공학사 졸업논문

NMR 스펙트럼 기반 분자 구조 예측을 위한 생성 모델 개발

Generative model for the prediction of molecular structures based on NMR spectra

2024년 2월

서울대학교 공과대학
화학생물공학부
석 정 헌

국문초록

본 연구에서는 NMR 스펙트럼과 분자식을 입력으로 받아 이에 해당하는 분자 구조를 예측하는 기계 학습 모델 개발을 목표로 연구를 진행했다. 모델은 크게 SMILES auto encoder와 NMR encoder로 나누어 학습을 진행했으며, 최종 모델은 NMR encoder와 SMILES decoder를 연결하여 만들었다. 모델 학습 후 테스트 결과, 생성된 대부분의 SMILES는 문법적으로 유효하지 않고, 유효한 것은 C 위주로 이루어진 것을 확인할 수 있었다. 데이터 및 파라미터 추가, 그리고 모델의 수정을 통해 결과를 개선할 수 있을 것으로 예상된다.

주요어 : 생성 모델, NMR, SMILES, 기계 학습

목 차

그 림 목 차

1. 서론

1-1. SMILES(Simplified Molecular-Input Line-Entry System)

SMILES는 Simplified Molecular-Input Line-Entry System의 약자로, 화학 물질의 구조를 짧은 ASCII string을 이용하여 문자열로 나타낸 것이다.[1] 분자를 SMILES, 즉 문자열로 바꾸어 컴퓨터를 이용해 다룰 수 있다. 이때 하나의 분자는 여러 개의 SMILES로 표현될 수 있는 문제가 있다. 예를 들어, “CCO”, “OCC”, “C(O)C”는 모두 에탄올을 표현하는 SMILES이다. 이를 보완한 것이 canonical SMILES이다. 이를 이용하면 하나의 분자를 표현하는 SMILES는 하나로 결정된다.

1-2. Molecular fingerprint, Tanimoto coefficient

분자식이나 SMILES만으로는 분자의 성질을 쉽게 유추하거나 유사성을 비교하기 어렵다. 이러한 역할들을 보다 쉽고 빠르게 수행하기 위해 각 분자에 일종의 지문을 부여하는 것이 Molecular fingerprint이다. 이를 위해 보편적으로 Morgan algorithm을 많이 사용하고, 이를 이용하여 생성한 분자 지문을 Morgan fingerprint라 한다.[2]

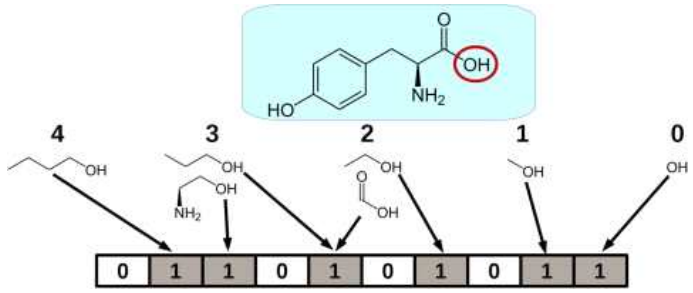

그림 1-1 Morgan fingerprint에 대한 Graphical abstract[2]

생성된 molecular fingerprint를 비교함을 통해 간단히 서로 다른 분자 사이의 유사도를 비교할 수 있다. 이때 fingerprint를 비교하는 방법 또한 여러 가지가 있다. 그 중 대표적으로 Tanimoto coefficient가 있다.[3] 이는 두 집합 사이의 유사도를 0에서 1 사이의 값으로 나타낸 것으로, 화합물 분석 외에도 이미지 분석과 같은 분야에도 활용될 수 있다. 본 연구에서는 두 화합물 사이의 유사도를 정량적으로 표현하기 위해, 두 화

[1] Weininger, D. (1988). "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules." Journal of chemical information and computer sciences **28**(1): 31-36.
[2] Cereto-Massagué, A., et al. (2015). "Molecular fingerprint similarity search in virtual screening." Methods **71**: 58-63.
[3] Bajusz, D., et al. (2015). "Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?" Journal of Cheminformatics **7**(1): 1-13.

합물의 molecular fingerprint를 구한 후, 두 fingerpirnt 사이의 Tanimoto coefficient를 구하여 유사도를 정량화했다.

1-3. 기계 학습 (Machine Learning, ML), 인공 신경망 (Artificial Neural Network)

기계 학습(Machine Learning, ML)은 컴퓨터 과학의 한 분야로, 데이터와 알고리즘을 사용하여 인간이 학습하는 방식을 모방하고 점차 정확도를 높이는 데 중점을 두는 분야로, 주어진 입력에 대해 원하는 출력을 내는 모델을 만드는 것을 목표로 한다. 최근에는 생성형 인공 신경망이 성능 면에서 이전의 많은 접근 방식을 능가하고 있다.[4]

인공 신경망은 행렬 수학을 기반으로, 인간의 신경계를 단순한 logical system으로 흉내낸 기계학습 모델이다. 인공 신경망은 퍼셉트론을 기본 단위로 하여 연산을 진행한다. 이는 주어진 입력값들을 선형 변환하는 행렬 연산 후, 이에 비선형성을 추가하는 활성함수를 연산시킨 값을 결과로 출력하는 형태이다. 이런 인공 뉴런을 여러 개 연결하면 인공 신경망이 되고, 이를 여러 층으로 배열했을 때 은닉층(hidden layer)들을 많이 추가하면 Deep Neural Network가 된다. 주어진 상황에 대해 원하는 결과를 잘 출력하는 weight matrix W를 수학적 배경을 토대로 경사 하강법과 같은 최적화 연산을 통해 구하는 것이 이 과정의 목표라고 말할 수 있다.[5]

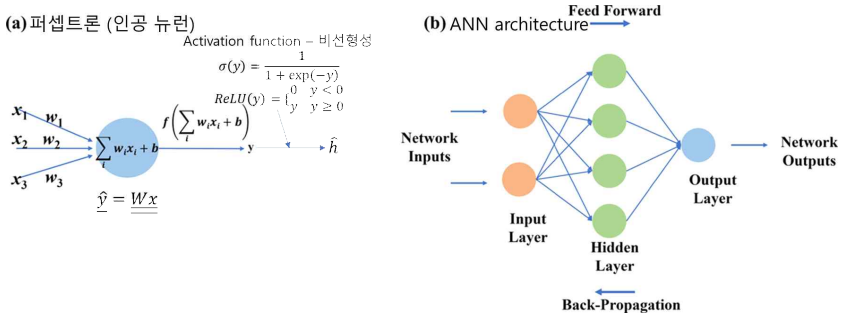

그림 1-2 퍼셉트론과 간단한 ANN architecture[5]

원하는 결과를 출력하기 위해 인공 뉴런을 배열하는 방법에 따라 다양한 신경망 모델들이 제시된다. 대표적으로 Recurrent Neural Network(RNN), Convolutional Neural Network(CNN)이 있다.

[4] IBM. What is machine learning? Retrieved 12-19, 2023, from https://www.ibm.com/topics/machine-learning.

[5] Xue, X., et al. (2023). "Advances in the Application of Artificial Intelligence-Based Spectral Data Interpretation: A Perspective." Analytical Chemistry **95**(37): 13733-13745.

Recurrent Neural Network(RNN)은 아래 그림과 같이 순차적 데이터에 대해 이전 단계의 출력값을 함께 입력값으로 받아 연산하도록 구조를 만든 것이다. 주로 순차적 데이터인 시간에 따른 측정값이나, 자연 언어와 같은 데이터를 처리하는데 주로 사용되는 모델이다.

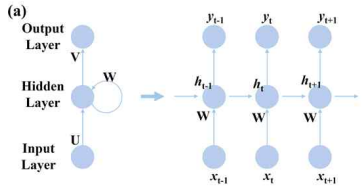

그림 1-3 Recurrent Neural Network의 간단한 모식도[5]

단, 단순 RNN은 파라미터 결정을 위한 기울기 연산 시 층이 깊어질수록 기울기(gradient)값이 0에 가까워져 의미가 없어지는 gradient vanishing 문제(Long-term dependency)가 있다. 이를 개선하기 위해 LSTM(Long Short-Term Memory)[6]이 제안되었다. 다만 LSTM은 복잡한 구조로 인해 파라미터가 많이 필요하게 되었고, 이로 인해 데이터가 충분하지 않은 경우 오버피팅이 발생하는 문제가 있다. 이를 개선하기 위해 GRU(Gated Recurrent Unit)[7]이 제안되었다. 본 연구에서는 순차적 데이터인 SMILES를 처리하기 위해 GRU를 이용했다.

Convolutional Neural Network(CNN)은 정형화된 입력 데이터에 대해 convolutional layer와의 dot product(합성곱(convolution) 연산)로 map을 만들어 계산하는 모델이다. 이미지 처리와 같은 grid화 된 데이터 처리에서 부분의 특징을 얻는 경우에 주로 사용된다.[8] 본 연구에서는 NMR 스펙트럼 정보 분석에 이를 이용했다.

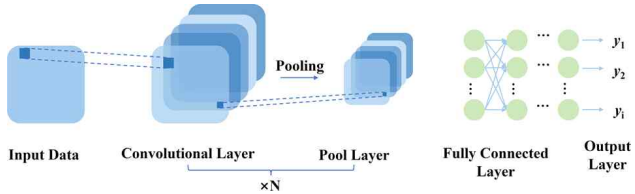

그림 1-4 Convolutional Nerual Network의 간단한 모식도[5]

[6] Sherstinsky, A. (2020). "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." Physica D: Nonlinear Phenomena **404**: 132306.

[7] Chung, J., et al. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555.

[8] Albawi, S., et al. (2017). Understanding of a convolutional neural network. 2017 international conference on engineering and technology (ICET), Ieee.

학습은 모델이 예측한 값과 참값 사이의 오차를 최소화 하는 방향으로 학습을 진행하는데, 둘 사이의 오차를 손실(Loss)이라는 목표함수로서 정의하고 이를 최소화하도록 수식적인 전개를 통해 이론적 토대를 세울 수 있다. 손실 함수로는 여러 종류가 있으나, 본 연구에서는 Mean Squared Error(MSE)와 Cross Entropy를 이용했다.

MSE는 참값과 예측된 값 사이의 차들의 제곱의 평균으로, 식으로 나타내면 아래와 같다. (i번째 참값 : y_i, 모델로 예측된 값 : $\hat{y_i}$)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

Cross entropy는 실제 분포와 모델로 예측된 분포 사이의 거리 정도에 해당하는 개념으로, 식으로 나타내면 아래와 같다.

$$H_p(q) = -\sum_{i=1}^{n} q(x_i)\log p(x_i)$$

1-4. NMR spectroscopy (Nuclear Magnetic Resonance)

핵 자기공명 분광법(Nuclear Magnetic Resonance(NMR) spectroscopy)는 유기화학자들이 사용하는 가장 중요한 분광학적 기술로, 구조에 관한 정보를 얻기 위해 이 방법에 우선적으로 의존한다.

NMR의 기본 원리는 스핀이 0이 아닌 원자핵(양성자와 중성자 개수가 모두 짝수인 것은 아닌 경우)에 자기장을 가하면 무작위로 배열되어 있던 핵 스핀이 외부 자기장 방향 혹은 이의 정 반대 방향으로 정렬되며 에너지 상태가 변화하는 것을 이용한다. 즉, 스핀이 0이 아닌 원자핵에 자기장을 가하게 되면 핵 스핀이 정렬되며 가해준 자기장의 크기에 비례하여 원자 핵의 에너지 준위가 둘로 나뉘게 된다. (이때 평행 배향이 역평형 배향보다 약간 더 에너지가 낮다) 만약 적절한 에너지의 전자기파가 배향된 핵에 조사되면 이를 흡수하며 spin flip이 일어난다. 이 현상이 곧 복사선과 핵과의 공명이기에 핵 자기공명 이라는 명칭이 붙게 되었다.

동일한 외부 자기장에 대해 한 분자 내에서도 핵 별 에너지 갈라짐의 크기가 다르다. 이는 핵이 전자에 둘러싸여 있기 때문이다. 분자에 외부 자기장이 가해질 때 전자들은 국부적 자기장을 형성하고, 이에 따라 핵에 의해 실질적으로 느껴지는 유효 자기장은 외부 자기장보다 약간 작아지게 된다. 이러한 전자의 가리움으로 인해 NMR spectroscopy에서는 서로 다른 흡광 주파수들을 확인할 수 있고, 이에 따라 분자 내의 화학적으로 서로 다른 핵을 확인할 수 있다. (화학적으로 다른 핵들의 주변의 전자 밀도와 성질은 다르다)

다만 앞서 언급했듯, 에너지 갈라짐의 크기는 외부 자기장의 크기에 비례한다. 다양한 자기장 세기에서 측정한 결과가 일정한 결과를 보일 수 있도록 화학적 이동을 델타 척도를 이용하여 나타낸다. 이 델타 척도의 단위는 ppm으로, 분광기의 작동 진동수

의 백만분의 일의 값에 해당한다. 또한 기준 물질을 설정하여 해당 물질과의 상대적인 흡광 진동수 차이를 기반으로 값을 계산하게 된다. 이때 기준 물질로는 가리움이 매우 큰 물질인 TMS(Tetramethylsilane, $(CH_3)_4Si$)를 이용한다. 이를 이용하여 델타 척도를 식으로 나타내면 아래와 같다.

$$\delta = \frac{\text{물질의 흡광 진동수}[Hz] - TMS\text{의 흡광 진동수}[Hz]}{MHz\text{로 나타낸 분광기의 진동수}}$$

NMR spectrum에서, 화학적 이동값으로부터 주위 환경 정보를, 적분 값으로부터 핵수를, 다중선(스핀-스핀 갈라짐)으로부터 이웃 원자핵에 관한 정보를 얻을 수 있다.[9]

1-5. NMR 분석을 위한 컴퓨터적 도구들에 관한 선행 연구

분석화학 분야에서 NMR 분석에 이용하기 위한 컴퓨터적 도구들은 다양한 방향으로 발전되어 왔다. 주어진 스펙트럼에 대해 피크가 어떤 것인지 할당하거나[10][11], DFT를 이용하여 분자 구조로부터 스펙트럼을 예측하거나[12], 스펙트럼으로부터 제안된 구조를 평가하거나[13], 사전 준비된 데이터 세트에서 주어진 스펙트럼에 해당하는 분자를 찾아주는[14][15] 등 다양하게 발전되어 있다.

그 중에서도 본 연구에서 관심 있는 바와 같이, 주어진 스펙트럼으로부터 구조를 생성해내는 연구 또한 다양하게 발전되어 왔다. 유전 알고리즘과 분자 계산을 이용하여 분자 구조를 제안하는 모델부터[16], 분자 생성 모델과 함께 DFT계산을 이용하여 구조를 제안하는 모델[17], CNN과 분자 그래프 생성 모델을 이용하여 구조를 제안하는 모

[9] McMurry, J. E. (2017). 맥머리의 유기화학, CENGAGE Learning, 사이플러스. p.429-447

[10] Smith, S. G. and J. M. Goodman (2010). "Assigning stereochemistry to single diastereoisomers by GIAO NMR calculation: The DP4 probability." Journal of the American Chemical Society **132**(37): 12946-12959.

[11] Zimmerman, D. E., et al. (1997). "Automated analysis of protein NMR assignments using methods from artificial intelligence." Journal of molecular biology **269**(4): 592-610.

[12] Gao, P., et al. (2020). "General protocol for the accurate prediction of molecular 13C/1H NMR chemical shifts via machine learning augmented DFT." Journal of Chemical Information and Modeling **60**(8): 3746-3754.

[13] Howarth, A. and J. M. Goodman (2022). "The DP5 probability, quantification and visualisation of structural uncertainty in single molecules." Chemical science **13**(12): 3507-3518.

[14] Zhang, C., et al. (2017). "Small molecule accurate recognition technology (SMART) to enhance natural products research." Scientific reports **7**(1): 14243.

[15] Bruguière, A., et al. (2020). "MixONat, a software for the dereplication of mixtures based on 13C NMR spectroscopy." Analytical Chemistry **92**(13): 8793-8801.

[16] Meiler, J. and M. Will (2002). "Genius: a genetic algorithm for automated structure elucidation from 13C NMR spectra." Journal of the American Chemical Society **124**(9): 1868-1870.

[17] Zhang, J., et al. (2020). "NMR-TS: de novo molecule identification from NMR spectra."

델[18], Transformer 모델을 이용하여 구조를 제안하는 모델[19], 이를 확장하여 BERT 모델을 이용하여 스펙트럼 정보와 함께 분자식이나 분자 조각과 같은 추가적인 정보를 함께 종합적으로 받아 구조를 제안하는 모델[20] 등 다양한 모델들이 제안되었다.

본 연구에서는 기계 학습을 사람이 생각하는 것과 유사한 방향으로 학습시켜보고자 했고, 그에 따라 본 연구에서는 스펙트럼에서 분자의 구조정보를 담은 잠재 벡터를 얻고, 이 벡터를 다시 구조로 복원하는 구조로 모델을 만들고자 했다. Litsa, Eleni, et al. 의 연구[21]에서 이와 같은 구조를 이용하여 MS/MS spectra로부터 SMILES를 얻는 모델(Spec2Mol)을 만들었다.

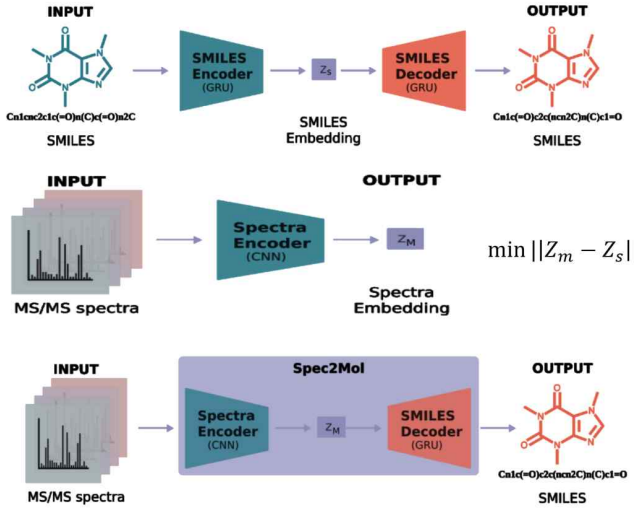

그림 1-5 Spec2Mol 모델의 구조[21]

위 모델에서 SMILES auto encoder와 Spectra encoder를 학습시킨 후, Spectra encoder와 SMILES decoder를 연결하여 최종 모델을 완성했다. 본 연구에서는 이와 유사한 형태로 모델을 구성했다.

2. 사용 데이터

사용한 데이터 세트는 크게 SMILES 데이터 세트와, NMR 데이터 세트 두 가지이다. 물질은 screening을 한 이후에 선택적으로 사용했다. 물질을 SMILES로 표현했을 때 기준으로, 길이가 120 이하이고, 탄소는 포함하고, 무기 원소 및 동위원소를 포함하지 않으며, 한 종류의 물질만을 포함하는 물질을 screening 했다. screening 이후에 SMILES들은 한 물질은 하나의 표현으로 나타날 수 있도록 canonical SMILES로 변환하여 이용했다.

학습 과정에서 SMILES들은 길이 상한인 120으로 길이를 맞추어 이용했다. 길이가 120이 되도록 문자열 뒤에 원소 기호와 겹치지 않는 문자 'X'를 넣어 길이를 120으로 padding하여 이용했다. 또한 SMILES 전체 데이터 세트에 대해 사용된 character들을 확인하고, 이를 dictionary로 대응시켜 one-hot-encoding하여 문자열을 정수 리스트로 변환하여 학습에 이용했다.

SMILES 길이 상한값 120은 이전 연구들과, SMILES dataset의 길이 분포를 통해 임의로 설정한 값이다. 길이 분포는 아래와 같고, 이에서 길이 120 이하의 문자열은 전체 중 93.96%임을 알 수 있다.

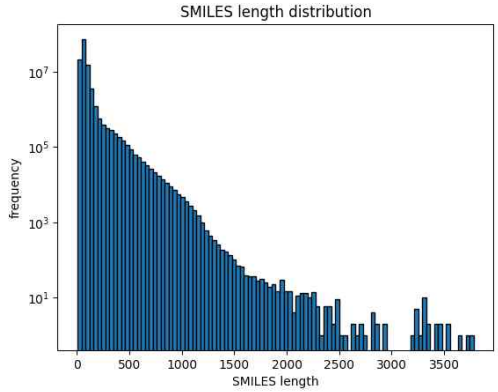

그림 2-1 PubChem의 CID-SMILES 파일의 SMILES 길이 분포. (개수는 로그 스케일로 그림)

유기 원소의 종류는 McMurry의 유기화학에서 자주 쓰이는 유기 원소 목록을 확인하여 C, H, O, N, F, Cl, Br, I, P, S로 설정하여 사용했다.[22]

Science and technology of advanced materials **21**(1): 552-561.

[18] Huang, Z., et al. (2021). "A framework for automated structure elucidation from routine NMR spectra." Chemical science **12**(46): 15329-15338.

[19] Alberts, M., et al. (2023). "Learning the Language of NMR: Structure Elucidation from NMR spectra using Transformer Models."

[20] Yao, L., et al. (2023). "Conditional molecular generation net enables automated structure elucidation based on 13C NMR spectra and prior knowledge." Analytical Chemistry **95**(12): 5393-5401.

[21] Litsa, E., et al. (2021). "Spec2Mol: An end-to-end deep learning framework for translating MS/MS Spectra to de-novo molecules."

[22] McMurry, J. E. (2017). 맥머리의 유기화학, CENGAGE Learning, 사이플러스. p. 25

2-1. SMILES

SMILES 데이터 세트는 PubChem의 CID-SMILES 파일을 기반으로 하여, 위와 같이 screening, canonical 변환, padding, one-hot-encoding 과정을 거쳐 이용했다. 이때 screening 이후 SMILES는 약 1억개 정도 있음을 확인할 수 있었고, 이들을 모두 학습에 이용하기에는 시간적, 성능적 제약이 따랐기에 이 중 100,000개를 선택하여 학습에 이용했다.

SMILES 데이터는 SMILES encoder, decoder에 train : valid = 9 : 1로 random split하여 사용했다.

2-2. NMR

NMR 데이터 세트는 크게 SMILES, 분자식, 스펙트럼 세 가지의 정보를 함께 담도록 했다. NMR 데이터 세트는 NP-MRD(the Natural Products Magnetic Resonance Database)[23]의 IUPAC 표준 스펙트럼 파일 형식인 JCAMP-DX[24] 파일(확장자 : .jdx, 텍스트 형태)을 이용했다. 해당 파일은 물질 종류와 이의 스펙트럼 정보가 함께 저장된 형태로, 물질 종류로부터 이의 SMILES를 얻을 수 있고, 위와 동일한 SMILES screening 과정을 거쳐 스펙트럼 데이터 또한 screening했다.

파일의 spectra raw data는 부호화 되어 있었고, 이를 해독할 알고리즘을 이해하지 못하여 부호화 되지 않은 peak list data(peak의 ppm 값과 해당 지점의 height 정보 포함)를 이용했다. 이 중 위의 screening과정과 함께, 1H-NMR, 유효 데이터, 중복 정리 과정을 추가로 거치고 남은 9,990개의 데이터 세트를 이용했다.

코드 학습 과정에서 입력 spectra는 1H-NMR의 ppm 0.01~13.00 데이터를 이용하여, 0.01 ppm마다의 스펙트럼 높이를 토대로 길이 1300의 벡터로 만들어 학습에 이용했다.

이와 함께 NMR dataset에는 SMILES와 분자식 정보도 함께 포함 시켰다. 이때 SMILES는 위와 동일한 과정으로 처리하여 길이 120으로 padding 및 one-hot-encoding된 형태로 이용했고, 분자식은 해당 분자에 들어있는 C, H, O, N, F, Cl, Br, I, P, S 원소 개수를 해당 순서로 나열하여 길이 10의 벡터로 만들어 학습에 이용했다.

NMR 데이터는 NMR encoder에 train : valid : test = 8 : 1 : 1로 나누어 사용했다.

3. 모델

본 연구에서는 사람이 NMR 스펙트럼을 분석할 때의 사고 과정을 모델의 아키텍처

[23] Wishart, D. S., et al. (2022). "NP-MRD: the natural products magnetic resonance database." Nucleic acids research **50**(D1): D665-D677.

[24] Lampen, P., et al. (1999). "An extension to the JCAMP-DX standard file format, JCAMP-DX V. 5.01." Pure and Applied Chemistry **71**(8): 1549-1556.

에 담아내고자 했다. 또한 모델 전체의 입력으로는 1H-NMR 스펙트럼 정보와 분자식 정보를, 출력으로는 이에 해당하는 분자의 SMILES를 출력하는 것을 목표로 했다.

3-1. 모델 구조

사람이 NMR 스펙트럼을 분석할 때의 사고 과정을 크게 두 단계로 나누어 보면 아래와 같이 나누어 볼 수 있다.

1. 스펙트럼으로부터 유효한 정보들의 모음을 얻는다 : peak의 위치, 적분값, 갈라짐 등의 자료들로부터 정답 분자의 작용기, 인접 구조 등에 대한 정보를 얻는다.
2. 스펙트럼으로부터 얻은 정보들을 토대로 합리적인 분자 구조를 추측한다.

이러한 의사 과정을 통해 정답 분자를 얻어내는 과정을 나타내기 위해 모델은 크게 세 가지 구조로 제안한다 : 1. SMILES를 latent vector로 압축하는 SMILES encoder, 2. latent vector를 SMILES로 복원하는 SMILES decoder, 3. 1H-NMR 스펙트럼 정보와 분자식 정보로부터 1과 동일한 latent vector를 출력하는 NMR encoder.

SMILES encoder와 decoder는 연결하여 SMILES auto encoder를 이룰 수 있고, 이로부터 latent vector와 SMILES decoder를 얻을 수 있으며, 얻은 latent vector로 NMR encoder를 학습시켜 결과적으로 NMR encoder와 SMILES decoder를 연결하여 NMR 스펙트럼으로부터 정답 분자를 추측하는 모델 구조를 완성할 수 있다.

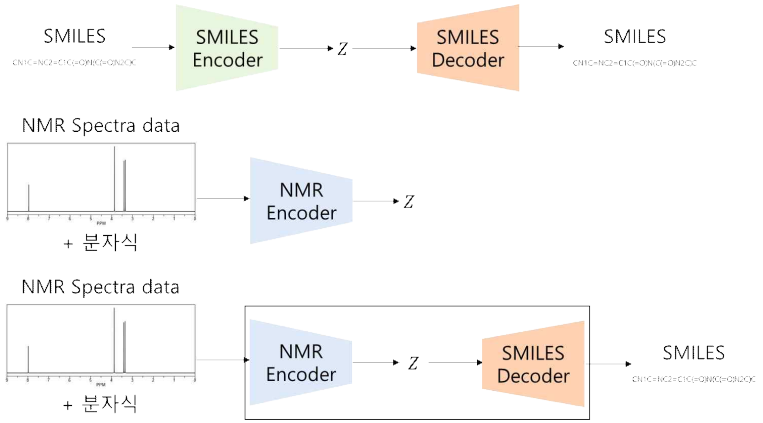

그림 3-1 본 연구에서 사용한 모델의 구조

이를 통해 사람의 의사 결정 과정과 유사하게《NMR 데이터로부터 적절한 잠재 정보 획득 → 획득한 잠재 정보로 분자 구조 결정》의 구조를 구상할 수 있다.

1. SMILES encoder, decoder

SMILES는 순차적 정보로 이루어져 있다. 이를 분석하는 모델을 만들기 위해 본 연구에서는 RNN을 개선한 GRU를 이용해서 이들을 구현했다. 이와 함께 embedding,

dropout, linear layer를 이용하여 아래 그림과 같이 SMILES encoder, decoder를 구성했다.

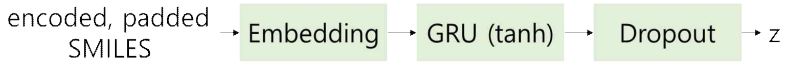

그림 3-2 SMILES encoder 구조

그림 3-3 SMILES decoder 구조

SMILES의 각 단어는 길이 256의 벡터로 embedding(각 단어를 특정 길이의 벡터로서 표현함) 했고, GRU의 activation function으로는 tanh 함수를 이용했다. 개선된 학습을 위해 dropout layer(확률적으로 특정 신경을 제거하고 학습)를 추가했으며, dropout probability는 0.2로 설정했다.

Batch 크기를 B, SMILES의 길이를 L(본 연구에서는 120으로 고정), embeding size를 F로 두면(본 연구에서는 256으로 고정), SMILES encoder의 입력 텐서 크기는 [B, L], 출력 텐서 크기는 [B, N]이 된다. 이때 출력 처리에 따라 latent vector의 크기(N)를 다르게 하여 조건을 변화시키며 실험을 진행할 수 있다. SMILES decoder의 입력 텐서 크기는 [B, N], 출력 텐서 크기는 [B, L, N_char] 이다. (N_char : SMILES에 포함된 모든 character 개수. 본 연구에서 N_char = 40)

2. NMR encoder

NMR encoder는 CNN과 DNN(Linear layer + ReLU activation function)를 이용하여 구성했다. 특히 CNN이 NMR 스펙트럼의 스핀 갈라짐을 사람의 사고 과정과 유사하게 인식할 수 있도록 일반적으로 구분되는 갈라짐 종류 개수와 유사하게 kernel 개수를 구성했다. 이 경우의 z는 SMILES encoder, decoder의 경우와 크기가 동일하다.

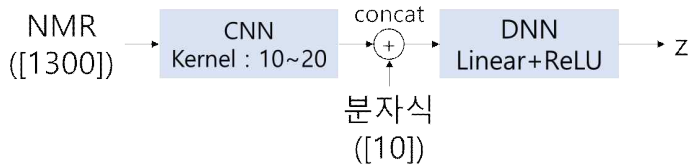

그림 3-4 NMR encoder 구조

3-2. 학습 과정

학습 과정은 아래와 같이 총 세 단계로 나누어 train 1, 2, 혹은 train 0, 1, 2의 순서로 진행했다.

1. SMILES autoencoder 학습

Train 1 과정은 SMILES 데이터 세트로 SMILES autoencoder를 학습하는 단계로, SMILES encoder와 decoder를 연결하여 입력 SMILES 데이터로부터 예측된 각 자리별 각 문자의 확률을 얻고, 이와 원래 SMILES 데이터 사이의 Cross Entropy를 최소화하는 방향으로 학습을 진행했다.

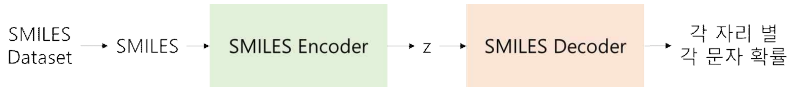

그림 3-5 Train 1 : SMILES encoder, decoder 학습

2. NMR encoder 학습

Train 2 과정은 NMR 데이터 세트로 NMR encoder를 학습하는 단계로, 하나의 분자에 대해, 이의 SMILES는 train 1에서 학습된 SMILES encoder를 통과시켜 latent vector를 얻고, NMR spectra 정보와 분자식 정보는 NMR encoder를 통과시켜 이의 latent vector를 얻은 후, 두 latent vector들의 Mean square error가 최소화되도록 NMR encoder의 학습을 진행했다. (학습된 SMILES encoder는 학습이 진행되지 않는다)

그림 3-6 Train 2 : NMR encoder 학습

0. SMILES encoder, decoder, NMR encoder 동시 학습

위 두 과정과 별도로, 전체 데이터 학습 효과와, latent vector에 NMR 데이터의 정보를 포함시키는 효과를 위하여[25], 세 가지 모델을 동시에 학습하는 Train 0 과정을 추가했다. 이 과정을 수행하는 경우는 train 0, 1, 2 순으로 학습이 진행된다. 이에서는 SMILES encoder - SMILES decoder, NMR encoder - SMILES decoder 모델 각각에 대한 Cross entropy 합을 총 loss로 활용하여, 세 모델 전체의 파라미터에 대한 학습을 진행했다.

[25] Gómez-Bombarelli, R., et al. (2018). "Automatic chemical design using a data-driven continuous representation of molecules." ACS central science **4**(2): 268-276.

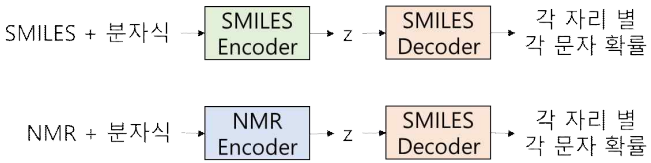

그림 3-7 Train 0 : SMILES encoder, decoder, NMR encoder 동시 학습

3-3. 테스트 과정

위의 학습 과정을 통해 훈련된 NMR encoder와 SMILES decoder를 연결하여 최종적으로 NMR 스펙트럼 정보로부터 SMILES를 생성하는 모델을 만들었다. 이를 통해 final test loss는 정답 SMILES와 생성된 문자열의 각 자리 별 각 문자 확률 사이의 Cross entropy로 확인했다.

그림 3-8 최종 NMR → SMILES 모델

4. 실험

4-1. 실험 환경

본 연구에서는 Windows 11이 설치된 PC에서 실험을 진행했다.

세부 성능 : CPU : i5-13400, RAM : DDR5 32GB 4800MHz, GPU : NVIDIA GeForce RTX 3060 Ti with GDDR6 8GB

이와 Google Colab의 T4 GPU도 함께 사용하였다.

4-2. 실험 결과

1. latent vector 변화

첫 시도에서는 latent vector로 SMILES encoder의 GRU output 전체인 [B, L*F]크기의 텐서를 이용했다. (NMR encoder의 CNN kernel 15개, DNN layer 수 4개, DNN hidden dimension 10000) train 1, 2를 이용하여 학습을 진행했다. epoch에 따른 loss변화 그래프는 아래와 같다. (train 1의 최소 validation loss : $4.267*10^{-6}$, train 2의 최소 validation loss : 0.2285)

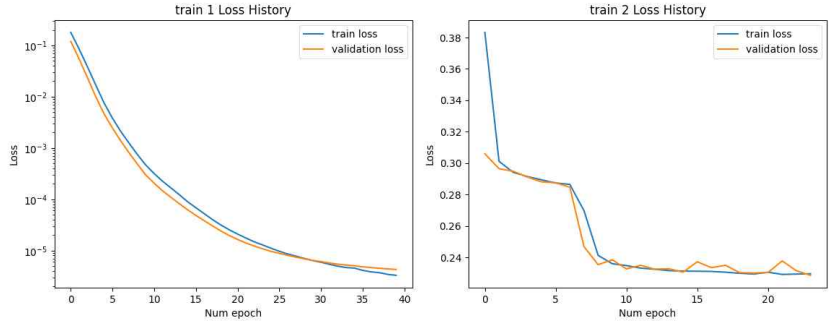

그림 4-1 [B, L*F] 크기의 latent vector로 train 1, 2 과정을 거쳐 학습한 경우의 loss history

이 경우 각 train의 loss는 작게 나오는 것을 확인할 수 있었으나, 파라미터 수가 굉장히 많아 학습에 과도한 시간이 소요되는 문제가 있었다. 이로 인해 목표한 학습을 완료하지 못하고 학습을 도중에 중단했다.

이러한 문제 해결을 위해 SMILES encoder와 decoder의 처음과 마지막에 L*F to F, F to L*F 차원의 linear layer를 추가하여 latent vector의 크기를 [B, F]로 수정하고, NMR encoder의 CNN kernel 수를 10개로 줄이고, DNN layer 수를 3개로 줄여 train 1, 2과정으로 학습을 진행했다. epoch에 따른 loss 변화 그래프는 아래와 같다. (train 1의 최소 validation loss : 0.004206, train 2의 최소 validation loss : 20.49)

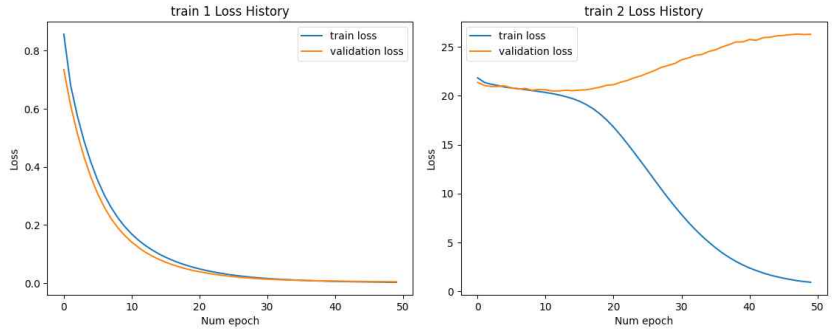

그림 4-2 [B, F] 크기의 latent vector로 train 1, 2 과정을 거쳐 학습한 경우의 loss history

이 경우에는 train 2에서 train loss는 감소하나 validation loss는 오히려 증가하는, over fitting 문제가 발생했음을 확인할 수 있었다.

over fitting 문제 개선을 위해 latent vector의 크기를 [B, 2F]로 수정하여 다시 실험을 진행해 보았다. (train 1의 최소 validation loss : 0.001193, train 2의 최소 validation loss : 16.04)

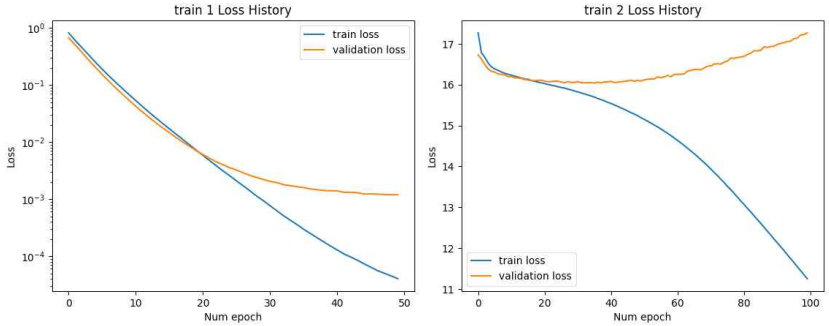

그림 4-3 [B, 2F] 크기의 latent vector로 train 1, 2 과정을 거쳐 학습한 경우의 loss history

Latent vector의 크기를 증가시킴에 따라 두 train 모두에서 최소 validation loss 값은 감소함을 확인할 수 있었으나, train 2에서 여전히 over fitting 문제가 발생함을 확인할 수 있었다. 이를 제외하면, latent vector의 크기가 증가함에 따라 최종 test loss도 감소하고, 생성된 SMILES 또한 C 이외에도 다양하게 나타남을 확인 가능했다. 다만 연산을 도중에 중단한 [B, L*F]크기의 latent vector를 이용한 경우를 제외한 두 경우에 대해서는 유효한 SMILES는 나타나지 않았다.

2. train 0, 1, 2 이용

다양한 실험 조건에서 train 0, 1, 2를 이용한 경우와 train 1, 2 과정을 이용한 경우의 최종 test loss를 비교했을 때 뚜렷한 차이를 보이는 경우는 없었음을 확인할 수 있었다. 그러나 단순히 train 0만 수행한 경우와, train 0, 1, 2 모두 수행한 경우의 최종 test loss에는 차이가 존재함을 확인할 수 있었다. 후자가 더 final test loss가 작음을 확인할 수 있었다.

3. hyper parameter 변화

학습을 개선하기 위해 latent vector의 크기는 [B, F]로 고정하고 다른 여러 요인들(hyper parameter)을 변화시키며 학습을 진행해 보았다. learning rate 변화, learning rate scheduler 및 annealing 이용, hidden dimension 및 layer 변화, weight decay 이용 등 다양한 요인들을 변화시키며 학습을 진행해 보았으나, 뚜렷한 변화는 확인할 수 없었다. 실험들 중 final test loss가 가장 작았던 경우의 loss history는 아래와 같다.
(train 1, 2 : train 1 best validation loss 0.1416, train 2 best validation loss 0.0710, final test loss 1.273, valid SMILES 0)
(train 0, 1, 2 : train 0 best validation loss 1.4104, train 1 best validation loss 0.1411, train 2 best validation loss 0.0682, final test loss 1.282, valid SMILES 2)

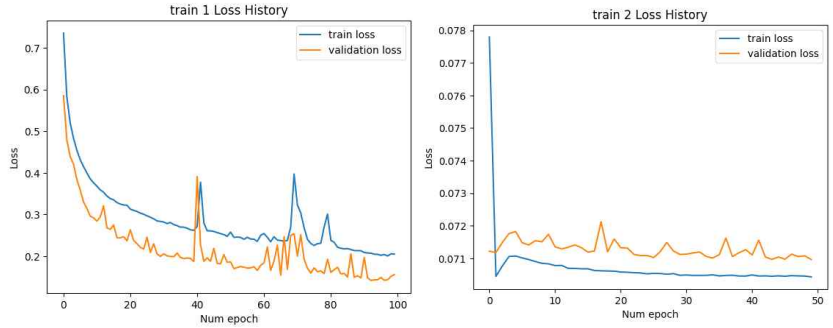

그림 4-4 final test loss가 가장 작았던 경우의 test1, 2를 이용한 학습 과정에서의 loss history

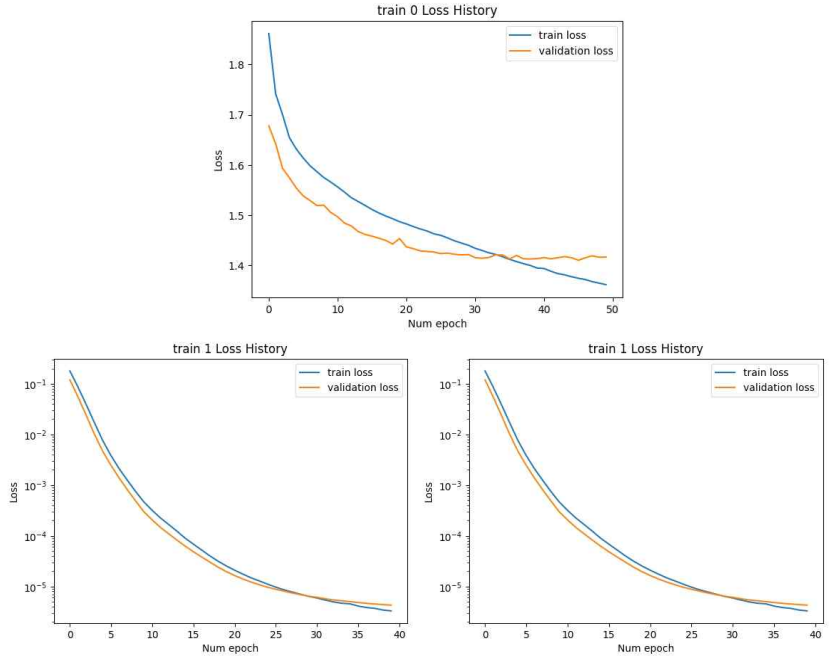

그림 4-5 final test loss가 가장 작았던 경우의 test0, 1, 2를 이용한 학습 과정에서의 loss history

4. Valid SMILES

실험 전반에 있어서 SMILES 문법에 맞는 유효한 SMILES는 거의 나타나지 않았

다. 생성된 유효한 SMILES는 문법적으로 틀릴 가능성이 적은 CCC...C 형태 뿐이었다.

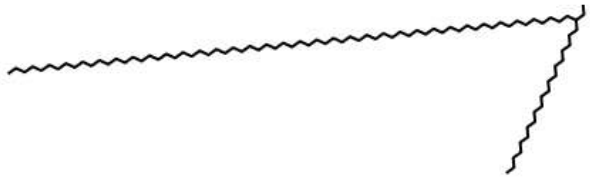

그림 4-6 생성된 valid SMILES : CCC...C 꼴의 분자 예시

이렇게 유효한 SMILES들에 대해 정답 분자와의 유사도를 확인하기 위해 tanimoto coefficient를 이용해보면 평균적으로 0.1에서 0.2 사이의 값을 나타냄을 확인할 수 있었다. 즉, 이와 같이 생성된 유효한 SMILES는 실제 정답 분자와 유사하지 않음을 확인할 수 있었다.

5. 논의

전반적으로 SMILES Dataset의 Auto encoder 모델은 적절한 방향으로 잘 학습됨을 loss 그래프를 통해 확인할 수 있으나, NMR encoder의 학습은 잘 되지 않았음을 확인할 수 있다. 이는 기계가 NMR spectra로부터 SMILES latent vector로 가는 과정을 잘 이해하지 못했다는 의미로, 모델에 수정이 필요할 것으로 예상된다. 혹은 본 실험 과정에서 스펙트럼 전체 데이터가 아닌 peak table 정보만을 이용하여 intensity와 같은 정보가 정확히 인식되지 않았을 가능성 또한 존재한다.

또한 SMILES Auto Encoder 모델의 경우도 두 경우 모두 최종 loss가 약 0.14 정도로 여전히 오차가 존재한다. 특히 이로 인해 모델이 SMILES 문법을 잘 학습하지 못하여 최종 생성 SMILES의 대부분이 문법에 맞지 않는 형태가 나왔을 것으로 추측된다. 이는 parameter수를 늘리거나 더 많은 dataset을 이용하여 학습을 하는 방법으로 개선할 수 있을 것으로 기대된다.

다만 1, 2 과정을 통한 학습에서 train 1의 경우 learning rate는 일정하게 설정했음에도 Loss가 중간에 급격히 변화하는 구간이 몇 개 있었다. 이에 대한 정확한 이유는 제시하지 못했다.

유효한 SMILES로 C만 있는 것이 남은 이유로는, decoder가 문법을 정확히 학습하지 못하여 가장 틀릴 확률이 적은 문자열만 남아 결론적으로 이런 경우만 남은 것으로 추측된다. 또한 유효하지 않은 전반적인 결과에서도 C가 많은 형태로 정답이 제시되었음을 알 수 있었는데, 이는 loss로 단순한 cross entropy와 MSE를 사용하여 단순히 오답일 확률이 적은 방향인, C가 많은 방향으로 모델이 학습된 결과로 추정된다. 더 많은 데이터 세트를 통해 개선할 수 있을 것으로 예상된다. 또한 선행 연구의 사례와 같이 Wasserstein distance나 Tanimoto coefficient를 직접적으로 loss에 이용하는 방법도 개선에 도움을 줄 수 있을 것이다.

혹은 최대 확률을 가지는 문자 조합만을 제시하는 것이 아닌, 제시된 자리 별 문자 확률들에서 문법에 맞으면서 확률이 가장 높은 문자열을 찾는 알고리즘을 제시하여 이용한다면 이러한 문제를 어느 정도 해결 가능하며, output 또한 확률별로 다양하게 제시할 수 있을 것으로 생각된다.

6. 결론

SMILES Auto Encoder와 NMR Encoder 모델을 이용하여 NMR spectra to SMILES 모델을 설계하고 학습해보았으나, 결과는 기대한 만큼 잘 나오지 않았다. 모델 수정과 더 많은 데이터 세트 이용을 이용하여 결과를 개선할 수 있을 것으로 기대된다.

7. 사용한 코드

기계학습은 python 언어로 pytorch를 기반으로 진행되었다. 학습에 이용한 코드는 ipython notebook 파일(.ipynb)의 형태로 정리하여 github에 업로드 해 두었다. 해당 파일은 본 연구에 관한 참고용으로, 지속적인 업데이트에 관한 특별한 계획은 없다.

https://github.com/JeongheonSeok/2024_02_undergraduate_thesis

참 고 문 헌

[1] Weininger, D. (1988). "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules." Journal of chemical information and computer sciences **28**(1): 31-36.

[2] Cereto-Massagué, A., et al. (2015). "Molecular fingerprint similarity search in virtual screening." Methods **71**: 58-63.

[3] Bajusz, D., et al. (2015). "Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?" Journal of Cheminformatics **7**(1): 1-13.

[4] IBM. What is machine learning? Retrieved 12-19, 2023, from https://www.ibm.com/topics/machine-learning.

[5] Xue, X., et al. (2023). "Advances in the Application of Artificial Intelligence-Based Spectral Data Interpretation: A Perspective." Analytical Chemistry **95**(37): 13733-13745.

[6] Sherstinsky, A. (2020). "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." Physica D: Nonlinear Phenomena **404**: 132306.

[7] Chung, J., et al. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555.

[8] Albawi, S., et al. (2017). Understanding of a convolutional neural network. 2017 international conference on engineering and technology (ICET), Ieee.

[9] McMurry, J. E. (2017). 맥머리의 유기화학, CENGAGE Learning, 사이플러스. p. 429-447

[10] Smith, S. G. and J. M. Goodman (2010). "Assigning stereochemistry to single diastereoisomers by GIAO NMR calculation: The DP4 probability." Journal of the American Chemical Society **132**(37): 12946-12959.

[11] Zimmerman, D. E., et al. (1997). "Automated analysis of protein NMR assignments using methods from artificial intelligence." Journal of molecular biology **269**(4): 592-610.

[12] Gao, P., et al. (2020). "General protocol for the accurate prediction of molecular 13C/1H NMR chemical shifts via machine learning augmented DFT." Journal of Chemical Information and Modeling **60**(8): 3746-3754.

[13] Howarth, A. and J. M. Goodman (2022). "The DP5 probability, quantification and visualisation of structural uncertainty in single molecules." Chemical science **13**(12): 3507-3518.

[14] Zhang, C., et al. (2017). "Small molecule accurate recognition technology (SMART) to enhance natural products research." Scientific reports **7**(1): 14243.

[15] Bruguière, A., et al. (2020). "MixONat, a software for the dereplication of mixtures based on 13C NMR spectroscopy." Analytical Chemistry **92**(13): 8793-8801.

[16] Meiler, J. and M. Will (2002). "Genius: a genetic algorithm for automated structure elucidation from 13C NMR spectra." Journal of the American Chemical Society **124**(9): 1868-1870.

[17] Zhang, J., et al. (2020). "NMR-TS: de novo molecule identification from NMR spectra." Science and technology of advanced materials **21**(1): 552-561.

[18] Huang, Z., et al. (2021). "A framework for automated structure elucidation from routine NMR spectra." Chemical science **12**(46): 15329-15338.

[19] Alberts, M., et al. (2023). "Learning the Language of NMR: Structure Elucidation from NMR spectra using Transformer Models."

[20] Yao, L., et al. (2023). "Conditional molecular generation net enables automated structure elucidation based on 13C NMR spectra and prior knowledge." Analytical Chemistry **95**(12): 5393-5401.

[21] Litsa, E., et al. (2021). "Spec2Mol: An end-to-end deep learning framework for translating MS/MS Spectra to de-novo molecules."

[22] McMurry, J. E. (2017). 맥머리의 유기화학, CENGAGE Learning, 사이플러스. p. 25

[23] Wishart, D. S., et al. (2022). "NP-MRD: the natural products magnetic resonance database." Nucleic acids research **50**(D1): D665-D677.

[24] Lampen, P., et al. (1999). "An extension to the JCAMP-DX standard file format, JCAMP-DX V. 5.01." Pure and Applied Chemistry **71**(8): 1549-1556.

[25] Gómez-Bombarelli, R., et al. (2018). "Automatic chemical design using a data-driven continuous representation of molecules." ACS central science **4**(2): 268-276.

Abstract

In this study, I conducted research with the goal of developing a maching learning model that receives NMR spectra and molecular formulas as input and predicts the corresponding molecular structure. The model was largely devided into SMILES auto encoder and NMR encoder for learning, and the final model was created by connecting the learned NMR encoder and SMILES decoder. As a result of testing after model training, I was able to confirm that most of generated SMILES were grammatically invalid, and those that were valid were mainly composed of C. It is expected that the results can be improved by adding data and parameters, and modifying the model.

keywords : Generative model, NMR, SMILES, Machine learning