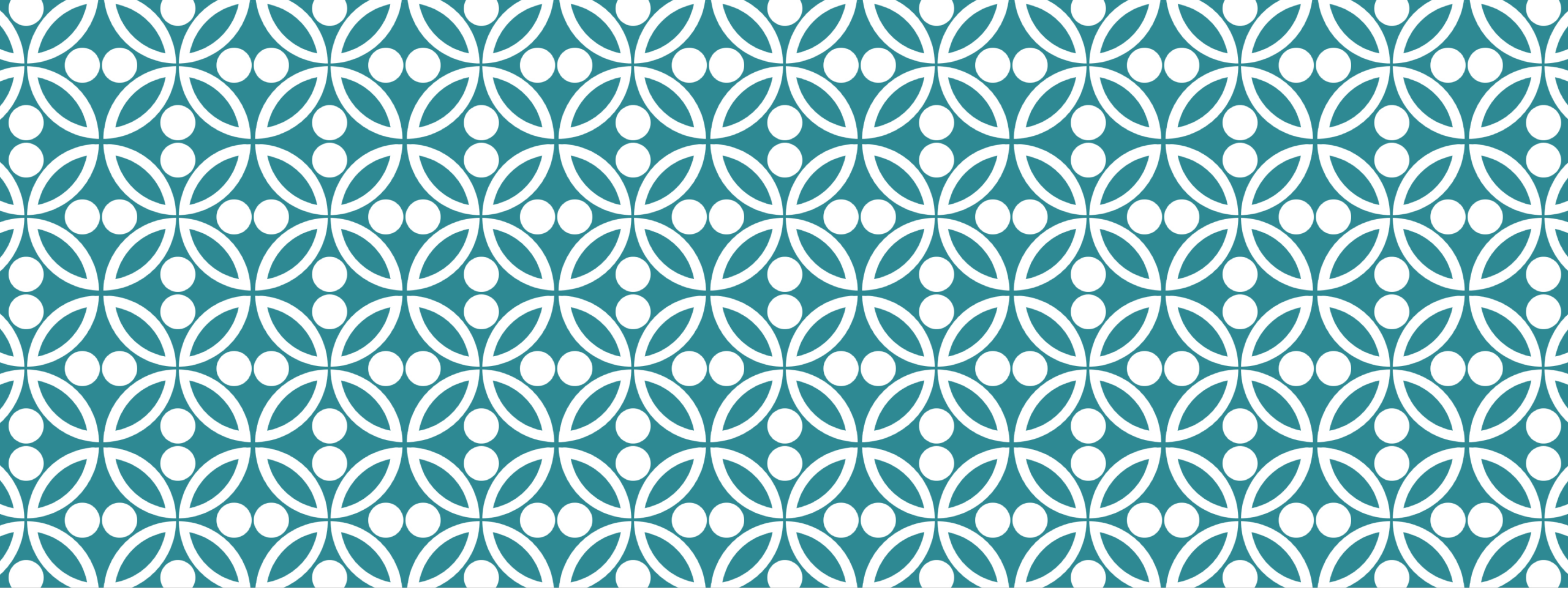
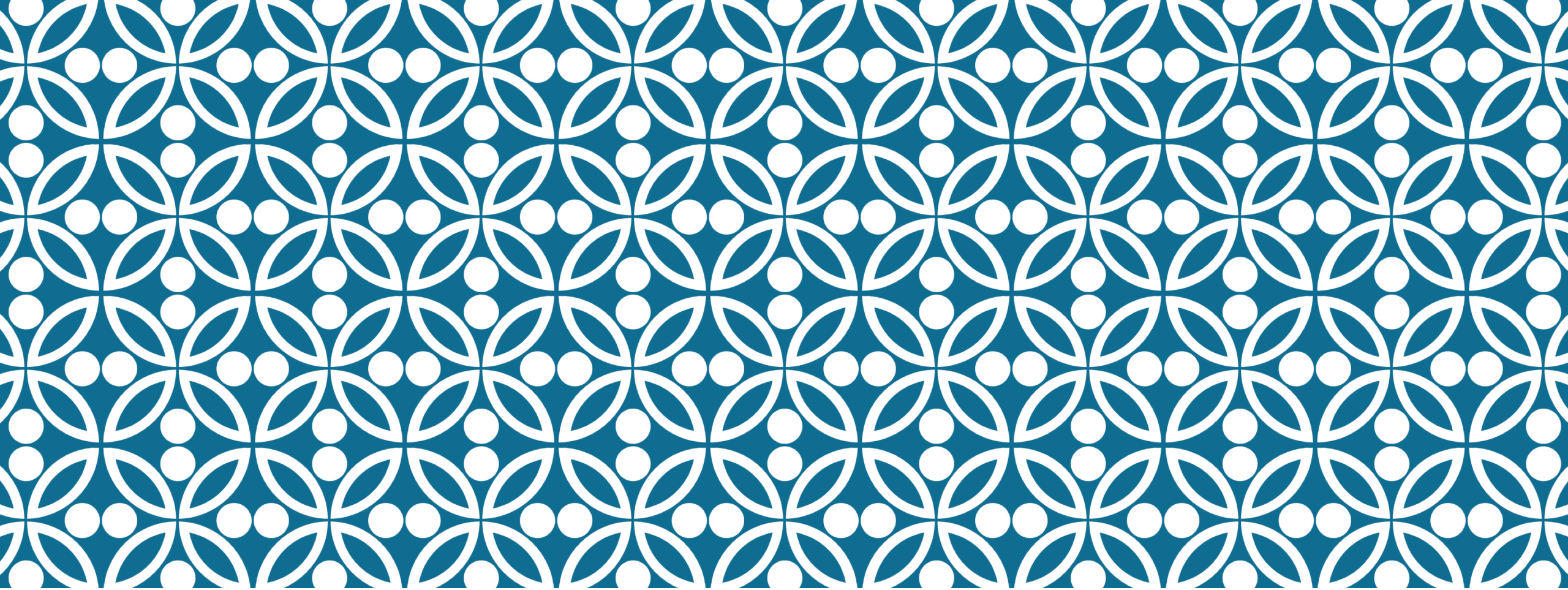


# DATA SCIENCE 2020 MIDTERM REVIEW

Speaker: Hong-Han Shuai



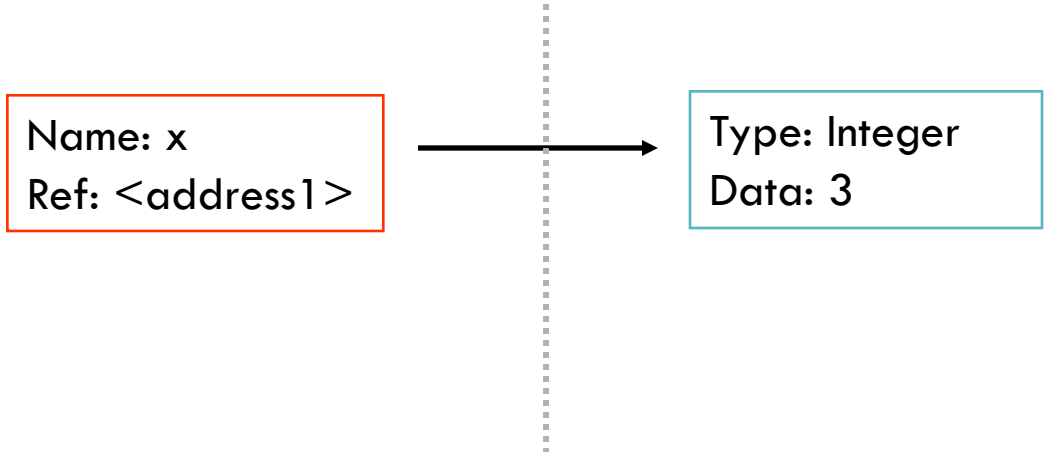
# LECTURE 1: INTRODUCTION TO CRAWLER

Speaker: Hong-Han Shuai
ECE, NCTU

# ASSIGNMENT

So, for simple built-in datatypes (integers, floats, strings), assignment behaves as you would expect:

```
>>> x = 3          # Creates 3, name x refers to 3
>>> y = x          # Creates name y, refers to 3.
>>> y = 4          # Creates ref for 4. Changes y.
>>> print x        # No effect on x, still ref 3.
3
```

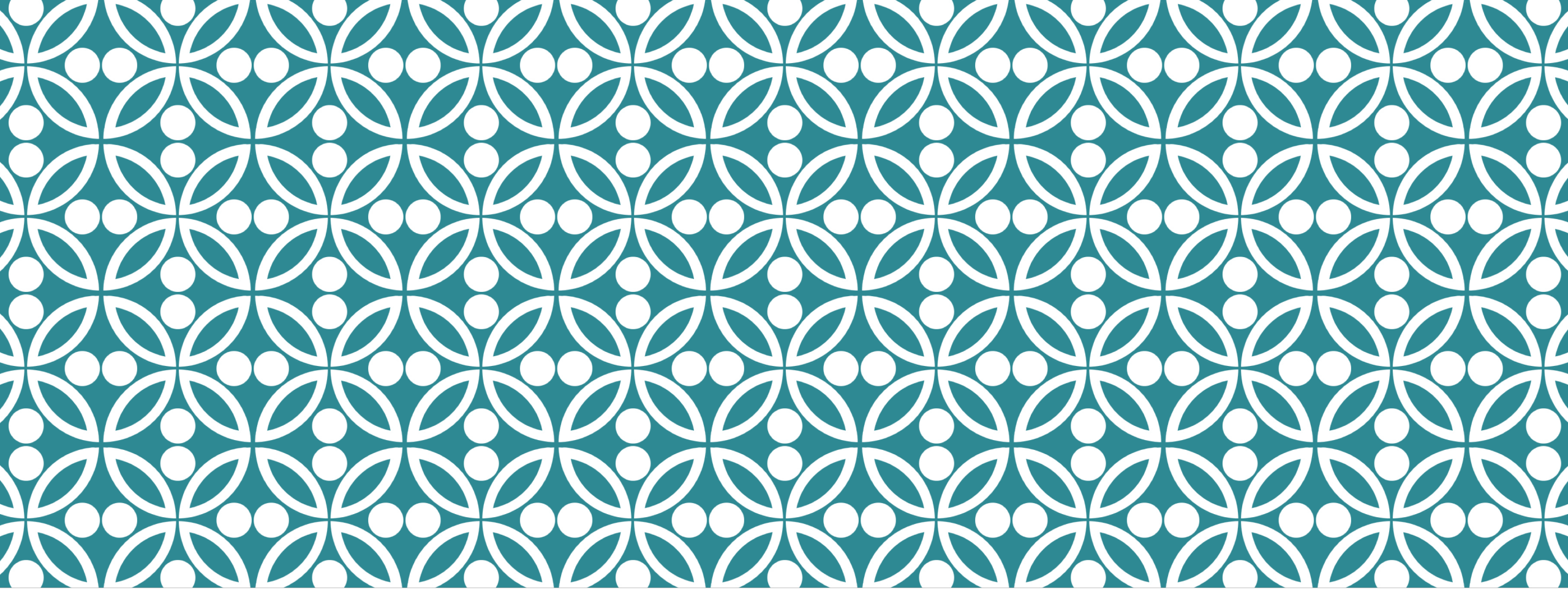


# PYTHON 的 RE PACKAGE

```
import re
pat = '[a-zA-Z]+'
text = 'Hello, hm...this is Tom speaking, who are you?'
r = re.findall(pat, text)
print r
```

```
['Hello', 'hm', 'this', 'is', 'Tom', 'speaking', 'who', 'are', 'you']
```

| 字元 | 描述 |
|---|---|
| [xyz] | 字元集合。匹配所包含的任意一個字元 |
| [^xyz] | 不含字元集合。 |
| [a-z] | 字元範圍。 |
| + | 匹配前面的子運算式一次或多次 |
| * | 匹配前面的子運算式零次或多次 |



# LECTURE 2:
# NATURAL LANGUAGE PROCESSING (I): WORD REPRESENTATION

Data Science, Fall 2020
Hong-Han Shuai

# HIGH DIMENSION EXAMPLE

[1 0 0 ……..]

[0 1 0 ……..]

[1 0 0 ……..]

# CURSE (CON'T)

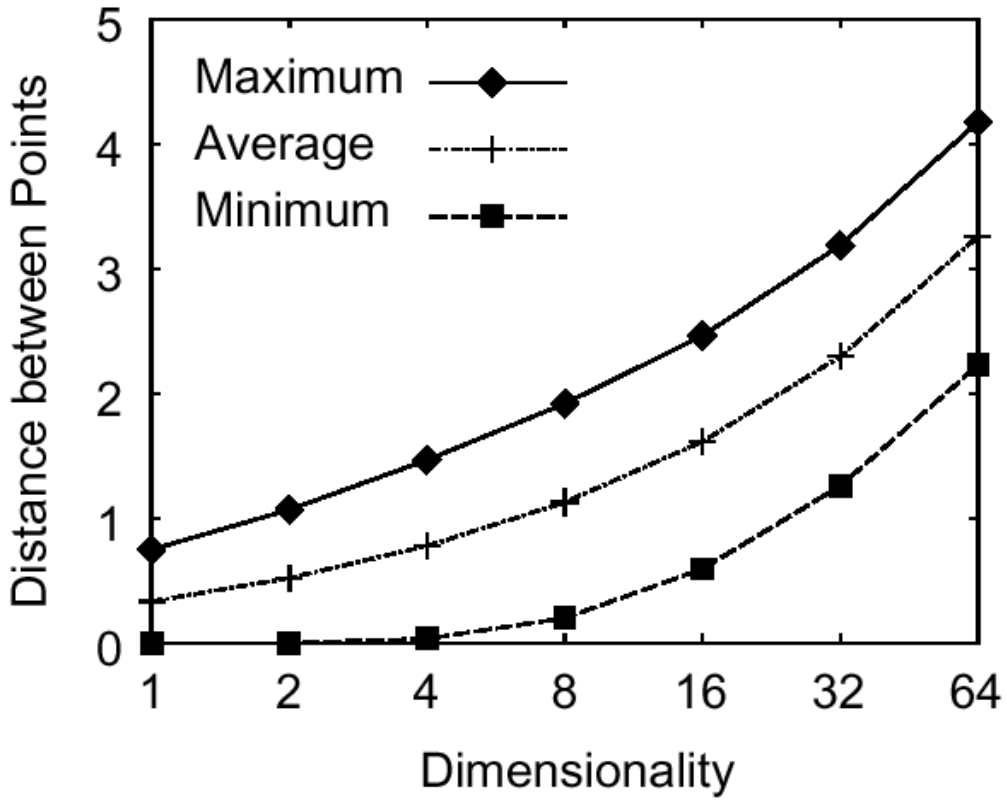
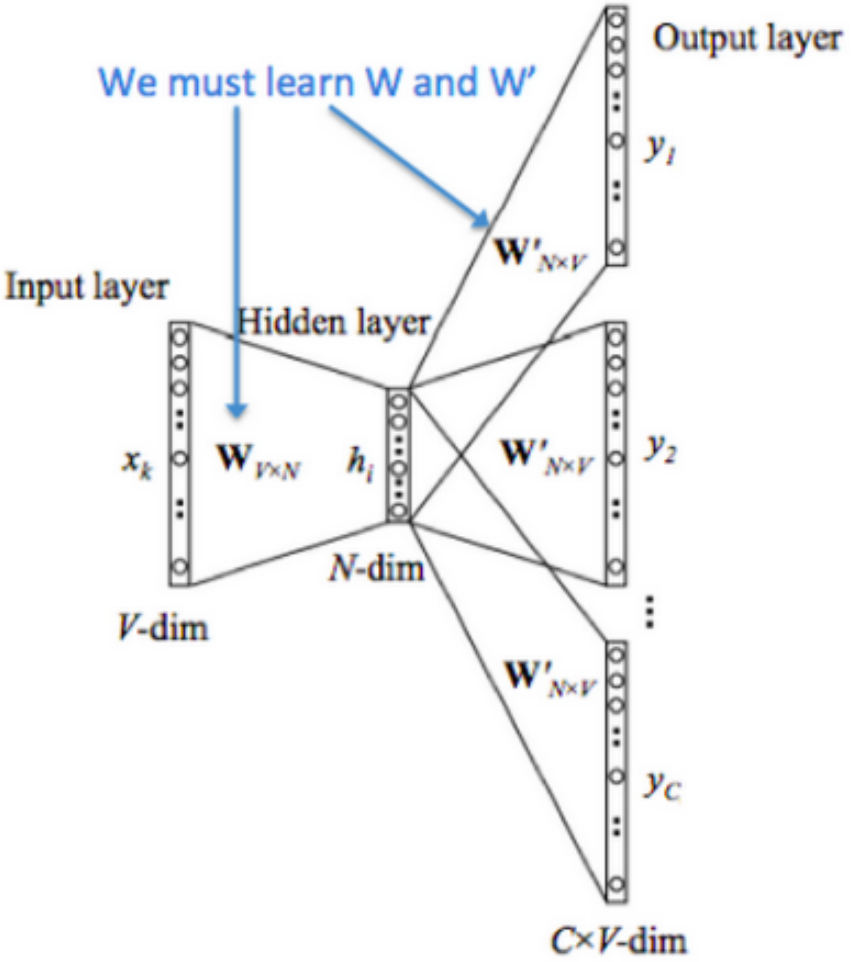



**Figure 1. Distances among 100k points generated at random in a unit hypercube**

Source: N. Katayama, S. Satoh. Distinctiveness Sensitive Nearest Neighbor Search for Efficient Similarity Retrieval of Multimedia Information. ICDE Conference, 2001.

# TAKE **SKIP-GRAM MODEL AS AN EXAMPLE**

$$\arg\max_{\theta} \prod_{w \in Text} \left[ \prod_{c \in C(w)} p(c|w;\theta) \right]$$



# NEGATIVE SAMPLING

Suppose we know that some words won't be put together.

- [Trump, NBA]
- [Trump, Vitamin C]
- [Trump, Washing Machine]

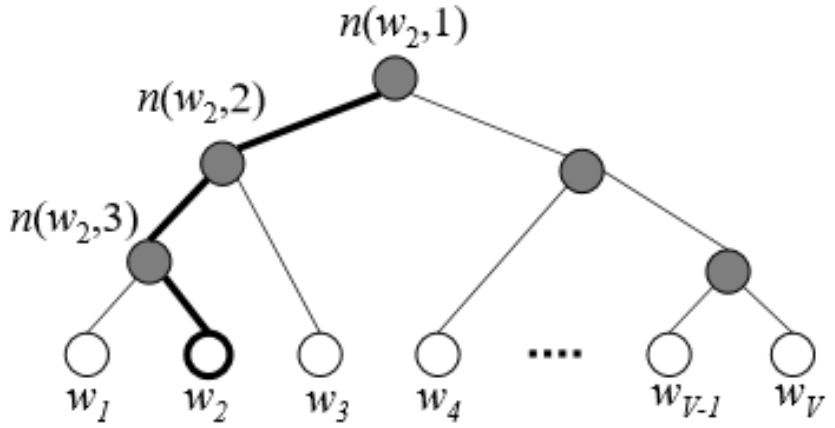
$$\arg\max_{\theta} \sum_{(w,c)\in D} \log \frac{1}{1+e^{-v_c \cdot v_w}} + \sum_{(w,c)\in D'} \log\left(\frac{1}{1+e^{v_c \cdot v_w}}\right)$$

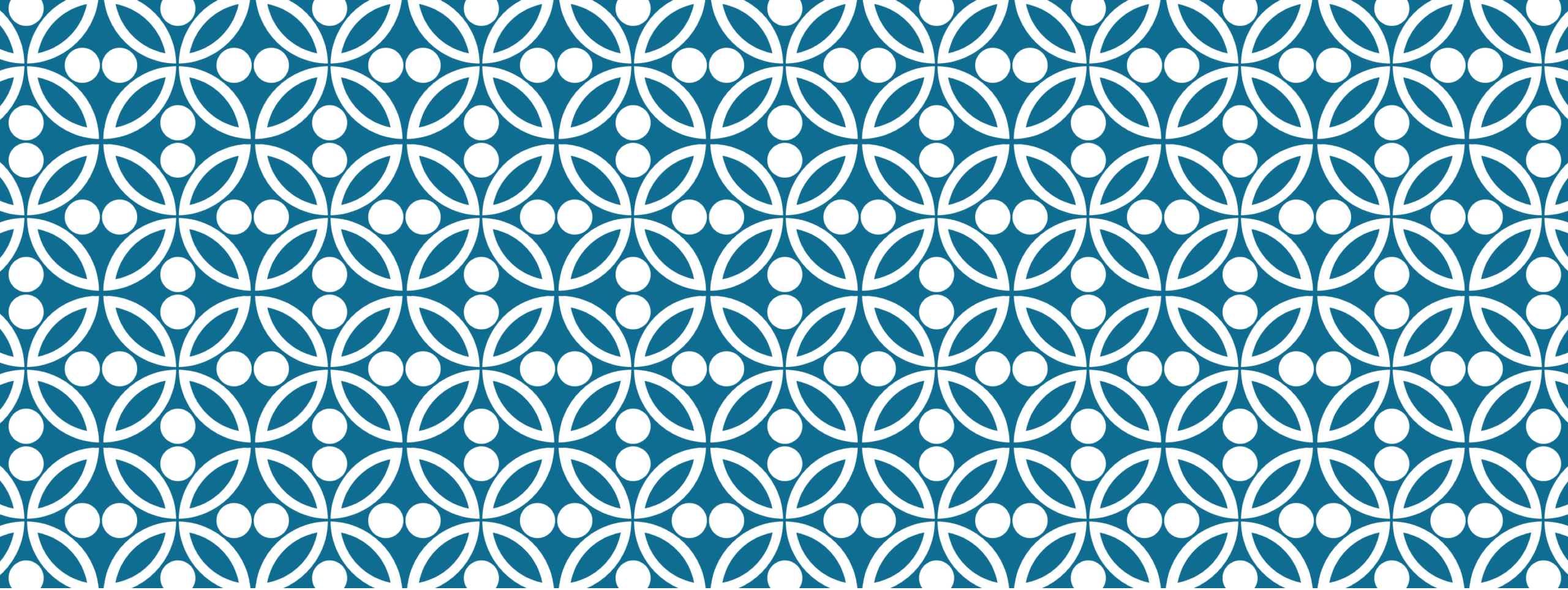
$$= \arg\max_{\theta} \sum_{(w,c)\in D} \log \sigma(v_c \cdot v_w) + \sum_{(w,c)\in D'} \log \sigma(-v_c \cdot v_w)$$

# HIERARCHICAL SOFTMAX (CBOW)

Key Idea: Modify the output layer so we can accelerate the training process.

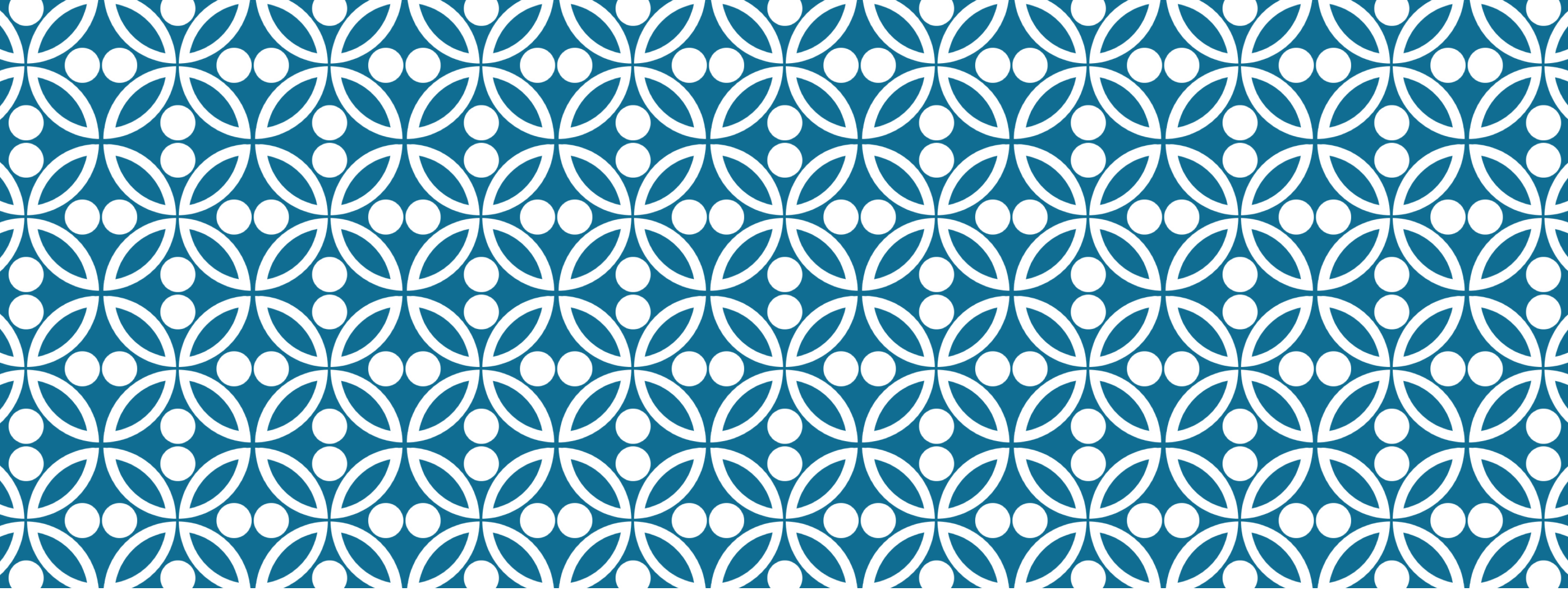
- Huffman coding
- O(n) -> O(logn)





# ADAPTIVE COMPRESSION OF WORD EMBEDDINGS

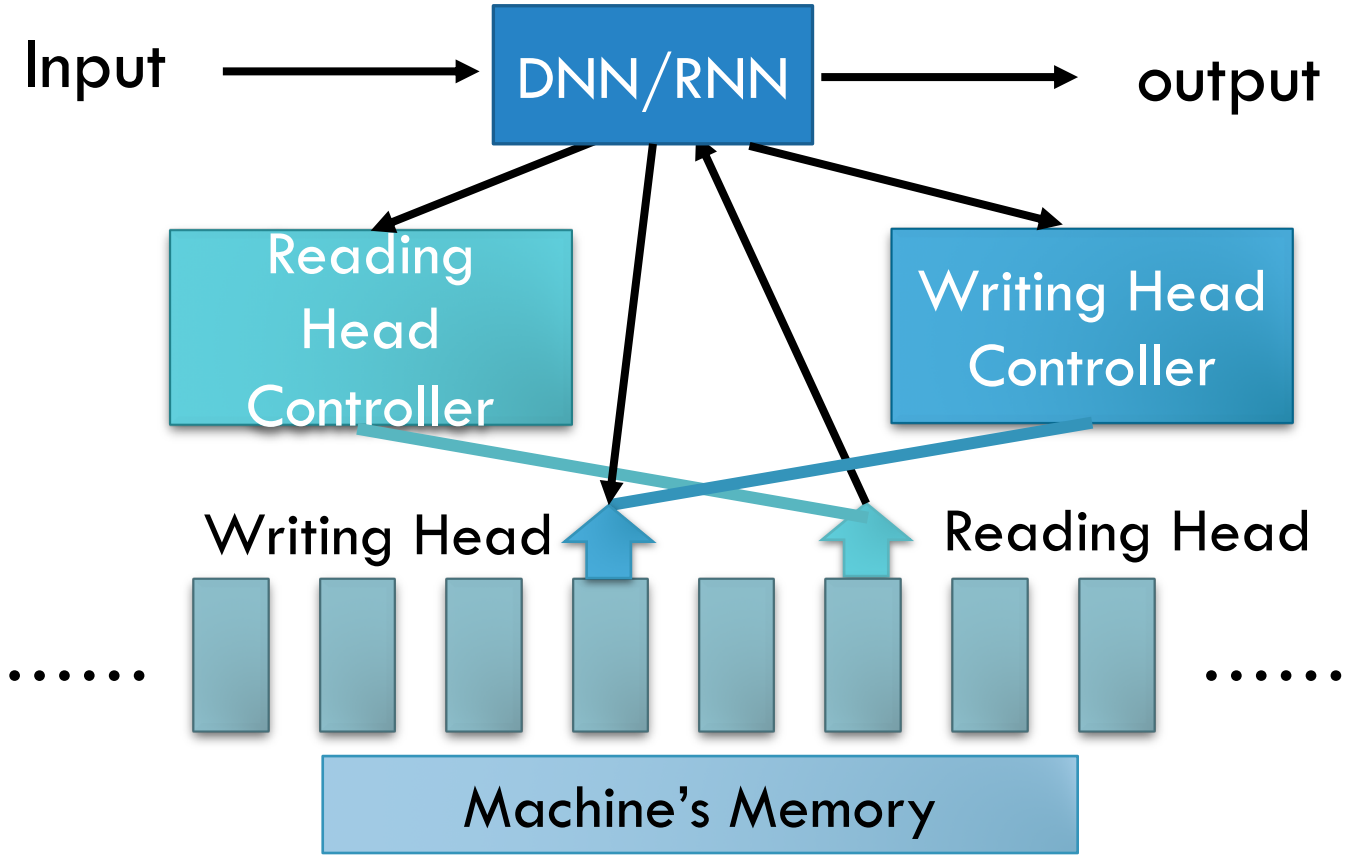
ACL'20



# DOUBLE-HARD DEBIAS: TAILORING WORD EMBEDDINGS FOR GENDER BIAS MITIGATION

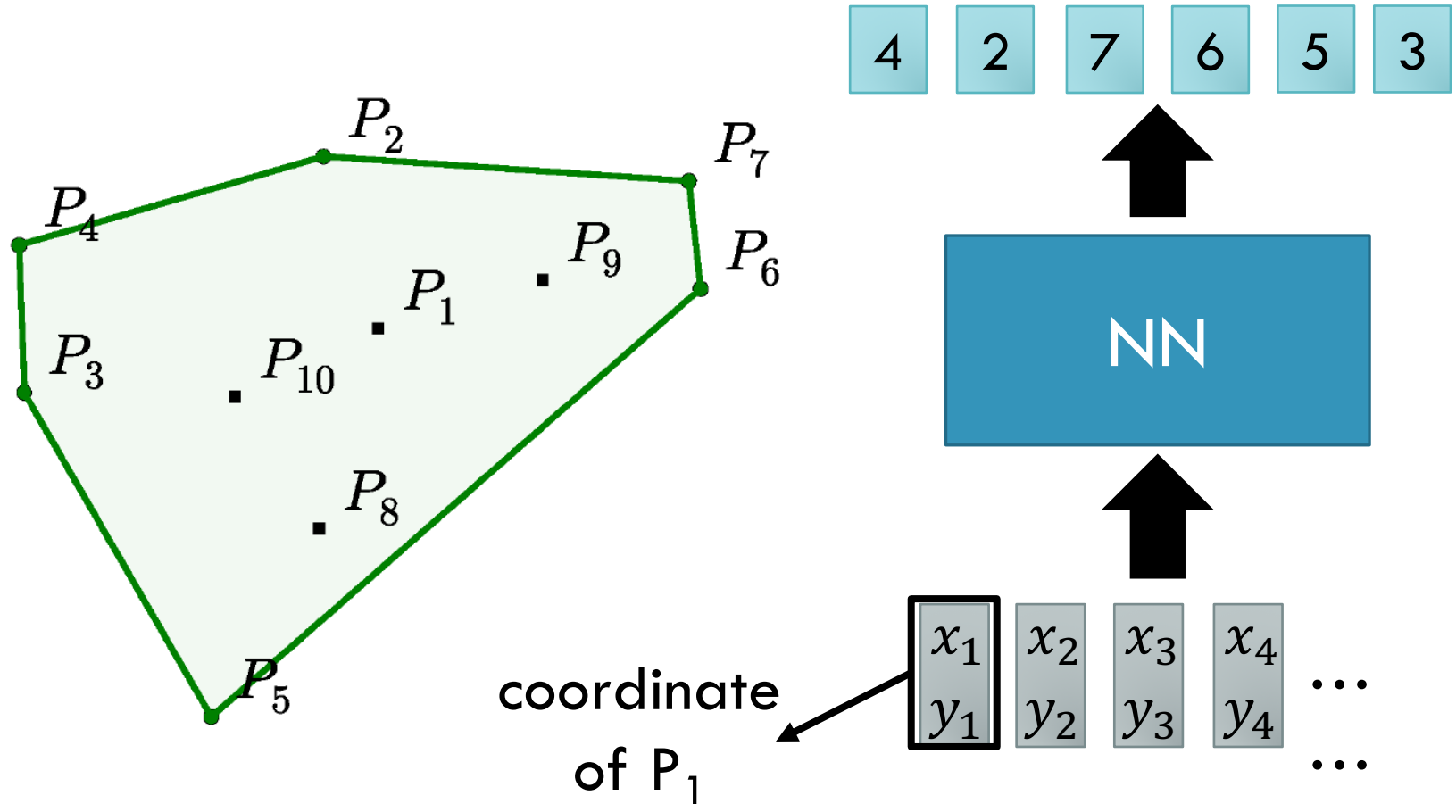
ACL'20

# EXTERNAL MEMORY V2




Neural Turing Machine

# POINTER NETWORK




planar convex hulls, computing Delaunay triangulations, and the planar Travelling Salesman Problem

# *Self-attention*

https://arxiv.org/abs/1706.03762


$q$: query (to match others)

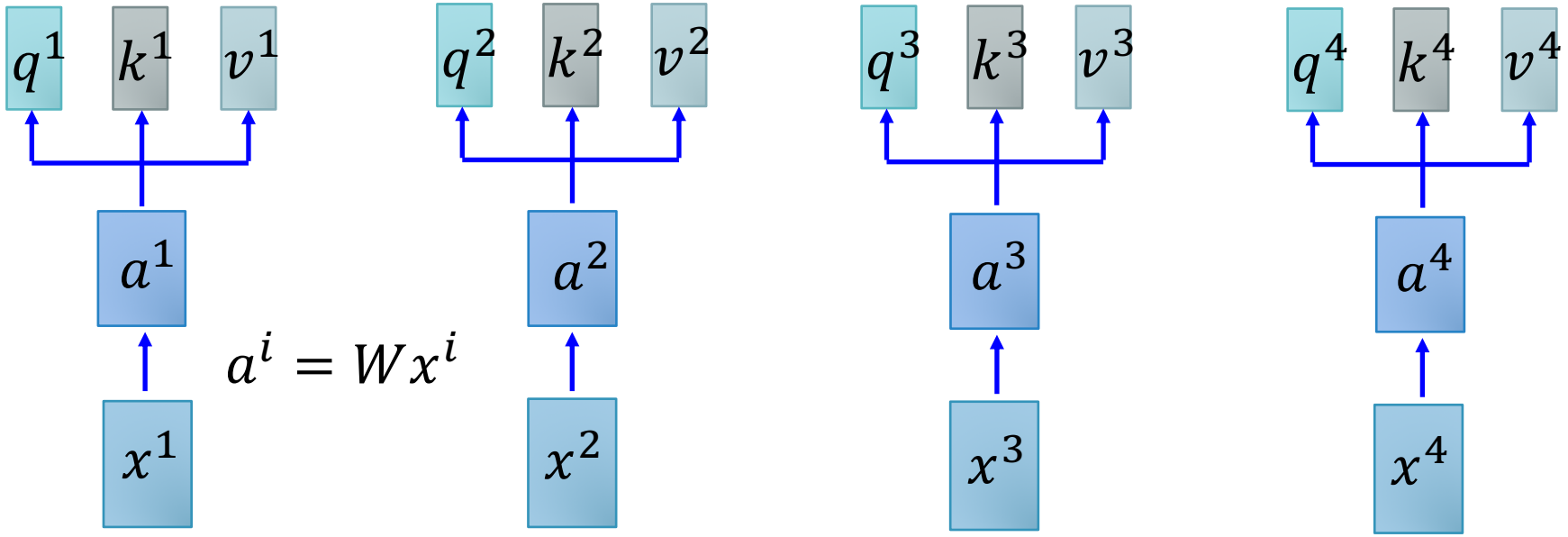
$$q^i = W^q a^i$$

$k$: key (to be matched)

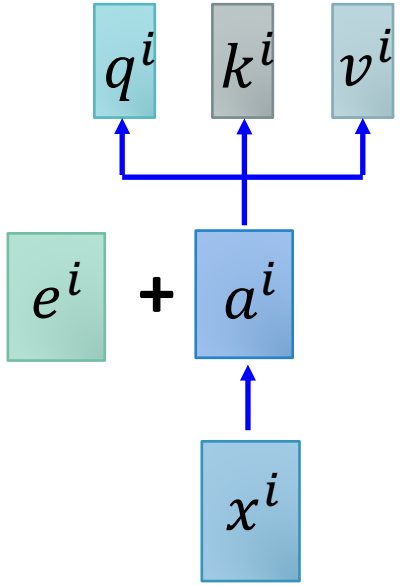
$$k^i = W^k a^i$$

$v$: information to be extracted

$$v^i = W^v a^i$$



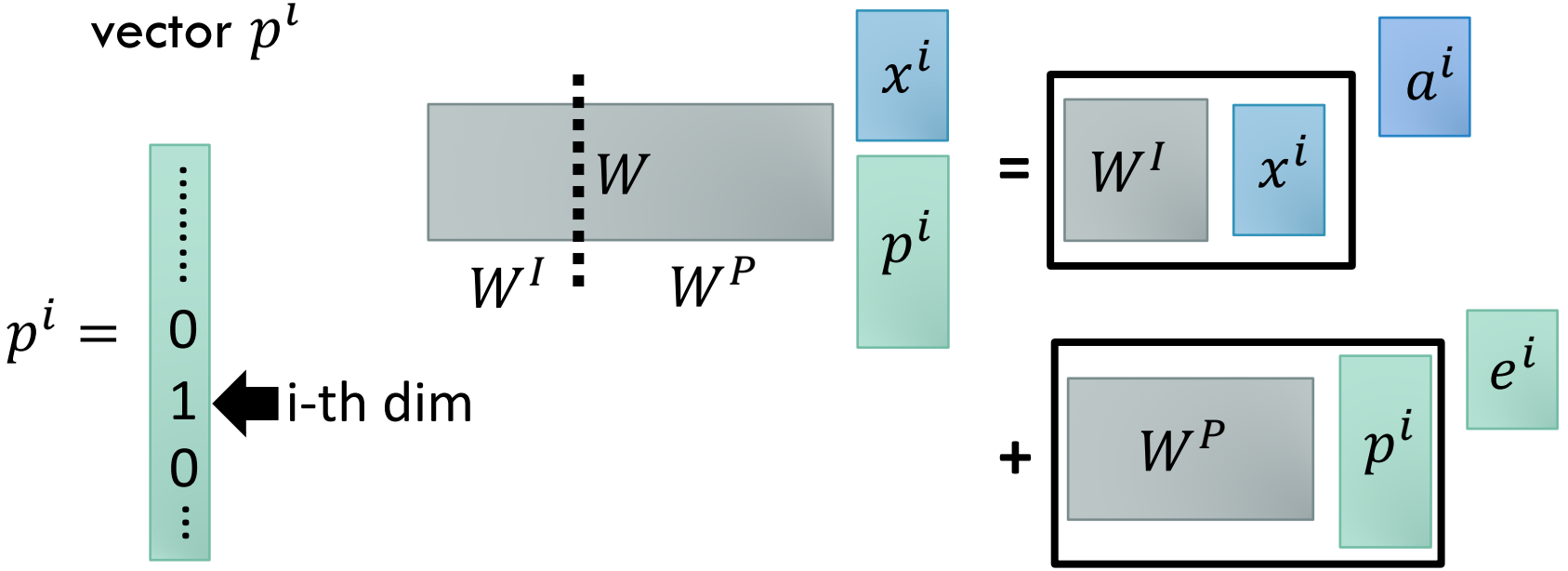
# POSITIONAL ENCODING

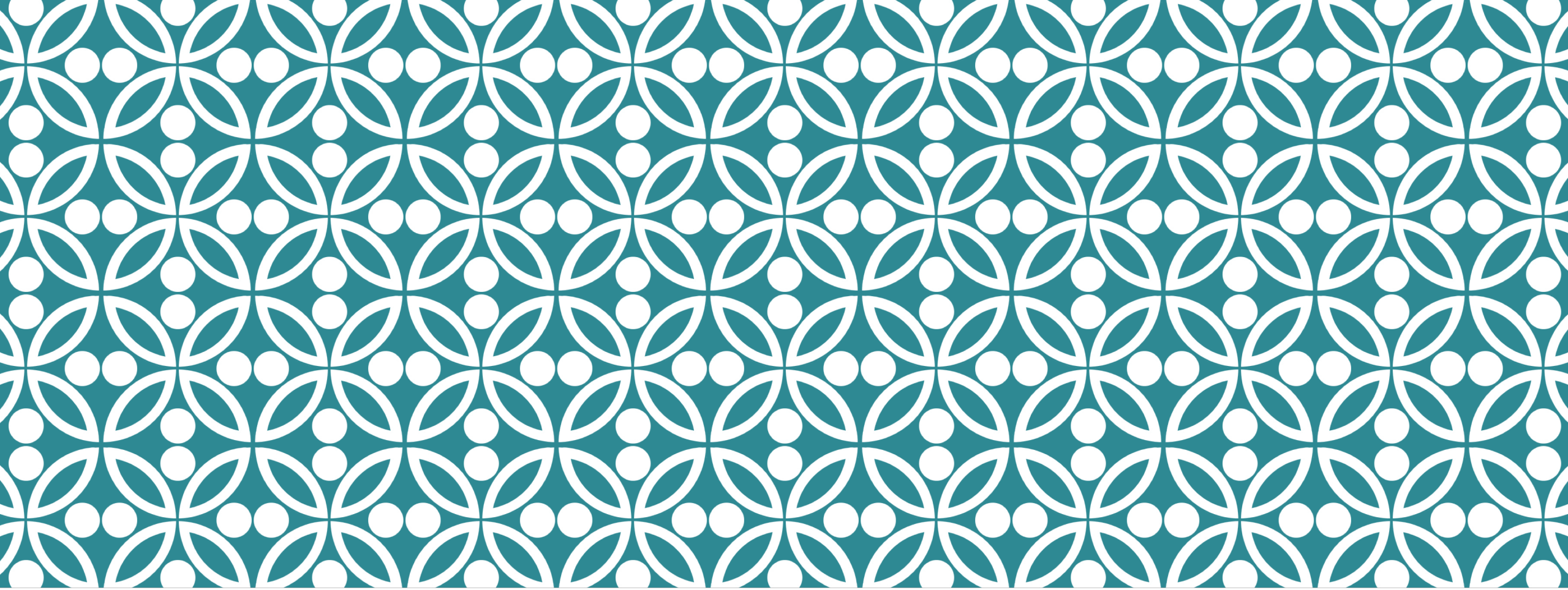



No position information in self-attention.

Original paper: each position has a unique positional vector $e^i$ (not learned from data)

In other words: each $x^i$ appends a one-hot vector $p^i$





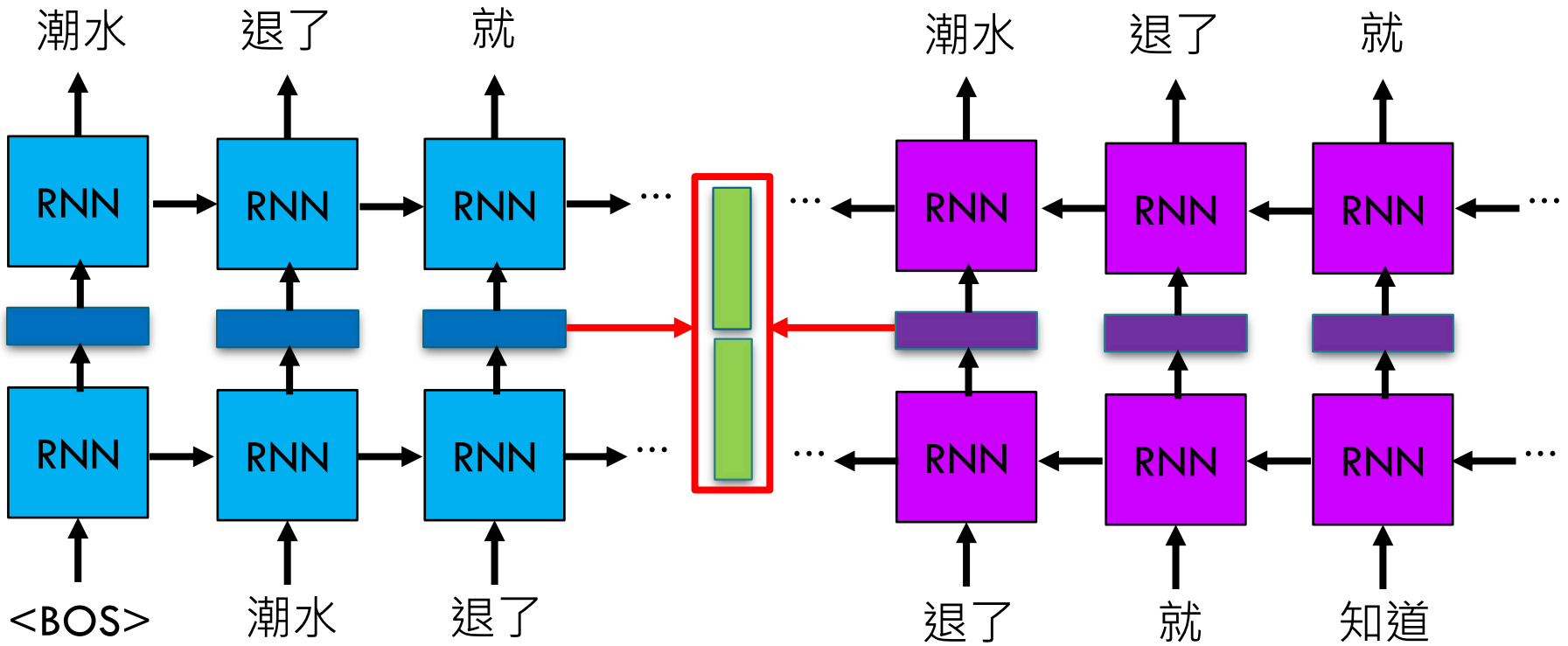
# ELMO, BERT, GPT, XLNET, AND ELECTRA

# EMBEDDINGS FROM LANGUAGE MODEL (EL...)

https://arxiv.org/abs/1802.05365

RNN-based language models (trained from lots of sentences)

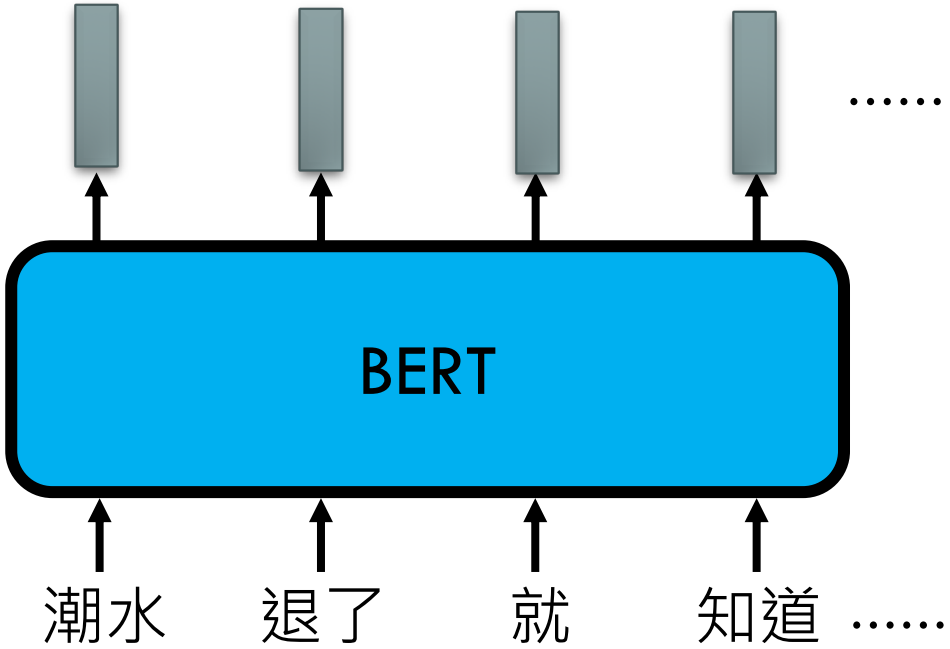
e.g. given "潮水 退了 就 知道 誰 沒穿 褲子"

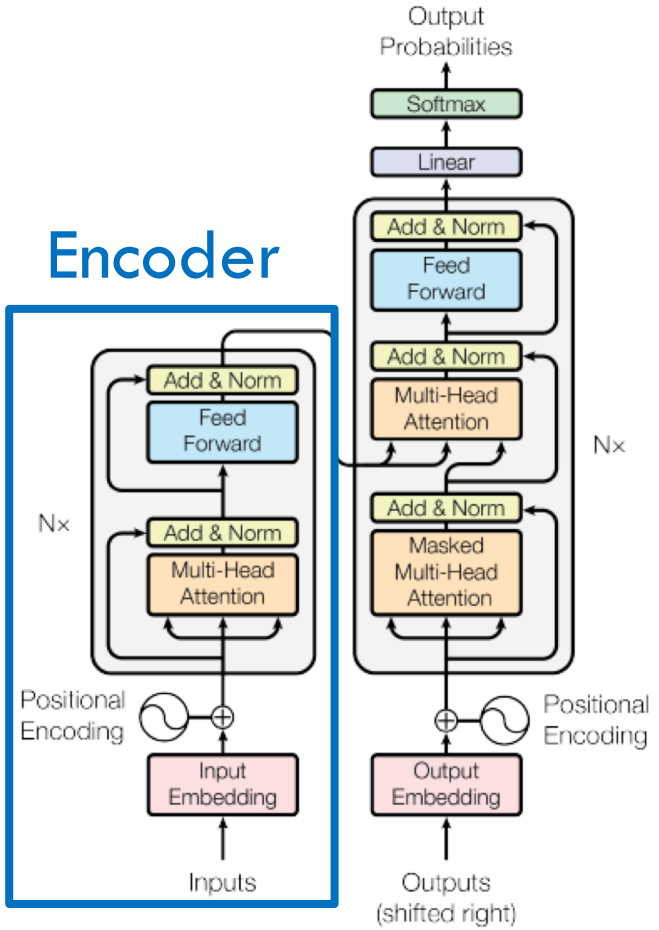


# BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT)

BERT = Encoder of Transformer

Learned from a large amount of text without annotation

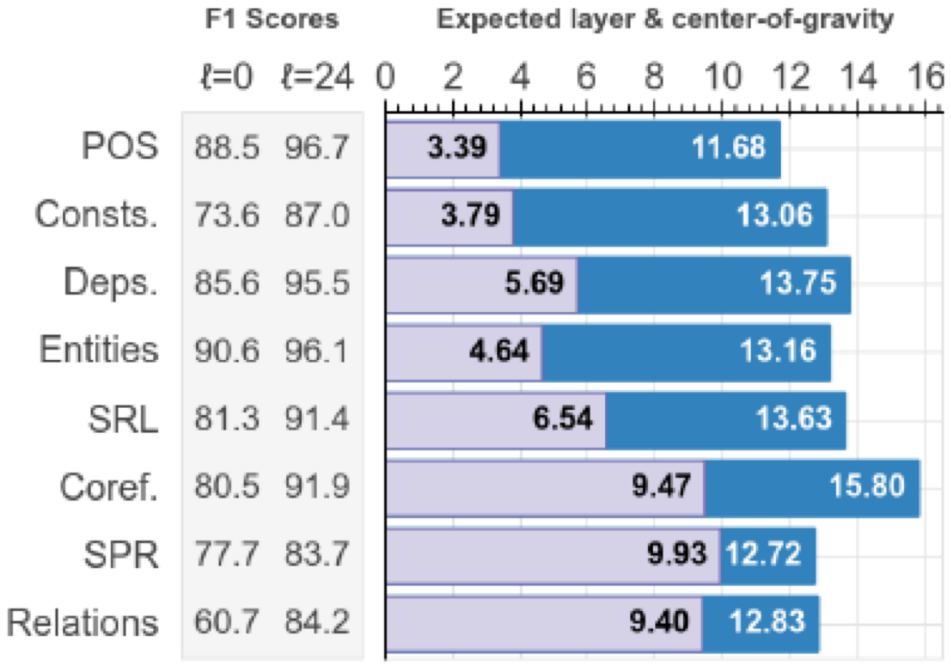


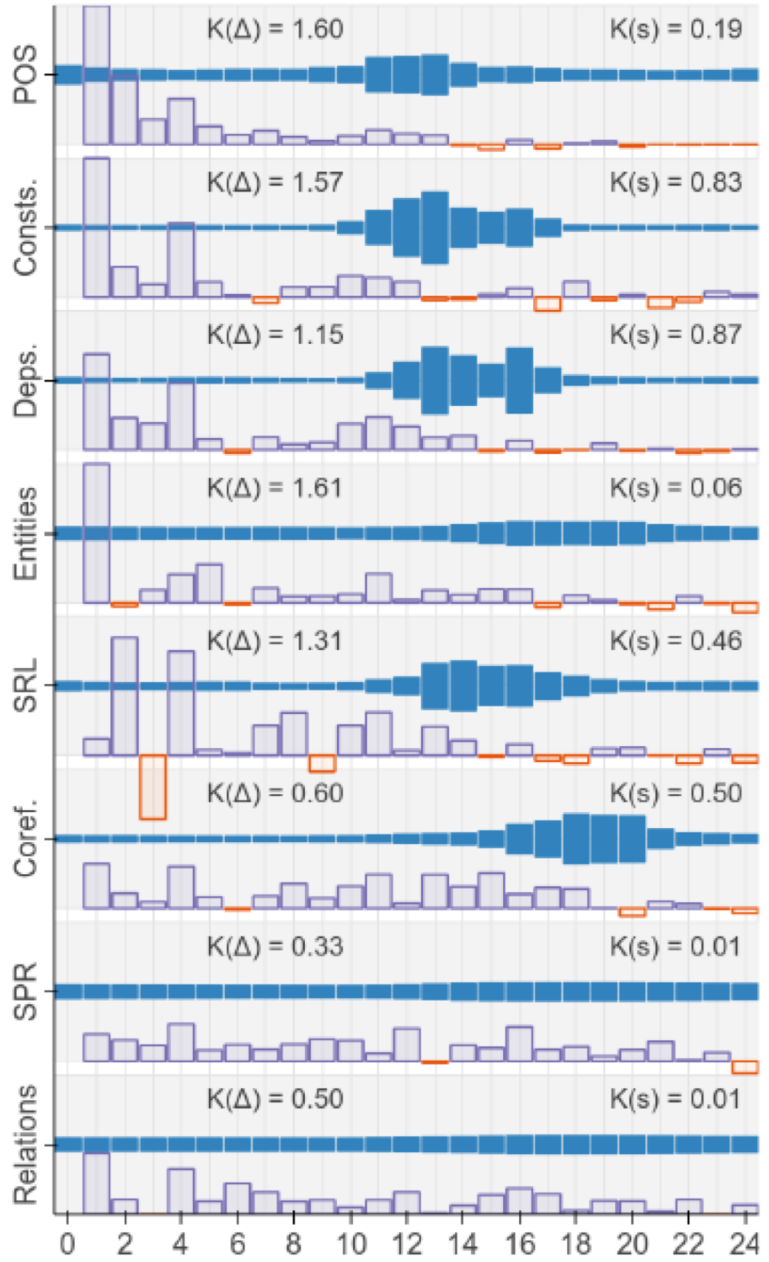



# WHAT DOES BERT LEARN?

**BERT Rediscovers the Classical NLP Pipeline**

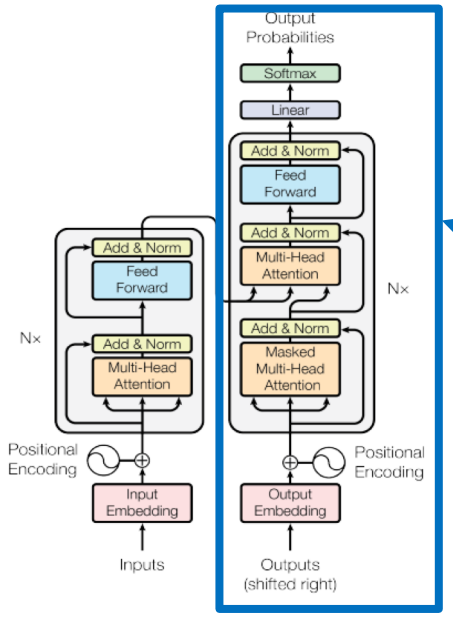
**Ian Tenney**[1] **Dipanjan Das**[1] **Ellie Pavlick**[1,2]
[1]Google Research [2]Brown University
{iftenney,dipanjand,epavlick}@google.com



https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

# GENERATIVE PRE-TRAINING (GPT)




Transformer Decoder

BERT (340M)

ELMO (94M)

GPT-2 (1542M)

Source of image: https://huaban.com/pins/1714071707/

# XLNET

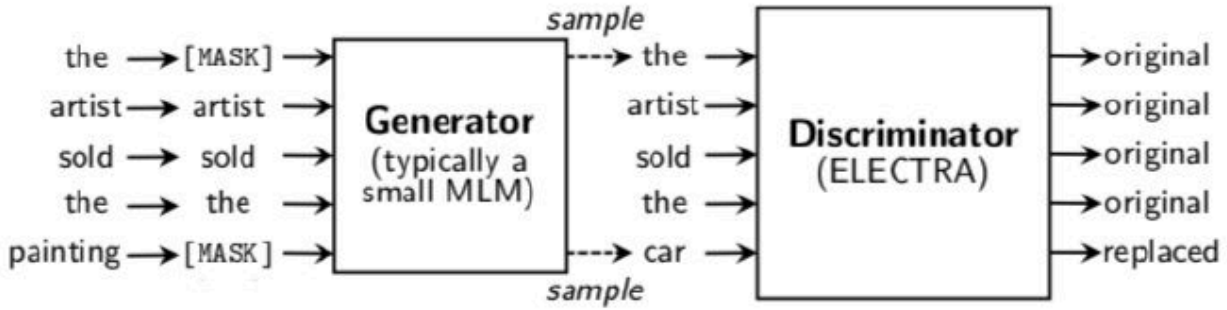
## XLNet: Generalized Autoregressive Pretraining for Language Understanding

**Zhilin Yang**[*1], **Zihang Dai**[*12], **Yiming Yang**[1], **Jaime Carbonell**[1],
**Ruslan Salakhutdinov**[1], **Quoc V. Le**[2]
[1]Carnegie Mellon University, [2]Google Brain
{zhiliny,dzihang,yiming,jgc,rsalakhu}@cs.cmu.edu, qvl@google.com

https://github.com/zihangdai/xlne

- ✓Predicting Y/N is easier than reconstruction
- ✓Every output position is used

# ELECTRA : Introduction




| | | |
|---|---|---|
| Generator | $p_G(x_t\|x) = \frac{exp(e(x_t)^\top h_G(x_t))}{\sum_i exp(e(x)_i^\top h_G(x_i)))}$ | $x_{masked} = REPLACE(x, m, [mask])$ |
| Discriminator | $p_D(x,t) = sigmoid(w^T h_D(x_t))$ | $x_{corrupt} = REPLACE(x, m, \bar{x})$ |

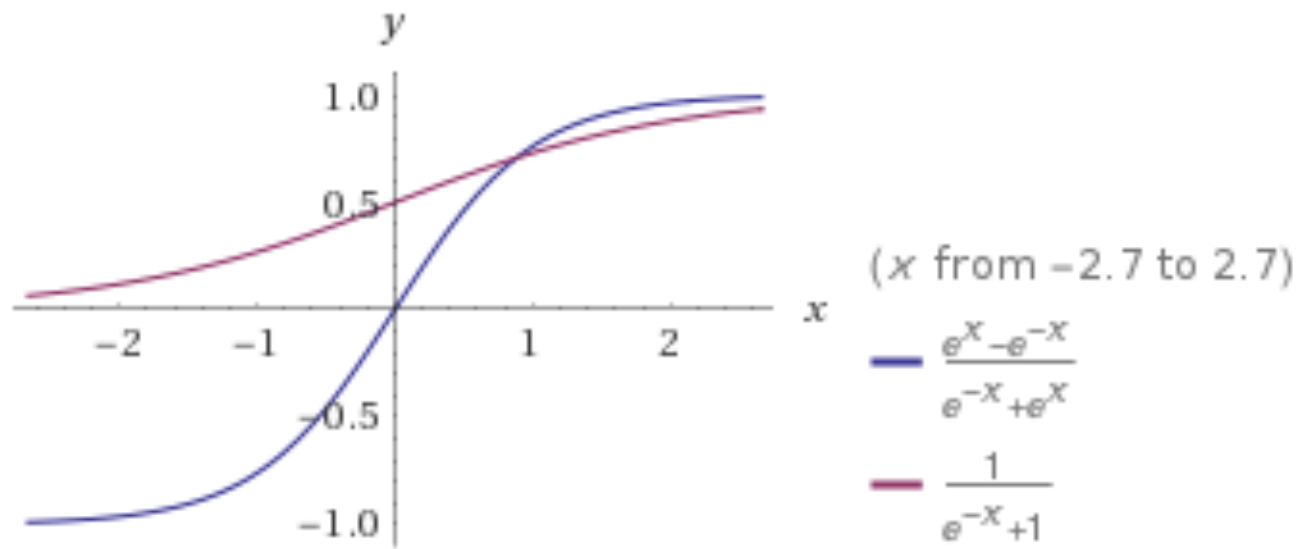
$$L_{MLM}(x, \theta_G) = \sum -log p_G(x_i|x^{masked}$$

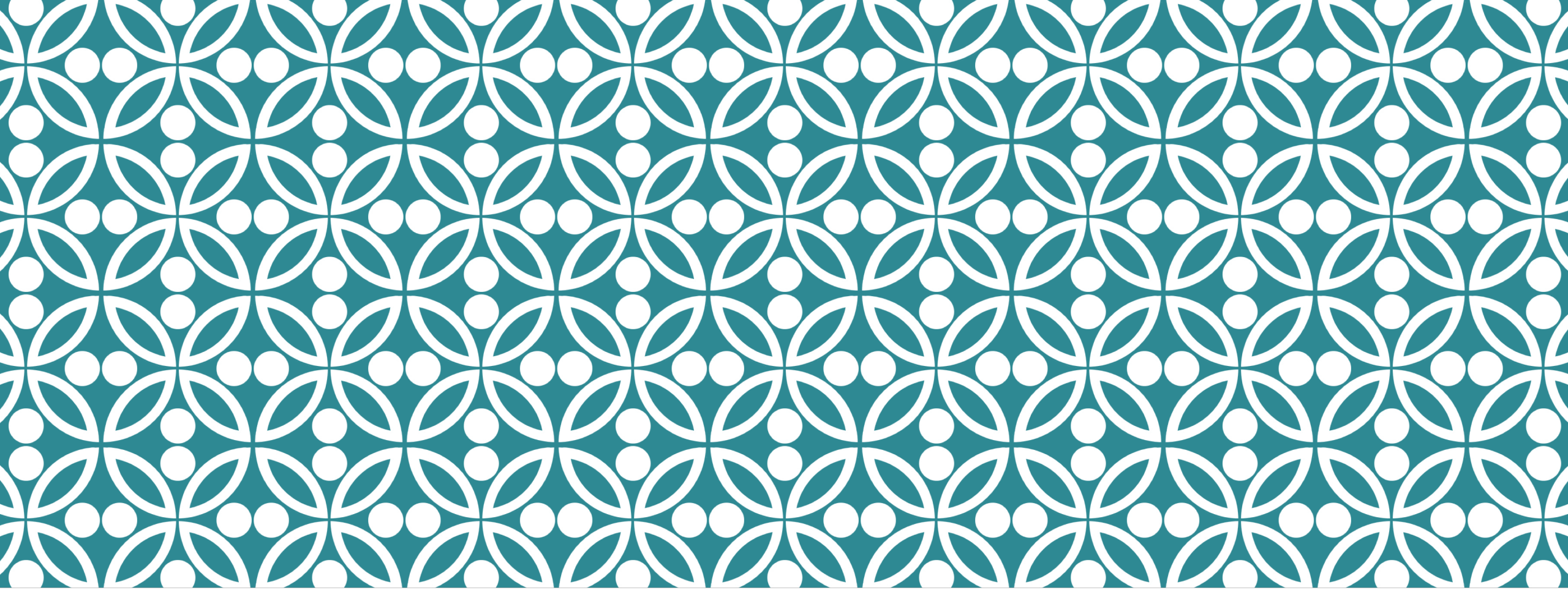
$$L_{Disc}(x, \theta_D) = \sum (x_t^{corrupt} = x_t) log D(x^{corrupt}, x_t) + (x_t^{corrupt} \neq x_t) log(1 - D(x^{corrupt}, x_t)$$

$$min_{\theta_G, \theta_D} \sum L_{MLM}(\theta_G) + \lambda L_{Disc}(\theta_D)$$

| Property | Description | Problems | Examples |
|---|---|---|---|
| derivative | $f'$ | $> 1$ exploding gradient (e)<br>$< 1$ vanishing (v) | sigmoid (v), tanh (v), cube (e) |
| zero-centered | range centered around zero? | if not, slower learning | tanh $(+)$, relu $(-)$ |
| saturating | finite limits | vanishing gradient in the limit | tanh, penalized tanh, sigmoid |
| monotonicity | $x > y \implies f(x) \geq f(y)$ | unclear | exceptions: sin, swish, minsin |

Table 2: Frequently cited properties of activation functions





# LECTURE 3: NATURAL LANGUAGE PROCESSING (III) SENTIMENT ANALYSIS AND RECOMMENDATION SYSTEM

Data Science, Fall 2020
Hong-Han Shuai

# A MORE PRACTICAL DEFINITION

## (HU AND LIU 2004; LIU, 2010, 2012)

An *opinion* is a quintuple

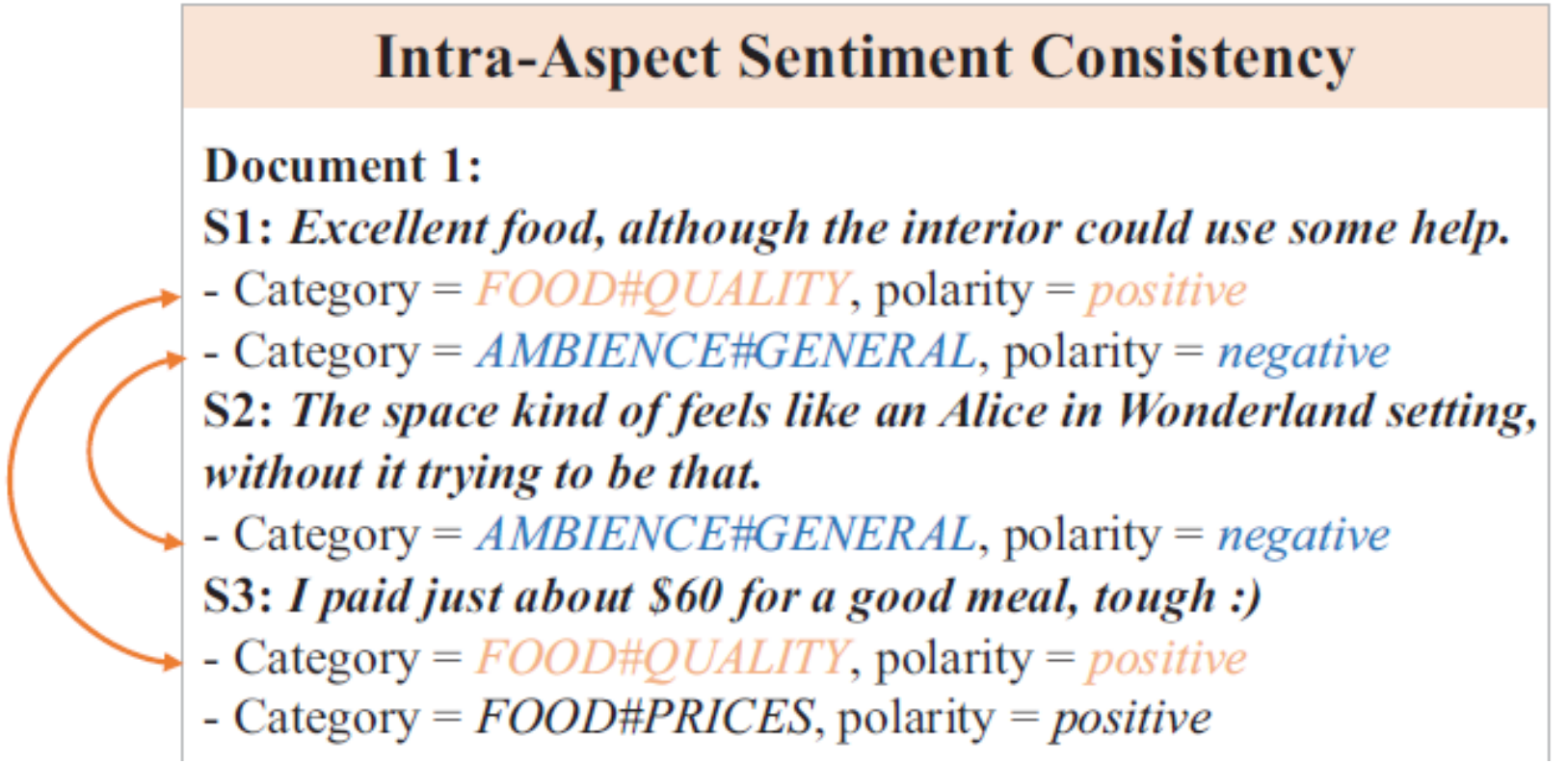
(*entity*, *aspect*, *sentiment*, *holder*, *time*)

where

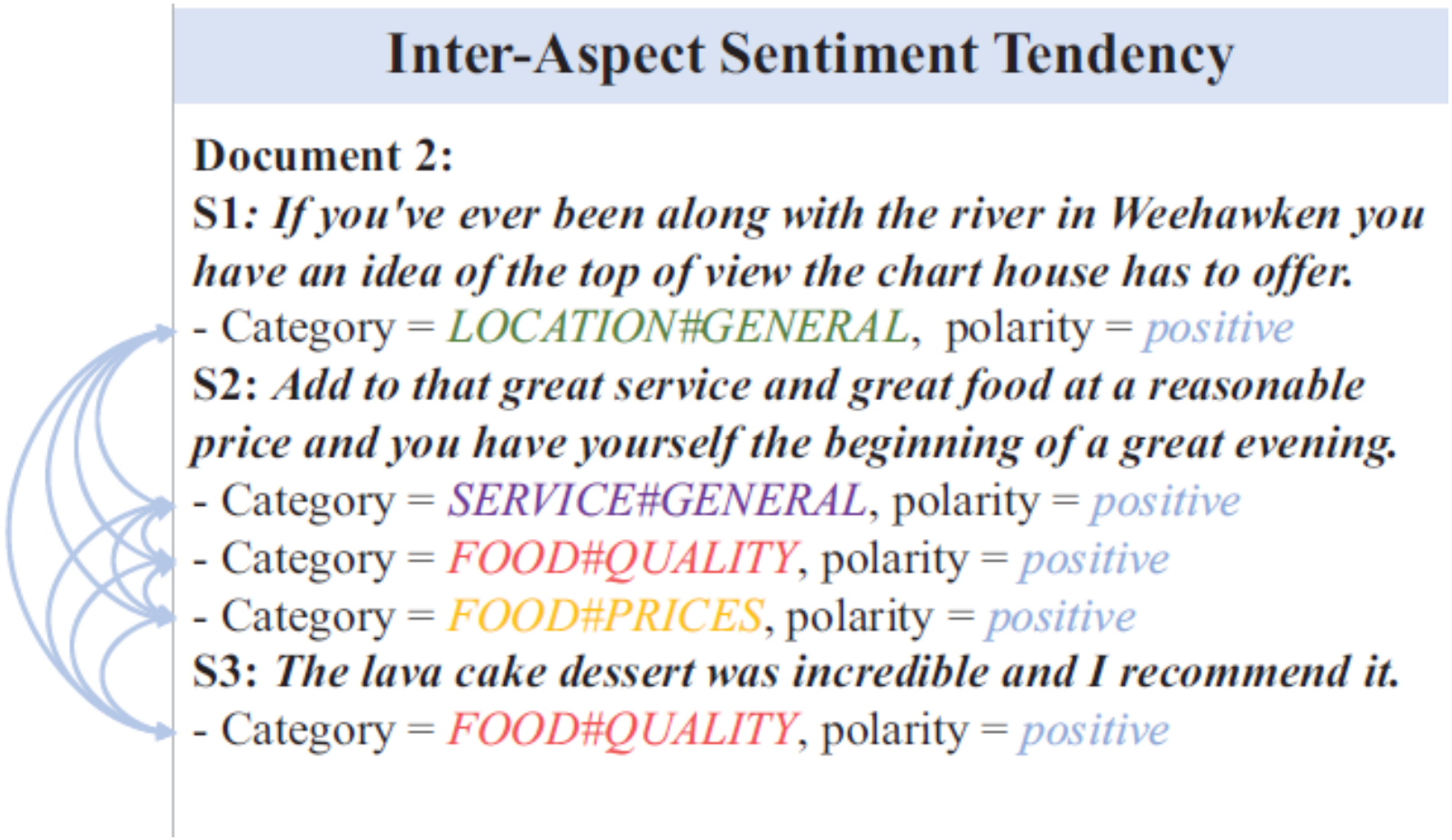
- ***entity***: target entity (or object).
- ***Aspect***: aspect (or feature) of the entity.
- ***Sentiment***: +, -, or neu, a rating, or an emotion.
- ***holder***: opinion holder.
- ***time***: time when the opinion was expressed.

*Aspect-based sentiment analysis*

# ASPECT SENTIMENT CLASSIFICATION WITH DOCUMENT-LEVEL SENTIMENT PREFERENCE MODELING (ACL'20)



# ASPECT SENTIMENT CLASSIFICATION WITH DOCUMENT-LEVEL SENTIMENT PREFERENCE MODELING (ACL'20)



# ASPECT EXTRACTION

Goal: Given an opinion corpus, extract all aspects

Four main approaches:

- (1) Finding frequent nouns and noun phrases
- (2) Exploiting opinion and target relations
- (3) Supervised learning
- (4) Topic modeling

# RECOMMENDATION SYSTEM

# THREE CHALLENGES

Cold start problem

- New users or new shops

Implicit feedback

- Has never seen vs. doesn't like

Multi-Source data integration

- Heterogeneous data sources
- Curse of dimensionality

# RELATED WORK

Collaborative filtering

Matrix factorization

Deep learning

Clustering