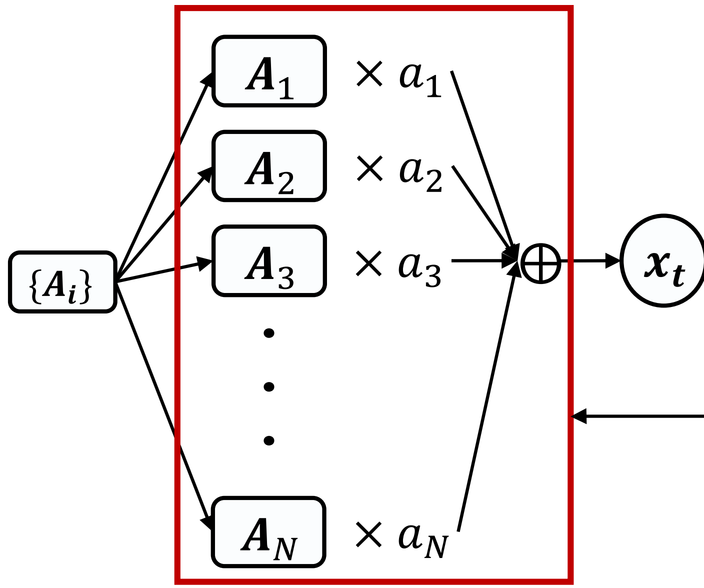
$$\alpha_t^i \propto exp\left((Ey_{t-1})^T U E y^i\right)$$

Data Mining Quality Analytics

Appendix



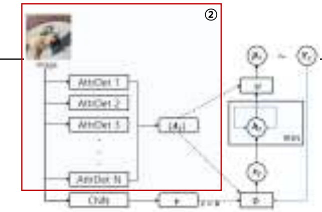
❖ Semantic Attention



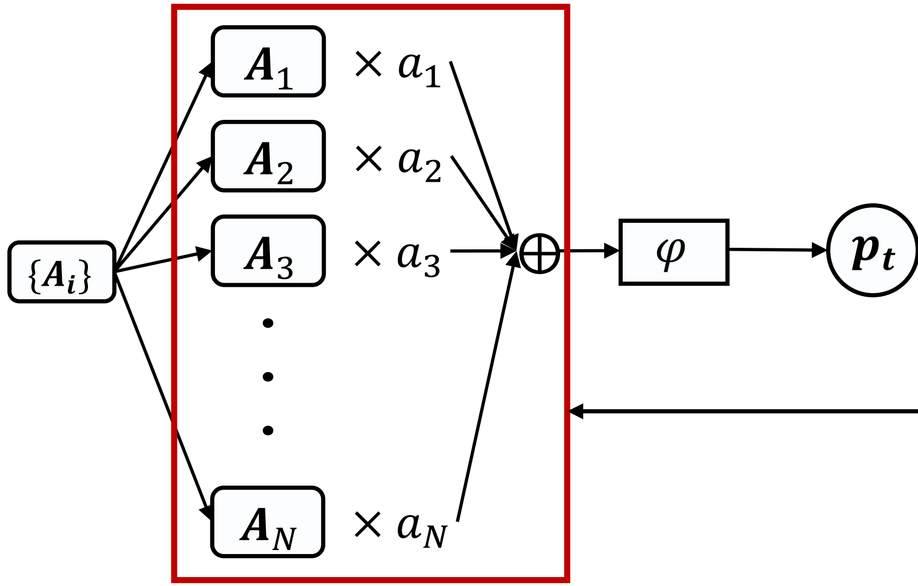
$$\alpha_t^i \propto exp\left((Ey_{t-1})^T UEy^i\right) = \exp(y_{t-1}^T E^T UEy^i),\ \widetilde{U} = E^T UE$$

$$x_t = W^{x,Y}(Ey_{t-1} + diag(w^{x,A}) \boxed{\sum_j \alpha_t^i Ey^i})$$

Appendix



❖ Semantic Attention



h_t^i : t state의 i 번째 hidden vector

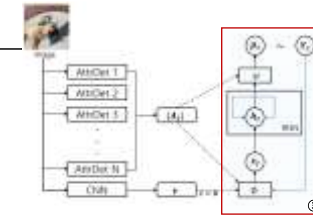
$\boldsymbol{E} \in \mathbb{R}^{d \times |\mathcal{Y}|}$

$\boldsymbol{V} \in \mathbb{R}^{n \times d}$

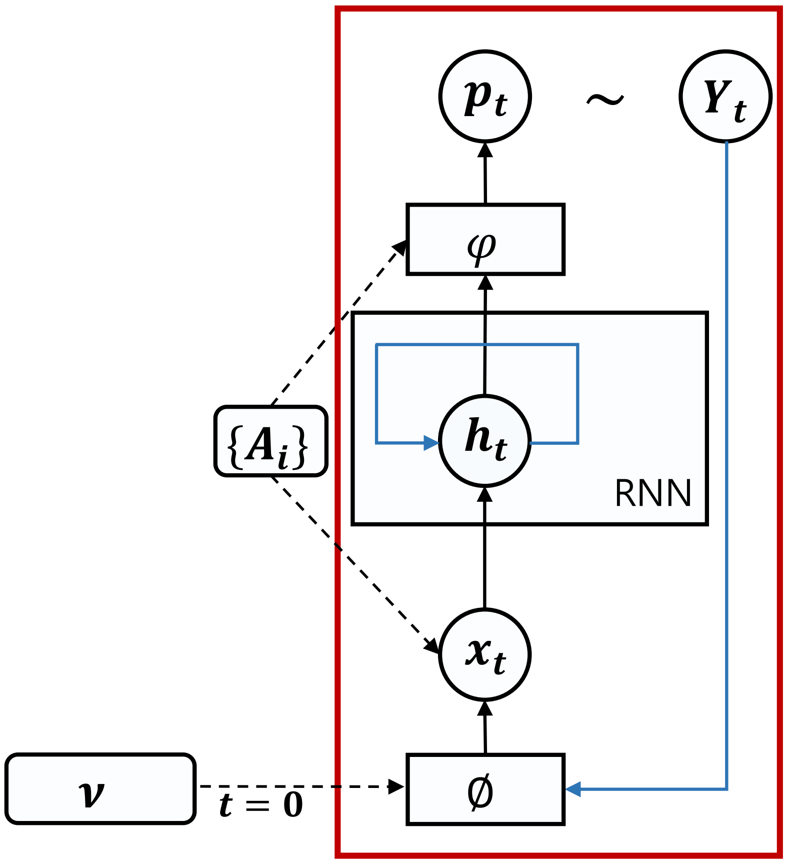
$$\beta_t^i \propto exp(h_t^i V \sigma(Ey^i))$$

$$p_t \propto exp(E^T W^{Y,h}(h_t + diag(w^{Y,A}) \boxed{\sum_i \beta_t^i \sigma(Ey^i)}))$$

Appendix



❖ RNN



Input : $x_0 = \emptyset_0(v) = W^{x,v}v, \ t = 0$

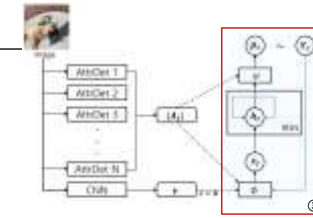
Calculation : $h_0 =$ 초기 hidden state vector

$Y_0 \sim p_0 = \varphi(h_0, \{A_i\})$

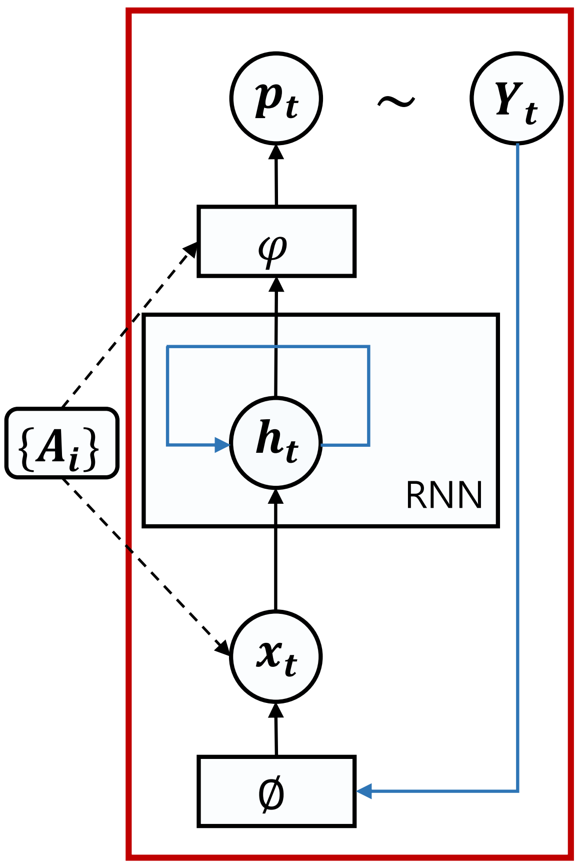
Output : $p_0 \sim Y_0$

- p_t : 단어 Y_t 가 나올 확률 벡터
- A_i : i 번째 단어
- h_t : State t의 hidden state vector

Appendix



❖ RNN



Input : $x_t = \emptyset(Y_{t-1}, \{A_i\}),\ t > 0$

Calculation : $h_t = RNN(h_{t-1,} x_t)$

$$Y_t \sim p_t = \varphi(h_t, \{A_i\})$$

Output : p_t

- p_t : 단어 Y_t 가 나올 확률 벡터
- A_i : i 번째 단어
- h_t : State t의 hidden state vector