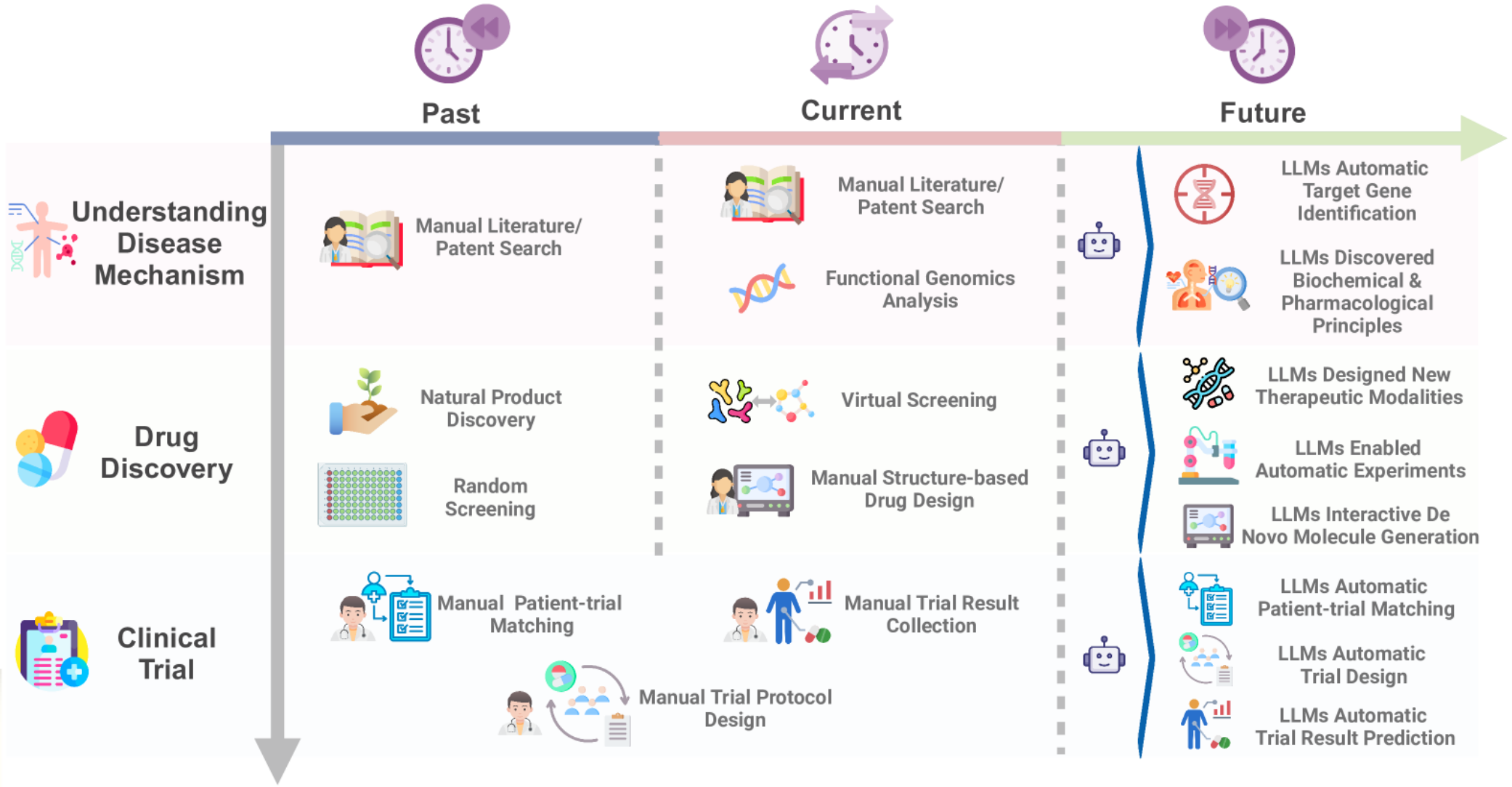
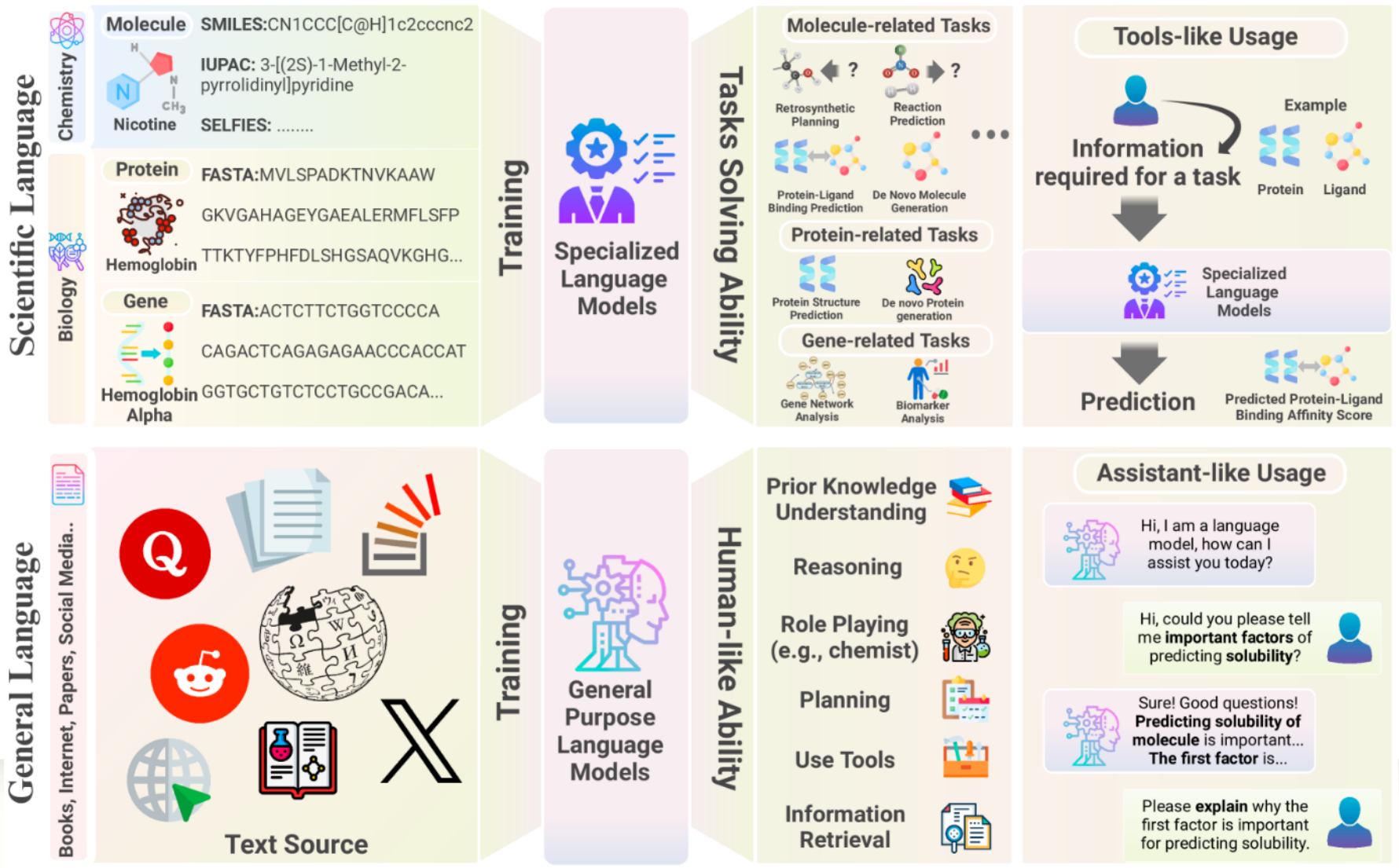
- 08

- **대규모 언어모델과 신약개발**

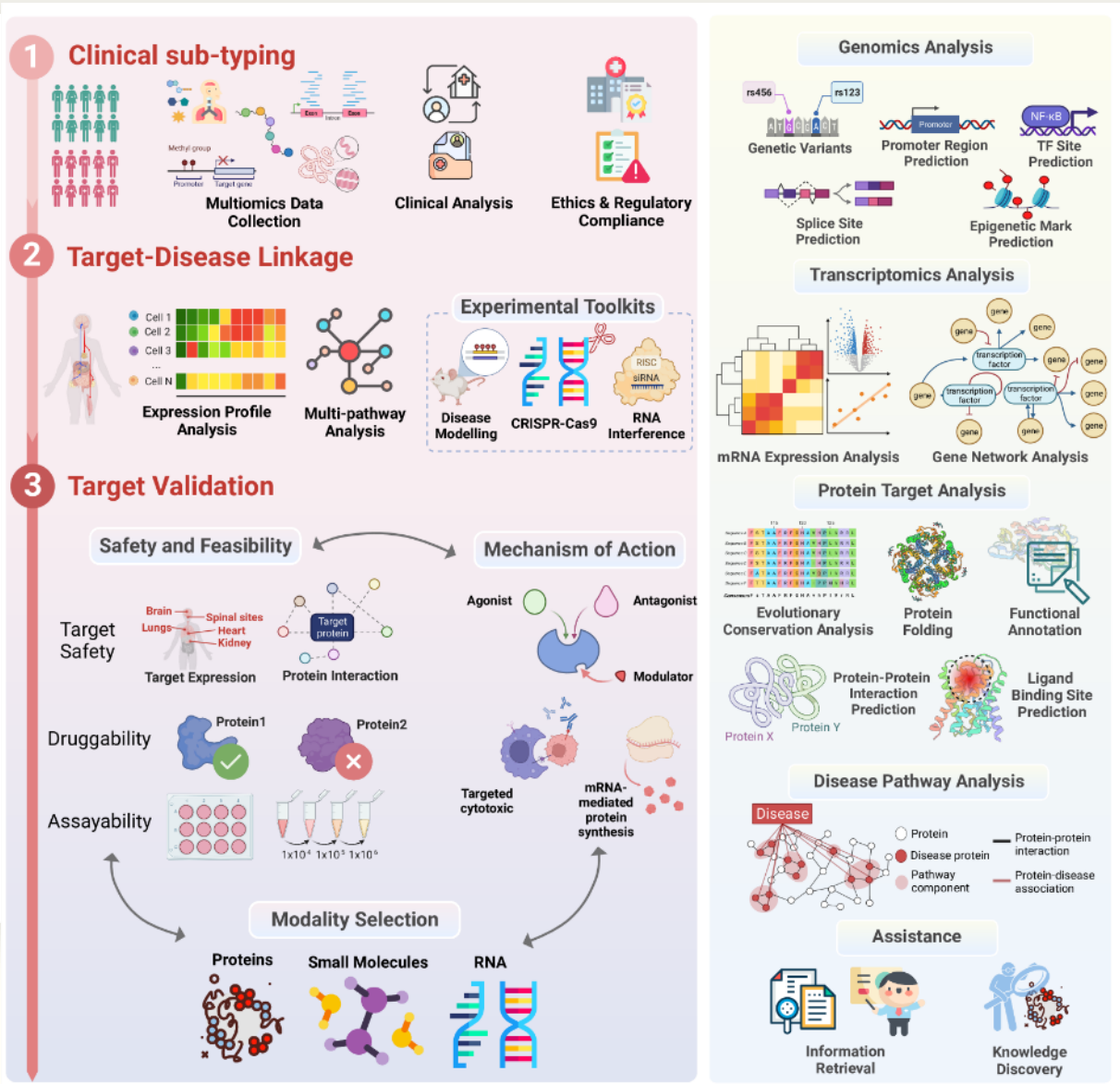
Large Language Models in Drug Discovery and Development: From Disease Mechanisms to Clinical Trials

Yizhen Zheng[1 *] **Huan Yee Koh**[1,2 *] **Maddie Yang**[4] **Li Li**[4,5] **Lauren T. May**[2] **Geoffrey I. Webb**[1]
Shirui Pan[3+] **George Church**[4,5+]
1. Department of Data Science and AI, Monash University
2. Drug Discovery Biology, Monash Institute of Pharmaceutical Sciences, Monash University
3. School of Information and Communication Technology, Griffith University
4. Harvard Medical School, Harvard University
5. Wyss Institute for Biologically Inspired Engineering, Harvard University
* indicates equal contribution and + indicates corresponding authors:
George Church(george_church@hms.harvard.edu), Shirui Pan(s.pan@griffith.edu.au)

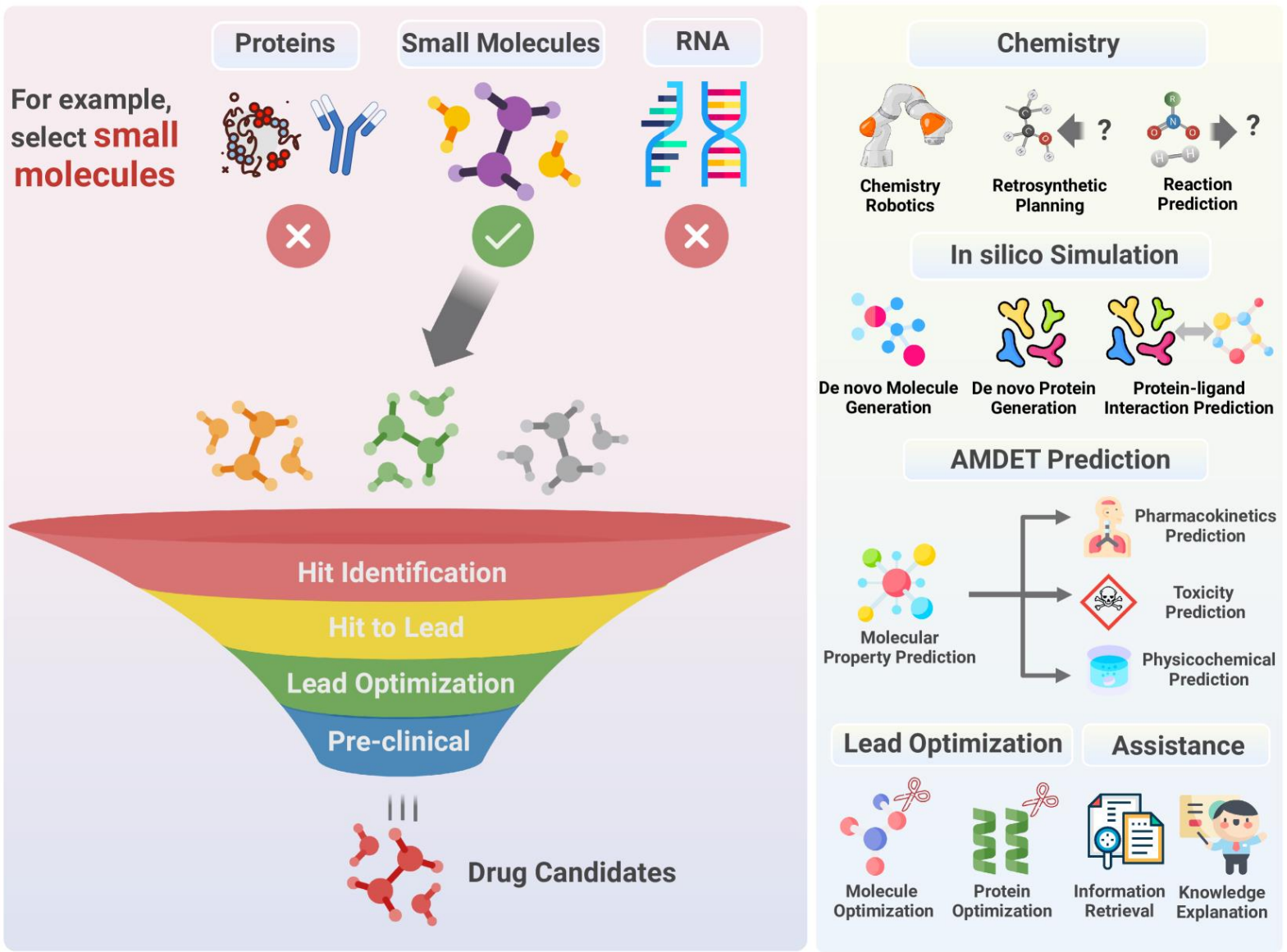

Past
Current
Future
Understanding Disease Mechanism
Drug Discovery
Clinical Trial
Manual Literature/ Patent Search
Natural Product Discovery
Random Screening
Manual Patient-trial Matching
Manual Trial Protocol Design
Manual Literature/ Patent Search
Functional Genomics Analysis
Virtual Screening
Manual Structure-based Drug Design
Manual Trial Result Collection
LLMs Automatic Target Gene Identification
LLMs Discovered Biochemical & Pharmacological Principles
LLMs Designed New Therapeutic Modalities
LLMs Enabled Automatic Experiments
LLMs Interactive De Novo Molecule Generation
LLMs Automatic Patient-trial Matching
LLMs Automatic Trial Design
LLMs Automatic Trial Result Prediction


Scientific Language
Chemistry
Molecule
SMILES:CN1CCC[C@H]1c2cccnc2
IUPAC: 3-[(2S)-1-Methyl-2-pyrrolidinyl]pyridine
Nicotine
SELFIES: .....
Biology
Protein
FASTA:MVLSPADKTNVKAAW
GKVGAHAGEYGAEALERMFLSFP
Hemoglobin
TTKTYFPHFDLSHGSAQVKGHG...
Gene
FASTA:ACTCTTCTGGTCCCCA
CAGACTCAGAGAGAACCCACCAT
Hemoglobin Alpha
GGTGCTGTCTCCTGCCGACA...
Training
Specialized Language Models
Tasks Solving Ability
Molecule-related Tasks
Retrosynthetic Planning
Reaction Prediction
Protein-Ligand Binding Prediction
De Novo Molecule Generation
Protein-related Tasks
Protein Structure Prediction
De novo Protein generation
Gene-related Tasks
Gene Network Analysis
Biomarker Analysis
Tools-like Usage
Information required for a task
Example
Protein
Ligand
Specialized Language Models
Prediction
Predicted Protein-Ligand Binding Affinity Score
General Language
Books, Internet, Papers, Social Media...
Text Source
Training
General Purpose Language Models
Human-like Ability
Prior Knowledge Understanding
Reasoning
Role Playing (e.g., chemist)
Planning
Use Tools
Information Retrieval
Assistant-like Usage
Hi, I am a language model, how can I assist you today?
Hi, could you please tell me important factors of predicting solubility?
Sure! Good questions! Predicting solubility of molecule is important... The first factor is...
Please explain why the first factor is important for predicting solubility.

- Understanding Disease Mechanisms



- Drug Discovery



- Clinical Trials

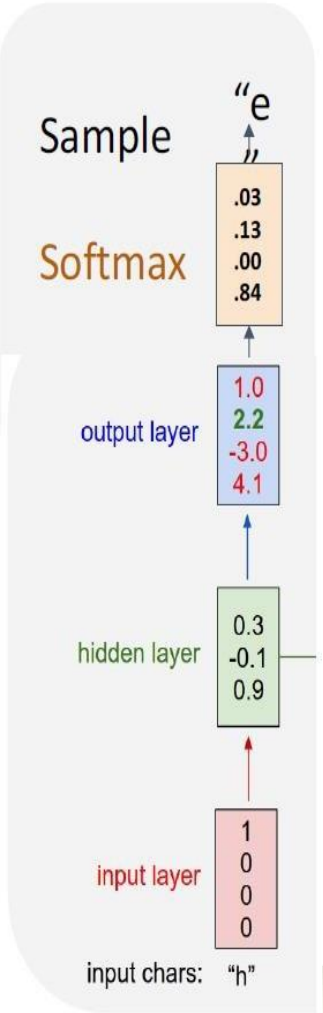
Example: Language Modeling

At test-time, **generate** new text: sample characters one at a time, feed back to model

★ 테스트타임에서 새로운 텍스트를 생성한다: 매 타임스텝에서 문자를 샘플링해서 모델에 피드백한다.

Training sequence: "hello"

Vocabulary: [h, e, l, o]



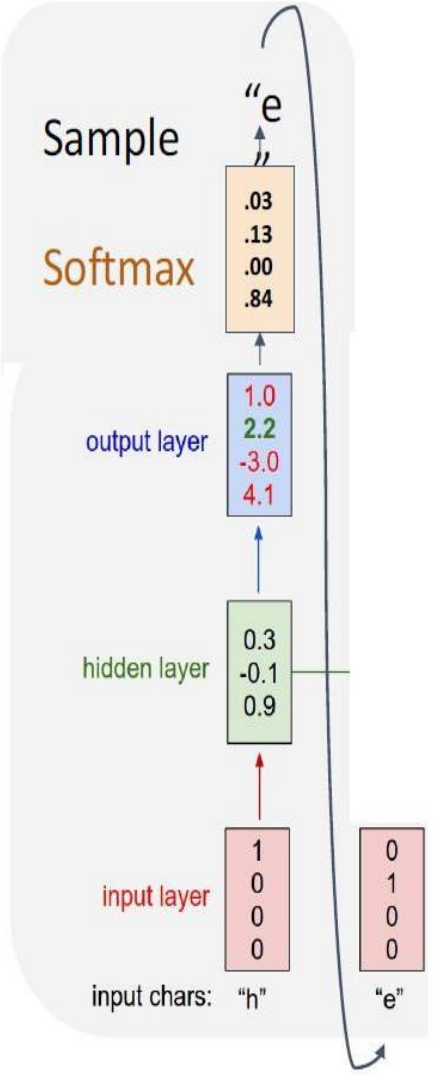
Example: Language Modeling

At test-time, **generate** new text: sample characters one at a time, feed back to model

★ 테스트타임에서 새로운 텍스트를 생성한다: 매 타임스텝에서 문자를 샘플링해서 모델에 피드백한다.

Training sequence: ”hello”

Vocabulary: [h, e, l, o]



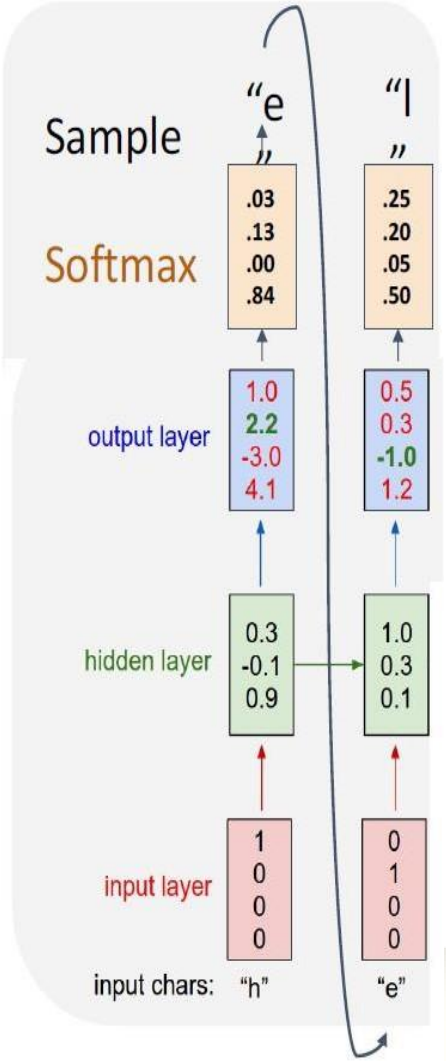
Example: Language Modeling

At test-time, **generate** new text: sample characters one at a time, feed back to model

★ 테스트타임에서 새로운 텍스트를 생성한다: 매 타임스텝에서 문자를 샘플링해서 모델에 피드백한다.

Training sequence: ”hello”

Vocabulary: [h, e, l, o]



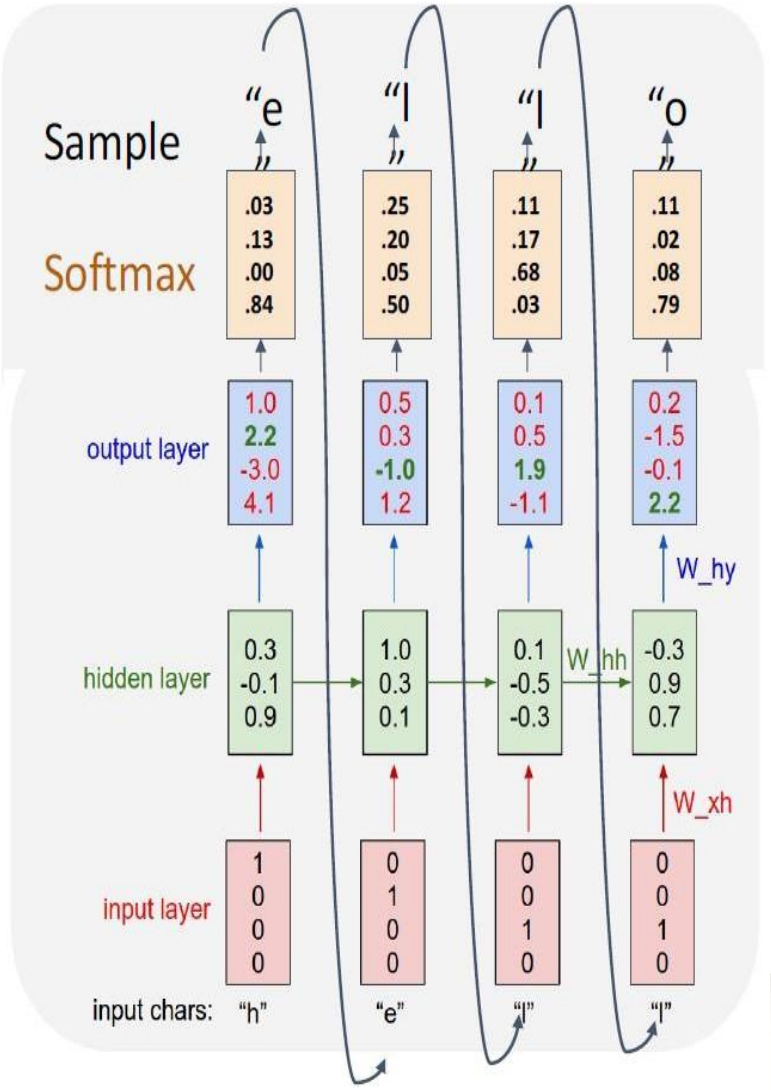
Example: Language Modeling

At test-time, **generate** new text: sample characters one at a time, feed back to model

★ 테스트타임에서 새로운 텍스트를 생성한다: 매 타임스텝에서 문자를 샘플링해서 모델에 피드백한다.

Training sequence: "hello"

Vocabulary: [h, e, l, o]



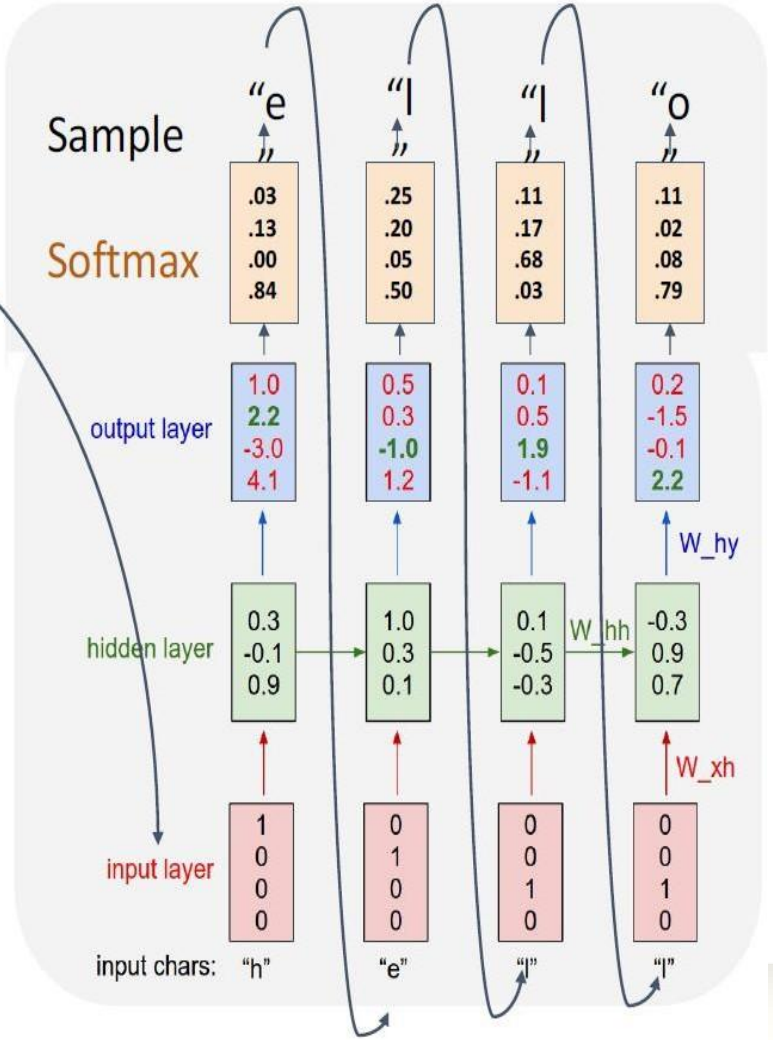
Example: Language Modeling

So far: encode inputs as **one-hot-vector**

★ 이제까지 입력을 원핫벡터로 인코드.

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{14} \\ w_{31} & w_{32} & w_{33} & w_{14} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} w_{11} \\ w_{21} \\ w_{31} \end{bmatrix}$$

Matrix multiply with a one-hot vector just extracts a column from the weight matrix. Often extract this into a separate **embedding** layer



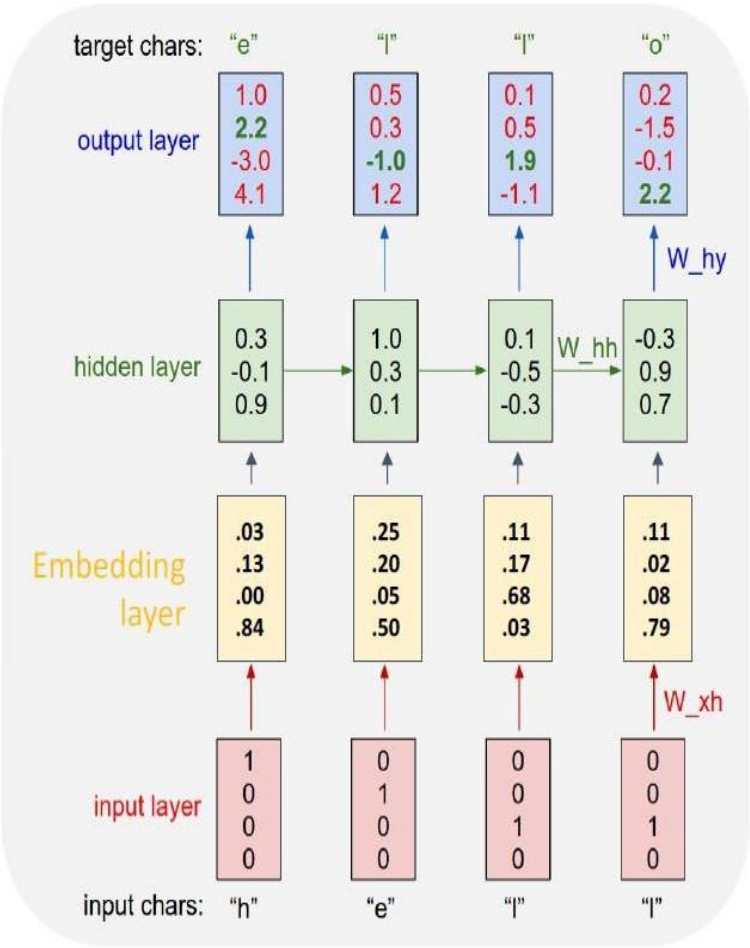
★ 별도의 임베딩층을 만들 수도 있다.

Example: Language Modeling

So far: encode inputs as **one-hot-vector**

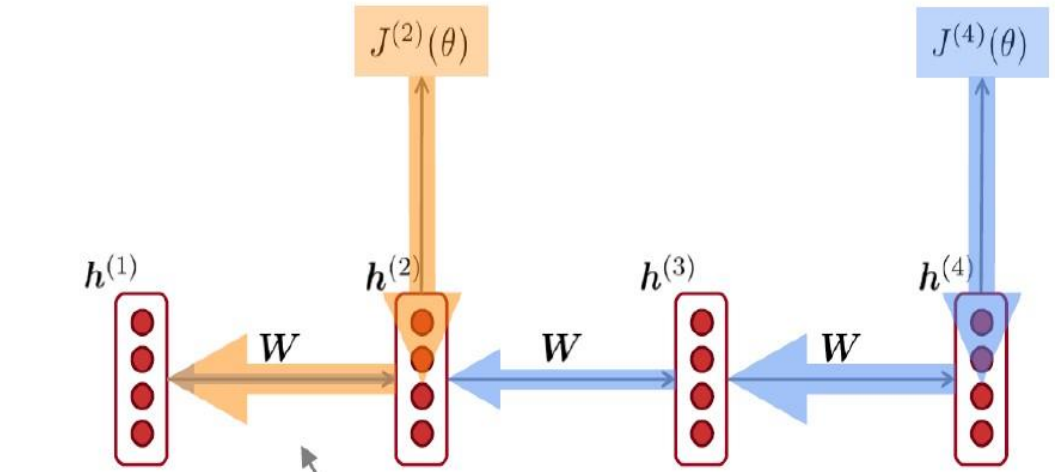
$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{14} \\ w_{31} & w_{32} & w_{33} & w_{14} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} w_{11} \\ w_{21} \\ w_{31} \end{bmatrix}$$

Matrix multiply with a one-hot vector just extracts a column from the weight matrix. Often extract this into a separate **embedding** layer



★ 별도의 임베딩층을 만들 수도 있다.

Why is Vanishing Gradient a Problem?



Gradient signal from faraway is lost because it's much smaller than gradient signal from close-by.

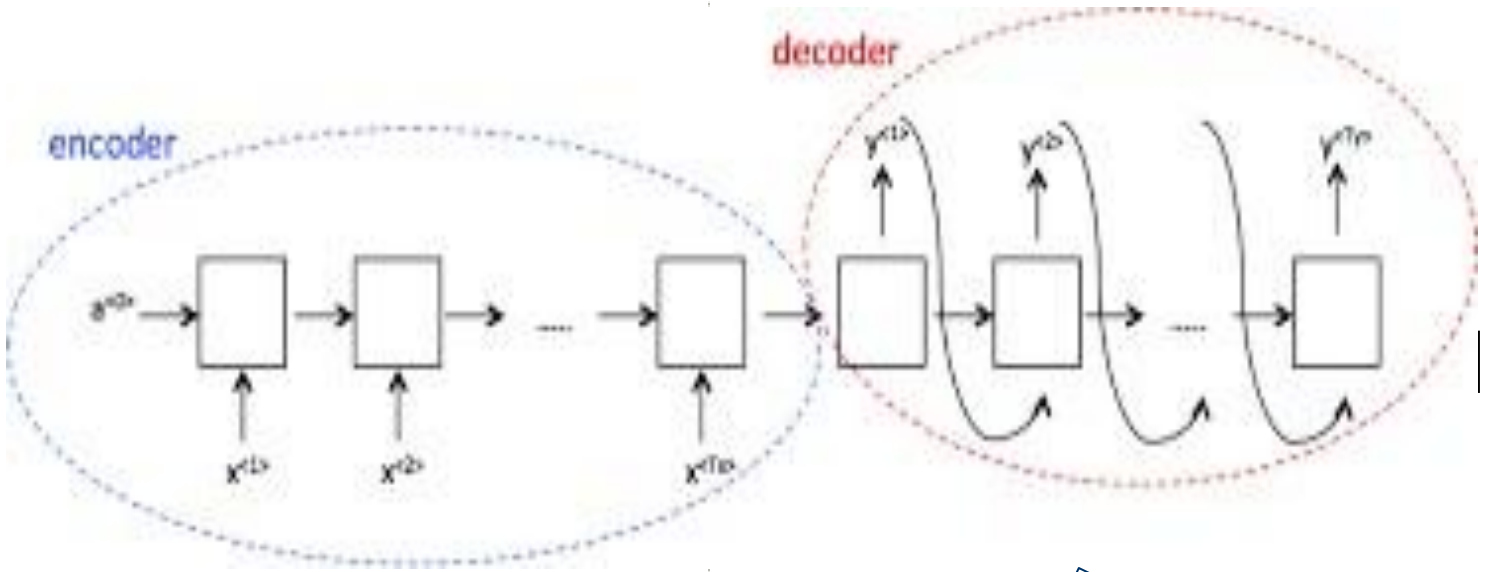
So model weights are only updated only with respect to near effects, not long-term effects.

Vanilla RNN의 역전파 – 그래디언트 소멸 문제

★ 그래디언트 소멸 문제

- 그래디언트는 과거의 미래에 대한 영향이다. 만약 기간가의 거리가 멀어 (예를 들면, 타임스텝 t과 타임스텝 t+n의 거리) 그래디언트가 매우 작으면 타임스텝 t와 t+n 사이의 상관관계가 없는 것인지, 타임스텝 t와 t+n 간의 참 파라미터를 잘 못 이해하는 것인지 구별할 수 없다.

그림



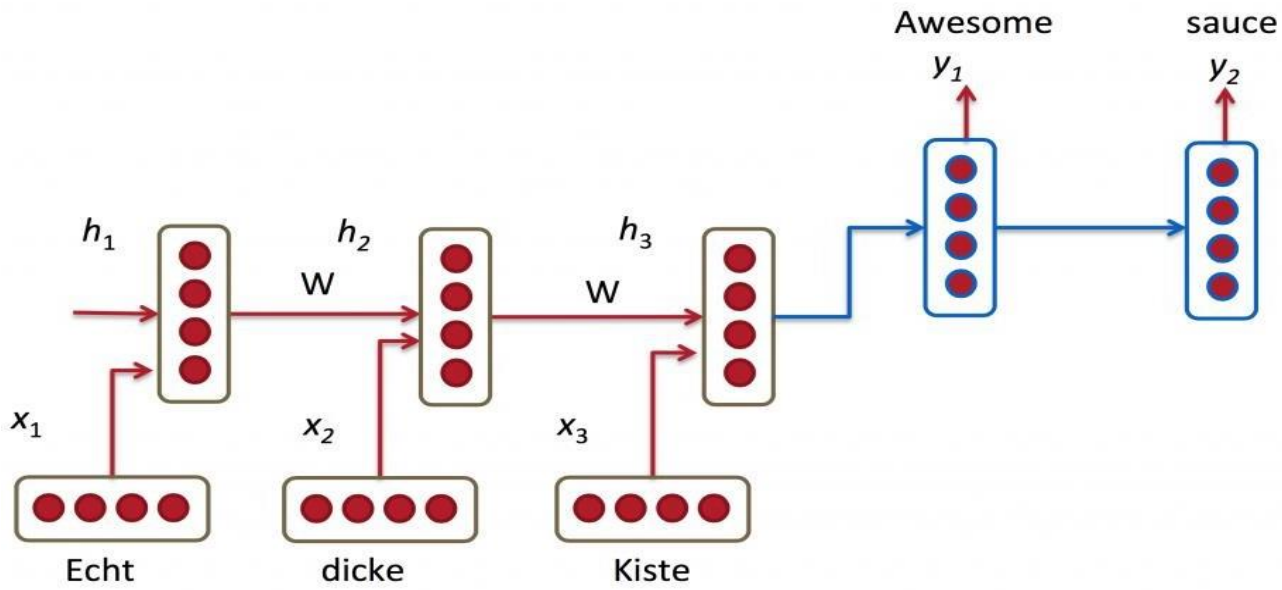
용

파트1

파트2

★ 어떻게 Keras로 구현할까 생각해보자.

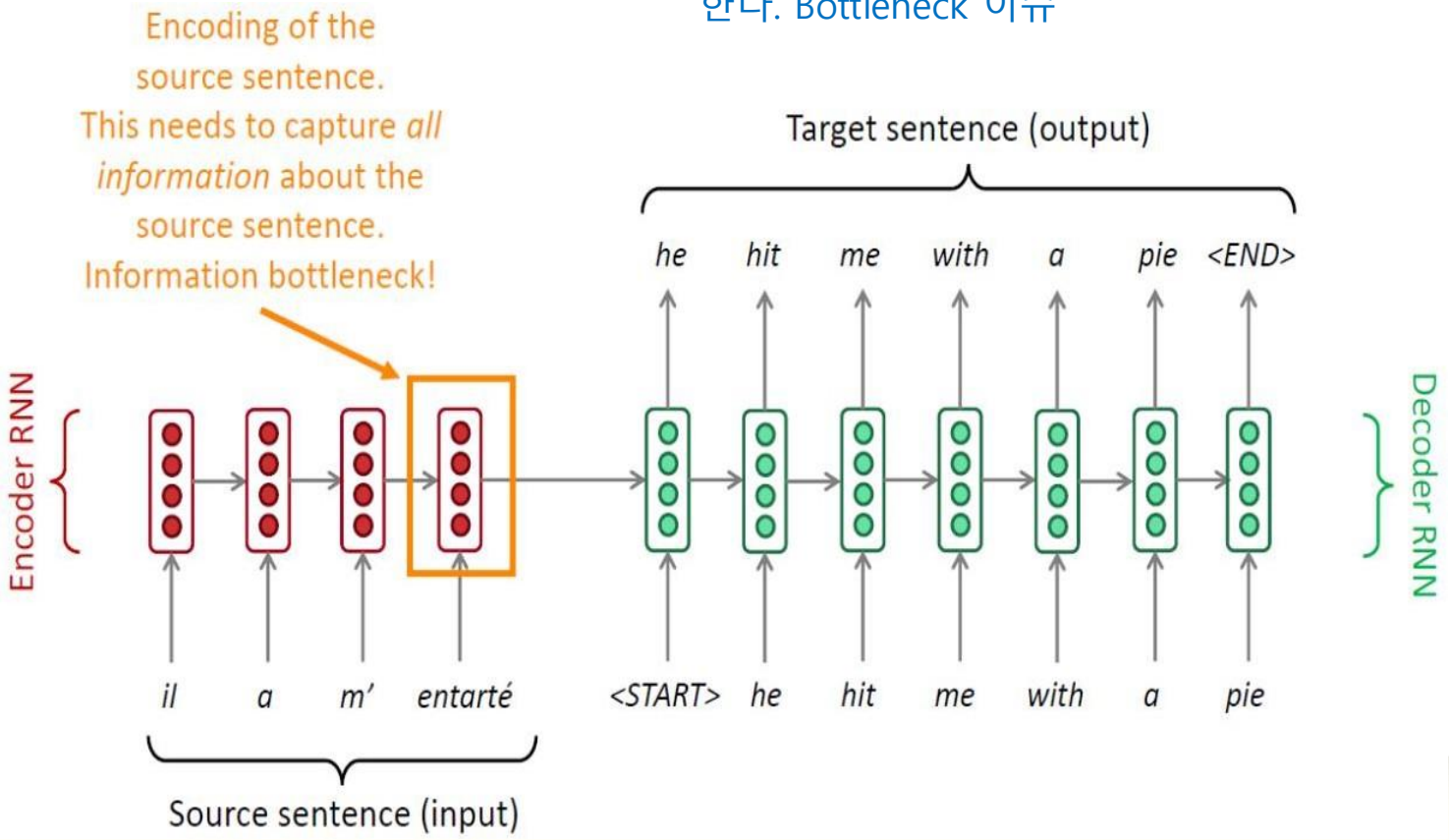
★ seq2seq 모델



ector)

★ seq2seq 모델: Bottleneck 이슈

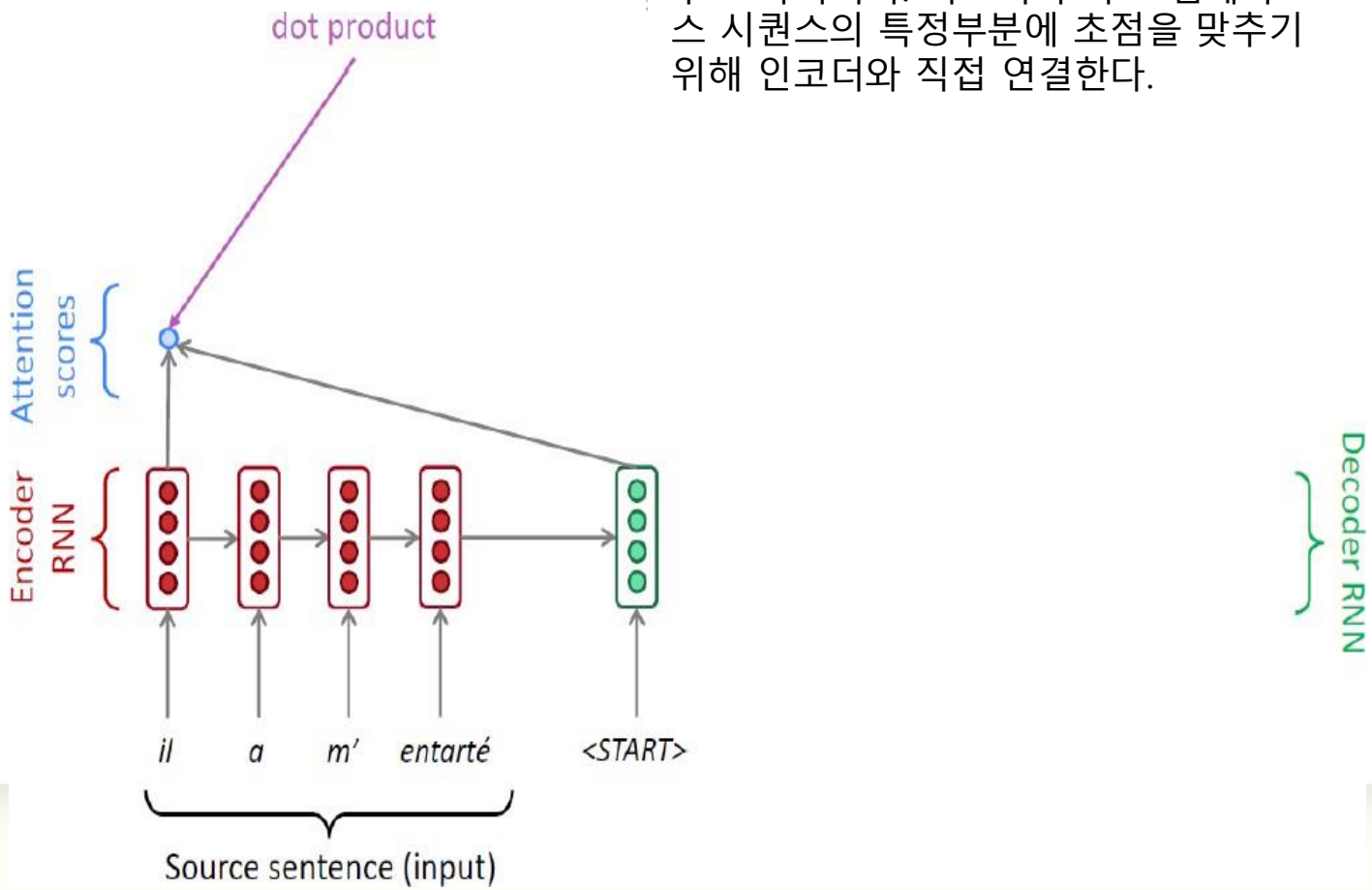
★ 소스 시퀀스의 인코딩: 소스 시퀀스의 모든 정보를 포착해야 한다. Bottleneck 이슈



★ seq2seq 모델: attention

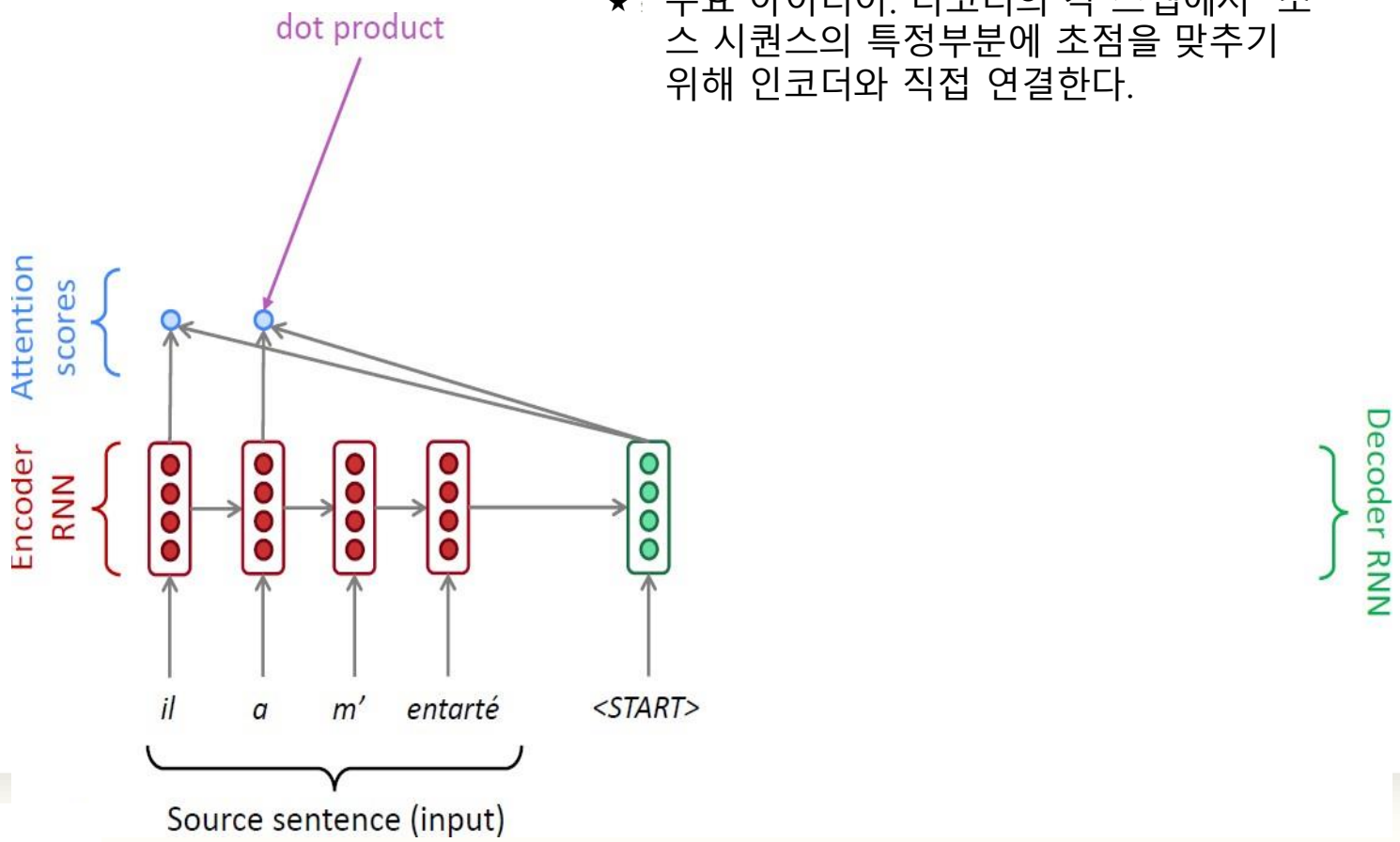
★ Attention은 bottleneck 문제에 좋은 해법을 제공한다.

★ 주요 아이디어: 디코더의 각 스텝에서 소스 시퀀스의 특정부분에 초점을 맞추기 위해 인코더와 직접 연결한다.



★ seq2seq 모델: attention

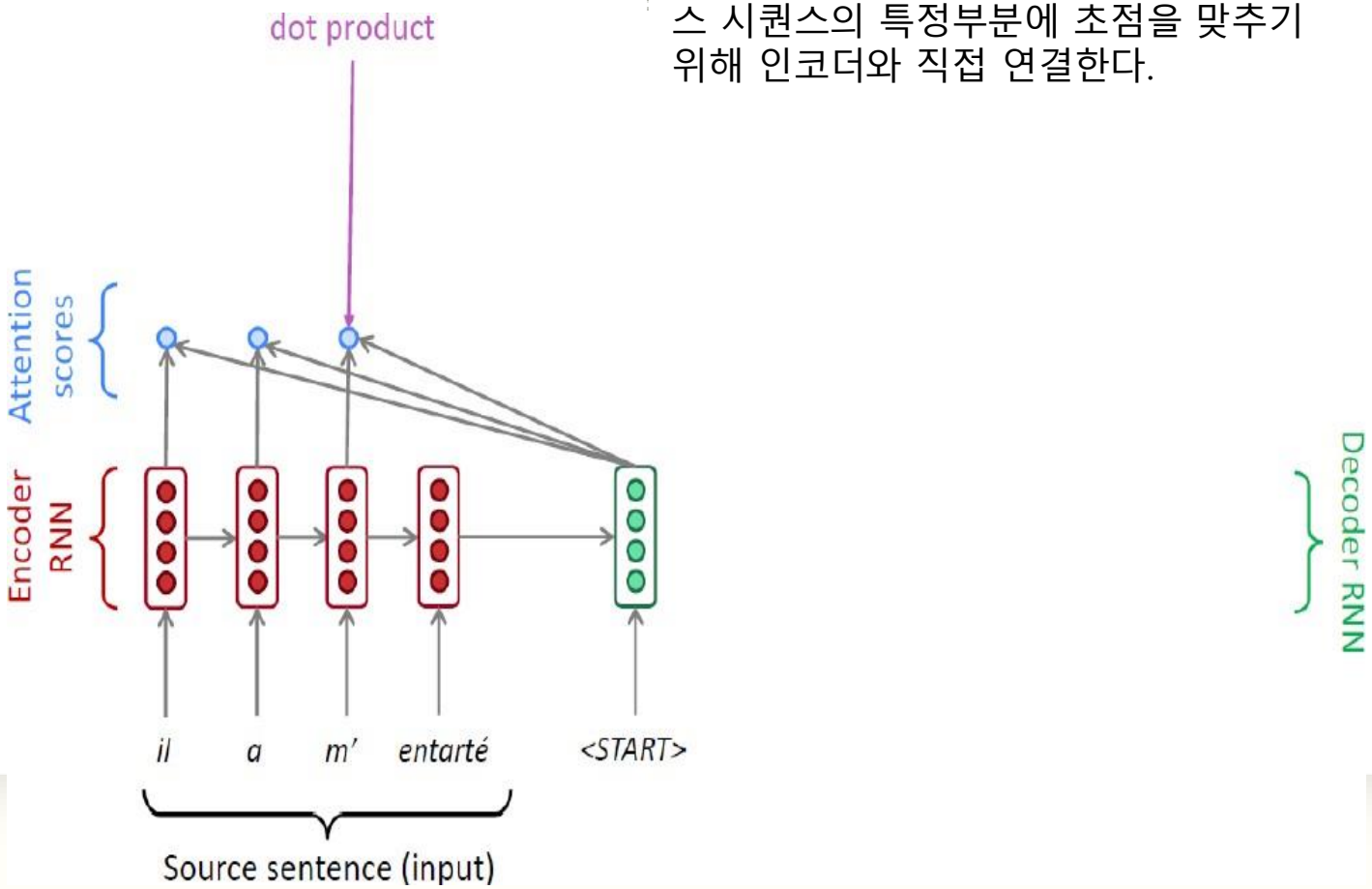
- ★ Attention은 bottleneck 문제에 좋은 해법을 제공한다.
- ★ 주요 아이디어: 디코더의 각 스텝에서 소스 시퀀스의 특정부분에 초점을 맞추기 위해 인코더와 직접 연결한다.



★ seq2seq 모델: attention

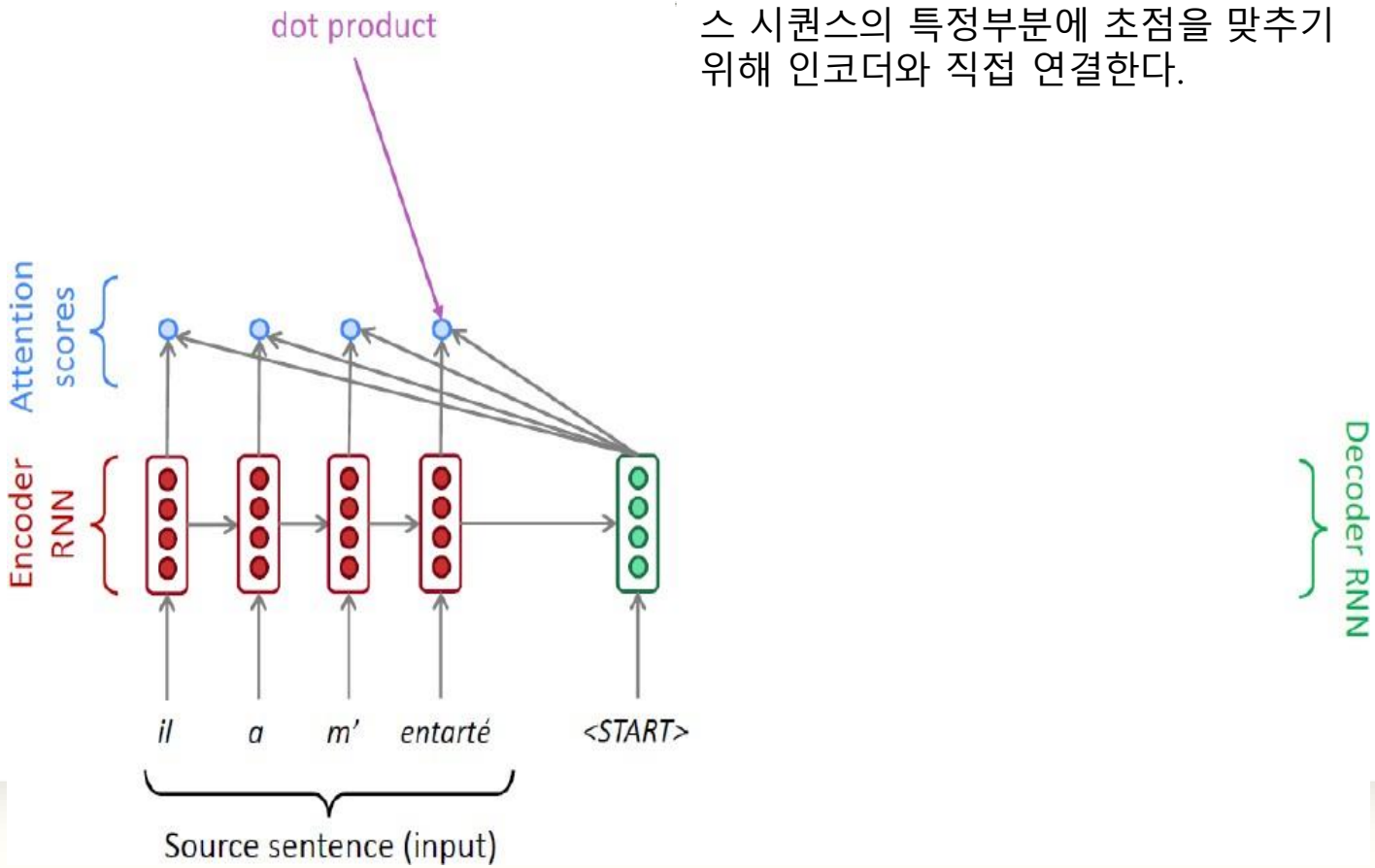
★ Attention은 bottleneck 문제에 좋은 해법을 제공한다.

★ 주요 아이디어: 디코더의 각 스텝에서 소스 시퀀스의 특정부분에 초점을 맞추기 위해 인코더와 직접 연결한다.

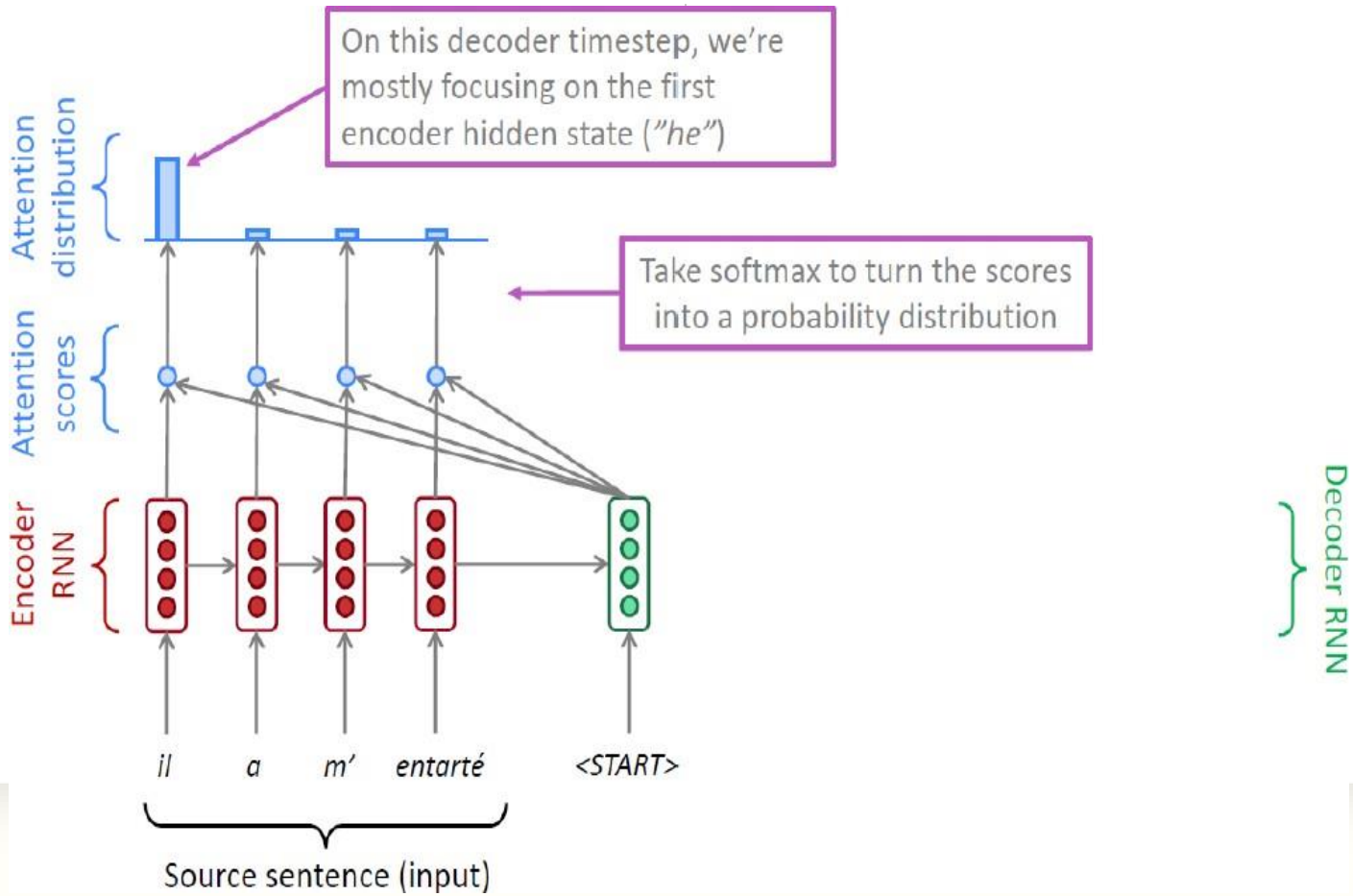


★ seq2seq 모델: attention

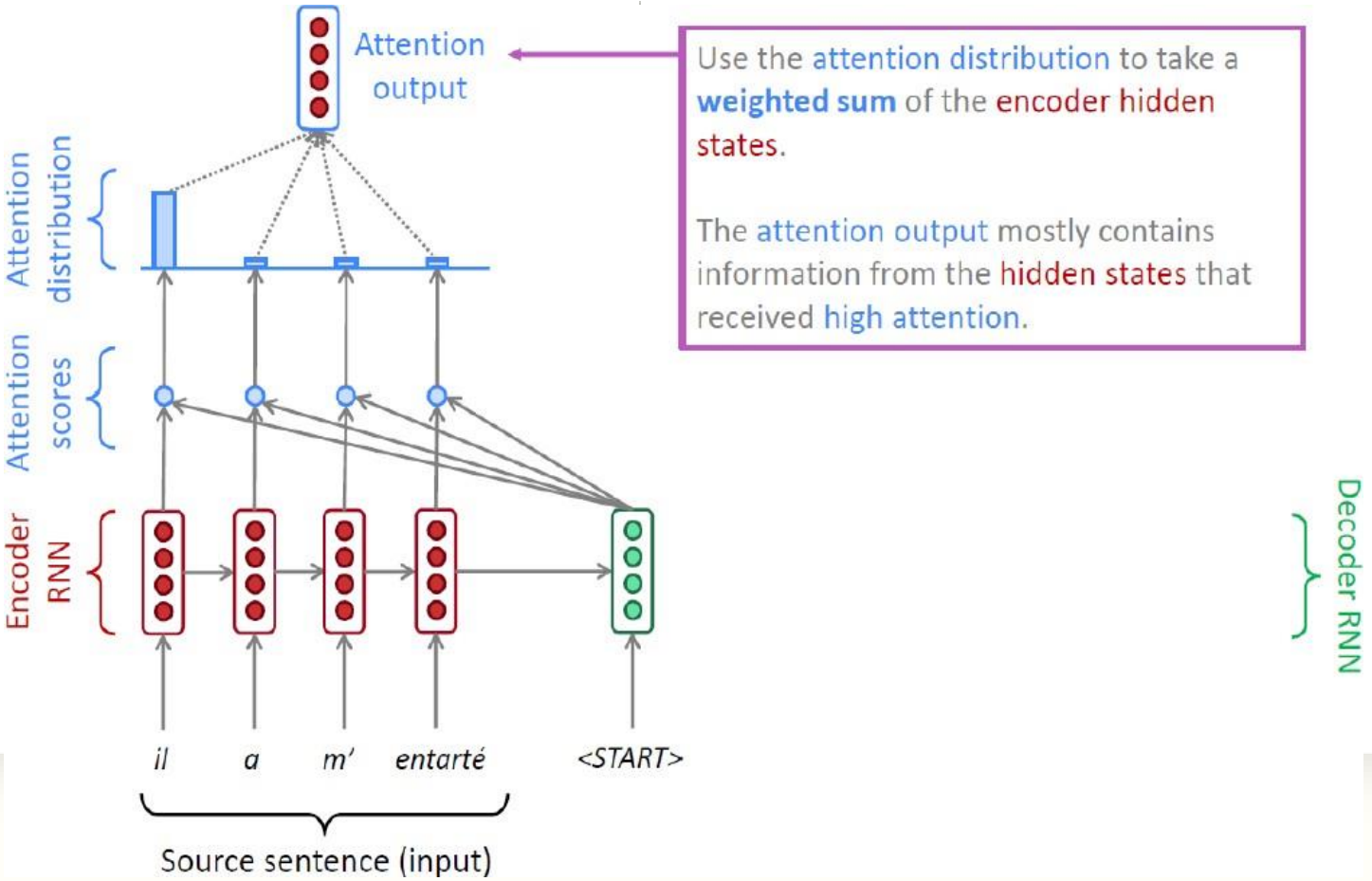
- ★ Attention은 bottleneck 문제에 좋은 해법을 제공한다.
- ★ 주요 아이디어: 디코더의 각 스텝에서 소스 시퀀스의 특정부분에 초점을 맞추기 위해 인코더와 직접 연결한다.



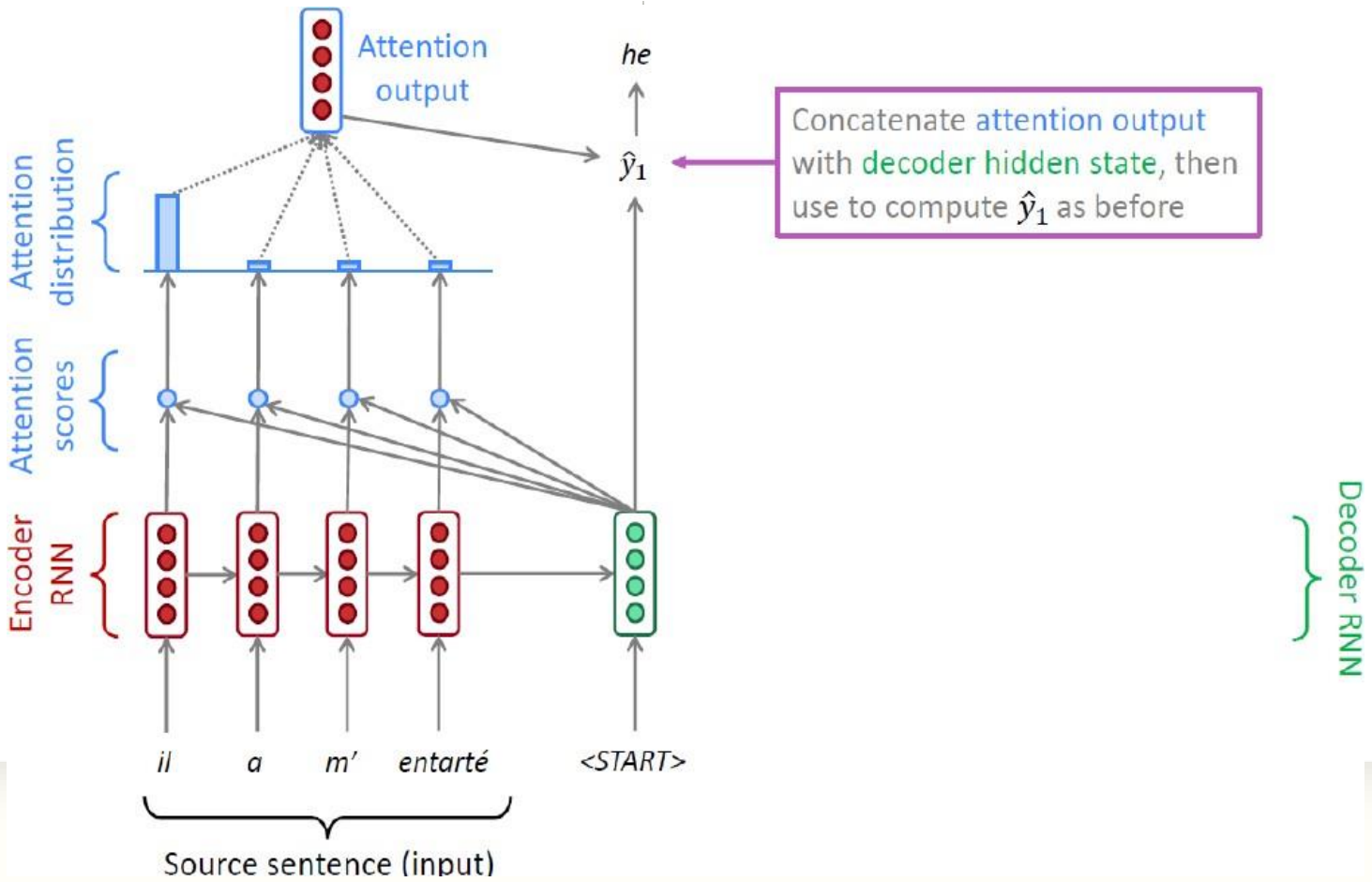
★ seq2seq 모델: attention



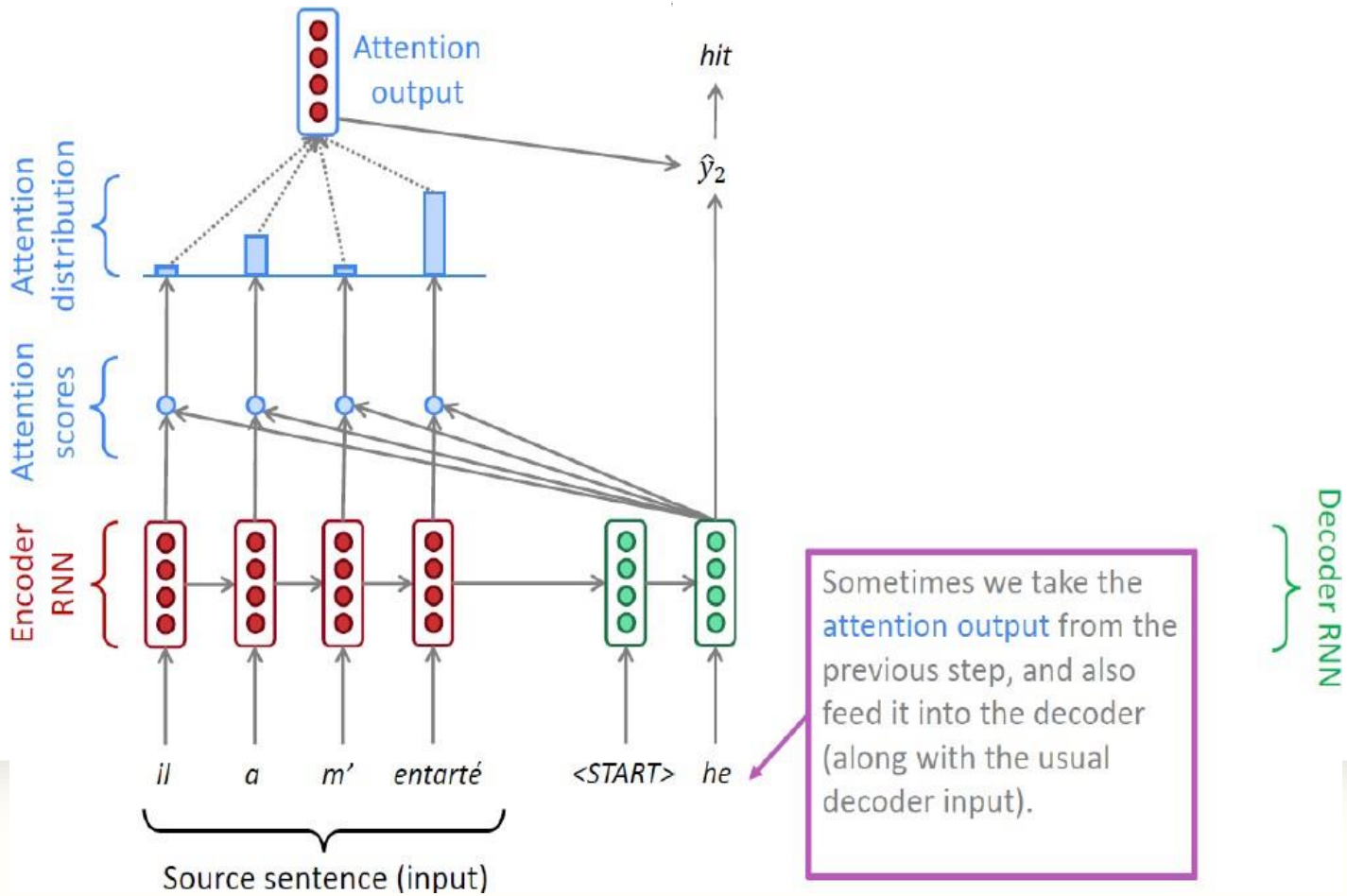
★ seq2seq 모델: attention



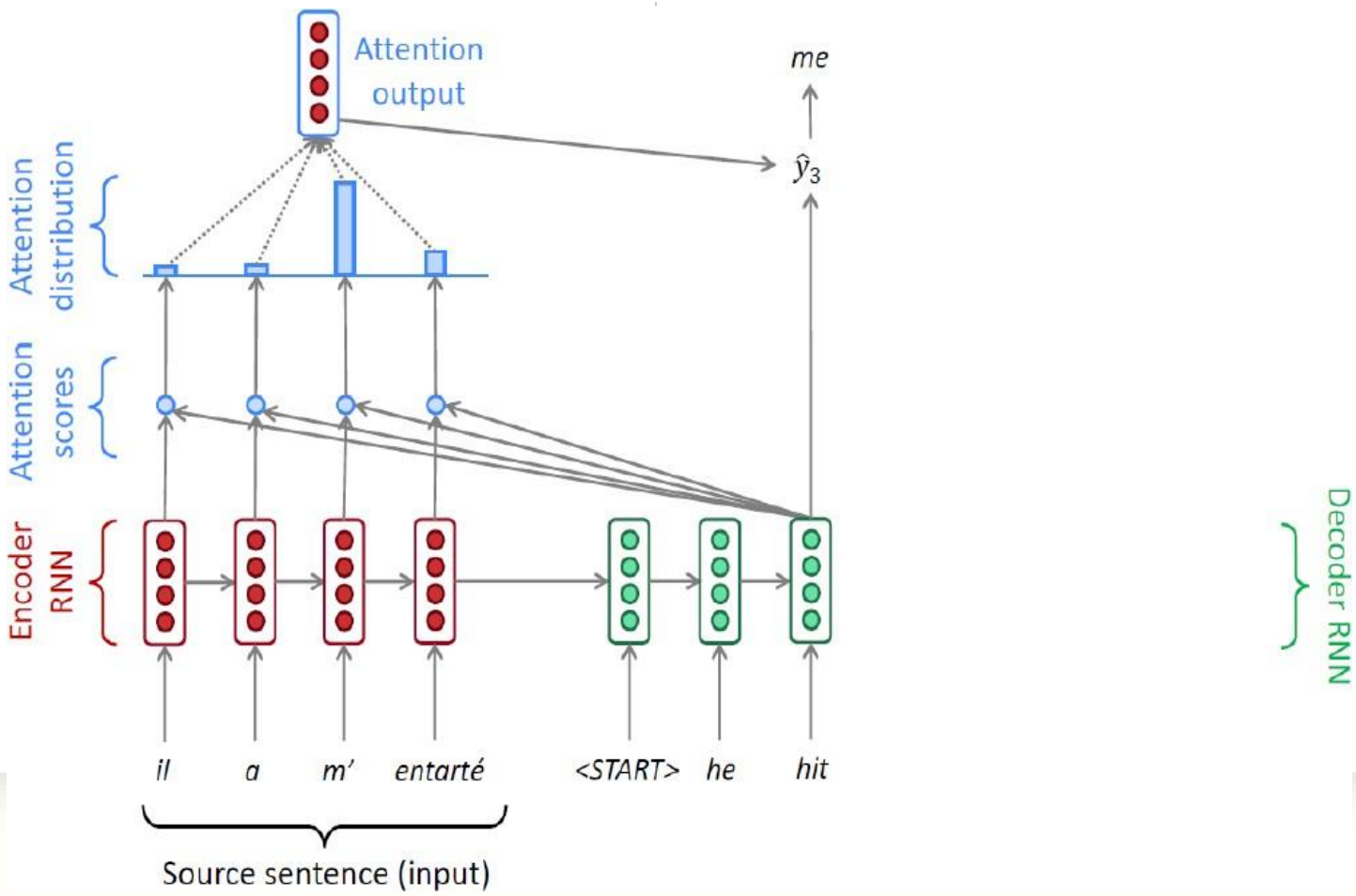
★ seq2seq 모델: attention



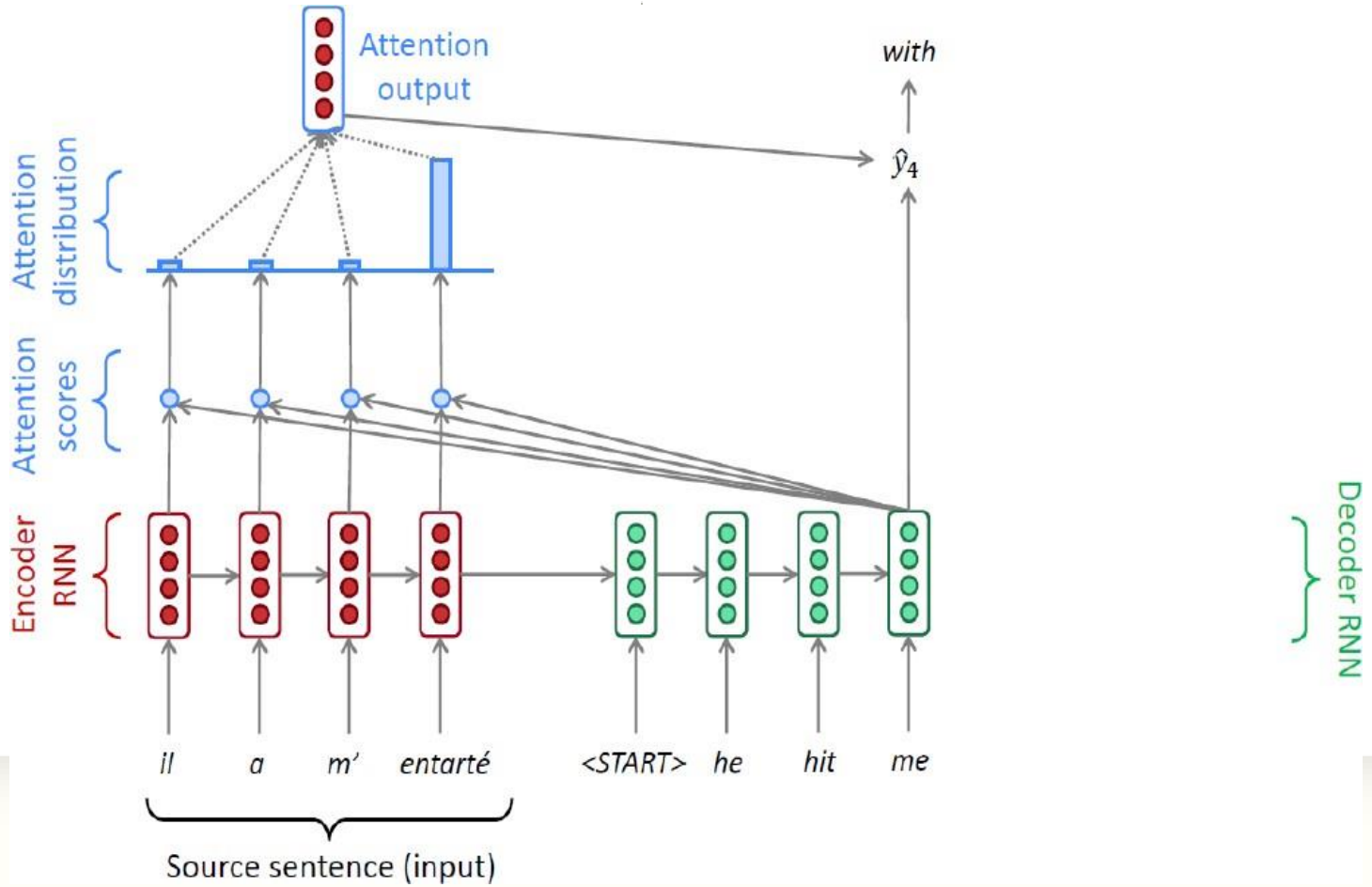
★ seq2seq 모델: attention



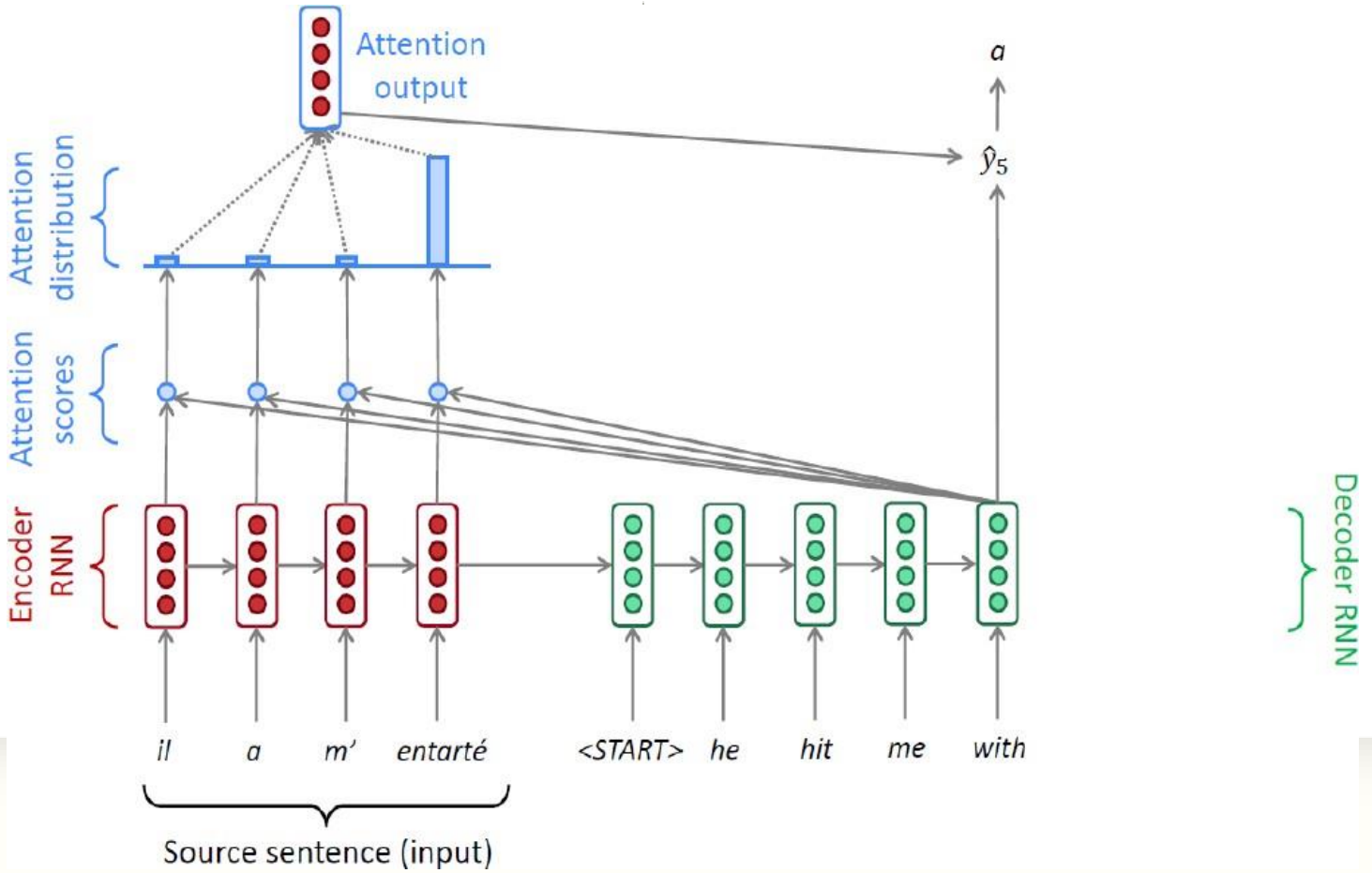
★ seq2seq 모델: attention



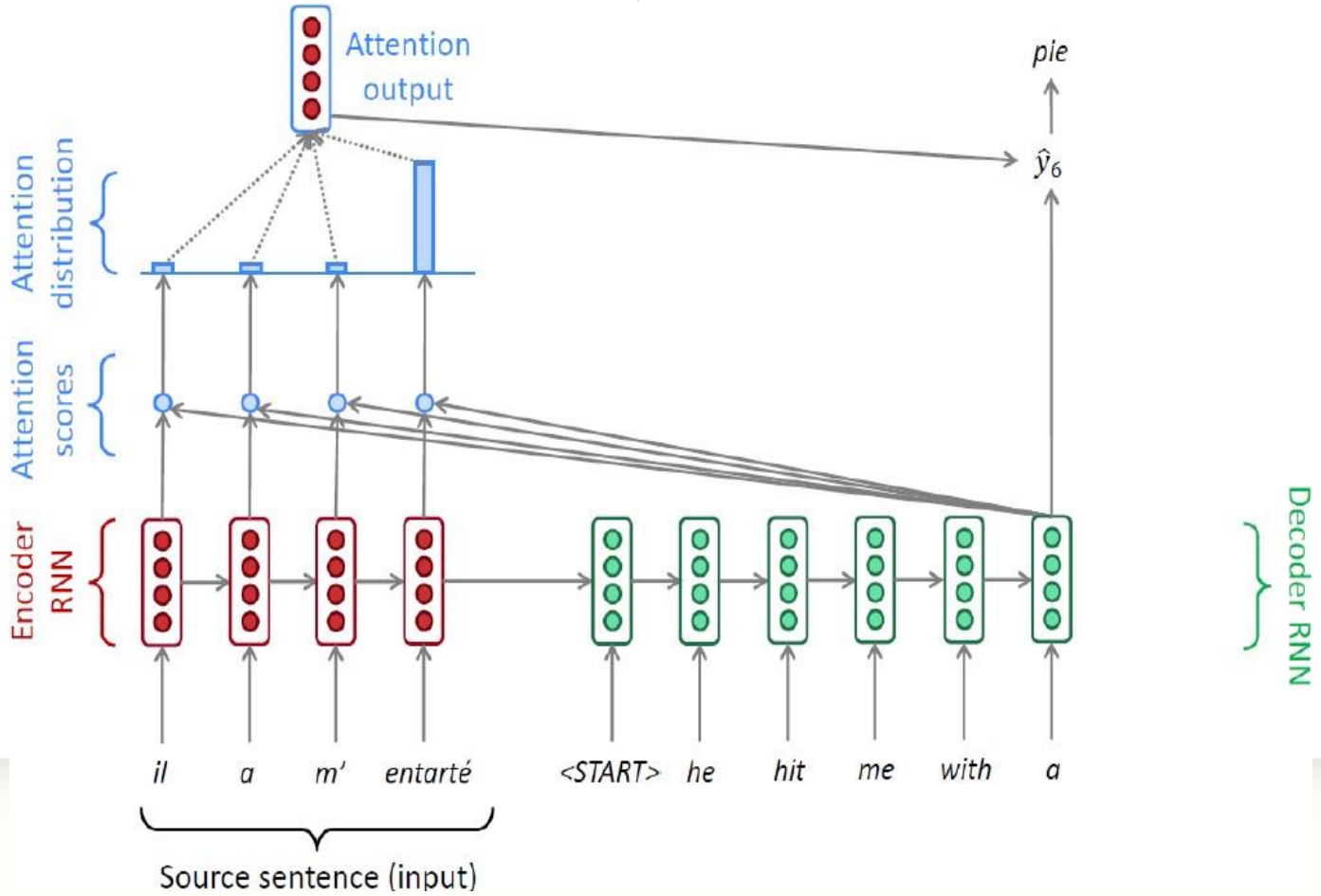
★ seq2seq 모델: attention



★ seq2seq 모델: attention



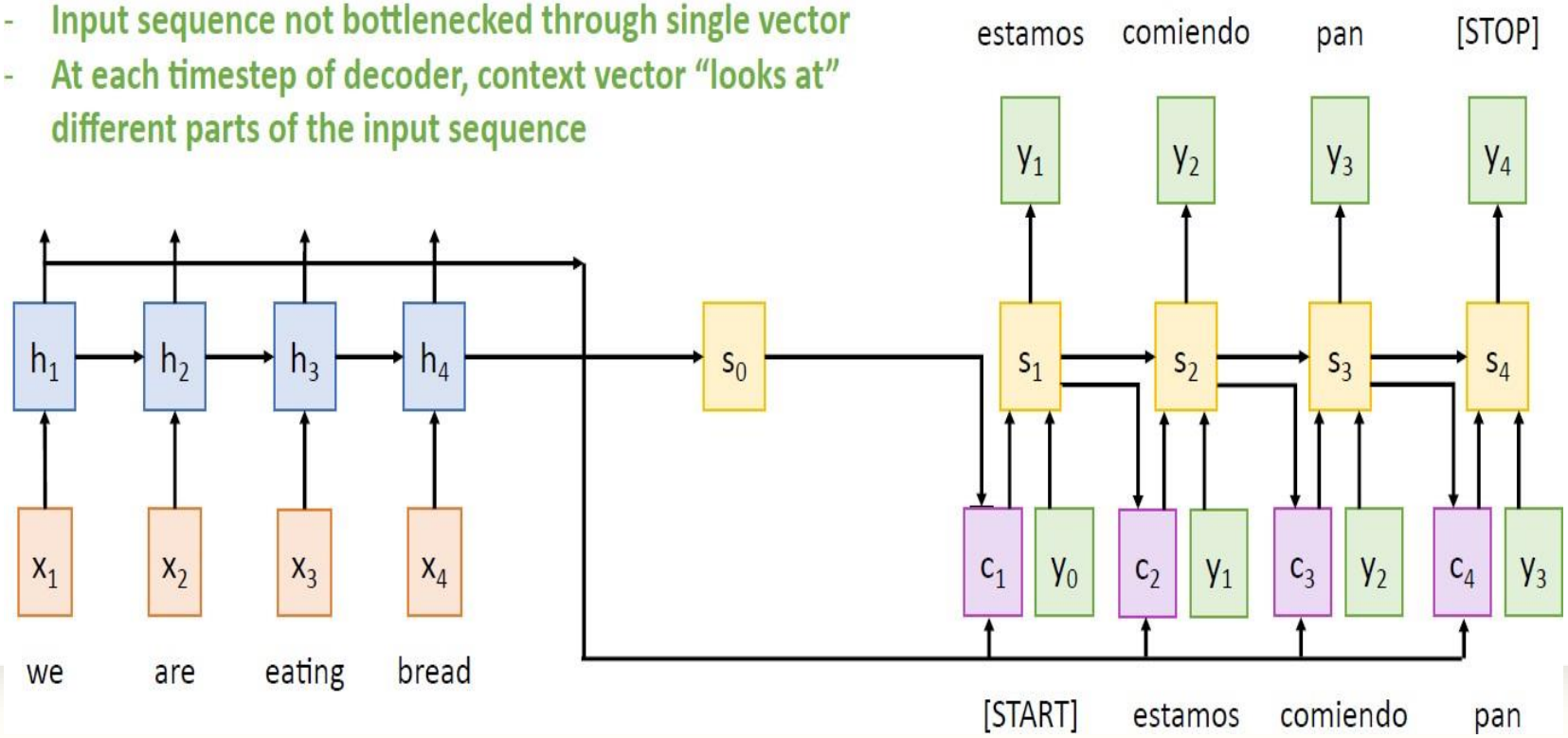
★ seq2seq 모델: attention



★ *디코더의 각 타임스텝에 다른 context 벡터를 사용하므로 no bottleneck*

★ *각 타임스텝에서 context 벡터는 입력시퀀스의 다른 부분을 본다.*

Use a different context vector in each timestep of decoder
- Input sequence not bottlenecked through single vector
- At each timestep of decoder, context vector "looks at" different parts of the input sequence



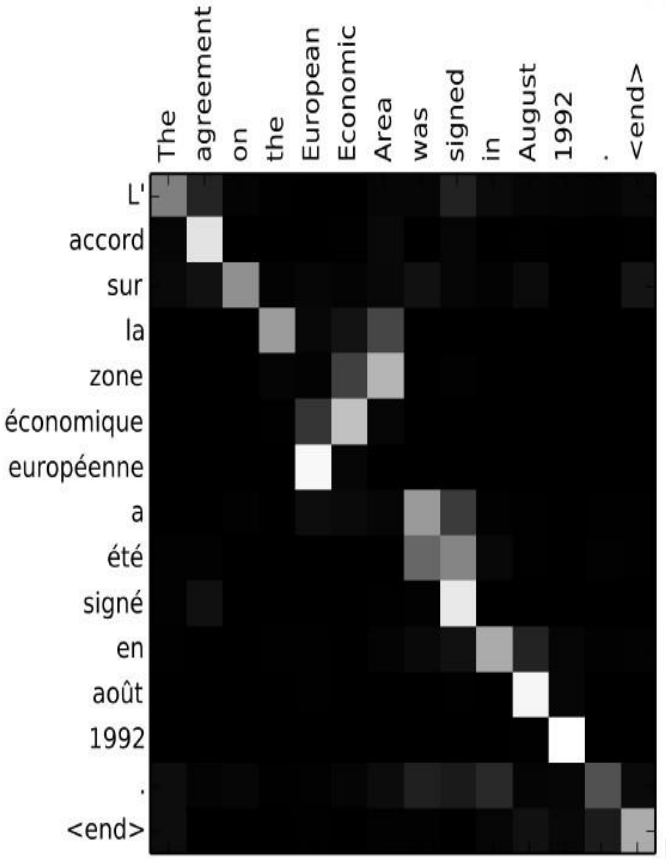
★ *영어를 불어로 번역*

Example: English to French translation

Input: "The agreement on the European Economic Area was signed in August 1992."

Output: "L'accord sur la zone économique européenne a été signé en août 1992."

Visualize attention weights $a_{t,i}$

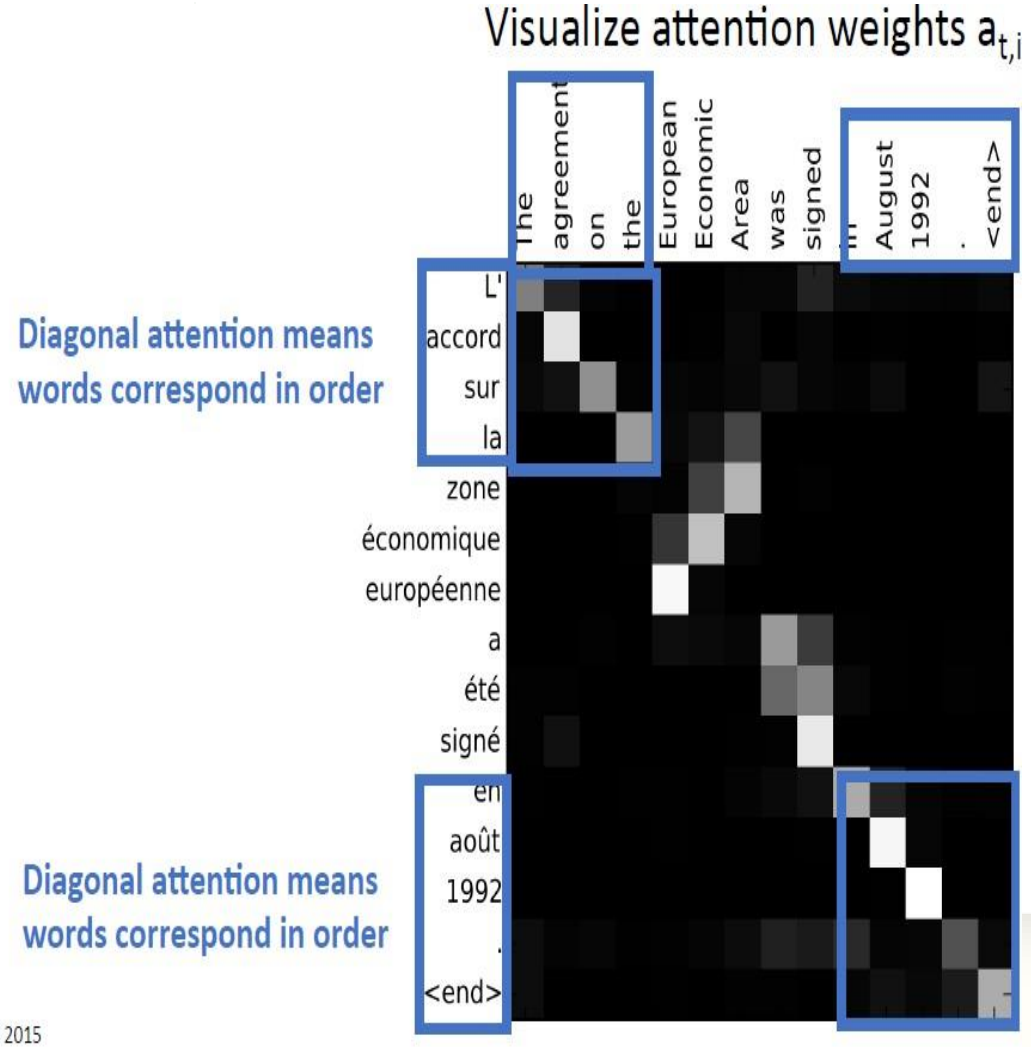


★ *영어를 불어로 번역*

Example: English to French translation

Input: "The agreement on the European Economic Area was signed in August 1992."

Output: "L'accord sur la zone économique européenne a été signé en août 1992."



ıdanau et al, "Neural machine translation by jointly learning to align and translate", ICLR 2015

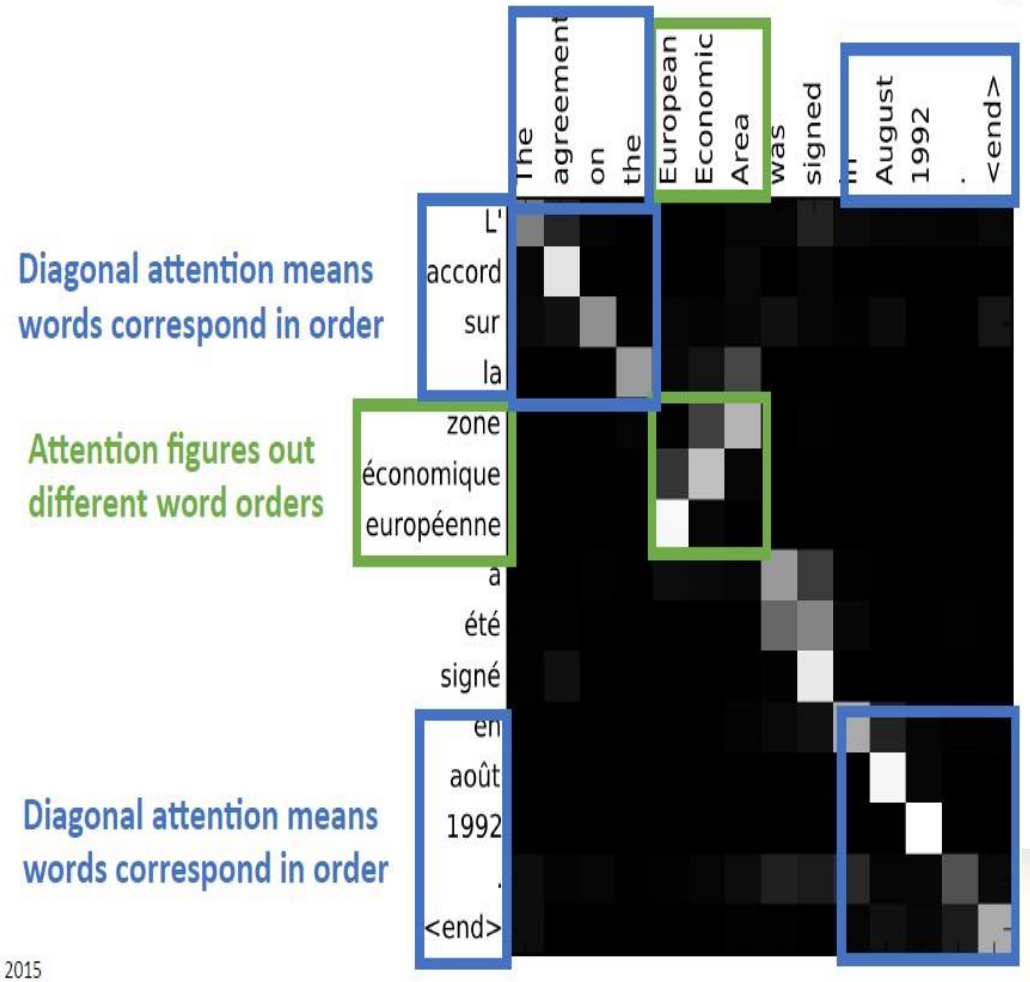
★ *영어를 불어로 번역*

Example: English to French translation

Input: "The agreement on the European Economic Area was signed in August 1992."

Output: "L'accord sur la zone économique européenne a été signé en août 1992."

Visualize attention weights $a_{t,i}$



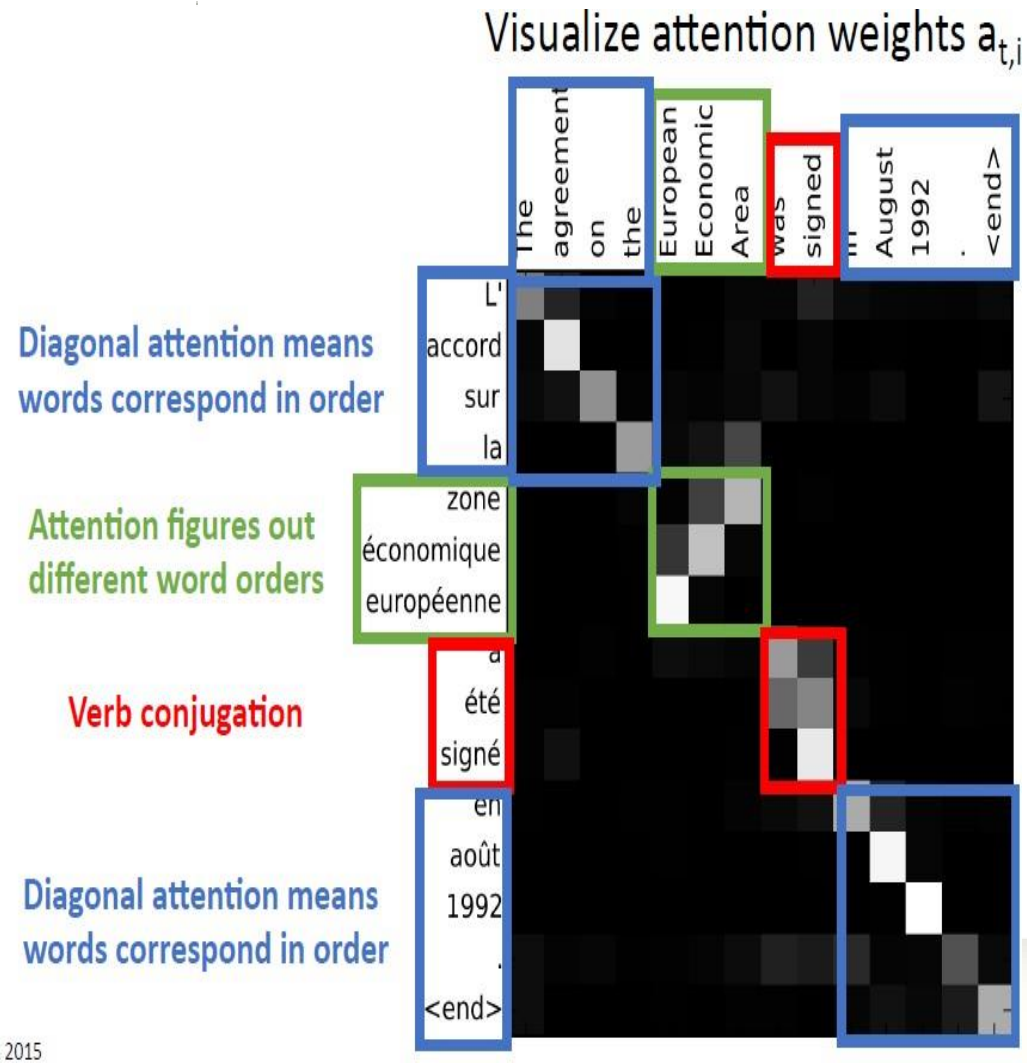
ahdanau et al, "Neural machine translation by jointly learning to align and translate", ICLR 2015

★ *영어를 불어로 번역*

Example: English to French translation

Input: "The agreement on the European Economic Area was signed in August 1992."

Output: "L'accord sur la zone économique européenne a été signé en août 1992."

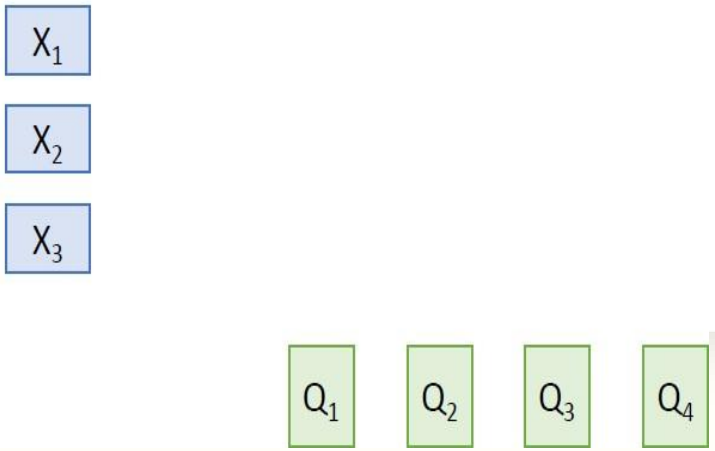


ahdanau et al, "Neural machine translation by jointly learning to align and translate", ICLR 2015

Attention Layer

Inputs:
Query vectors: Q (Shape: $N_Q \times D_Q$)
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)

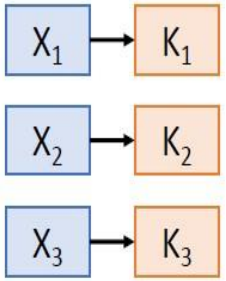
Computation:
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T$ (Shape: $N_Q \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_Q \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_Q \times D_V$) $Y_i = \sum_j A_{i,j} V_j$

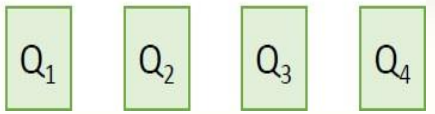


Attention Layer

Inputs:
Query vectors: Q (Shape: $N_Q \times D_Q$)
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)

Computation:
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T$ (Shape: $N_Q \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: A = softmax(E, dim=1) (Shape: $N_Q \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_Q \times D_V$) $Y_i = \sum_j A_{i,j} V_j$

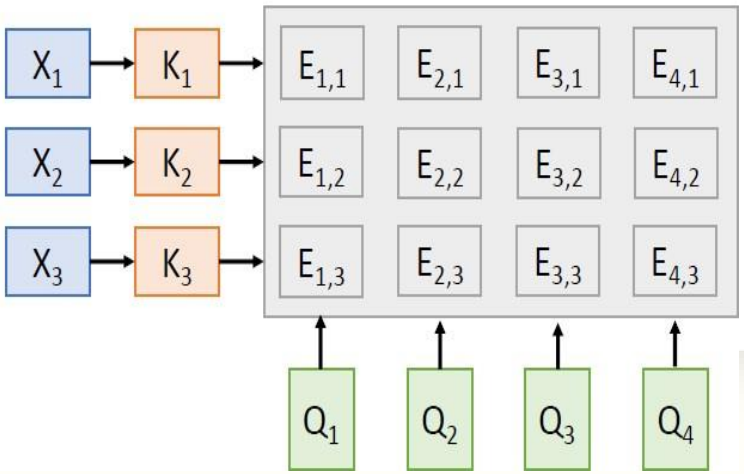




Attention Layer

Inputs:
Query vectors: Q (Shape: $N_Q \times D_Q$)
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)

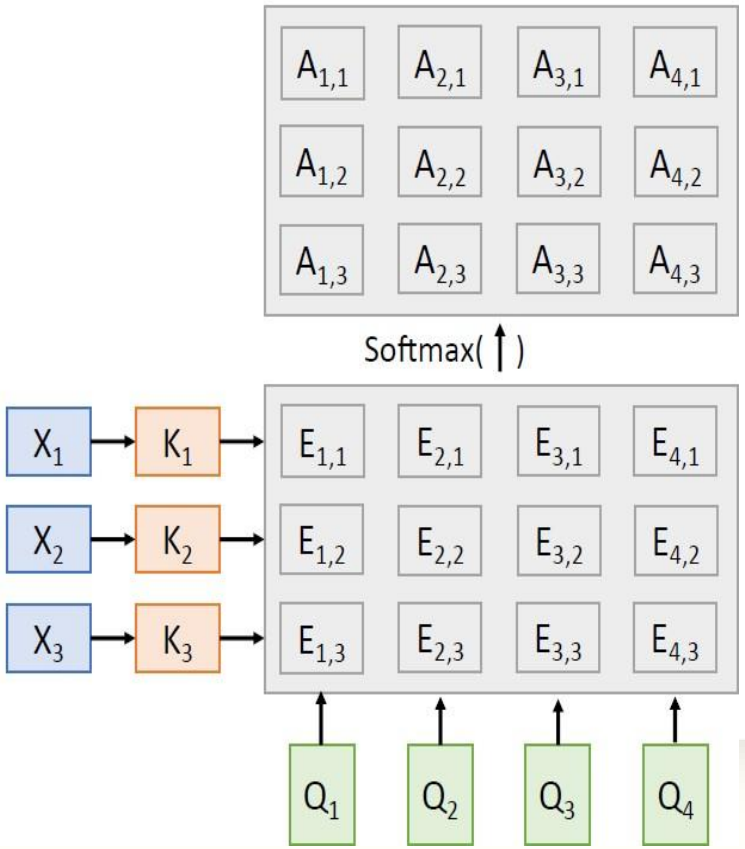
Computation:
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T$ (Shape: $N_Q \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: A = softmax(E, dim=1) (Shape: $N_Q \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_Q \times D_V$) $Y_i = \sum_j A_{i,j} V_j$



Attention Layer

Inputs:
Query vectors: Q (Shape: N_Q x D_Q)
Input vectors: X (Shape: N_X x D_X)
Key matrix: W_K (Shape: D_X x D_Q)
Value matrix: W_V (Shape: D_X x D_V)

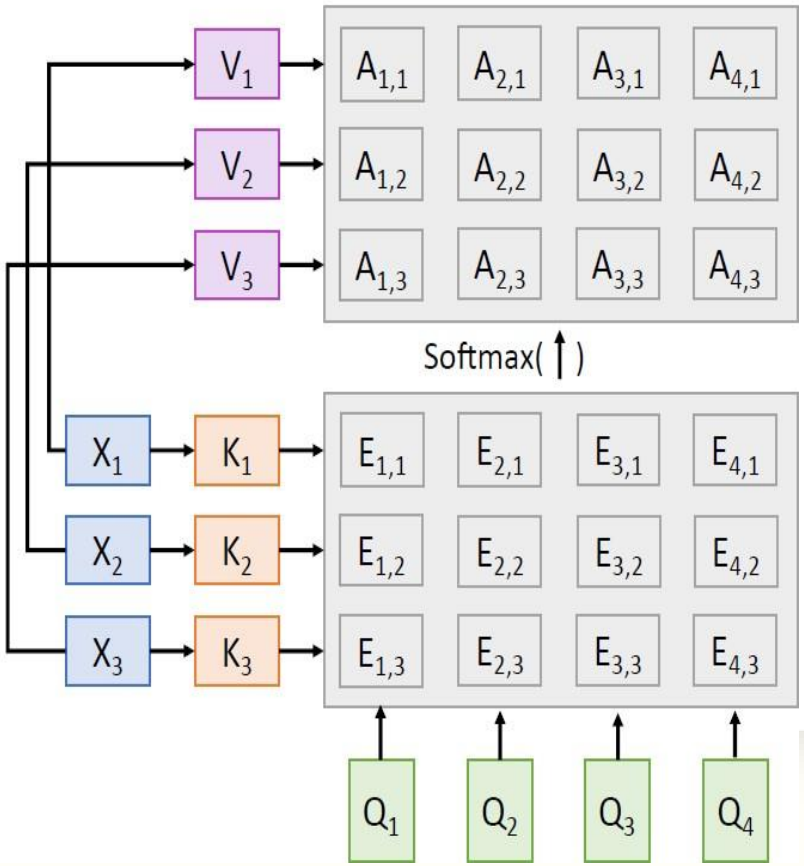
Computation:
Key vectors: $K = XW_K$ (Shape: N_X x D_Q)
Value Vectors: $V = XW_V$ (Shape: N_X x D_V)
Similarities: $E = QK^T$ (Shape: N_Q x N_X) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: A = softmax(E, dim=1) (Shape: N_Q x N_X)
Output vectors: Y = AV (Shape: N_Q x D_V) $Y_i = \sum_j A_{i,j} V_j$



Attention Layer

Inputs:
Query vectors: Q (Shape: N_Q x D_Q)
Input vectors: X (Shape: N_X x D_X)
Key matrix: W_K (Shape: D_X x D_Q)
Value matrix: W_V (Shape: D_X x D_V)

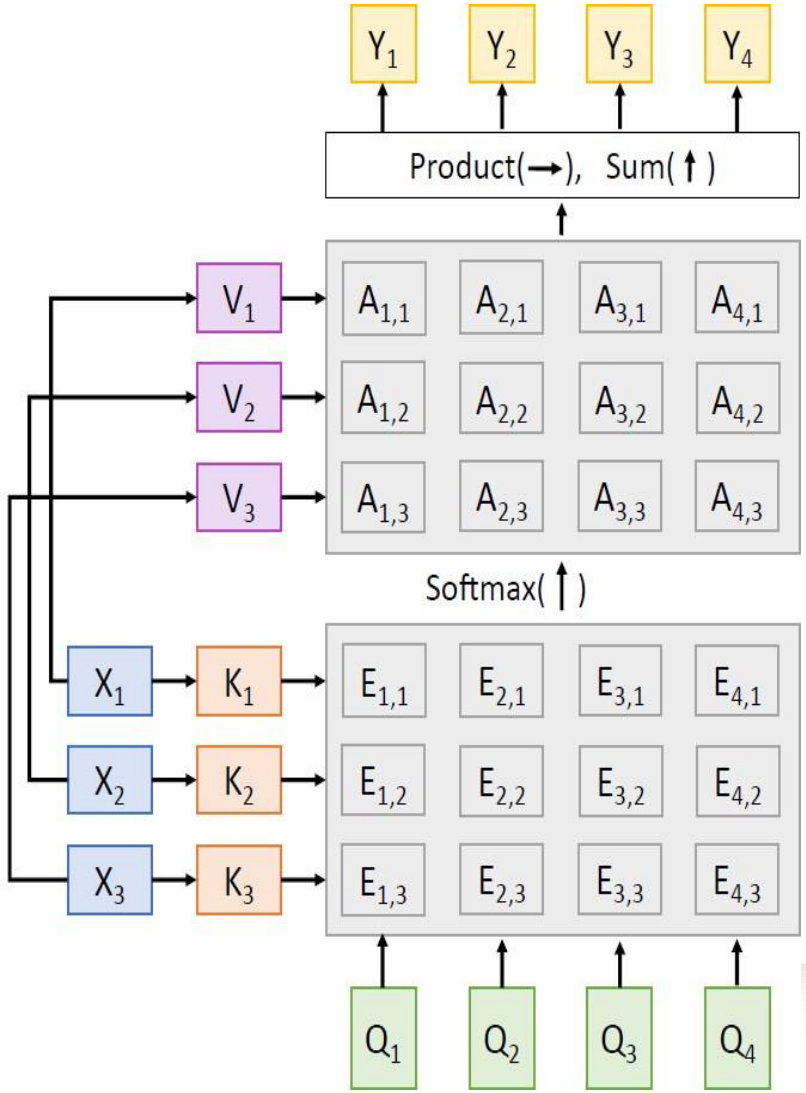
Computation:
Key vectors: $K = XW_K$ (Shape: N_X x D_Q)
Value Vectors: $V = XW_V$ (Shape: N_X x D_V)
Similarities: $E = QK^T$ (Shape: N_Q x N_X) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: A = softmax(E, dim=1) (Shape: N_Q x N_X)
Output vectors: Y = AV (Shape: N_Q x D_V) $Y_i = \sum_j A_{i,j} V_j$



Attention Layer

Inputs:
Query vectors: Q (Shape: N_Q x D_Q)
Input vectors: X (Shape: N_X x D_X)
Key matrix: W_K (Shape: D_X x D_Q)
Value matrix: W_V (Shape: D_X x D_V)

Computation:
Key vectors: $K = XW_K$ (Shape: N_X x D_Q)
Value Vectors: $V = XW_V$ (Shape: N_X x D_V)
Similarities: $E = QK^T$ (Shape: N_Q x N_X) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: A = softmax(E, dim=1) (Shape: N_Q x N_X)
Output vectors: Y = AV (Shape: N_Q x D_V) $Y_i = \sum_j A_{i,j} V_j$



Self-Attention Layer

One **query** per **input vector**

★ *입력벡터마다 하나의* **Query** *벡터를 사용한다.*

Inputs:

Input vectors: X (Shape: $N_X \times D_X$)

Key matrix: W_K (Shape: $D_X \times D_Q$)

Value matrix: W_V (Shape: $D_X \times D_V$)

Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:

Query vectors: $Q = XW_Q$

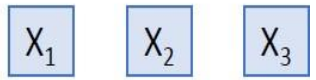
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)

Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)

Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$

Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)

Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$

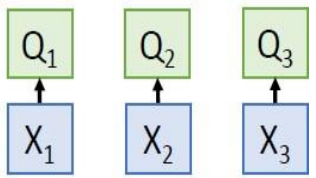


Self-Attention Layer

One **query** per **input vector**

Inputs:
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)
Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:
Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$

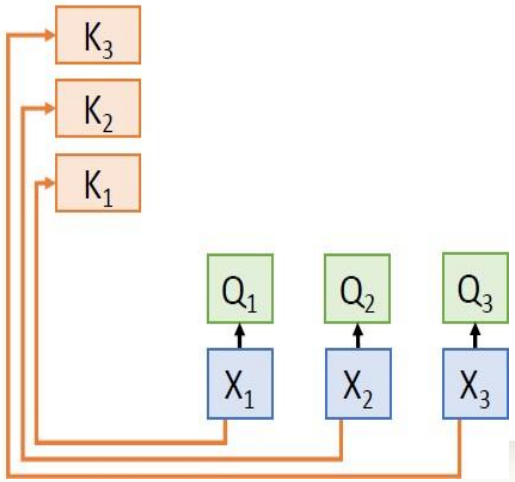


Self-Attention Layer

One **query** per **input vector**

Inputs:
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)
Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:
Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$

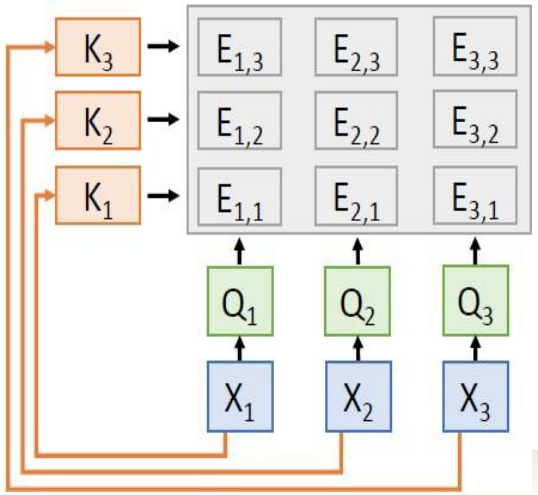


Self-Attention Layer

One **query** per **input vector**

Inputs:
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)
Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:
Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$

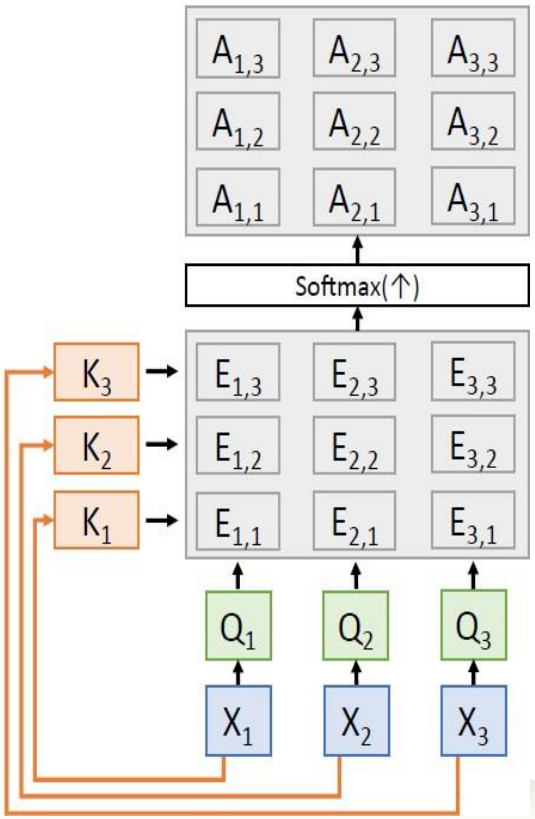


Self-Attention Layer

One **query** per **input vector**

Inputs:
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)
Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:
Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$

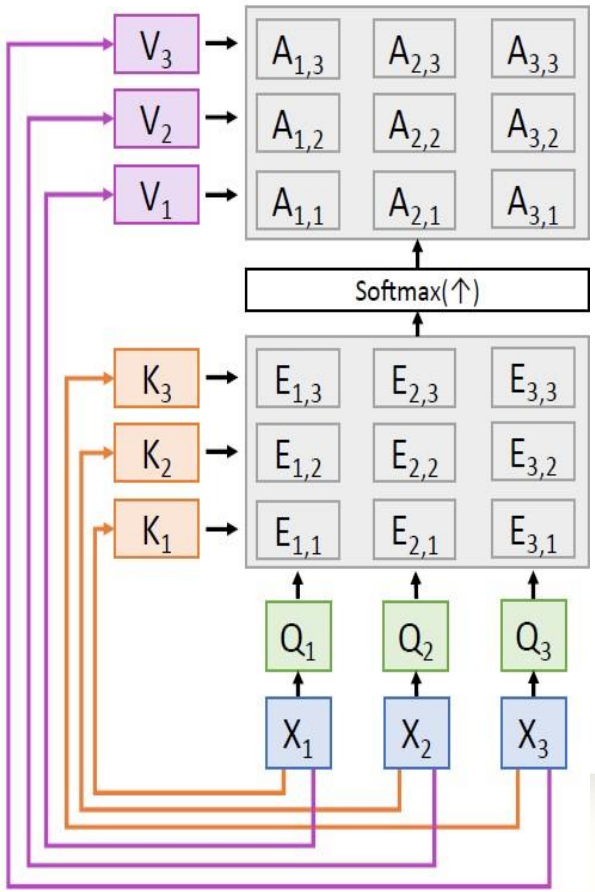


Self-Attention Layer

One **query** per **input vector**

Inputs:
Input vectors: X (Shape: N_X x D_X)
Key matrix: W_K (Shape: D_X x D_Q)
Value matrix: W_V (Shape: D_X x D_V)
Query matrix: W_Q (Shape: D_X x D_Q)

Computation:
Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: N_X x D_Q)
Value Vectors: $V = XW_V$ (Shape: N_X x D_V)
Similarities: $E = QK^T$ (Shape: N_X x N_X) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: A = softmax(E, dim=1) (Shape: N_X x N_X)
Output vectors: Y = AV (Shape: N_X x D_V) $Y_i = \sum_j A_{i,j} V_j$

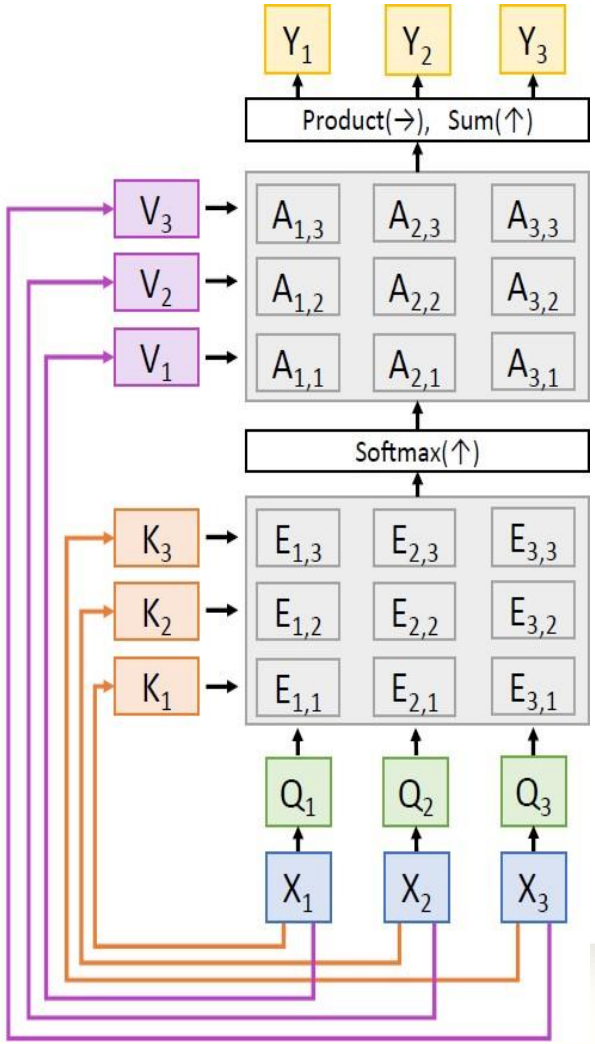


Self-Attention Layer

One **query** per **input vector**

Inputs:
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)
Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:
Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$



Self-Attention Layer

Consider **permuting** the input vectors:

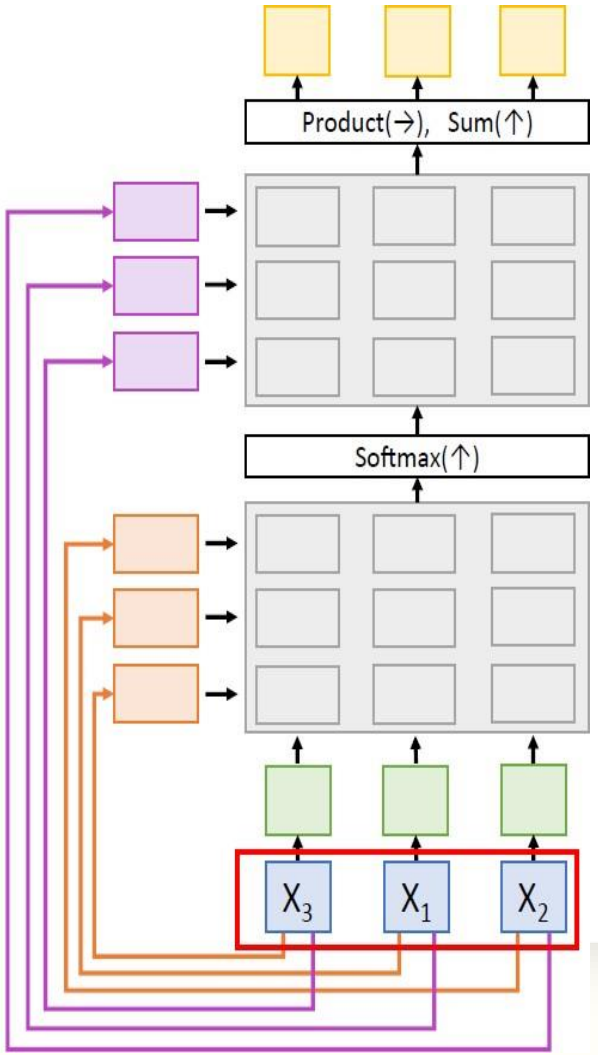
★ *입력벡터의 순열을 고려*

Inputs:

Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)
Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:

Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$



Self-Attention Layer

Consider **permuting** the input vectors:

Queries and Keys will be the same, but permuted

★ ***key도 동일한 순열을 사용하며, 값은 동일하다***

Inputs:

Input vectors: X (Shape: $N_X \times D_X$)

Key matrix: W_K (Shape: $D_X \times D_Q$)

Value matrix: W_V (Shape: $D_X \times D_V$)

Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:

Query vectors: $Q = XW_Q$

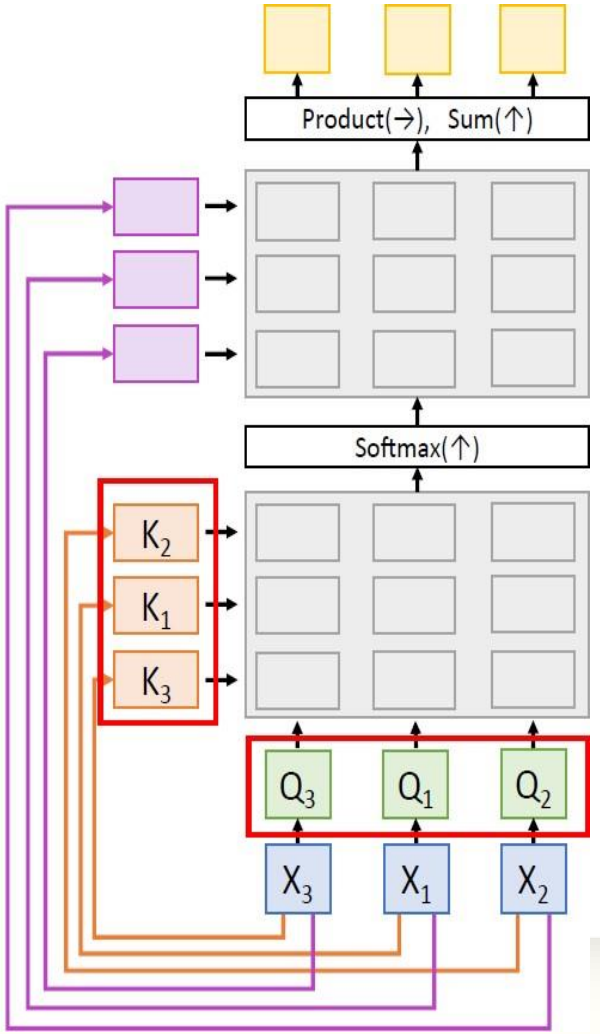
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)

Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)

Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$

Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)

Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$



Self-Attention Layer

Consider **permuting** the input vectors:

Similarities will be the same, but permuted

★ *유사도도 동일한 순열을 사용하며, 값은 동일하다*

Inputs:

Input vectors: X (Shape: $N_X \times D_X$)

Key matrix: W_K (Shape: $D_X \times D_Q$)

Value matrix: W_V (Shape: $D_X \times D_V$)

Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:

Query vectors: $Q = XW_Q$

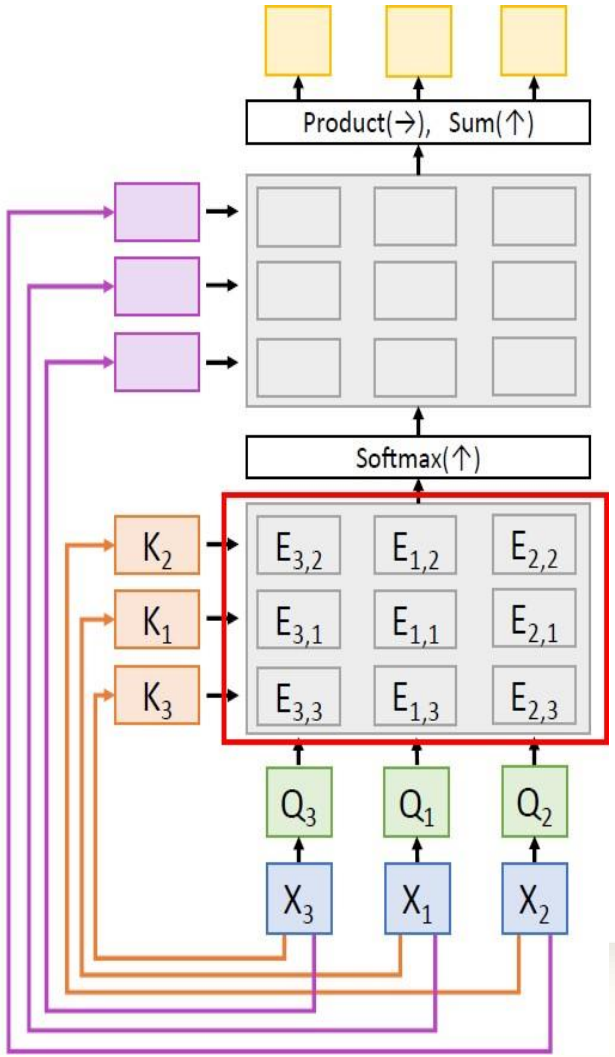
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)

Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)

Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$

Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)

Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$



Self-Attention Layer

Consider **permuting** the input vectors:

Attention weights will be the same, but permuted

★ **Attention** *가중치도 동일한 순열을 사용하며, 값은 동일하다*

Inputs:

Input vectors: X (Shape: $N_X \times D_X$)

Key matrix: W_K (Shape: $D_X \times D_Q$)

Value matrix: W_V (Shape: $D_X \times D_V$)

Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:

Query vectors: $Q = XW_Q$

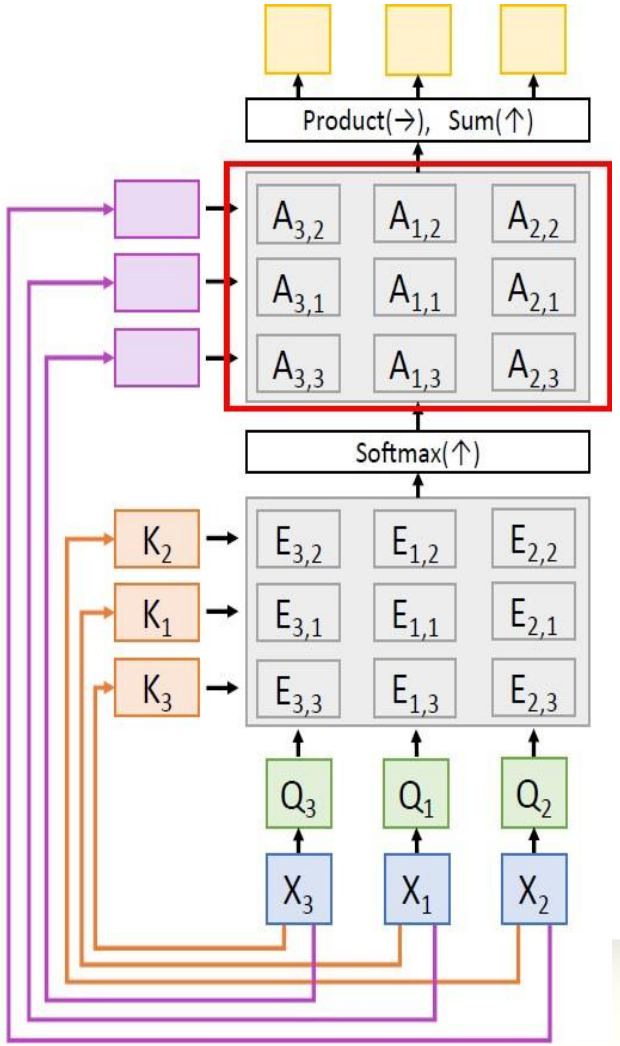
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)

Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)

Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$

Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)

Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$



Self-Attention Layer

Consider **permuting** the input vectors:

Values will be the same, but permuted

★ **value도 동일한 순열을 사용하며, 값은 동일하다**

Inputs:

Input vectors: X (Shape: $N_X \times D_X$)

Key matrix: W_K (Shape: $D_X \times D_Q$)

Value matrix: W_V (Shape: $D_X \times D_V$)

Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:

Query vectors: $Q = XW_Q$

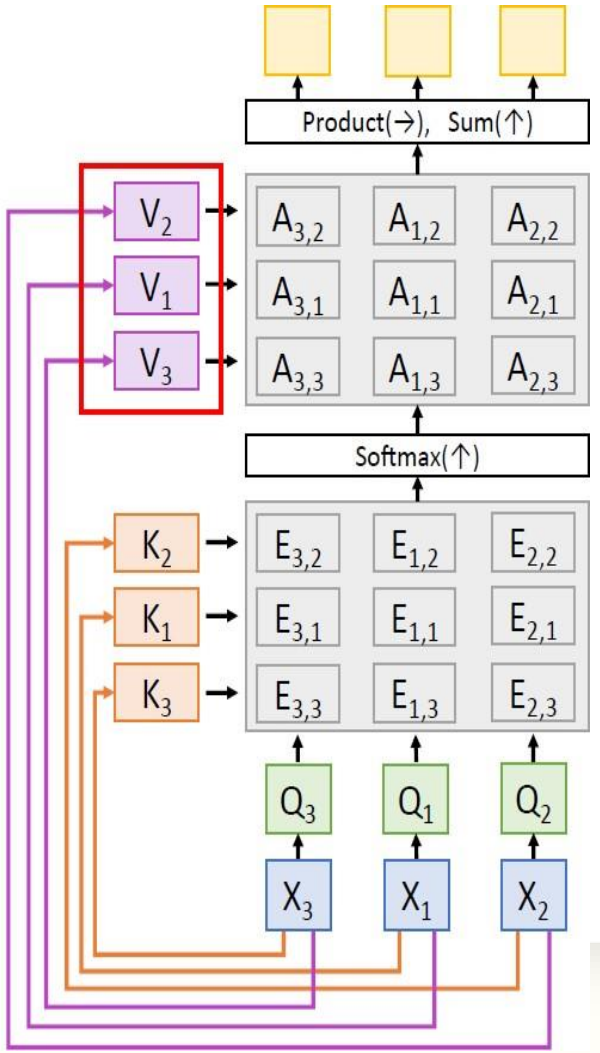
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)

Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)

Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$

Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)

Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$



Self-Attention Layer

Consider **permuting** the input vectors:

Outputs will be the same, but permuted

★ *출력도 동일한 순열을 사용하며, 값은 동일하다*

Inputs:

Input vectors: X (Shape: $N_X \times D_X$)

Key matrix: W_K (Shape: $D_X \times D_Q$)

Value matrix: W_V (Shape: $D_X \times D_V$)

Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:

Query vectors: $Q = XW_Q$

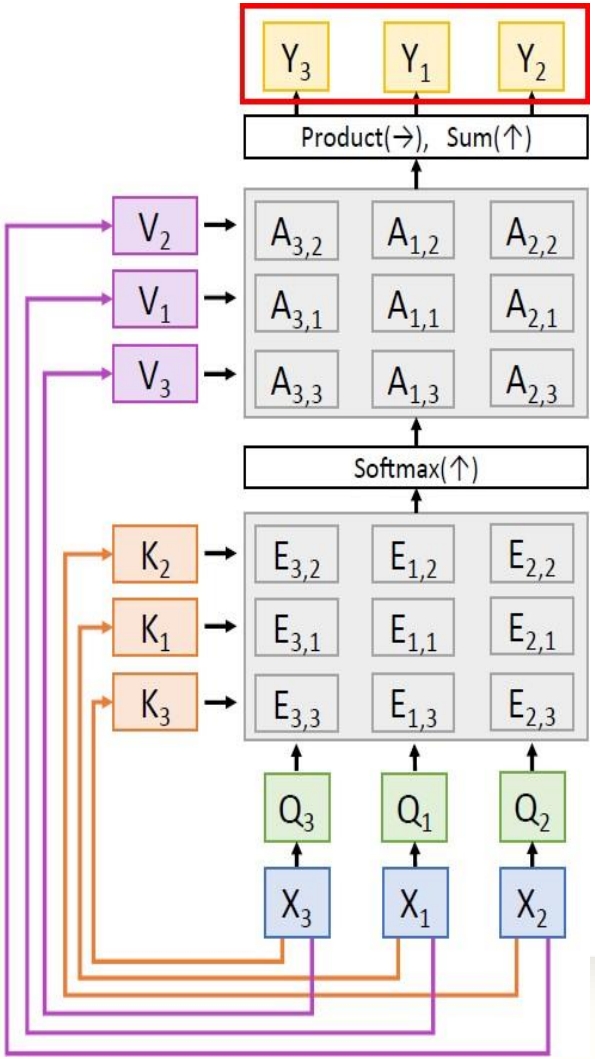
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)

Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)

Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$

Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)

Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$



Self-Attention Layer

Inputs:
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)
Query matrix: W_Q (Shape: $D_X \times D_Q$)

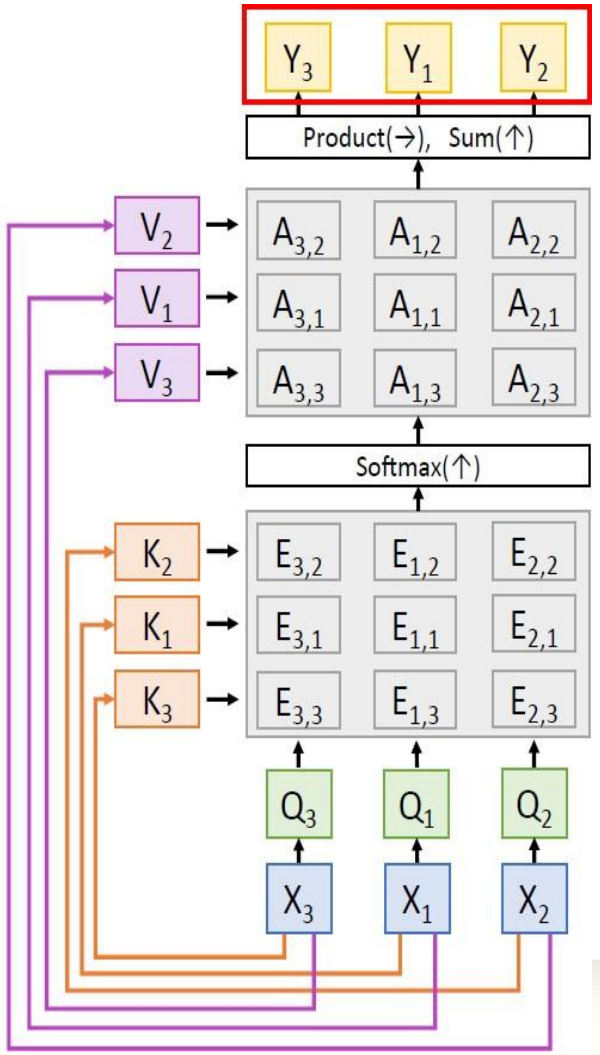
Computation:
Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$

Consider **permuting** the input vectors:

Outputs will be the same, but permuted

Self-attention layer is **Permutation Equivariant**
$f(s(x)) = s(f(x))$

Self-Attention layer works on **sets** of vectors



★ **self-attention*은 순열동등하다. 즉* f(s(x))= s(f(x))**
–self-attention*은 벡터셋에 대해 적용할 수 있다.*

Self-Attention Layer

Self attention doesn't "know" the order of the vectors it is processing!

Inputs:

Input vectors: X (Shape: $N_X \times D_X$)

Key matrix: W_K (Shape: $D_X \times D_Q$)

Value matrix: W_V (Shape: $D_X \times D_V$)

Query matrix: W_Q (Shape: $D_X \times D_Q$)

★ **self-attention은** ***벡터의 순서를 모른다.***

Computation:

Query vectors: $Q = XW_Q$

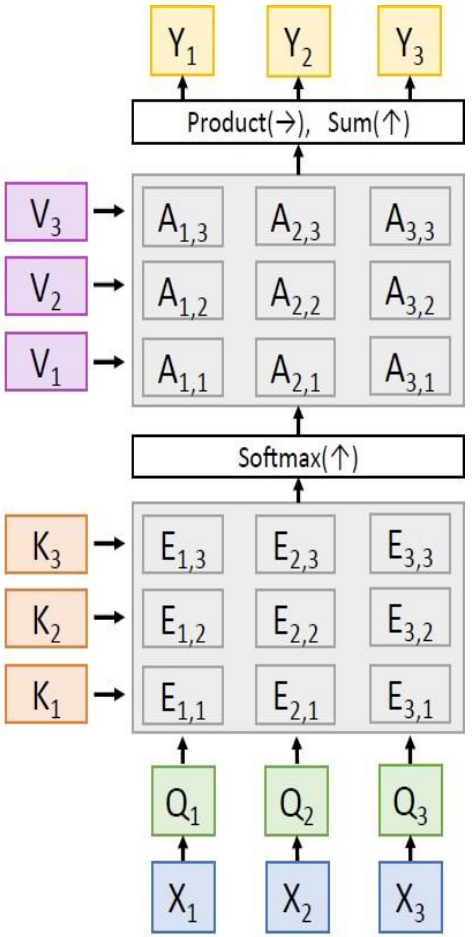
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)

Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)

Similarities: $E = QK^T$ (Shape: $N_X \times N_X$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$

Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)

Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$



Self-Attention Layer

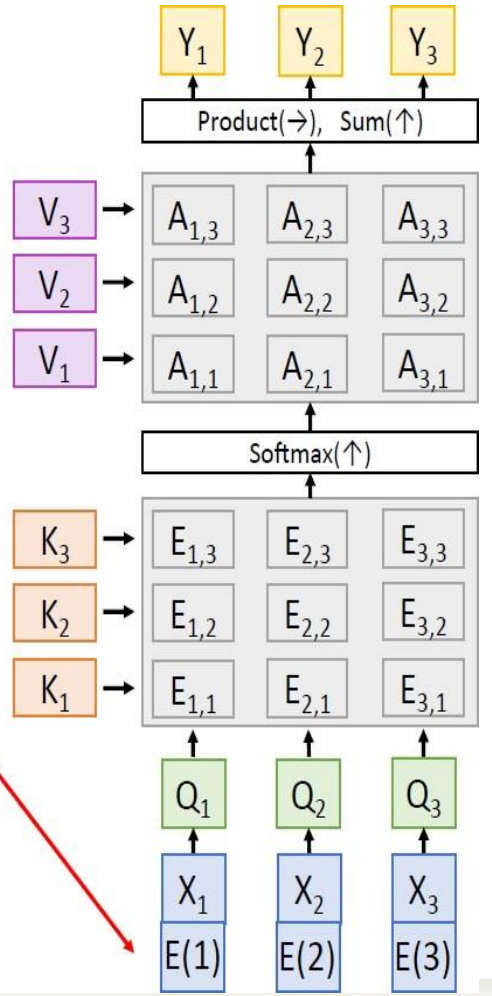
Inputs:
Input vectors: X (Shape: N_X x D_X)
Key matrix: W_K (Shape: D_X x D_Q)
Value matrix: W_V (Shape: D_X x D_V)
Query matrix: W_Q (Shape: D_X x D_Q)

Computation:
Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: N_X x D_Q)
Value Vectors: $V = XW_V$ (Shape: N_X x D_V)
Similarities: $E = QK^T$ (Shape: N_X x N_X) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: A = softmax(E, dim=1) (Shape: N_X x N_X)
Output vectors: $Y = AV$ (Shape: N_X x D_V) $Y_i = \sum_j A_{i,j} V_j$

Self attention doesn't "know" the order of the vectors it is processing!

In order to make processing position-aware, concatenate input with **positional encoding**

E can be learned lookup table, or fixed function



★ *위치를 고려하기 위해 위치 인코딩을 한다.* **E는 lookup table** *또는 함수를 통해 구한다.*

Masked Self-Attention Layer

Don't let vectors "look ahead" in the sequence

★ *"선견" 편향을 막기 위해 마스크를 사용*

Inputs:

Input vectors: X (Shape: N_X x D_X)

Key matrix: W_K (Shape: D_X x D_Q)

Value matrix: W_V (Shape: D_X x D_V)

Query matrix: W_Q (Shape: D_X x D_Q)

Computation:

Query vectors: $Q = XW_Q$

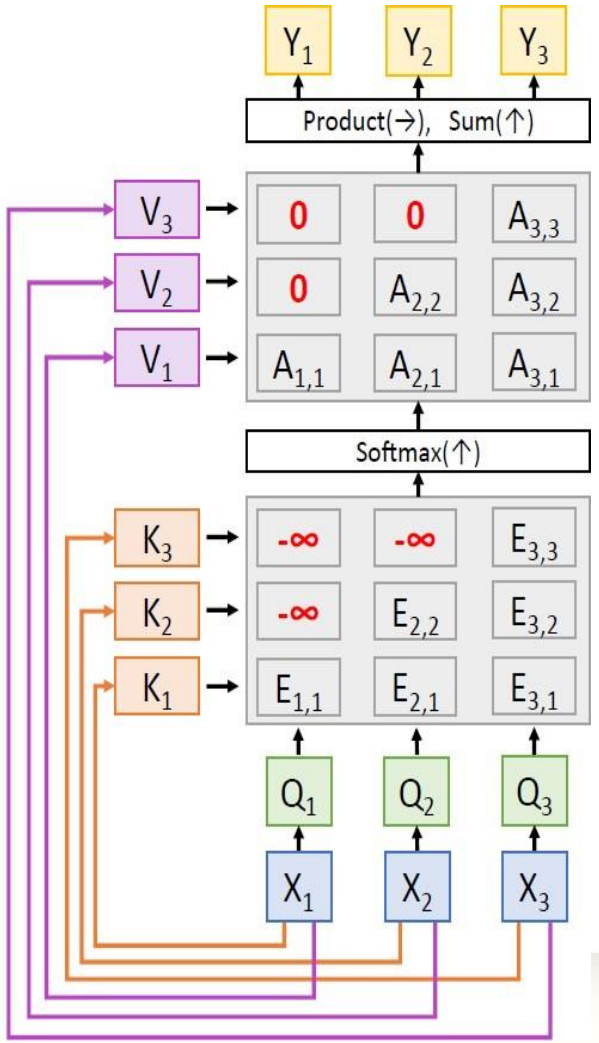
Key vectors: $K = XW_K$ (Shape: N_X x D_Q)

Value Vectors: $V = XW_V$ (Shape: N_X x D_V)

Similarities: $E = QK^T$ (Shape: N_X x N_X) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$

Attention weights: A = softmax(E, dim=1) (Shape: N_X x N_X)

Output vectors: $Y = AV$ (Shape: N_X x D_V) $Y_i = \sum_j A_{i,j} V_j$



Masked Self-Attention Layer

Don't let vectors "look ahead" in the sequence

Used for language modeling (predict next word)

Inputs:

Input vectors: X (Shape: N_X x D_X)

Key matrix: W_K (Shape: D_X x D_Q)

Value matrix: W_V (Shape: D_X x D_V)

Query matrix: W_Q (Shape: D_X x D_Q)

★ *언어모델에서 다음 단어 예측 및 시계열에서 미래 값 예측에 사용*

Computation:

Query vectors: $Q = XW_Q$

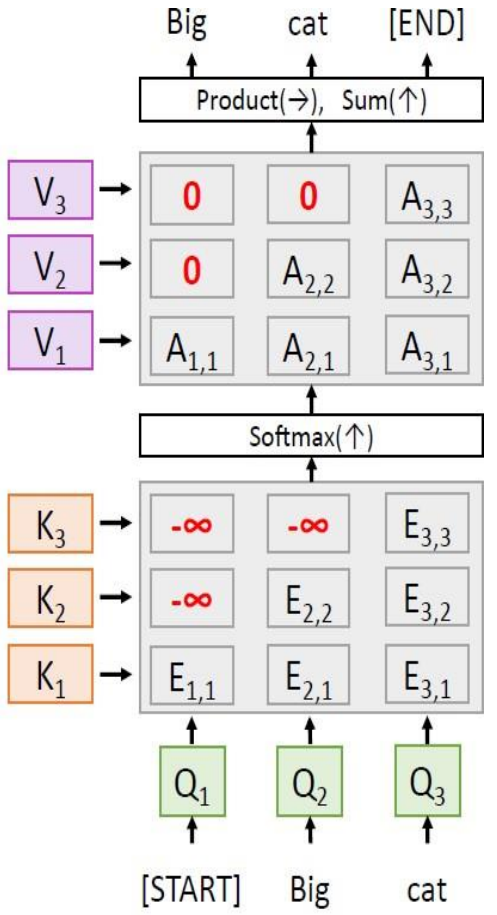
Key vectors: $K = XW_K$ (Shape: N_X x D_Q)

Value Vectors: $V = XW_V$ (Shape: N_X x D_V)

Similarities: $E = QK^T$ (Shape: N_X x N_X) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$

Attention weights: A = softmax(E, dim=1) (Shape: N_X x N_X)

Output vectors: $Y = AV$ (Shape: N_X x D_V) $Y_i = \sum_j A_{i,j} V_j$



Multihead Self-Attention Layer

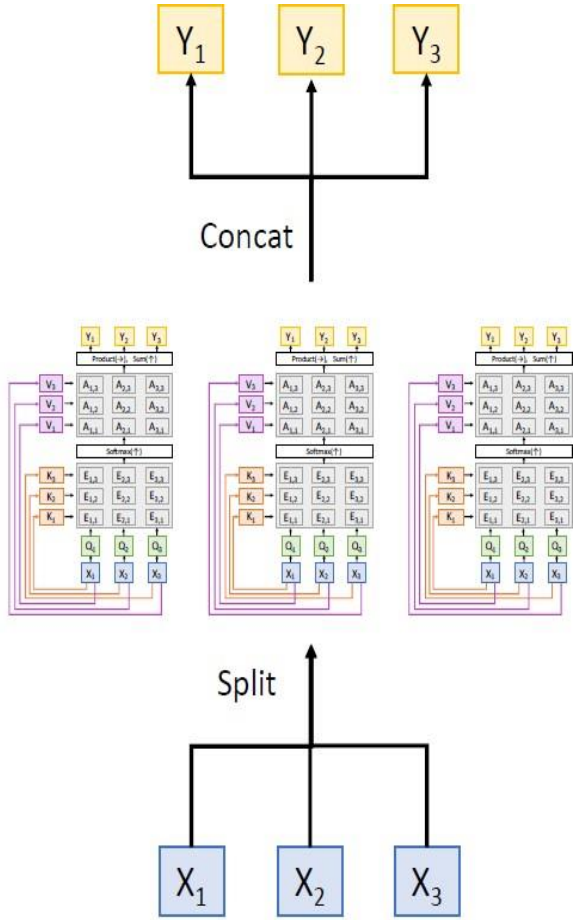
Use H independent
"Attention Heads" in parallel

Inputs:
Input vectors: X (Shape: N_X x D_X)
Key matrix: W_K (Shape: D_X x D_Q)
Value matrix: W_V (Shape: D_X x D_V)
Query matrix: W_Q (Shape: D_X x D_Q)

★ **H*개의 독립* attention head*를 병렬로 사용***

Hyperparameters:
Query dimension D_Q
Number of heads H

Computation:
Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: N_X x D_Q)
Value Vectors: $V = XW_V$ (Shape: N_X x D_V)
Similarities: $E = QK^T$ (Shape: N_X x N_X) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$
Attention weights: A = softmax(E, dim=1) (Shape: N_X x N_X)
Output vectors: Y = AV (Shape: N_X x D_V) $Y_i = \sum_j A_{i,j} V_j$



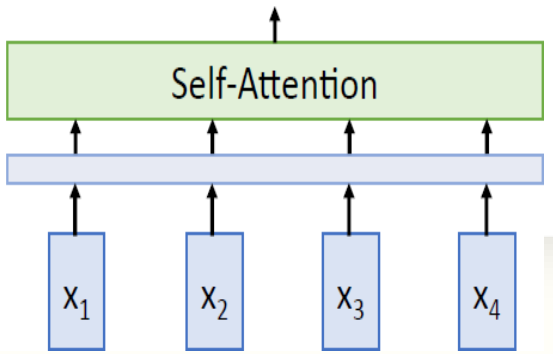
The Transformer

x_1 x_2 x_3 x_4

The Transformer

★ ***모는 벡터가 상호 영향***

All vectors interact with each other

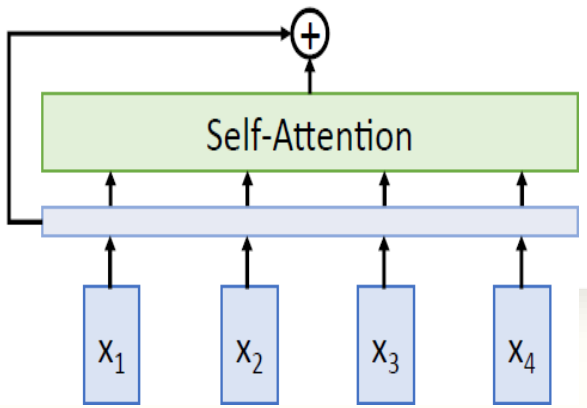


The Transformer

★ ***잔여 연결 구조 추가***

Residual connection

All vectors interact with each other



The Transformer

★ *층 정규화 추가*

Recall **Layer Normalization**:

Given $h_1, \ldots, h_N$ (Shape: D)

scale: γ (Shape: D)

shift: β (Shape: D)

$\mu_i = (1/D)\sum_j h_{i,j}$ (scalar)

$\sigma_i = (\sum_j (h_{i,j} - \mu_i)^2)^{1/2}$ (scalar)

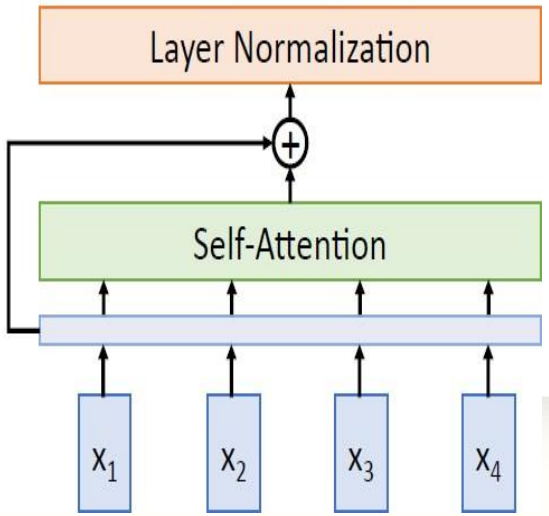
$z_i = (h_i - \mu_i) / \sigma_i$

$y_i = \gamma * z_i + \beta$

Ba et al, 2016

Residual connection

All vectors interact with each other



The Transformer

Recall **Layer Normalization**:

Given $h_1, \ldots, h_N$ (Shape: D)

scale: γ (Shape: D)

shift: β (Shape: D)

$\mu_i = (1/D)\sum_j h_{i,j}$ (scalar)

$\sigma_i = (\sum_j (h_{i,j} - \mu_i)^2)^{1/2}$ (scalar)

$z_i = (h_i - \mu_i) / \sigma_i$

$y_i = \gamma * z_i + \beta$

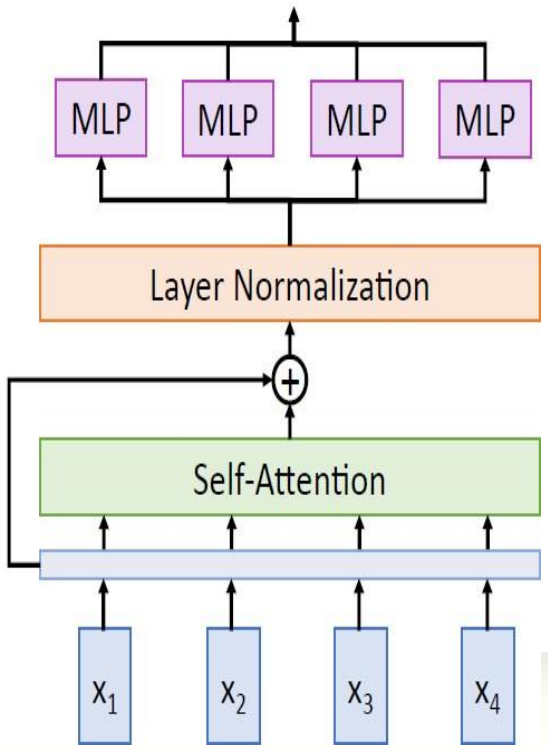
Ba et al, 2016

★ *각 벡터에 대해* **MLP** *독립적으로 적용*

MLP independently on each vector

Residual connection

All vectors interact with each other



The Transformer

★ *잔여 연결 구조 추가*

Recall **Layer Normalization**:

Given $h_1, \ldots, h_N$ (Shape: D)

scale: γ (Shape: D)

shift: β (Shape: D)

$\mu_i = (1/D)\sum_j h_{i,j}$ (scalar)

$\sigma_i = (\sum_j (h_{i,j} - \mu_i)^2)^{1/2}$ (scalar)

$z_i = (h_i - \mu_i) / \sigma_i$

$y_i = \gamma * z_i + \beta$

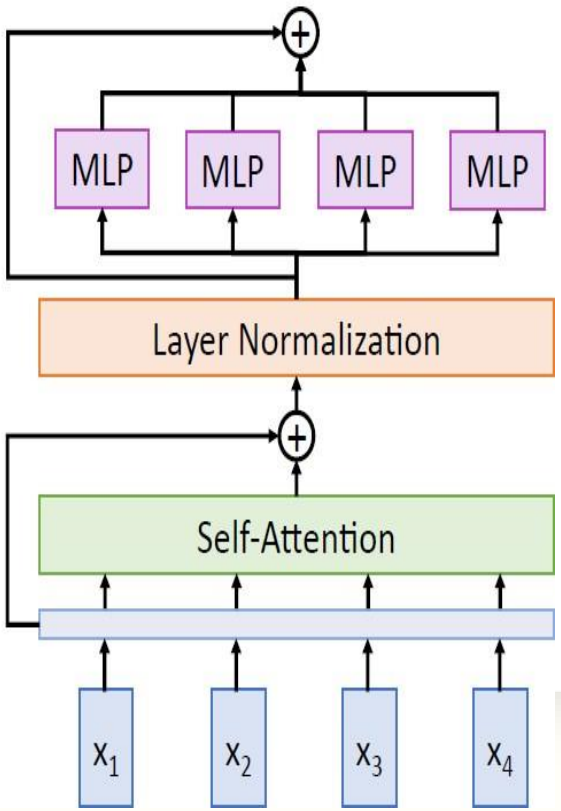
Ba et al, 2016

Residual connection

MLP independently on each vector

Residual connection

All vectors interact with each other



The Transformer

Recall **Layer Normalization**:

Given $h_1, \ldots, h_N$ (Shape: D)

scale: γ (Shape: D)

shift: β (Shape: D)

$\mu_i = (1/D)\sum_j h_{i,j}$ (scalar)

$\sigma_i = (\sum_j (h_{i,j} - \mu_i)^2)^{1/2}$ (scalar)

$z_i = (h_i - \mu_i) / \sigma_i$

$y_i = \gamma * z_i + \beta$

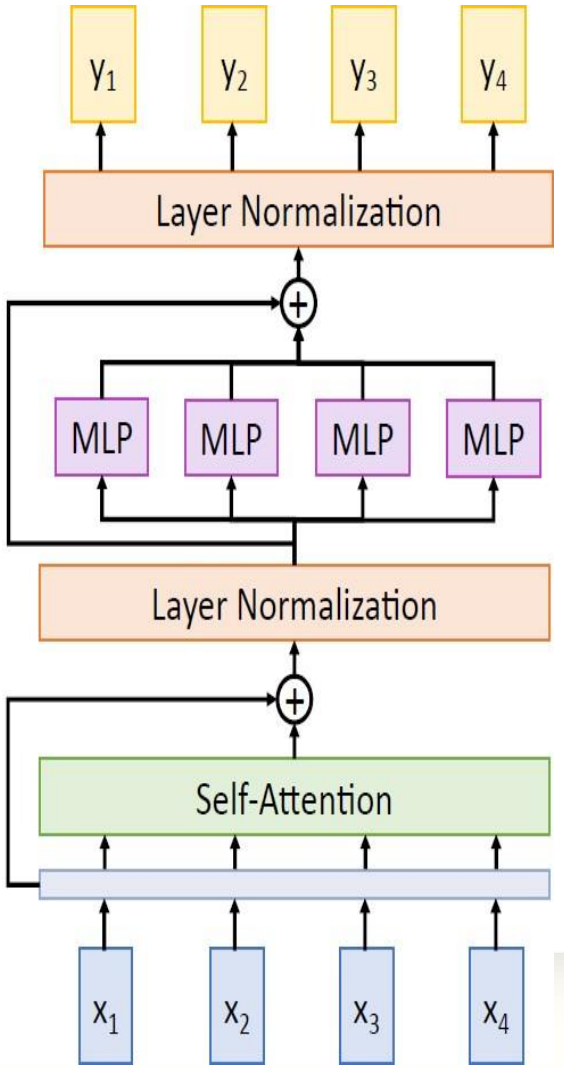
Ba et al, 2016

Residual connection

MLP independently on each vector

Residual connection

All vectors interact with each other



The Transformer

Transformer Block:

Input: Set of vectors x

Output: Set of vectors y

Self-attention is the only interaction between vectors!

Layer norm and MLP work independently per vector

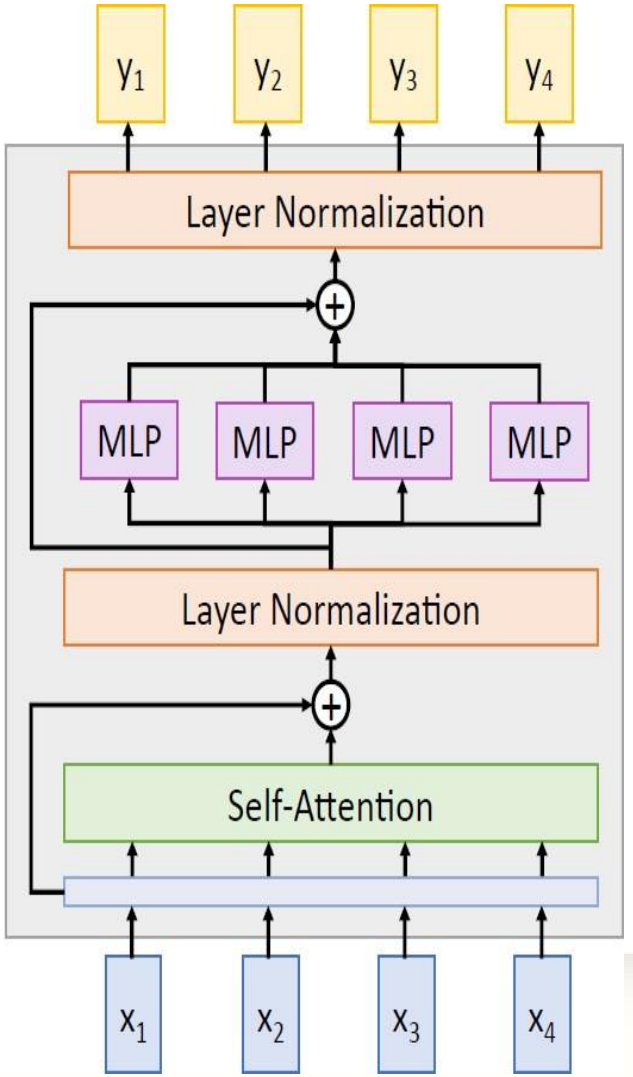
Highly scalable, highly parallelizable

★ *트랜스포머 블럭*
- *입력*: **X**
- *출력*: **y**

★ **self-attention** *벡터간의 상호작용만*

★ *층정규화와* **MLP***는 벡터별로 독립적으로 적용*

★ *규모확장+ 병렬화*



The Transformer

Transformer Block:

Input: Set of vectors x

Output: Set of vectors y

Self-attention is the only interaction between vectors!

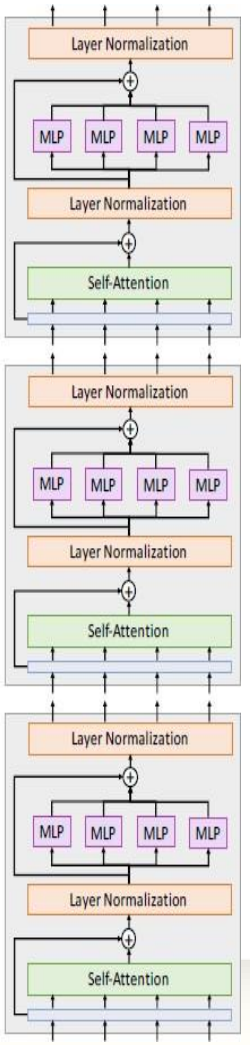
Layer norm and MLP work independently per vector

Highly scalable, highly parallelizable

★ *트랜스포머는 일련의 트랜스포머 블럭이다.*

A **Transformer** is a sequence of transformer blocks

Vaswani et al:
12 blocks, D_Q=512, 6 heads



★ *이미지넷에서 봤던 혁신을* **NLP** *분야에서 경험*

The Transformer: Transfer Learning

"ImageNet Moment for Natural Language Processing"

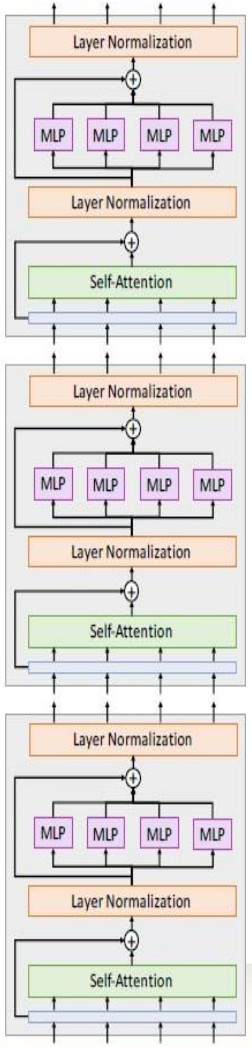
Pretraining:

Download a lot of text from the internet

Train a giant Transformer model for language modeling

Finetuning:

Fine-tune the Transformer on your own NLP task

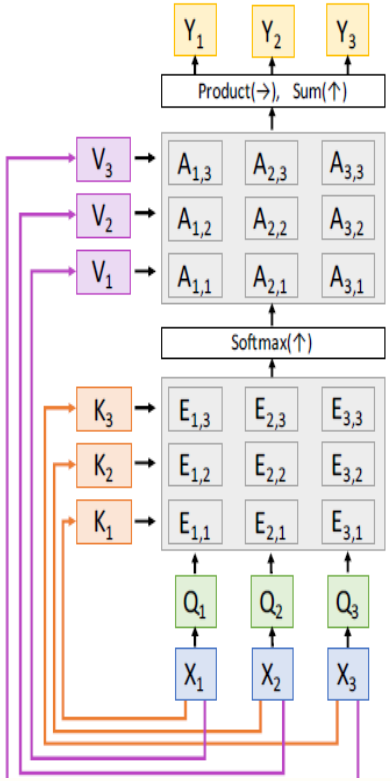


★ **RNN+attention -> Self-Attention -> Transformer(attention만 사용)**

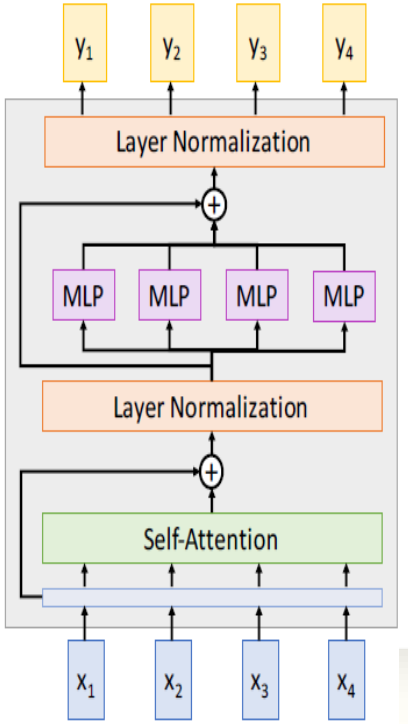
Adding **Attention** to RNN models lets them look at different parts of the input at each timestep

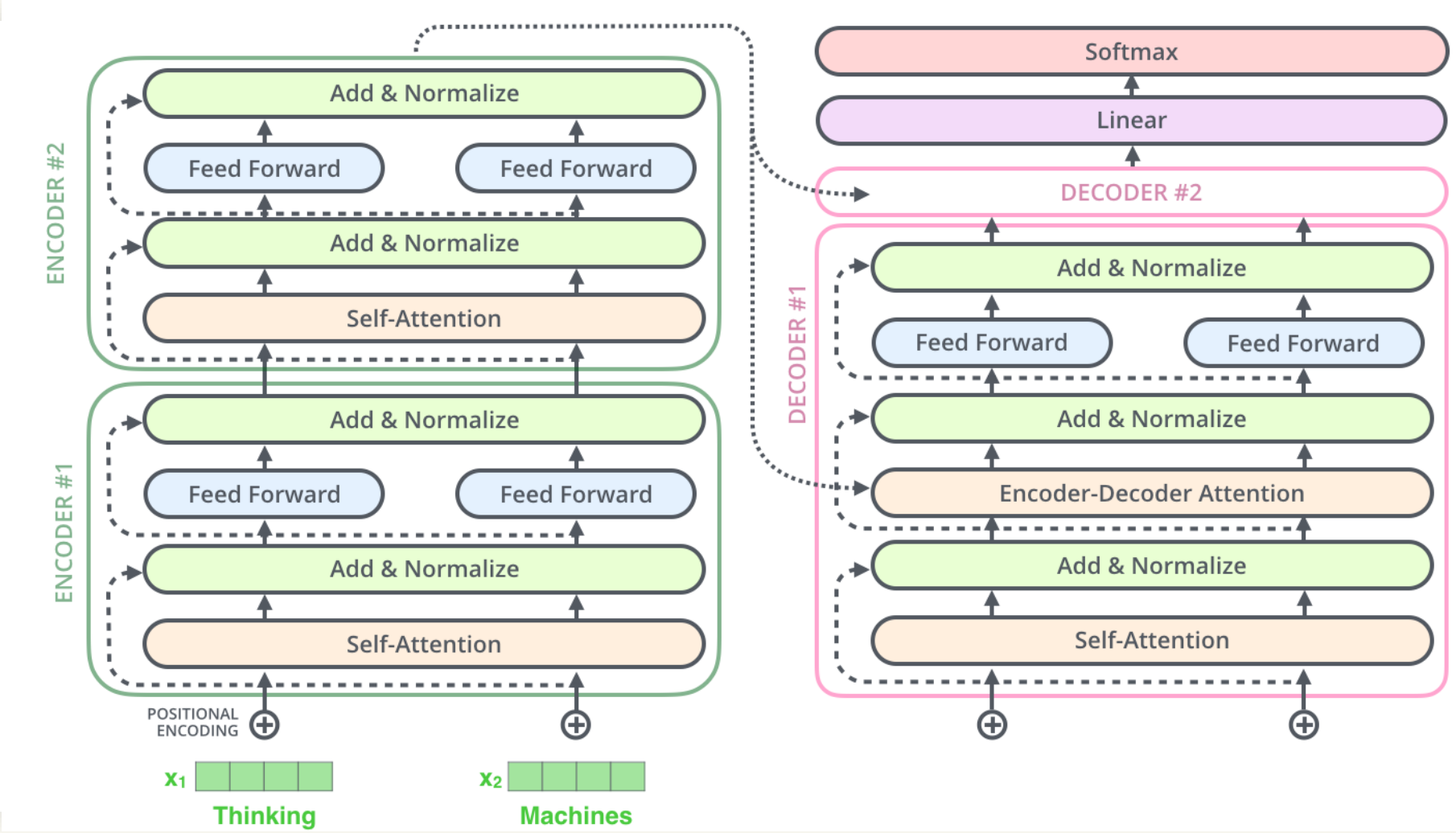
A dog is standing on a hardwood floor.

Generalized **Self-Attention** is new, powerful neural network primitive

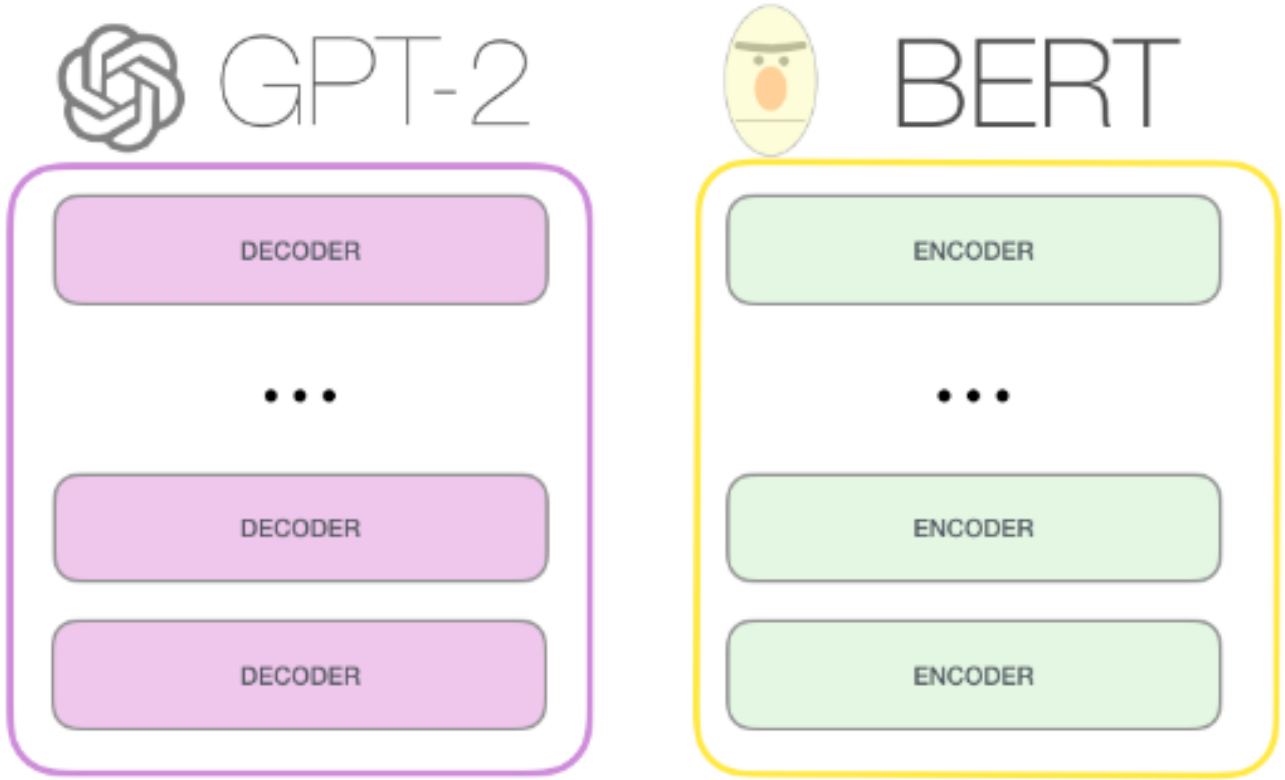


Transformers are a new neural network model that only uses attention




Softmax
Linear
DECODER #2
DECODER #1
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Encoder-Decoder Attention
Add & Normalize
Self-Attention
ENCODER #2
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Self-Attention
ENCODER #1
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Self-Attention
POSITIONAL ENCODING
x1
x2
Thinking
Machines

GPT-2, BERT

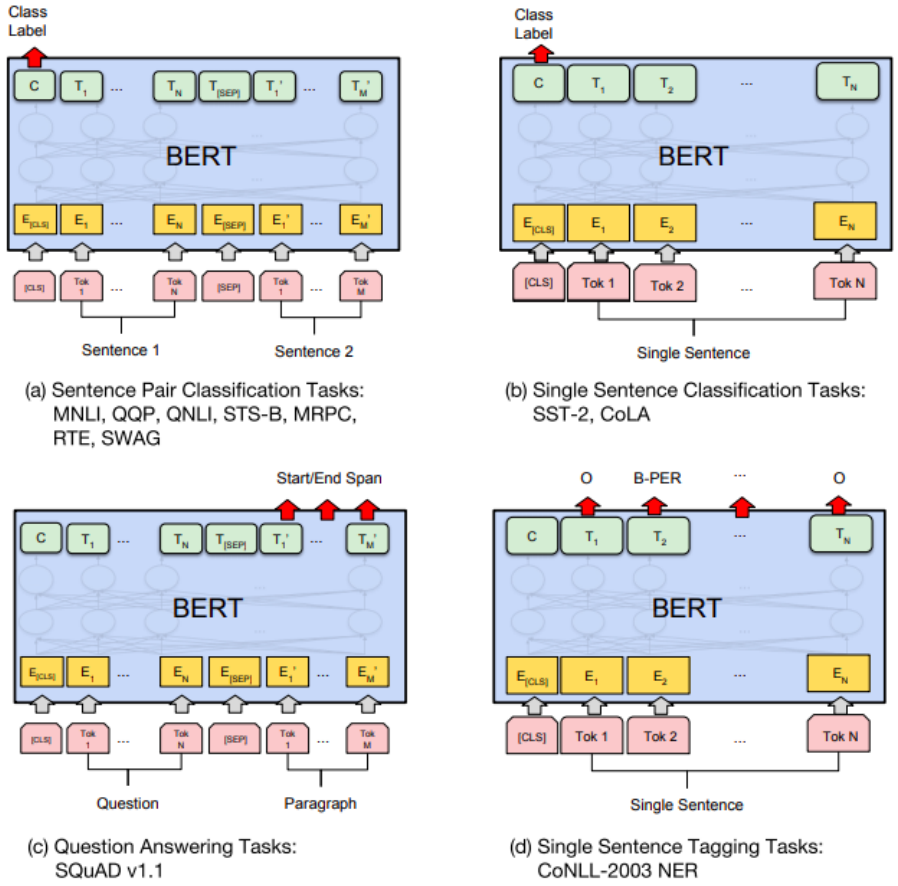

GPT-2
DECODER
DECODER
DECODER
BERT
ENCODER
ENCODER
ENCODER

BERT pretraining

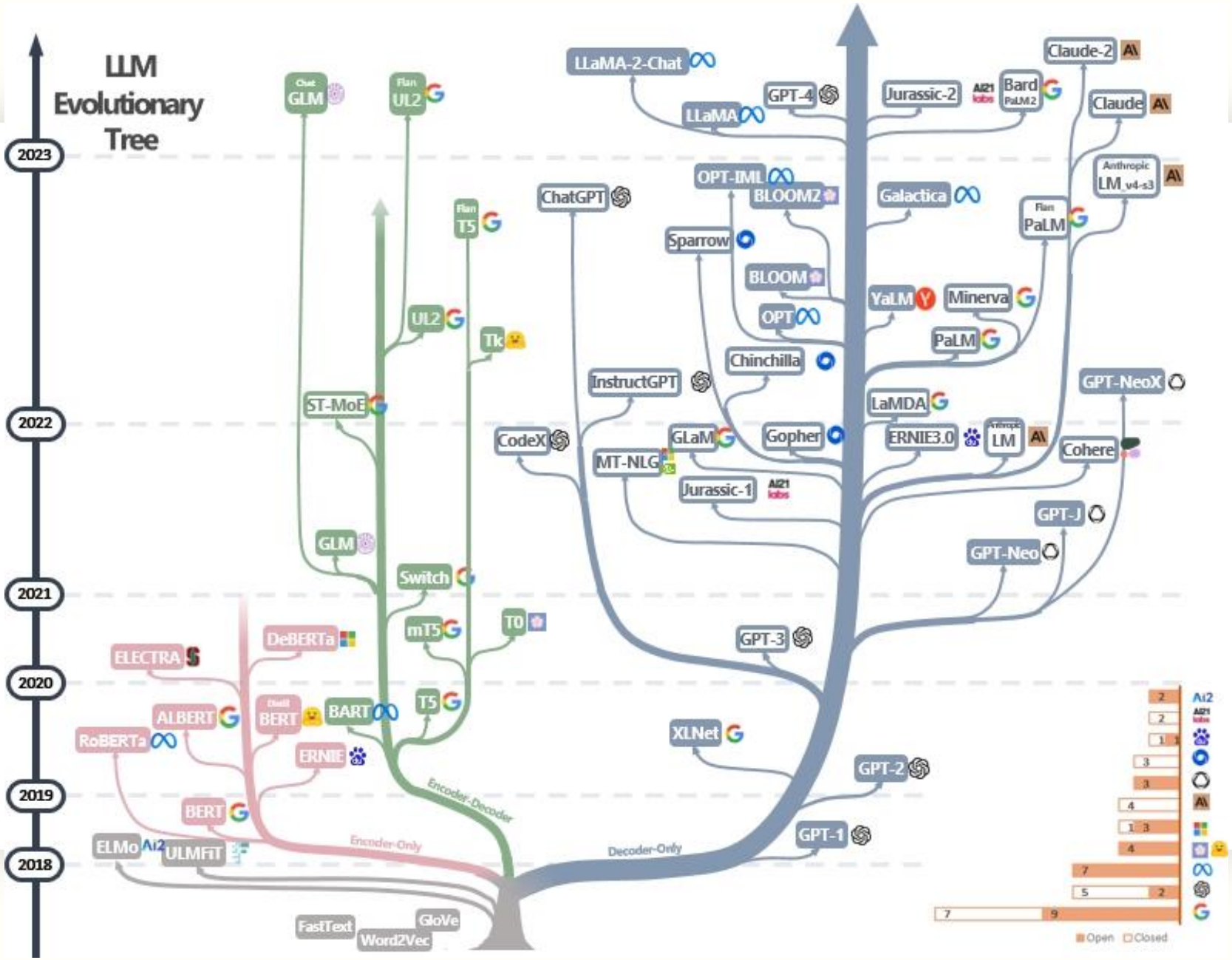
ULM-FiT (2018): Pre-training ideas, transfer learning in NLP.
ELMo: Bidirectional training (LSTM)
Transformer: Although used things from left, but still missing fr
om the right.
GPT: Use Transformer Decoder half.
BERT: Switches from Decoder to Encoder, so that it can use bot
h sides in training and invented corresponding training tasks:
masked language model

Fine-tuning BERT for other specific tasks

MNLI
QQP (Quaro Question Pairs)
Semantic equivalence)
QNLI (NL inference dataset)
STS-B (texture similarity)
MRPC (paraphrase, Microsoft)
RTE (textual entailment)
SWAG (commonsense inference)
SST-2 (sentiment)
CoLA (linguistic acceptability
SQuAD (question and answer)

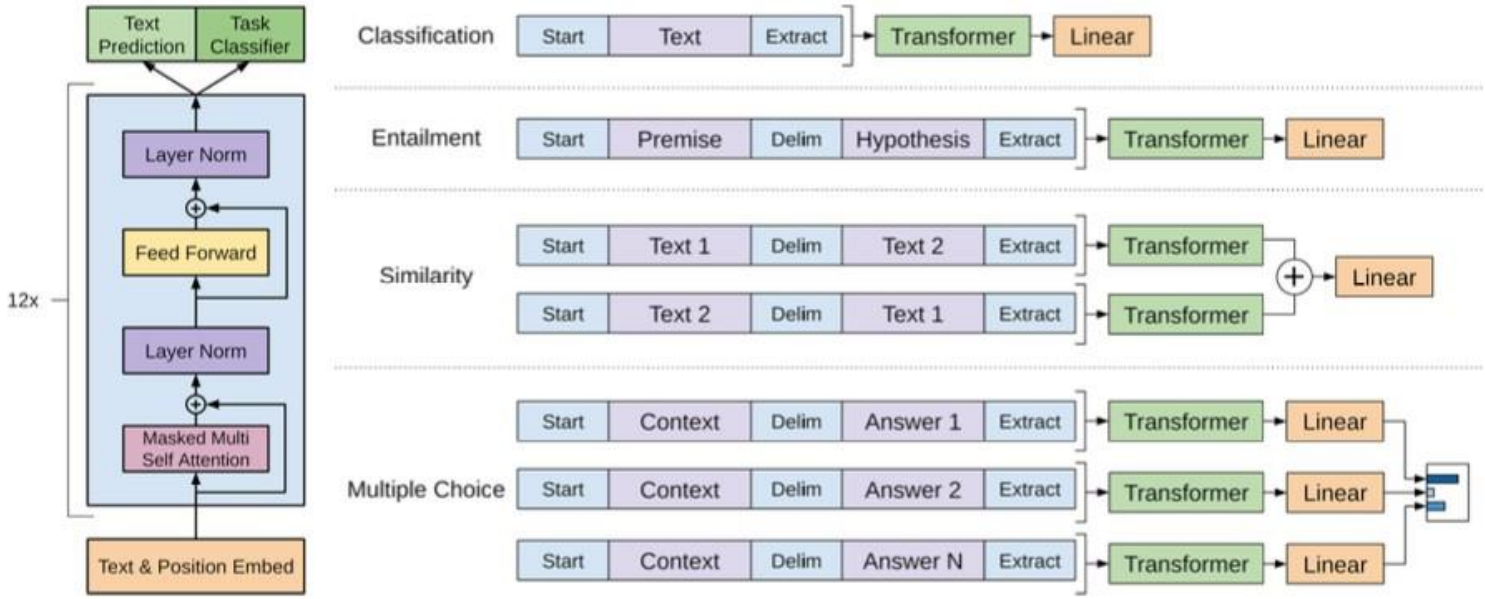


SST (Stanford sentiment treebank): 215k phrases with fine-grained sentiment labels in the parse trees of 11k sentences.



Yang et.al. 2023

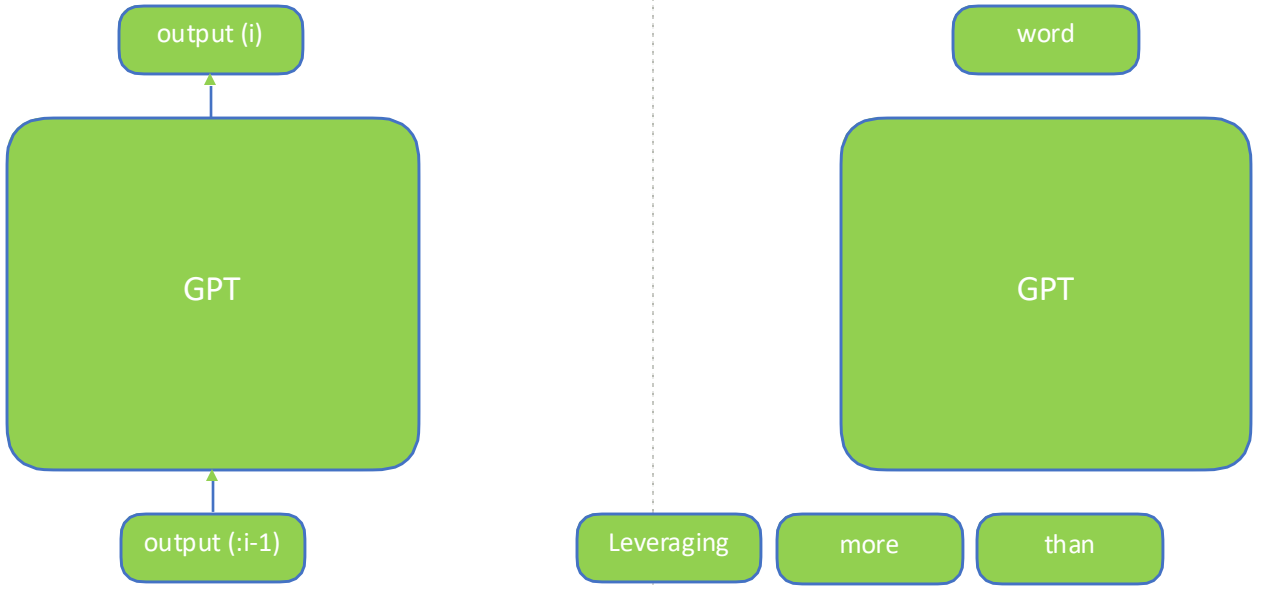
GPT: Task-specific input transformations



$[z; q; \$; a_k]$, z : 지문, q : 문제, $\$$: 구분자, a_k : k번째 정답 후보

Radford et al., Improving Language Understanding by Generative Pre-Training, 2017

GPT: outline

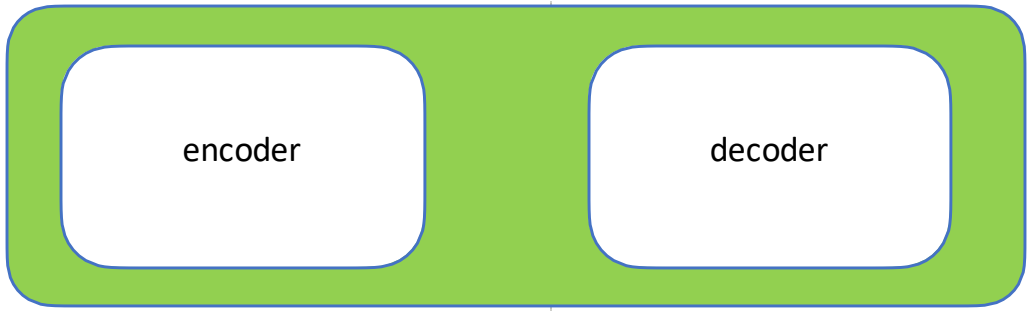


Radford et al., Improving Language Understanding by Generative Pre-Training, 2017

BART: outline

- BART: Bidirectional Auto-Regressive Transformer
  - denoising autoencoder for pretraining sequence-to-sequence model
  - Bidirectional encoder
  - Autoregressive decoder

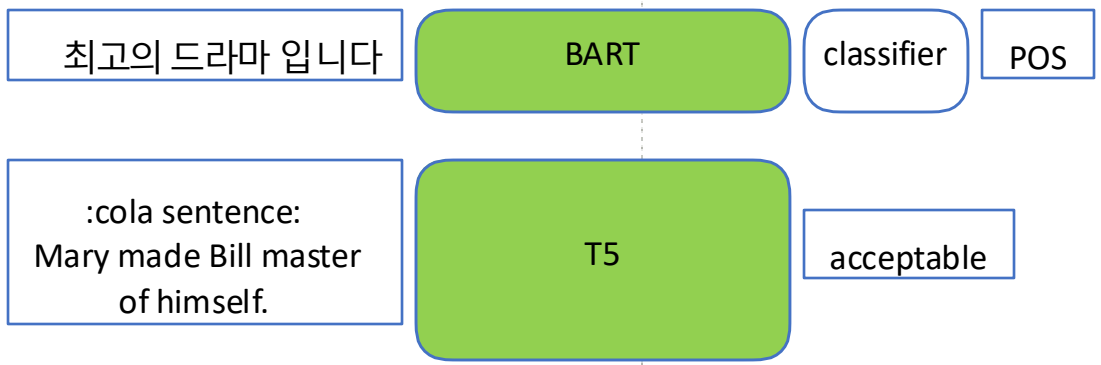
Part.1



Lewis et al., BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, 2019

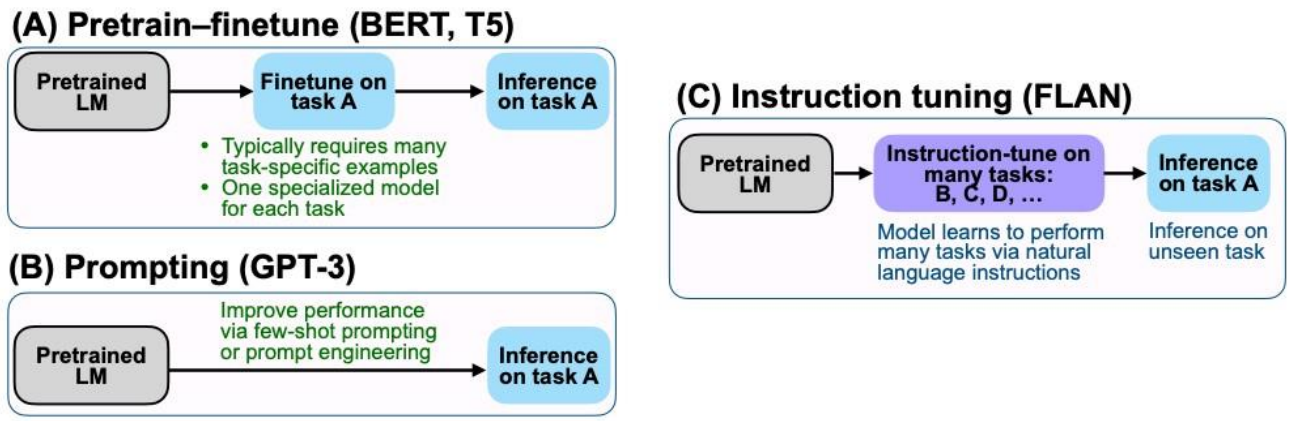
T5: outline

- Text to Text model



Raffel et al., Exploring the Limits of Transfer Learning with a Unified Text-to-text Transformer, 2019

FLAN: instruction tuning



Wei et al., Finetuned Language Models are Zero-shot Learners, 2021

Aligning LM Model

LMs should be..
* helpful
* honest
* harmless

Ouyang et al., Training Language Models to Follow Instructions with Human Feedback, 2022

Reinforcement Learning

Reinforcement Learning: 경험을 통해서 더 나은 결정을 내리도록 학습하는 방식

Agent: 학습하고자 하는 모델 State: Agent 의 현재 상태
Environment: Agent와 상호작용하는 환경 Reward: 환경이 Agent에게 주는 피드백

Ouyang et al., Training Language Models to Follow Instructions with Human Feedback, 2022

InstructGPT: outline

Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

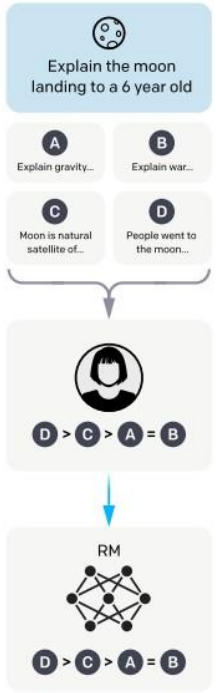
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

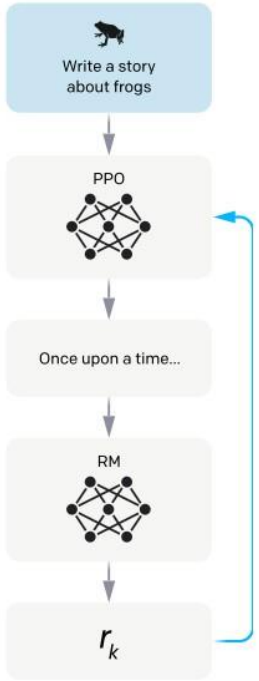
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

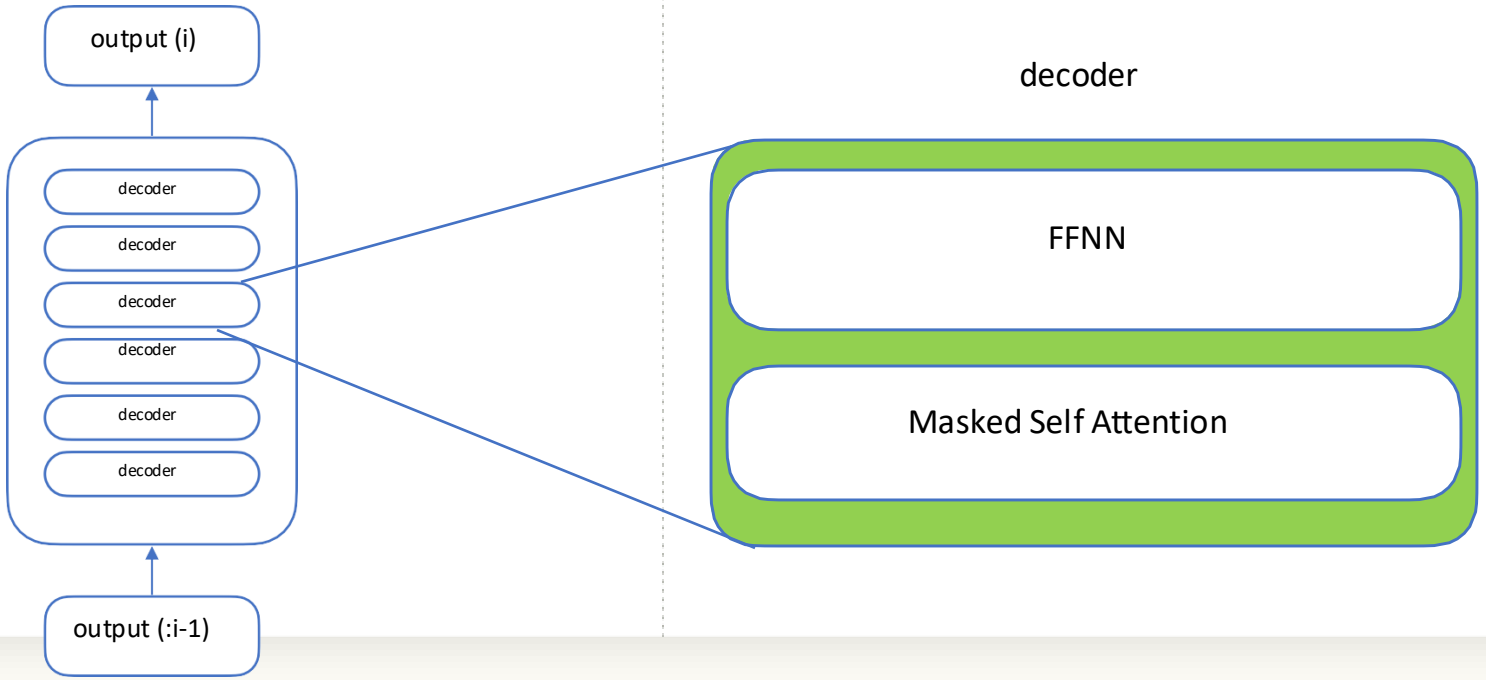
The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Touvron et al., LLaMA: Open and Efficient Foundation Language Models, 2023

Architecture: Decoder only Transformer



Touvron et al., LLaMA: Open and Efficient Foundation Language Models, 2023

Data & Training

LLaMA

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Unsupervised pretraining

GPT

Dataset	Quantity (tokens)
Common Crawl (filtered)	410 billion
WebText2	19 billion
Books1	12 billion
Books2	55 billion
Wikipedia	3 billion

Unsupervised pretraining
+ supervised finetuning
+ reinforcement learning w. human feedback

Drug discovery companies are customizing ChatGPT: here's how

Large language models are helping scientists to converse with artificial intelligence and even to generate potential drug targets.

By Neil Savage

Companies are adopting large language models to aid drug discovery.

CREDIT: REUTERS / ALAMY STOCK PHOTO

Much of the world has been transfixed in recent months by the appearance of text generation engines such as OpenAI's ChatGPT, artificial intelligence (AI) algorithms capable of producing text that seems as if it were written by a human. While tech companies like Microsoft and Google are focused on using such engines as a way to improve search and others worry they could cause a rash of plagiarized essays, fake news and bad poetry, biotechs are looking at these algorithms to bolster their businesses, as a method to contribute to drug discovery in a variety of ways.

Biotechs that already rely on AI in their search for new drugs can turn to text generation as a simple, intuitive way to interact with some of their other AI and machine learning tools. Andrew Beam, an AI researcher in the epidemiology department at the Harvard T.H. Chan School of Public Health and a scientific advisor at Generate Biomedicines, calls ChatGPT "a really interesting interface" that allows users to work more easily with other forms of AI than their current interfaces.

For example, Insilico Medicine of New York and Hong Kong, a company set up to search for potential drug targets with its AI-driven platform, is now using ChatGPT as a new way to interact with their target discovery platform, augmenting the relationships and integration provided by knowledge graphs – previously the main method for integrating data. Petrina Kamya, a computational chemist who is head of AI platforms and president at Insilico Medicine in Montreal, says they can talk to their own discovery system thanks to ChatGPT: "Instead of clicking and clicking and clicking, you just ask a question and it composes this text that you read and you understand."

Beyond embracing chatbots to help produce written materials, such as papers, patents or grant applications, others can repurpose them specifically for drug discovery, as a sort of advanced search engine specifically geared to biological science. "We can have a more specific, for example, Bio ChatGPT or Med ChatGPT," says Lurong Pan, a computational chemist at the University of Alabama, Birmingham and founder and CEO of Ainnocence, a biotech with a platform to aid drug discovery. "It may change the way people are searching." For instance, Google and DeepMind earlier this year released Med-PaLM, a chatbot designed to provide answers to medical questions.

All these chatbots are based on large language models (LLMs), algorithms trained on millions of examples of text collected from the internet. LLMs are one type of generative AI – algorithms capable of creating data that did not previously exist. For text, LLMs learn the statistical relationships between words. Then, given a prompt such as a question, they generate text by predicting which word is most likely to follow the previous word. The results seem remarkably natural, though the chatbots often make statements at odds with reality, essentially 'hallucinating' facts. ChatGPT is based on an LLM called Generative Pre-trained Transformer, Med-PaLM draws on Google's Pathways Language Model, and Bard, a more generalized chatbot that Google is incorporating into its search engine, relies on Language Model for Dialogue Applications (LaMDA).

These LLMs are already proving useful for drug hunters, says Kamya. Previously, users of Insilico's platform were able to look at a knowledge graph, a visual representation of the genes linked to a particular disease and the substances known to interact with those genes. That was useful information, but the way researchers worked with it was limited. Now, with the addition of a chat function, Kamya says the data have become much more accessible. "Being able to have a conversation with the tool is very empowering. It makes it more interesting and more fun if you're able to query our biomedical knowledge graphs in the way you want to," she says.

If a scientist wants to investigate psoriasis, for example, the chat function can look at the knowledge graph for that disease. It will deliver a text description that includes the major signaling pathways and genes involved in psoriasis and the compounds known to interact with them. The user can then ask any

Cureus Open Access Original Article DOI: 10.7759/cureus.36272

The Capability of ChatGPT in Predicting and Explaining Common Drug-Drug Interactions

Ayesha Juhi [1], Neha Pipil [2], Soumya Santra [3], Shaikat Mondal [4], Joshil Kumar Behera [5], Himel Mondal [1]

1. Physiology, All India Institute of Medical Sciences, Deoghar, Deoghar, IND 2. Pharmacology, All India Institute of Medical Sciences, Bilaspur, Bilaspur, IND 3. Pharmacology, College of Medicine and JNM Hospital, Kalyani, IND 4. Physiology, Raiganj Government Medical College and Hospital, Raiganj, IND 5. Physiology, Dharanidhar Medical College, Keonjhar, Keonjhar, IND

Corresponding author: Himel Mondal, himelmkcg@gmail.com

Review began 03/13/2023
Review ended 03/14/2023
Published 03/17/2023

Abstract

Background

Drug-drug interactions (DDIs) can have serious consequences for patient health and well-being. Patients who are taking multiple medications may be at an increased risk of experiencing adverse events or drug toxicity if they are not aware of potential interactions between their medications. Many times, patients self-prescribe medications without knowing DDI.

Objective

The objective is to investigate the effectiveness of ChatGPT, a large language model, in predicting and explaining common DDIs.

Methods

A total of 40 DDIs lists were prepared from previously published literature. This list was used to converse with ChatGPT with a two-stage question. The first question was asked as "can I take X and Y together?" with two drug names. After storing the output, the next question was asked. The second question was asked as "why should I not take X and Y together?" The output was stored for further analysis. The responses were checked by two pharmacologists and the consensus output was categorized as "correct" and "incorrect." The "correct" ones were further classified as "conclusive" and "inconclusive." The text was checked for reading ease scores and grades of education required to understand the text. Data were tested by descriptive and inferential statistics.

Results

Among the 40 DDI pairs, one answer was incorrect in the first question. Among correct answers, 19 were conclusive and 20 were inconclusive. For the second question, one answer was wrong. Among correct answers, 17 were conclusive and 22 were inconclusive. The mean Flesch reading ease score was 27.64±10.85 in answers to the first question and 29.35±10.16 in answers to the second question, $p = 0.47$. The mean Flesh-Kincaid grade level was 15.06±2.79 in answers to the first question and 14.85±1.97 in answers to the second question, $p = 0.69$. When we compared the reading levels with hypothetical 6th grade, the grades were significantly higher than expected ($t = 20.57$, $p < 0.0001$ for first answers and $t = 28.43$, $p < 0.0001$ for second answers).

Conclusion

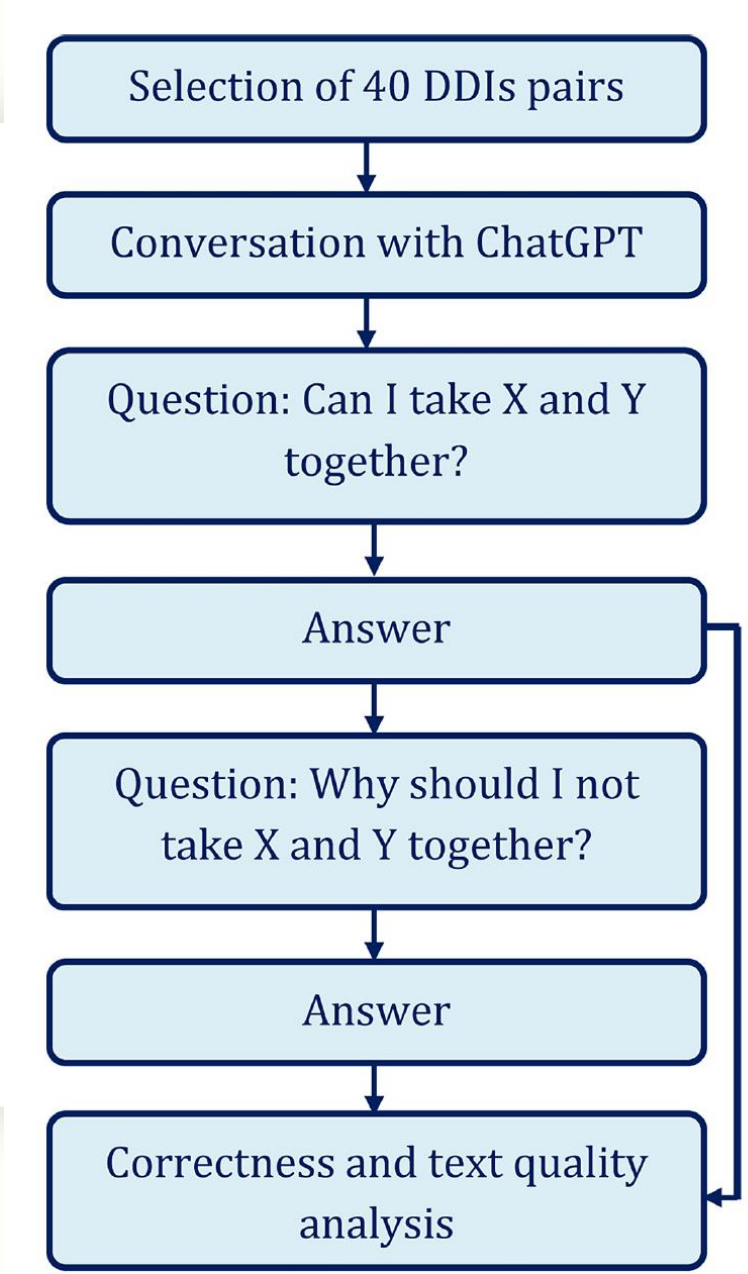
ChatGPT is a partially effective tool for predicting and explaining DDIs. Patients, who may not have immediate access to the healthcare facility for getting information about DDIs, may take help from ChatGPT. However, on several occasions, it may provide incomplete guidance. Further improvement is required for potential usage by patients for getting ideas about DDI.

Categories: Family/General Practice, Medical Education, Epidemiology/Public Health
Keywords: artificial intelligence, patient education, language model, chatgpt, adverse reactions, side effects, explaining, predicting, drug-drug interaction, drug interactions

Introduction

Drug-drug interactions (DDIs) can have serious consequences for patient health and well-being. Patients who are taking multiple medications may be at an increased risk of experiencing adverse events or drug toxicity if they are not aware of potential interactions between their medications. Therefore, patient education on the risks and consequences of DDIs is essential for promoting safe and effective medication use [1]. Patients should also be advised to keep an up-to-date list of all medications they are taking, including over-the-counter drugs, vitamins, and supplements, and to share this information with their

How to cite this article
Juhi A, Pipil N, Santra S, et al. (March 17, 2023) The Capability of ChatGPT in Predicting and Explaining Common Drug-Drug Interactions. Cureus 15(3): e36272. DOI 10.7759/cureus.36272

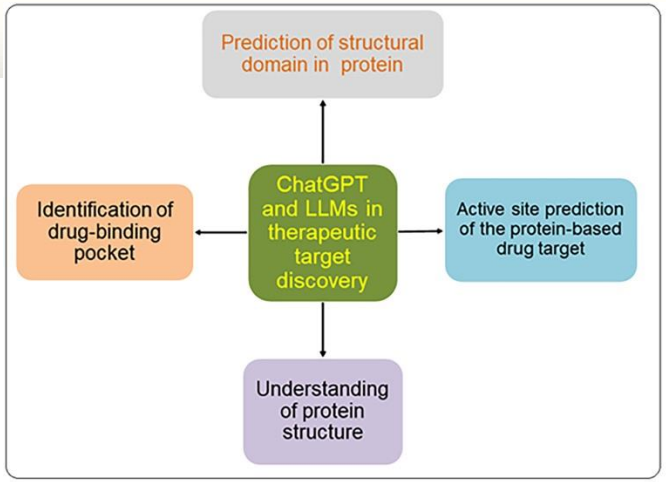


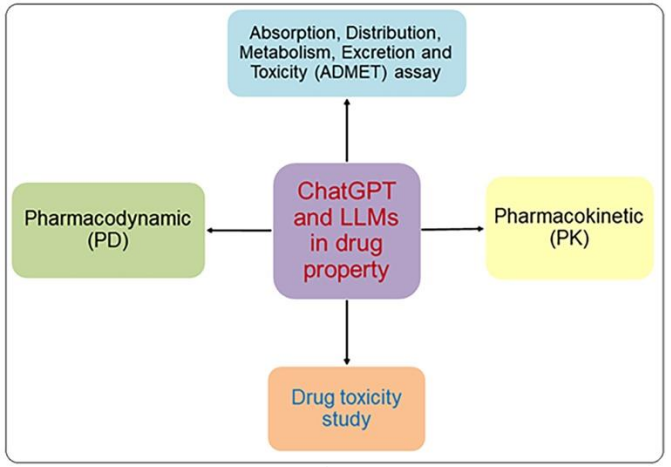
number of incorrect, correct-conclusive, correct - inconclusive answers is shown in Table 1.

Category	Correctness		Number	P-value
Answer to "Can I take?"	Incorrect		1	
	Correct	Conclusive	19	0.003*
		Inconclusive	20	
Answer to "Why should I not take?"	Incorrect		1	
	Correct	Conclusive	17	0.002*
		Inconclusive	22	

Molecular Therapy
Nucleic Acids
Editorial

Artificial intelligence enabled ChatGPT and large language models in drug target discovery, drug discovery, and development





Molecular Therapy - Nucleic Acids 2023 33866-868DOI: (10.1016/j.omtn.2023.08.009)

DrugChat: Towards Enabling ChatGPT-Like Capabilities on Drug Molecule Graphs

SUBMISSION DATE / POSTED DATE

19-05-2023 / 22-05-2023

CITATION

Liang, Youwei; Zhang, Ruiyi; Zhang, Li; Xie, Pengtao (2023): DrugChat: Towards Enabling ChatGPT-Like Capabilities on Drug Molecule Graphs. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.22945922.v1

ChatGPT in Drug Discovery

Gaurav Sharma[*a], Abhishek Thakur[b]

[a] Department of Chemistry, Michigan State University, USA

[b] Center for Biophysics and Computational Biology, Temple University, USA

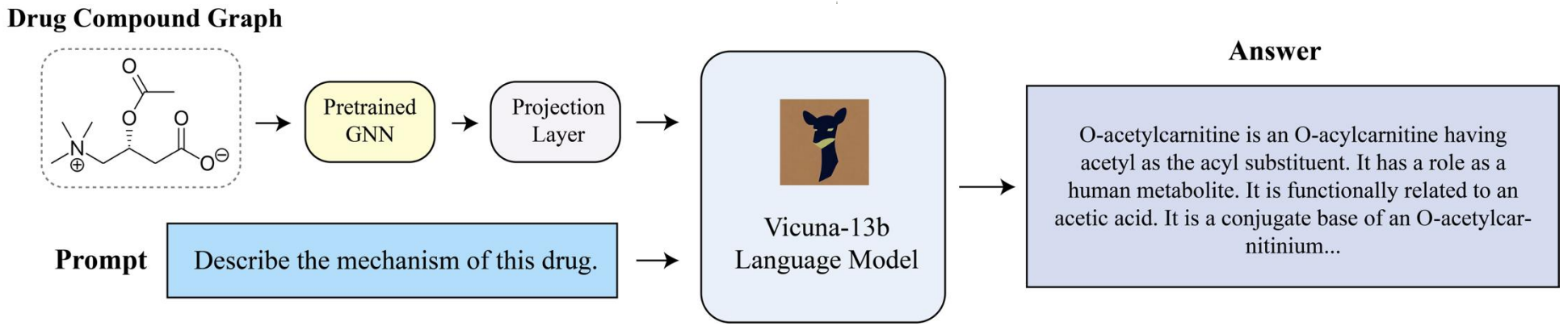
**Corresponding Author:* Gaurav Sharma

gaurav.sharmapsit@gmail.com

Table 1: Dataset statistics.

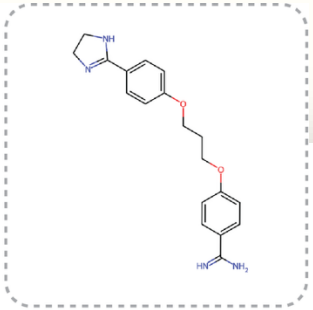
Dataset	Number of Drugs	Number of Question-Answer Pairs
ChEMBL	3,892	129,699
PubChem	6,942	13,818

SMILES of the drug	CC(=O)/N=c1/sc(S(N)(=O)=O)nn1C
QA Pair 1	**Q:** Please provide a description of this drug's mechanism of action. **A:** Carbonic anhydrase II inhibitor.
QA Pair 2	**Q:** Is this compound an acid, a base, or neutral? **A:** Neutral.
QA Pair 3	**Q:** How many rotatable bonds does this compound have? **A:** 1.
QA Pair 4	**Q:** Determine if this drug is administered as a racemic mixture, a single stereoisomer, an achiral molecule, or has an unknown chirality. **A:** An achiral molecule.
QA Pair 5	**Q:** Does this compound satisfy the rule-of-three criteria? **A:** No.


Drug Compound Graph
Pretrained GNN
Projection Layer
Prompt
Describe the mechanism of this drug.
Vicuna-13b Language Model
Answer
O-acetylcarnitine is an O-acylcarnitine having acetyl as the acyl substituent. It has a role as a human metabolite. It is functionally related to an acetic acid. It is a conjugate base of an O-acetylcarnitinium...

SMILES

N=C(N)c1ccc(OCCCOc2ccc(C3=NCCN3)cc2)cc1



Start Chatting

Restart

Beam search numbers) 1

Temperature 1

what makes this compound unique?

This compound is unique because it combines both isoenzymes of AChE inhibition (donepezil and galantamine) in a single molecule, which theoretically could lead to greater synergy in AChE inhibition, resulting in better clinical outcomes.

what diseases can this compound potentially treat?

This compound has the potential to treat neurodegenerative disorders such as Alzheimer's disease and other dementias, as well as other conditions such as myasthenia gravis and mild cognitive impairment.

why can it treat Alzheimer's disease?

This compound can potentially treat Alzheimer's disease by inhibiting the breakdown of the neurotransmitter acetylcholine in the brain. Acetylcholine is involved in memory and cognition, and a deficiency in this neurotransmitter is thought to contribute to the symptoms of Alzheimer's disease. By inhibiting AChE, this compound may increase the levels of acetylcholine in the brain, which could potentially improve cognitive function in people with Alzheimer's disease.

what are the potential side effects of this compound?

The potential side effects of this compound are not yet known. It is important to note that all drugs have the potential to cause side effects, and it is always possible that this compound could have unexpected or unintended side effects. Further research is needed to determine the safety and efficacy of this compound in humans.

Published in final edited form as:
Nat Biotechnol. 2023 August ; 41(8): 1099–1106. doi:10.1038/s41587-022-01618-2.

Large language models generate functional protein sequences across diverse families

Ali Madani[1,2,*], **Ben Krause**[1,†], **Eric R. Greene**[3,†], **Subu Subramanian**[4,5], **Benjamin P. Mohr**[6], **James M. Holton**[7,8,9], **Jose Luis Olmos Jr.**[3], **Caiming Xiong**[1], **Zachary Z. Sun**[6], **Richard Socher**[1], **James S. Fraser**[3], **Nikhil Naik**[1,*]

[1.] Salesforce Research, Palo Alto CA 94301, USA.

[2.] Profluent Bio, San Francisco, CA 94110, USA.

[3.] Department of Bioengineering and Therapeutic Sciences, University of California, San Francisco, San Francisco, CA 94158, USA.

[4.] Department of Molecular and Cell Biology, University of California, Berkeley, Berkeley, CA 94720, USA.

[5.] Howard Hughes Medical Institute, University of California, Berkeley, Berkeley, CA 94720, USA.

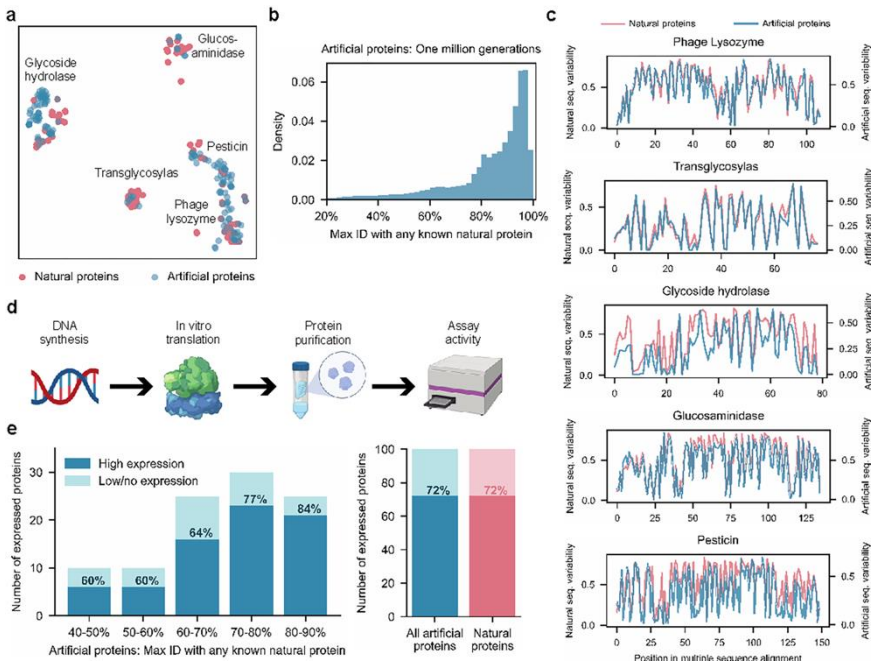
[6.] Tierra Biosciences, San Leandro, CA 94577, USA.

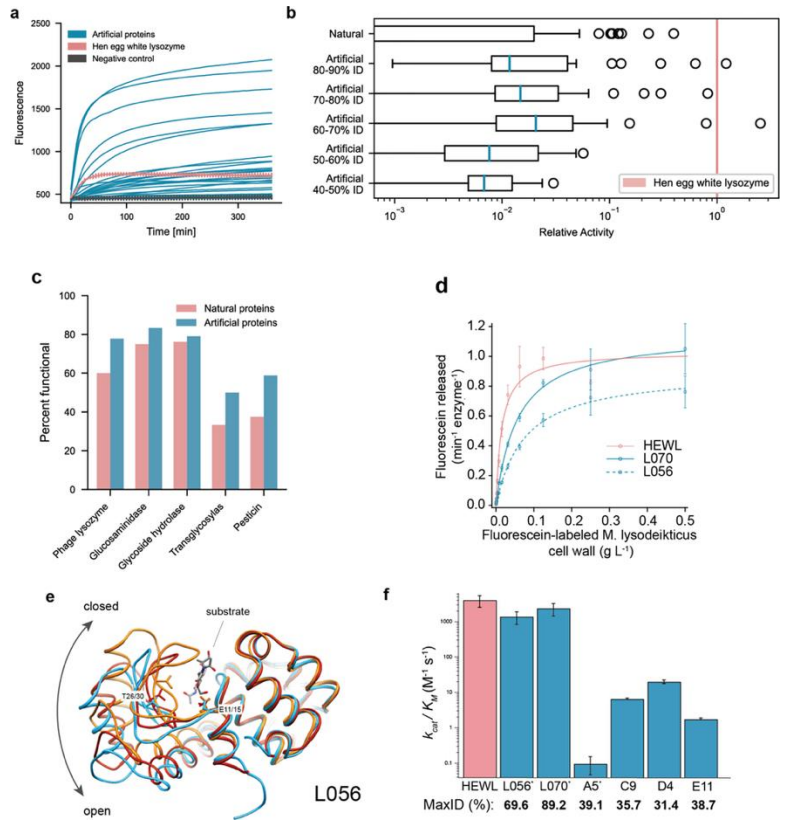
[7.] Molecular Biophysics and Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA.

[8.] Stanford Synchrotron Radiation Lightsource, SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA.

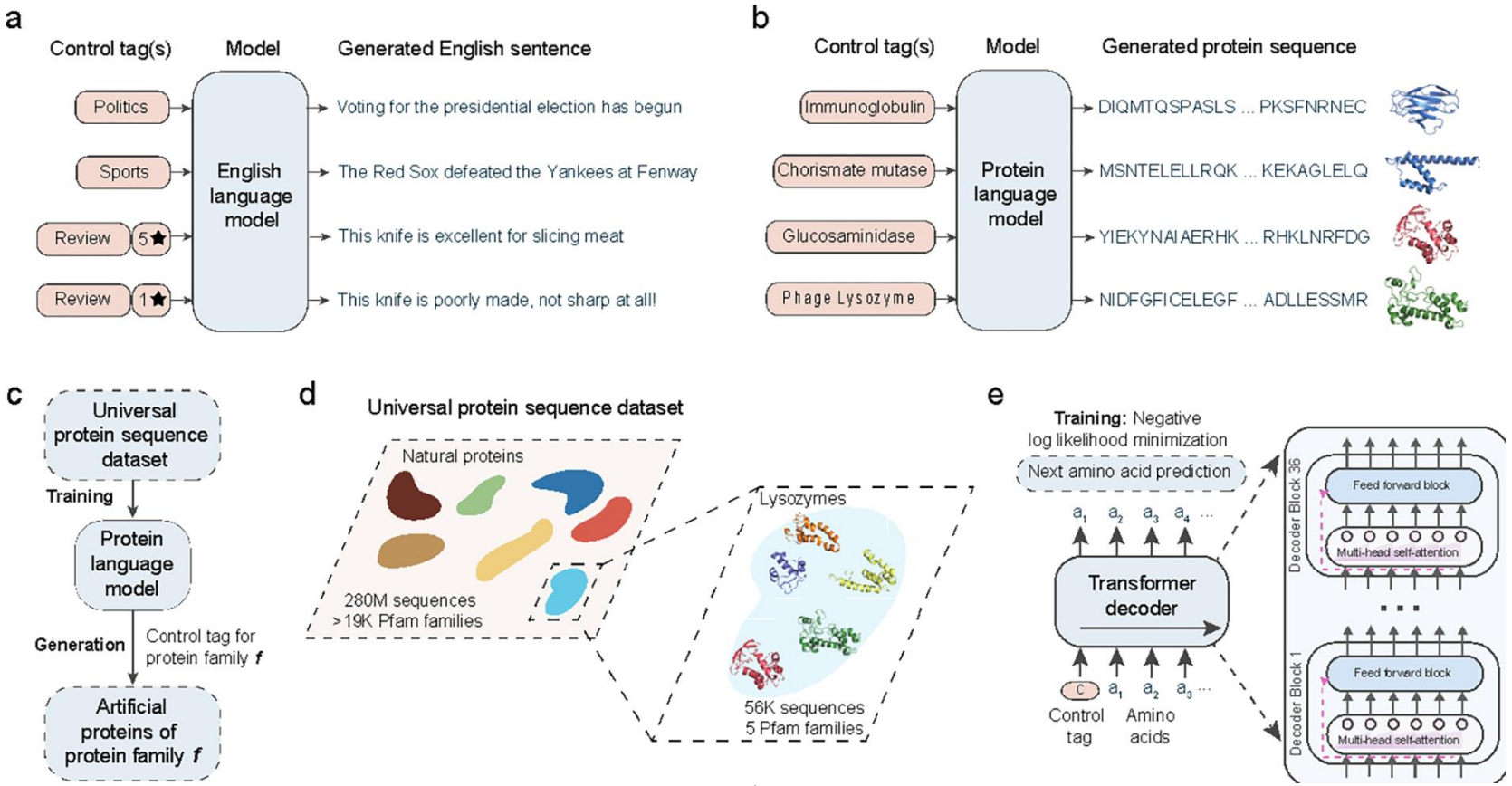
[9.] Department of Biochemistry and Biophysics, University of California, San Francisco, San Francisco, CA 94158, USA.

- prog
  en


a
Glycoside hydrolase
Glucos-aminidase
Pesticin
Transglycosylas
Phage lysozyme
Natural proteins
Artificial proteins
b
Artificial proteins: One million generations
Density
Max ID with any known natural protein
c
Natural proteins
Artificial proteins
Phage Lysozyme
Transglycosylas
Glycoside hydrolase
Glucosaminidase
Pesticin
Position in multiple sequence alignment
d
DNA synthesis
In vitro translation
Protein purification
Assay activity
e
High expression
Low/no expression
Number of expressed proteins
60%
60%
64%
77%
84%
40-50%
50-60%
60-70%
70-80%
80-90%
Artificial proteins: Max ID with any known natural protein
72%
72%
All artificial proteins
Natural proteins


a
Artificial proteins
Hen egg white lysozyme
Negative control
Fluorescence
Time [min]
b
Natural
Artificial 80-90% ID
Artificial 70-80% ID
Artificial 60-70% ID
Artificial 50-60% ID
Artificial 40-50% ID
Hen egg white lysozyme
Relative Activity
c
Natural proteins
Artificial proteins
Percent functional
Phage lysozyme
Glucosaminidase
Glycoside hydrolase
Transglycosylas
Pesticin
d
Fluorescein released (min⁻¹ enzyme⁻¹)
HEWL
L070
L056
Fluorescein-labeled M. lysodeikticus cell wall (g L⁻¹)
e
closed
substrate
open
L056
f
HEWL L056* L070* A5* C9 D4 E11
MaxID (%): 69.6 89.2 39.1 35.7 31.4 38.7

- prog
  en


a
Control tag(s)
Model
Generated English sentence
Politics
Sports
Review 5★
Review 1★
English language model
Voting for the presidential election has begun
The Red Sox defeated the Yankees at Fenway
This knife is excellent for slicing meat
This knife is poorly made, not sharp at all!
b
Control tag(s)
Model
Generated protein sequence
Immunoglobulin
Chorismate mutase
Glucosaminidase
Phage Lysozyme
Protein language model
DIQMTQSPASLS ... PKSFNRNEC
MSNTELELLRQK ... KEKAGLELQ
YIEKYNAIAERHK ... RHKLNRFDG
NIDFGFICELEGF ... ADLLESSMR
c
Universal protein sequence dataset
Training
Protein language model
Generation
Control tag for protein family f
Artificial proteins of protein family f
d
Universal protein sequence dataset
Natural proteins
280M sequences
>19K Pfam families
Lysozymes
56K sequences
5 Pfam families
e
Training: Negative log likelihood minimization
Next amino acid prediction
Transformer decoder
Control tag
Amino acids
Decoder Block 36
Decoder Block 1
Feed forward block
Multi-head self-attention

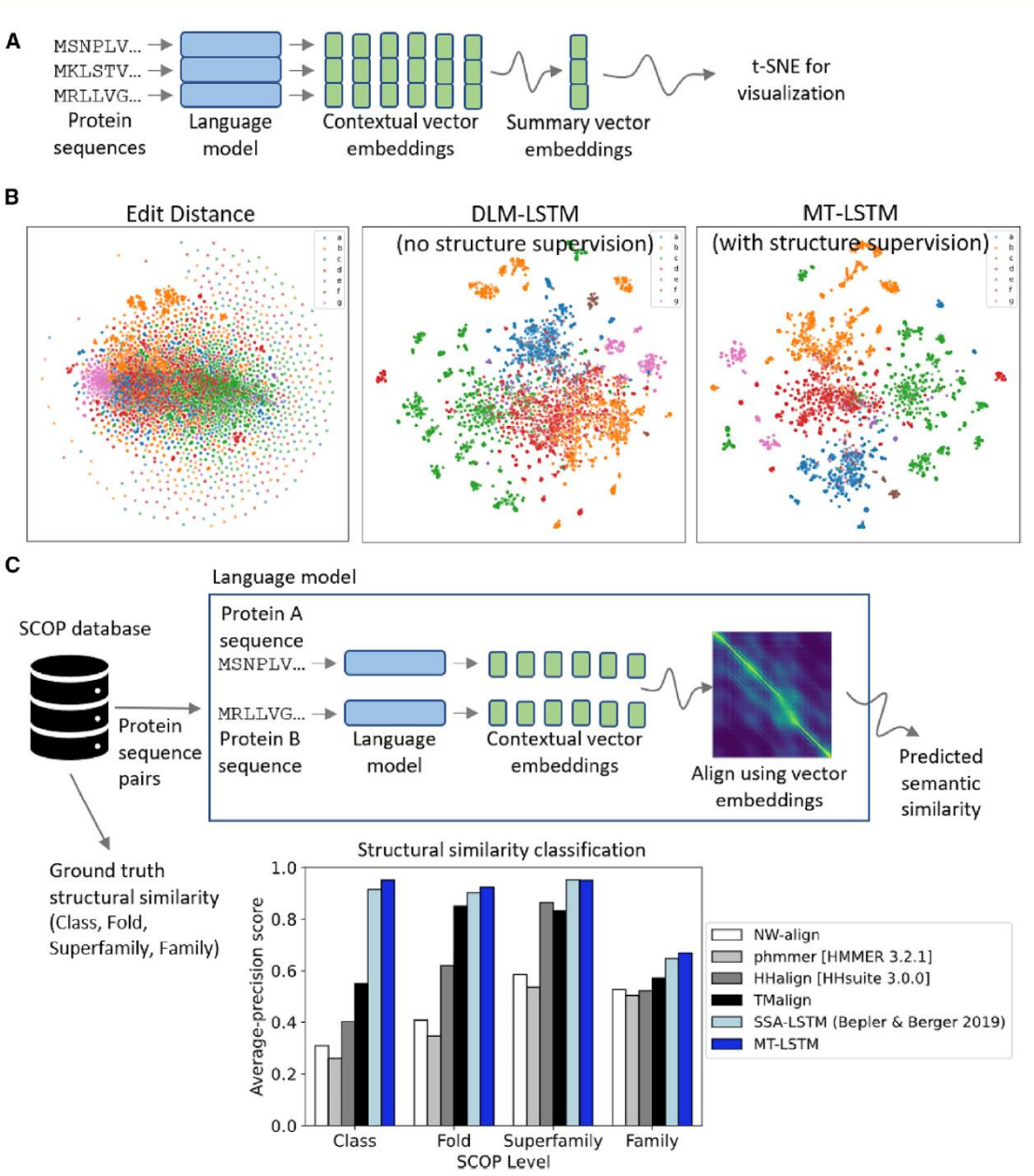
CellPress
OPEN ACCESS

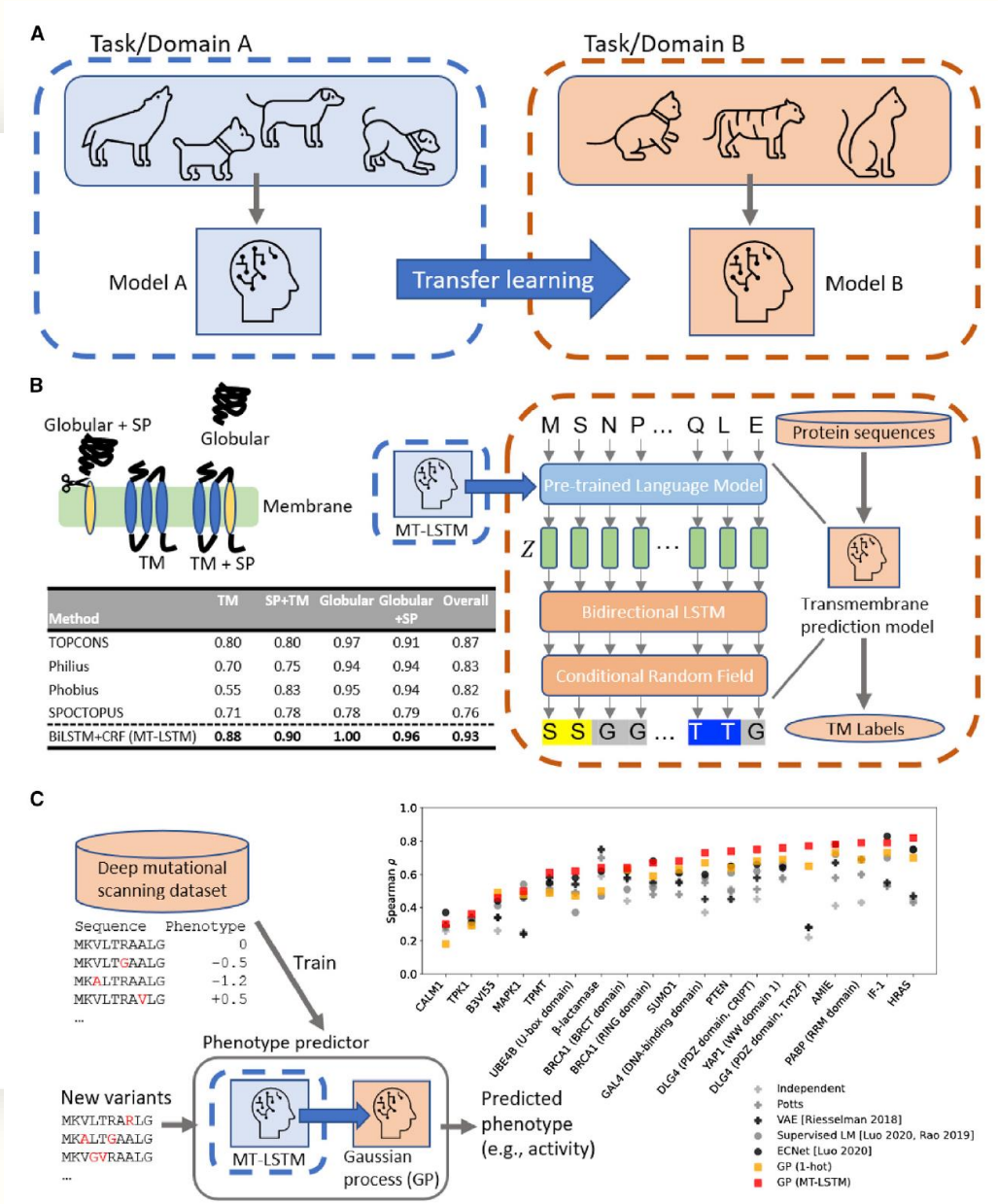
Cell Systems

Synthesis

Learning the protein language: Evolution, structure, and function

Tristan Bepler[1,2,3,*] and Bonnie Berger[2,4,5,*]
[1]Simons Machine Learning Center, New York Structural Biology Center, New York, NY, USA
[2]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA
[3]Computational and Systems Biology Program, Massachusetts Institute of Technology, Cambridge, MA, USA
[4]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA
[5]Lead contact
*Correspondence: tbepler@nysbc.org (T.B.), bab@mit.edu (B.B.)
https://doi.org/10.1016/j.cels.2021.05.017


A
MSNPLV...
MKLSTV...
MRLLVG...
Protein sequences
Language model
Contextual vector embeddings
Summary vector embeddings
t-SNE for visualization
B
Edit Distance
DLM-LSTM
(no structure supervision)
MT-LSTM
(with structure supervision)
C
Language model
SCOP database
Protein sequence pairs
Protein A sequence
MSNPLV...
MRLLVG...
Protein B sequence
Language model
Contextual vector embeddings
Align using vector embeddings
Predicted semantic similarity
Ground truth structural similarity (Class, Fold, Superfamily, Family)
Structural similarity classification
Average-precision score
1.0
0.8
0.6
0.4
0.2
0.0
Class
Fold
Superfamily
Family
SCOP Level
NW-align
phmmer [HMMER 3.2.1]
HHalign [HHsuite 3.0.0]
TMalign
SSA-LSTM (Bepler & Berger 2019)
MT-LSTM



Method	TM	SP+TM	Globular	Globular +SP	Overall
TOPCONS	0.80	0.80	0.97	0.91	0.87
Philius	0.70	0.75	0.94	0.94	0.83
Phobius	0.55	0.83	0.95	0.94	0.82
SPOCTOPUS	0.71	0.78	0.78	0.79	0.76
BiLSTM+CRF (MT-LSTM)	**0.88**	**0.90**	**1.00**	**0.96**	**0.93**

1. ChemBERTa:
•**논문**: "ChemBERTa: Large-Scale Self-Supervised Pretraining of Transformer Models for Molecular Property Prediction"
•**특징**: RoBERTa 기반의 모델로, SMILES 코드를 학습해 분자 특성 예측에 활용됩니다.
•**응용 분야**: 약물 특성 예측, 화합물 설계
2. ChemGPT:
•**논문**: "Generative Pre-trained Transformer (GPT) Models for Molecular Design"
•**특징**: GPT 모델의 구조를 활용해 화학 데이터에서 새로운 분자를 생성하고 최적화합니다.
•**응용 분야**: 새로운 분자 생성, 약물 설계
3. MolBART:
•**논문**: "MolBART: Molecular Representation Learning with Bidirectional AutoRegressive Transformers"
•**특징**: BART 기반의 모델로, 화합물 변형 및 생성에 탁월한 성능을 보입니다.
•**응용 분야**: 약물-타겟 상호작용 예측, 화합물 최적화
4. SMILES Transformer:
•**논문**: "Attention-Based Transformers for Predicting Molecular Properties"
•**특징**: SMILES 표현을 통해 분자 특성을 예측하고 생성하는 모델로, 다양한 화학적 특성 예측에 활용됩니다.
•**응용 분야**: 분자 구조 생성, 약물 설계
5. MoLFormer:
•**논문**: "MolFormer: Self-Supervised Transformer-Based Representation Learning for Molecules"
•**특징**: Microsoft에서 개발한 모델로, 약물-타겟 상호작용 예측과 신약 후보 발굴에 효과적입니다.
•**응용 분야**: 신약 후보 발굴, 화합물의 생물학적 특성 예측
6. MolT5:
•**논문**: "MolT5: A Molecular Language Model for Translating Chemical Texts"
•**특징**: T5 모델을 기반으로 분자 구조 변환과 신약 후보 생성에 활용됩니다.
•**응용 분야**: 분자 구조 변환, 화합물 라이브러리 확장

- Chemical entity Recognition

- **PharmaMind-MolMiner**
  - 객체 감지 기술을 활용한 AI 기반 광학 화학 구조 인식(OCSR) 시스템
  - 화학 요소 추출에 특화
  - GitHub 링크
- **ChemSpot**
  - 텍스트 문서에서 화학 명칭 및 화합물 식별자 추출
  - 화학 엔티티 인식에 사용
- **SciSpacy**
  - 과학 텍스트 처리를 위한 SpaCy 확장판
  - PubChem 등에서 화학 엔티티 추출 가능
- GitHub 링크
- **ChemDataExtractor**
  - 과학 논문에서 화학 구조와 이름 추출
  - NLP와 머신러닝 기술 활용
  - GitHub 링크
- **OSCAR4**
  - 텍스트에서 화학 엔티티 및 반응 식별
  - 화학 텍스트 마이닝 시스템
  - 웹사이트 링크

IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 015022 https://doi.org/10.1088/2632-2153/ac3ffb

MACHINE LEARNING
Science and Technology

PAPER

Chemformer: a pre-trained transformer for computational chemistry

Ross Irwin[1], Spyridon Dimitriadis[1,2], Jiazhen He[1] and Esben Jannik Bjerrum[1,*]

[1] Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg, Sweden
[2] Department of Computer and Information Science, Linköping University, Linköping, Sweden
* Author to whom any correspondence should be addressed.

E-mail: esben.bjerrum@astrazeneca.com

Keywords: transformer, self-supervision, chemistry, reaction prediction, molecular optimization, QSAR

Supplementary material for this article is available online

OPEN ACCESS

RECEIVED
6 September 2021

REVISED
15 November 2021

ACCEPTED FOR PUBLICATION
3 December 2021

PUBLISHED
31 January 2022

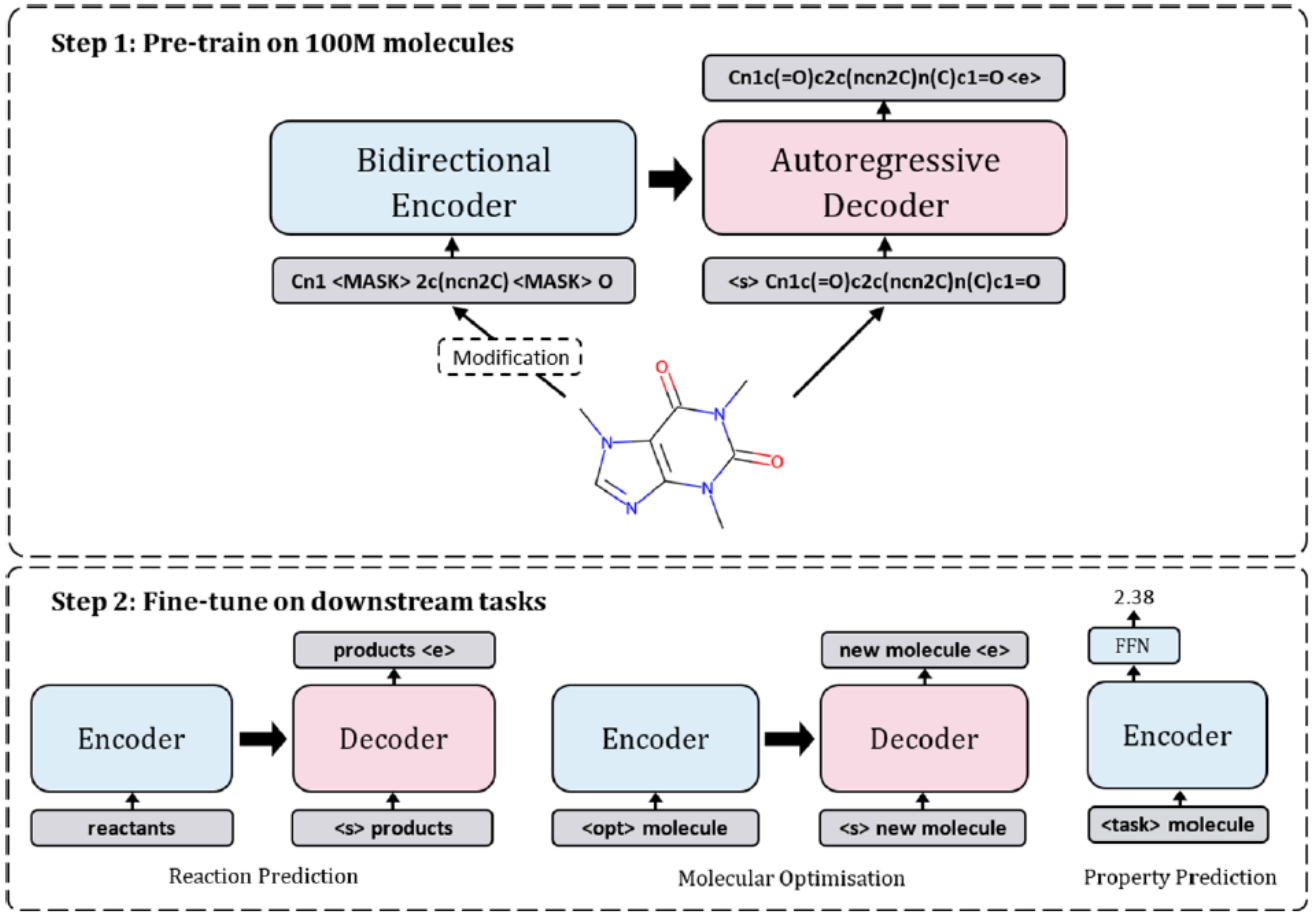


Figure 1. An illustration of the pre-training and fine-tuning procedures for downstream tasks. Here, we use the term pre-training to refer to training on a large dataset in an unsupervised manner before fine-tuning the weights of the model on the specific cheminformatics task of interest. FFN in the figure refers to a feed-forward neural network—a sequence of fully-connected layers of artificial neurons with non-linear activations.

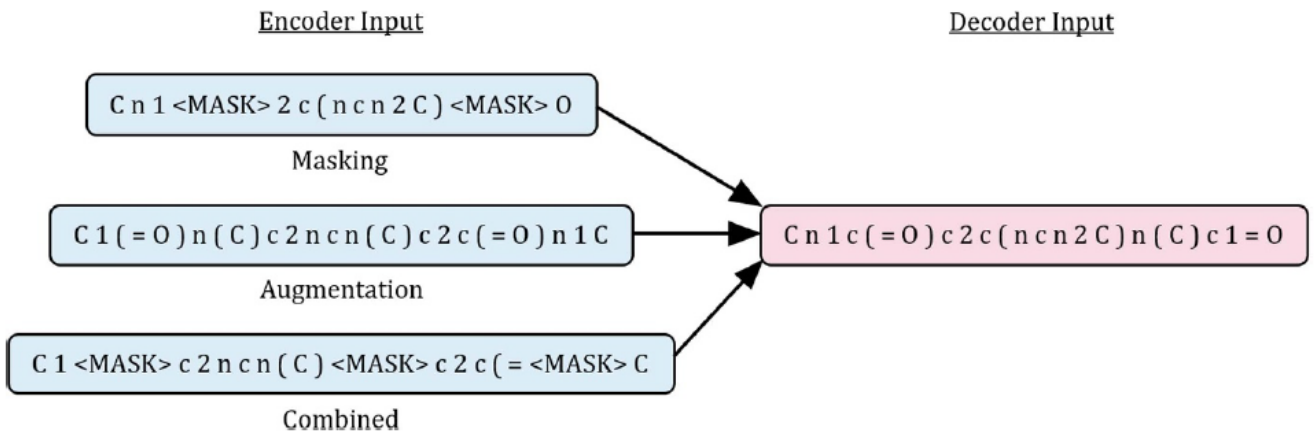


Figure 2. Comparison of the three pre-training tasks for a caffeine molecule. On the left is the modified SMILES given to the encoder, and on the right is the original SMILES, which we train the decoder to predict.

Model	Sequence-to-sequence (%)			Discriminative (Mean R^2)	
	Direct	Retro	Mol opt	Mol prop	Bioactivity
Random	91.1	50.8	73.1	0.680	0.480
Mask	91.2	52.1	**75.0**	0.843	0.603
Augment	91.1	51.8	74.3	0.848	0.606
Combined	**91.8**	**53.6**	72.2	**0.857**	**0.631**

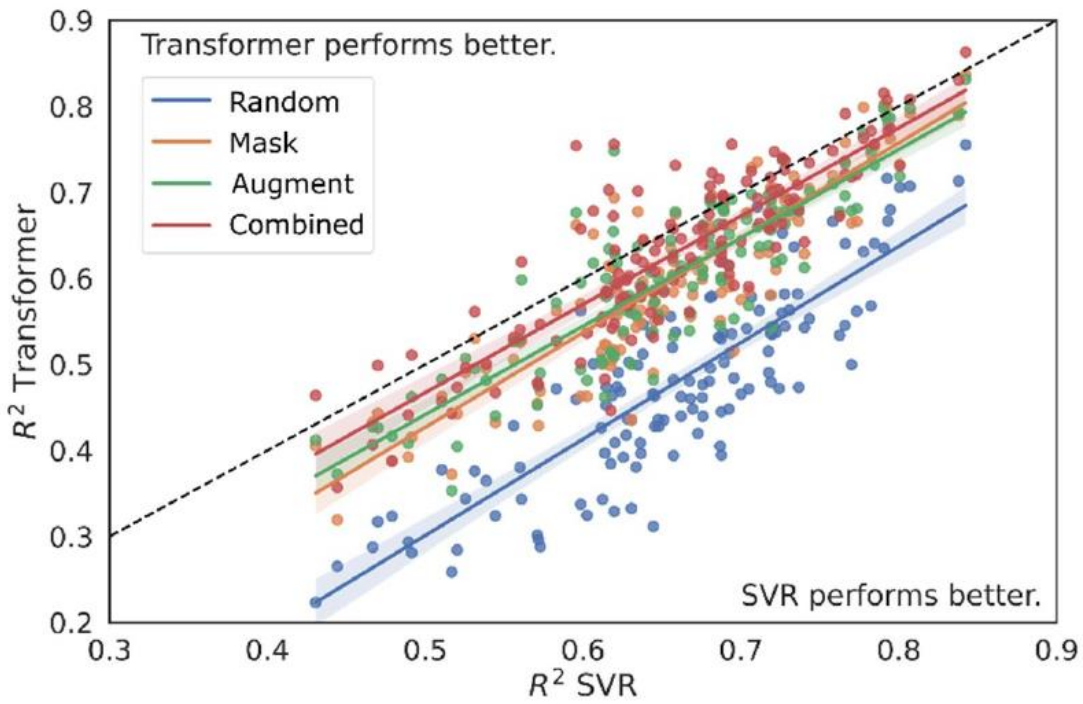


Figure 3. Comparison of the performance of Chemformer models with that of an SVR baseline across 133 bioactivity prediction tasks. Each dot corresponds to the bioactivity prediction result for a single gene. If the dot is above the dashed line the Chemformer model is a better predictor for that gene.

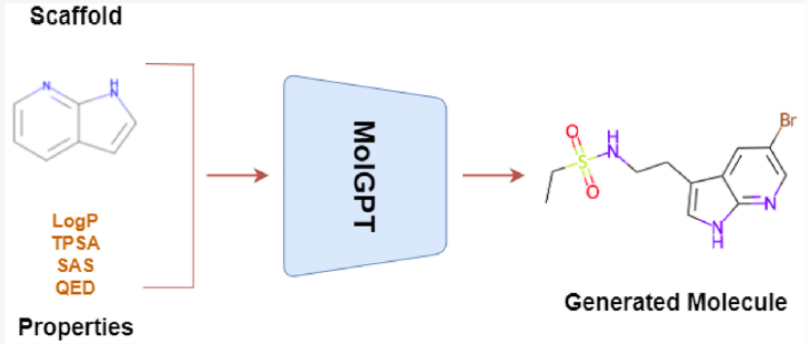
JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

MolGPT: Molecular Generation Using a Transformer-Decoder Model

Viraj Bagal, Rishal Aggarwal, P. K. Vinod, and U. Deva Priyakumar*

Cite This: J. Chem. Inf. Model. 2022, 62, 2064−2076 Read Online



ABSTRACT: Application of deep learning techniques for *de novo* generation of molecules, termed as inverse molecular design, has been gaining enormous traction in drug design. The representation of molecules in SMILES notation as a string of characters enables the usage of state of the art models in natural language processing, such as Transformers, for molecular design in general. Inspired by generative pre-training (GPT) models that have been shown to be successful in generating meaningful text, we train a transformer-decoder on the next token prediction task using masked self-attention for the generation of druglike molecules in this study. We show that our model, MolGPT, performs on par with other previously proposed modern machine learning frameworks for molecular generation in terms of generating valid, unique, and novel molecules. Furthermore, we demonstrate that the model can be trained conditionally to control multiple properties of the generated molecules. We also show that the model can be used to generate molecules with desired scaffolds as well as desired molecular properties by conditioning the generation on scaffold SMILES strings of desired scaffolds and property values. Using saliency maps, we highlight the interpretability of the generative process of the model.

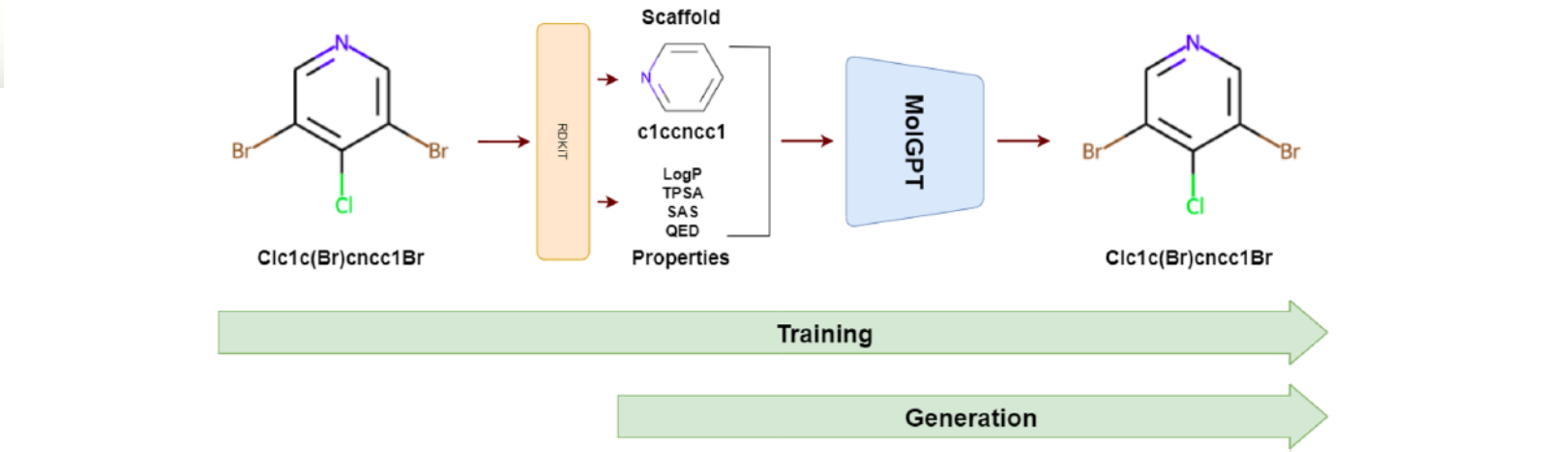


Figure 2. Pipeline for training and generation using the MolGPT model.

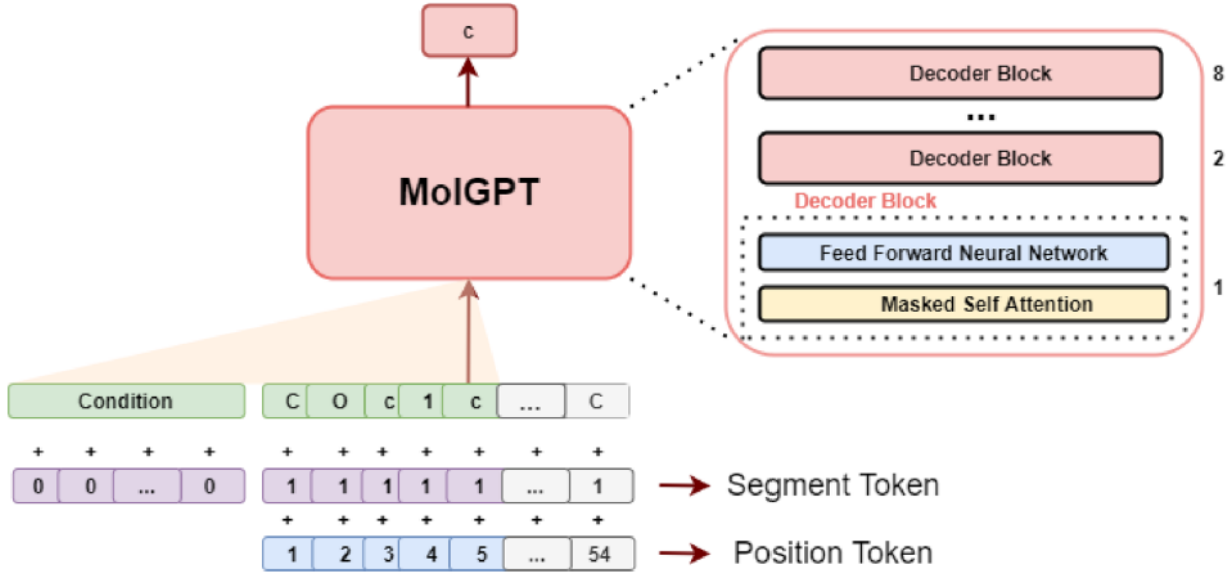


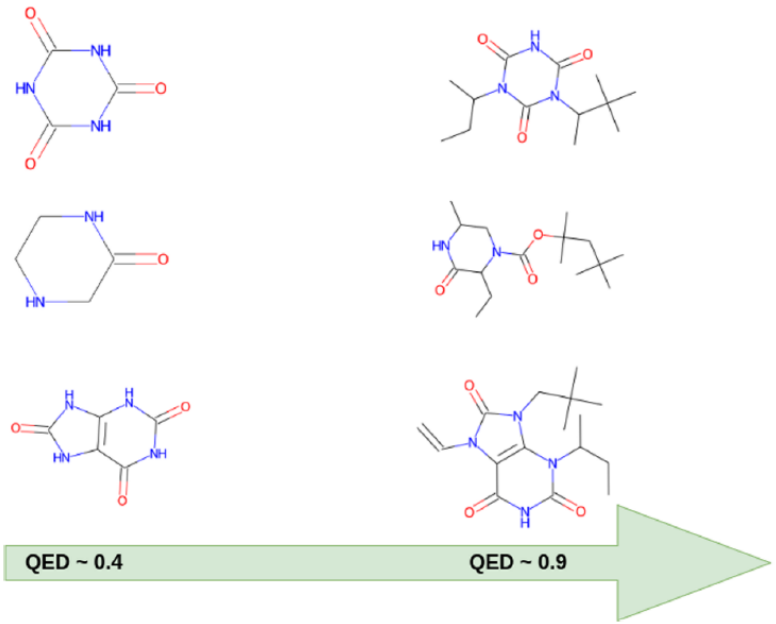
Figure 3. MolGPT model architecture.

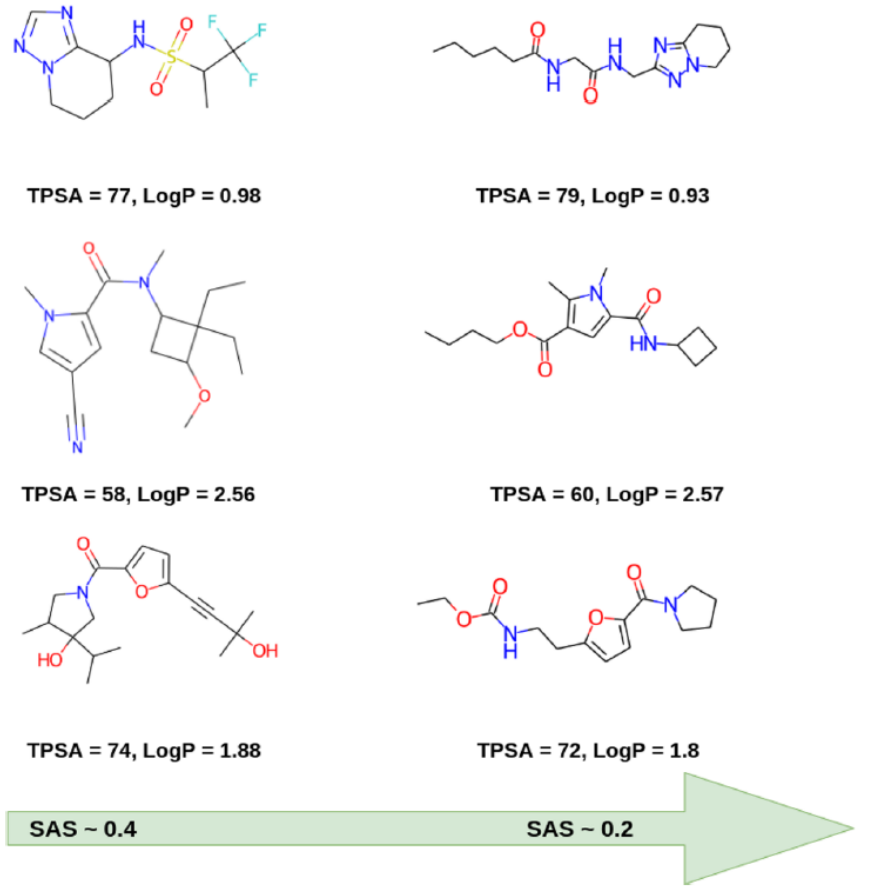
Table 1. Comparison of the Different Metrics Corresponding to Nonconditioned Generation of Molecules Using Different Approaches Trained on MOSES Data Set

models	validity	unique@10K	novelty	$IntDiv_1$	$IntDiv_2$	FCD/Test	FCD/TestSF
CharRNN	0.975	**0.999**	0.842	0.856	0.85	0.0732	0.5204
VAE	0.977	0.998	0.695	0.856	0.85	0.099	0.567
AAE	0.937	0.997	0.793	0.856	0.85	0.555	1.057
LatentGAN	0.897	0.997	**0.949**	0.857	0.85	0.2968	0.8281
JT-VAE	**1.0**	**0.999**	0.914	0.855	0.849	0.395	0.938
MolGPT	0.994	**1.0**	0.797	0.857	0.851	**0.067**	**0.507**

Table 2. Comparison of the Different Metrics Corresponding to Nonconditioned Generation of Molecules Using Different Approaches Trained on GuacaMol Data Set

models	validity	unique	novelty	FCD	KL divergence
SMILES LSTM	0.959	**1.0**	0.912	**0.913**	0.991
AAE	0.822	**1.0**	0.998	0.529	0.886
Organ	0.379	0.841	0.687	0.000	0.267
VAE	0.870	0.999	0.974	0.863	0.982
MolGPT	**0.981**	0.998	**1.0**	0.907	**0.992**


QED ~ 0.4
QED ~ 0.9


TPSA = 77, LogP = 0.98
TPSA = 79, LogP = 0.93
TPSA = 58, LogP = 2.56
TPSA = 60, LogP = 2.57
TPSA = 74, LogP = 1.88
TPSA = 72, LogP = 1.8
SAS ~ 0.4
SAS ~ 0.2

Available online at www.sciencedirect.com

ScienceDirect

Current Opinion in Structural Biology

Chemical language models for de novo drug design: Challenges and opportunities

Francesca Grisoni[a,b]

Abstract

Generative deep learning is accelerating de novo drug design, by allowing the generation of molecules with desired properties on demand. Chemical language models – which generate new molecules in the form of strings using deep learning – have been particularly successful in this endeavour. Thanks to advances in natural language processing methods and interdisciplinary collaborations, chemical language models are expected to become increasingly relevant in drug discovery. This minireview provides an overview of the current state-of-the-art of chemical language models for de novo design, and analyses current limitations, challenges, and advantages. Finally, a perspective on future opportunities is provided.

Addresses

[a] Eindhoven University of Technology, Institute for Complex Molecular Systems and Dept. Biomedical Engineering, Eindhoven, Netherlands
[b] Centre for Living Technologies, Alliance TU/e, WUR, UU, UMC Utrecht, Netherlands

Corresponding author: Grisoni, Francesca (f.grisoni@tue.nl)

Current Opinion in Structural Biology 2023, **79**:102527

This review comes from a themed issue on **Artificial Intelligence (AI) Methodology in Structural Biology (2023)**

Edited by **Andreas Bender**, **Chris de Graaf** and **Noel O'Boyle**

For complete overview of the section, please refer the article collection - Artificial Intelligence (AI) Methodology in Structural Biology (2023)

Available online 2 February 2023

https://doi.org/10.1016/j.sbi.2023.102527

Abbreviations

AI, Artificial Intelligence; CLM, Chemical language model; RNN, Recurrent neural network; SELFIES, Self-referencing embedded strings; SMILES, Simplified Molecular Input Line Entry Systems.

Introduction

Chemical biology is populated with linguistics analogies [1]: the genetic code is transcribed and translated, and cells communicate via chemical signals. Molecules can be considered as the elements of a 'chemical language' [1]. Like human language, chemical language possesses a *syntax*: finite elements (atoms, like words) can relate (form bonds) to one another only in specific ways to produce 'chemically valid' molecules (Fig. 1a). Molecules also have *semantical properties*: based on which elements are present and how they are connected, different high-level properties (*e.g.*, physicochemical, biological) will emerge (Fig. 1b).

Learning the chemical language is crucial for de novo drug design [2], which addresses the difficult question of how to generate molecules from scratch that are chemically valid (*syntax*) and possess desired pharmacological properties (*semantics*). De novo design is confronted with an extremely vast 'chemical universe', estimated to contain up to 10^{60} small molecular entities one could consider [3], which makes extensive enumeration practically impossible. De novo design has highly benefitted from the recent artificial intelligence (AI) *renaissance*, in the form of deep learning [4]. 'Generative' deep learning allows for generating 'raw' representations of molecules (*e.g.*, molecular graphs), and circumvents the need for molecule assembly and construction rules of conventional design algorithms [5].

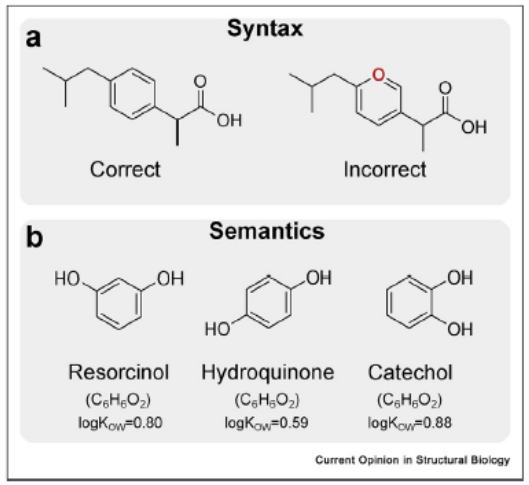
Among the many flavours of deep learning for drug discovery, chemical language models (CLMs) [6,7] have spearheaded AI-driven de novo design (Fig. 2). CLMs borrow and adapt algorithms developed for natural language processing to learning the chemical language. This is allowed by the usage of string notations, such as Simplified Molecular Input Line Entry Systems (SMILES [8]) strings (Fig. 2).

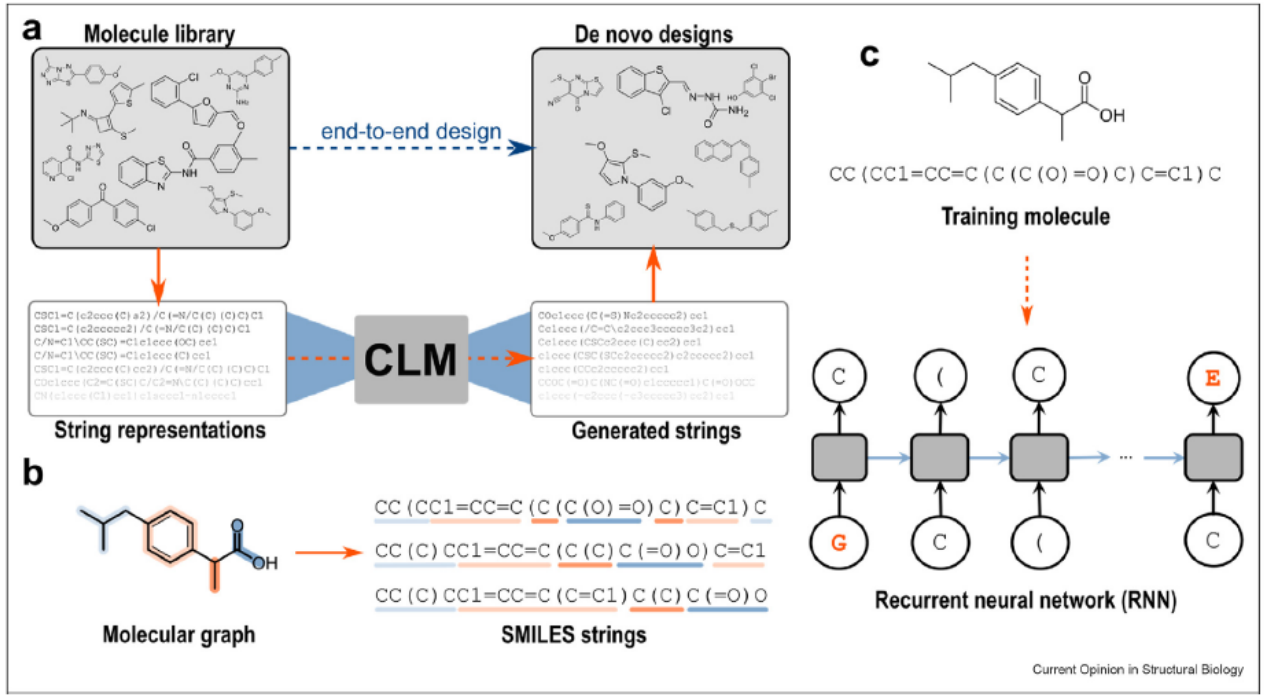
In the last few years, CLMs have been successful in designing experimentally-validated bioactive molecules [7,9–11], and are providing increasing evidence of their capacities to explore the uncharted biochemical matter. This minireview will focus on deep-learning-based CLMs for de novo molecule design, although other exciting applications have been reported [12]. After discussing the state-of-the-art of CLM-driven de novo design, this minireview describes current gaps and future opportunities in the field of drug discovery.

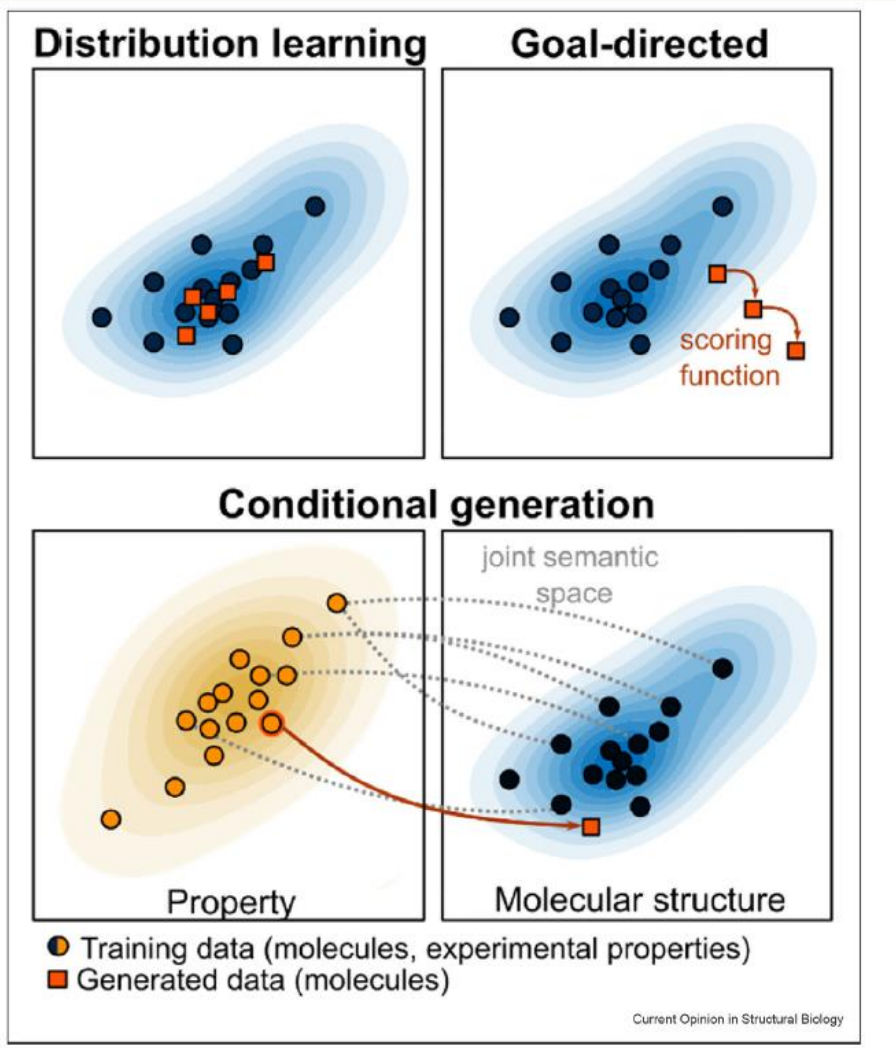
State of the art

Molecular string representations: advantages and drawbacks

Molecular string representations were originally developed for database storage and molecule identification [13], but they have found a *renaissance* thanks to deep


a
Syntax
O
OH
Correct
O
O
OH
Incorrect
b
Semantics
HO
OH
Resorcinol
($C_6H_6O_2$)
$\log K_{OW}$=0.80
OH
HO
Hydroquinone
($C_6H_6O_2$)
$\log K_{OW}$=0.59
OH
OH
Catechol
($C_6H_6O_2$)
$\log K_{OW}$=0.88
Current Opinion in Structural Biology


a
Molecule library
De novo designs
end-to-end design
CLM
String representations
Generated strings
b
Molecular graph
CC(CC1=CC=C(C(C(O)=O)C)C=C1)C
CC(C)CC1=CC=C(C(C)C(=O)O)C=C1
CC(C)CC1=CC=C(C=C1)C(C)C(=O)O
SMILES strings
c
CC(CC1=CC=C(C(C(O)=O)C)C=C1)C
Training molecule
Recurrent neural network (RNN)
Current Opinion in Structural Biology


Distribution learning
Goal-directed
scoring function
Conditional generation
joint semantic space
Property
Molecular structure
Training data (molecules, experimental properties)
Generated data (molecules)
Current Opinion in Structural Biology

nature machine intelligence

Article

https://doi.org/10.1038/s42256-022-00534-z

scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data

Received: 3 February 2022

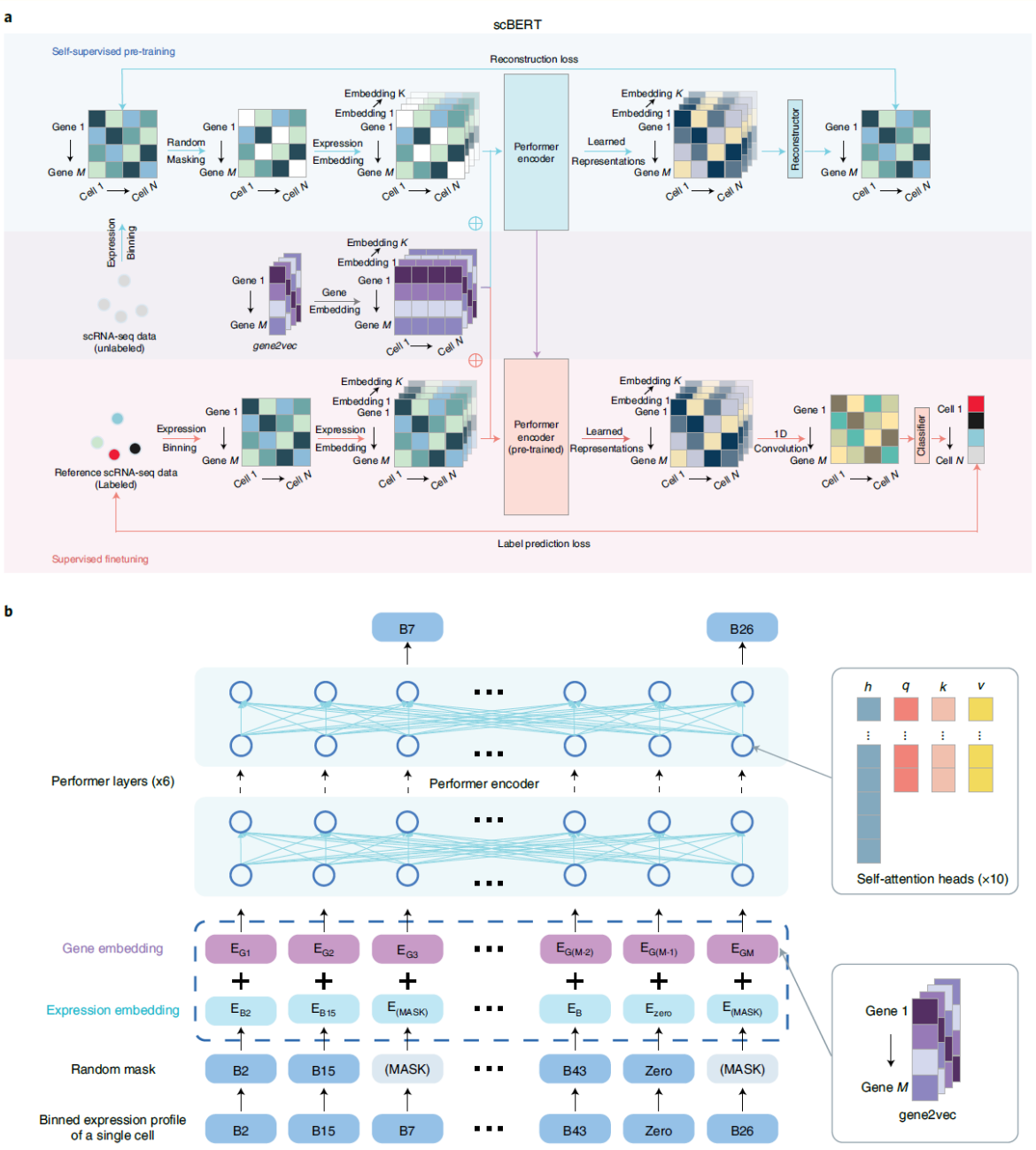
Accepted: 19 August 2022

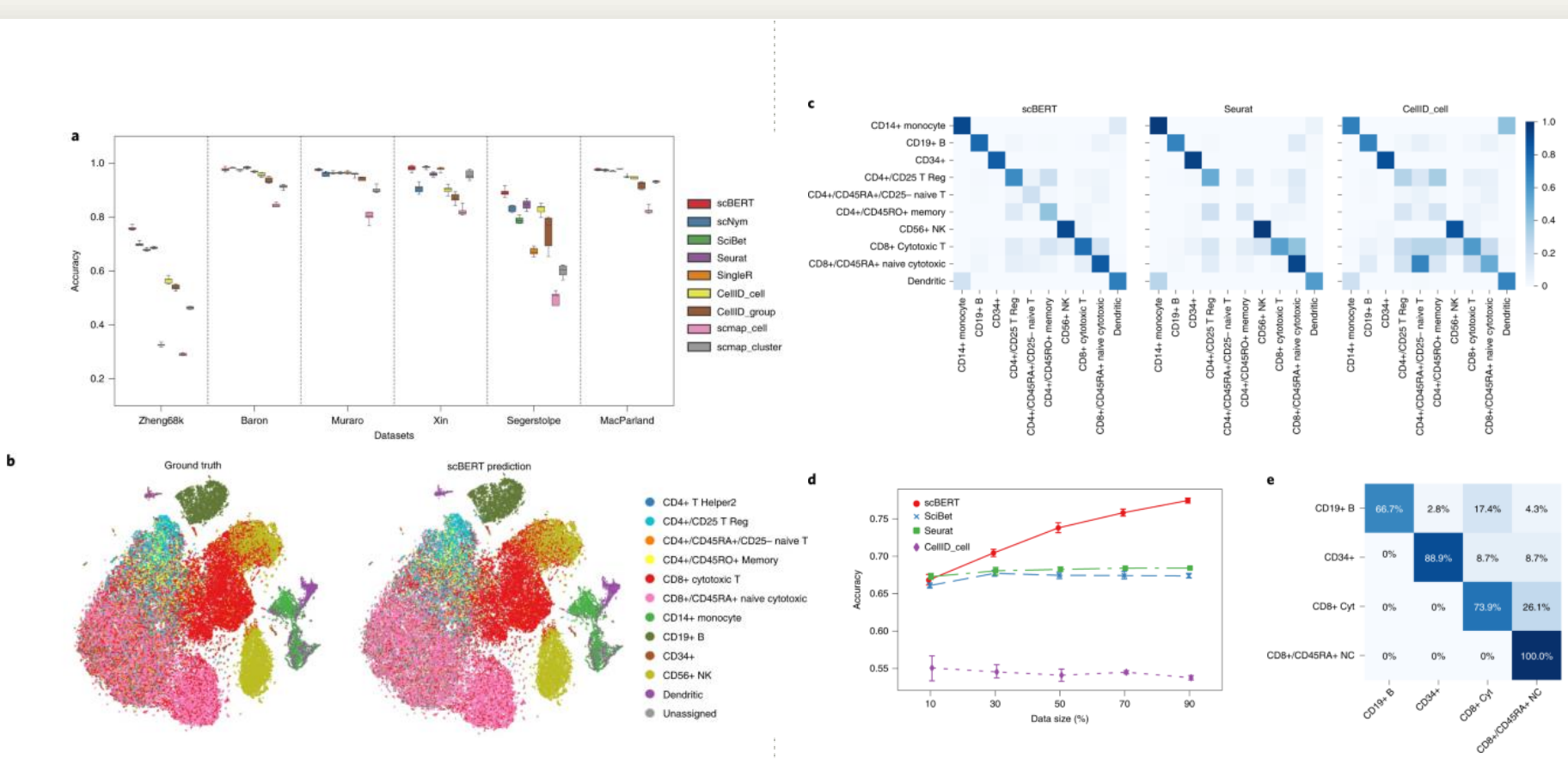
Published online: 26 September 2022

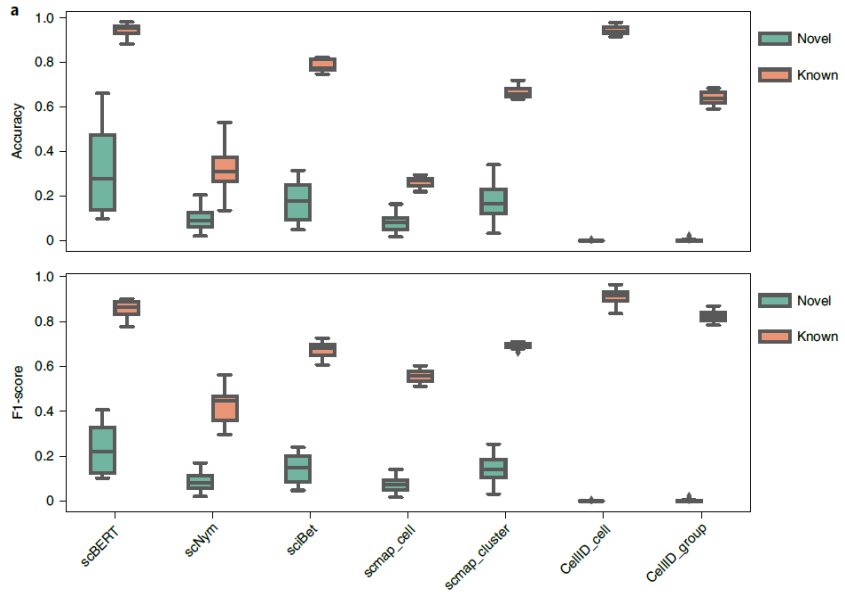
Check for updates

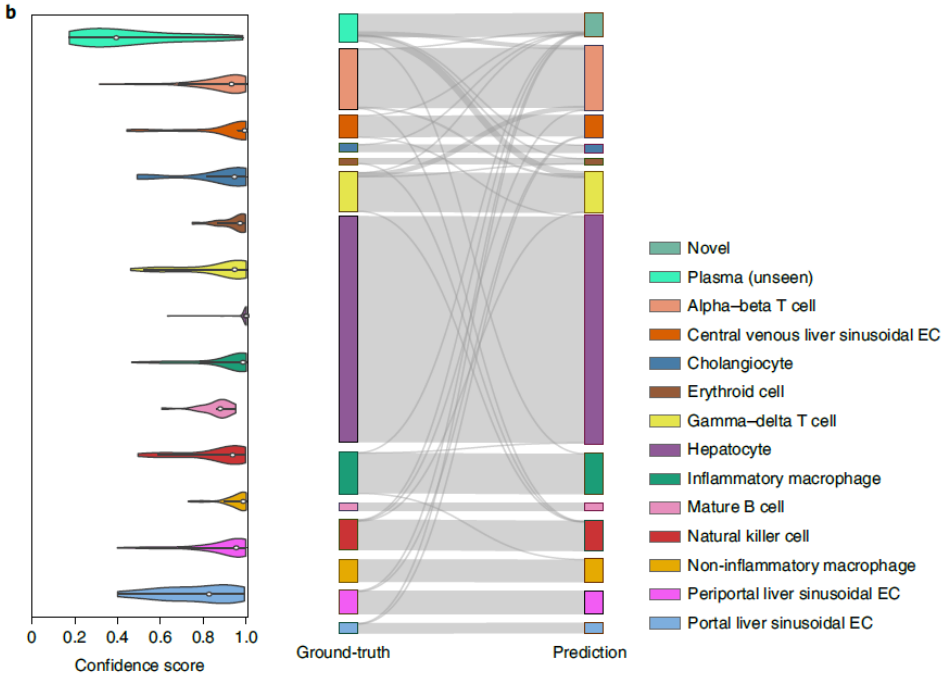
Fan Yang[1,7], Wenchuan Wang[1,2,7], Fang Wang[1,7], Yuan Fang[1,3,4], Duyu Tang[1], Junzhou Huang[5], Hui Lu[2,6] ✉ and Jianhua Yao[1] ✉

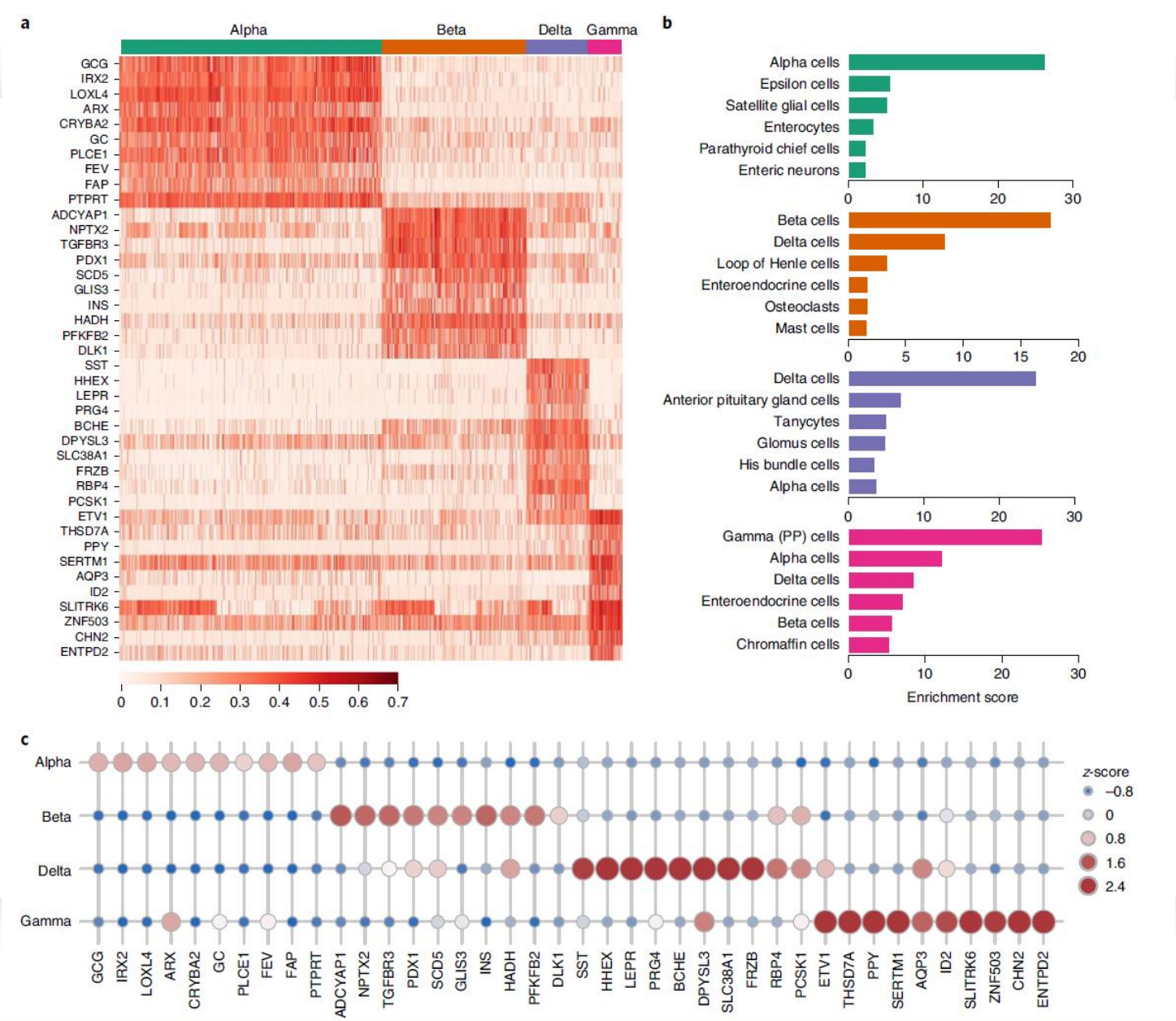
Annotating cell types on the basis of single-cell RNA-seq data is a prerequisite for research on disease progress and tumour microenvironments. Here we show that existing annotation methods typically suffer from a lack of curated marker gene lists, improper handling of batch effects and difficulty in leveraging the latent gene–gene interaction information, impairing their generalization and robustness. We developed a pretrained deep neural network-based model, single-cell bidirectional encoder representations from transformers (scBERT), to overcome the challenges. Following BERT's approach to pretraining and fine-tuning, scBERT attains a general understanding of gene–gene interactions by being pretrained on huge amounts of unlabelled scRNA-seq data; it is then transferred to the cell type annotation task of unseen and user-specific scRNA-seq data for supervised fine-tuning. Extensive and rigorous benchmark studies validated the superior performance of scBERT on cell type annotation, novel cell type discovery, robustness to batch effects and model interpretability.


a
scBERT
Self-supervised pre-training
Reconstruction loss
Gene 1
Gene M
Cell 1
Cell N
Random Masking
Expression Embedding
Embedding K
Embedding 1
Performer encoder
Learned Representations
Reconstructor
Expression Binning
scRNA-seq data (unlabeled)
Gene Embedding
gene2vec
Reference scRNA-seq data (Labeled)
Performer encoder (pre-trained)
1D Convolution
Classifier
Label prediction loss
Supervised finetuning
b
B7
B26
Performer layers (x6)
Performer encoder
h
q
k
v
Self-attention heads (×10)
Gene embedding
Expression embedding
Random mask
Binned expression profile of a single cell
B2
B15
(MASK)
B43
Zero
gene2vec


a
Accuracy
1.0
0.8
0.6
0.4
0.2
Zheng68k
Baron
Muraro
Xin
Segerstolpe
MacParland
Datasets
scBERT
scNym
SciBet
Seurat
SingleR
CellID_cell
CellID_group
scmap_cell
scmap_cluster
b
Ground truth
scBERT prediction
CD4+ T Helper2
CD4+/CD25 T Reg
CD4+/CD45RA+/CD25− naive T
CD4+/CD45RO+ Memory
CD8+ cytotoxic T
CD8+/CD45RA+ naive cytotoxic
CD14+ monocyte
CD19+ B
CD34+
CD56+ NK
Dendritic
Unassigned
c
scBERT
Seurat
CellID_cell
CD14+ monocyte
CD19+ B
CD34+
CD4+/CD25 T Reg
CD4+/CD45RA+/CD25− naive T
CD4+/CD45RO+ memory
CD56+ NK
CD8+ Cytotoxic T
CD8+/CD45RA+ naive cytotoxic
Dendritic
1.0
0.8
0.6
0.4
0.2
0
d
scBERT
SciBet
Seurat
CellID_cell
Accuracy
0.75
0.70
0.65
0.60
0.55
10
30
50
70
90
Data size (%)
e
CD19+ B
CD34+
CD8+ Cyt
CD8+/CD45RA+ NC
66.7%
2.8%
17.4%
4.3%
0%
88.9%
8.7%
8.7%
0%
0%
73.9%
26.1%
0%
0%
0%
100.0%


a
Accuracy
F1-score
1.0
0.8
0.6
0.4
0.2
0
Novel
Known
scBERT
scNym
scBet
scmap_cell
scmap_cluster
CellID_cell
CellID_group


b
0
0.2
0.4
0.6
0.8
1.0
Confidence score
Ground-truth
Prediction
Novel
Plasma (unseen)
Alpha–beta T cell
Central venous liver sinusoidal EC
Cholangiocyte
Erythroid cell
Gamma–delta T cell
Hepatocyte
Inflammatory macrophage
Mature B cell
Natural killer cell
Non-inflammatory macrophage
Periportal liver sinusoidal EC
Portal liver sinusoidal EC


a
Alpha
Beta
Delta
Gamma
GCG
IRX2
LOXL4
ARX
CRYBA2
GC
PLCE1
FEV
FAP
PTPRT
ADCYAP1
NPTX2
TGFBR3
PDX1
SCD5
GLIS3
INS
HADH
PFKFB2
DLK1
SST
HHEX
LEPR
PRG4
BCHE
DPYSL3
SLC38A1
FRZB
RBP4
PCSK1
ETV1
THSD7A
PPY
SERTM1
AQP3
ID2
SLITRK6
ZNF503
CHN2
ENTPD2
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7
b
Alpha cells
Epsilon cells
Satellite glial cells
Enterocytes
Parathyroid chief cells
Enteric neurons
0 10 20 30
Beta cells
Delta cells
Loop of Henle cells
Enteroendocrine cells
Osteoclasts
Mast cells
0 5 10 15 20
Delta cells
Anterior pituitary gland cells
Tanycytes
Glomus cells
His bundle cells
Alpha cells
0 10 20 30
Gamma (PP) cells
Alpha cells
Delta cells
Enteroendocrine cells
Beta cells
Chromaffin cells
0 10 20 30
Enrichment score
c
Alpha
Beta
Delta
Gamma
z-score
−0.8
0
0.8
1.6
2.4

PNAS RESEARCH ARTICLE | BIOPHYSICS AND COMPUTATIONAL BIOLOGY OPEN ACCESS

Contrastive learning in protein language space predicts interactions between drugs and protein targets

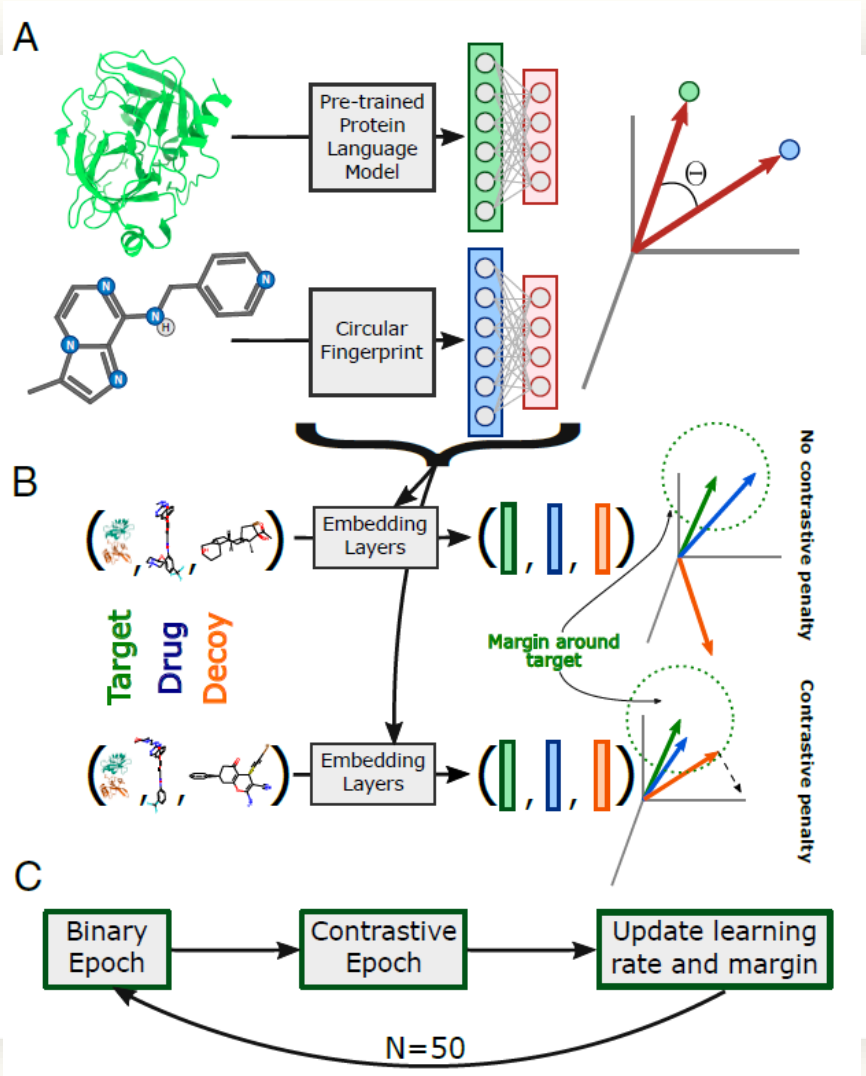
Rohit Singh[a,1], Samuel Sledzieski[a,1], Bryan Bryson[b,c], Lenore Cowen[d,2], and Bonnie Berger[a,e,2]

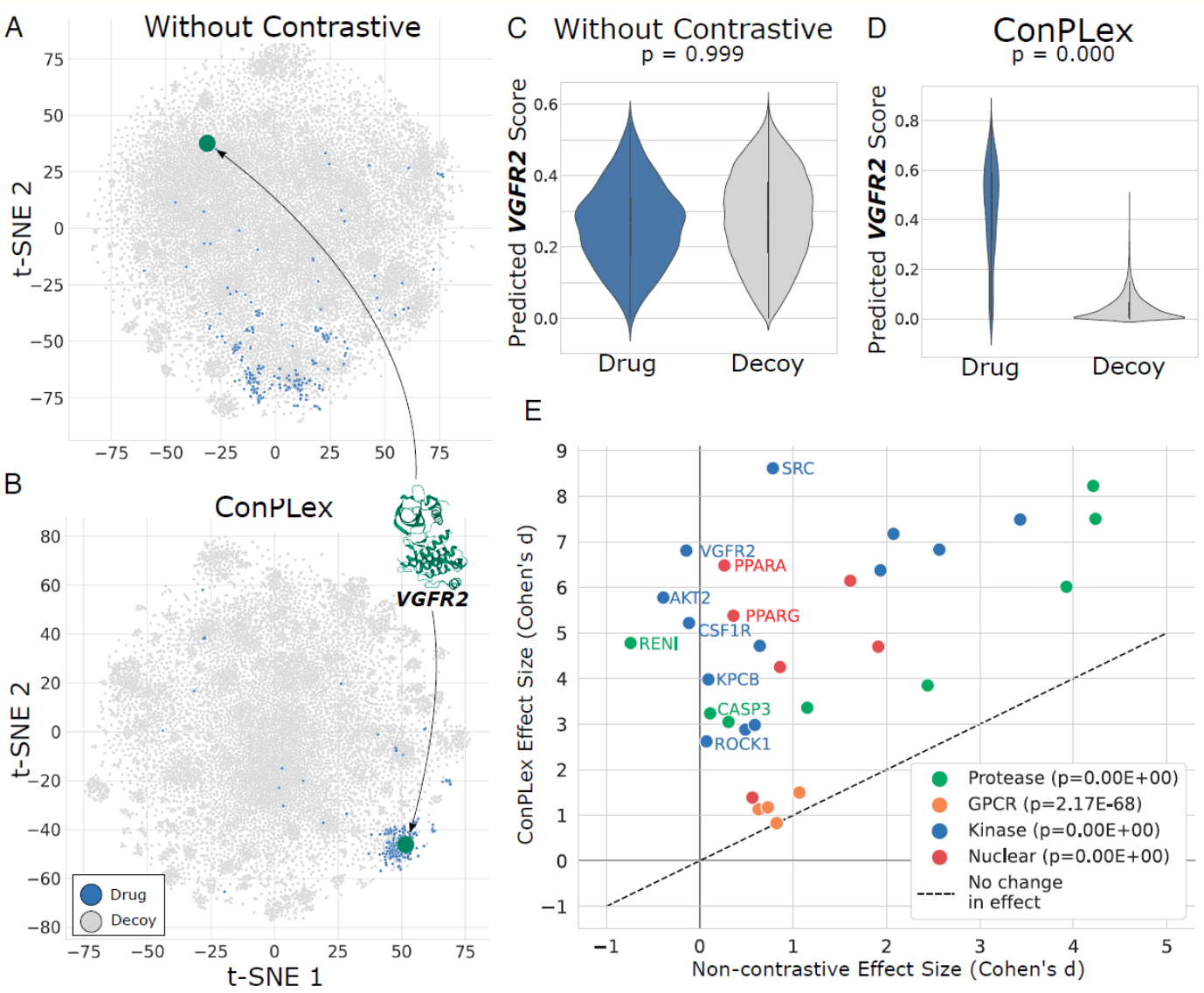
Edited by Barry Honig, Columbia University, New York, NY; received December 6, 2022; accepted April 10, 2023

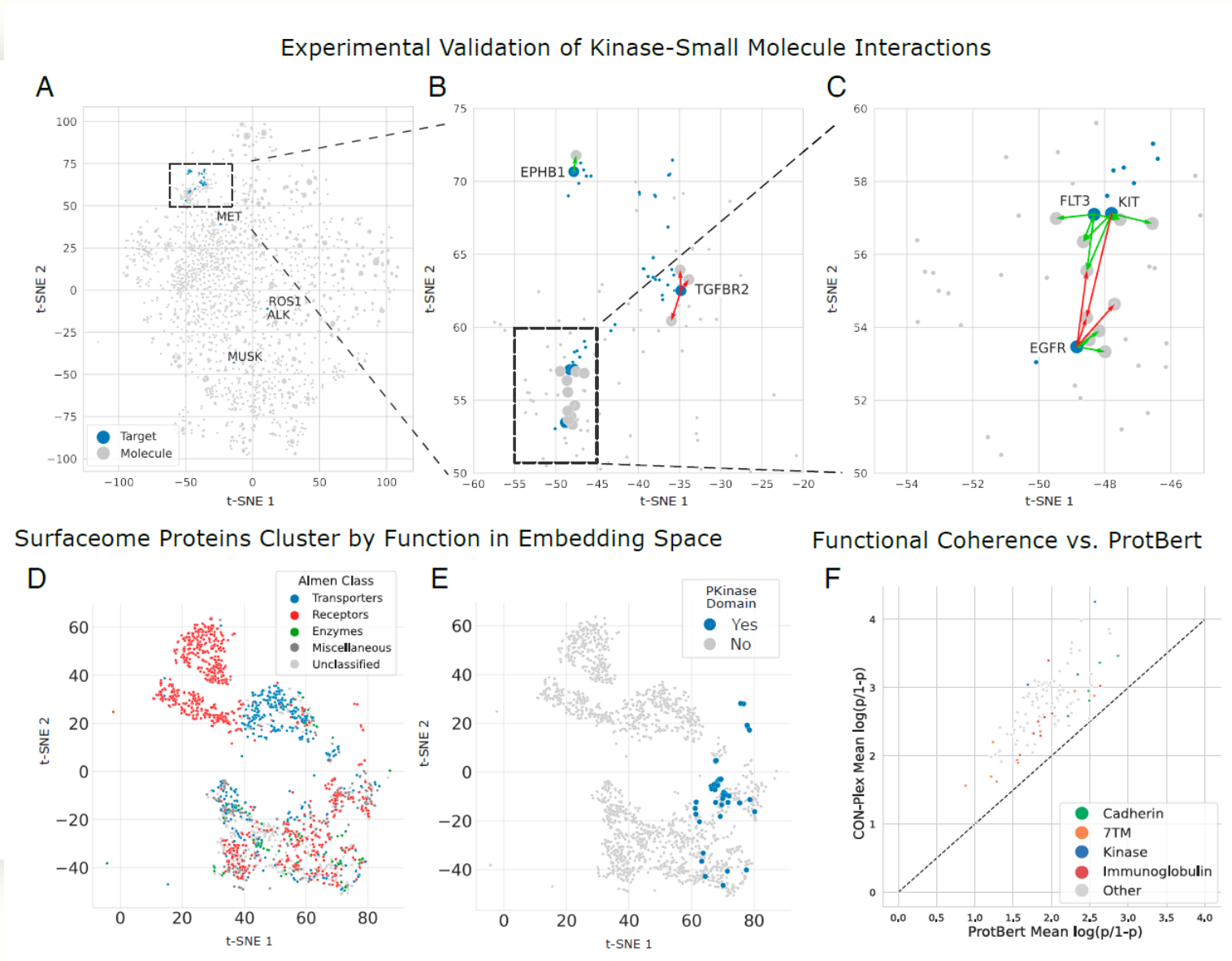
Sequence-based prediction of drug–target interactions has the potential to accelerate drug discovery by complementing experimental screens. Such computational prediction needs to be generalizable and scalable while remaining sensitive to subtle variations in the inputs. However, current computational techniques fail to simultaneously meet these goals, often sacrificing performance of one to achieve the others. We develop a deep learning model, ConPLex, successfully leveraging the advances in pretrained protein language models ("PLex") and employing a protein-anchored contrastive coembedding ("Con") to outperform state-of-the-art approaches. ConPLex achieves high accuracy, broad adaptivity to unseen data, and specificity against decoy compounds. It makes predictions of binding based on the distance between learned representations, enabling predictions at the scale of massive compound libraries and the human proteome. Experimental testing of 19 kinase-drug interaction predictions validated 12 interactions, including four with subnanomolar affinity, plus a strongly binding EPHB1 inhibitor ($K_D = 1.3$ nM). Furthermore, ConPLex embeddings are interpretable, which enables us to visualize the drug–target embedding space and use embeddings to characterize the function of human cell-surface proteins. We anticipate that ConPLex will facilitate efficient drug discovery by making highly sensitive in silico drug screening feasible at the genome scale. ConPLex is available open source at ConPLex.csail.mit.edu.

Significance

In time and money, one of the most expensive steps of the drug discovery pipeline is the experimental screening of small molecules to determine binding to a protein target of interest. Therefore, accurate high-throughput computational prediction of drug-target interactions would unlock significant value, guiding and prioritizing promising candidates for experimental screening. We introduce ConPLex, a machine learning method for predicting


A
Pre-trained Protein Language Model
Circular Fingerprint
Θ
B
Embedding Layers
Embedding Layers
Target
Drug
Decoy
Margin around target
No contrastive penalty
Contrastive penalty
C
Binary Epoch
Contrastive Epoch
Update learning rate and margin
N=50


A
Without Contrastive
t-SNE 2
B
ConPLex
VGFR2
t-SNE 2
t-SNE 1
Drug
Decoy
C
Without Contrastive
p = 0.999
Predicted VGFR2 Score
Drug
Decoy
D
ConPLex
p = 0.000
Predicted VGFR2 Score
Drug
Decoy
E
SRC
VGFR2
PPARA
AKT2
PPARG
CSF1R
RENI
KPCB
CASP3
ROCK1
ConPLex Effect Size (Cohen's d)
Non-contrastive Effect Size (Cohen's d)
Protease (p=0.00E+00)
GPCR (p=2.17E-68)
Kinase (p=0.00E+00)
Nuclear (p=0.00E+00)
No change in effect


Experimental Validation of Kinase-Small Molecule Interactions
A
B
C
MET
ROS1
ALK
MUSK
EPHB1
TGFBR2
FLT3
KIT
EGFR
Target
Molecule
t-SNE 1
t-SNE 2
Surfaceome Proteins Cluster by Function in Embedding Space
Functional Coherence vs. ProtBert
D
E
F
Almen Class
Transporters
Receptors
Enzymes
Miscellaneous
Unclassified
PKinase Domain
Yes
No
CON-Plex Mean log(p/1-p)
ProtBert Mean log(p/1-p)
Cadherin
7TM
Kinase
Immunoglobulin
Other

ACS OMEGA

http://pubs.acs.org/journal/acsodf

Article

TransDTI: Transformer-Based Language Models for Estimating DTIs and Building a Drug Recommendation Workflow

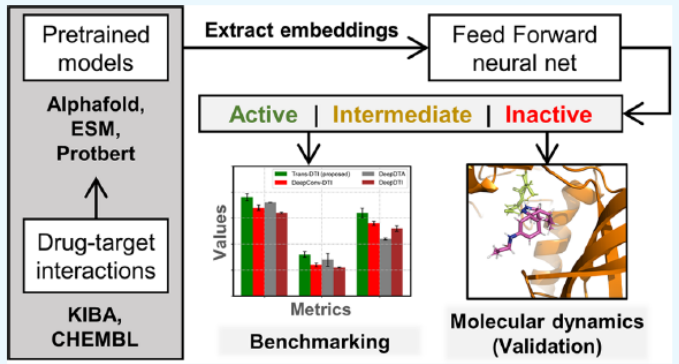
Yogesh Kalakoti, Shashank Yadav, and Durai Sundar*

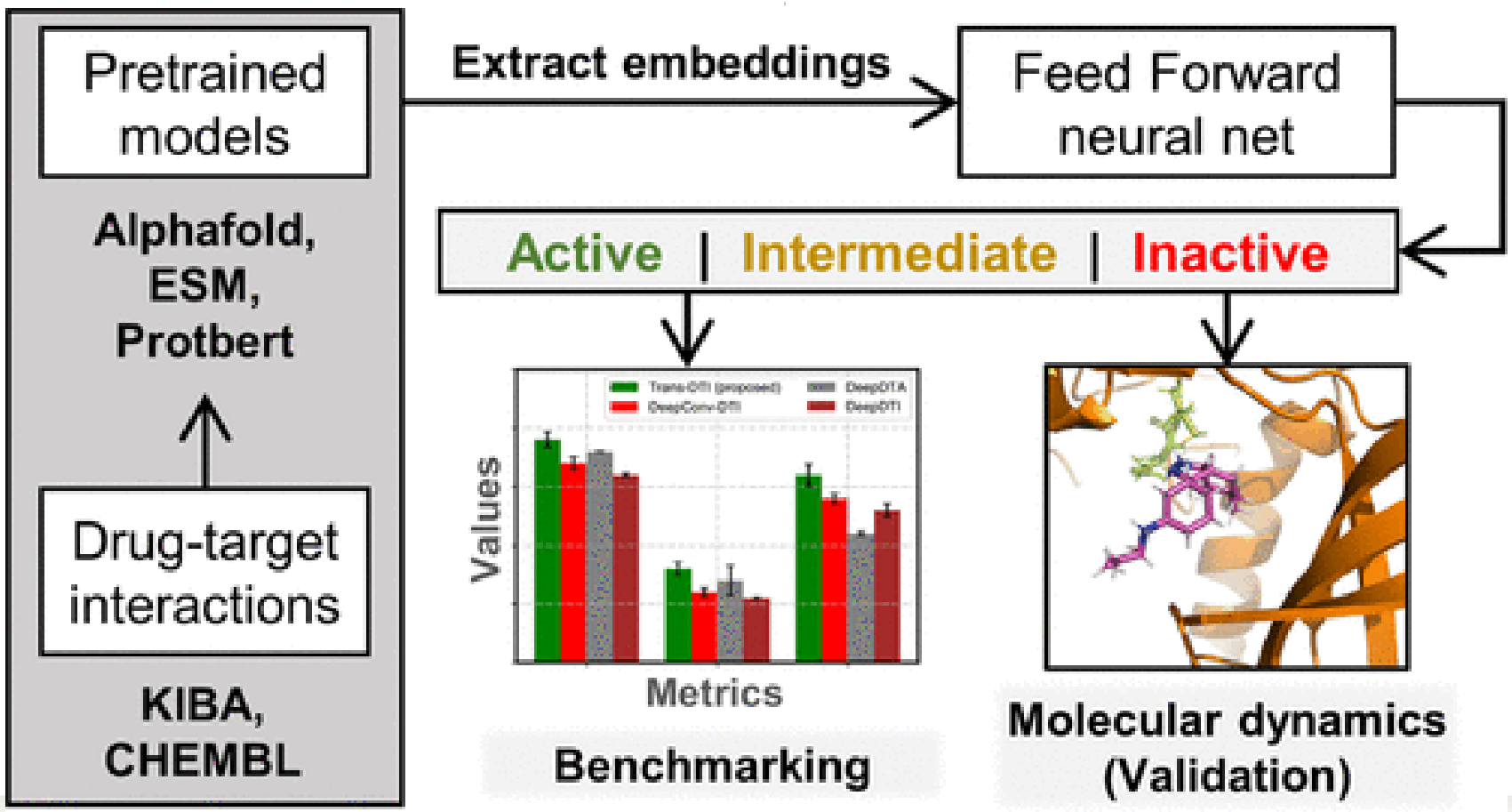
Cite This: *ACS Omega* 2022, 7, 2706−2717

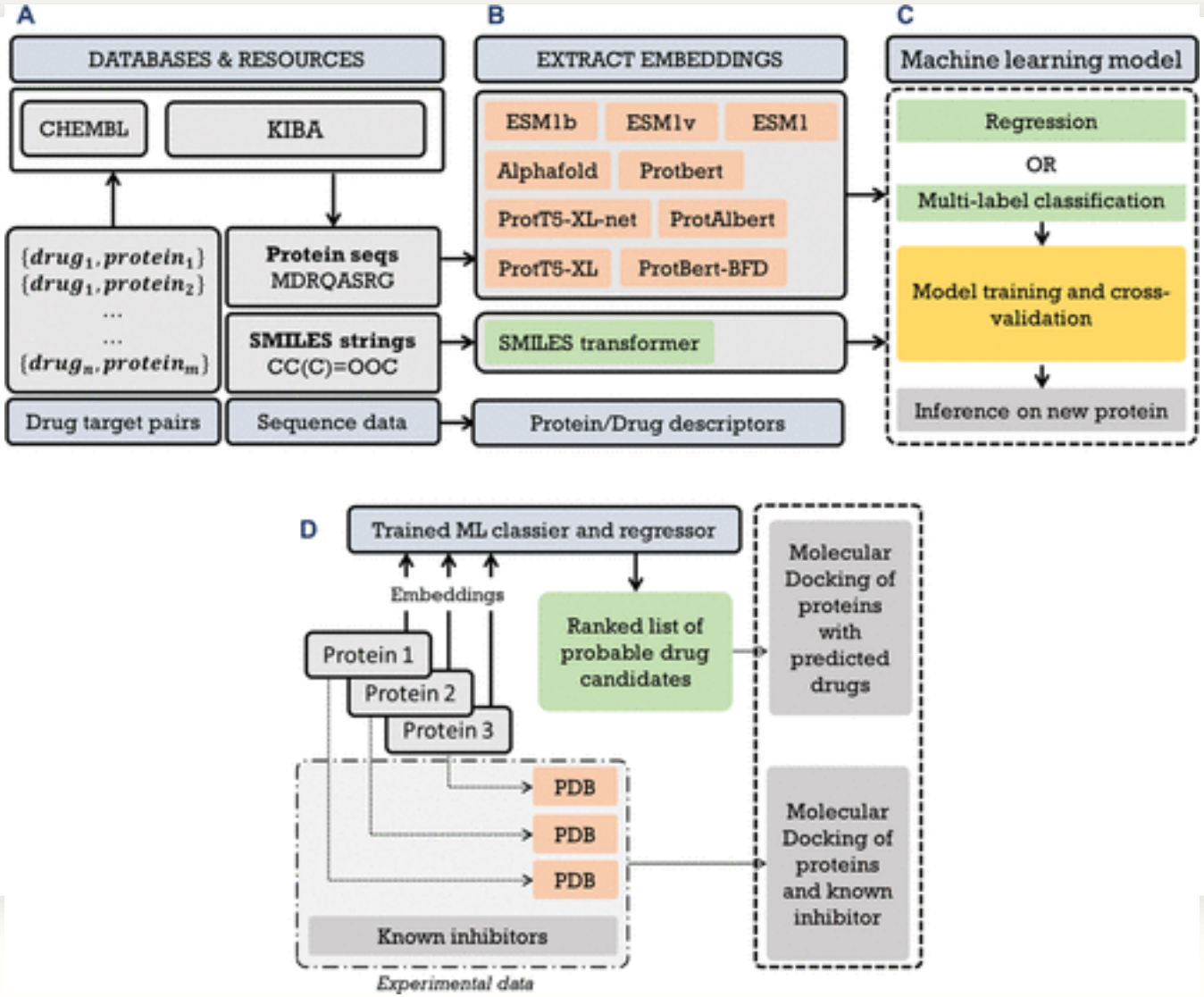
Read Online

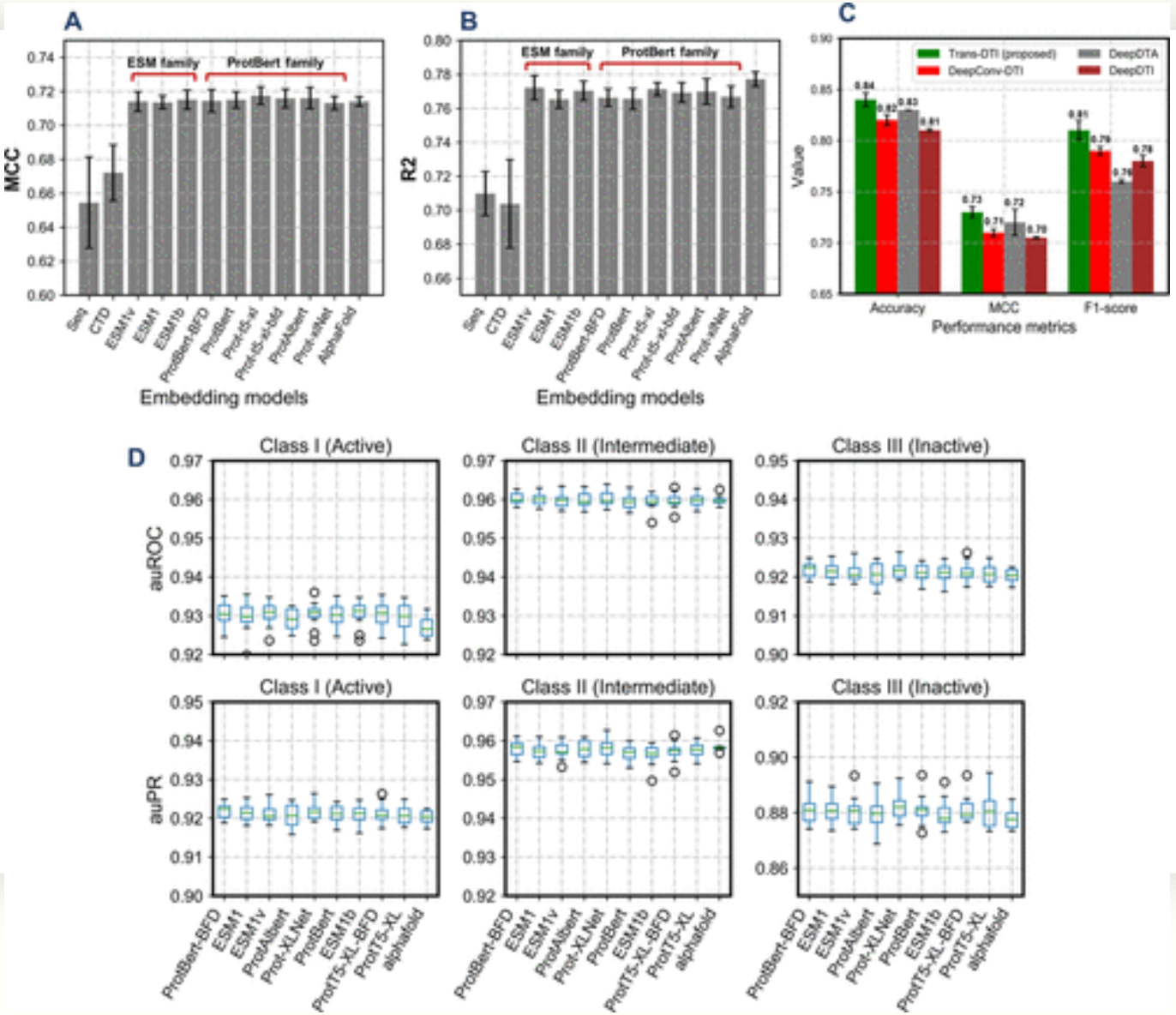
ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: The identification of novel drug−target interactions is a labor-intensive and low-throughput process. In silico alternatives have proved to be of immense importance in assisting the drug discovery process. Here, we present TransDTI, a multiclass classification and regression workflow employing transformer-based language models to segregate interactions between drug−target pairs as active, inactive, and intermediate. The models were trained with large-scale drug−target interaction (DTI) data sets, which reported an improvement in performance in terms of the area under receiver operating characteristic (auROC), the area under precision recall (auPR), Matthew's correlation coefficient (MCC), and *R*2 over baseline methods. The results showed that models based on transformer-based language models effectively predict novel drug−target interactions from sequence data. The proposed models significantly outperformed existing methods like DeepConvDTI, DeepDTA, and DeepDTI on a test data set. Further, the validity of novel interactions predicted by TransDTI was found to be backed by molecular docking and simulation analysis, where the model prediction had similar or better interaction potential for MAP2k and transforming growth factor-β (TGFβ) and their known inhibitors. Proposed approaches can have a significant impact on the development of personalized therapy and clinical decision making.




Pretrained models
Alphafold, ESM, Protbert
Drug-target interactions
KIBA, CHEMBL
Extract embeddings
Feed Forward neural net
Active | Intermediate | Inactive
Values
Metrics
Benchmarking
Molecular dynamics (Validation)


A
B
C
DATABASES & RESOURCES
CHEMBL
KIBA
EXTRACT EMBEDDINGS
ESM1b
ESM1v
ESM1
Alphafold
Protbert
ProtT5-XL-net
ProtAlbert
ProtT5-XL
ProtBert-BFD
Machine learning model
Regression
OR
Multi-label classification
Model training and cross-validation
Inference on new protein
{drug1, protein1}
{drug1, protein2}
...
...
{drugn, proteinm}
Protein seqs
MDRQASRG
SMILES strings
CC(C)=OOC
SMILES transformer
Drug target pairs
Sequence data
Protein/Drug descriptors
D
Trained ML classier and regressor
Embeddings
Protein 1
Protein 2
Protein 3
Ranked list of probable drug candidates
Molecular Docking of proteins with predicted drugs
PDB
PDB
PDB
Known inhibitors
Molecular Docking of proteins and known inhibitor
Experimental data


A
MCC
ESM family
ProtBert family
Seq
CTD
ESM1v
ESM1
ESM1b
ProtBert-BFD
ProtBert
Prot-t5-xl
Prot-t5-xl-bfd
ProtAlbert
Prot-xlNet
AlphaFold
Embedding models
B
R2
ESM family
ProtBert family
Embedding models
C
Trans-DTI (proposed)
DeepConv-DTI
DeepDTA
DeepDTI
Value
Accuracy
MCC
F1-score
Performance metrics
D
auROC
auPR
Class I (Active)
Class II (Intermediate)
Class III (Inactive)
ProtBert-BFD
ESM1
ESM1v
ProtAlbert
Prot-XLNet
ProtBert
ESM1b
ProtT5-XL-BFD
ProtT5-XL
alphafold

validation set	metric	CTD	Seq	DeepDTA	DeepConvDTI	DeepDTI	MolTrans	TransforerCPI	TransDTI[b]
GPCR	accuracy	0.61	0.57	0.73	0.68	0.74	0.72	0.69	**0.77**
	MCC[d]	0.43	0.41	0.58	0.67	0.61	0.59	0.56	**0.69**
	F1 score	0.56	0.53	0.63	**0.68**	0.59	0.62	0.59	0.60
	sensitivity	0.57	0.52	**0.67**	**0.67**	0.54	0.55	0.54	0.58
	specificity	0.69	0.64	0.78	0.72	0.83	0.84	0.80	**0.85**
enzyme	accuracy	0.58	0.53	0.62	0.57	0.71	0.69	0.67	**0.75**
	MCC[d]	0.45	0.38	0.47	0.41	0.53	0.55	0.57	**0.59**
	F1 score	0.53	0.51	0.59	0.56	0.68	0.65	0.64	**0.70**
	sensitivity	0.56	0.50	0.53	0.53	0.67	0.61	0.59	**0.68**
	specificity	0.60	0.57	0.69	0.64	0.64	0.77	0.77	**0.81**
ion channel	accuracy	0.62	0.56	**0.67**	0.59	0.64	0.65	0.62	0.63
	MCC[d]	0.39	0.37	**0.49**	0.38	0.46	0.45	0.49	0.48
	F1 score	0.51	0.48	**0.62**	0.56	0.60	0.56	0.55	0.59
	sensitivity	0.47	0.44	**0.60**	0.52	0.57	0.51	0.50	0.57
	specificity	0.72	0.67	0.72	0.67	0.69	**0.76**	0.72	0.70
nuclear receptors	accuracy	0.54	0.57	0.58	0.61	0.65	0.71	0.64	**0.74**
	MCC[d]	0.34	0.31	0.35	0.43	0.47	0.52	0.48	**0.55**
	F1 score	0.51	0.52	0.53	0.56	0.62	0.58	0.50	**0.68**
	sensitivity	0.41	0.47	0.50	0.50	0.61	0.52	0.45	**0.63**
	specificity	0.65	0.66	0.67	0.72	0.69	0.83	0.77	**0.84**

Chemical Science

ROYAL SOCIETY OF CHEMISTRY

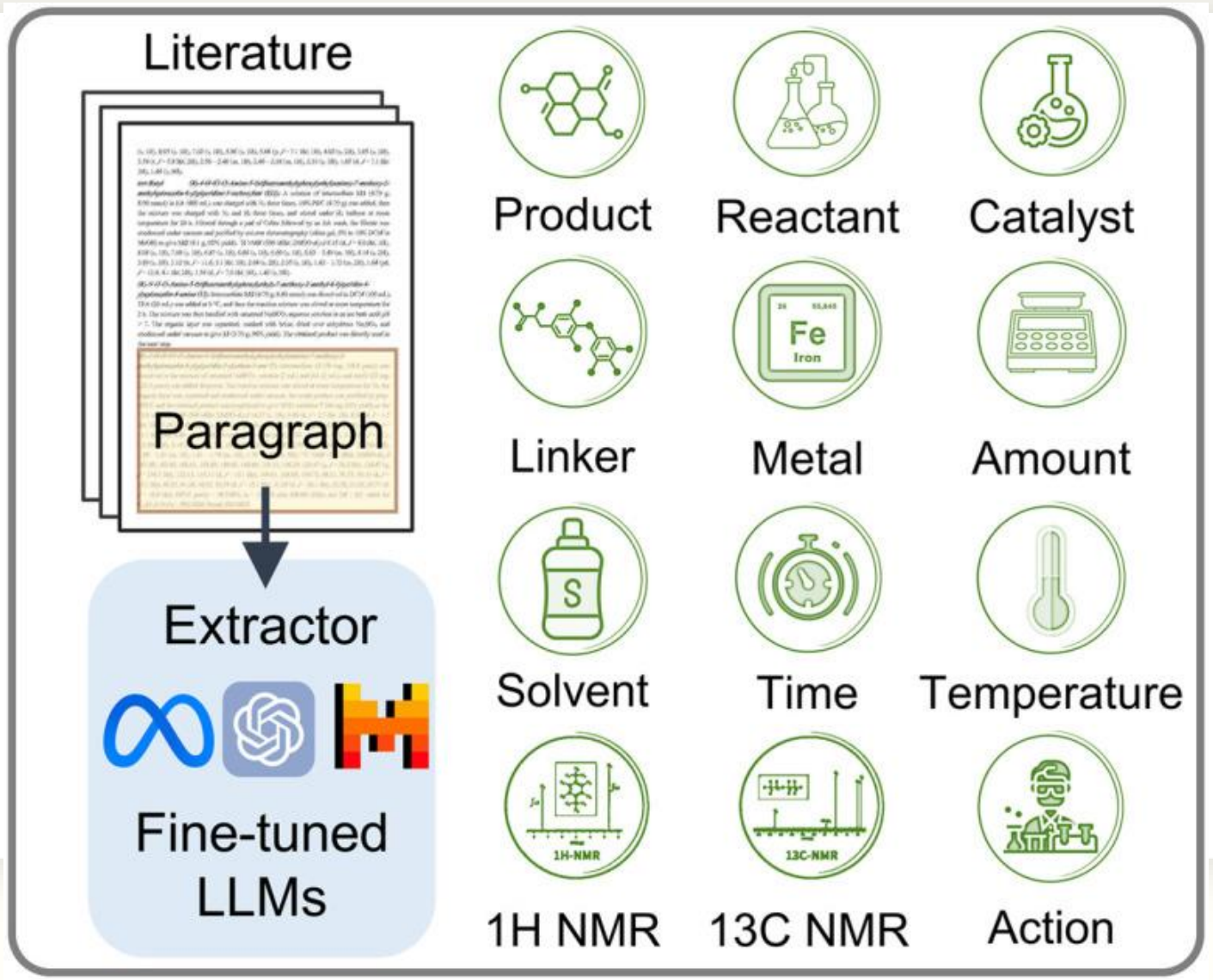
EDGE ARTICLE

Cite this: *Chem. Sci.*, 2024, **15**, 10600

All publication charges for this article have been paid for by the Royal Society of Chemistry

Fine-tuning large language models for chemical text mining†

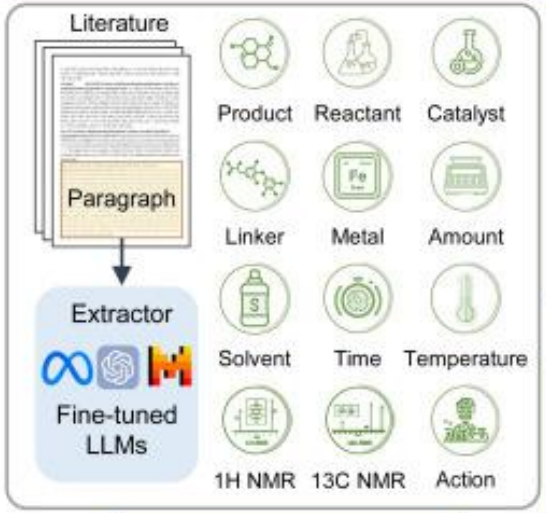
Wei Zhang,‡[ab] Qinggong Wang,‡[c] Xiangtai Kong,[ab] Jiacheng Xiong,[ab] Shengkun Ni,[ab] Duanhua Cao,[ad] Buying Niu,[ab] Mingan Chen,[aef] Yameng Li,[g] Runze Zhang,[ab] Yitian Wang,[ab] Lehan Zhang,[ab] Xutong Li,[ab] Zhaoping Xiong,[g] Qian Shi,[f] Ziming Huang,[h] Zunyun Fu*[a] and Mingyue Zheng *[abc]


Literature
Paragraph
Extractor
Fine-tuned LLMs
Product
Reactant
Catalyst
Linker
Fe
Iron
Metal
Amount
S
Solvent
Time
Temperature
1H-NMR
13C-NMR
1H NMR
13C NMR
Action

Chemical Text Mining

Paragraph2Compound

- 1-[4-(3,4-difluoro-1H-pyrrol-1-yl)-2-hydroxyphenyl]-5-methoxy-3-(1-phenyl-1H-pyrazol-5-yl)pyridazin-4(1H)-one
- Iodomethane
- potassium carbonate
- N,N-dimethylformamide
- ethyl acetate
- brine
- magnesium sulfate
- tetrahydrofuran



Paragraph2Action

- PARTITION with saturated aqueous solution of NH4Cl and EtOAc;
- COLLECTLAYER organic;
- DRYSOLUTION over Na2SO4;
- CONCENTRATE;
- PURIFY;
- YIELD compound X (44 mg, 85%)

Paragraph2MOFInfo

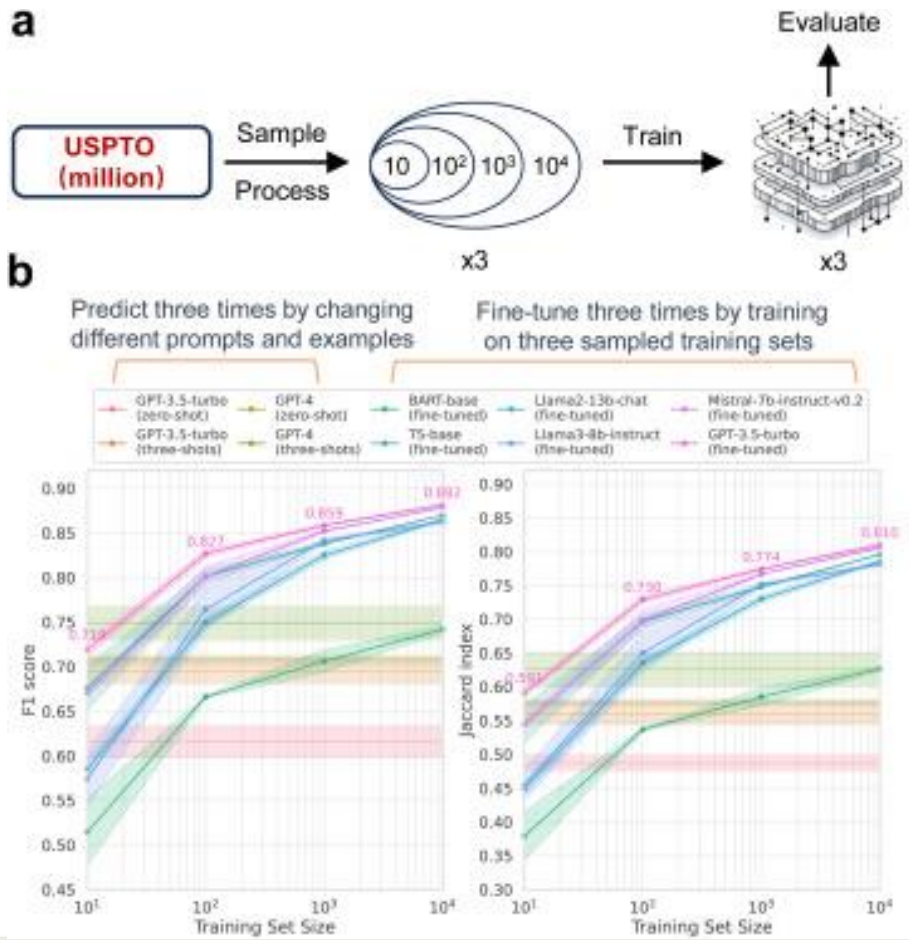
Compound name : MOF-808
Metal source: ZrOCl2·8H2O
Metal amount: 970 mg
Linker : 1,3,5-benzenetricarboxylic acid
Linker amount : 210 mg
Modulator : formic acid
Modulator amount or volume : 30 mL
Solvent: DMF
Solvent volume: 30 mL
Reaction temperature: 100 °C
Reaction time: a day

Paragraph2RXNRole

Product: (E)-3-methyleneisoindolin-1-one
Reactant: 3a
Catalysts: K_2CO_3, Bu_3P
Temperature: 60℃

Paragraph2NMR

IUPAC name: 2,5-Dioxocyclopentyl 2-(2,2,2-trifluoroacetamido)acetate
1H NMR cond.: 270 MHz, CD3OD
1H NMR data: 4.44, 2.84
1H NMR text: 1H-NMR (270 MHz, CD3OD) δ: 4.44 (s, 2H, CH2NH), 2.84 (s, 4H, 2×CH2) ppm
13C NMR cond.: 67.5 MHz, ACETONE-d6
13C NMR data: 170.1, 165.7, 158.2, 116.8, 39.5, 26.2
13C NMR text: 13C NMR (67.5 MHz, ACETONE-d6) δ: 170.1 (2×CO), 165.7, 158.2 (q, 2JCF = 37.4 Hz), 116.8 (q, 1JCF = 287.0 Hz), 39.5, 26.2 (2×CH2) ppm



c

Zero-shot prompt:

Please just extract all compound names in the paragraph, the compound names should be split in " | ". If there is no compound name in whole paragraph, please return "N/A".

{Input Paragraph}

Few-shots prompt:

Please just extract all compound names in the paragraph, the compound names should be split in " | ". If there is no compound name in whole paragraph, please return "N/A".

Example x N:

Input:

Compound 610 (102 mg, 0.366 mmol) was dissolved in DMF (3 mL), and 3-pyridylacetic acid hydrochloride (635 mg, 3.66 mmol), EDC hydrochloride (702 mg, 3.66 mmol), 1-hydroxybenzotriazole monohydrate (561 mg, 3.66 mmol) and triethylamine (0.510 mL, 3.66 mmol) were added thereto, followed by stirring at 80° C. for 10 hours.

Output:

Compound 610 | DMF | 3-pyridylacetic acid hydrochloride | EDC hydrochloride | 1-hydroxybenzotriazole monohydrate | triethylamine

{Input Paragraph}

Tx-LLM: A Large Language Model for Therapeutics

Juan Manuel Zambrano Chaves[*,1], Eric Wang[*,2], Tao Tu[2], Eeshit Dhaval Vaishnav, Byron Lee, S. Sara Mahdavi[2], Christopher Semturs[1], David Fleet[2], Vivek Natarajan[†,1] and Shekoofeh Azizi[†,‡,2]

[1]Google Research, [2]Google DeepMind

Therapeutics Data Commons (TDC)

- Small molecules
- Target discovery
- Macromolecules
- Efficacy & safety
- Cell & gene therapies
- Manufacturing

Classification

Drug	Y
Cc1nnsc1SCC(=O)O	0
CCn1c(N)nc2ccccc21	0
O=C(O)CC(O)C(=O)O	1
...	...

Generation

Input	Output
NC1C2ccccc2C(=O)N1Cc1ccccc1	CCOC(=O)CBr
[NH2:1][C:2]1[S:3][C:4]([S:7][C:8]#[N:9])=[CH:5][N:6]=1	[NH2:1][C:2]1[S:3][CH:4]=[CH:5][N:6]=1.[S-:7][...
...	...

Regression

Drug1	Drug2	Cell Line ID	Synergy
C(=O)(N)NO	N.N.Cl[Pt+2]Cl	786-0	1.059215
C(=O)(N)NO	N.N.Cl[Pt+2]Cl	ACHN	-1.159521
CS(=O)(=O)CCCCO	N.N.Cl[Pt+2]Cl	OVCAR3	2.033875
...	...	...	...

Therapeutics Instruction Tuning (TxT)

66 Tasks **709 Datasets**

10 Million (Instructions, Answer) Pairs

Classification

Given a drug SMILES string, predict whether it (A) is not mutagenic (B) Is mutagenic

Drug: Cc1nnsc1SCC(=O)O

Answer: (A)

Generation

Given a product SMILES string, predict the reactant SMILES string.

Product: NC1C2ccccc2C(=O)N1Cc1ccccc1

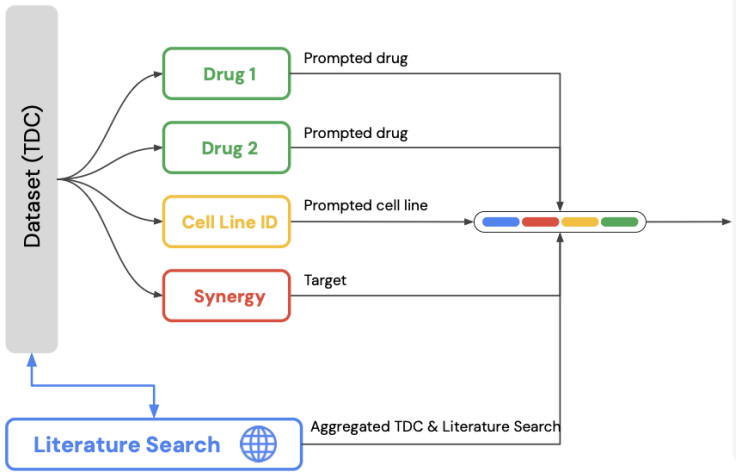
Answer: CCOC(=O)CBr

Regression

Given two drug SMILES strings and a cell line, predict the drug synergy from 0 to 1000.

Drug 1: C(=O)(N)NO
Drug 2: N.N.Cl[Pt+2]Cl
Cell line: 786-0, renal cell adenocarcinoma

Answer: 562



TxT Prompt

Instructions: Answer the following question about drug synergy

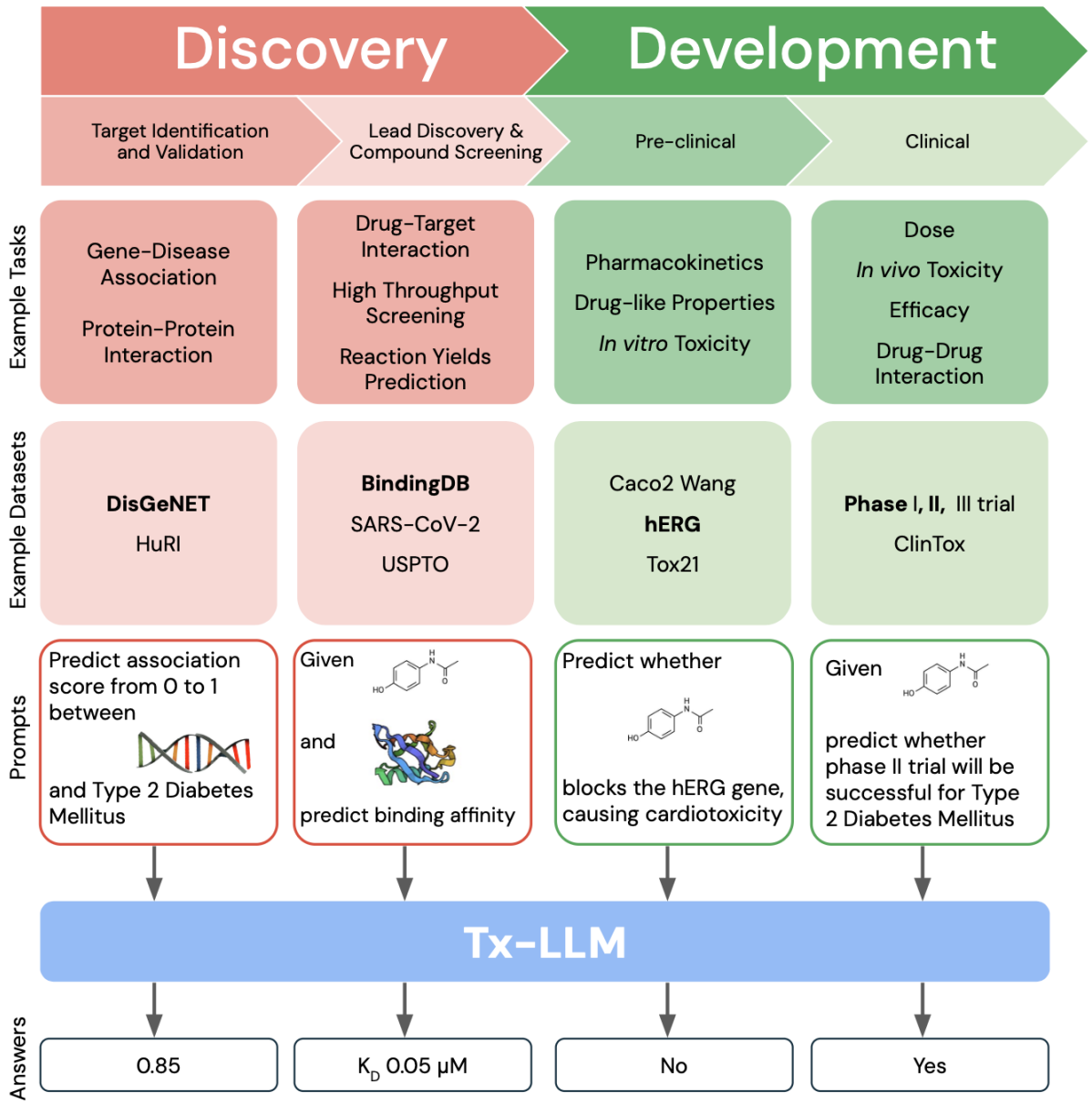
Context: Synergy is a dimensionless measure of deviation of an observed drug combination response from the expected effect of non-interaction. Drug combinations were tested against cancer cell lines...

Question: Given two drug SMILES strings and a cell line description, predict the normalized drug synergy from 000 to 1000, where 000 is minimum drug synergy and 1000 is maximum drug synergy.

Drug1 SMILES: C(=O)(N)NO
Drug2 SMILES: N.N.Cl[Pt+2]Cl
Cell line description: 786-0, renal cell adenocarcinoma
Answer: 562

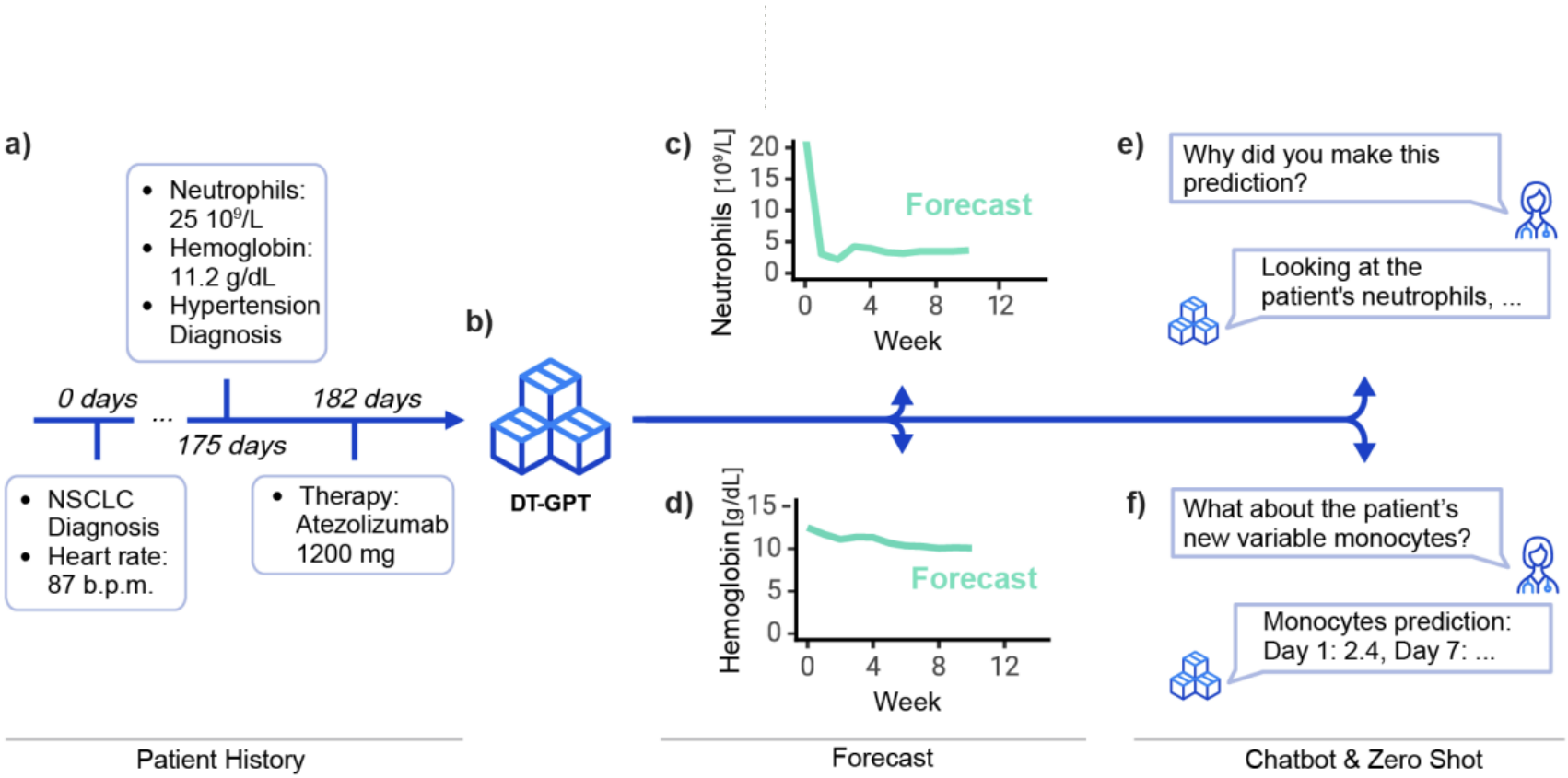
Drug1 SMILES: CS(=O)(=O)CCCCO
Drug2 SMILES: N.N.Cl[Pt+2]Cl
Cell line description: OVCAR3, ovarian cell adenocarcinoma
Answer: 678

...


Discovery
Development
Target Identification and Validation
Lead Discovery & Compound Screening
Pre-clinical
Clinical
Example Tasks
Gene-Disease Association
Protein-Protein Interaction
Drug-Target Interaction
High Throughput Screening
Reaction Yields Prediction
Pharmacokinetics
Drug-like Properties
In vitro Toxicity
Dose
In vivo Toxicity
Efficacy
Drug-Drug Interaction
Example Datasets
DisGeNET
HuRI
BindingDB
SARS-CoV-2
USPTO
Caco2 Wang
hERG
Tox21
Phase I, II, III trial
ClinTox
Prompts
Predict association score from 0 to 1 between
and Type 2 Diabetes Mellitus
Given
and
predict binding affinity
Predict whether
blocks the hERG gene, causing cardiotoxicity
Given
predict whether phase II trial will be successful for Type 2 Diabetes Mellitus
Tx-LLM
Answers
0.85
K_D 0.05 μM
No
Yes

Large Language Models forecast Patient Health Trajectories enabling Digital Twins

Nikita Makarov[1,2,3,†], Maria Bordukova[1,2,3,†],
Raul Rodriguez-Esteban[4,#], Fabian Schmich[1,#], Michael P. Menden[2,5,#]



LLM 발전 방향

LLM 기술 핵심 동향

- 멀티모달 확장과 에이전트 기술로 인간-AI 상호작용 혁신
- RAG, Function Calling으로 지식 활용 및 작업 수행 능력 향상
- Context Length 확장, MoE 기술로 더 복잡한 문제 해결
- PEFT, RLHF로 맞춤형 고품질 모델 구현 용이

기술 발전 방향성

- 모델 경량화와 친환경 AI가 대중화 촉진
- 오픈소스 LLM의 성장으로 접근성 확대
- 개인화 LLM과 안전성이 미래 핵심 과제

LLM 기술의 미래

개인화
개인 맞춤형 AI

안전성
신뢰와 편향 제거

효율성
저전력 고성능

LLM은 분산 지능형 시스템으로 발전하며
인간-AI 협업 생태계를 구축할 것입니다

산업적 영향

모든 산업 분야에 지능형 자동화와 의사 결정 지원 시스템으로 혁신 촉진

사회적 영향

정보 접근성 증대와 함께 AI 윤리, 데이터 프라이버시 등 사회적 고려 필요

"LLM 기술은 이제 단순한 언어 모델을 넘어 인간의 지적 파트너로 진화하고 있습니다"

멀티모달& 에이전틱LLM

멀티모달 LLM

- 이미지, 오디오, 비디오와 텍스트 등 복수의 데이터 타입을 동시에 이해 및 생성
- 자연스러운 다중 모달 대화 및 콘텐츠 이해 가능
- 주요 사례: GPT-4 Turbo, Gemini, Claude 3

에이전틱 LLM

- 자율 판단 및 일련의 작업 수행이 가능한 AI 에이전트
- 외부 도구 활용과 복잡한 문제 해결 자동화
- 대표 프레임워크: AutoGPT, BabyAGI

이미지 이해 & 생성

시각 데이터 이해 및 이미지 기반 답변 생성

"이 차트는 작년 대비 15% 매출 증가를 보여줍니다."

오디오/비디오 처리

음성, 영상 내용 이해 및 텍스트 변환

"회의 내용을 자동 요약하고 핵심 작업 목록 생성"

코드 자동화

코드 생성 및 디버깅 자동화

"API 스펙을 읽고 자동으로 클라이언트 코드 생성"

복합 작업 수행

순차적 작업 계획 및 실행

"데이터 수집, 분석, 시각화 자동 파이프라인"

RAG & Function Calling

RAG (Retrieval-Augmented Generation)

- 외부 지식 소스에서 관련 정보를 검색하여 LLM 응답 생성 보강
- 최신 정보 활용 및 사실 기반 응답으로 정확성 향상
- 주요 도구: LangChain, LlamaIndex

Function Calling & Tool Use

- LLM이 외부 API와 도구를 호출하여 작업 자동화
- 구조화된 출력 형식으로 시스템 통합 용이
- 대표 사례: OpenAI Functions, GPTs, Gemini Tools

지식 검색 및 응답 생성

최신 외부 데이터로 정확한 정보 제공

"2023년 최신 연구 데이터를 검색하여 답변 생성"

벡터 데이터베이스 활용

의미적 유사성 기반 검색 및 응답 증강

"사내 문서 저장소에서 관련 정책 검색 및 요약"

API 연동 자동화

다양한 서비스 API 연결 및 작업 수행

"날씨 API에서 데이터 가져와 일정 조정 제안"

구조화된 출력 생성

JSON 형식 응답으로 시스템 통합 지원

"사용자 요청을 분석하여 JSON 형식 매개변수로 변환"

LangChain LlamaIndex OpenAI Functions Gemini Tools

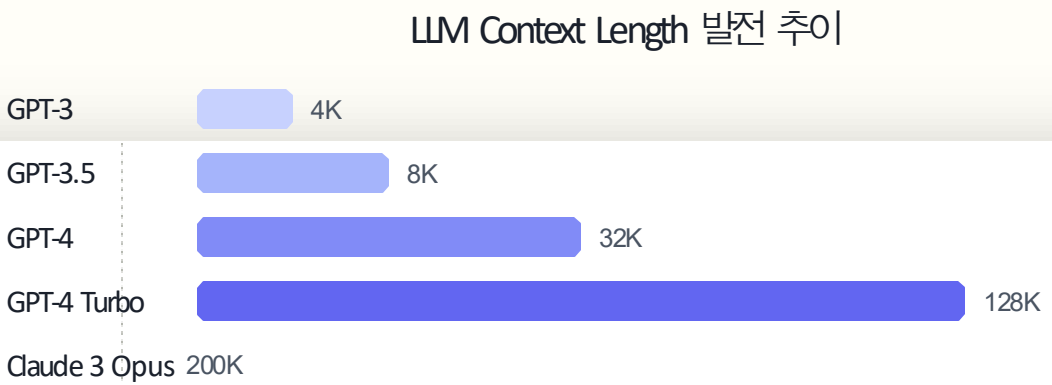
Context Length 확장 &Mixture of Experts

Context Length 확장

- LLM이 동시에 처리 가능한 텍스트 길이의 비약적 증가
- 책 전체, 긴 문서, 대화 기록 등 장문 처리 능력
- 주요 사례: GPT-4 Turbo (128k), Claude 3 Opus (200k)

Mixture of Experts (MoE)

- 작업별 특화된 전문가 네트워크들을 선택적으로 활용
- 모델 크기 증가 대비 계산 효율성과 성능 향상
- 적용 사례: Gemini 1.5, GLaM, Switch Transformer



MoE 작동 방식

라우터가 입력에 적합한 전문가 모듈만 활성화

"계산 비용 절감 및 성능 향상 동시 달성"

효율적 스케일링

대규모 모델의 효율적 확장 가능

"Gemini 1.5: 고품질 MoE 기반 멀티모달 모델"

GPT-4 Turbo Claude 3 Opus Gemini 1.5 GLaM

파라미터 효율적 튜닝 & RLHF

파라미터 효율적 튜닝

- 전체 모델이 아닌 소수의 파라미터만 학습하여 자원 효율화
- 기본 모델 성능 유지하며 특정 도메인 적응성 향상
- 주요 기법: LoRA, QLoRA, Prefix Tuning

RLHF

- 인간 피드백 기반 강화학습 통한 모델 정렬(Alignment)
- 유해하지 않고 유용한 응답 생성 능력 강화
- 대표 적용: ChatGPT, Claude, Gemini

파라미터 효율적 튜닝 (PEFT) 방식 비교

전체 파인튜닝
모든 파라미터 학습 (100%)
높은 계산 비용

→

PEFT (LoRA 등)
일부 파라미터만 (< 1%)
낮은 계산 비용

LoRA 작동 원리:
원본 가중치는 고정된 상태로 유지하고, 저차원 적응 매트릭스를 학습하여 원본 레이어에 추가

RLHF 프로세스

초기 LLM → 인간 평가 → 보상 모델 → 강화학습

RLHF 주요 성과:
인간 선호도에 부합하는 응답 생성, 유해한 내용 회피, 유용성과 정확성 향상

LoRA QLoRA ChatGPT Claude

Hallucination 해결& 프롬프트 엔지니어링

Hallucination 해결

- 사실 오류(Hallucination) 감소를 위한 검증 메커니즘 강화 Self-Consistency 기법으로 응답 일관성 검증
- 주요 기법: Multi-Step Reasoning, Knowledge Grounding

프롬프트 엔지니어링 자동화

- 최적의 프롬프트 자동 탐색 및 발견 기술
- 자체 최적화 및 진화적 프롬프트 생성
- 대표 기술: APE, Promptbreeder

Self-Consistency

여러 경로로 추론 후 일관된 답변 도출

"다양한 사고 경로를 통해 도출된 답변의 일치도 검증"

Multi-Step Reasoning

단계적 추론으로 결론 품질 향상

"각 추론 단계의 정확성을 검증하며 논리적 결론 도출"

Automatic Prompt Engineer

LLM으로 최적 프롬프트 자동 탐색

"메타-학습을 통한 프롬프트 성능 최적화"

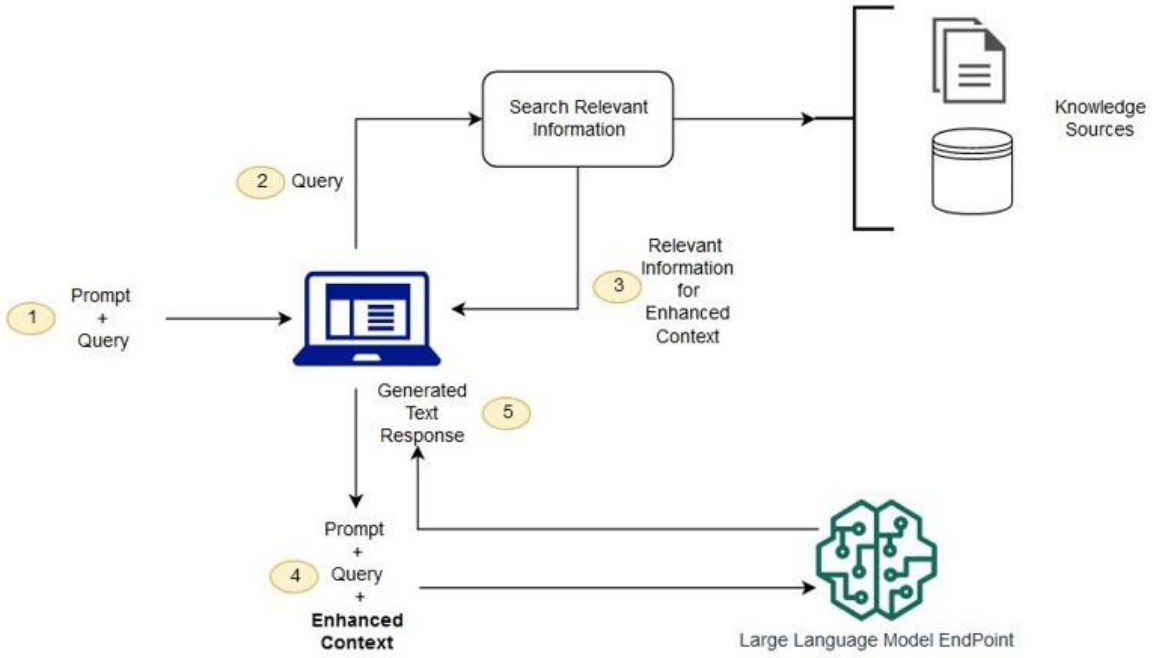
Promptbreeder

진화 알고리즘 활용 프롬프트 개선

"자가 진화하는 프롬프트로 복잡한 태스크 해결력 향상"

Hallucination 감소 효과 비교

- 검색 증강 생성(Retrieval-Augmented Generation, RAG)



- 외부데이터생성
- 관련정보 검색
- LLM 프롬프트 확장
- 외부데이터 업데이트

- RAG vs. Fine-tuning

항목	RAG	파인튜닝
사용 데이터	외부 데이터 검색	사전 학습 데이터
시간·비용 소모	적음	많음
베이스 모델 개선	불가	가능
환각 현상 발생 가능성	낮음	사전 학습되지 않은 데이터에서 발생 가능성 있음
데이터 변동성	역동적 데이터	정적 데이터
의사 결정 과정	검색된 문서 확인 가능	알 수 없음

멀티모달 생의학 파운데이션 모델 상세 분석

Geneformer, scGPT, BioGPT 등 주요 모델 비교 및 성능 평가

주요 생의학 파운데이션 모델

Geneformer

초기 버전(V1)은 대규모 단일세포 RNA 데이터로 사전학습. V2는 이전보다 크게 확장된 세포 데이터 활용

특징: 유전자 발현 프로파일 입력, BERT 구조, Masked Gene Prediction

scGPT

다수의 단일세포 데이터셋, 방대한 규모의 세포 데이터로 훈련 **특**

징: GPT 구조, 전사체 프로파일, 공간 정보 통합 (scGPT-spatial)

BioGPT

PubMed에서 추출한 광범위한 생의학 문헌 데이터로 훈련

특징: GPT 기반 생물의학 텍스트 처리, 분자 구조 이해

Med-PaLM

의학 텍스트, 임상 기록, 의료 문헌 대규모 학습

특징: 의학 질의응답 특화, 자연어 기반 임상 추론

멀티모달 통합 접근법

Mixture of Experts(MoE) 구조로 효율적 데이터 통합

모델 성능 비교 및 적용 사례

모델명	세포 유형 분류	유전자 상호작용	통합 분석력
Geneformer	우수	보통	보통
scGPT	매우 우수	우수	우수
BioGPT	보통	우수	보통
Med-PaLM	보통	보통	매우 우수

주요 적용 사례

희귀질환 변이 영향 예측

유전자 변이의 기능적 영향을 정확하게 예측, 희귀 변이 관련 질병 진단에 활용

약물-표적 상호작용 발견

신약 타겟 발굴 시 후보 약물과 타겟 단백질 간의 결합 친화도와 작용 기전 예측

세포 분화경로 예측

다양한 세포 상태 간 전환 과정 예측, 줄기세포 분화 및 세포 운명 결정 연구에 활용

에이전틱 AI 자가 진화 메커니즘

자기 진화(Self-evolving) 메커니즘

- **데이터 소스 통합**으로 신규 가설 생성
  문헌, 실험 데이터, 공간오믹스 정보 종합 분석
- **자체 검증 및 평가** 시스템 구축
  결과의 일관성, 신뢰성, 통계적 유의성 자동 평가
- **자율적 피드백 루프**로 지속적 개선
  실험 결과 기반 가설 수정 및 반복적 개선

DREAM 사례 연구

- 생물의학 데이터 기반 **자율적 연구**
  데이터 중심 추론으로 가설 도출 및 검증
  [Nature: CoScientist (2023)]
- **우수한 효율성** 입증
  넓은 탐색 공간에서 최적 가설 선별 가능
- **자기 교정 및 최적화** 능력
  실험-분석 반복 과정에서 자가 개선

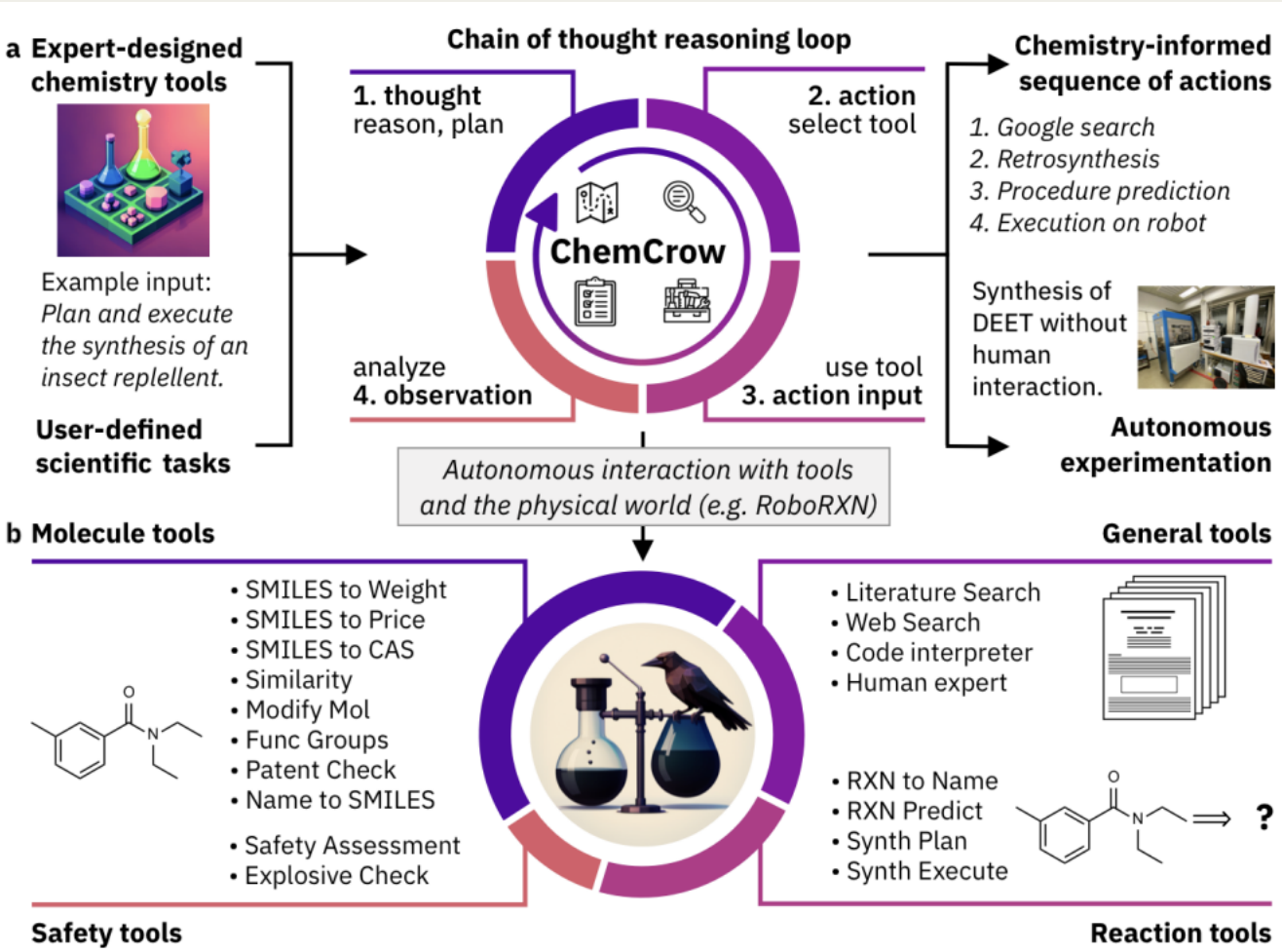
자가 진화 사이클 구조

바이오마커 발굴
자동 검증 과정으로 바이오마커 발굴 효율성 향상

약물 반응성 예측
새로운 약물-타겟 관계 발견 능력 향상

Augmenting large language models with chemistry tools

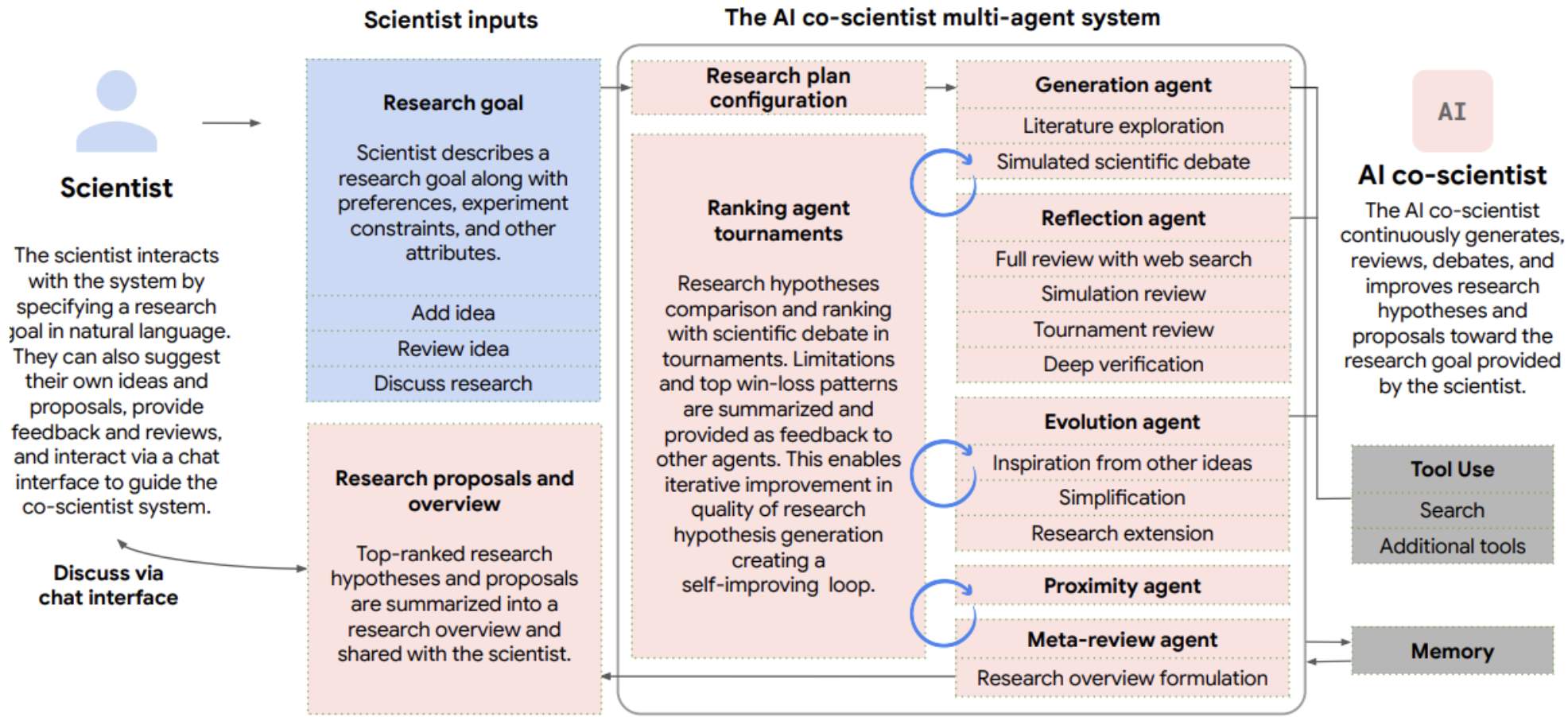
Andres M. Bran[12*] **Sam Cox**[3*] **Oliver Schilter**[24]
Carlo Baldassari[4] **Andrew D. White**[3] **Philippe Schwaller**[12]
[1] Laboratory of Artificial Chemical Intelligence (LIAC), ISIC, EPFL
[2]National Centre of Competence in Research (NCCR) Catalysis, EPFL
[3] Department of Chemical Engineering, University of Rochester
[4] Accelerated Discovery, IBM Research – Europe
*Contributed equally.
andrew.white@rochester.edu
philippe.schwaller@epfl.ch


a Expert-designed chemistry tools
Example input: Plan and execute the synthesis of an insect replellent.
User-defined scientific tasks
Chain of thought reasoning loop
1. thought reason, plan
2. action select tool
use tool 3. action input
analyze 4. observation
ChemCrow
Chemistry-informed sequence of actions
1. Google search
2. Retrosynthesis
3. Procedure prediction
4. Execution on robot
Synthesis of DEET without human interaction.
Autonomous experimentation
Autonomous interaction with tools and the physical world (e.g. RoboRXN)
b Molecule tools
• SMILES to Weight
• SMILES to Price
• SMILES to CAS
• Similarity
• Modify Mol
• Func Groups
• Patent Check
• Name to SMILES
• Safety Assessment
• Explosive Check
Safety tools
General tools
• Literature Search
• Web Search
• Code interpreter
• Human expert
• RXN to Name
• RXN Predict
• Synth Plan
• Synth Execute
Reaction tools

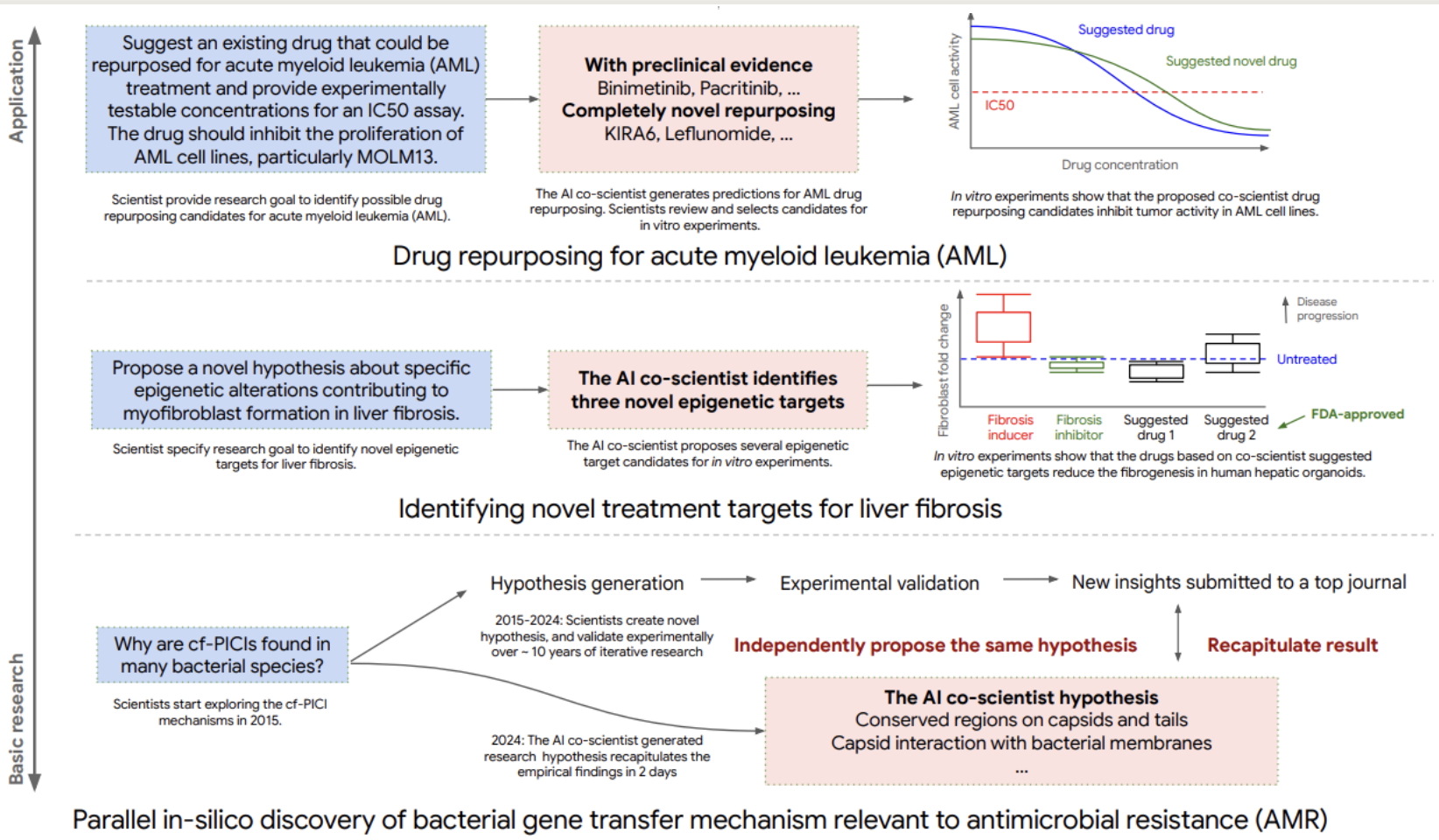
Towards an AI co-scientist

Juraj Gottweis[*, ‡, 1], Wei-Hung Weng[*, ‡, 2], Alexander Daryin[*,1], Tao Tu[*,3],
Anil Palepu[2], Petar Sirkovic[1], Artiom Myaskovsky[1], Felix Weissenberger[1],
Keran Rong[3], Ryutaro Tanno[3], Khaled Saab[3], Dan Popovici[2], Jacob Blum[7], Fan Zhang[2],
Katherine Chou[2], Avinatan Hassidim[2], Burak Gokturk[1],
Amin Vahdat[1], Pushmeet Kohli[3], Yossi Matias[2],
Andrew Carroll[2], Kavita Kulkarni[2], Nenad Tomasev[3], Yuan Guan[7],
Vikram Dhillon[4], Eeshit Dhaval Vaishnav[5], Byron Lee[5],
Tiago R D Costa[6], José R Penadés[6], Gary Peltz[7],
Yunhan Xu[3], Annalisa Pawlosky[1, ‡], Alan Karthikesalingam[2, ‡] and Vivek Natarajan[2, ‡]

[1]Google Cloud AI Research, [2]Google Research, [3]Google DeepMind,
[4]Houston Methodist, [5]Sequome,
[6]Fleming Initiative and Imperial College London,
[7]Stanford University School of Medicine



The AI co-scientist system design


Application
Basic research
Suggest an existing drug that could be repurposed for acute myeloid leukemia (AML) treatment and provide experimentally testable concentrations for an IC50 assay. The drug should inhibit the proliferation of AML cell lines, particularly MOLM13.
Scientist provide research goal to identify possible drug repurposing candidates for acute myeloid leukemia (AML).
With preclinical evidence
Binimetinib, Pacritinib, ...
Completely novel repurposing
KIRA6, Leflunomide, ...
The AI co-scientist generates predictions for AML drug repurposing. Scientists review and selects candidates for in vitro experiments.
Suggested drug
Suggested novel drug
IC50
AML cell activity
Drug concentration
In vitro experiments show that the proposed co-scientist drug repurposing candidates inhibit tumor activity in AML cell lines.
Drug repurposing for acute myeloid leukemia (AML)
Propose a novel hypothesis about specific epigenetic alterations contributing to myofibroblast formation in liver fibrosis.
Scientist specify research goal to identify novel epigenetic targets for liver fibrosis.
The AI co-scientist identifies three novel epigenetic targets
The AI co-scientist proposes several epigenetic target candidates for in vitro experiments.
Fibroblast fold change
Disease progression
Untreated
Fibrosis inducer
Fibrosis inhibitor
Suggested drug 1
Suggested drug 2
FDA-approved
In vitro experiments show that the drugs based on co-scientist suggested epigenetic targets reduce the fibrogenesis in human hepatic organoids.
Identifying novel treatment targets for liver fibrosis
Why are cf-PICIs found in many bacterial species?
Scientists start exploring the cf-PICI mechanisms in 2015.
Hypothesis generation
Experimental validation
New insights submitted to a top journal
2015-2024: Scientists create novel hypothesis, and validate experimentally over ~ 10 years of iterative research
Independently propose the same hypothesis
Recapitulate result
2024: The AI co-scientist generated research hypothesis recapitulates the empirical findings in 2 days
The AI co-scientist hypothesis
Conserved regions on capsids and tails
Capsid interaction with bacterial membranes
...
Parallel in-silico discovery of bacterial gene transfer mechanism relevant to antimicrobial resistance (AMR)