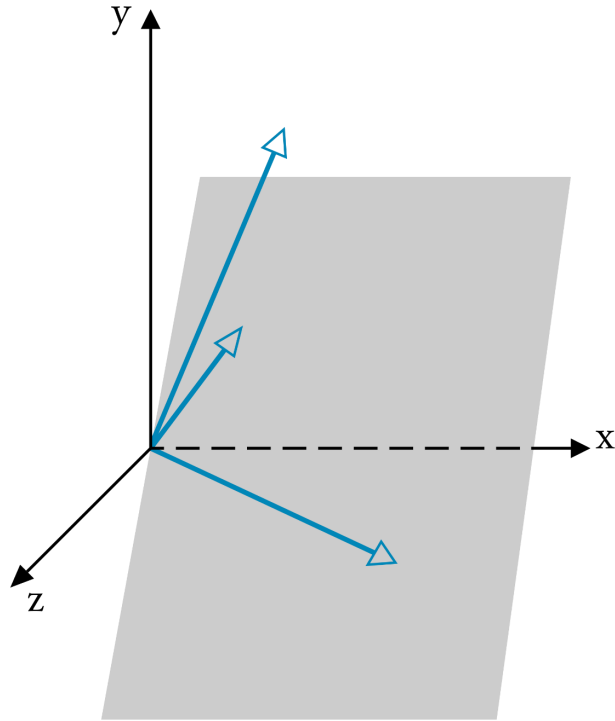


Word2vect and NCE loss

陳慶豐

Why using word vector?

- Trying to reduce dimension of word and make word with similar meaning clustered.



Distributional Hypothesis

- words that appear in the same contexts share semantic meaning.

Data preprocessing: Subsampling

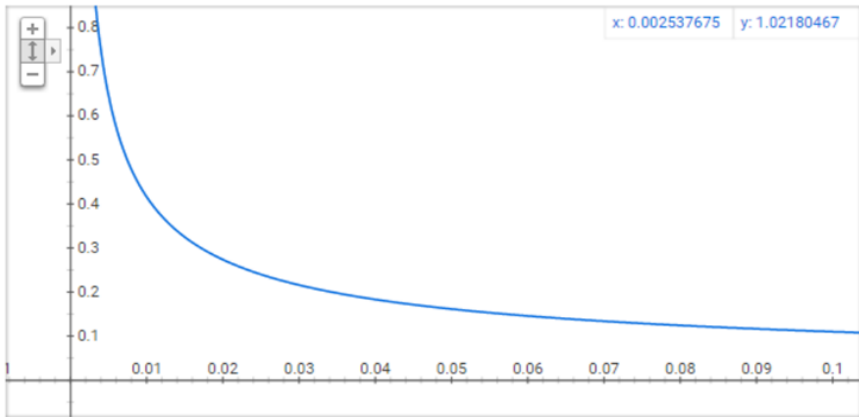
'sample' mean words are less likely to be kept.

$P(w_i)$ is the probability of *keeping* the word:

$$P(w_i) = \left(\sqrt{\frac{z(w_i)}{0.001}} + 1 \right) \cdot \frac{0.001}{z(w_i)}$$

You can plot this quickly in Google to see the shape.

Graph for $(\sqrt{x/0.001}+1)*0.001/x$

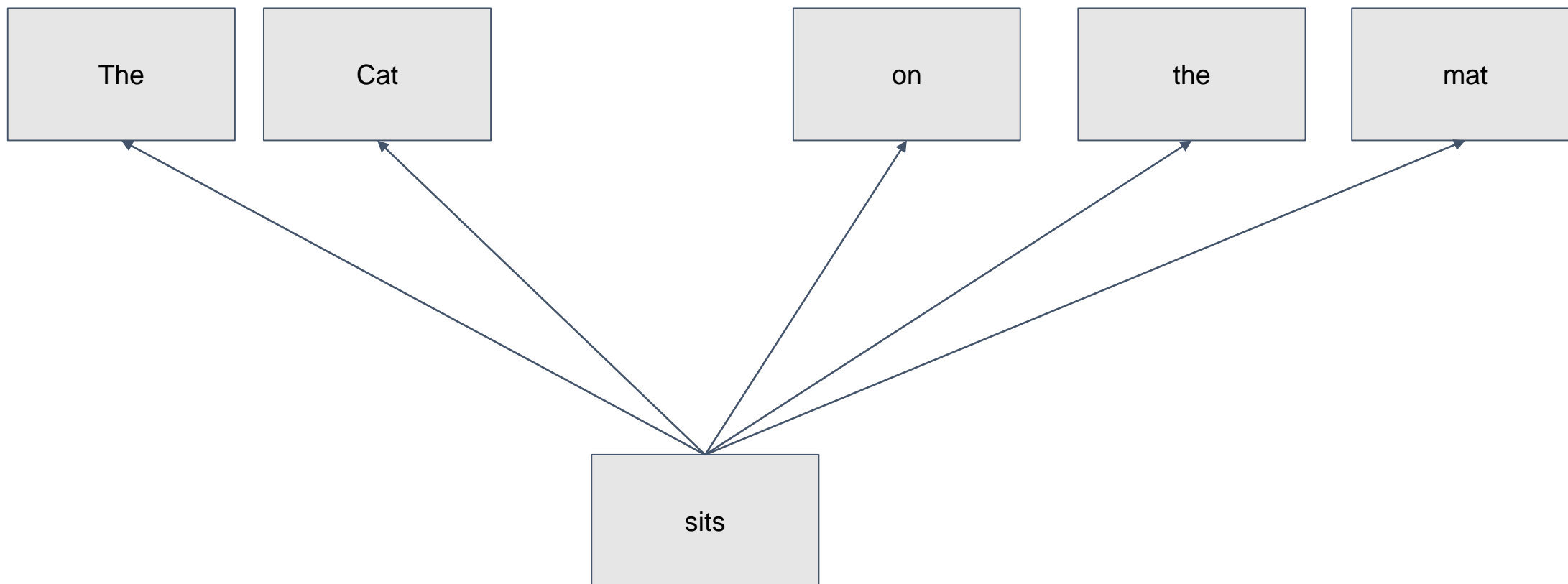


No single word should be a very large percentage of the corpus, so we want to look at pretty small values on the x-axis.

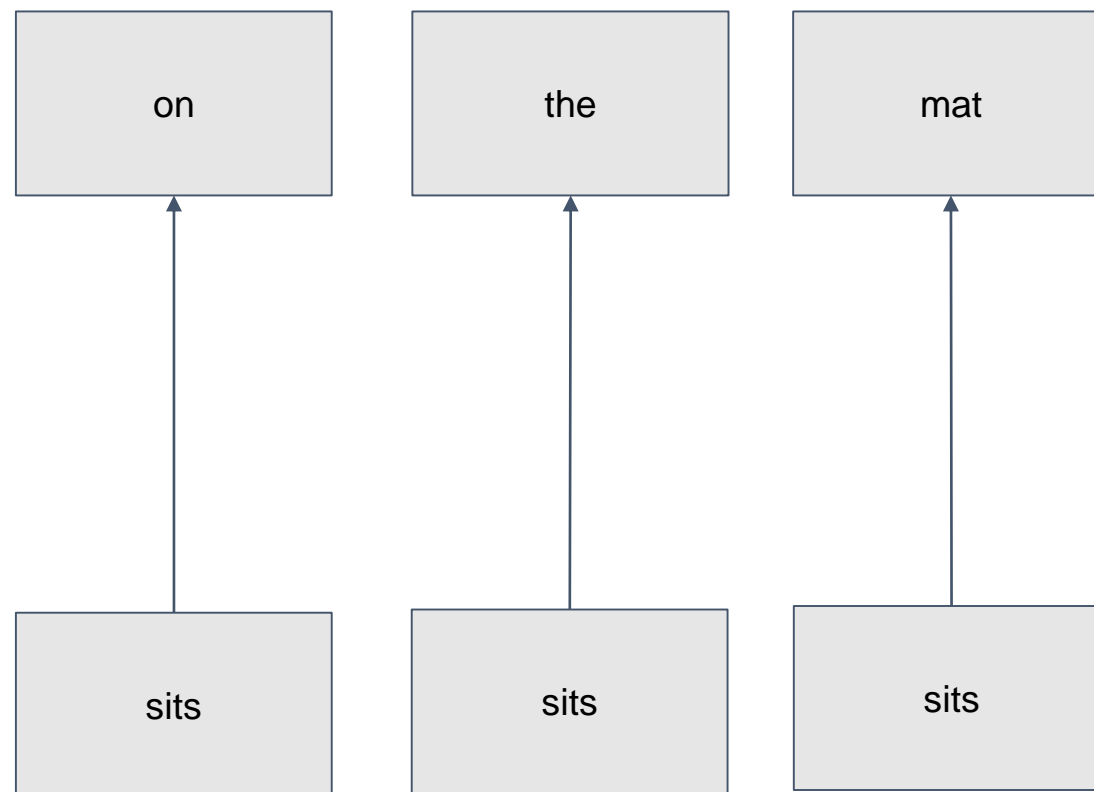
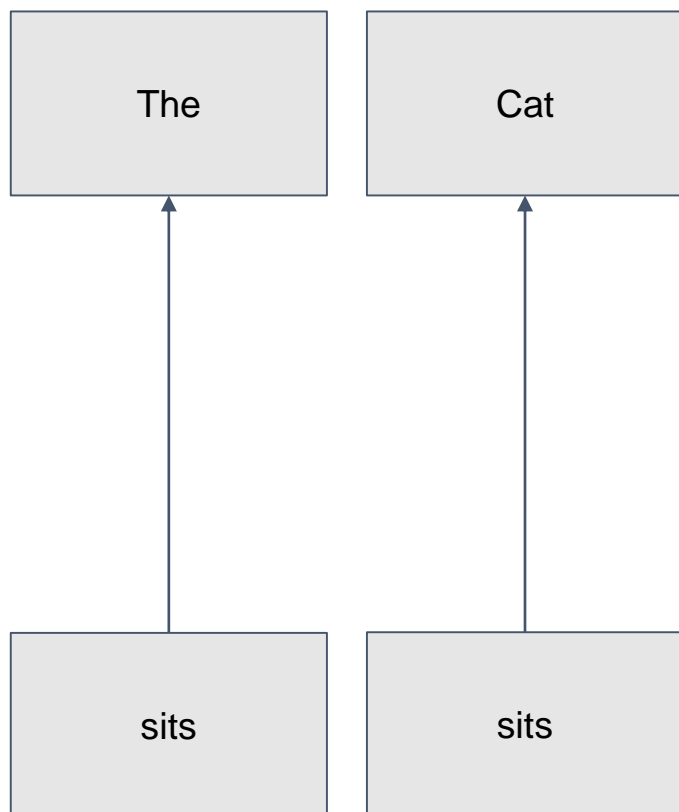
Skip-gram model



Skip-gram model

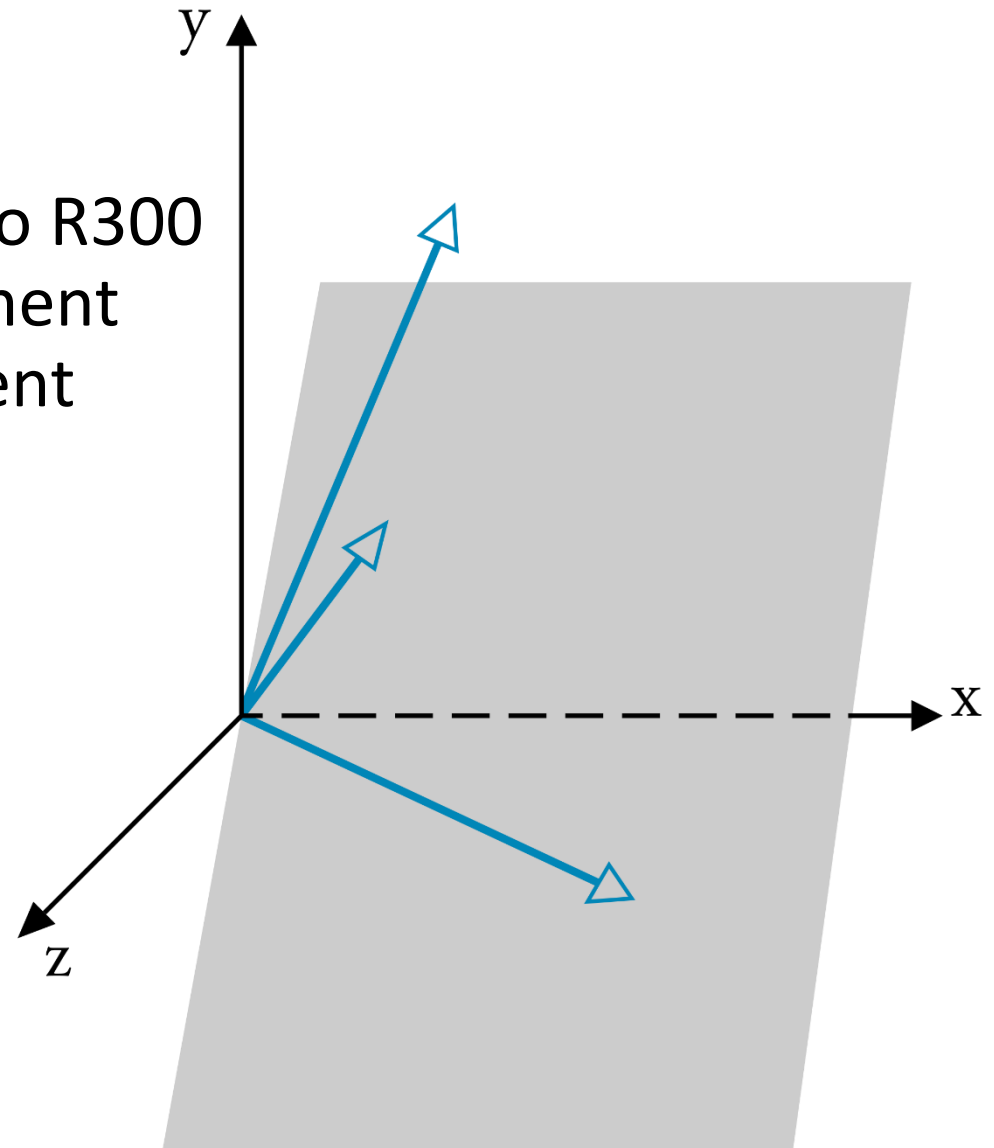


Skip-gram model



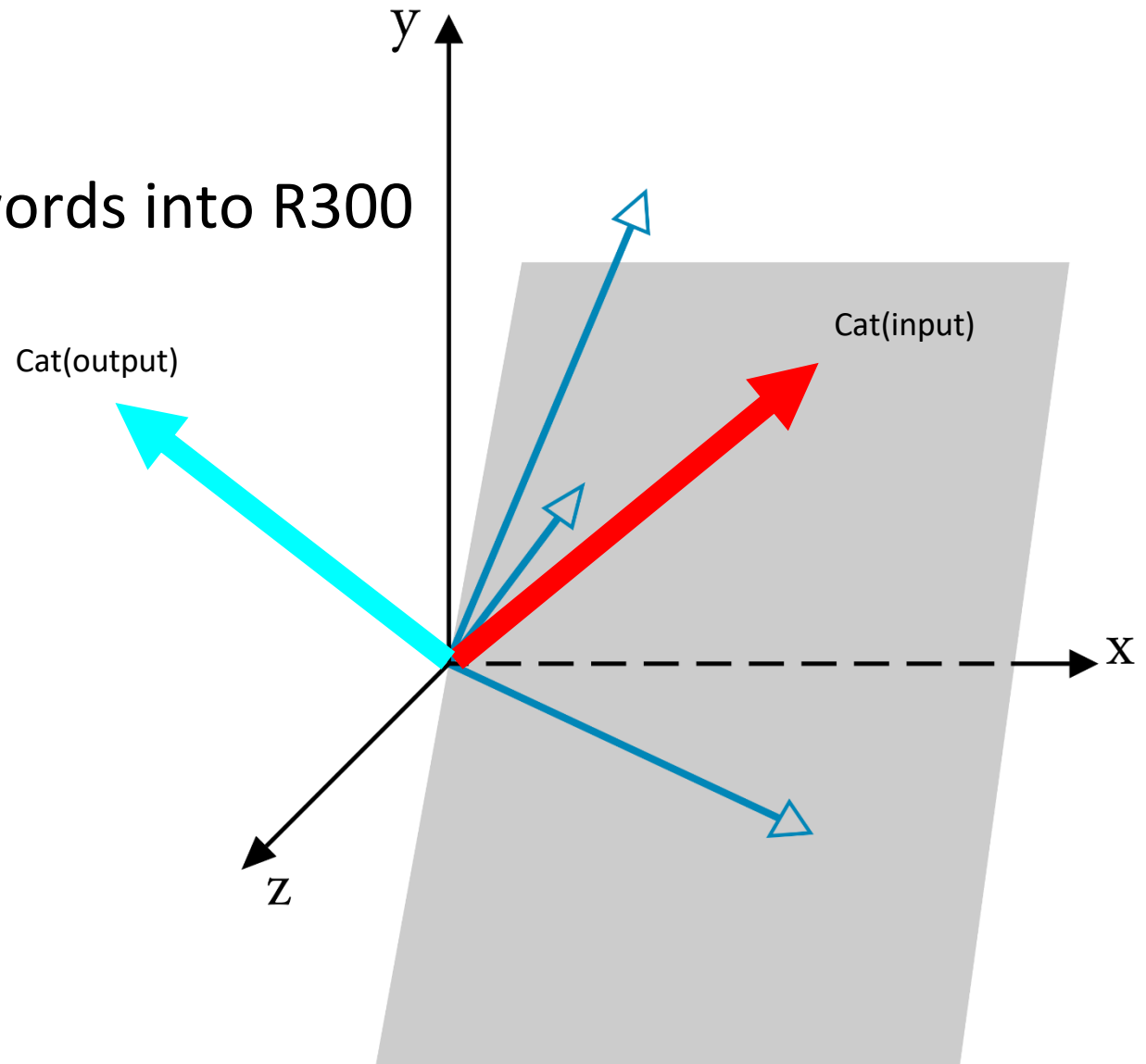
Skip-gram model

- Encode input and output words into R300
- So the first question is, why implement
- input and output words into different
- vectors?

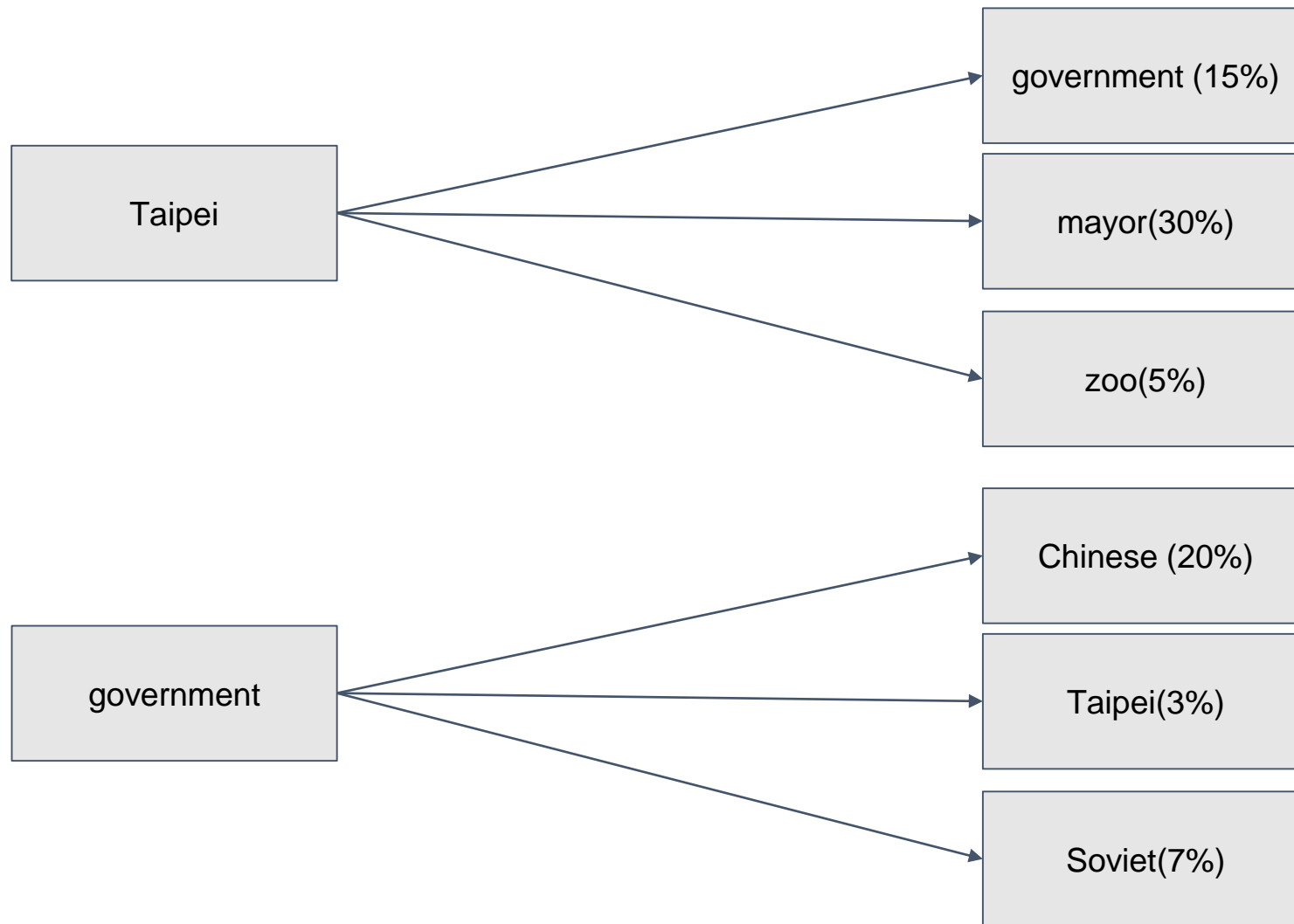


Skip-gram model

- Encode input and output words into R300



Skip-gram model



So intuitively we can set it as softmax function

$$p(w_O|w_I) = \frac{\exp\left(v'_{w_O}{}^\top v_{w_I}\right)}{\sum_{w=1}^W \exp\left(v'_w{}^\top v_{w_I}\right)}$$

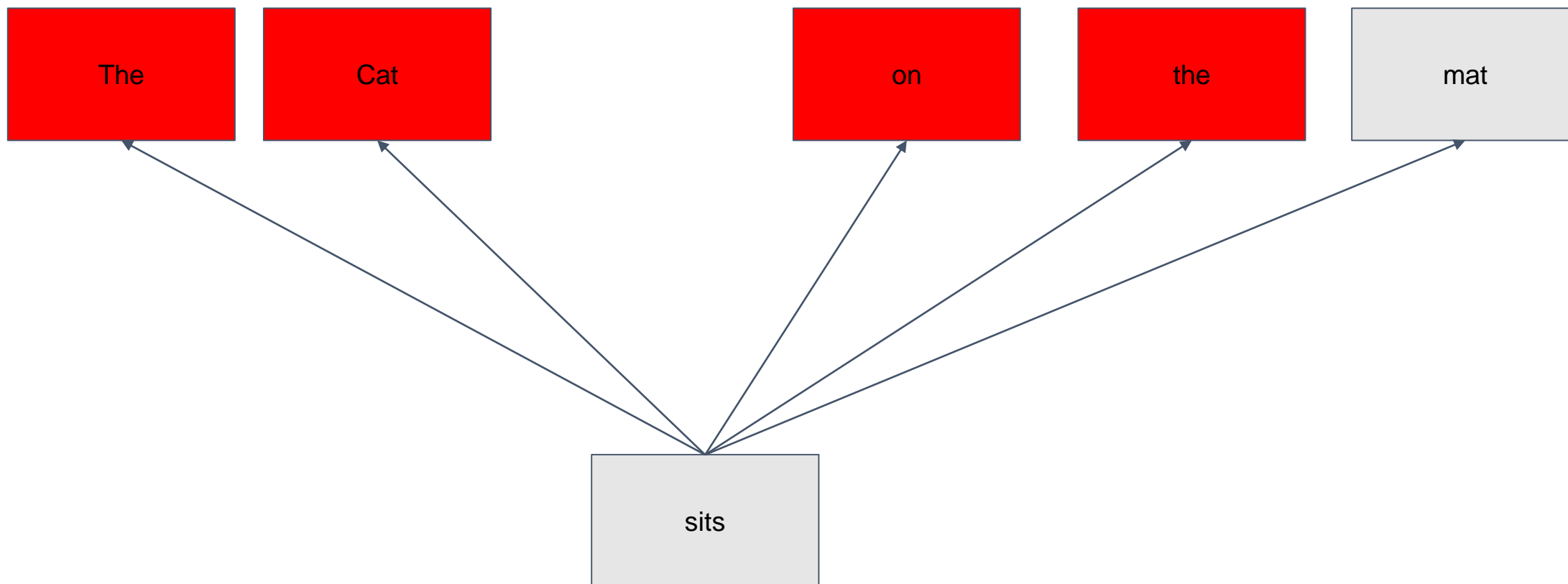
Goal: maximizing MLE

- our goal is to maximize the MLE:

$$\prod p(w_{input} | w_{output}) \quad (\text{一個強迫中獎的multinomial})$$

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t)$$

Window size: c ($c=2$)



Impractical!

- $\nabla \log p(w_O | w_I)$ is proportional to W ,

$$p(w_O | w_I) = \frac{\exp \left(v'_{w_O}{}^\top v_{w_I} \right)}{\sum_{w=1}^W \exp \left(v'_w{}^\top v_{w_I} \right)}$$

A little observation:

$$p(w_O|w_I) = \frac{\exp\left(v'_{w_O}{}^\top v_{w_I}\right)}{\sum_{w=1}^W \exp\left(v'_w{}^\top v_{w_I}\right)}$$

A huge stuff, set it to be C a constant

A little observation:

$$\frac{\exp(w'_{input} * w_{output})}{\sum_j \exp(w'_{input} * w_j)} = \frac{\exp(w'_{input} * w_{output})}{C + \exp(w'_{input} * w_{output})}$$
$$= \frac{1}{1 + C * \exp(-w'_{input} * w_{output})}$$

- Which become logistic regression between $v_o * v_i$ and a constant c

So from the observation:

- We can set

$$p(w_o|w_i) = \sigma(w_i' * w_o) * c(w_o)$$

to lower the dimension of softmax which is unnecessary needed.

But, Unnormalized probability function will face a problem:

$$MLE = \prod p(w_o | w_i)$$

$$= \prod \sigma(w_i' * w_o) * c(w_o)$$

Where MLE \rightarrow infinity when $c \rightarrow$ infinity, can't use MLE to find the

Noise-Contrastive Estimation of Unnormalized Statistical Models (NCE loss)

This is the most important tool in this paper!!!!!!!

Unnormalized probability function:

Suppose we have an unnormalized probability function

$$p(\bullet, \theta)$$

for example,

$$p(\bullet, \theta) = \bullet * \theta \text{ or}$$

$$p(\bullet, \theta) = \sigma(w_i' * w_o)$$

Normalized unnormalized probability function:

Suppose we have an unnormalized probability function

$$p(\bullet, \theta)$$

$$\text{Set } \frac{1}{\int p(\bullet, \theta)} = c(\theta) , \text{ then } p(\bullet, \theta) * c$$

is normalized

But

1.

$$\int p(\bullet, \theta) \quad \int p(\cdot, \theta) = \sum_{j=1}^V \sigma(Vj * Vi) \text{ in our case.}$$

is usually not easy to compute in high dimension space.

2.

$$\text{MLE} = \prod p(\bullet, \theta) * c(\theta)$$

which stands for we can't use gradient descent to find the actual theta.

Goal of NCE loss:

Try to find a way to train a unnormalized probability function

$$p(\bullet, \theta) * c(\theta)$$

and get theta and c simultaneously.

Idea of NCE loss

- Now we have a probability function

$$p(\cdot, \theta) * c(\theta)$$

which we want to train.

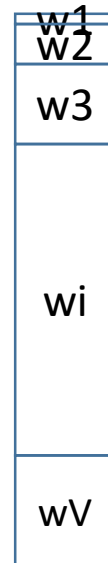
And given a noise distribution $p_n(\cdot)$

we draw k sample from $p(\cdot, \theta) * c(\theta)$ and k samples form $p_n(\cdot)$

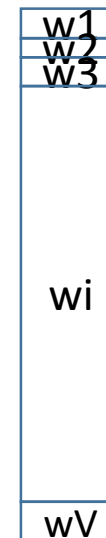
In a more understandable picture:

- Given w_i : we have a distribution of $p(x;w_i)$ and $p_n(x)$, suppose p_n is uniform.
- And after collecting k data points from $p(x;w_i)$ we collect k data points from noise too. (Total $2k$ datas)

$p(x;w_i)$ distribution



$p_n(x)$ distribution



In a more understandable picture:

- So after collecting those data, we have: (Given input: w_i , output: w_1)

- $P(C = 1) = \frac{T_d}{T_d + T_n} = (w_{11}) / (w_{11} + w_{12})$ which is an estimator of

- $p(w_1 | w_i) / (p(w_1 | w_i) + p_n(w_1))$

w11	w12
w21	w22
w31	w32
wi1	wi2
wV1	wV2

So now, we turn it into a binomial model

$$p(x \text{ is from } p(\bullet, \theta))$$

$$= \frac{p(\bullet, \theta)}{p(\bullet, \theta) + pn(\bullet)} \quad \text{-----} \rightarrow \quad \frac{w_{11}}{w_{11} + w_{12}}$$

$$p(x \text{ is from } pn(\bullet))$$

$$= \frac{pn(\bullet)}{p(\bullet, \theta) + pn(\bullet)} \quad \text{-----} \rightarrow \quad \frac{w_{12}}{w_{11} + w_{12}}$$

w11	w12
w21	w22
w31	w32
wi1	wi2
wV1	wV2

Good news! Now we can do MLE already!

$$\begin{aligned} MLE &= \prod p(x_i \text{ is from } p(\cdot, \theta))^{x_i} * p(x_i \text{ is from } pn(\cdot))^{(1-x_i)} \\ &= \prod \left(\frac{p(\cdot, \theta) * c(\theta)}{p(\cdot, \theta) * c(\theta) + pn(\cdot)} \right)^{x_i} * \left(\frac{pn(\cdot)}{p(\cdot, \theta) * c(\theta) + pn(\cdot)} \right)^{(1-x_i)} \end{aligned}$$

$$l = \sum x_i * \left(\frac{p(\cdot, \theta) * c(\theta)}{p(\cdot, \theta) * c(\theta) + pn(\cdot)} \right) + (1 - x_i) * \left(\frac{pn(\cdot)}{p(\cdot, \theta) * c(\theta) + pn(\cdot)} \right)$$

MLE result in:

$$\begin{aligned} & p(x \text{ is from } p(\bullet, \theta)) \\ &= \frac{p(\bullet, \theta)}{p(\bullet, \theta) + pn(\bullet)} \quad \text{-----} > \frac{w_{11}}{w_{11} + w_{12}} \end{aligned}$$

$$\begin{aligned} & p(x \text{ is from } pn(\bullet)) \\ &= \frac{pn(\bullet)}{p(\bullet, \theta) + pn(\bullet)} \quad \text{-----} > \frac{w_{12}}{w_{11} + w_{12}} \end{aligned}$$

MLE result in:

$$\textit{Because } \text{pn}(\cdot) = \frac{w_{12}}{k}$$

$$\therefore p(\cdot, \theta) \dashrightarrow \frac{w_{11}}{k}$$

Conclusion on NCE loss

$$p(\bullet, \theta) * c(\theta)$$

$$p(w_o|w_i) = \sigma(w_i' * w_o) * c(w_o)$$

Conclusion on word2vect

1. Greatly reduce the training time through NCE loss and NCE loss liked NGE loss (“We successfully trained models on several orders of magnitude more data than the previously published models, thanks to the computationally efficient model architecture.”)
2. These trained word vectors have additive property i.e.
American + idol = Madonna
3. Subsampling greatly improve the quality of word vector representative especially for those rare words.

Additive property explanation

$$V_{\text{russia}} + V_{\text{river}} = V_{\text{vodka river}}$$

$$P(\text{Russia} | \text{vodka river}) * p(\text{river} | \text{vodka river}) =$$
$$p(\text{vodka river} | \text{vodka river})^2$$

=

$$e^{(v_3 * v_1 + v_3 * v_2) / s^2} = e^{(v_3^2) / s^2}$$

So

$$v_3 * v_1 + v_3 * v_2 = v_3^2$$

$$V_1 + V_2 = V_3$$

中文實作

Wiki中文



中文斷詞系統

中文斷詞系統

[\[授權辦法\]](#)

相關系統：[輿情分析](#)

[實體辨識](#)

[中文詞彙特性速描系統](#)

[斷詞系統](#)

[剖析系統](#)

[詞首詞尾](#)

[平衡語料庫](#)

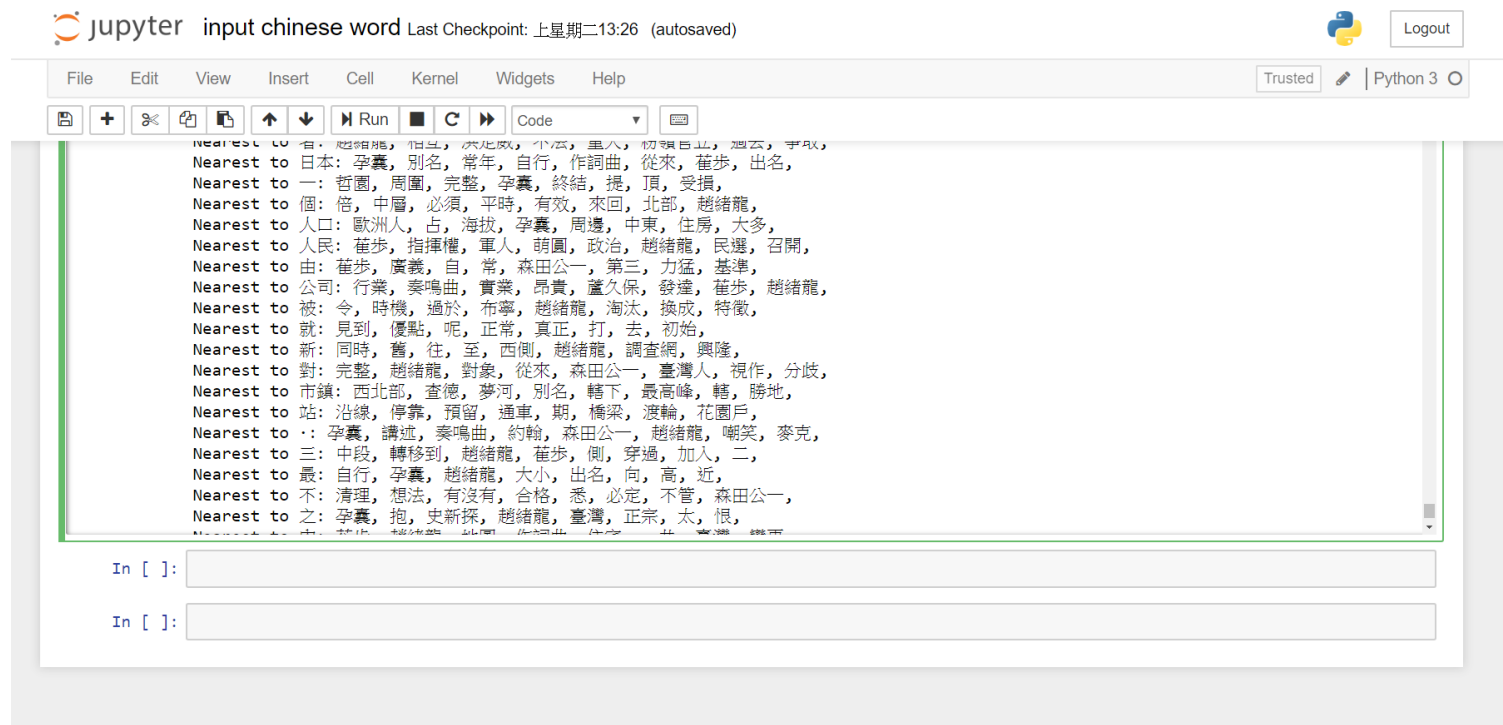
[廣義知網](#)

➔ [簡介](#)

➔ [未知詞擷取做法](#)

未知詞偵測 : 清華大學()(Nc) 資工()(Na) 系(?)(Nc) 在()(P) 資工()(Na) 館(?)(Nc)
複合詞 : 清華大學()(Nc) 資工系()(Nc) 在()(P) 資工()(Na) 館(?)(Nc)

Send it into word2vect



Without subsampling 跟只有做一次subsampling 都無法收斂

Nearest to 時：長癩，去世，緊急，復出，之前，考慮，然而，期間，
Nearest to 國：全，協商，第十二，人選，屆，同時，選為，六，
Nearest to 與：王位，郝，兩，戰敗，中，學者，爭奪，儘管，
Nearest to 第一：成為，世界，歷史，第三，生涯，以前，最近，終於，
Nearest to 由：轉移，做為，負責，則，依照，屬於，此後，該，
Nearest to 者：必須，提倡，情形，真正，方式，暴力，強制，透過，
Nearest to 國家：東歐，貢獻，進口，不得，伊朗，爭，有效，最終，
Nearest to 則：可，方，大大，而，家庭，由，其餘，為，
Nearest to 在：只，當地，帶來，從，狀況，時常，生活，集中，
Nearest to 個：這，但，不，之下，仍然，足球，它，真正，
Nearest to 三：約瑟夫，終結，取名為，長子，七，國王，世，育有，
Nearest to 和：努力，分析，是，提供，重要性，試圖，當時，教育，
Nearest to 能：只，從，它，不過，必須，或許，生命，毫無，
Nearest to 使用：用以，規範，比如，編碼，知識，輸入，詞彙，語音，
Nearest to 從：能，跟隨，成為，存在，以外，希臘，只，但是，
Nearest to 次：連續，場，世界，拿下，這，戰績，接下來，並且，
Nearest to 又：名，於，她們，說，不幸，該，有關，只，
Nearest to 地：見到，換，水，準備，不，其中，出發，以東，
Nearest to 月：起，隨後，春，同年，返回，召開，併，赴，
Nearest to 及：教育，時常，近年，用以，生活，效果，民間，布魯克蘭，

Without subsampling 跟只有做一次subsampling 都無法收斂 (隨著train 到不同的data 一直轉圈 圈)

```
Average loss at step 4998000 : 3.2418958055973053
Average loss at step 5000000 : 3.305725517988205
Nearest to 最: 為止, 度, 最多, 全, 之內, 達, 季, 有史以來,
Nearest to 二: 世, 去世, 達, 大公, 六, 壽, 伯爵, 生,
Nearest to 多: 另外, 被, 月底, 種, 因為, 所以, 人數, 順利,
Nearest to 向: 但是, 給, 迫使, 建議, 過去, 提出, 威脅, 更多,
Nearest to 就: 讓, 比較, 自己, 難, 想法, 過去, 都, 夠,
Nearest to 時: 錯, 而, 亦, 之前, 夠, 開會, 長達, 一般,
Nearest to 國: 全, 協商, 保護, 關於, 大會, 委員長, 批准, 第十二,
Nearest to 與: 電腦, 幸福, 贊助, 運作, 業者, 如何, 郝, 以,
Nearest to 第一: 第二, 新聞網, 歷史, 十三, 作戰, 大戰, 次, 郝,
Nearest to 由: 上, 為, 隊, 管, 正式, 贊助, 負責人, 負責,
Nearest to 者: 放棄, 更多, 亦, 對抗, 真正, 宣稱, 但, 地位,
Nearest to 國家: 重點, 有效, 能力, 計劃, 貢獻, 探索, 國土, 羽毛,
Nearest to 則: 一般, 可, 出現, 其, 該, 限, 或, 投入,
Nearest to 在: 卻, 亦, 下, 郝, 領先, 曼聯, 突然, 自此,
Nearest to 個: 叫做, 找, 真正, 或者, 以前, 種, 時候, 什麼,
Nearest to 三: 十四, 從此, 中間, 例, 最早, 嚴, 贊助, 建,
Nearest to 和: 被迫, 一般, 不得不, 但是, 給, 運作, 更多, 努力,
Nearest to 能: 才, 盛, 卻, 保持, 毫無, 應付, 只, 然而,
```

I assume the reason for not converging is it's too dense

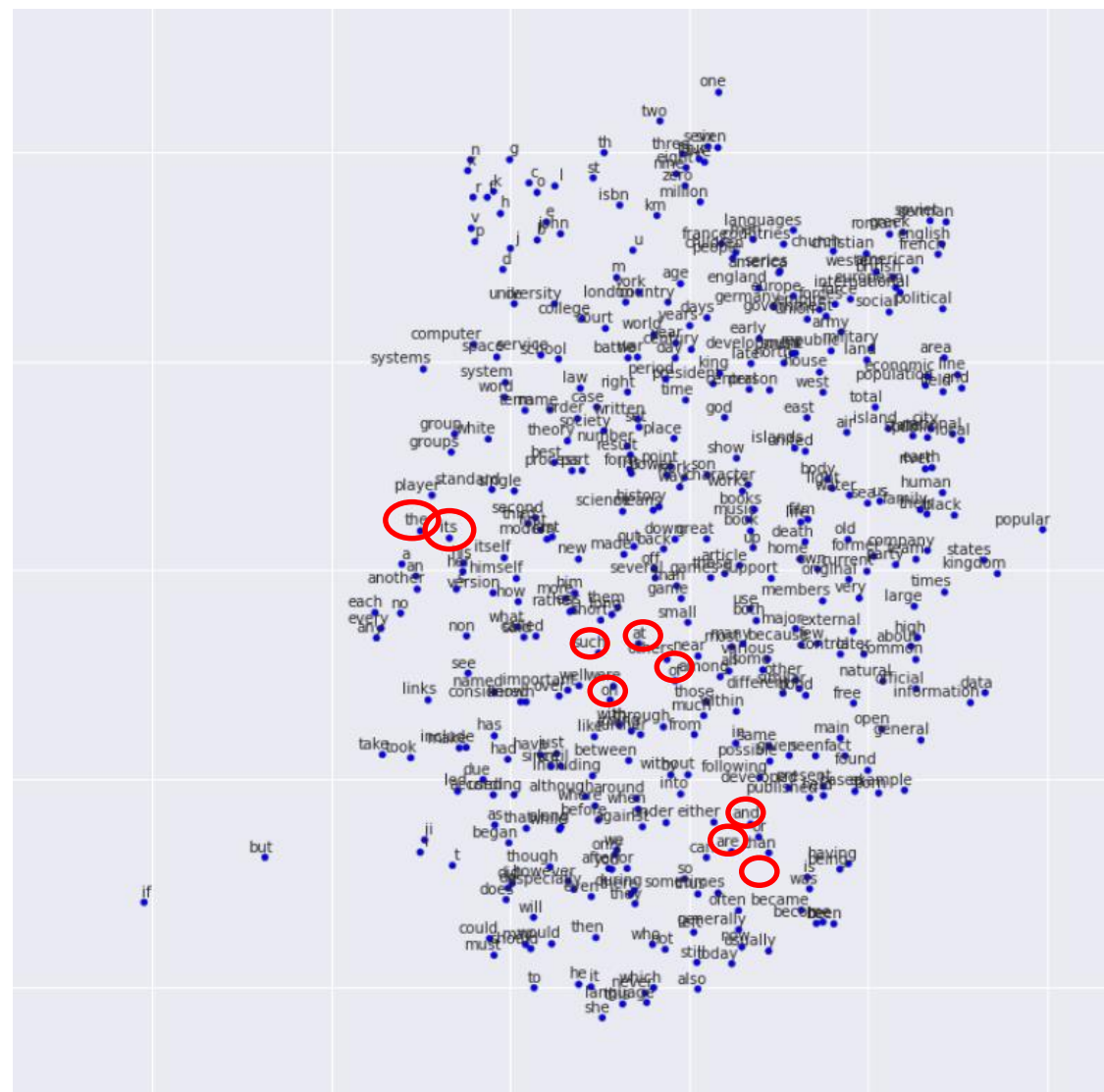
Stuff words

('的' , 5034580)
('在' , 1306559)
('是' , 1026858)
('一' , 1000255)
('年' , 785646)
('為' , 738728)
('了' , 701669)
('個' , 632132)
('有' , 631158)
('和' , 623109)
('中' , 522652)
('與' , 508076)
('之' , 503987)
('於' , 483496)

Distributional Hypothesis

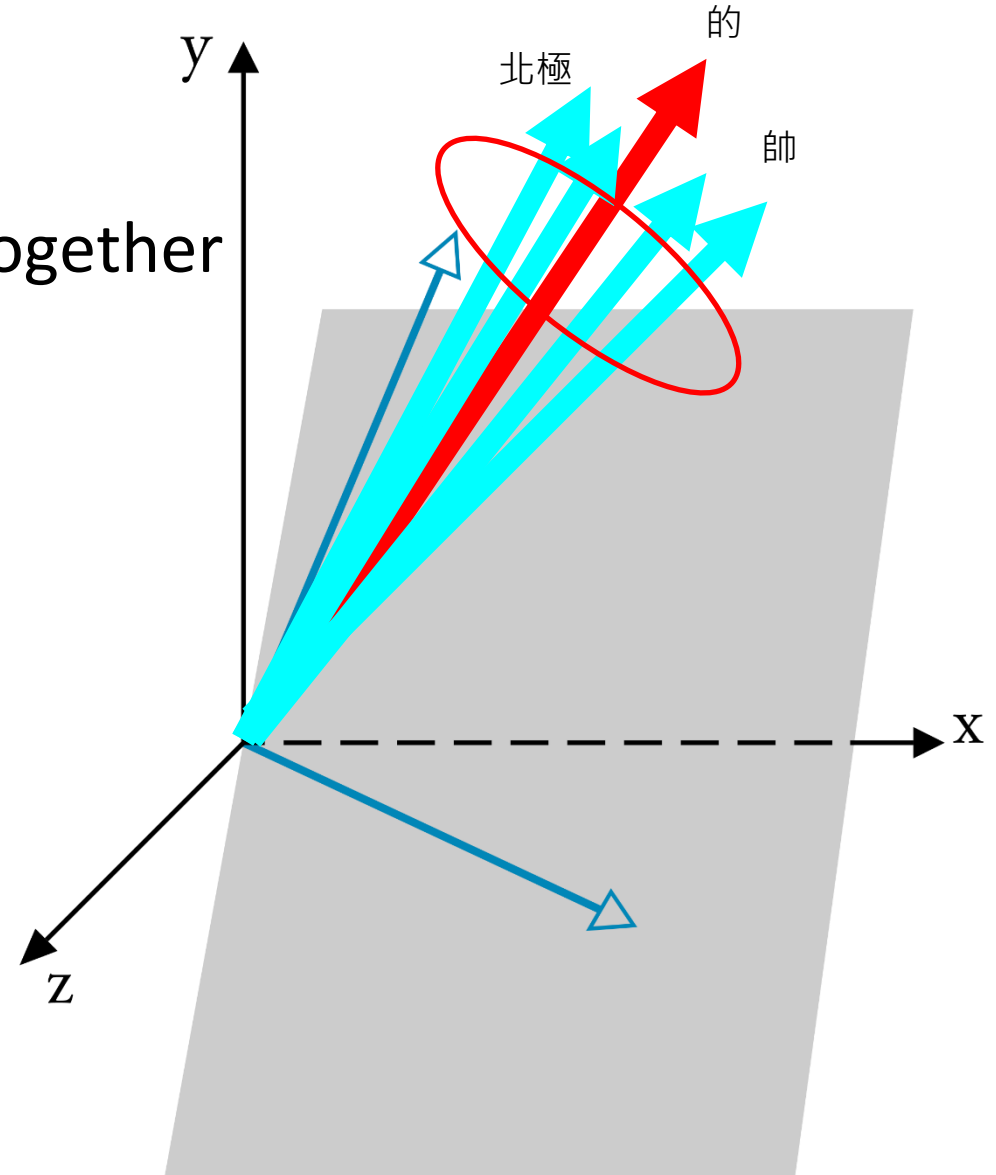
- There are still some exceptions such as 介係詞 Be動詞 or something else, which doesn't obey distributional hypothesis. They occur everywhere, making other vector stick to them:

Stuff word



Distributional Hypothesis: Destroy by stuff words

Different meaning words clustered together due to they surround “的”



After doing subsampling several time it become more stable, which I consider not enough. They have only bad effect.

- 這些stuff words還是很多

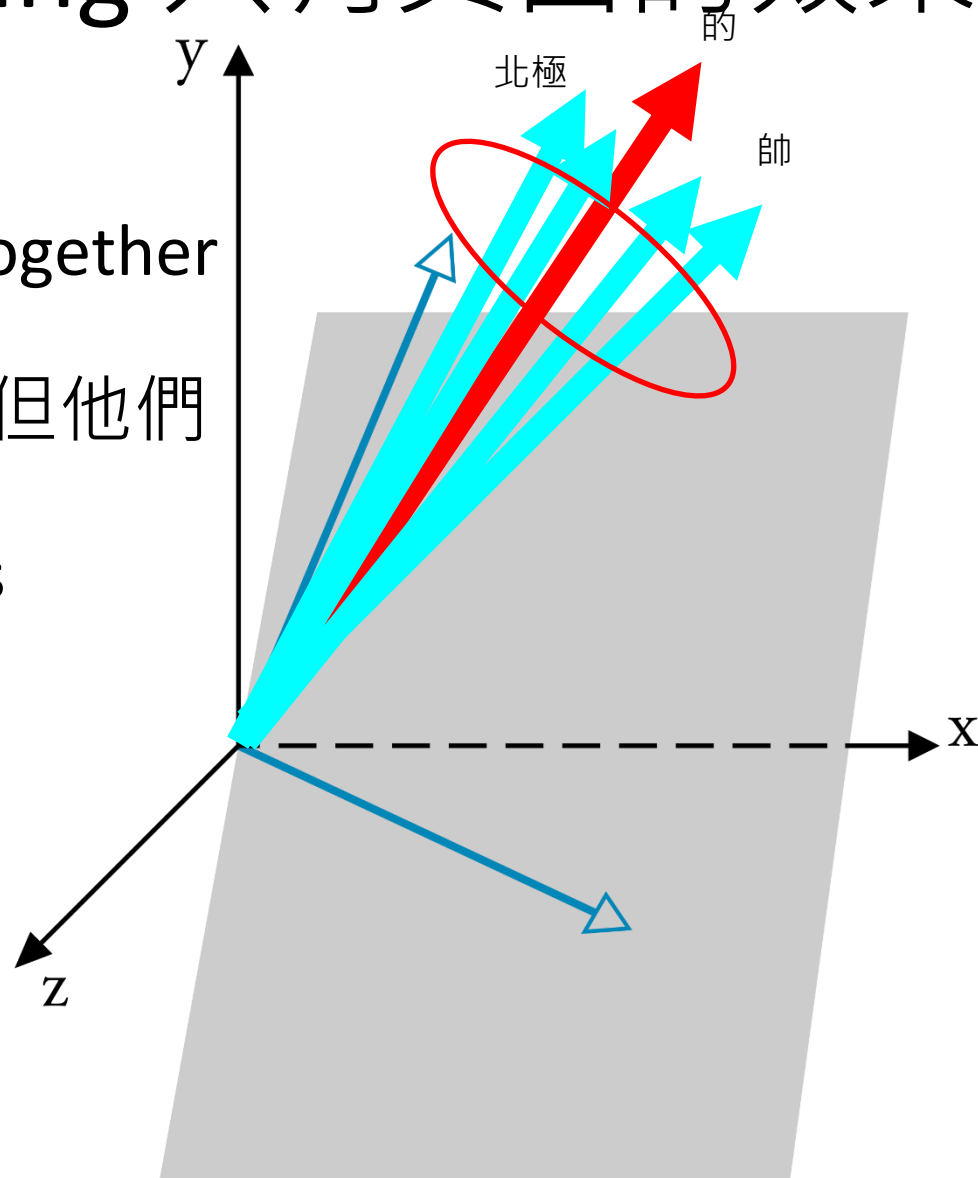
```
('的', 20624)
('在', 17716)
('一', 17446)
('是', 17246)
('年', 17012)
('為', 16881)
('了', 16876)
('和', 16746)
('有', 16624)
('個', 16502)
after subsampling
('的', 14354)
('在', 13582)
('一', 13494)
('是', 13385)
('了', 13352)
('年', 13331)
('個', 13237)
('有', 13230)
('和', 13212)
('為', 13193)
```

Stuff word 對這個training 只有負面的效果。

Different meaning words clustered together due to they surround “的”

“的”跟周遭的word vector很相近，但他們的意思仍舊相差很遠啊！

他還會破壞distributional hypothesis
把意思很不一樣的字拉近。



I assume that after paring them off, the quality of word vector will improve significantly

Thanks!