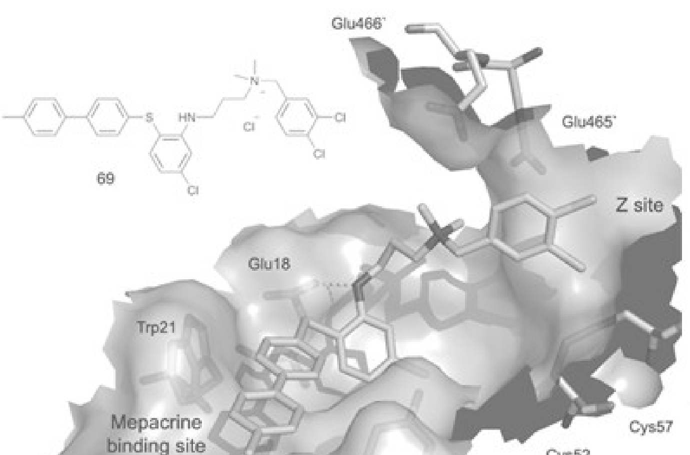
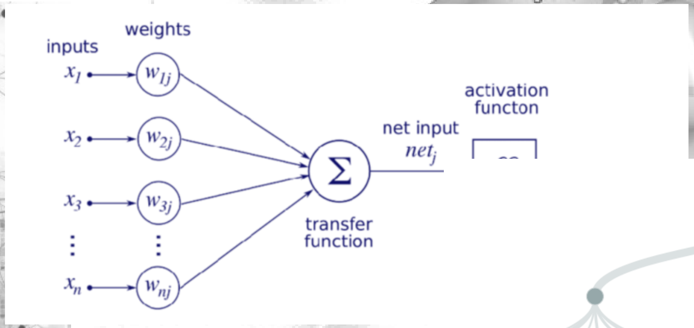
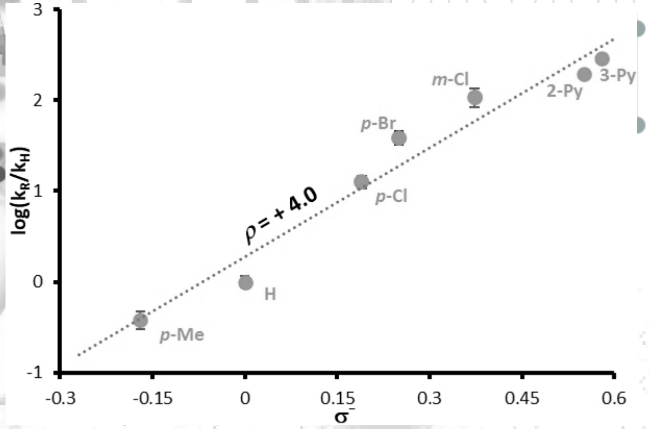
Introduzione ai metodi di Intelligenza Artificiale…







(Q)SAR: we are ready to this…

Real world

Chemical Compound (CC)

↓

Chemical Structure (CS)

↓

Chemical Properties (CP)

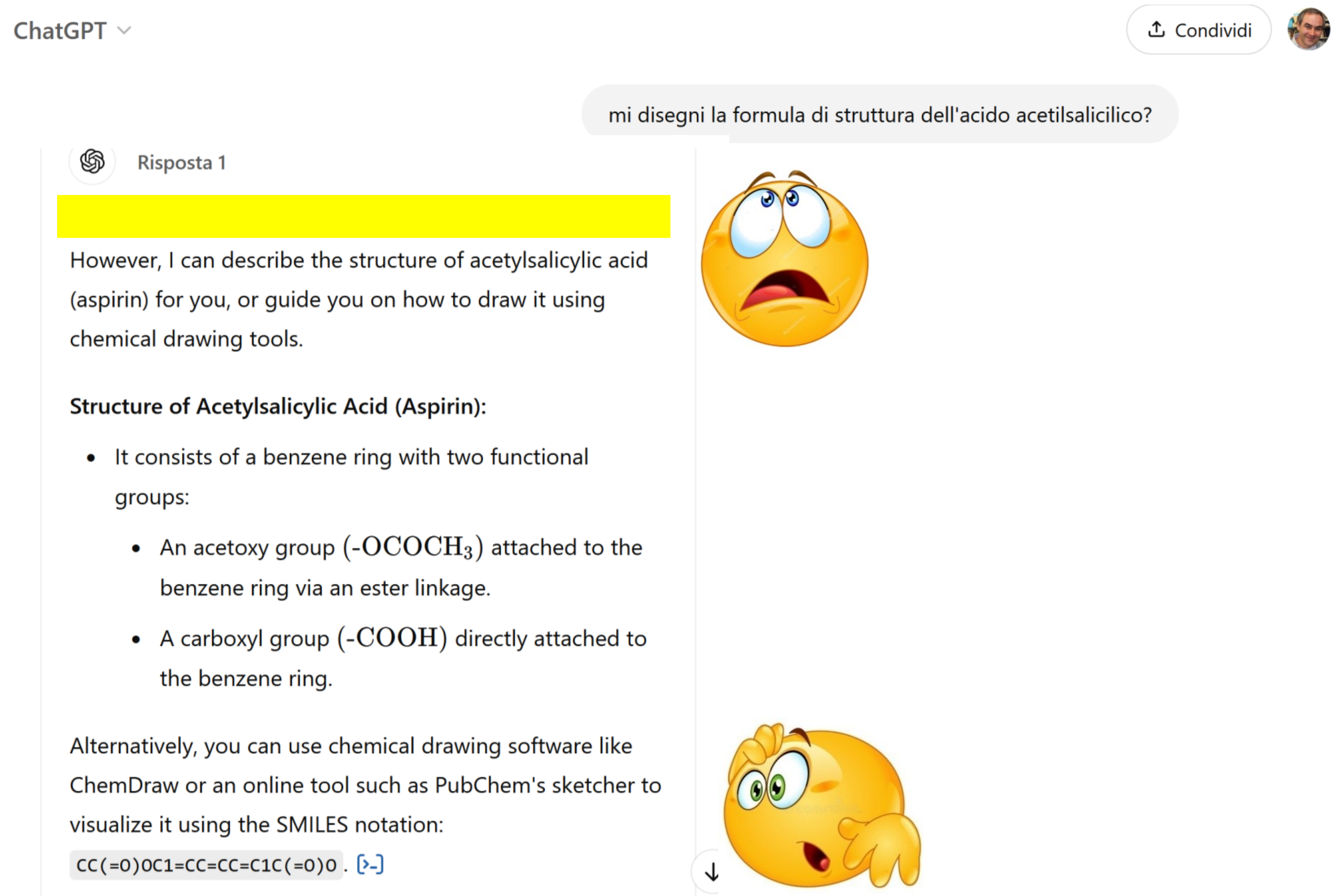
Virtual world

Numerical representazions of CS

↓

Molecular Descriptors (MD)

Stefano&ChatGPT… on the table:



mi disegni la formula di struttura dell'acido acetilsalicilico?

Risposta 1

[redacted]

However, I can describe the structure of acetylsalicylic acid (aspirin) for you, or guide you on how to draw it using chemical drawing tools.

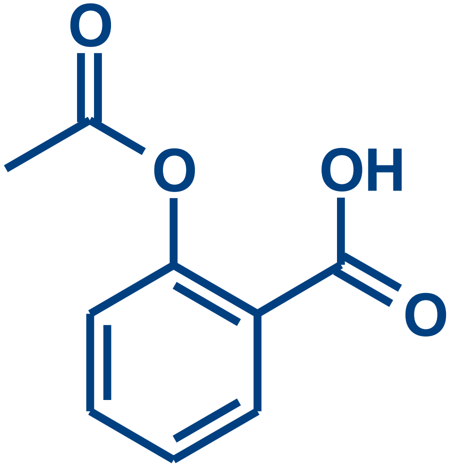
Structure of Acetylsalicylic Acid (Aspirin):

- It consists of a benzene ring with two functional groups:
    - An acetoxy group (-OCOCH_3) attached to the benzene ring via an ester linkage.
    - A carboxyl group (-COOH) directly attached to the benzene ring.

Alternatively, you can use chemical drawing software like ChemDraw or an online tool such as PubChem's sketcher to visualize it using the SMILES notation: `CC(=O)OC1=CC=CC=C1C(=O)O` .

With how many chemical representations we can deal?

C9H8O4



...the acetylsalicylic acid?

The crucial informatics differences:

acetylsalicylic acid

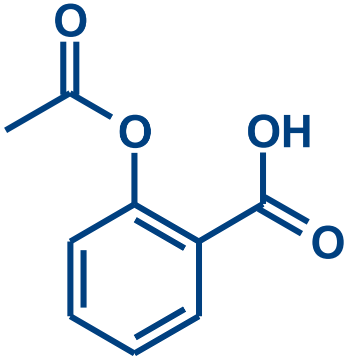
$C_9H_8O_4$

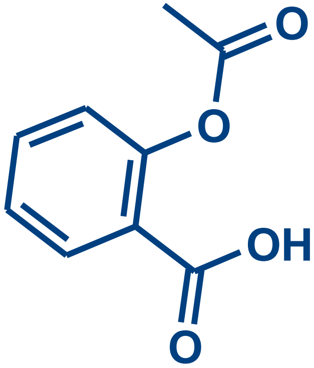
… these are simple *strings* (sequences) of alphanumeric characters and they are very easy to manage… informatically speaking!!!

Just a simple example: are these two representations identical?

$C9H8O4$

$C9H8O4$

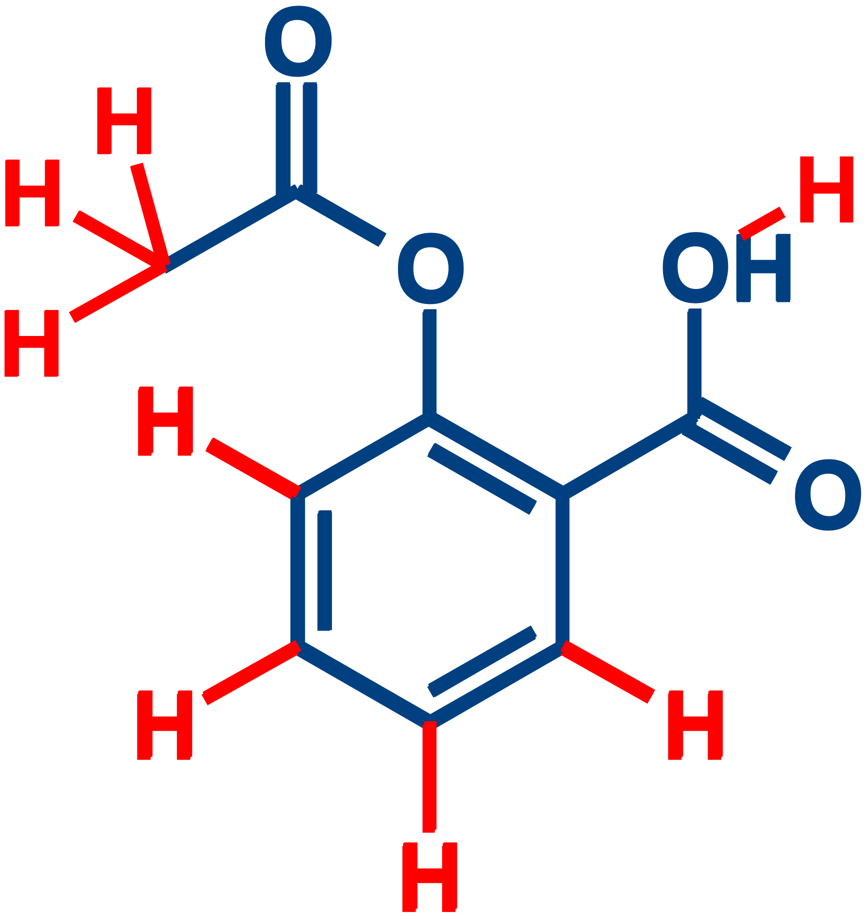




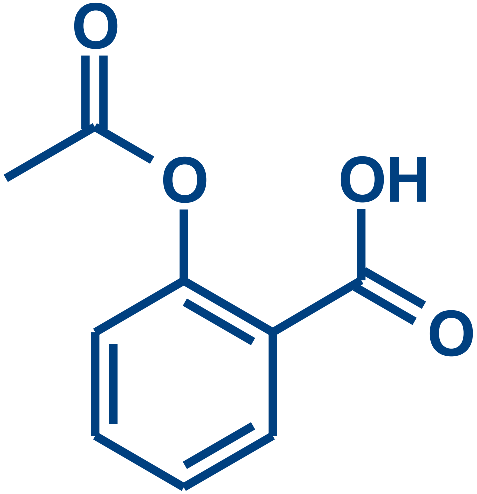
Time of answer (sec):

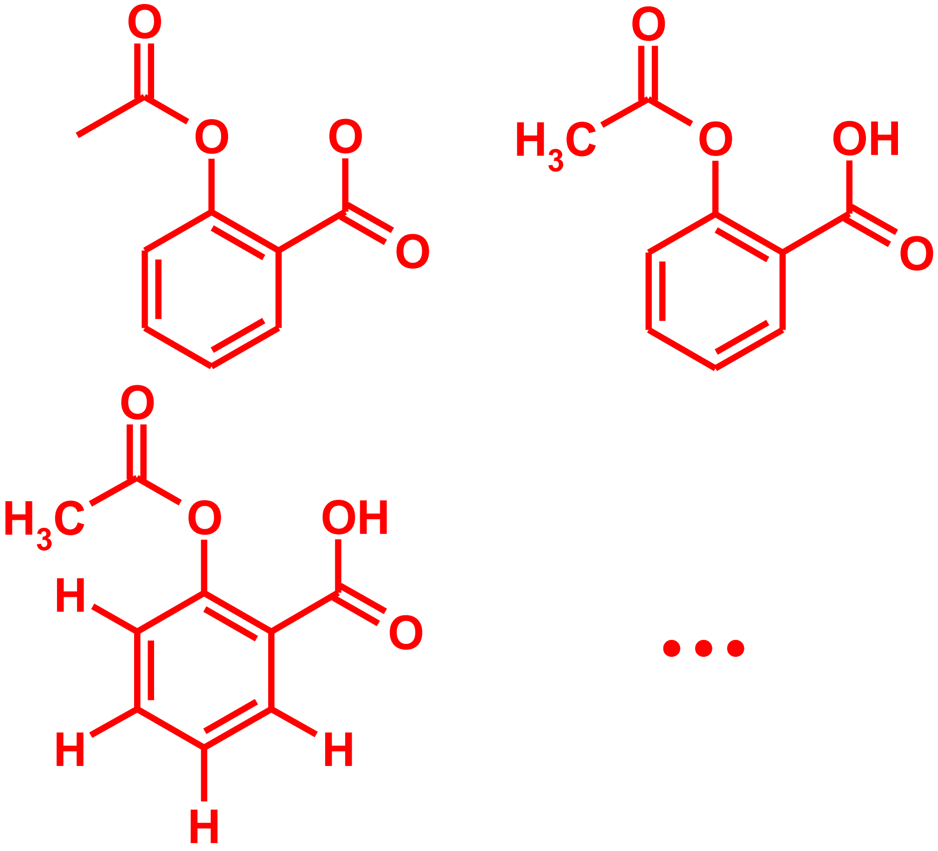
Time of answer (sec):

Be careful to the chemical *slang*…



Remember, all of these are not identical… informatically specking!

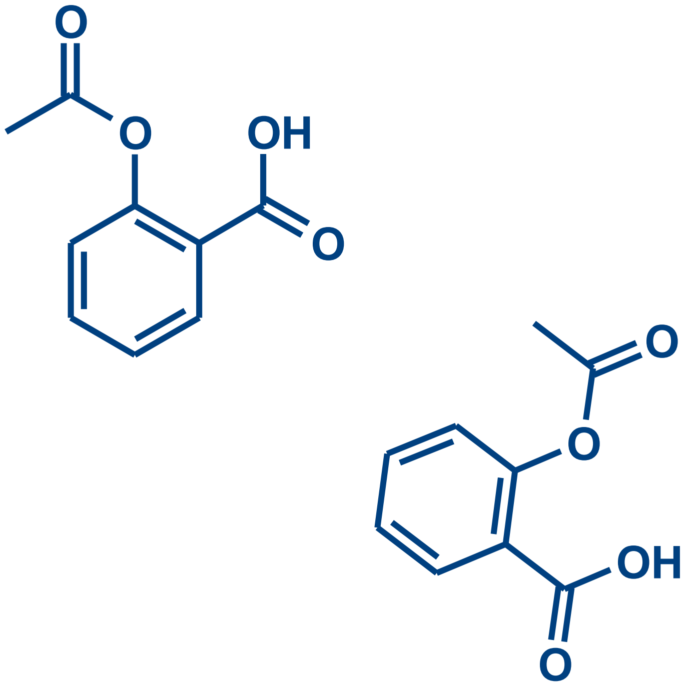




Just a simple example: are these two representations identical?

C9H8O4

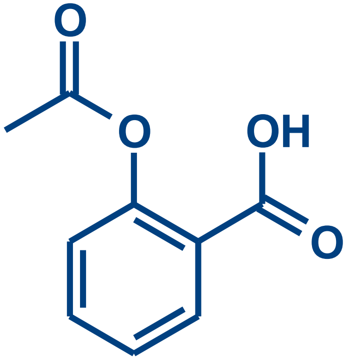
C9H8O4

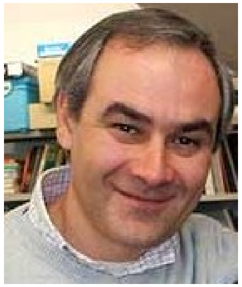


Time of answer (sec):

Time of answer (sec):

Think about...





C9H8O4

Stefano Moro

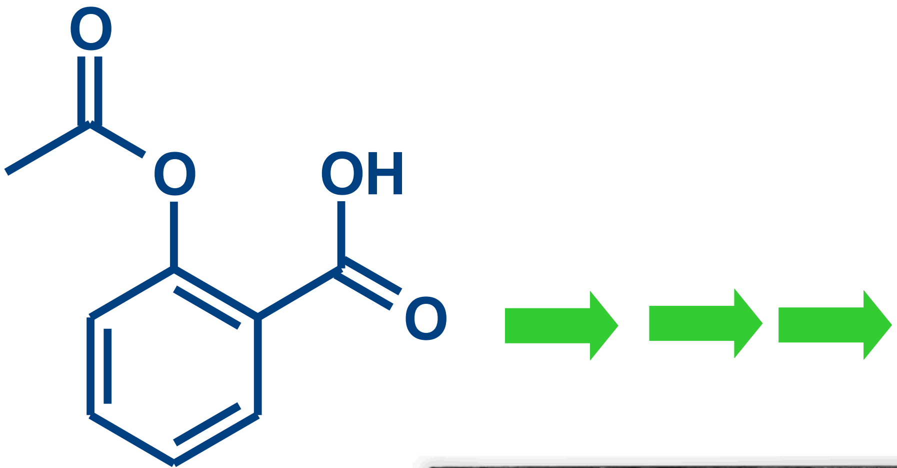
⬇ ⬇

????

MROSFN65B05X407Y

Confidential and Property of ©2012 Molecular Modeling Section

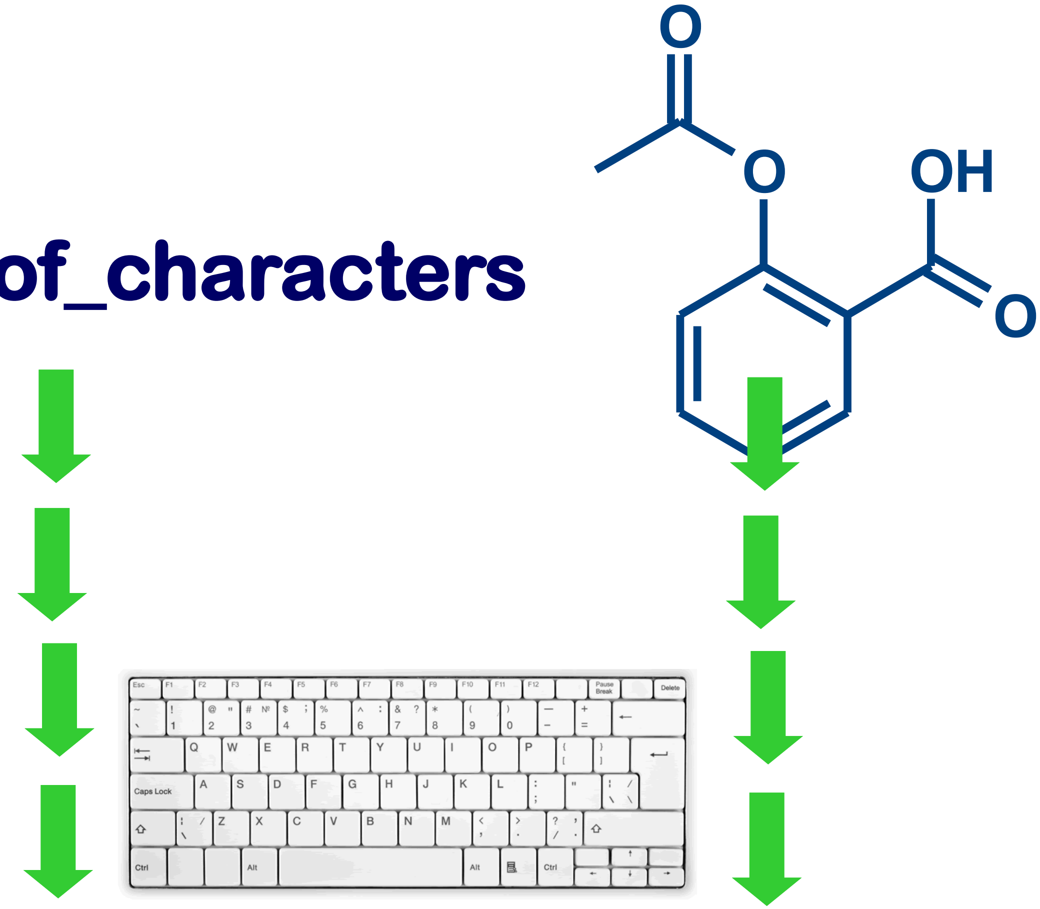
Combining business with pleasure ?



String_of_characters

… possibly, using only a keyboard?

Combining business with pleasure ?



String_of_characters

SMILES (**S**implified **M**olecular **I**nput **L**ine **E**ntry **S**pecification)

SMILES (Simplified Molecular Input Line Entry Specification)

The original SMILES specification was initiated by *David Weininger* at the USEPA Mid-Continent Ecology Division Laboratory in Duluth in the 1980s.

Anderson E, Veith GD, Weininger D (1987). SMILES: A line notation and computerized interpreter for chemical structures. Duluth, MN: U.S. EPA, Environmental Research Laboratory-Duluth. Report No. EPA/600/M-87/021.

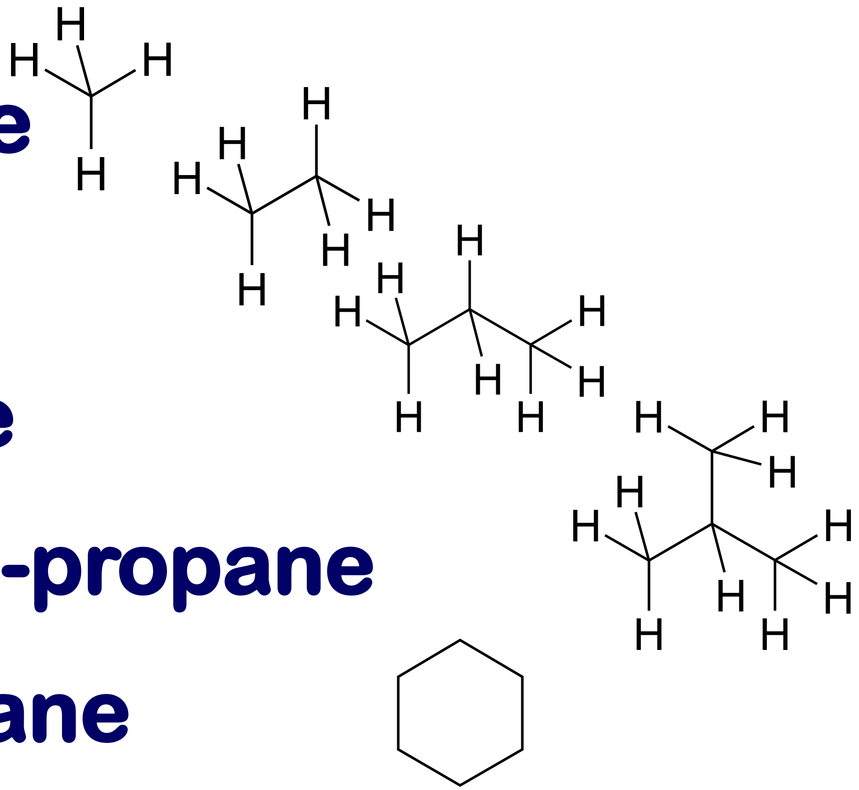
Using simple rules it is possible to represent the "connections" between "molecular fragments" (as in the *structural formula*) in a simple "string" of "alphanumeric characters" (as in the *bruta formula*).

Here is some examples:

SMILES (Simplified Molecular Input Line Entry Specification)

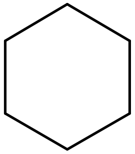
Some SMILES rules:

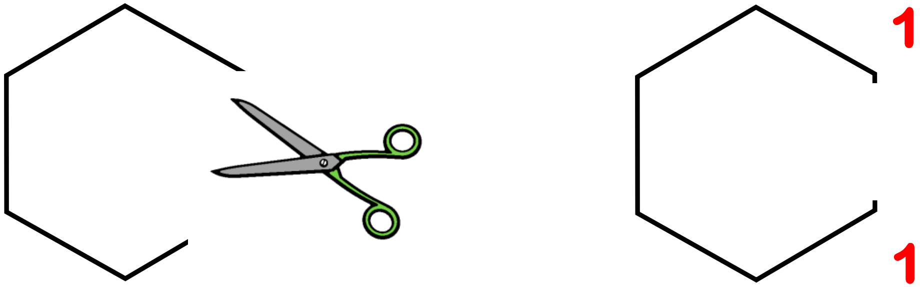
C	**methane**
CC	**ethane**
CCC	**propane**
CC(C)C	**2-methil-propane**
C1CCCCC1	**cycloesane**



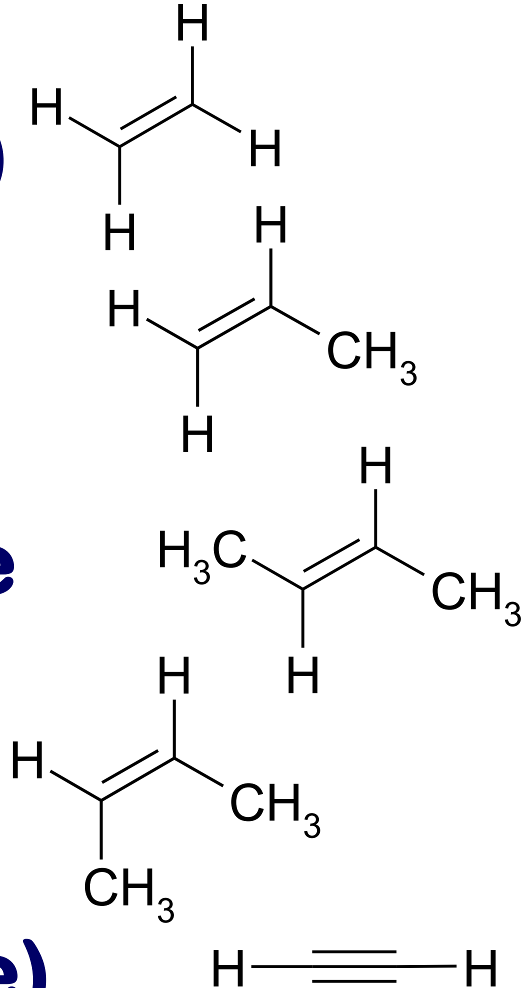
SMILES (Simplified Molecular Input Line Entry Specification)

C1CCCCC1 **cycloesane**



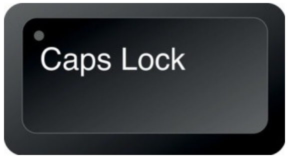


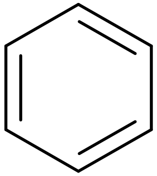
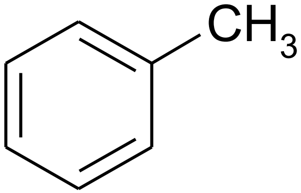
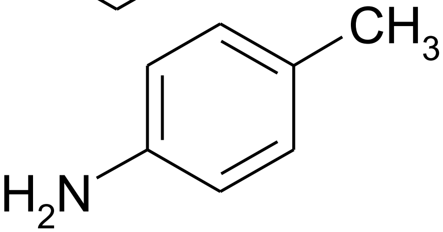
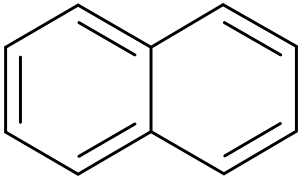
SMILES (**S**implified **M**olecular **I**nput **L**ine **E**ntry **S**pecification)



C=C	**ethene (ethylene)**
C=CC	**propene**
C/C=C/C	***trans* (E)-2-butane**
C/C=C\C	***cis* (Z)-2-butane**
C#C	**ethyne (acetylene)**

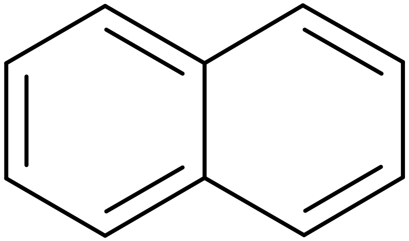
SMILES (Simplified Molecular Input Line Entry Specification)

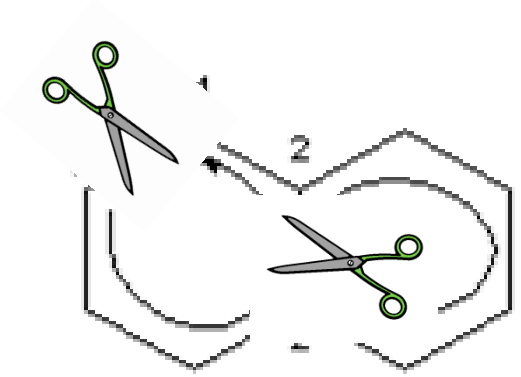


c1ccccc1	**benzene**	
Cc1ccccc1	**toluene**	
Cc1ccc(N)cc1 **Nc1ccc(C)cc1**	**4-methil-aniline**	
c12c(cccc1)cccc2	**naphthalene**	

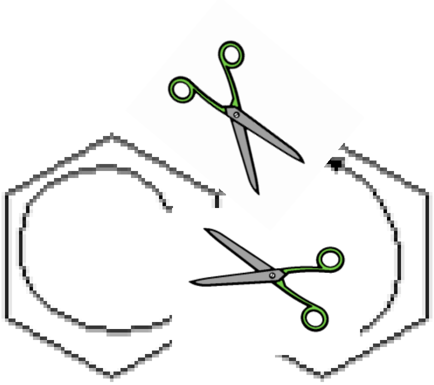
SMILES (Simplified Molecular Input Line Entry Specification)

Examples for naphtalene:

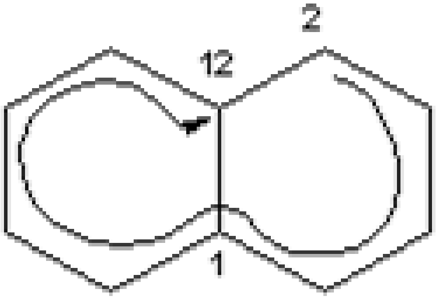




c1ccc2ccccc2c1



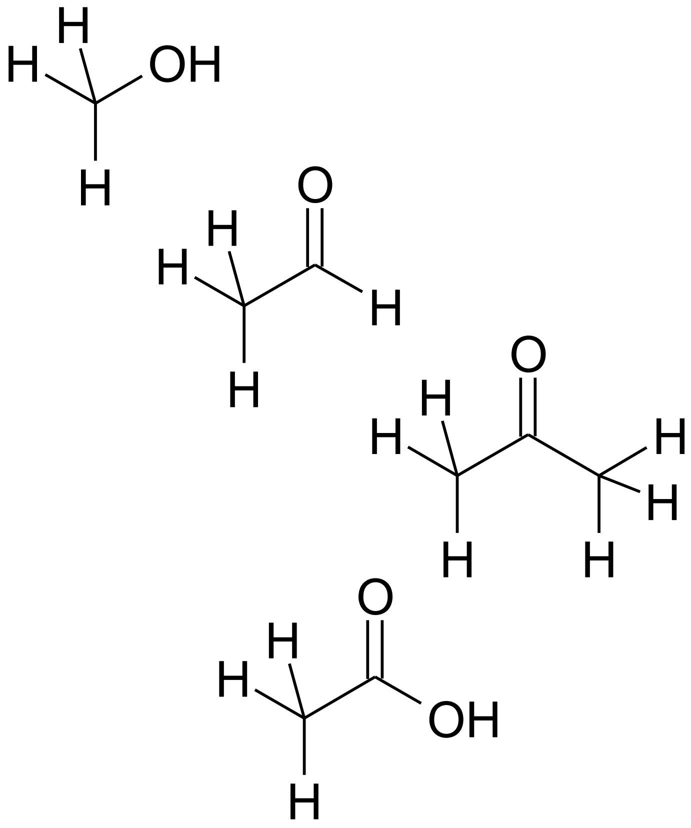
c12ccccc1cccc2



c2cccc1ccccc12

SMILES (Simplified Molecular Input Line Entry Specification)

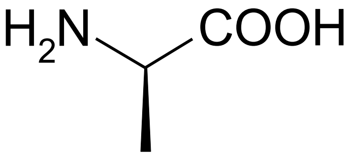
CO	**methanol**
CC=O	**ethanal**
CC(=O)C	**acetone**
CC(=O)O	**acetic acid**



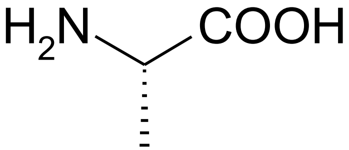
SMILES (**S**implified **M**olecular **I**nput **L**ine **E**ntry **S**pecification)

In SMILES, tetrahedral centers may be indicated by a simplified chrial specification (@ or @@) written as an atomic property following the atomic symbol of the chiral atom.

Looking at the chiral center from the direction of the "from" atom (as per atom order in SMILES), @ means "the other three atoms are listed *anti-clockwise*; @@ means *clockwise*.



C[C@@H](N)C(=O)O

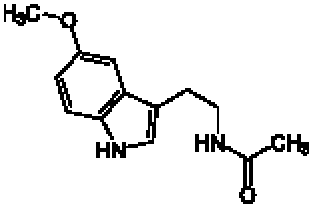


C[C@H](N)C(=O)O

SMILES (Simplified Molecular Input Line Entry Specification)

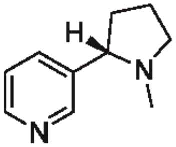
Some medchem examples:

Melatonin ($C_{13}H_{16}N_2O_2$)



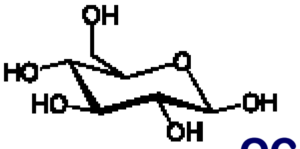
CC(=O)NCCc1c[nH]c2ccc(OC)cc12

Nicotine ($C_{10}H_{14}N_2$)



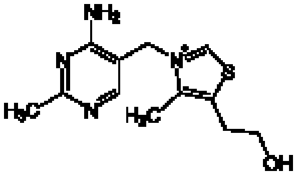
CN1CCC[C@H]1c2cccnc2

Glucose (glucopyranose) ($C_6H_{12}O_6$)



OC[C@@H](O1)[C@@H](O)[C@H](O)[C@@H](O)[C@@H](O)1

Thiamine ($C_{12}H_{17}N_4OS^+$) (vitamin B1)

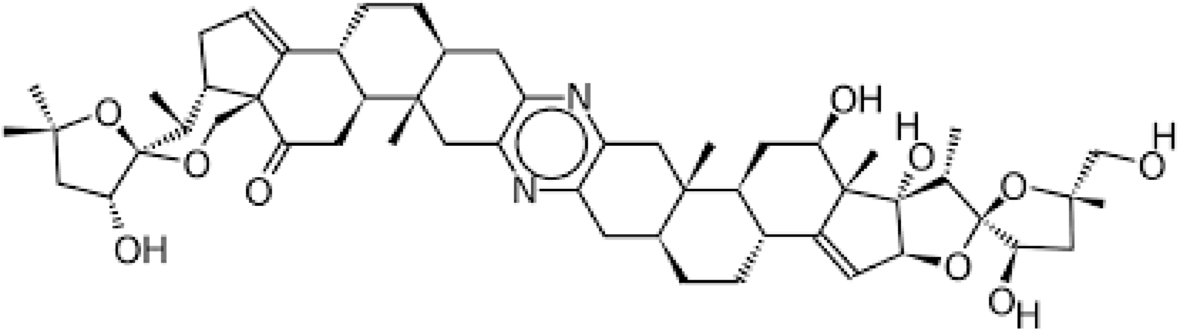


OCCc1c(C)[n+](cs1)Cc2cnc(C)nc2N

SMILES (Simplified Molecular Input Line Entry Specification)

Some examples:

Cephalostatin-1, a steroidic trisdecacyclic pyrazine with the empirical formula $C_{54}H_{74}N_2O_{10}$



Starting with the left-most methyl group in the figure:

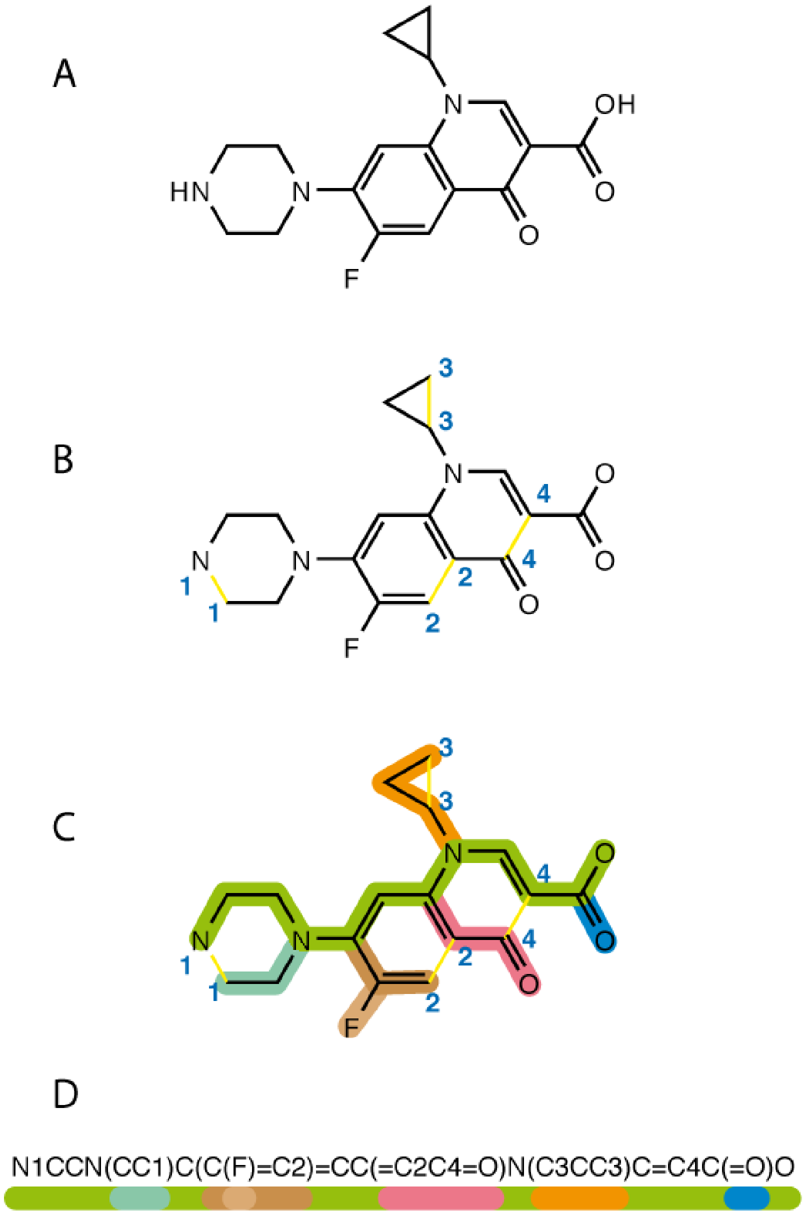
```
C[C@H]1[C@H]2CC=C3[C@]2(CO[C@]14[C@@H](CC(O4)(C)C)O)C(=O)C[C@H]5[C@H]3CC[C@@H]6[C@@]5(CC7=NC8=C(C[C@]9([C@H](C8)CC[C@@H]1[C@@H]9C[C@H]([C@]2(C1=C[C@H]1[C@@]2([C@@H]([C@@]2(O1)[C@@H](C[C@@](O2)(C)CO)O)C)O)C)O)C)N=C7C6)C
```

220 char

SMILES (Simplified Molecular Input Line Entry Specification)

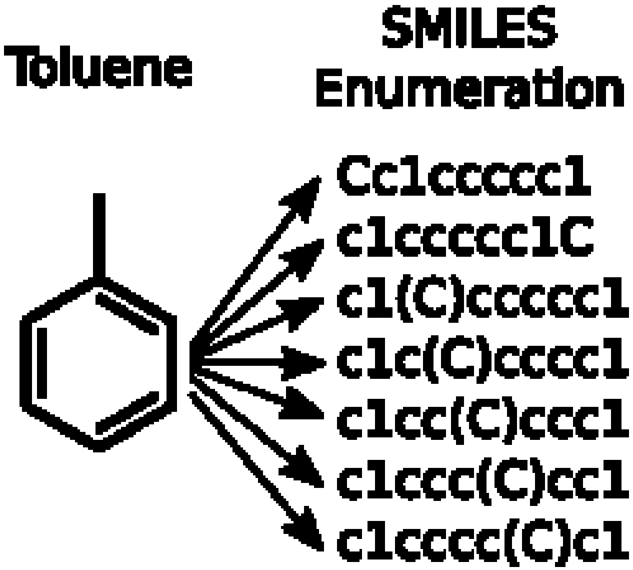
Generation of SMILES:

Break cycles, then write as branches off a main backbone. (Ciprofloxacin)



N1CCN(CC1)C(C(F)=C2)=CC(=C2C4=O)N(C3CC3)C=C4C(=O)O

SMILES (Simplified Molecular Input Line Entry Specification)



Canonical SMILES is a unique way of writing a SMILES for a molecule, where some rules about numbering defines the ordering of the atoms. This ensures that there is only one unique SMILES corresponding to one unique molecule. It is often useful to have this 1:1 correspondence:

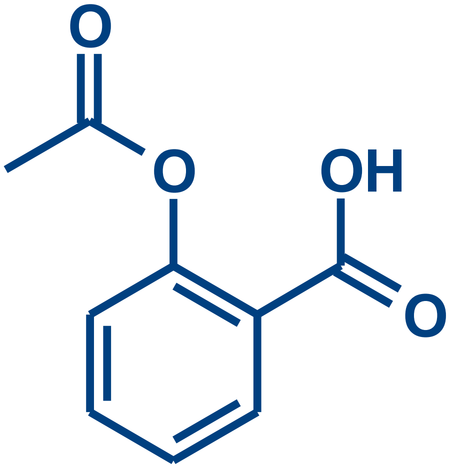
- One chemical one SMILES string;
- Same SMILES string coming from different programs;
- Improving search provess in chemical databases.

SMILES (Simplified Molecular Input Line Entry Specification)

NERDS ONLY

CANONALISING SMILES: **please check at the end for this file… and enjoy the Morgan's algorithm!**

reassuming:



O(O=)Cc1ccccc1OC(=O)C

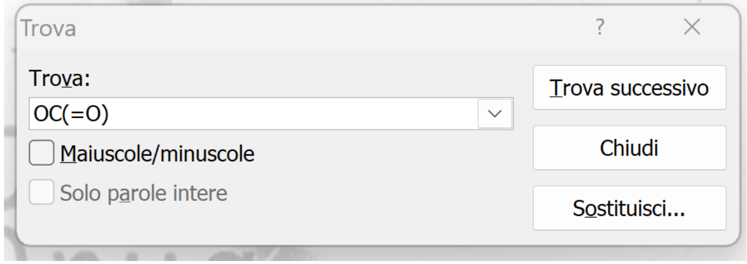
Two faces of the same medal!!!

Confidential and Property of ©2012 Molecular Modeling Section

SMILES (Simplified Molecular Input Line Entry Specification)

A powerful “**searching**” strategy:

O(O=)Cc1ccccc1OC(=O)C



Trova:
OC(=O)
☐ Maiuscole/minuscole
☐ Solo parole intere

O(O=)Cc1ccccc1OC(=O)C

SMILES and… Natural Language Processing (NLP)

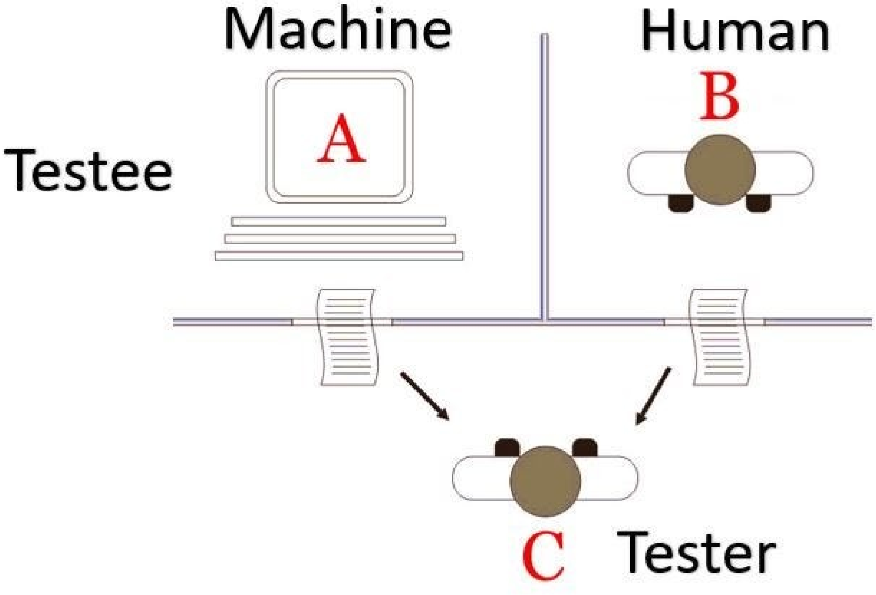
One of the essential things in the life of a human being is communication. We must communicate with others to deliver information, express our emotions, present ideas, and much more. The key to communication is language.

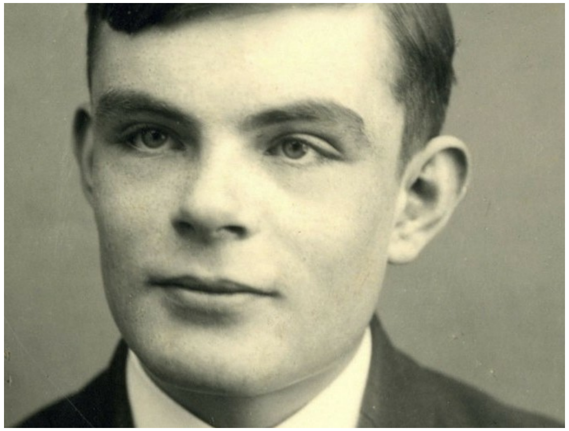
We need a common language to communicate, which both ends of the conversation can understand. Doing this is possible for humans, but it might seem a bit difficult if we talk about communicating with a computer system or the computer system communicating with us.

But we have a solution for that, Artificial Intelligence, or more specifically, a branch of Artificial Intelligence known as Natural Language Processing (NLP). It enables the computer system to understand and comprehend information like humans do.

It helps the computer system understand the literal meaning and recognize the sentiments, tone, opinions, thoughts, and other components that construct a proper conversation.

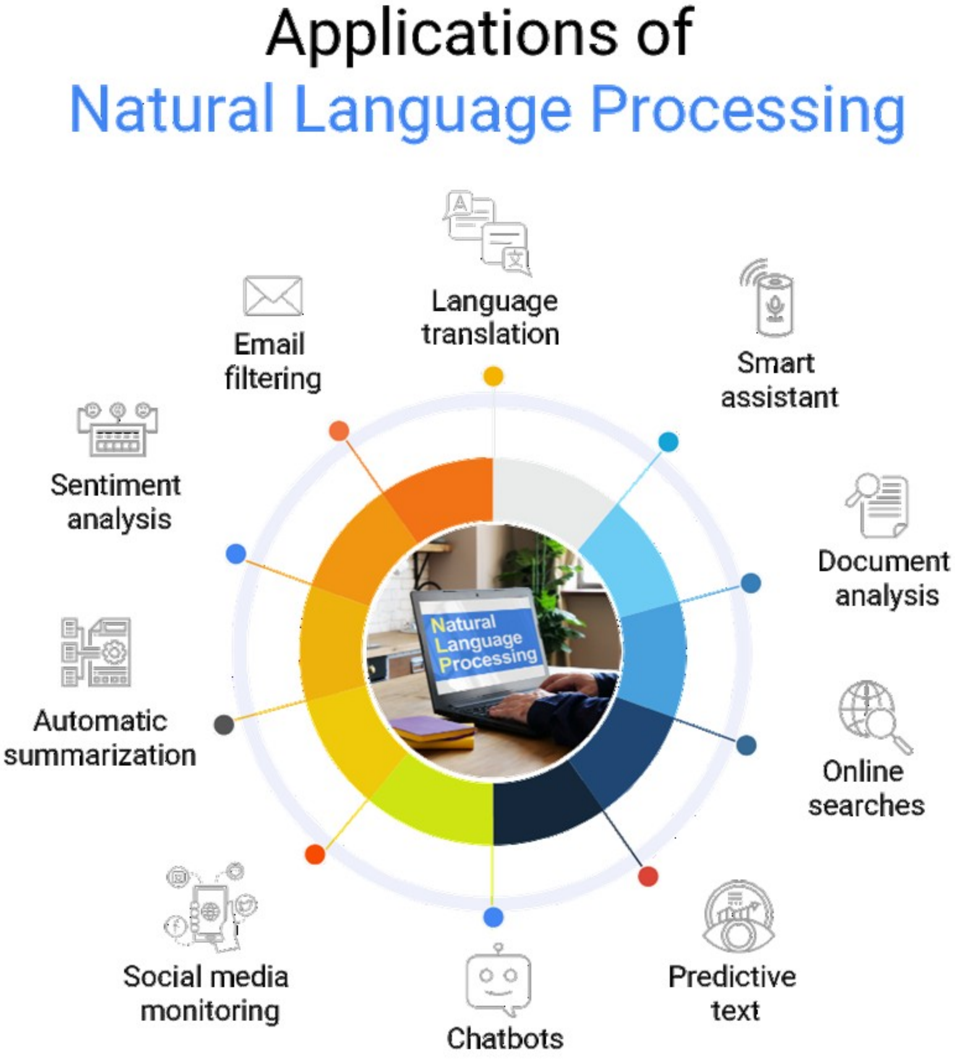
A piece of a wonderful story:





The "standard interpretation" of the Turing test, originally called the *imitation game* by Alan Turing in 1949, in which tester C, the interrogator, is given the task of trying to determine which testee – A or B – is a computer and which is a human. The interrogator is limited to using the responses to written questions to make the determination.

SMILES and... Natural Language Processing (NLP)



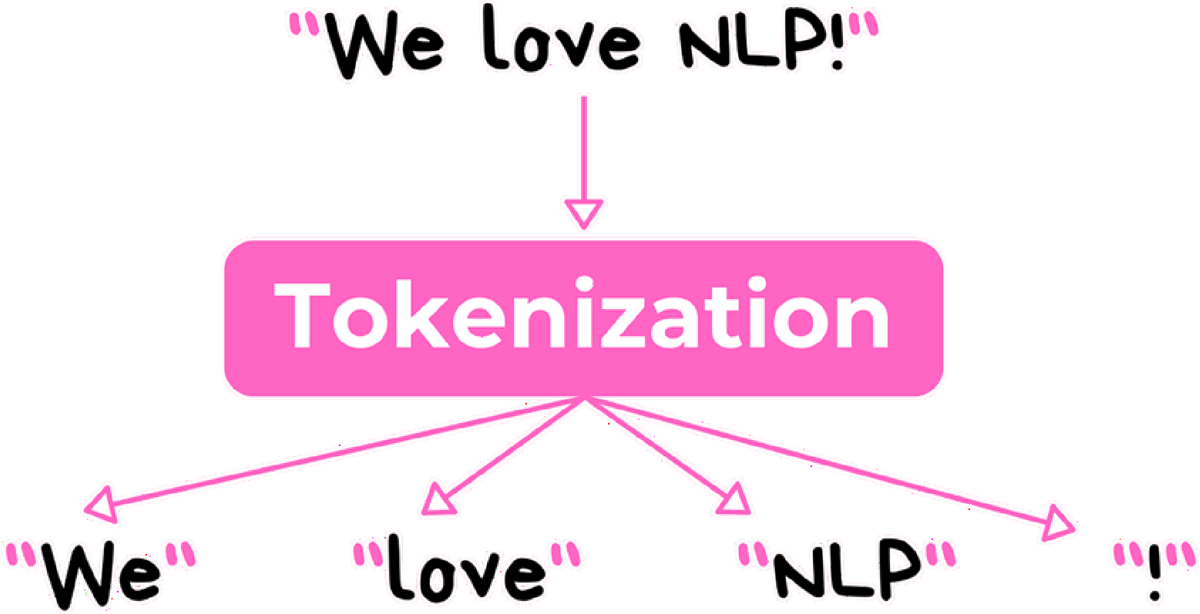
Natural Language Processing (NLP): the concept of token

In Natural Language Processing (NLP), a **token** is a single unit of text that the system considers meaningful. Tokens are usually words, but they can also be punctuation marks, parts of words, or subwords, depending on how the text is split.

This process is defined as **tokenization.**

Natural Language Processing (NLP): the concept of token

A very simple example of tokenization:



Confidential and Property of ©2012 Molecular Modeling Section

Natural Language Processing (NLP): the concept of token

And an exmple of tokenization, using SMILES

O(O=)Cc1ccccc1OC(=O)C

↓

“O” “(“ “=“ “)” “C” … “c1ccccc1” “OC(=O)C”

Confidential and Property of ©2012 Molecular Modeling Section

Natural Language Processing (NLP): from token to identifier (ID)

Each token is matched with a unique number (ID) from a vocabulary:

token	ID
“We”	101
“love”	202
“NLP”	303
“!”	404

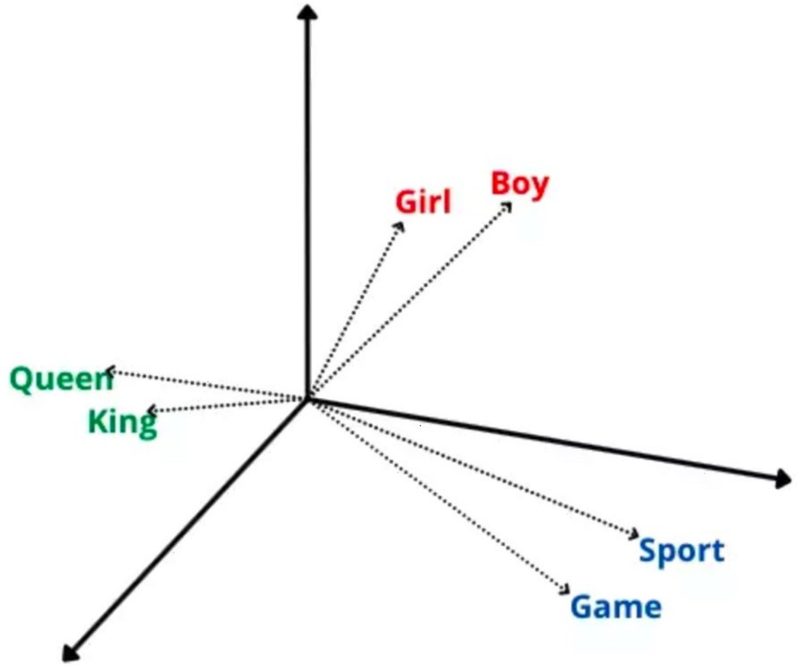
This gives us:

We love NPL! → [101, 202, 303, 404]

but these are just IDs - not very useful on their own.

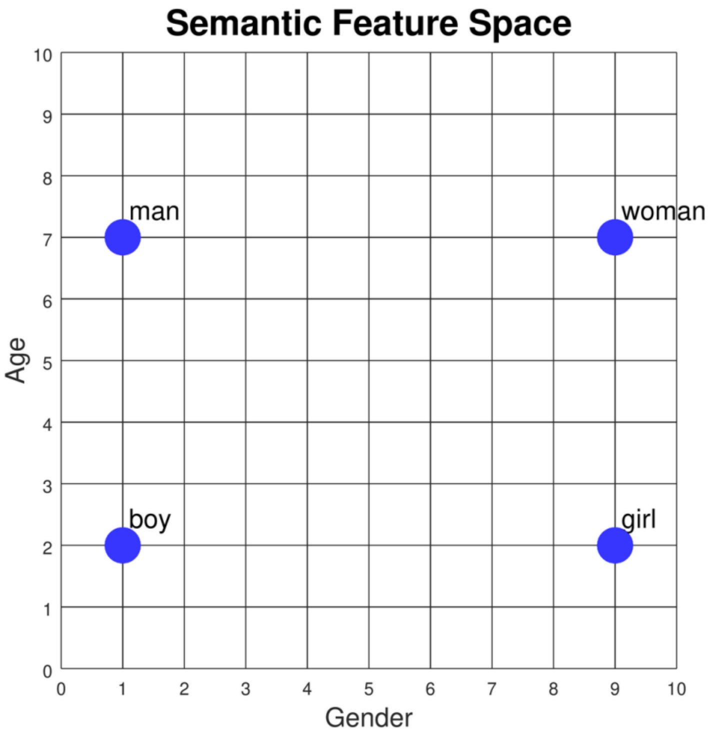
Natural Language Processing (NLP): turning IDs into vectors (Embeddings)

We use something called an embedding layer, which maps each token ID to a vector of numbers.



Natural Language Processing (NLP): turning IDs into vectors (Embeddings)

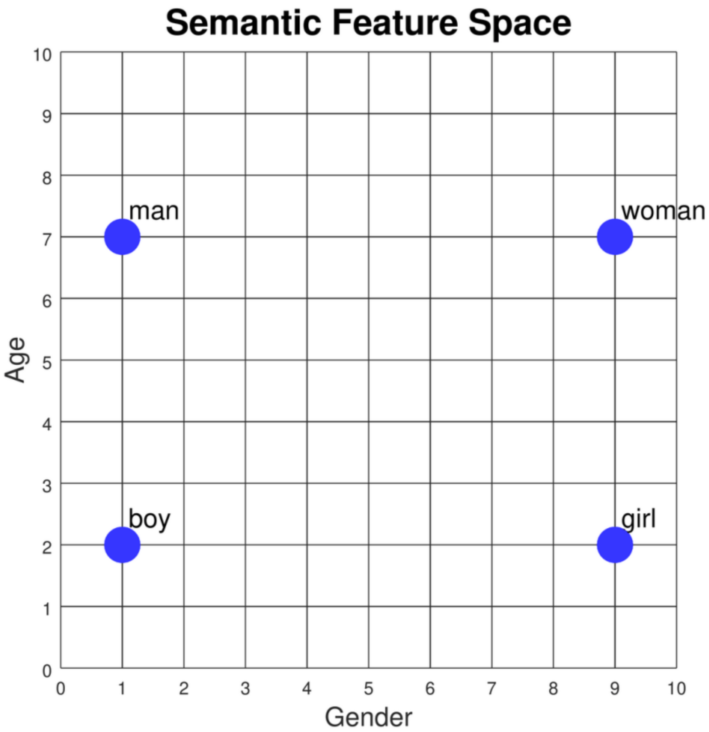
Semantic Feature Space: **consider the words "man", "woman", "boy", and "girl". Two of them refer to males, and two to females. Also, two of them refer to adults, and two to children. We can plot these worlds as points on a graph where the *x* axis axis represents gender and the *y* axis represents age:**



credits:h ttps://www.cs.cmu.edu/~dst/WordEmbeddingDemo/tutorial.html

Natural Language Processing (NLP): turning IDs into vectors (Embeddings)

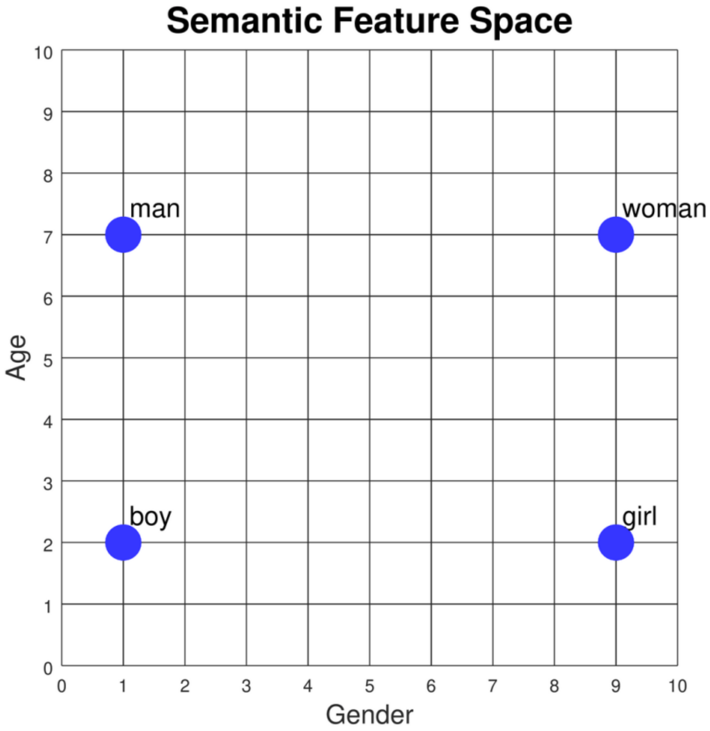
Gender and age are called *semantic features*: they represent part of the meaning of each word. If we associate a numerical scale with each feature, then we can assign coordinates to each word:

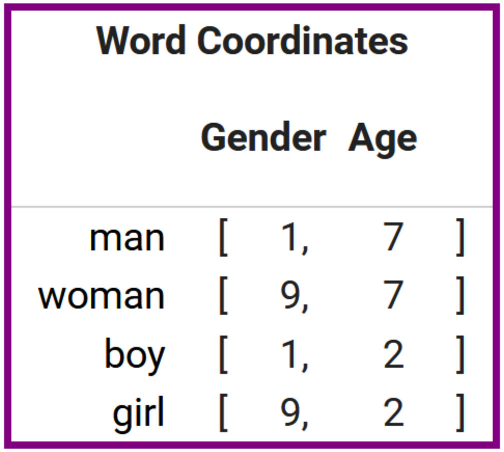


discrete space *versus* continuous space

Natural Language Processing (NLP): turning IDs into vectors (Embeddings)

Gender and age are called *semantic features*: they represent part of the meaning of each word. If we associate a numerical scale with each feature, then we can assign coordinates to each word:





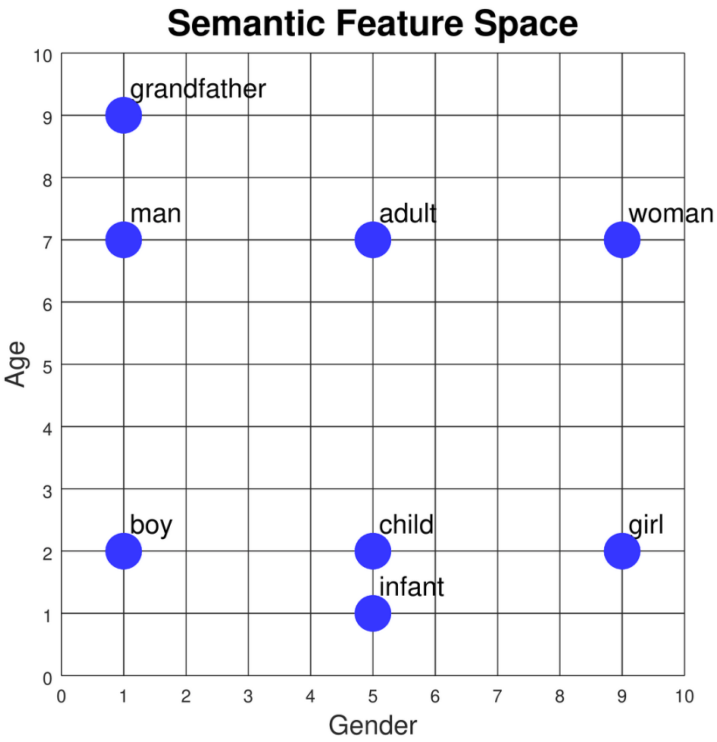
Word Coordinates

	Gender	Age
man	[1,	7]
woman	[9,	7]
boy	[1,	2]
girl	[9,	2]

credits:h ttps://www.cs.cmu.edu/~dst/WordEmbeddingDemo/tutorial.html

Natural Language Processing (NLP): turning IDs into vectors (Embeddings)

We can add new words to the plot based on their meanings. For example, where should the words "adult" and "child" go? How about "infant"? Or "grandfather"?



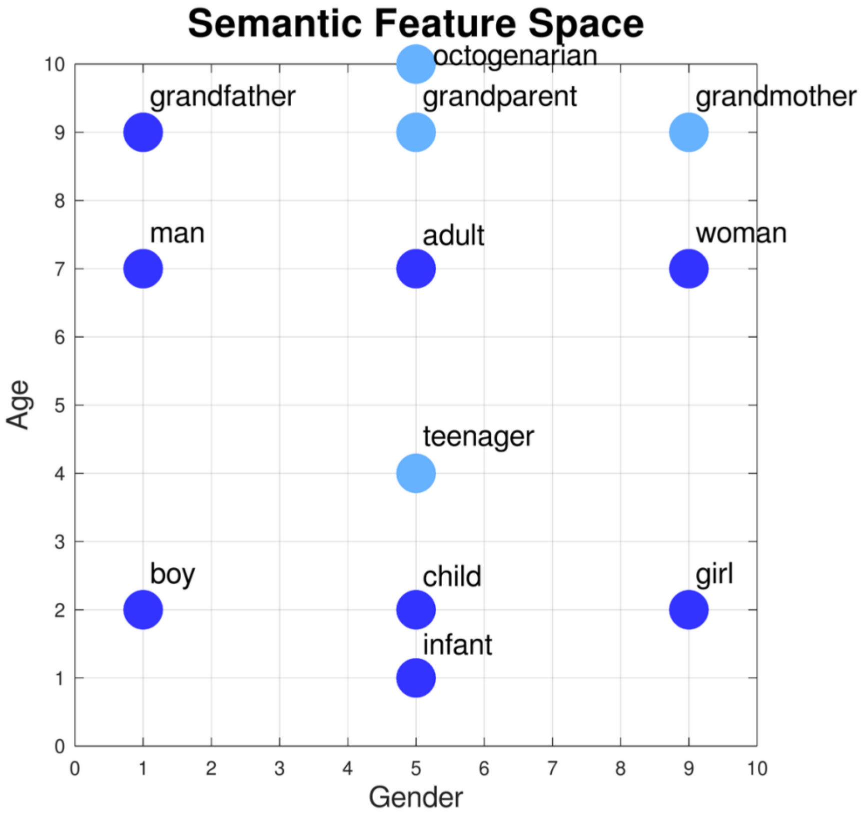
Word Coordinates

	Gender	Age
grandfather	[1,	9]
man	[1,	7]
adult	[5,	7]
woman	[9,	7]
boy	[1,	2]
child	[5,	2]
girl	[9,	2]
infant	[5,	1]

credits:h ttps://www.cs.cmu.edu/~dst/WordEmbeddingDemo/tutorial.html

Natural Language Processing (NLP): turning IDs into vectors (Embeddings)

How would you represent the words "grandmother", "grandparent", "teenager", and "octogenarian"?



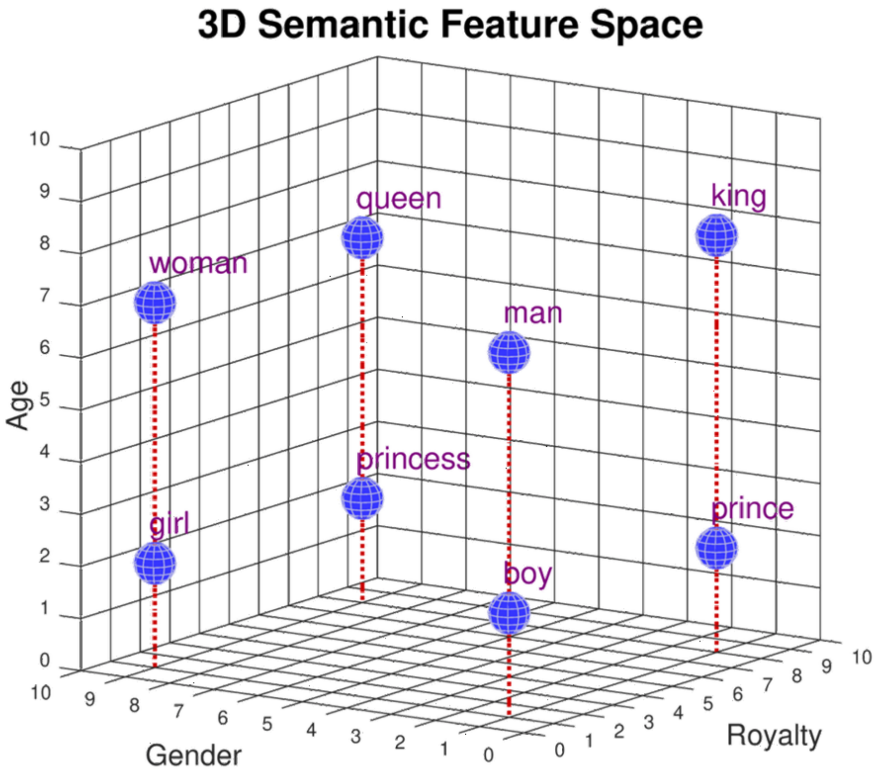
Word Coordinates

	Gender	Age
grandmother	[9,	9]
grandparent	[5,	9]
octogenarian	[5,	10]
teenager	[5,	4]

credits:h ttps://www.cs.cmu.edu/~dst/WordEmbeddingDemo/tutorial.html

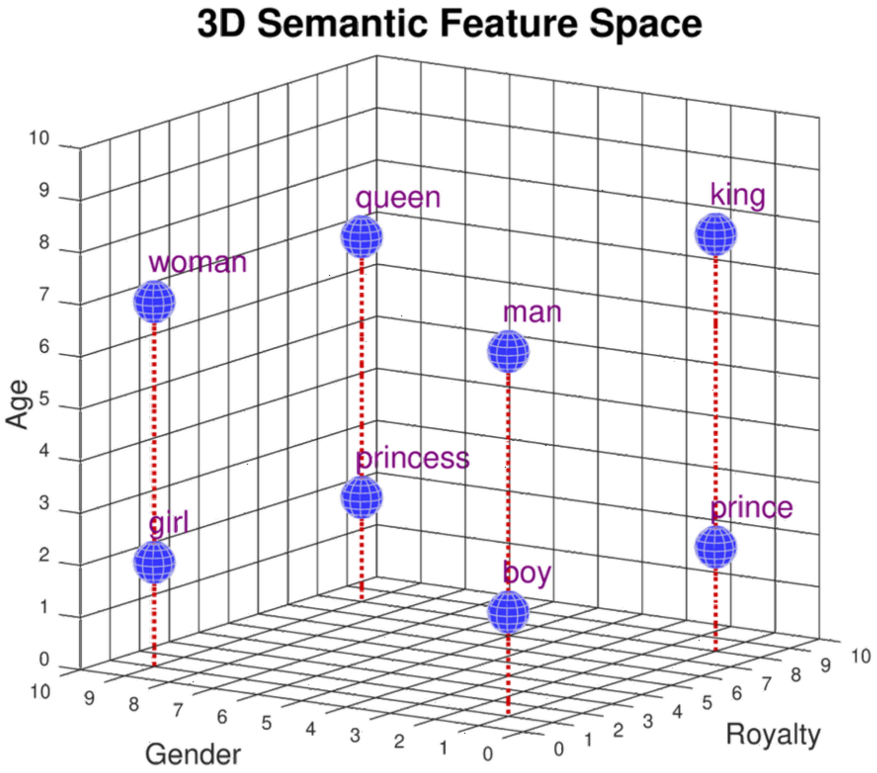
Natural Language Processing (NLP): turning IDs into vectors (Embeddings)

Now let's consider the words "king", "queen", "prince", and "princess". They have the same gender and age attributes as "man", "woman", "boy', and "girl". But they don't mean the same thing. In order to distinguish "man" from "king", "woman" from "queen", and so on, we need to introduce a new semantic feature in which they differ. Let's call it "royalty". Now we have to plot the points in a 3-dimensional space:



credits:h ttps://www.cs.cmu.edu/~dst/WordEmbeddingDemo/tutorial.html

Natural Language Processing (NLP): turning IDs into vectors (Embeddings)



Word Coordinates

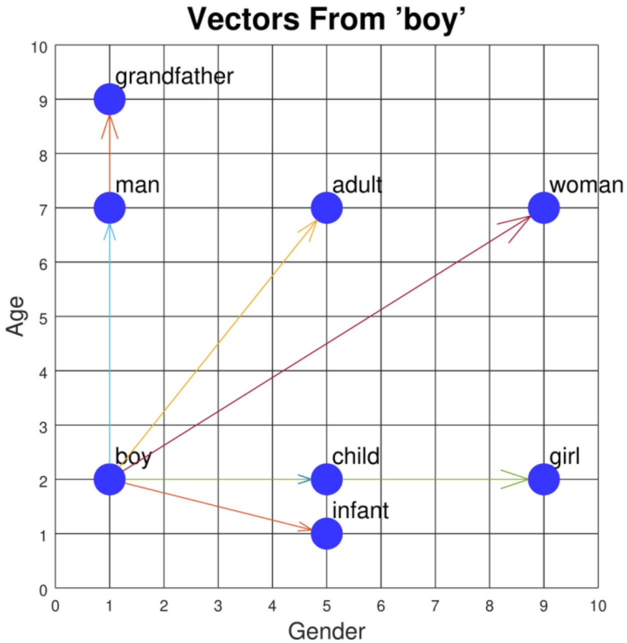
	Gender	Age	Royalty
man	[1,	7,	1]
woman	[9,	7,	1]
boy	[1,	2,	1]
girl	[9,	2,	1]
king	[1,	8,	8]
queen	[9,	7,	8]
prince	[1,	2,	8]
princess	[9,	2,	8]

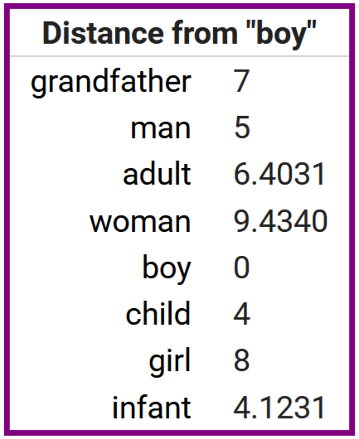
credits:h ttps://www.cs.cmu.edu/~dst/WordEmbeddingDemo/tutorial.html

Natural Language Processing (NLP): turning IDs into vectors (Embeddings)

Uses of Semantic Feature Vectors

What can we do with these numerical representations? One thing we can use them for is judging similarity between words. For example, "boy" is more similar to "girl" than to "queen" because the *distance* from "boy" to "girl" is less than the distance from "boy" to "queen". There are several ways to measure distance.



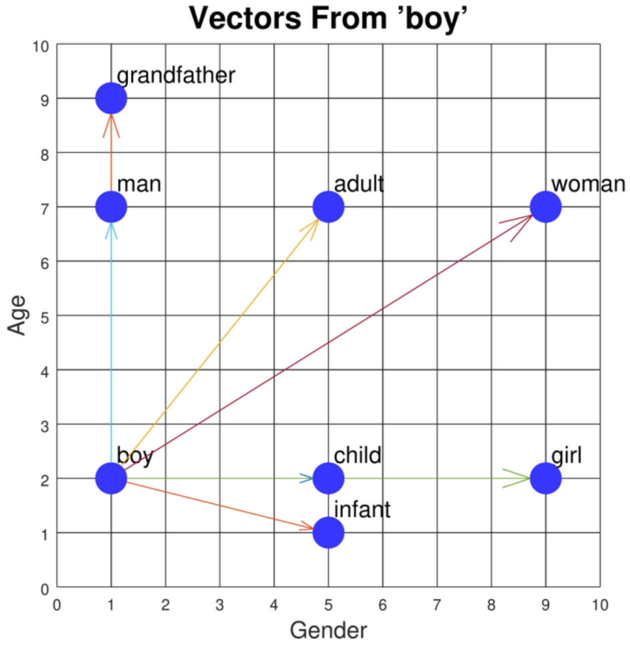


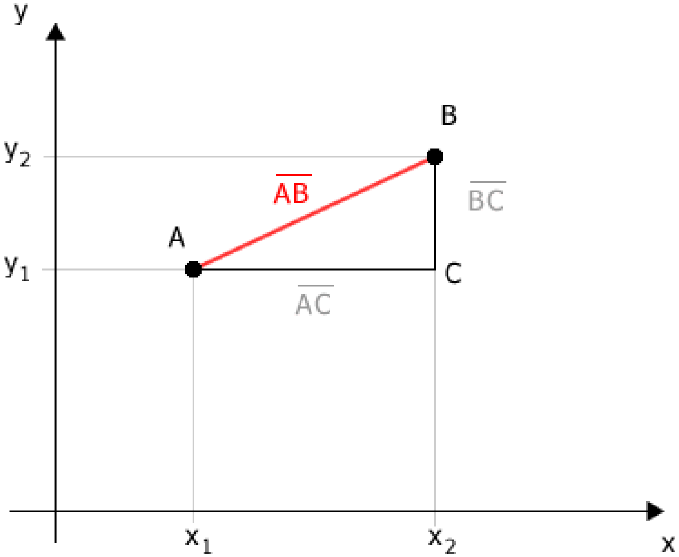
Distance from "boy"	
grandfather	7
man	5
adult	6.4031
woman	9.4340
boy	0
child	4
girl	8
infant	4.1231

credits: h ttps://www.cs.cmu.edu/~dst/WordEmbeddingDemo/tutorial.html

Natural Language Processing (NLP): turning IDs into vectors (Embeddings)

Remember Pitagora's theorem?



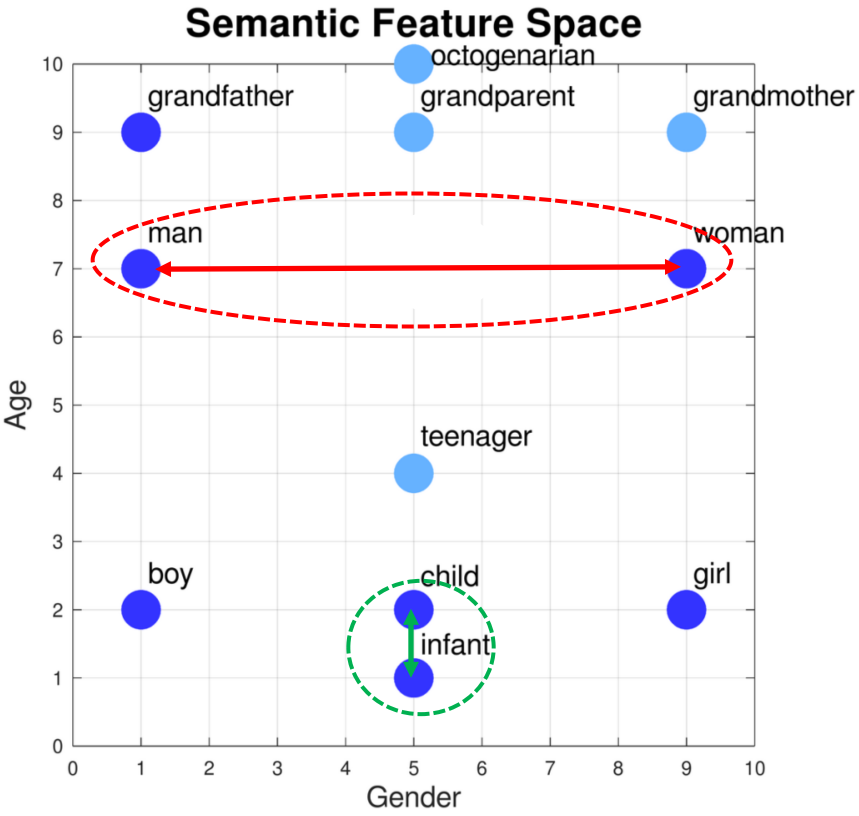


$$\overline{AB} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

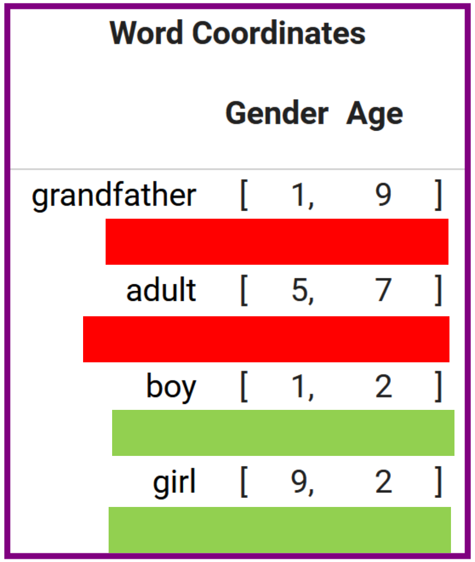
Natural Language Processing (NLP): turning IDs into vectors (Embeddings)

Uses of Semantic Feature Vectors

A very interesting application of the vector's distance:



synonymous:

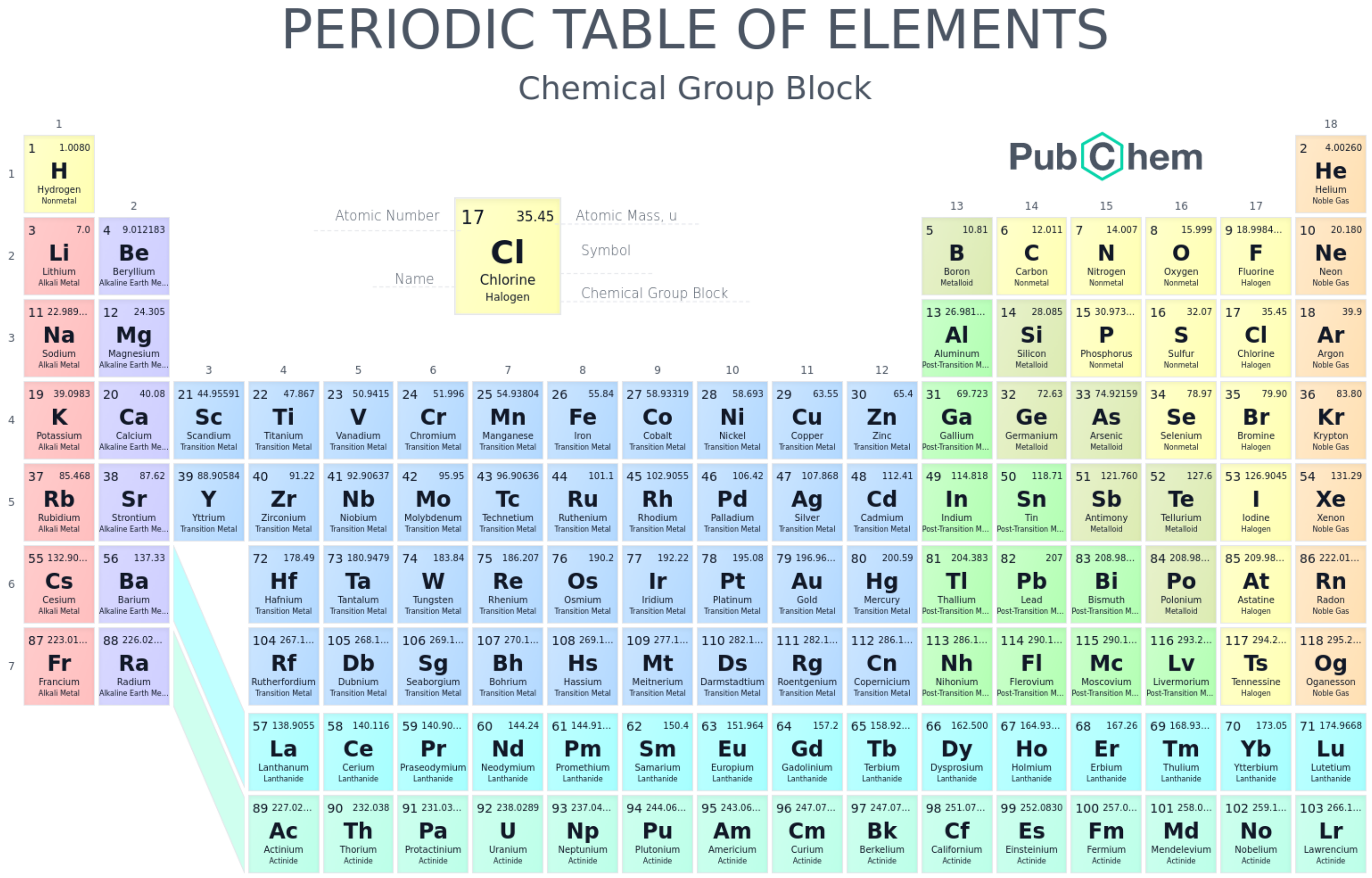


Word Coordinates

	Gender	Age
grandfather	[1,	9]
adult	[5,	7]
boy	[1,	2]
girl	[9,	2]

antonym:

back to chemistry… our first simple tokenization space



PERIODIC TABLE OF ELEMENTS

Chemical Group Block

PubChem

Atomic Number — 17 35.45 — Atomic Mass, u

Cl — Symbol

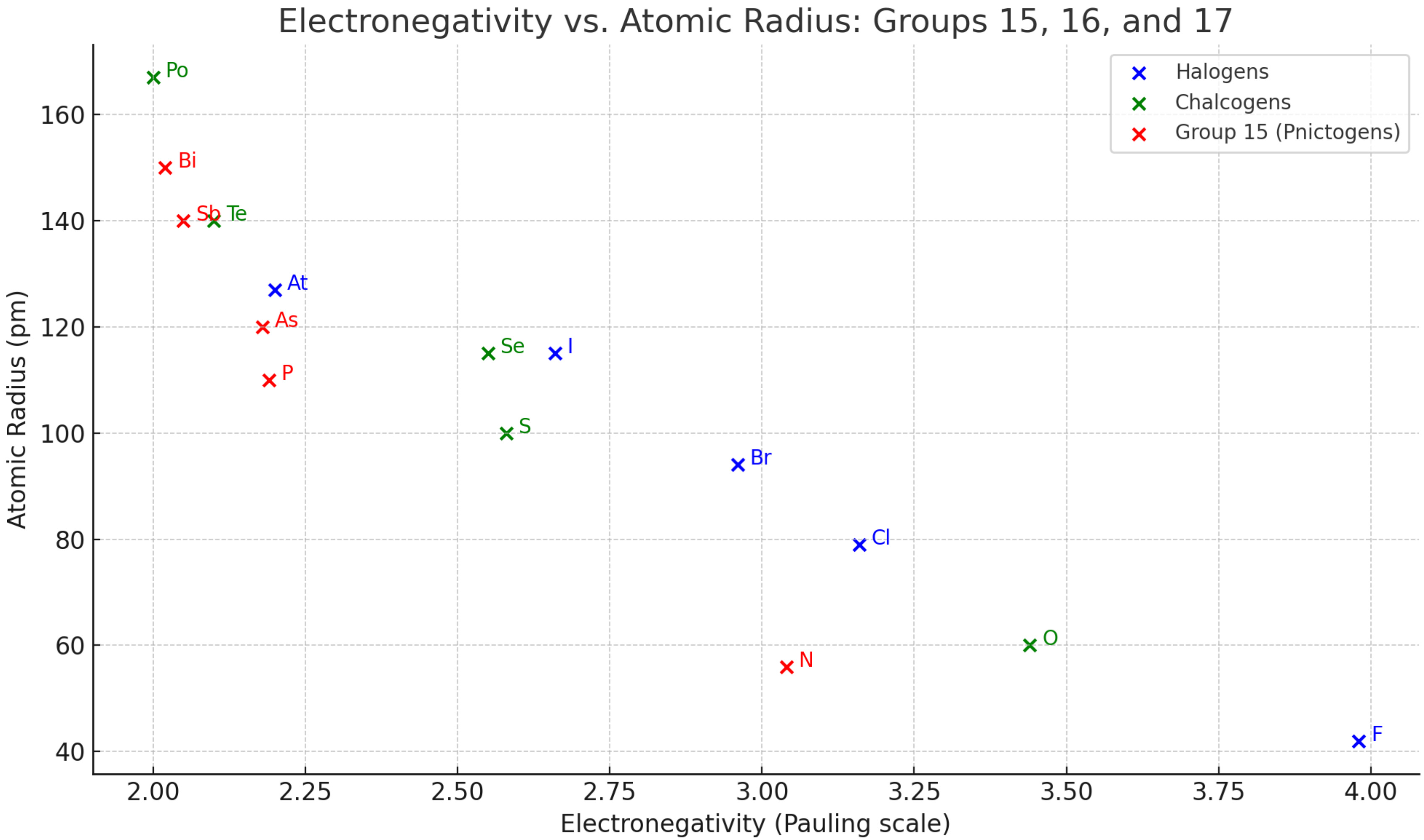
Name — Chlorine

Halogen — Chemical Group Block

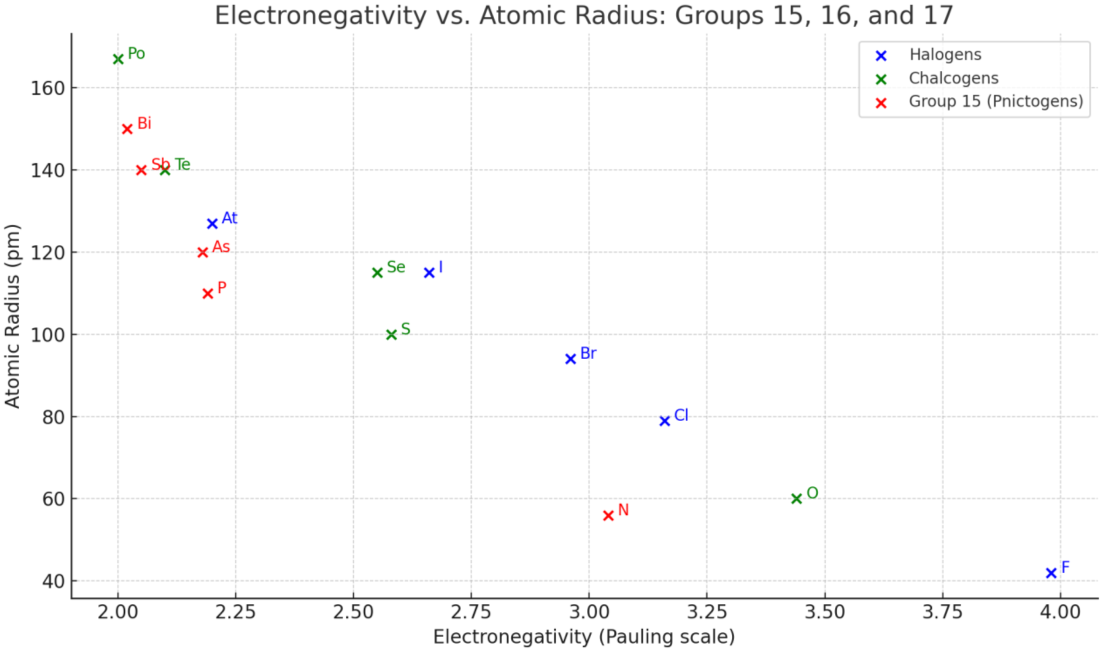
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1 1.0080 H Hydrogen Nonmetal																	2 4.00260 He Helium Noble Gas
2	3 7.0 Li Lithium Alkali Metal	4 9.012183 Be Beryllium Alkaline Earth Me...											5 10.81 B Boron Metalloid	6 12.011 C Carbon Nonmetal	7 14.007 N Nitrogen Nonmetal	8 15.999 O Oxygen Nonmetal	9 18.9984... F Fluorine Halogen	10 20.180 Ne Neon Noble Gas
3	11 22.989... Na Sodium Alkali Metal	12 24.305 Mg Magnesium Alkaline Earth Me...											13 26.981... Al Aluminum Post-Transition M...	14 28.085 Si Silicon Metalloid	15 30.973... P Phosphorus Nonmetal	16 32.07 S Sulfur Nonmetal	17 35.45 Cl Chlorine Halogen	18 39.9 Ar Argon Noble Gas
4	19 39.0983 K Potassium Alkali Metal	20 40.08 Ca Calcium Alkaline Earth Me...	21 44.95591 Sc Scandium Transition Metal	22 47.867 Ti Titanium Transition Metal	23 50.9415 V Vanadium Transition Metal	24 51.996 Cr Chromium Transition Metal	25 54.93804 Mn Manganese Transition Metal	26 55.84 Fe Iron Transition Metal	27 58.93319 Co Cobalt Transition Metal	28 58.693 Ni Nickel Transition Metal	29 63.55 Cu Copper Transition Metal	30 65.4 Zn Zinc Transition Metal	31 69.723 Ga Gallium Post-Transition M...	32 72.63 Ge Germanium Metalloid	33 74.92159 As Arsenic Metalloid	34 78.97 Se Selenium Nonmetal	35 79.90 Br Bromine Halogen	36 83.80 Kr Krypton Noble Gas
5	37 85.468 Rb Rubidium Alkali Metal	38 87.62 Sr Strontium Alkaline Earth Me...	39 88.90584 Y Yttrium Transition Metal	40 91.22 Zr Zirconium Transition Metal	41 92.90637 Nb Niobium Transition Metal	42 95.95 Mo Molybdenum Transition Metal	43 96.90636 Tc Technetium Transition Metal	44 101.1 Ru Ruthenium Transition Metal	45 102.9055 Rh Rhodium Transition Metal	46 106.42 Pd Palladium Transition Metal	47 107.868 Ag Silver Transition Metal	48 112.41 Cd Cadmium Transition Metal	49 114.818 In Indium Post-Transition M...	50 118.71 Sn Tin Post-Transition M...	51 121.760 Sb Antimony Metalloid	52 127.6 Te Tellurium Metalloid	53 126.9045 I Iodine Halogen	54 131.29 Xe Xenon Noble Gas
6	55 132.90... Cs Cesium Alkali Metal	56 137.33 Ba Barium Alkaline Earth Me...		72 178.49 Hf Hafnium Transition Metal	73 180.9479 Ta Tantalum Transition Metal	74 183.84 W Tungsten Transition Metal	75 186.207 Re Rhenium Transition Metal	76 190.2 Os Osmium Transition Metal	77 192.22 Ir Iridium Transition Metal	78 195.08 Pt Platinum Transition Metal	79 196.96... Au Gold Transition Metal	80 200.59 Hg Mercury Transition Metal	81 204.383 Tl Thallium Post-Transition M...	82 207 Pb Lead Post-Transition M...	83 208.98... Bi Bismuth Post-Transition M...	84 208.98... Po Polonium Metalloid	85 209.98... At Astatine Halogen	86 222.01... Rn Radon Noble Gas
7	87 223.01... Fr Francium Alkali Metal	88 226.02... Ra Radium Alkaline Earth Me...		104 267.1... Rf Rutherfordium Transition Metal	105 268.1... Db Dubnium Transition Metal	106 269.1... Sg Seaborgium Transition Metal	107 270.1... Bh Bohrium Transition Metal	108 269.1... Hs Hassium Transition Metal	109 277.1... Mt Meitnerium Transition Metal	110 282.1... Ds Darmstadtium Transition Metal	111 282.1... Rg Roentgenium Transition Metal	112 286.1... Cn Copernicium Transition Metal	113 286.1... Nh Nihonium Post-Transition M...	114 290.1... Fl Flerovium Post-Transition M...	115 290.1... Mc Moscovium Post-Transition M...	116 293.2... Lv Livermorium Post-Transition M...	117 294.2... Ts Tennessine Halogen	118 295.2... Og Oganesson Noble Gas
				57 138.9055 La Lanthanum Lanthanide	58 140.116 Ce Cerium Lanthanide	59 140.90... Pr Praseodymium Lanthanide	60 144.24 Nd Neodymium Lanthanide	61 144.91... Pm Promethium Lanthanide	62 150.4 Sm Samarium Lanthanide	63 151.964 Eu Europium Lanthanide	64 157.2 Gd Gadolinium Lanthanide	65 158.92... Tb Terbium Lanthanide	66 162.500 Dy Dysprosium Lanthanide	67 164.93... Ho Holmium Lanthanide	68 167.26 Er Erbium Lanthanide	69 168.93... Tm Thulium Lanthanide	70 173.05 Yb Ytterbium Lanthanide	71 174.9668 Lu Lutetium Lanthanide
				89 227.02... Ac Actinium Actinide	90 232.038 Th Thorium Actinide	91 231.03... Pa Protactinium Actinide	92 238.0289 U Uranium Actinide	93 237.04... Np Neptunium Actinide	94 244.06... Pu Plutonium Actinide	95 243.06... Am Americium Actinide	96 247.07... Cm Curium Actinide	97 247.07... Bk Berkelium Actinide	98 251.07... Cf Californium Actinide	99 252.0830 Es Einsteinium Actinide	100 257.0... Fm Fermium Actinide	101 258.0... Md Mendelevium Actinide	102 259.1... No Nobelium Actinide	103 266.1... Lr Lawrencium Actinide

credits: PubChem - https://pubchem.ncbi.nlm.nih.gov/periodic-table/

… and a first simple example of semantic chemical space



… and easy to embedded!!!



Chemical Word Coordinates

	EN	AR
F	[4.0,	64]
Cl	[3.5,	100]
Br	[2.9,	114]
I	[2.6,	133]
At	[2.2,	140]

…

In principle we can copy/paste the same approach for any chemical token!

… and easy to embedded!!!

Some example in organic chemistry, from SMILES’s tokenization:

Liner *versus* Cyclic

Aromatic *versus* Non Aromatic

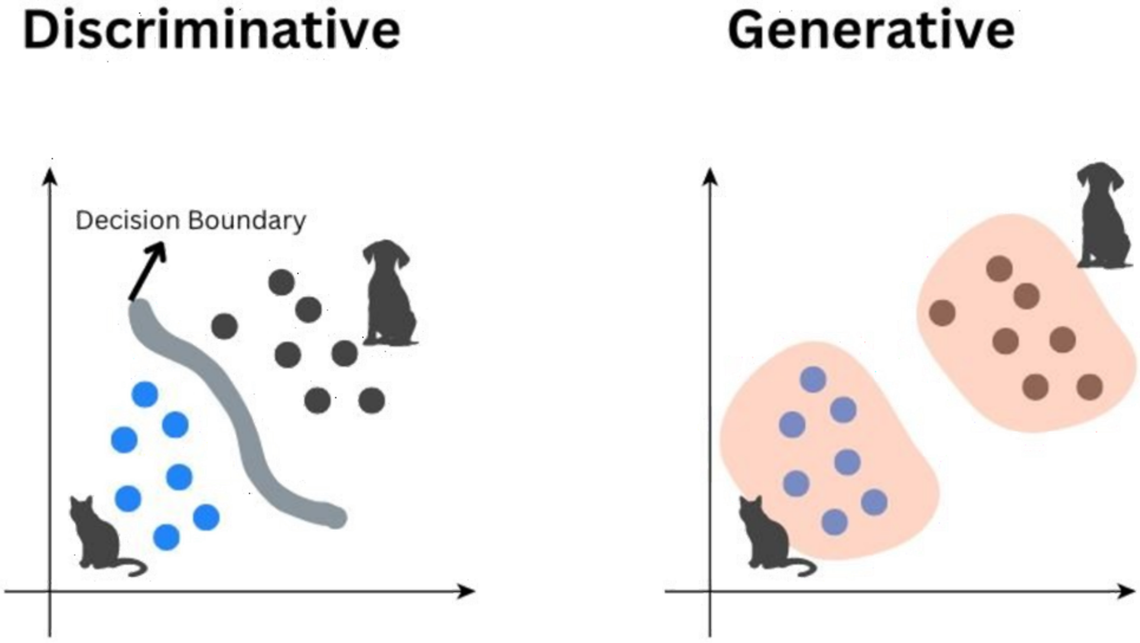
Polar *versus* Apolar

With *or* Without Nitrogen/Oxygen/…

…

We are ready to GENERATE: lesson 0

credits: https://viso.ai/deep-learning/generative-ai/



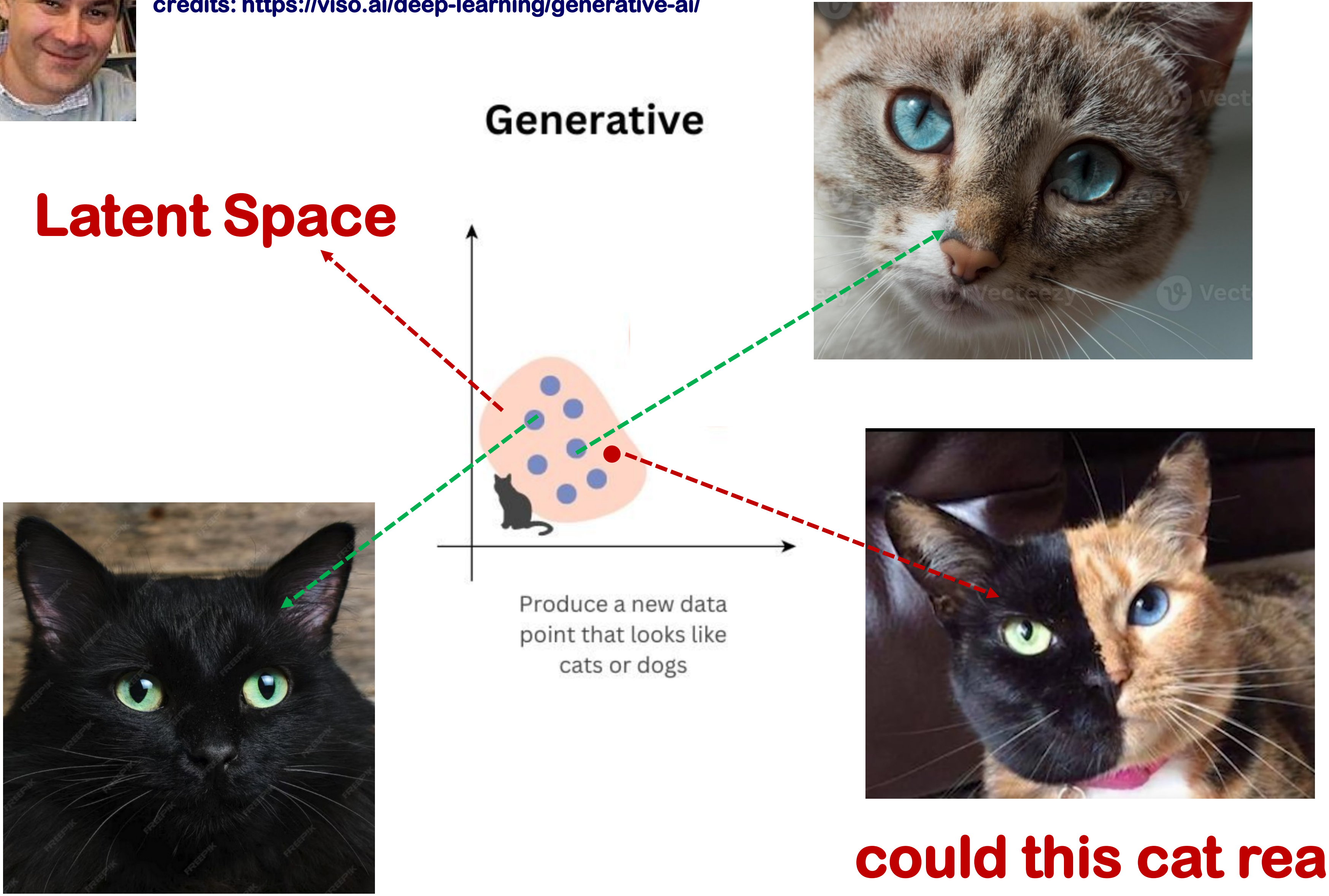
Classify or Label data point as cat or dog

Produce a new data point that looks like cats or dogs

Consider the x and y axis as a space where data points exist, each data point is either a cat or a dog. A *discriminative* model task is to predict what each data point is, even with new data. On the other hand, the *generative AI task* is to create new data points that look like the existing ones.

We are ready to GENERATE: lesson 0

credits: https://viso.ai/deep-learning/generative-ai/



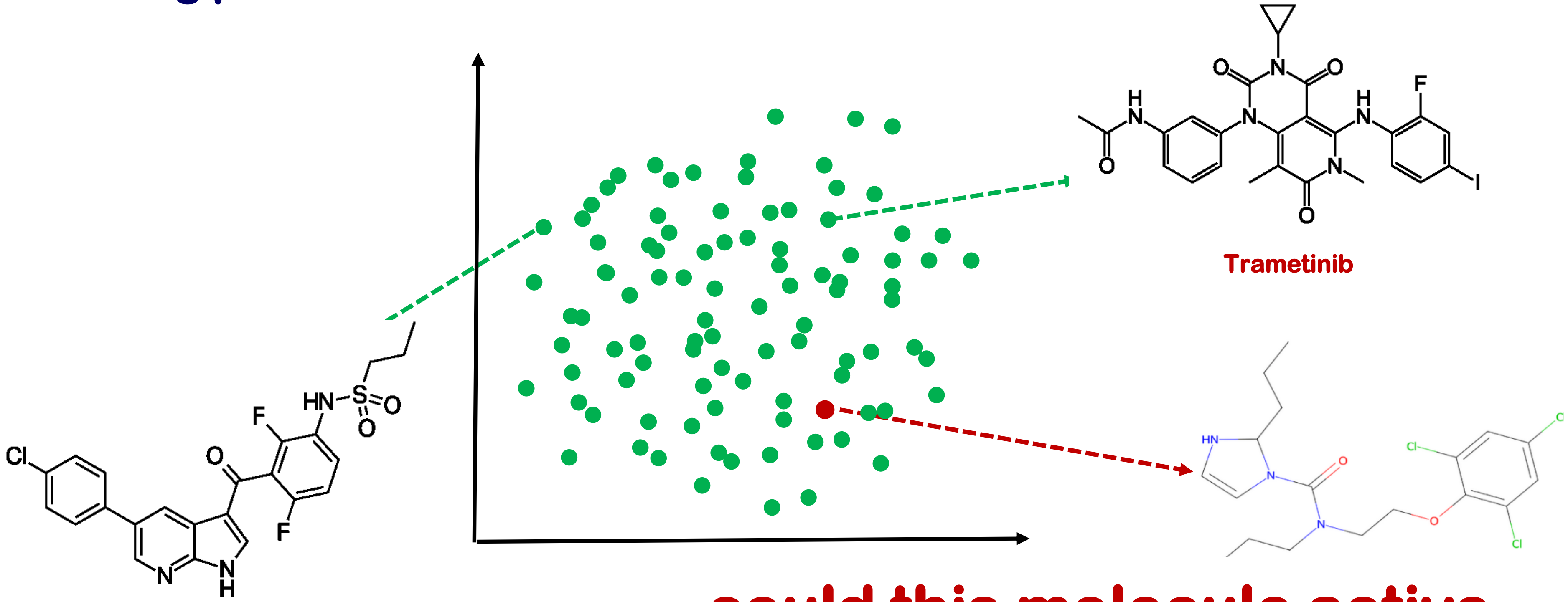
Latent Space

Produce a new data point that looks like cats or dogs

could this cat real?

We are ready to GENERATE: lesson 0 – a medchem example:

Melanoma is the third most common cancer in young adults under 30 years of age. 140 molecules with higher antitumor efficacy were used as a starting point.



could this molecule active against melanoma?

Rigoni D, et al. Int J Mol Sci. 2024 Jun 4;25(11):6186.

GENERATIVE models: a couple of approaches...

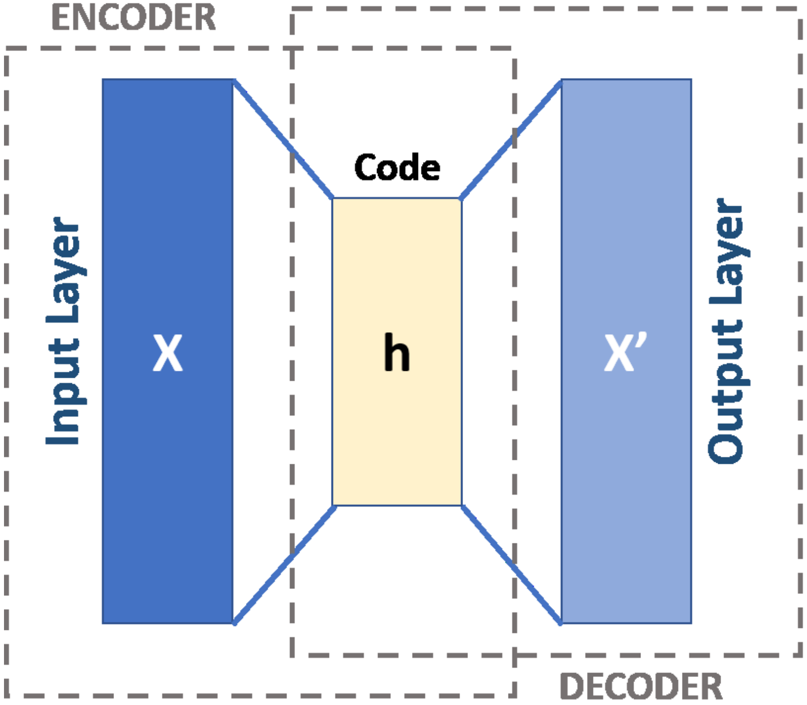
- **Autoencoder (AE)** → compress SMILES into a latent space (global embedding of the molecule) and generate new molecules by sampling from there

- **Transformer** (ex. GPT) → generate complete SMILES as if they were sentences

credits: https://viso.ai/deep-learning/generative-ai/

GENERATIVE models: what is an Autoencoder (AE)

credits: https://viso.ai/deep-learning/generative-ai/



A schema of an *autoencoder*. An autoencoder has two main parts: an *encoder* that maps the message to a code, and a *decoder* that reconstructs the message from the code.

GENERATIVE models: what is an encoder

credits: https://viso.ai/deep-learning/generative-ai/

Encoders: Transforming Inputs into *Latent Representations*

Encoders are the initial half of the generative process. They are responsible for transforming raw input data, such as images or text (SMILES), into a compact, latent representation. This latent space representation captures the essence of the input data in a lower-dimensional form, highlighting the crucial features that define it. This process is akin to how the human brain processes information — abstracting away irrelevant details to focus on the essence of an object.

Encoders are particularly useful for dimensionality reduction, feature extraction, and anomaly detection. In generative AI, these encoded representations serve as a bridge between the raw data and the generative model, making it easier to manipulate and transform data for creative purposes.

GENERATIVE models: what is an encoder

credits: https://viso.ai/deep-learning/generative-ai/

SMILES Encoders workflow:

SMILES : CCO

Tokens : “C” “C” “O”

Token ID: [101, 101, 202] (*vocabulary*)

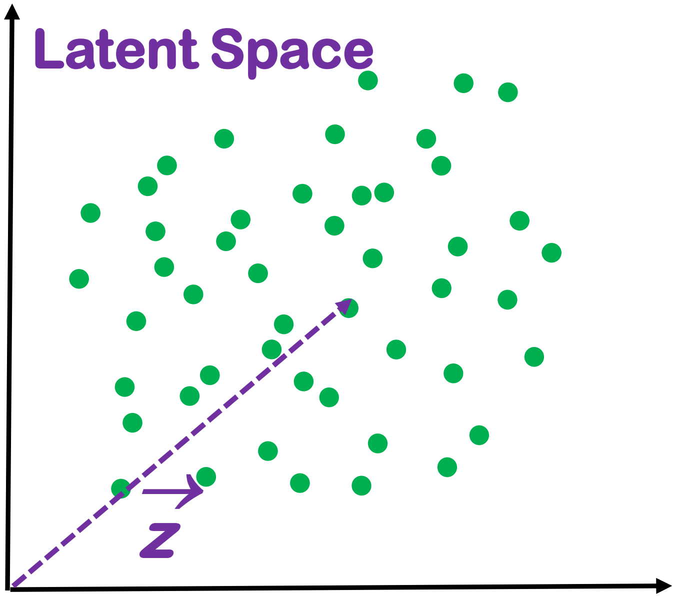
Embedding: [101, 101, 202] → [0.2, 0.8], [0.2, 0.8], [0.1, 0.3] (*if we use only 2 chemical features, is equal to a 3 x 2 matrix*)

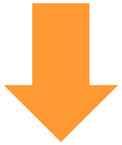
ENCODER: [0.2, 0.8], [0.2, 0.8], [0.1, 0.3] → [0.17, -0.42, …, 1.23] **LATENT VECTOR, z** *(from a 3x2 matrix to a single vector with a dimensionality that depend from the model)*

GENERATIVE models: what is an encoder

SMILES Encoders workflow:

SMILES → Tokens → Token ID → Embedding → ENCODER (*latent vector, z*)

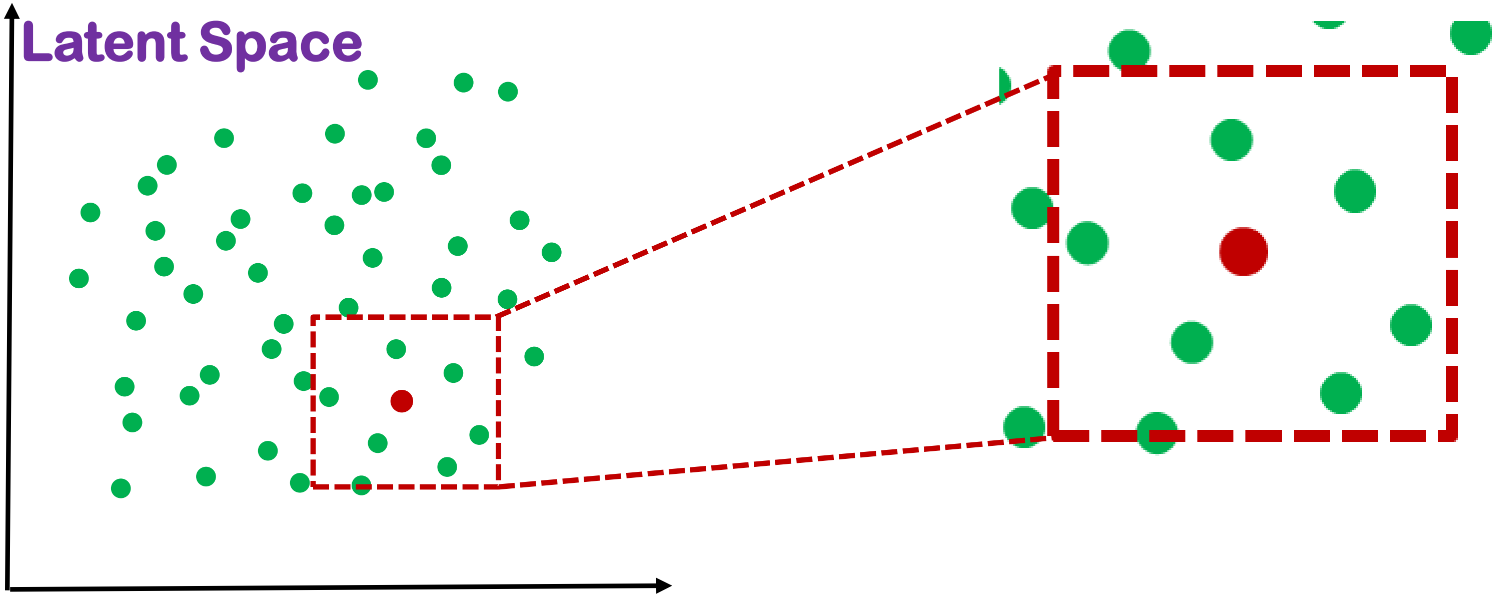




PCA

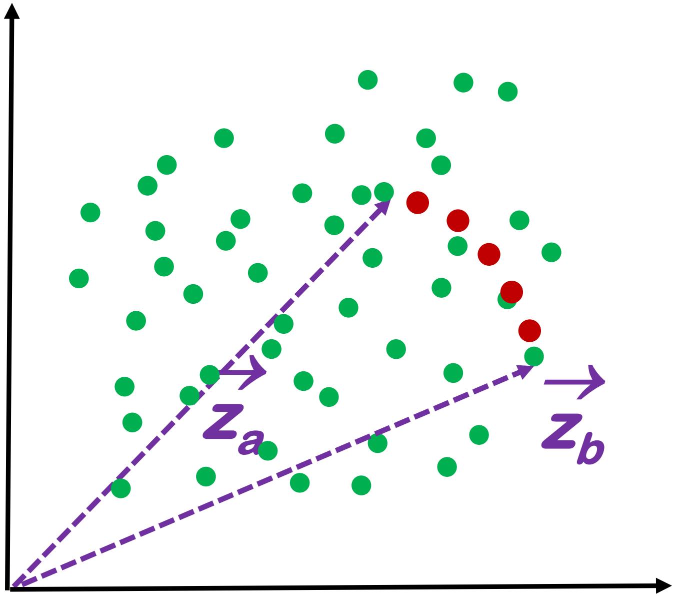
Artifical Neuronal Network (ANN)

GENERATIVE models: how we *navigate* in the *latent space*?



Navigate: it means modifying the latent vector (the famous *z*, like [0.17, -0.42, ..., 1.23]) and seeing what molecules come out when you pass it to the decoder. Here some examples:

GENERATIVE models: how we *navigate* in the *latent space*?



Molecule A → z_a

Molecule B → z_b

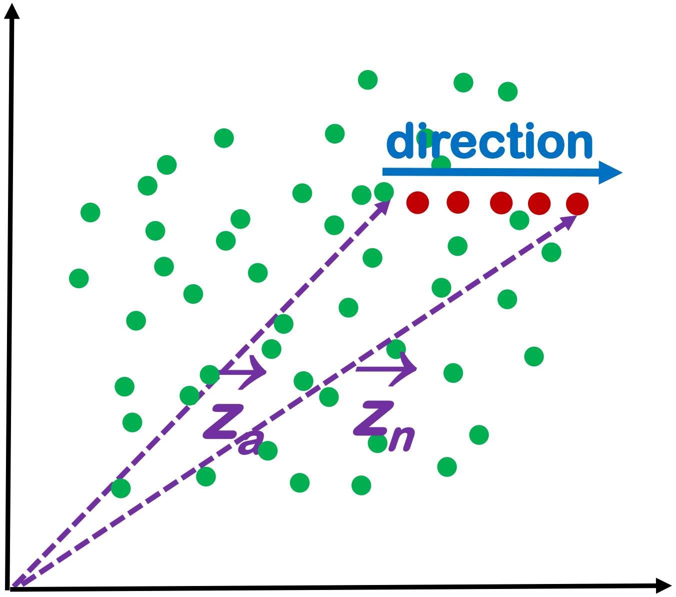
Molecule n → z_n

$$(1 - t) * z_a + t * z_b$$

Interpolation: for different values of t, you get new molecules that "*lie in the middle*" between A and B.

This is useful if you want to see which molecule is in the middle between two other molecules!

GENERATIVE models: how we *navigate* in the *latent space*?



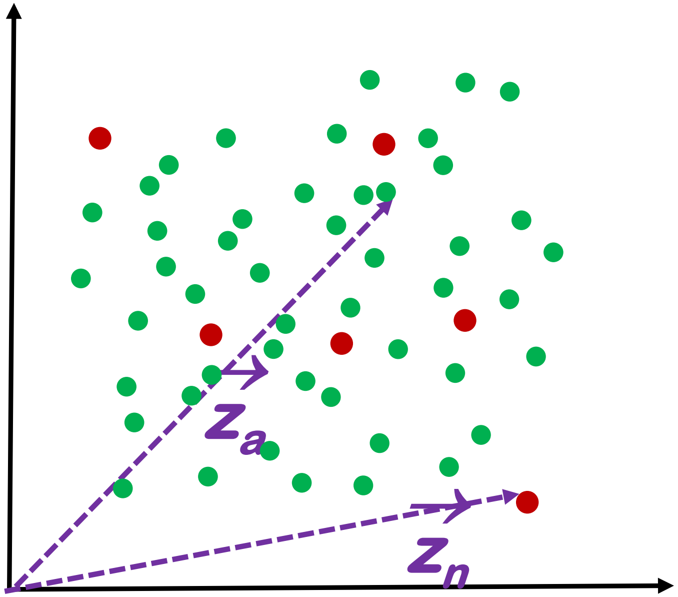
Molecule A → z_a

Molecule n → z_n

$$z_n = z_a + \alpha * \text{direction}$$

Direct Navigation: If you know that a certain direction corresponds to increase lipophilicity or increase basicity or increase BBB permeability, **it can take small steps (α) in one direction of the latent vector, z:**

GENERATIVE models: how we *navigate* in the *latent space*?



Molecule A → z_a

Molecule n → z_n

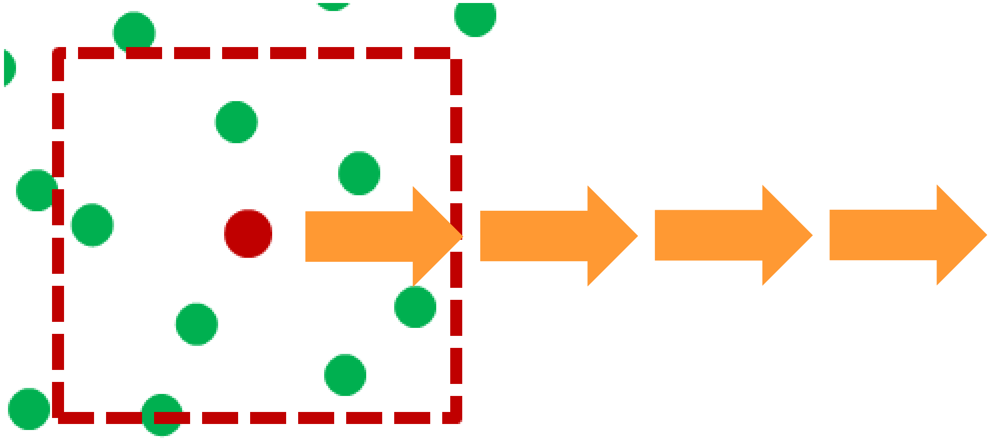
z_n = random z_a

Random Sampling: a random number generator can be used to apply to each component of the latent vector, z. This method is used to generate new molecules with respect to those that have mapped the latent space

GENERATIVE models: what is an decoder

credits: https://viso.ai/deep-learning/generative-ai/

Decoders: Bringing Latent Representations to Life



CC(=O)

GENERATIVE models: what is an decoder

credits: https://viso.ai/deep-learning/generative-ai/

Decoders: Bringing Latent Representations to Life

Once the data has been encoded into a latent representation, decoders take the stage. Decoders, also known as generators, are responsible for translating these latent vectors back into meaningful output data. They reconstruct the data based on the learned patterns and relationships from the encoded space, resulting in outputs that often exhibit remarkable creativity.

Decoders are crucial in applications like image generation, text synthesis, music composition and SMILES. They enable the model to generate content that is both novel and coherent, producing outputs that align with the characteristics of the original input data.

GENERATIVE models: what is an decoder

credits: https://viso.ai/deep-learning/generative-ai/

SMILES Decoders workflow:

The decoder receives the new *latent vector z* as the initial state of the decoding process. Usually the deconding process starts with a special token like "START":

Latent Vector, z: "START" [0.21, -0.19, ..., 0,97]

Translate the first embedded point: [0.2, 0.8] and read its token ID [101]

Translate the second embedded point: [0.2, 0.8] and read its token ID [101]

...

Translate the last embedded point: [0.4, 0.9] dnd read its token ID [250]

Usually the deconding process stops with a special token like "END".

GENERATIVE models: what is an decoder

credits: https://viso.ai/deep-learning/generative-ai/

Converting token ID's in tokens:

[101, 101, 205, 306, 155, 250] → ["C", "C", "(", "=", "O“, “)”]

Translating token in SMILES:

"CC(=O)"

GENERATIVE models: what is an dencoder

credits: https://viso.ai/deep-learning/generative-ai/

Converting token ID’s in tokens:

[101, 101, 205, 306, 155, 250] → ["C", "C", "(", "=", "O“, “)”]

Translating token in SMILES:

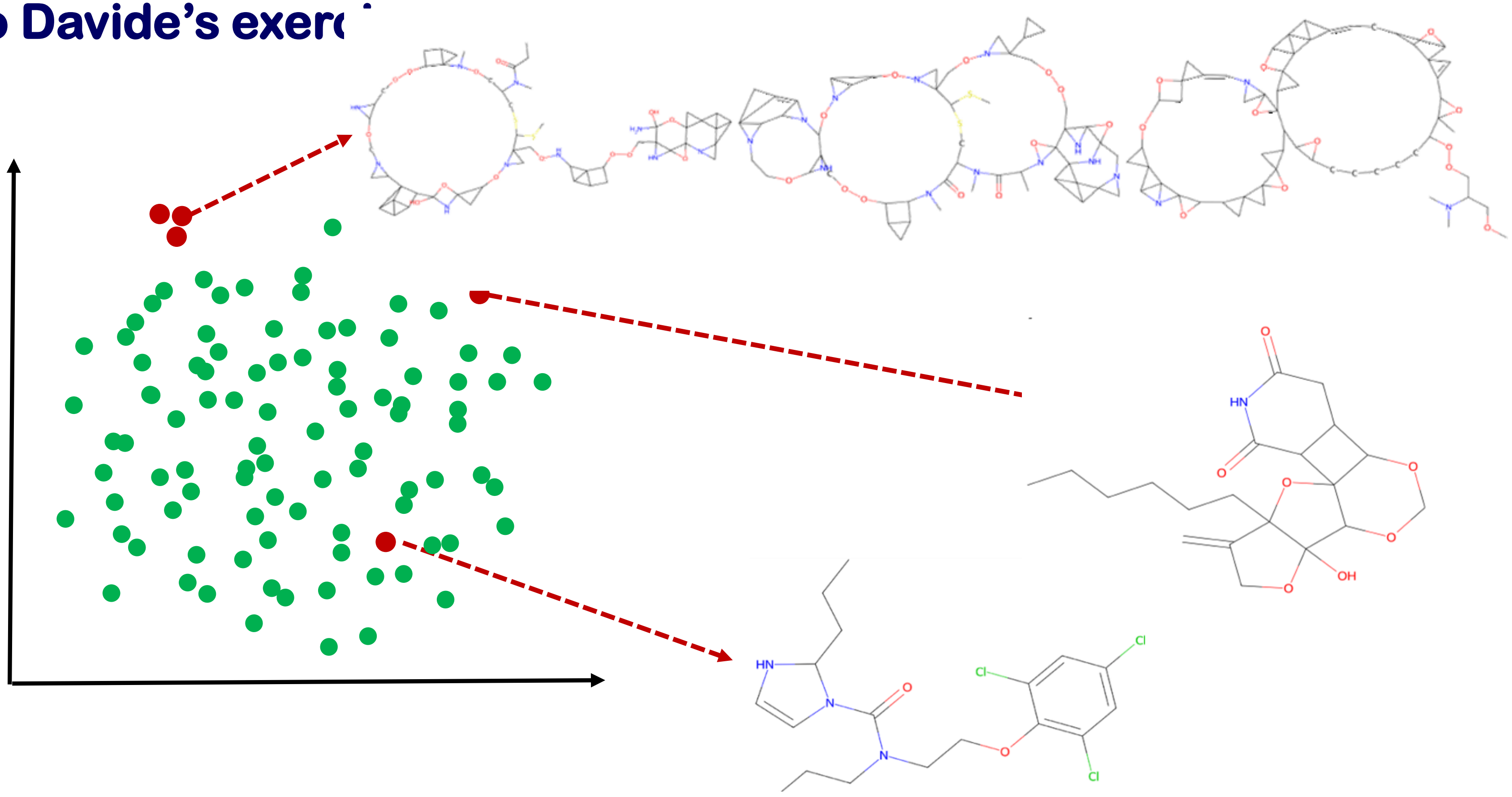
"CC(=O)"

CC(=O)

could this SMILES correspond to a real chemical compound?

We are ready to GENERATE: lesson 0 – a medchem example:

Back to Davide's exerc

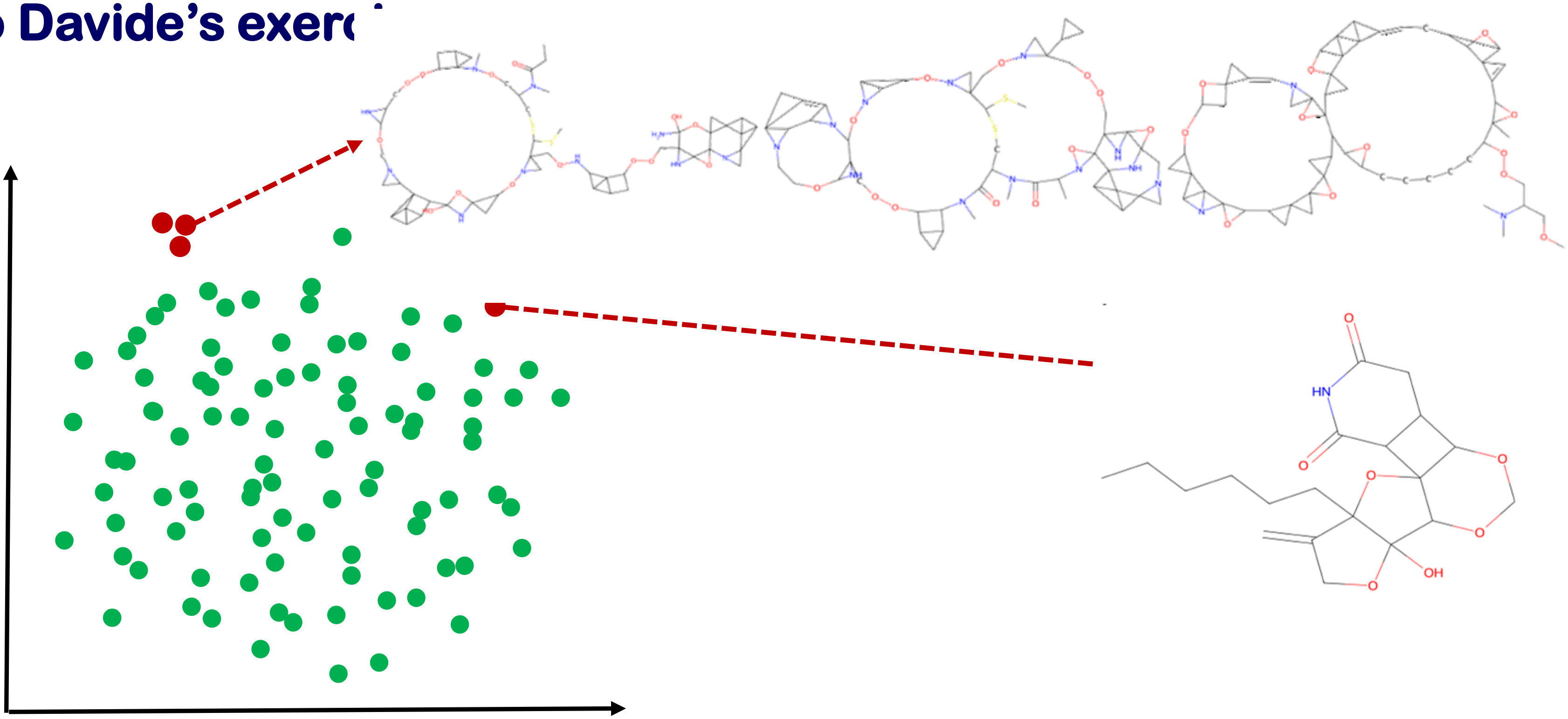


could this molecule active against melanoma?

Rigoni D, et al. Int J Mol Sci. 2024 Jun 4;25(11):6186.

We are ready to GENERATE: lesson 0 – a medchem example:

Back to Davide's exerc



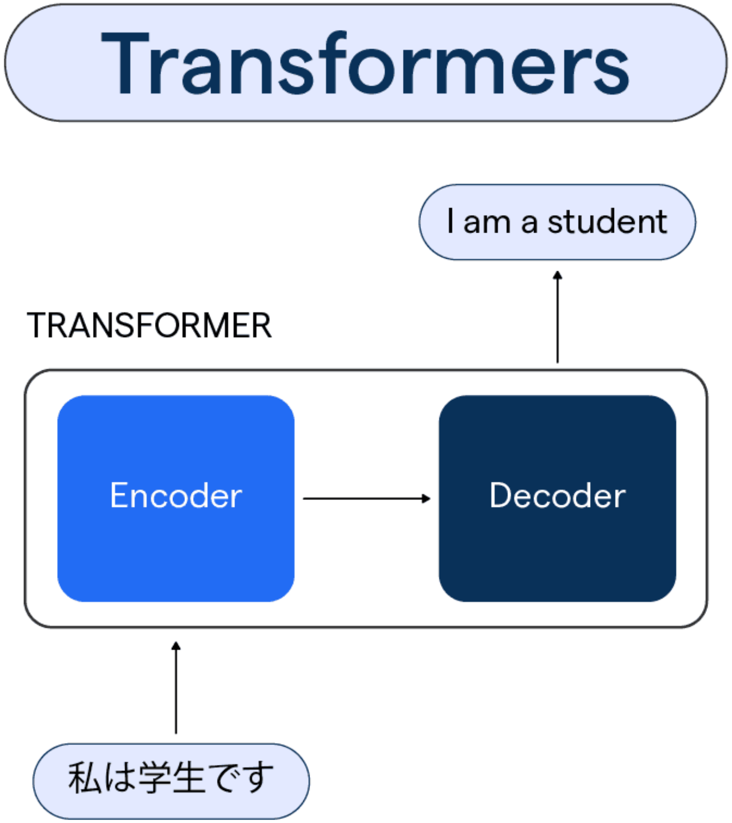
Before to be active against melanoma, could these molecules exit in the real life?

Rigoni D, et al. Int J Mol Sci. 2024 Jun 4;25(11):6186.

GENERATIVE models: what is an TRASFORMER

… even if famous, not this!!!

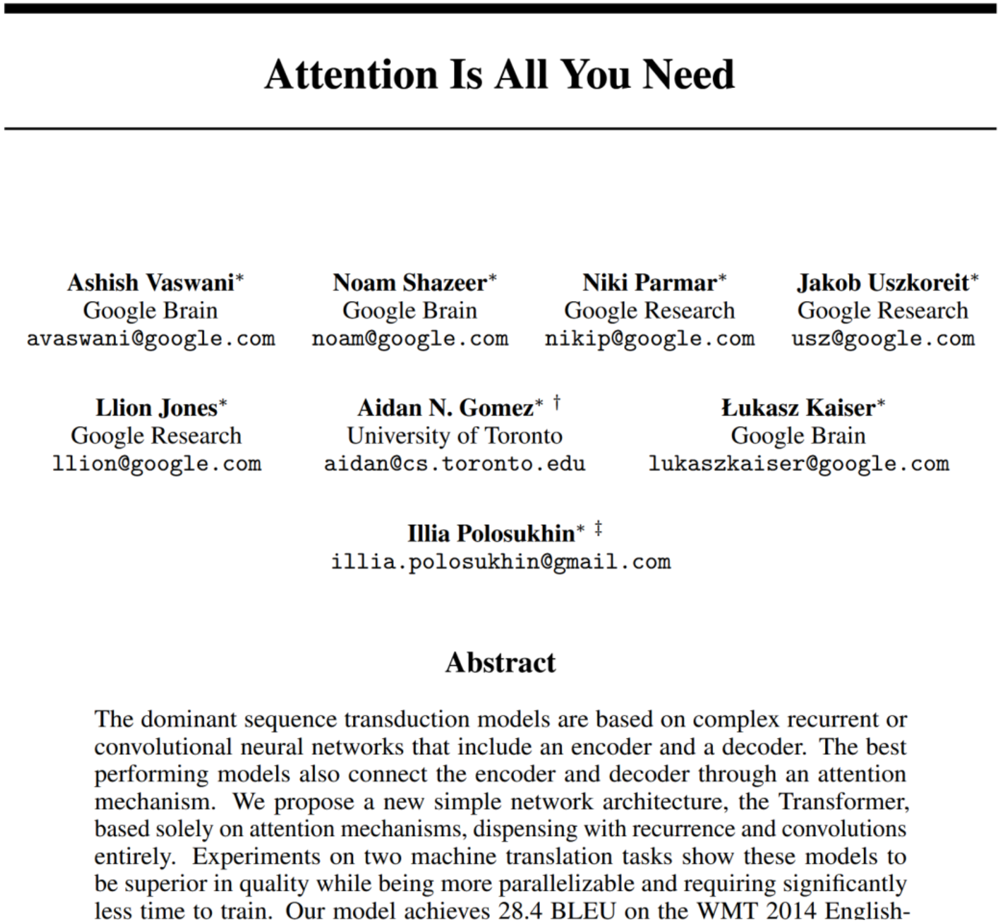
GENERATIVE models: what is an TRASFORMER in NLP:



Transformers are models in NLP that take on language processing tasks with a novel approach. They focus on the context of words in a sentence, providing a more accurate understanding of the text compared to traditional methods.

credits: https://botpenguin.com/glossary/transformers

GENERATIVE models: what is an TRASFORMER in NLP:



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-

credits: Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N; Kaiser, Łukasz; Polosukhin, Illia (December 2017). "Attention is All you Need". In I. Guyon and U. Von Luxburg and S. Bengio and H. Wallach and R. Fergus and S. Vishwanathan and R. Garnett (ed.). 31st Conference on Neural Information Processing Systems (NIPS). Advances in Neural Information Processing Systems. Vol. 30. Curran Associates, Inc.

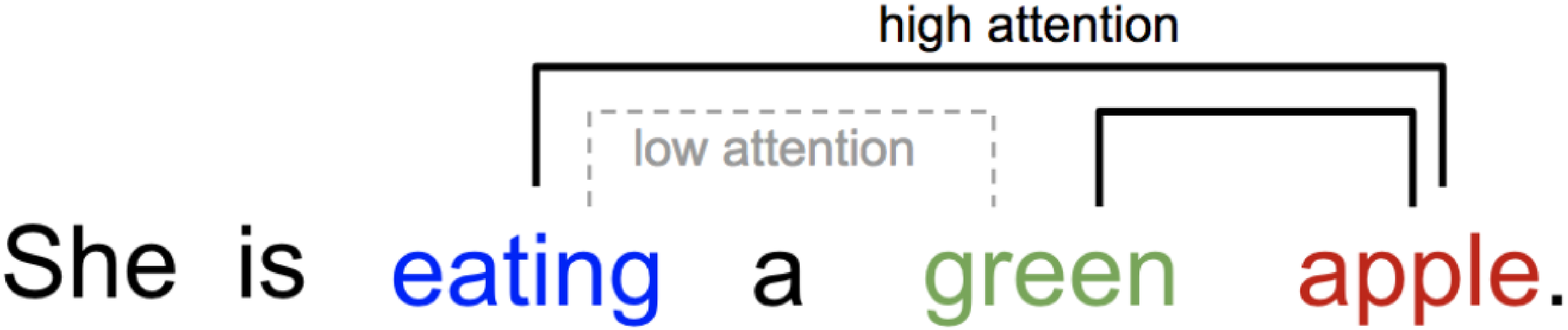
GENERATIVE models: what is an TRASFORMER in NLP:

The *attention mechanism* is a fundamental part of transformer models that allows them to weigh the importance of different tokens (words or symbols) in an input sequence when making predictions.

It assigns varying degrees of importance to different tokens to help the model focus on the most relevant portions of the data, like how certain words have more meaning in a sentence.

credits: https://botpenguin.com/glossary/transformers

GENERATIVE models: what is an TRASFORMER in NLP:



We can explain the relationship between words in one sentence or close context. When we see "eating", we expect to encounter a food word very soon. The color term describes the food, but probably not so much with "eating" directly.

credits: https://botpenguin.com/glossary/transformers

GENERATIVE models: what is an TRASFORMER in NLP:

How *attention mechanism* works: QUERY, KEYS and VALUES.

“Anthony Hopkins admired Michael Bay as a great director”

tokenization

“Anthony “ “Hopkins” “admired” “Michael” “Bay” “as” “a” “great director”

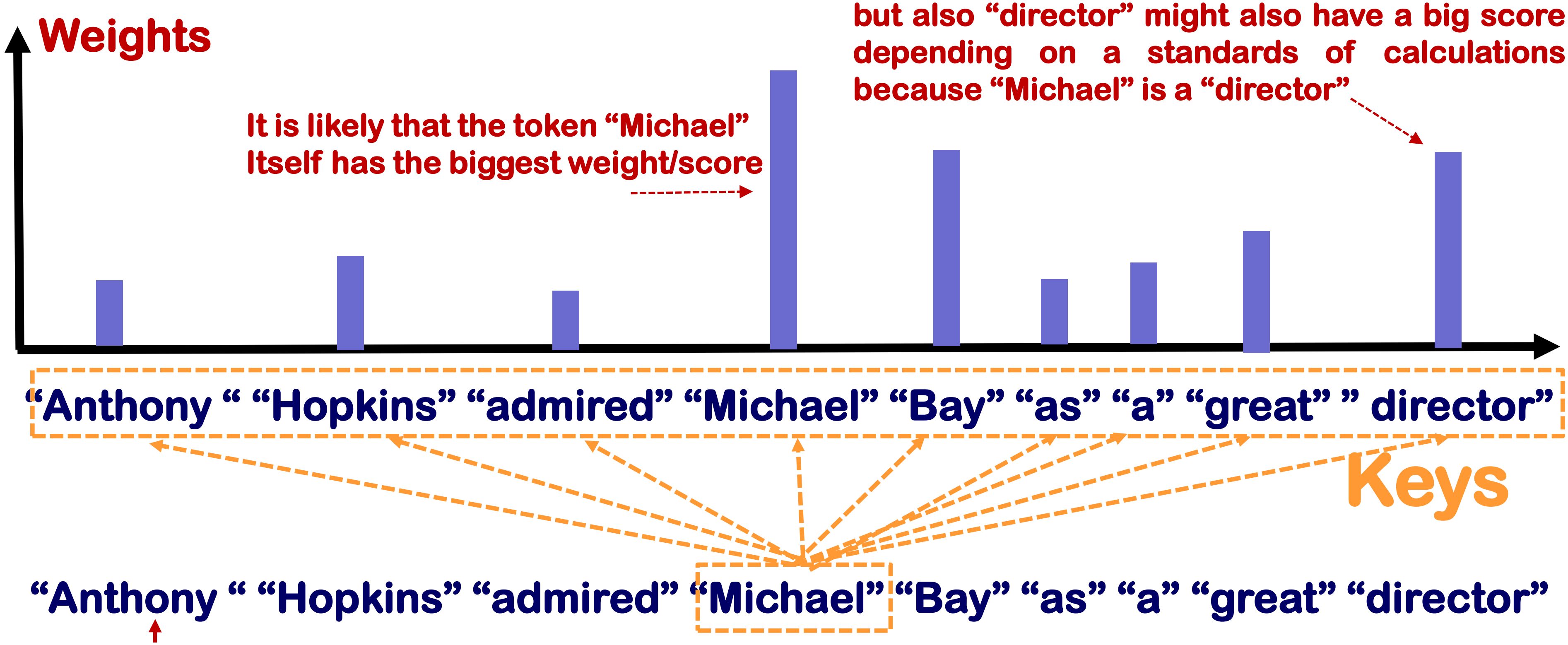
embedding

[0.5, 0.2, -1.4], [0.2, -0.2, 0..4], [-0.1, 0.2, -0.3], [0.7, 0.6, 1.2], [1.5, 0.2, 1.4], …, [-0.5, 0.7, 1.0]

Just an example of embedded vector

GENERATIVE models: what is an TRASFORMER in NLP:

Only as an example, we will use the tokens instead of the corresponding embedded vectors:



Weights

It is likely that the token “Michael” Itself has the biggest weight/score

but also “director” might also have a big score depending on a standards of calculations because “Michael” is a “director”

“Anthony “ “Hopkins” “admired” “Michael” “Bay” “as” “a” “great” ” director”

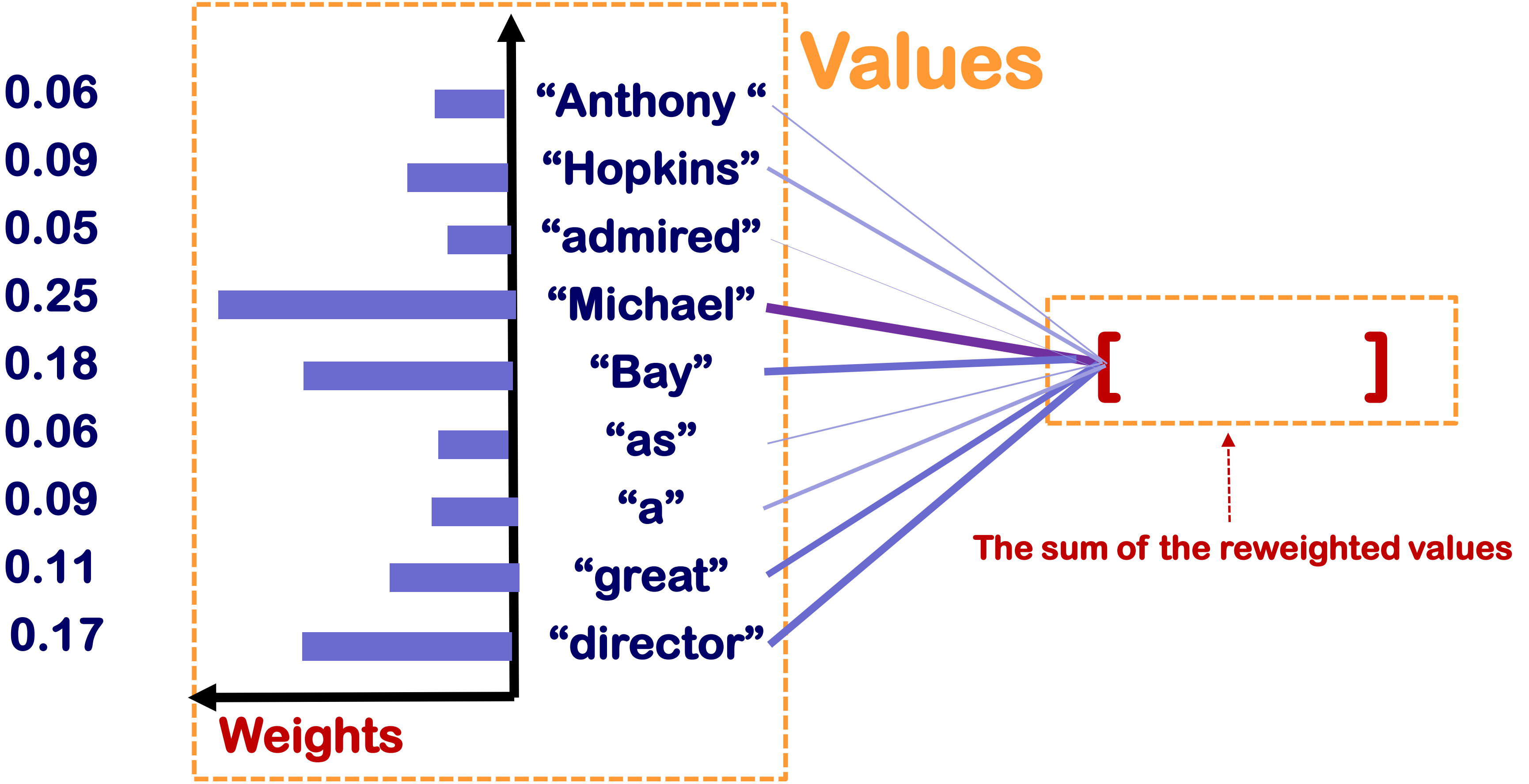
Keys

“Anthony “ “Hopkins” “admired” “Michael” “Bay” “as” “a” “great” “director”

Query

Just an example of embedded vector

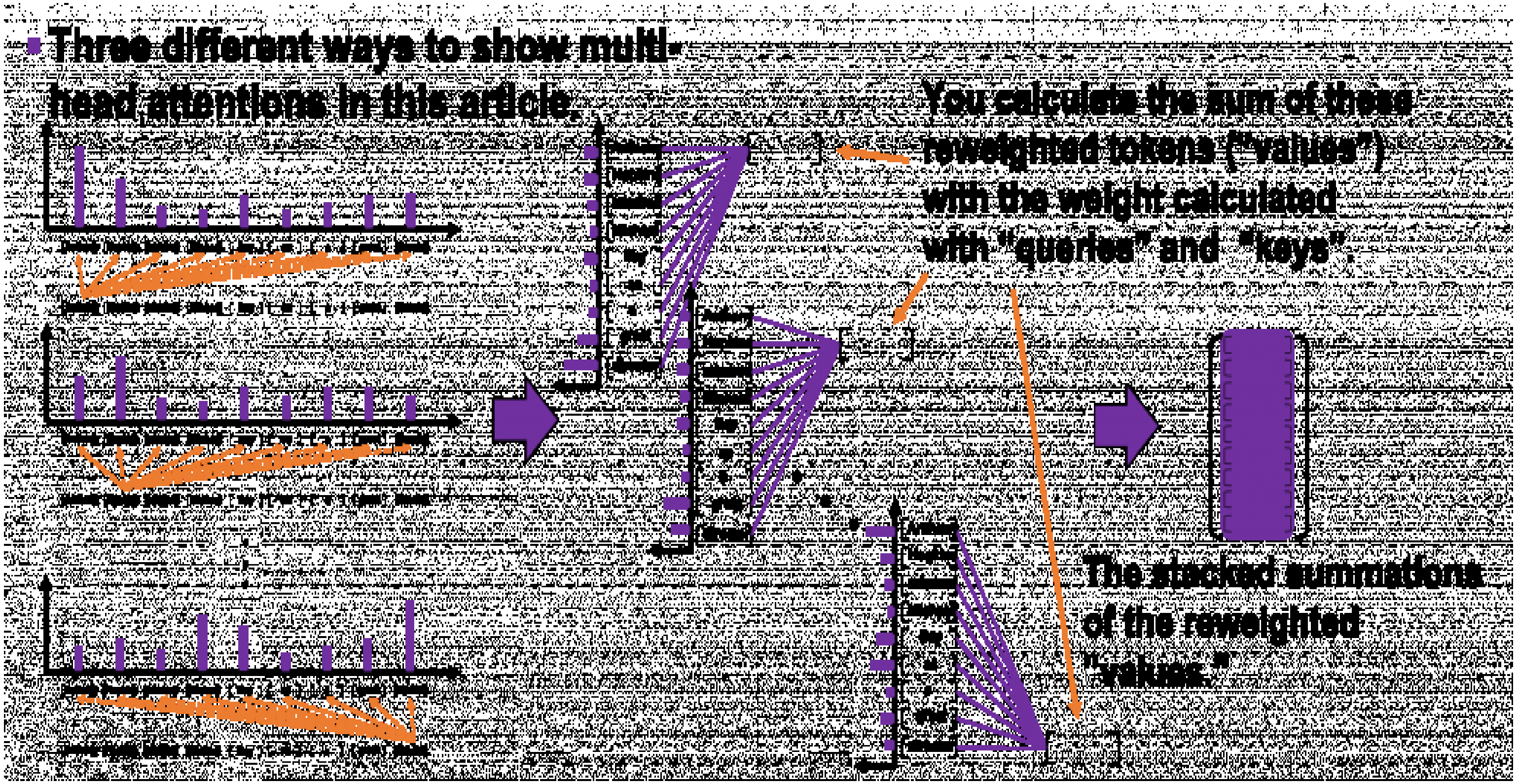
GENERATIVE models: what is an TRASFORMER in NLP:



In this case the sum of the reweighted token is: 0.06"Anthony” + 0.09"Hopkins” + 0.05"admired” + 0.25"Michael” + 0.18"Bay” + 0.06"as” + 0.09"a” + 0.11"great” 0.17"director”, and this sum is the what we e actually use.

GENERATIVE models: what is an TRASFORMER in NLP:

You repeat this process for all the "queries." As you can see in the figure below, you get summations of 9 pairs of reweighted "values" because you use every token of the input sentence "Anthony Hopkins admired Michael Bay as a great director." as a "query." You stack the sum of reweighted "values" like the matrix in purple in the figure below, and this is the output of a one head multi-head attention.



- Three different ways to show multi-head attentions in this article.

You calculate the sum of these reweighted tokens ("values") with the weight calculated with "queries" and "keys".

The stacked summations of the reweighted "values."

GENERATIVE models: what is an TRASFORMER in NLP:

How *attention mechanism* works with SMILES: QUERY, KEYS and VALUES.

“CC(=O)O”

tokenization

“C “ “C” “(” “=” “O” “)” “O”

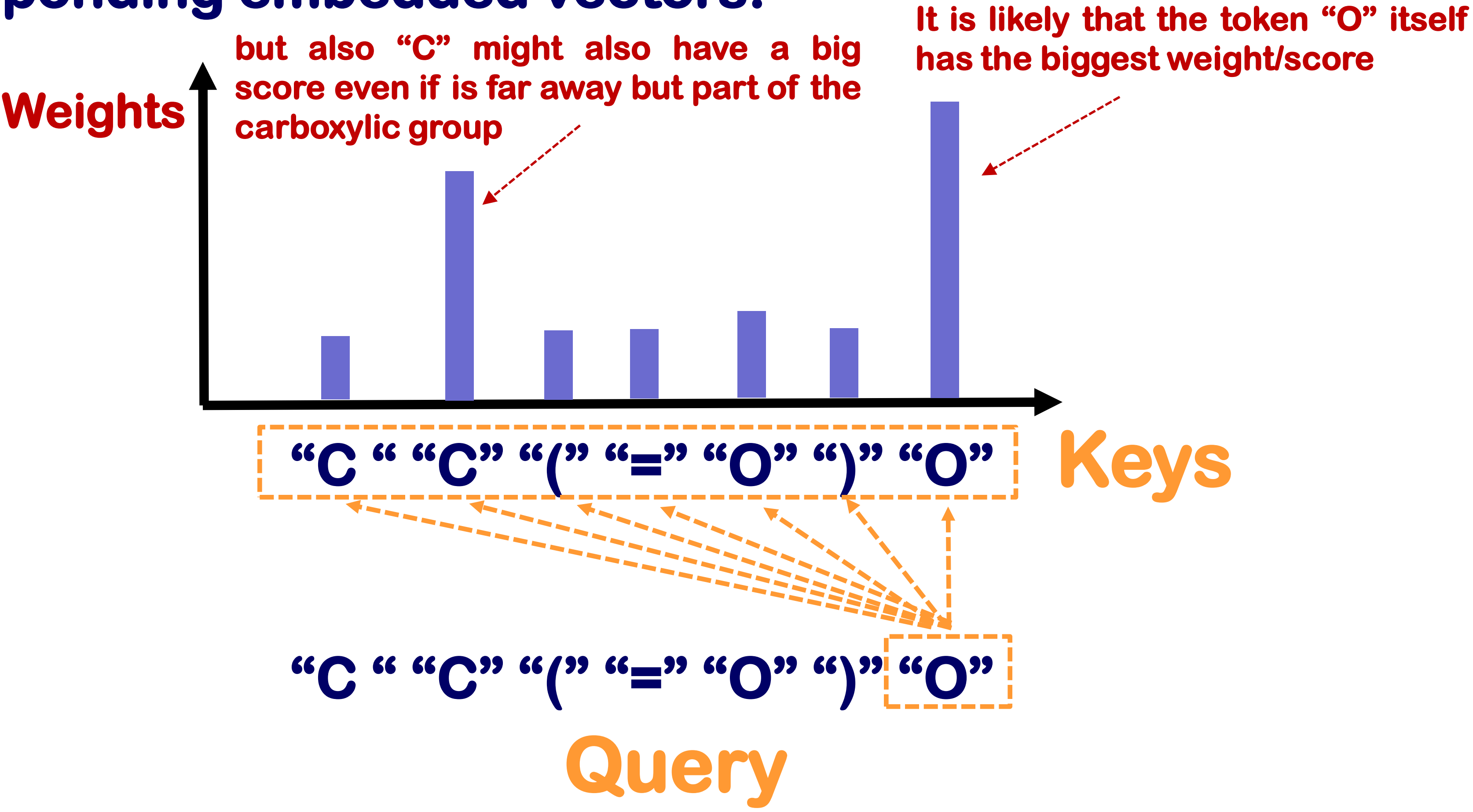
embedding

[0.5, 0.2, -1.4], [0.2, -0.2, 0..4], [-0.1, 0.2, -0.3], [0.7, 0.6, 1.2], [1.5, 0.2, 1.4], …, [-0.5, 0.7, 1.0]

Also in this case, ust an example of embedded vector

GENERATIVE models: what is an TRASFORMER in NLP:

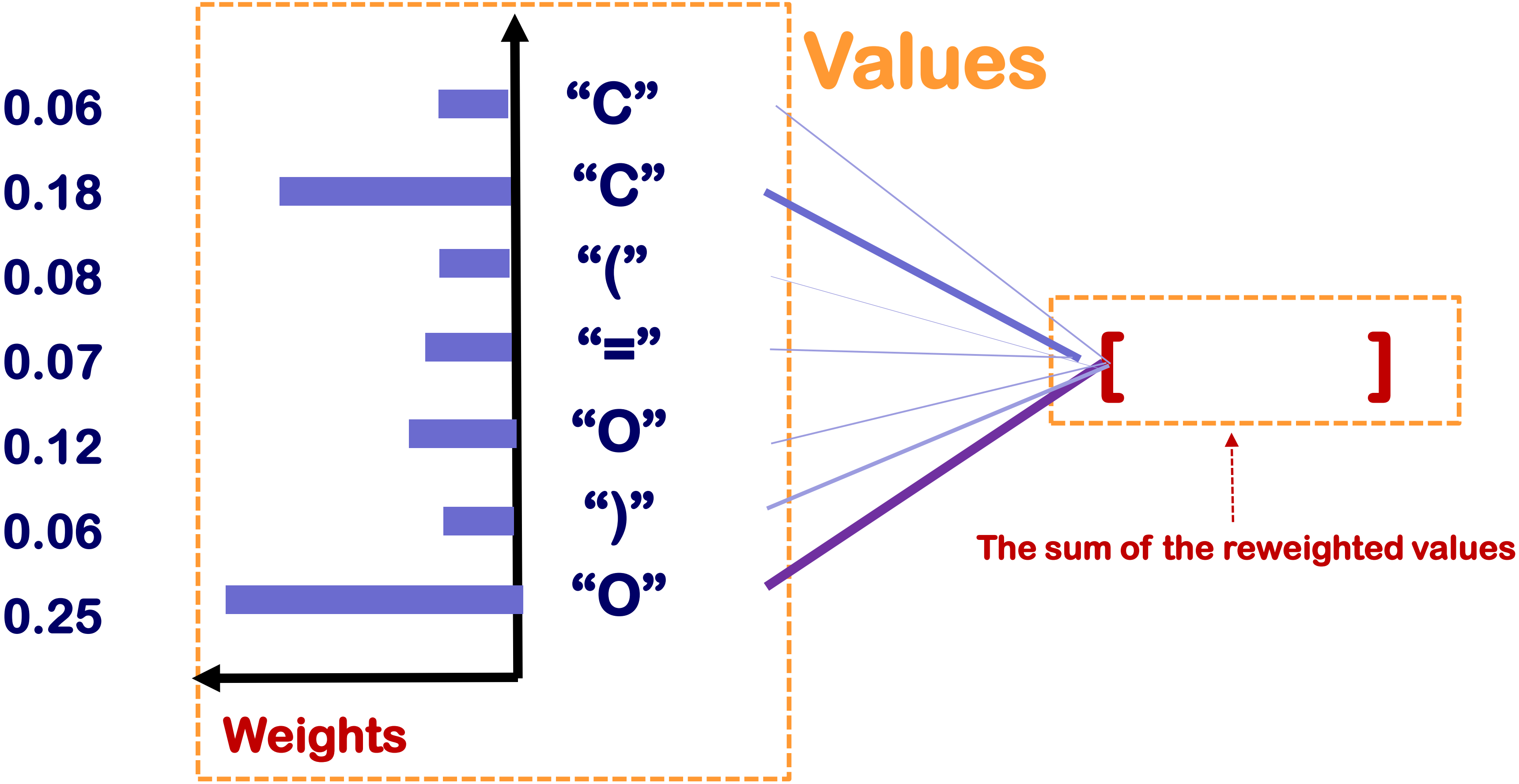
Only as an example, we will use the tokens instead of the corresponding embedded vectors:



but also “C” might also have a big score even if is far away but part of the carboxylic group

It is likely that the token “O” itself has the biggest weight/score

GENERATIVE models: what is an TRASFORMER in NLP:



Values

0.06 "C"

0.18 "C"

0.08 "("

0.07 "="

0.12 "O"

0.06 ")"

0.25 "O"

Weights

[ ]

The sum of the reweighted values

In this case the sum of the reweighted token is: 0.06"C" + 0.18"C" + 0.08"(" + 0.07"=" + 0.12"O" + 0.06")" + 0.25"O", and this sum is the what we e actually use.

Before to complete the story of Trasformers…