Text Classification

2/2565: FRA501 Introduction to Natural Language Processing with Deep learning

Week 06

Paisit Khanarsa, Ph.D.

Institute of Field Robotics (FIBO), King Mongkut's University of Technology Thonburi

Outlines

- Introduction to text classification task
- Bag of words model
  - Naïve Bayes (A traditional model)
  - Neural methods
    - Deep Averaging Networks(DAN)
    - Universal Sentence Encoder (USE)
    - Unsupervised pre-training
- Topic modeling

Introduction

- Wongnai Challenge
  - Predict star rating from review text

Introduction (cont.)

- Yelp reviews
- Document Modeling with Gated Recurrent Neural Network for Sentiment Classification

Corpus	#docs	#s/d	#w/d	$\|V\|$	#class	Class Distribution
Yelp 2013	335,018	8.90	151.6	211,245	5	.09/.09/.14/.33/.36
Yelp 2014	1,125,457	9.22	156.9	476,191	5	.10/.09/.15/.30/.36
Yelp 2015	1,569,264	8.97	151.9	612,636	5	.10/.09/.14/.30/.37
IMDB	348,415	14.02	325.6	115,831	10	.07/.04/.05/.05/.08/.11/.15/.17/.12/.18

	Yelp 2013		Yelp 2014		Yelp 2015		IMDB	
	Accuracy	MSE	Accuracy	MSE	Accuracy	MSE	Accuracy	MSE
Majority	0.356	3.06	0.361	3.28	0.369	3.30	0.179	17.46
SVM + Unigrams	0.589	0.79	0.600	0.78	0.611	0.75	0.399	4.23
SVM + Bigrams	0.576	0.75	0.616	0.65	0.624	0.63	0.409	3.74
SVM + TextFeatures	0.598	0.68	0.618	0.63	0.624	0.60	0.405	3.56
SVM + AverageSG	0.543	1.11	0.557	1.08	0.568	1.04	0.319	5.57
SVM + SSWE	0.535	1.12	0.543	1.13	0.554	1.11	0.262	9.16
JMARS	N/A	–	N/A	–	N/A	–	N/A	4.97
Paragraph Vector	0.577	0.86	0.592	0.70	0.605	0.61	0.341	4.69
Convolutional NN	0.597	0.76	0.610	0.68	0.615	0.68	0.376	3.30
Conv-GRNN	0.637	0.56	0.655	0.51	0.660	0.50	0.425	**2.71**
LSTM-GRNN	**0.651**	**0.50**	**0.671**	**0.48**	**0.676**	**0.49**	**0.453**	3.00

Tang, D., Qin, B., & Liu, T. (2015, September). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1422-1432).

Introduction (cont.)

- Document classification

Type	Focus	Example
Topic	Subject matter	Sport vs Technology
Sentiment/opinion	Emotion (current state)	Negative vs Positive
Intent	Action (future state)	Order vs Inquiry

- Other classification application
  - Spam filtering
  - Authorship id
  - Auto tagging (information retrieval)
  - Trend analysis

Introduction (cont.)

- Text classification definition
  - Input
    - Set of documents: $D = \{d_1, d_2, d_3, \dots, d_M\}$
    - Each document is composed of words: $d_1 = [w_{11}, w_{12}, \dots, w_{1N}]$
    - Set of classes: $C = \{c_1, c_2, c_3, \dots, c_m\}$
  - Output
    - The predicted class c_i from the set C

Introduction (cont.)

- Rule-based classification
  - Rule based on phrases or other features
  - Wongnai rating
    - “อร่อย” → ★★★★
    - “ไม่อร่อย” → ★★
    - “สกปรก” → ★
    - ….

Introduction (cont.)

- Rule-based classification
  - Rule based on phrases or other features
  - Wongnai rating
    - “อร่อย” → ★★★★
    - “ไม่อร่อย” → ★★
    - “สกปรก” → ★
    - ….
  - What rating of this phase is “ไม่ค่อยอร่อย” → maybe ★★

Introduction (cont.)

- Rule-based classification
  - Rule based on phrases or other features
  - Wongnai rating
    - “อร่อย” → ★★★★
    - “ไม่อร่อย” → ★★
    - “สกปรก” → ★
    - ….
  - What rating of this phase is “ไม่ค่อยอร่อย” → maybe ★★
  - What rating of this phase is “ไม่ถูกแต่อร่อย” → ???

Introduction (cont.)

- Rule-based classification
  - Rule based on phrases or other features
  - Wongnai rating
    - "อร่อย" → ★★★★
    - "ไม่อร่อย" → ★★
    - "สกปรก" → ★
    - ….
  - What rating of this phase is "ไม่ค่อยอร่อย" → maybe ★★
  - What rating of this phase is "ไม่ถูกแต่อร่อย" → ???
- Pros: easy to implement, can yield very good results
- Cons: building and maintaining rules is expensive

Introduction (cont.)

- Text classification definition
  - Input
    - Set of documents: $D = \{d_1, d_2, d_3, \dots, d_M\}$
    - Labels: $Y = \{y_1, y_2, y_3, \dots, y_M\}$
    - Each document is composed of words: $d_1 = [w_{11}, w_{12}, \dots, w_{1N}]$
    - Set of classes: $C = \{c_1, c_2, c_3, \dots, c_m\}$
  - Output
    - The predicted class c_i from the set C
    - A classifier $H: d \rightarrow c$

Introduction (cont.)

- Text classification definition
  - Input
    - Set of documents: $D = \{d_1, d_2, d_3,$
    - Labels: $Y = \{y_1, y_2, y_3, \dots, y_M\}$
    - Each document is composed of wor
    - Set of classes: $C = \{c_1, c_2, c_3, \dots, c_m$

  - Output
    - The predicted class c_i from the set C
    - A classifier $H: d \rightarrow c$

Classifiers

- k-NN
- Naïve Bayes
- Logistic regression
- SVM
- Neural networks

Bag of words representation

$$H\left(\begin{array}{c}\text{"วันนี้เป็นครั้งแรกที่ได้ทานเค้กของ } \textbf{Farm Desing} \text{ ครับ} \\ \text{เดินผ่านหลายครั้งแล้วแต่ไม่ได้คิดที่จะ ..."}\end{array}\right) = 4$$

Bag of words representation (cont.)

$$H\left(\text{"วันนี้เป็นครั้งแรกที่ได้ทานเค้กของ Farm Desing ครับ เดินผ่านหลายครั้งแล้วแต่ไม่ได้คิดที่จะ ..."}\right) = 4$$

Bag of words that just consider word or feature existence while ignoring word position and context

$$H(\underbrace{\text{ชอบ, ชอบ, อร่อย, อร่อย, สะอาด, ไม่, ไม่, สุดยอด, ...}}_{\text{Bag of words}}) = 4$$

Bag of words representation (cont.)

$$H\left(\text{"วันนี้เป็นครั้งแรกที่ได้ทานเค้กของ Farm Desing ครับ เดินผ่านหลายครั้งแล้วแต่ไม่ได้คิดที่จะ ..."}\right) = 4$$

Bag of words that just consider word or feature existence while ignoring word position and context

$H($ชอบ, ชอบ, … อด, ….$) = 4$

Word	Count
ชอบ	2
อร่อย	2
สะอาด	1
ไม่	2
สุดยอด	1
…	…

Bag of words representation (cont.)

Training

- สกปรก, แย่, เหม็น ★
- ถูก, กลางๆ, อร่อย, ใช้ได้ ★★★
- อร่อย, มาก, สุดยอด, ยอดเยี่ยม ★★★★★
- ....

Testing

Rating ..?

ถูก, อร่อย, ใช้ได้, แย่

Bag of words representation (cont.)

Training

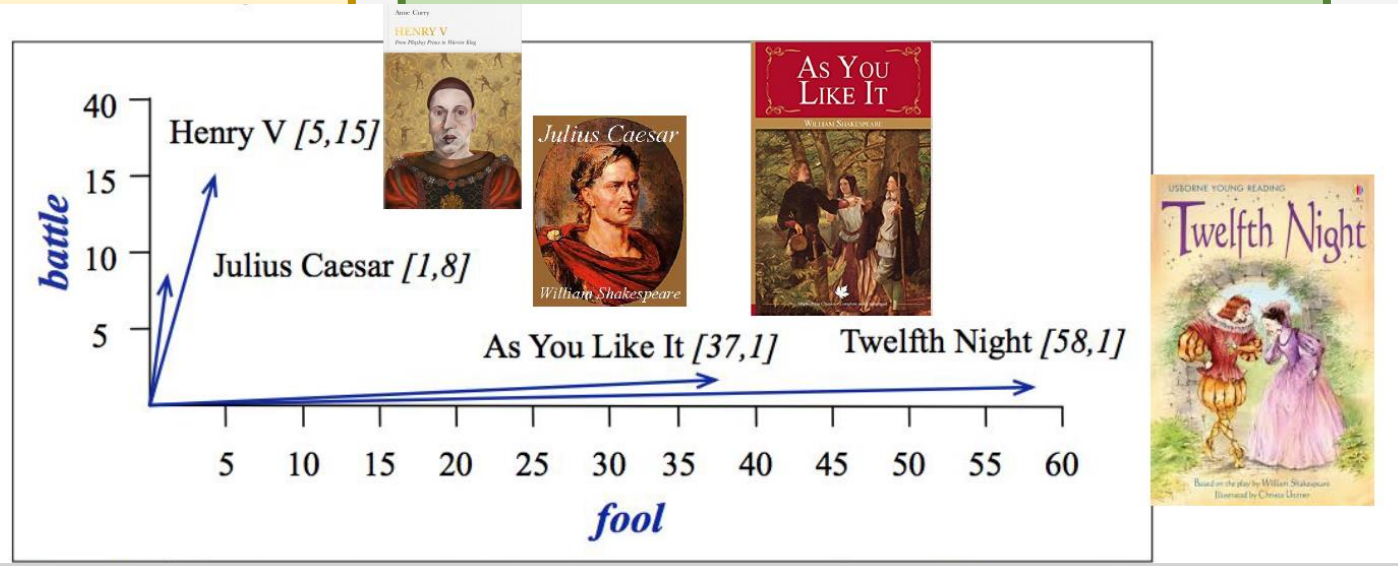
- สกปรก, แย่, เหม็น ★
- ถูก, กลางๆ, อร่อย, ใช้ได้ ★★★
- อร่อย, มาก, สุดยอด, ยอดเยี่ยม ★★★★★

Word	Count	
ถูก	1	★★★
อร่อย	★ 1	★★★
ใช้ได้	1	★★★
แย่	1	★★★★★
...	...	

Testing

Rating ..?

ถูก, อร่อย, ใช้ได้, แย่



Bayes' Rule for classification

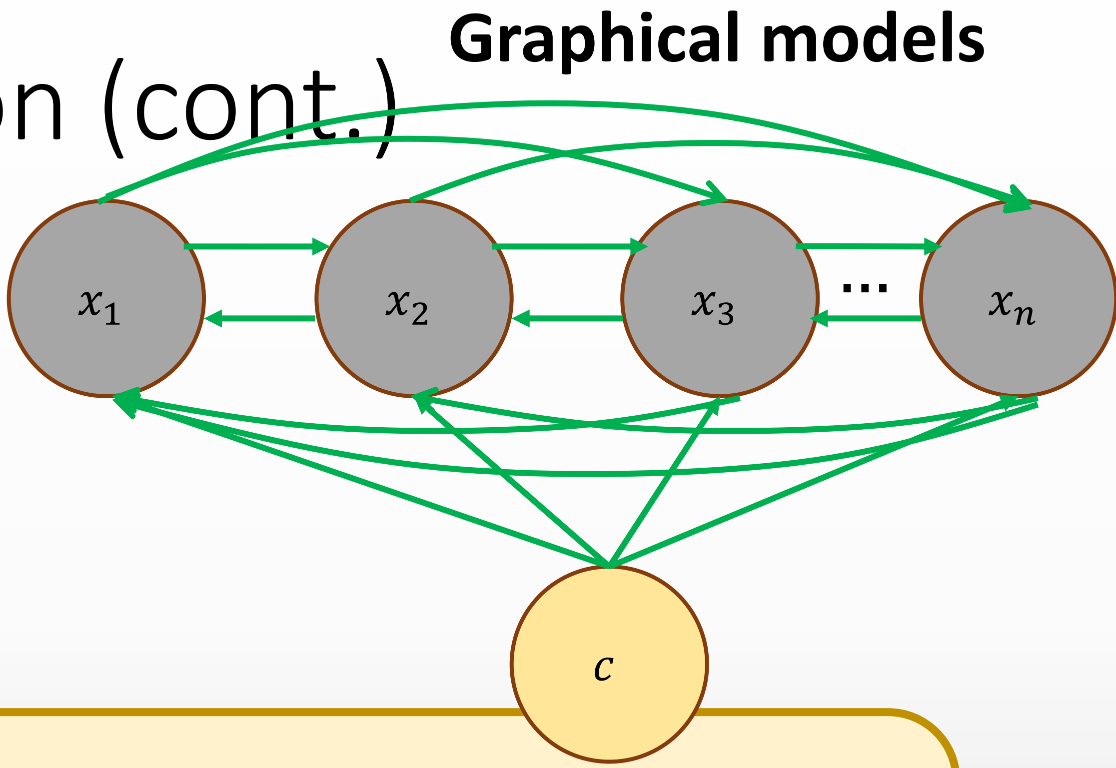
- A simple classification model
  - Set of documents: $D = \{d_1, d_2, d_3, \dots, d_M\}$
  - Set of classes: $C = \{c_1, c_2, c_3, \dots, c_m\}$
  - The predicted class c_i from the set C
  - A classifier $H: d \rightarrow c$

Bayes' Rule

$$Argmax_c P(c|d) = Argmax_c \frac{P(d|c)P(c)}{P(d)} \Rightarrow Argmax_c P(d|c)P(c) =$$
$$Argmax_c P(x_1, x_2, x_3, \dots x_n|c)P(c)$$

- The document is represented by features $x_1, x_2, x_3, \dots x_n$

Bayes' Rule for classification (cont.)



- A simple classification model
  - Set of documents: $D = \{d_1, d_2, d_3, \dots, d_M\}$
  - Set of classes: $C = \{c_1, c_2, c_3, \dots, c_m\}$
  - The predicted class c_i from the set C
  - A classifier $H: d \rightarrow c$

Bayes' Rule

$$Argmax_c P(c|d) = Argmax_c \frac{P(d|c)P(c)}{P(d)} \Rightarrow Argmax_c P(d|c)P(c) =$$

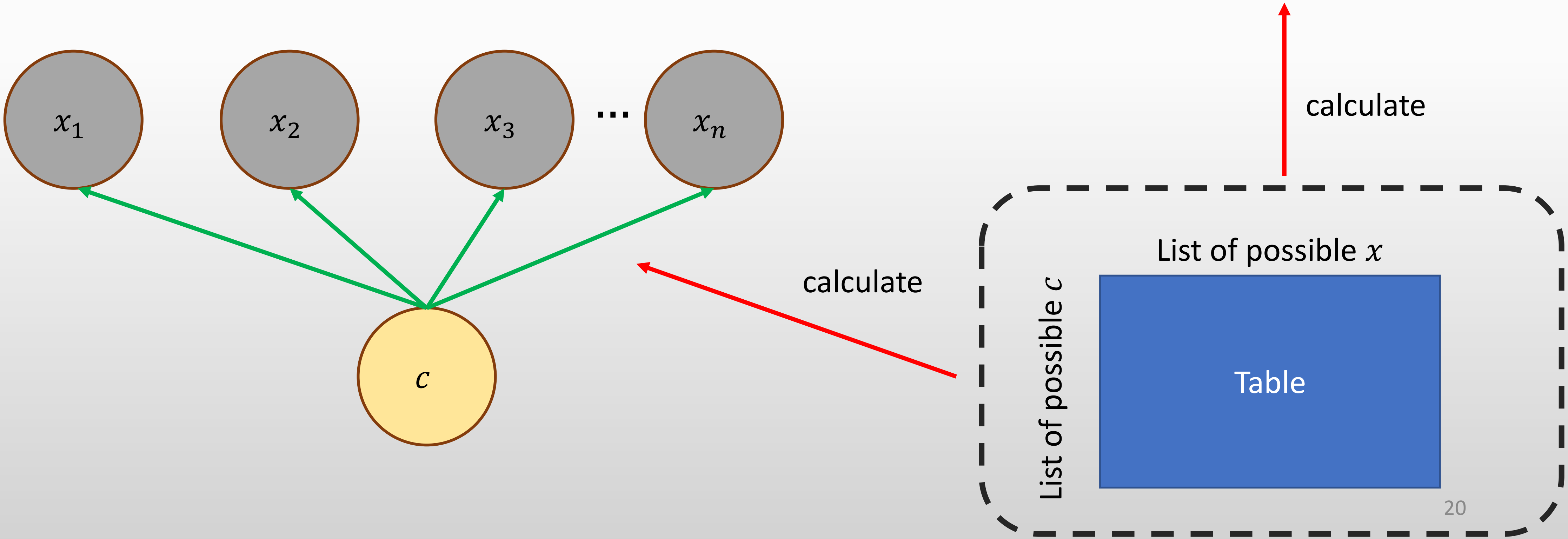
$$\boxed{Argmax_c P(x_1, x_2, x_3, \dots x_n|c)P(c)}$$ **Hard to train !!!**

- The document is represented by features $x_1, x_2, x_3, \dots x_n$

Bag of words assumption and Naïve Bayes

- Conditional independence

$$Argmax_c P(x_1, x_2, x_3, \ldots x_n | c) P(c) = Argmax_c P(x_1|c) P(x_2|c) \ldots P(x_n|c) P(c)$$



Bag of words assumption and Naïve Bayes (cont.)

- Probability of drawing with replacement words from the bag of word distribution
- Example:

Word	Distribution of Class 5
ชอบ	0.3
อร่อย	0.3
ไม่	0.05
กลมกล่อม	0.25
ทานง่าย	0.1

$$P(\text{ไม่อร่อยไม่ชอบ}|c = 1)$$

$$= P(\text{ไม่}|c = 1)P(\text{อร่อย}|c = 1)P(\text{ไม่}|c = 1)P(\text{ชอบ}|c = 1)$$
$$= 0.05 \times 0.3 \times 0.05 \times 0.3 \times \frac{4!}{2!}$$
$$= 0.0027$$

Naïve Bayes Learning

How to find ?

Word	Distribution of Class 5
ชอบ	0.3
อร่อย	0.3
ไม่	0.05
กลมกล่อม	0.25
ทานง่าย	0.1

Naïve Bayes Learning (cont.)

How to find ?

Word	Probability	Distribution of Class 5
ชอบ	$P(\text{ชอบ}\|c=5)$	0.3
อร่อย	$P(\text{อร่อย}\|c=5)$	0.3
ไม่	$P(\text{ไม่}\|c=5)$	0.05
กลมกล่อม	$P(\text{กลมกล่อม}\|c=5)$	0.25
ทานง่าย	$P(\text{ทานง่าย}\|c=5)$	0.1

- $P(x|c)$
- $P(x = \text{"ชอบ"}|c = 5) = \frac{count(x=\text{"ชอบ"},c=5)}{count(c=5)}$
- $P(c)$
- $P(c = 5) = \frac{count(c=5)}{count(all\ reviews)}$

Naïve Bayes Learning (cont.)

How to find ?

Word	Probability	Distribution of Class 5
ชอบ	$P(\text{ชอบ}\|c = 5)$	0.3
อร่อย	$P(\text{อร่อย}\|c = 5)$	0.3
ไม่	$P(\text{ไม่}\|c = 5)$	0.05
กลมกล่อม	$P(\text{กลมกล่อม}\|c = 5)$	0.25
ทานง่าย	$P(\text{ทานง่าย}\|c = 5)$	0.1

- $P(x|c)$
- $P(x = \text{"ชอบ"}|c = 5) = \frac{count(x=\text{"ชอบ"},c=5)}{count(c=5)}$
- $P(c)$
- $P(c = 5) = \frac{count(c=5)}{count(all\ reviews)}$

Problems !!!

$P(\text{อาหารร้านนี้รสชาติไม่กลมกล่อมเลย}|c = 1)$
$= P(\text{ไม่}|c = 1)\boxed{P(\text{กลมกล่อม}|c = 1)}$
$= 0$

Hard to appear

Naïve Bayes Learning (cont.)

How to find ?

Word	Probability	Distribution of Class 5
ชอบ	$P(\text{ชอบ}\|c=5)$	0.3
อร่อย	$P(\text{อร่อย}\|c=5)$	0.3
ไม่	$P(\text{ไม่}\|c=5)$	0.05
กลมกล่อม	$P(\text{กลมกล่อม}\|c=5)$	0.25
ทานง่าย	$P(\text{ทานง่าย}\|c=5)$	0.1

Solution

Smoothing techniques

- Add-one estimation
- Back-off
- Interpolation
- Kneser-Ney Smoothing

- $P(x|c)$
- $P(x=\text{"ชอบ"}|c=5)=\frac{count(x=\text{"ชอบ"},c=5)}{count(c=5)}$
- $P(c)$
- $P(c=5)=\frac{count(c=5)}{count(all\ reviews)}$

Problems !!!

$$P(\text{อาหารร้านนี้รสชาติไม่กลมกล่อมเลย}|c=1)$$
$$=P(\text{ไม่}|c=1)\boxed{P(\text{กลมกล่อม}|c=1)}$$
$$=0$$

Hard to appear

Naïve Bayes

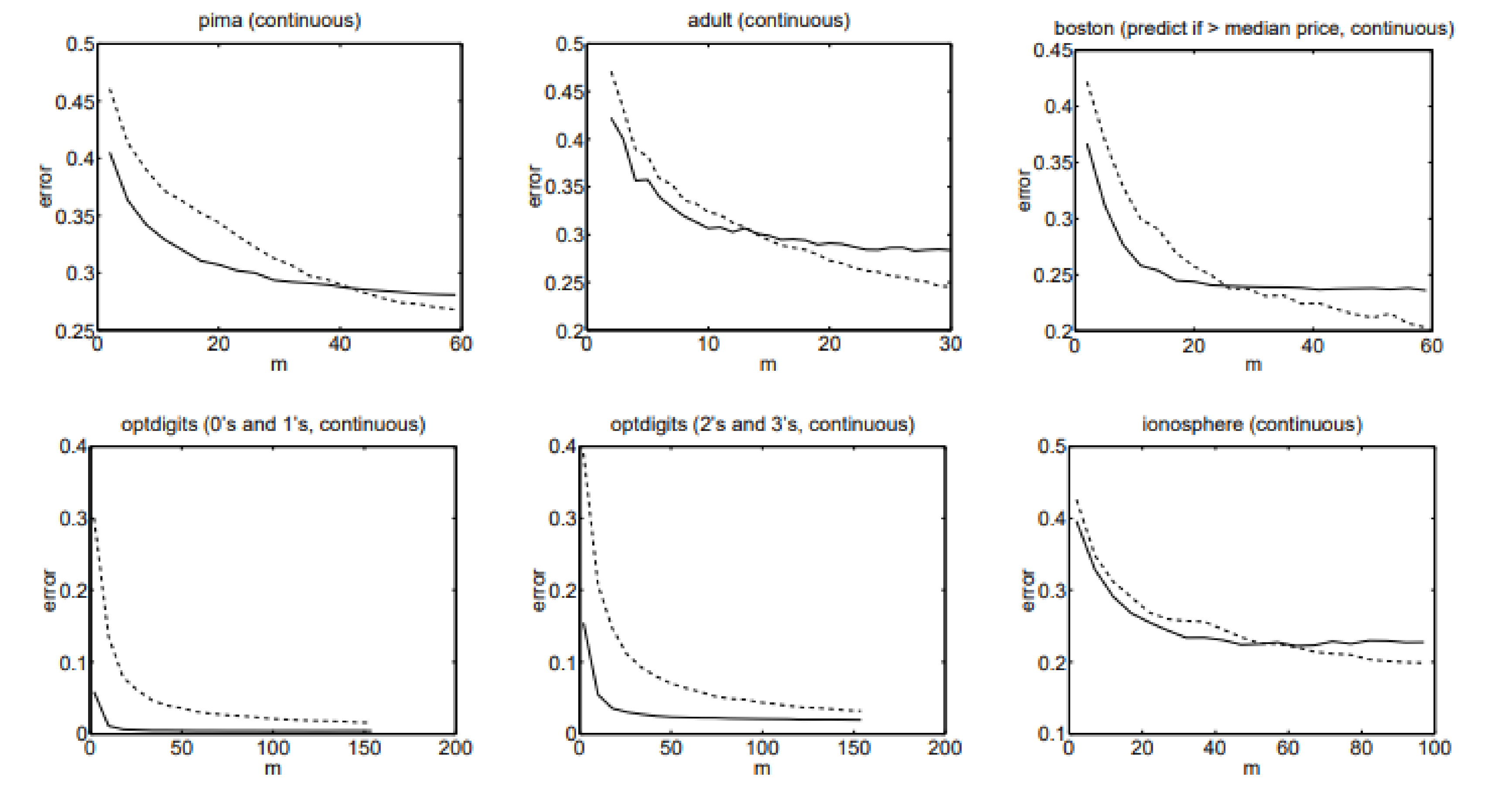
- Feature engineering: restaurant name, location, price range, reviewer id, date of review
- More 1000 features
- Pros: very fast, very small model
- Robust especially for small training data
- A good fast baseline. Always try Naive Bayes or logistic regression in model search.

Naïve Bayes vs Logistic regression

- Naïve Bayes are generative models
  - $\hat{c} = Argmax_c \frac{P(d|c)P(c)}{P(d)}$
- Logistic regression are discriminative models
  - $\hat{c} = Argmax_c P(c|d)$
- Logistic regression and Naive Bayes are linear models (linear decision boundary)
- They are quite interchangeable.

Naïve Bayes vs Logistic regression (cont.)

- Dashed line is logistic regression
- Solid line is Naïve Bayes



Ng, A., & Jordan, M. (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, *14*.

https://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf

Naïve Bayes vs Logistic regression (cont.)

- Features: n-grams (bag of phrases)
- Model: logistic regression
- Very competitive results

Model	Yelp'13	Yelp'14	Yelp'15	IMDB
SVM+TF	59.8	61.8	62.4	40.5
CNN	59.7	61.0	61.5	37.5
Conv-GRNN	63.7	65.5	66.0	42.5
LSTM-GRNN	65.1	67.1	67.6	45.3
`fastText`	64.2	66.2	66.6	45.2

Table 3: Comparision with Tang et al. (2015). The hyper-parameters are chosen on the validation set. We report the test accuracy.

	Zhang and LeCun (2015)		Conneau et al. (2016)			`fastText`
	small char-CNN	big char-CNN	depth=9	depth=17	depth=29	$h = 10$, bigram
AG	1h	3h	24m	37m	51m	1s
Sogou	-	-	25m	41m	56m	7s
DBpedia	2h	5h	27m	44m	1h	2s
Yelp P.	-	-	28m	43m	1h09	3s
Yelp F.	-	-	29m	45m	1h12	4s
Yah. A.	8h	1d	1h	1h33	2h	5s
Amz. F.	2d	5d	2h45	4h20	7h	9s
Amz. P.	2d	5d	2h45	4h25	7h	10s

Table 2: Training time for a single epoch on sentiment analysis datasets compared to char-CNN and VDCNN.

Model	prec@1	Running time	
		Train	Test
Freq. baseline	2.2	-	-
Tagspace, $h = 50$	30.1	3h8	6h
Tagspace, $h = 200$	35.6	5h32	15h
`fastText`, $h = 50$	31.2	6m40	48s
`fastText`, $h = 50$, bigram	36.7	7m47	50s
`fastText`, $h = 200$	41.1	10m34	1m29
`fastText`, $h = 200$, bigram	46.1	13m38	1m37

Table 5: Prec@1 on the test set for tag prediction on YFCC100M. We also report the training time and test time. Test time is reported for a single thread, while training uses 20 threads for both models.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

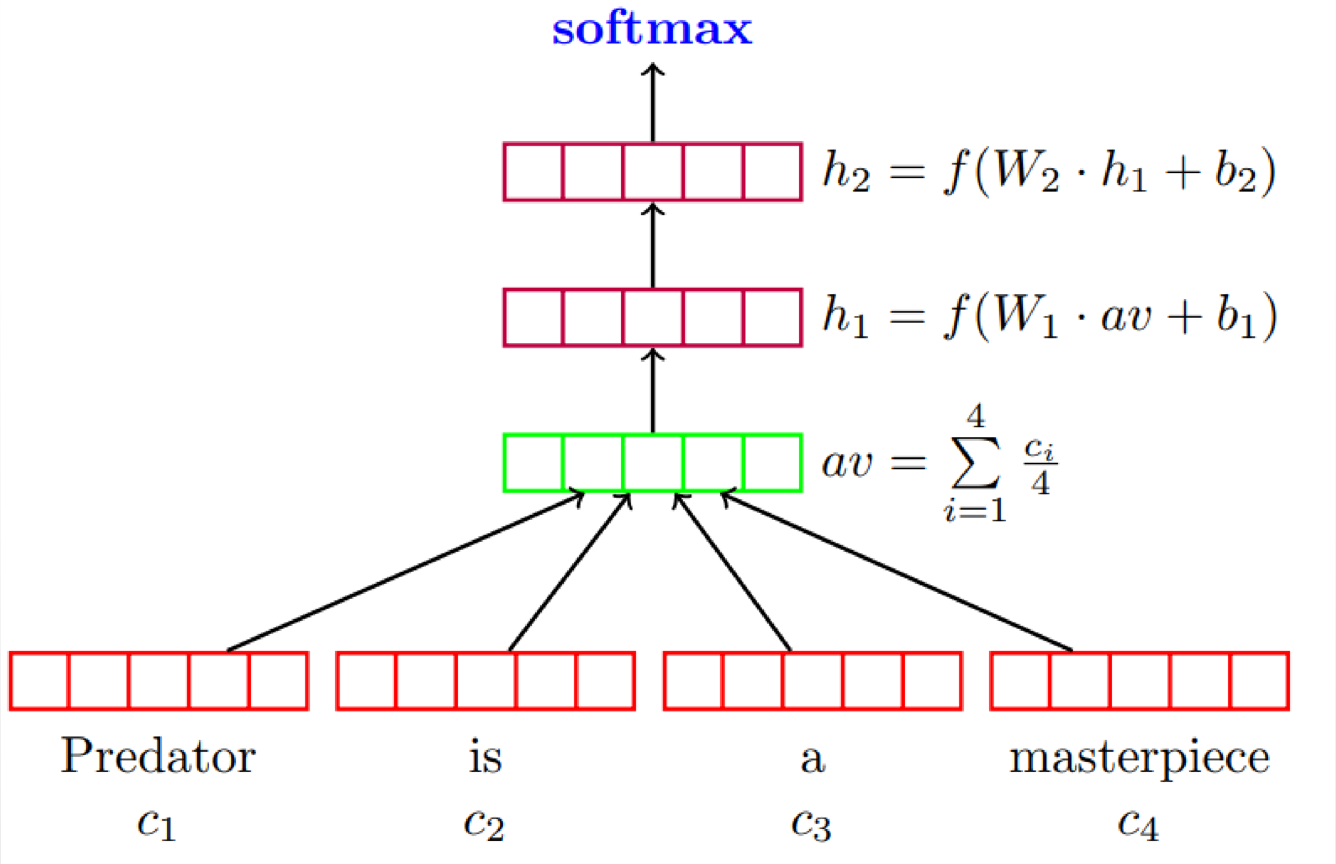
https://arxiv.org/pdf/1607.01759.pdf

Naïve Bayes tricks for text classification

- Count words after “not” as a different word
  - I don’t go there. → I don’t go_not there_not
- Upweighting: double counting words at important locations
  - Words in titles
  - First sentence of each paragraph
  - Sentences that contain title words

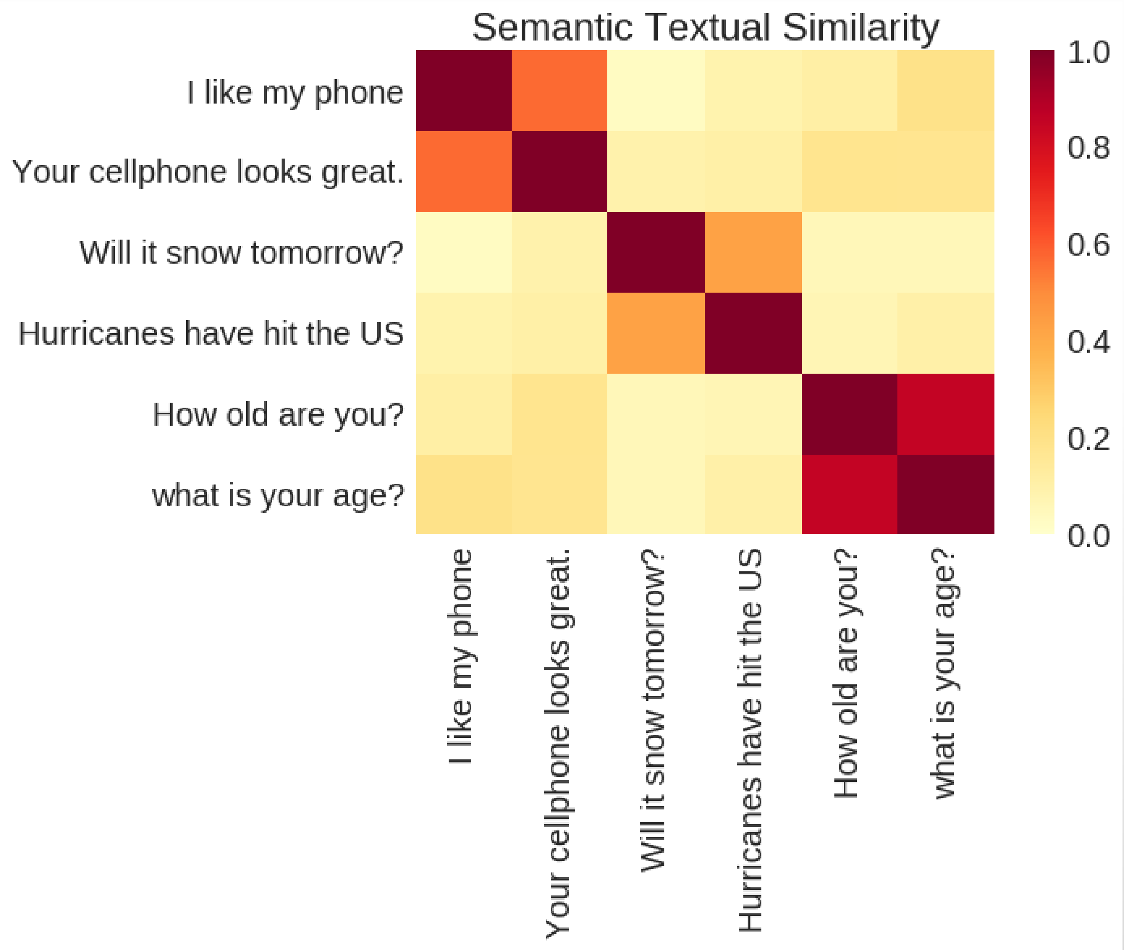
Neural methods: Deep Averaging Networks(DAN)

- Deep Averaging Networks (DAN)



Universal Sentence Encoder (USE)

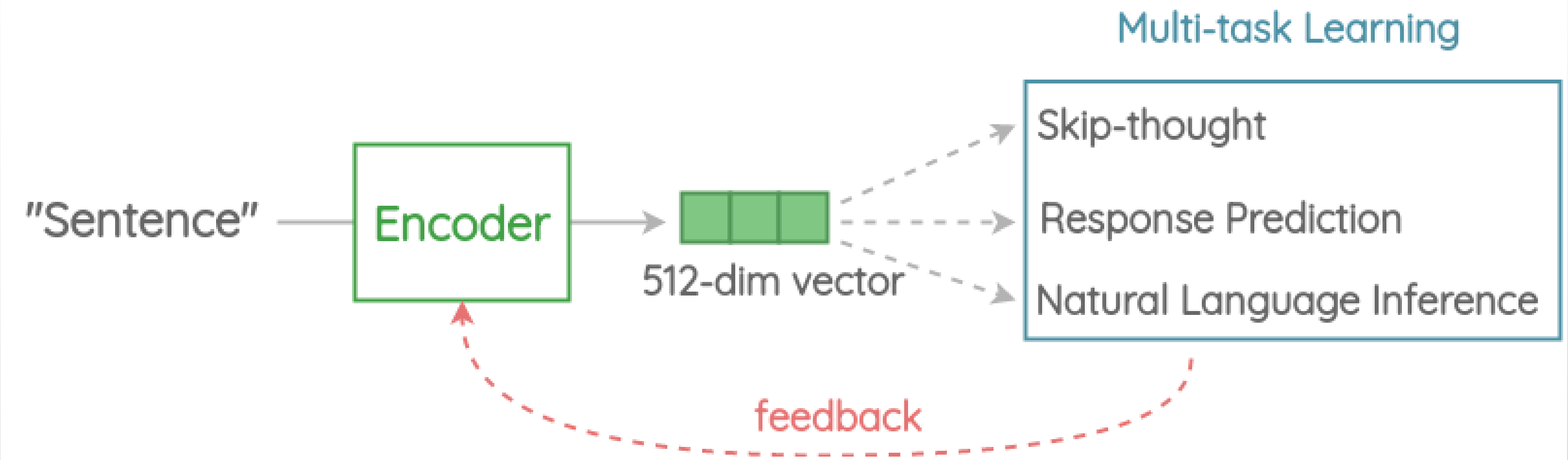
- A model focusing on sentence representation
- Use sentence piece tokenization
- Pre-trained then used anywhere
- Based on
  - (1) DAN (lite version)
  - (2) Transformer



https://ai.googleblog.com/2018/05/advances-in-semantic-textual-similarity.html

Pretraining USE

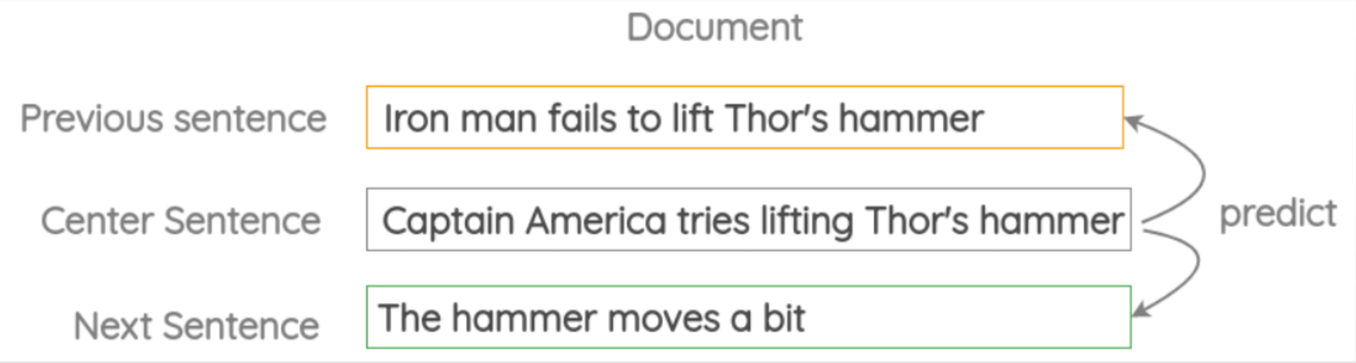
- Training USE done using multi-task
  - Skip-thought
  - Response prediction
  - Natural language inference (NLI)

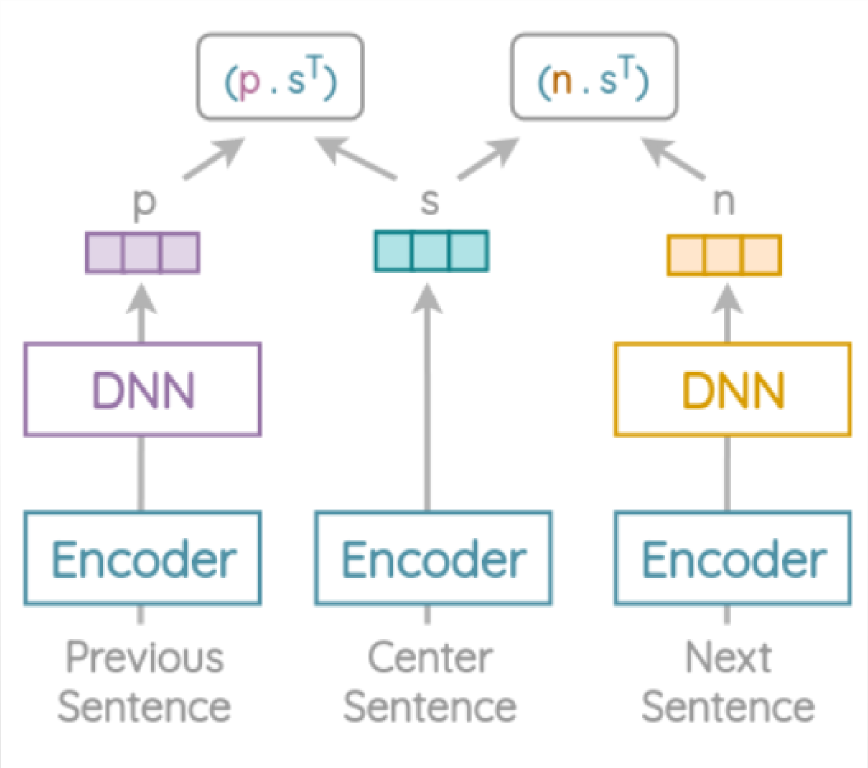


https://amitness.com/2020/06/universal-sentence-encoder/

Pretraining USE: Skip-thought

- Similar to skip-gram
- Use the current sentence to predict the previous and next sentence
- Proposed by Kiros et al.



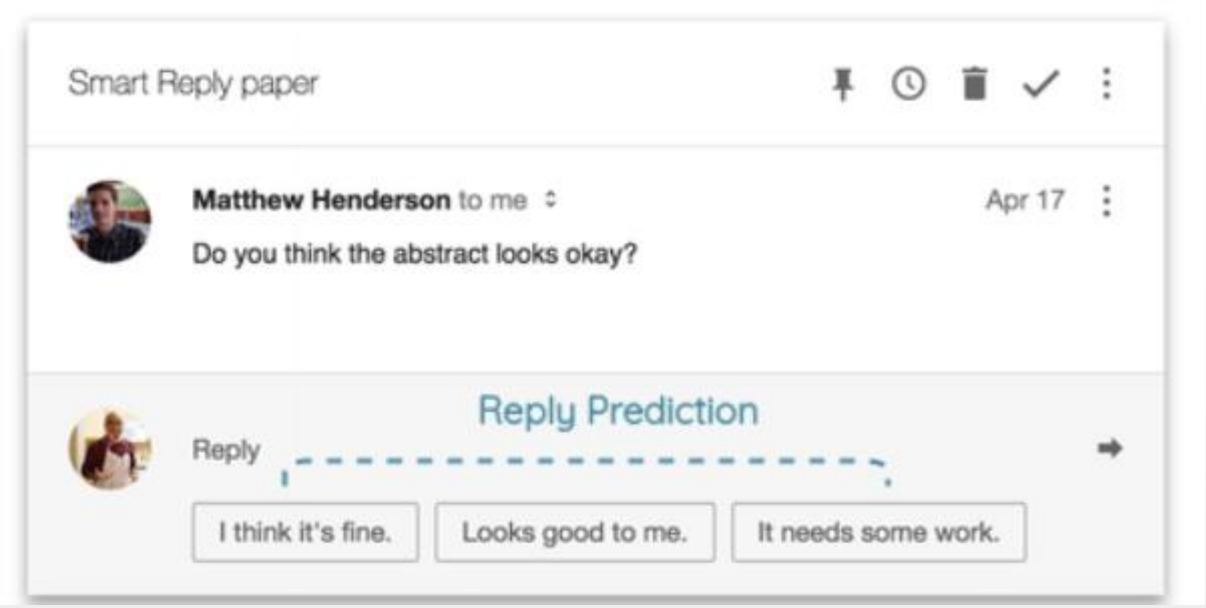


Skip-thought Task Structure

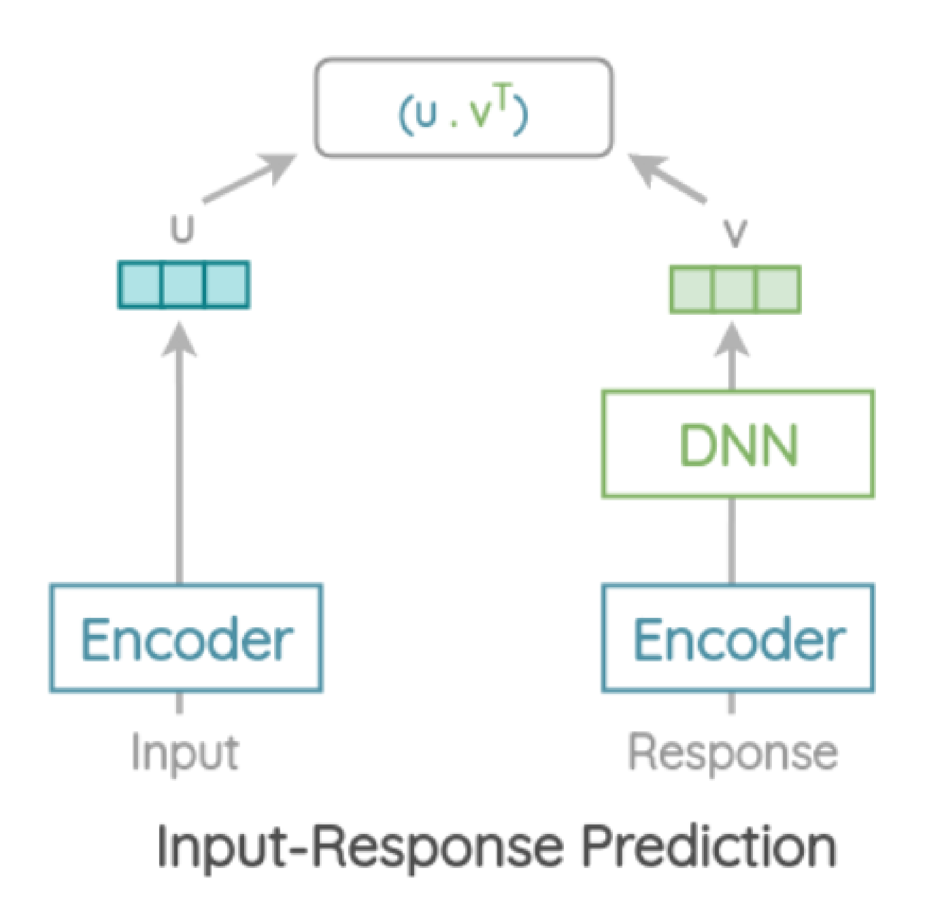
Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. *Advances in neural information processing systems*, *28*.

https://amitness.com/2020/06/universal-sentence-encoder/

Pretraining USE: Response prediction



- Predict the correct response for a given input among a list of correct responses
- Proposed by Henderson et al.



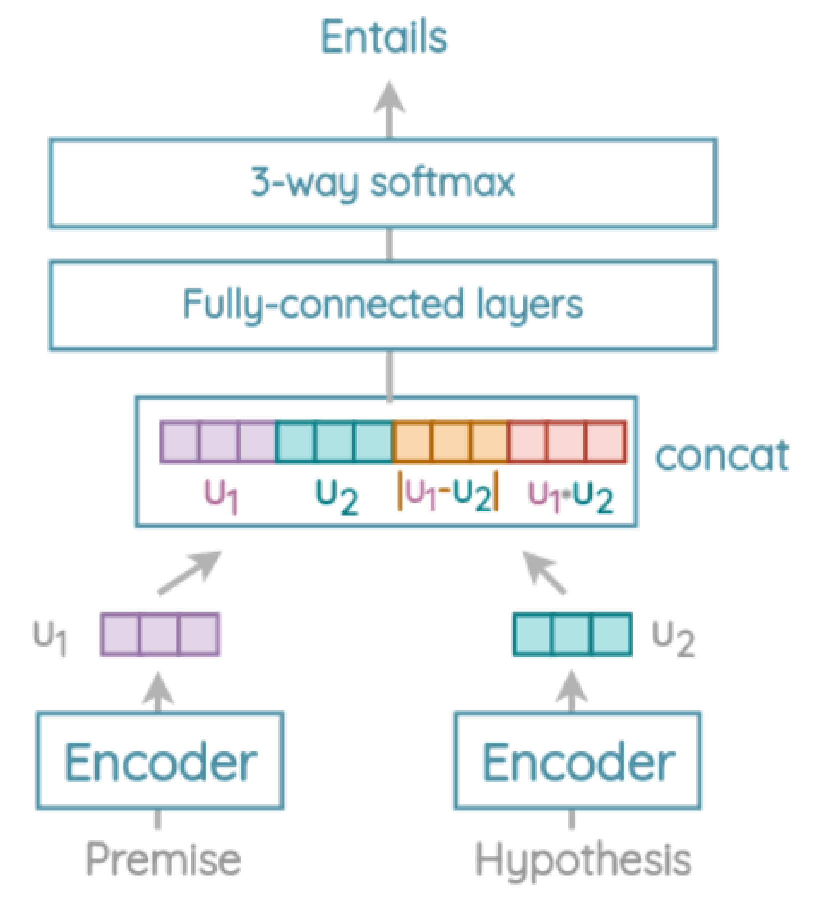
Henderson, M., Al-Rfou, R., Strope, B., Sung, Y. H., Lukács, L., Guo, R., ... & Kurzweil, R. (2017). Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

https://amitness.com/2020/06/universal-sentence-encoder/

Pretraining USE: Natural language inference (NLI)

Premise	Hypothesis	Judgement
A soccer game with multiple males playing	Some men are playing a sport	entailment
I love Marvel movies	I hate Marvel movies	contradiction
I love Marvel movies	A ship arrived	neutral

- Predict relationship between sentence
- Proposed by Conneau et al.

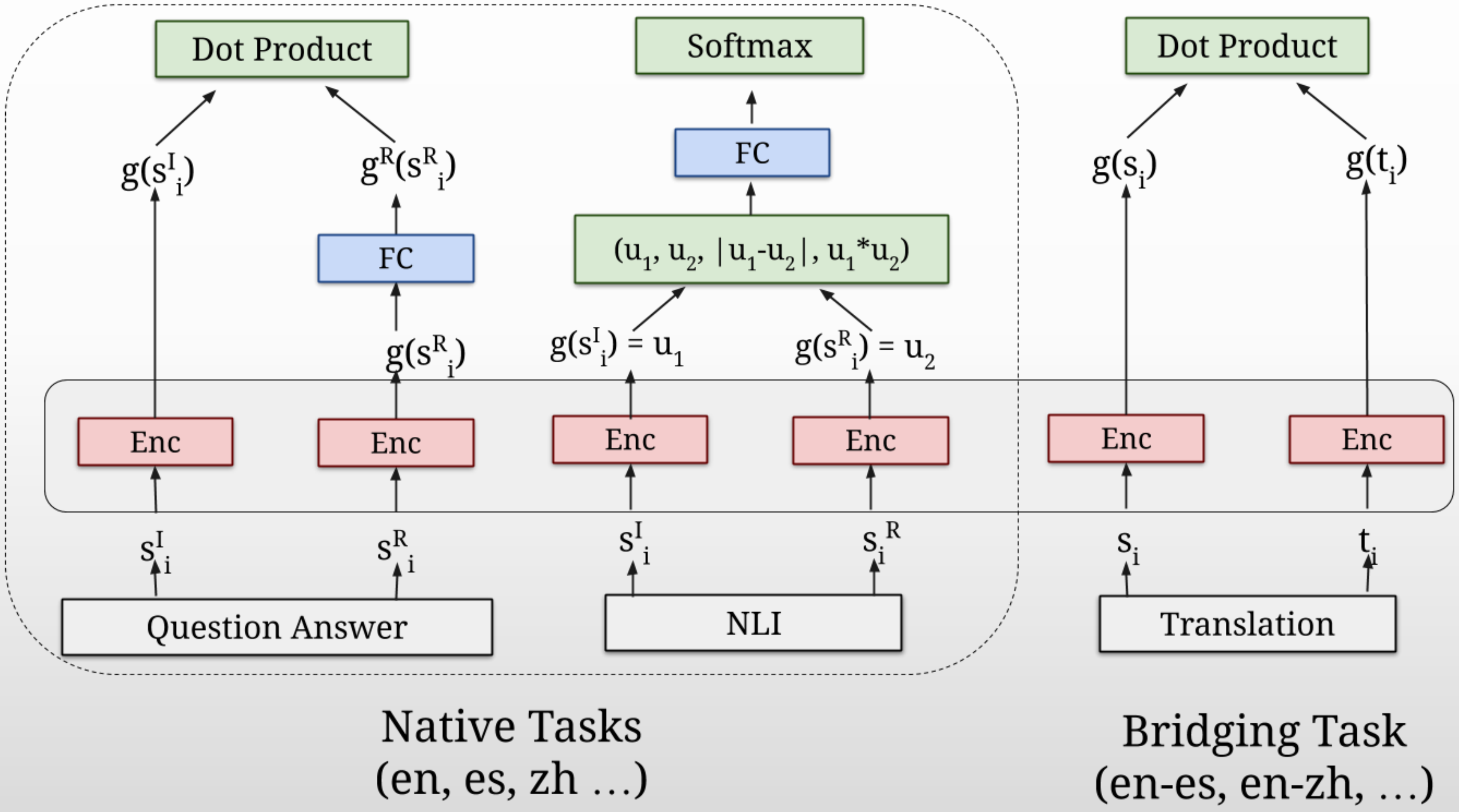


Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

https://amitness.com/2020/06/universal-sentence-encoder/

Multilingual USE

- Can be trained to translate a presentation into several languages.

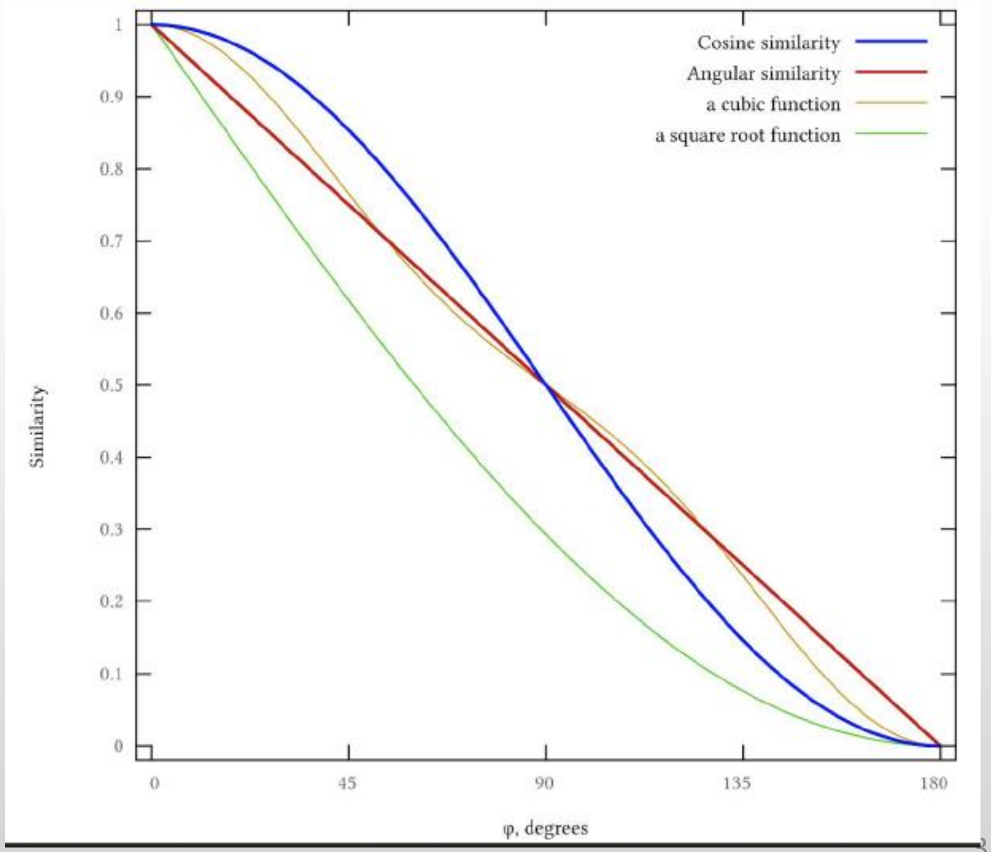


https://ai.googleblog.com/2019/07/multilingual-universal-sentence-encoder.html

Measuring distance between vectors

- Use angular similarity (arccos) rather than cosine similarity

$$sim(u,v) = \left(1 - \frac{arccoss\left(\frac{u \cdot v}{|u||v|}\right)}{\pi}\right)$$



Download USE

TensorFlow Hub

Search for models, collections & publishers

← Back

Model	Comments
universal-sentence-encoder	
universal-sentence-encoder-large	
universal-sentence-encoder-lite	
universal-sentence-encoder-qa	Question answering
universal-sentence-encoder-multilingual	16 languages
universal-sentence-encoder-multilingual-large	16 languages
universal-sentence-encoder-multilingual-qa	16 languages , Question answering

https://tfhub.dev/google/collections/universal-sentence-encoder/1

Unsupervised pre-training: Benchmarks

prachathai-67k: body_text

We benchmark prachathai-67k by using `body_text` as text features and construct a 12-label multi-label classification. The performance is measured by macro-averaged accuracy and F1 score. Codes can be run to confirm performance at this notebook. We also provide performance metrics by class in the notebook.

model	macro-accuracy	macro-F1
fastText	0.9302	0.5529
LinearSVC	0.513277	0.552801
ULMFit	**0.948737**	**0.744875**
USE	0.856091	0.696172

https://github.com/PyThaiNLP/classification-benchmarks

Unsupervised pre-training: Benchmarks (cont.)

truevoice-intent: destination

We benchmark truevoice-intent by using `destination` as target and construct a 7-class multi-class classification. The performance is measured by micro-averaged and macro-averaged accuracy and F1 score. Codes can be run to confirm performance at this notebook. We also provide performance metrics by class in the notebook.

model	macro-accuracy	micro-accuracy	macro-F1	micro-F1
LinearSVC	**0.957806**	**0.95747712**	**0.869411**	**0.85116993**
ULMFit	0.955066	0.84273111	0.852149	0.84273111
BERT	0.8921	0.85	0.87	0.85
USE	0.943559	0.94355855	0.787686	0.802455

https://github.com/PyThaiNLP/classification-benchmarks

Unsupervised pre-training: Benchmarks (cont.)

wongnai-corpus

Performance of wongnai-corpus is based on the test set of Wongnai Challenge: Review Rating Prediction. Codes can be run to confirm performance at this notebook.

Model	Public Micro-F1	Private Micro-F1
ULMFit Knight	**0.61109**	**0.62580**
ULMFit	0.59313	0.60322
fastText	0.5145	0.5109
LinearSVC	0.5022	0.4976
Kaggle Score	0.59139	0.58139
BERT	0.56612	0.57057
USE	0.42688	0.41031

https://github.com/PyThaiNLP/classification-benchmarks

Relationship to language modeling

How to find ?

Word	Probability	Distribution of Class 5
ชอบ	$P(\text{ชอบ}\|c=5)$	0.3
อร่อย	$P(\text{อร่อย}\|c=5)$	0.3
ไม่	$P(\text{ไม่}\|c=5)$	0.05
กลมกล่อม	$P(\text{กลมกล่อม}\|c=5)$	0.25
ทานง่าย	$P(\text{ทานง่าย}\|c=5)$	0.1

- $P(x|c)$
- $P(x = \text{"ชอบ"}|c = 5) = \frac{count(x=\text{"ชอบ"},c=5)}{count(c=5)}$
- $P(c)$
- $P(c = 5) = \frac{count(c=5)}{count(all\ reviews)}$

- Looks like… n-grams
- Bag of words model for topic modeling (**unigram with topic**)

Relationship to language modeling (cont.)

Word	Distribution of Class 5	Distribution of Class 1
ชอบ	0.3	0.05
อร่อย	0.3	0.05
ไม่	0.05	0.6
กลมกล่อม	0.25	0.1
ทานง่าย	0.05	0.1
แต่	0.05	0.1

- Example: S = อร่อยทานง่ายแต่ไม่กลมกล่อม
  - $P(S|c = 1) = 0.05 \times 0.1 \times 0.1 \times 0.6 \times 0.1$
  - $= 0.00003$
  - $P(S|c = 5) = 0.3 \times 0.05 \times 0.05 \times 0.05 \times 0.25$
  - $= 0.000009375$

Topic Modeling

Word	Class = บรรยากาศ	Class = อาหาร
ชอบ	0.3	0.05
อร่อย	0.3	0.05
ไม่	0.05	0.6
กลมกล่อม	0.25	0.1
ทานง่าย	0.05	0.1
แต่	0.05	0.1

Topic

อาหารร้านนี้อร่อยทานง่ายแต่รสชาติยังไม่กลมกลม กล่อม แต่ฉันชอบขนมมากรสชาติดีแต่ว่าให้น้อย

การบริการยังไม่น่าประทับใจ แต่ชอบการตกแต่งร้านที่ทันสมัย

$$P(S|c = \text{บรรยากาศ}) = ?$$
$$P(S|c = \text{อาหาร}) = ?$$

Naïve Bayes for Topic Modeling

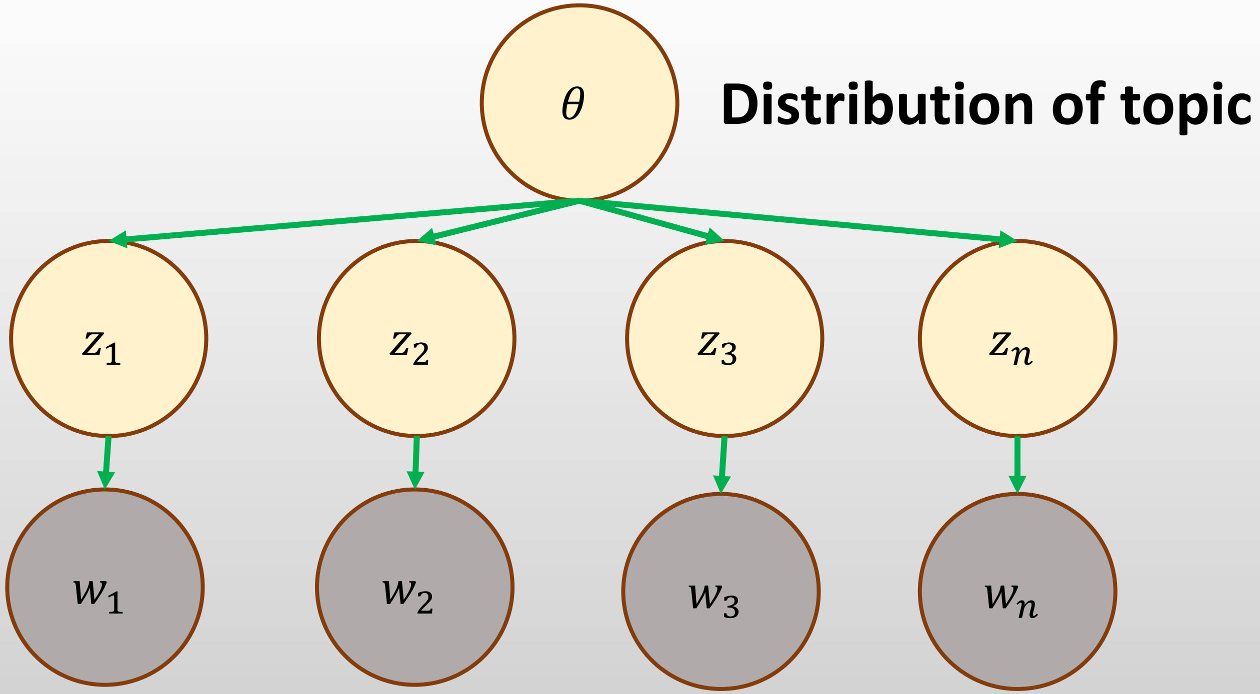
- Old assumption is 1 document 1 topic (multi-classes).
- Let a document be a mixture of topics (multi-labels).
- Each word has its own topic (z)
- There are 2 different topics (A and B)

- $P(w) = P(z = A)P(w|z = A) + P(z = B)P(w|z = B)$
- $P(z = A) + P(z = B) = 1$

Naïve Bayes for Topic Modeling (cont.)

- Old assumption is 1 document 1 topic (multi-classes).
- Let a document be a mixture of topics (multi-labels).
- There are 2 different topics (A and B)

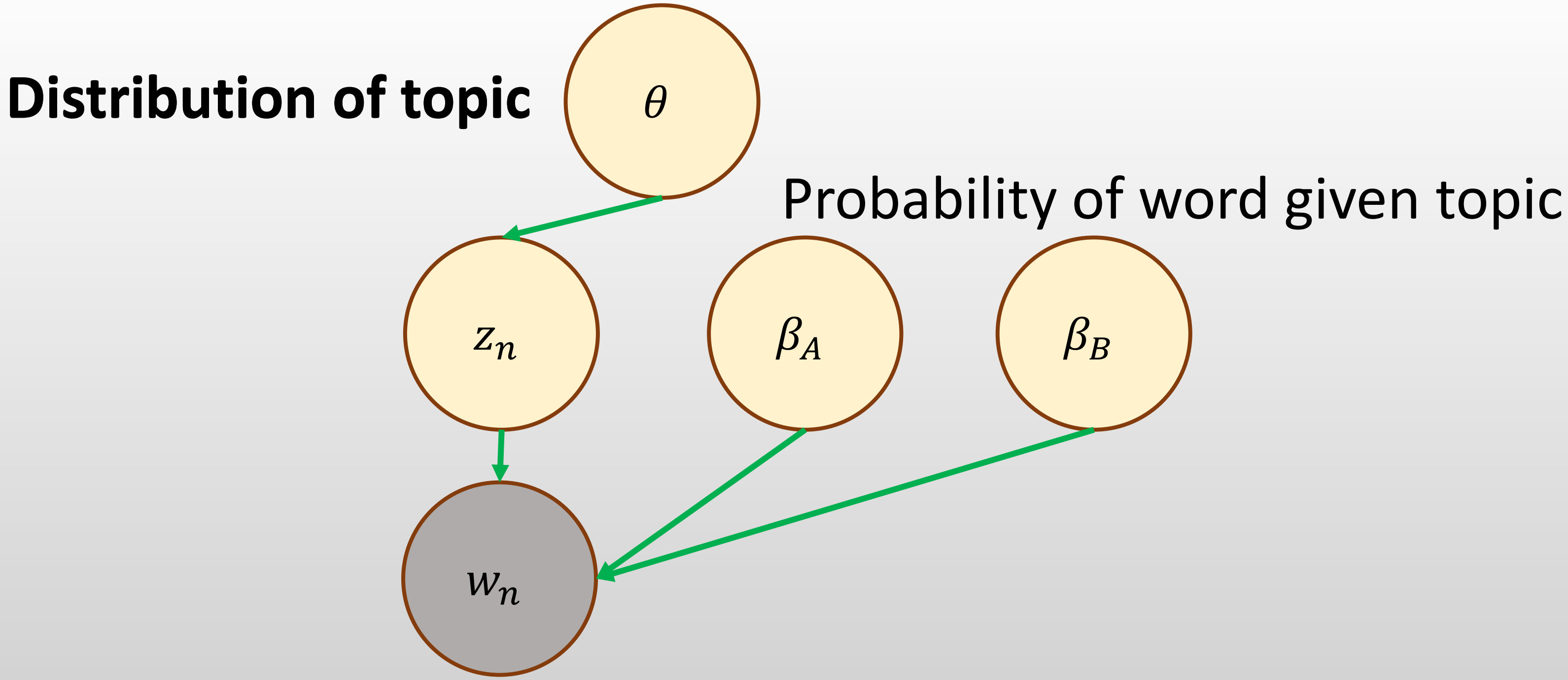
- $P(w) = P(z = A)P(w|z = A) + P(z = B)P(w|z = B)$
- $P(z = A) + P(z = B) = 1,\ \theta = P(z = A)$



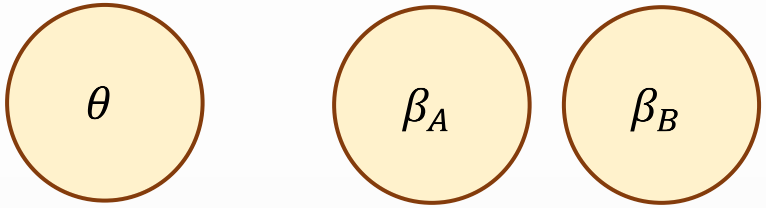
Naïve Bayes for Topic Modeling (cont.)

- Old assumption is 1 document 1 topic (multi-classes).
- Let a document be a mixture of topics (multi-labels).
- There are 2 different topics (A and B)

- $P(w) = P(z = A)P(w|z = A) + P(z = B)P(w|z = B)$
- $P(z = A) + P(z = B) = 1,\ \theta = P(z = A), \beta_A = P(w|z = A)$



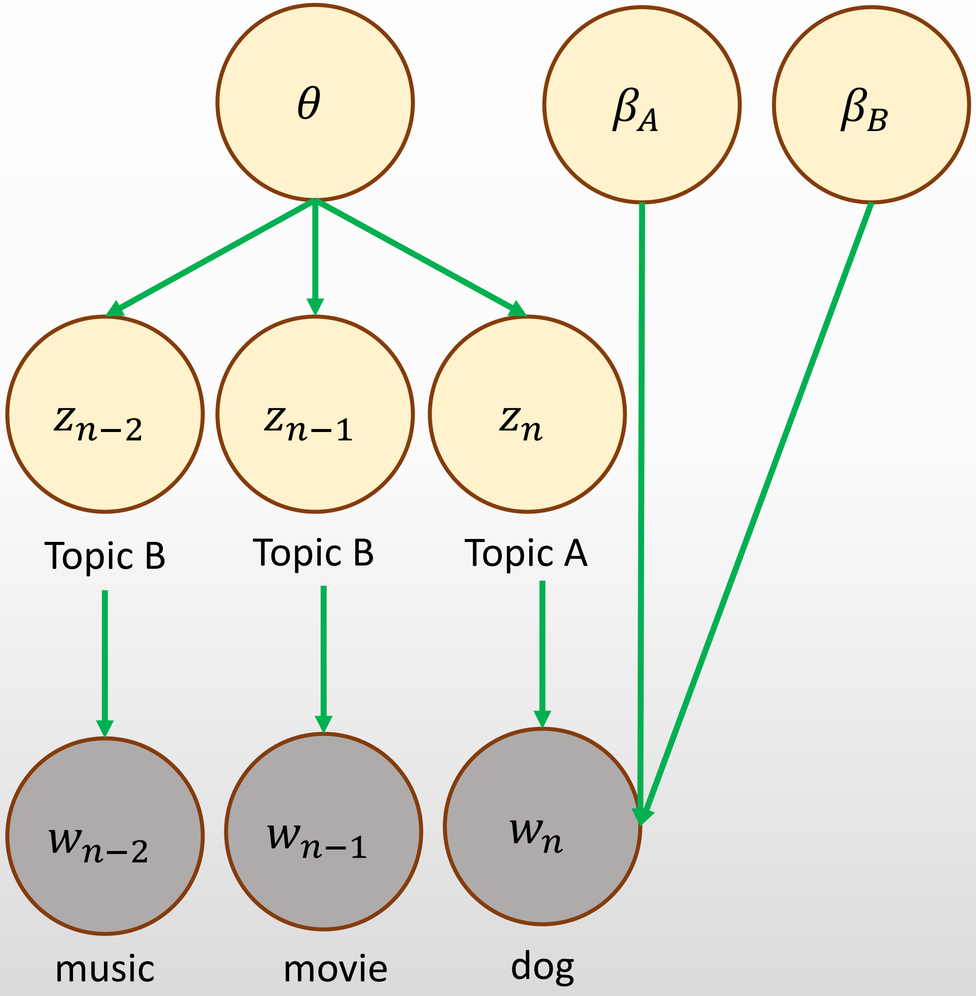
Naïve Bayes for Topic Modeling (cont.)



- Example: Given
  - $\theta; P(z = A) = 0.3, P(z = B) = 0.7$
  - $\beta_A: P(w = cat|z = A) = 0.5, P(w = dog|z = A) = 0.5$
  - $\beta_B: P(w = movie|z = B) = 0.7, P(w = music|z = B) = 0.3$

Naïve Bayes for Topic Modeling (cont.)

- Example: Given
  - $\theta; P(z = A) = 0.3, P(z = B) = 0.7$
  - $\beta_A: P(w = cat|z = A) = 0.5, P(w = dog|z = A) = 0.5$
  - $\beta_B: P(w = movie|z = B) = 0.7, P(w = music|z = B) = 0.3$
- What is the probability of $P(music\ movie\ dog, B, B\ A)$ and $P(music\ movie\ dog)$?



Naïve Bayes for Topic Modeling (cont.)

- Example: Given
  - $\theta; P(z = A) = 0.3, P(z = B) = 0.7$
  - $\beta_A: P(w = cat|z = A) = 0.5, P(w = dog|z = A) = 0.5$
  - $\beta_B: P(w = movie|z = B) = 0.7, P(w = music|z = B) = 0.3$
- What is the probability of $P(music\ movie\ dog, B, B\ A)$ and $P(music\ movie\ dog)$?
- $P(music\ movie\ dog, B, B\ A) = P(B)P(B)P(A)P(music|B)P(movie|B)P(dog|A)$
- $P(music\ movie\ dog) =$

$P(music\ movie\ dog, A, A, A) +$
$P(music\ movie\ dog, A, A, B) +$
$P(music\ movie\ dog, A, B, A) + \cdots +$
$P(music\ movie\ dog, B, B, B)$

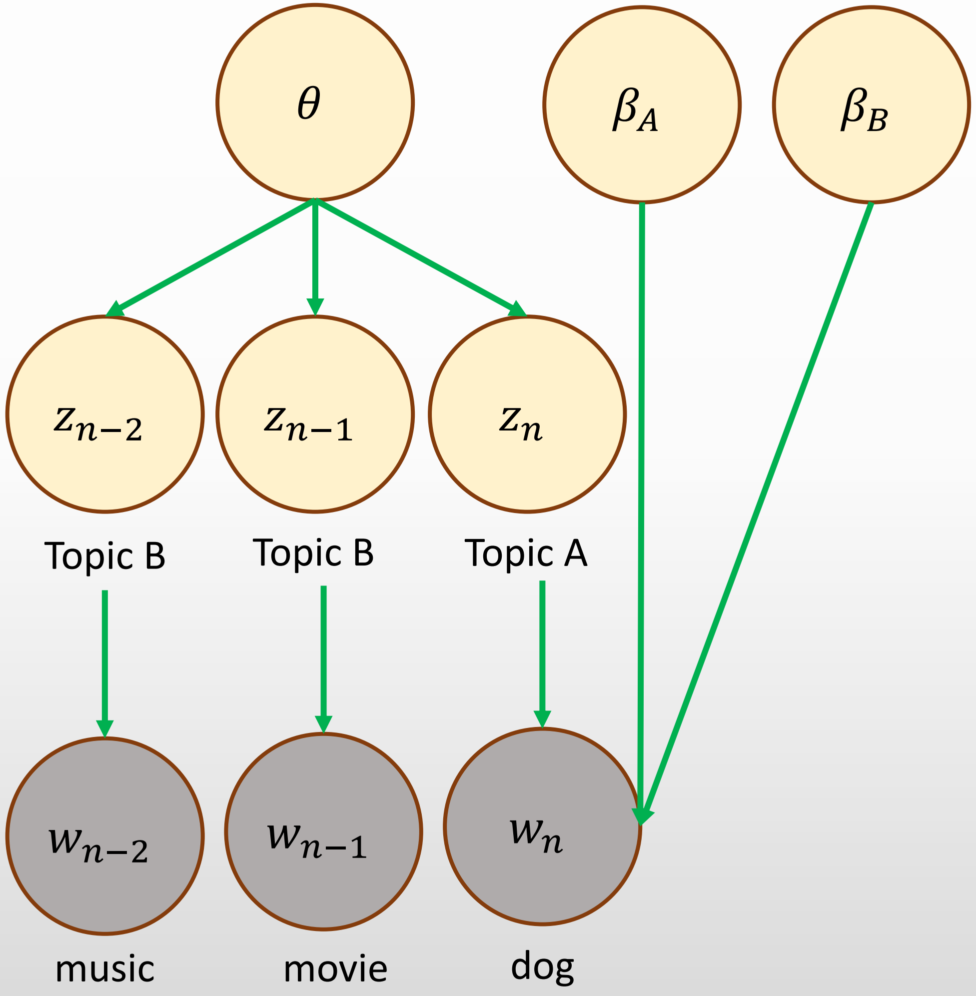


Plate notation in Graphical model

- Summarize by using a square box with number

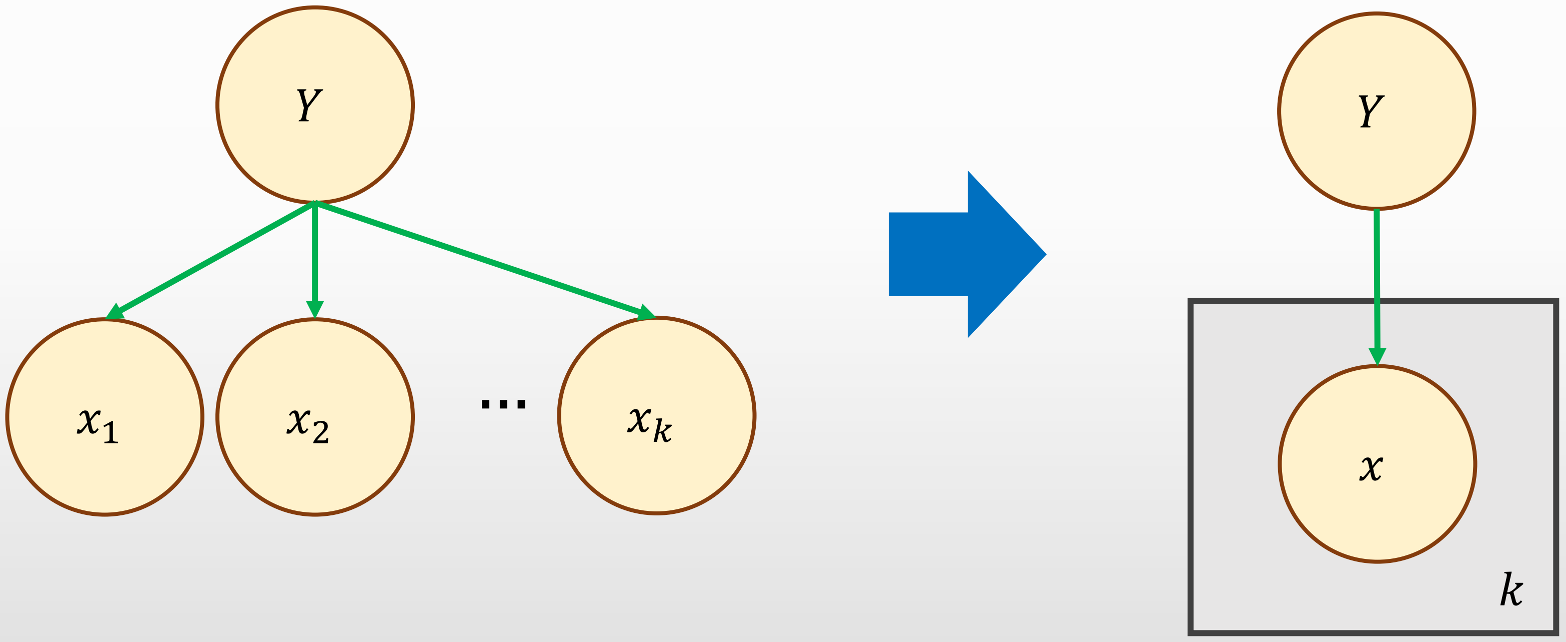
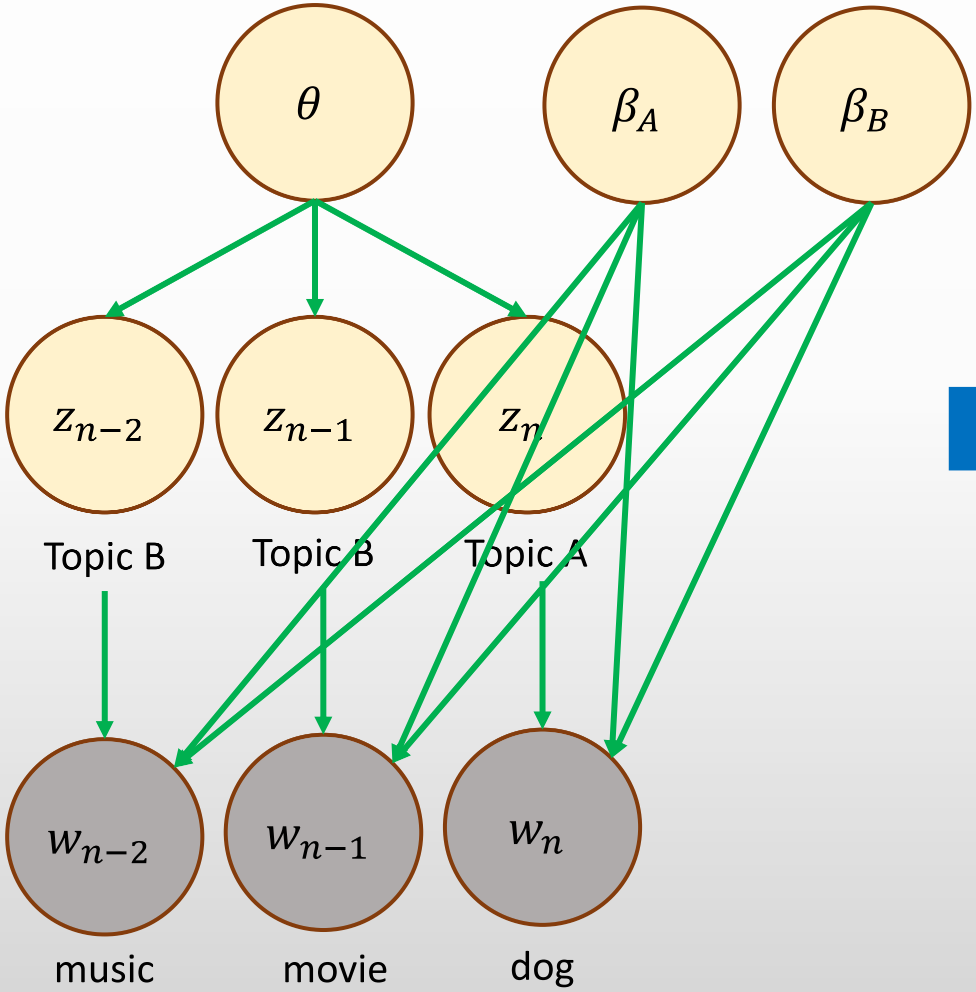
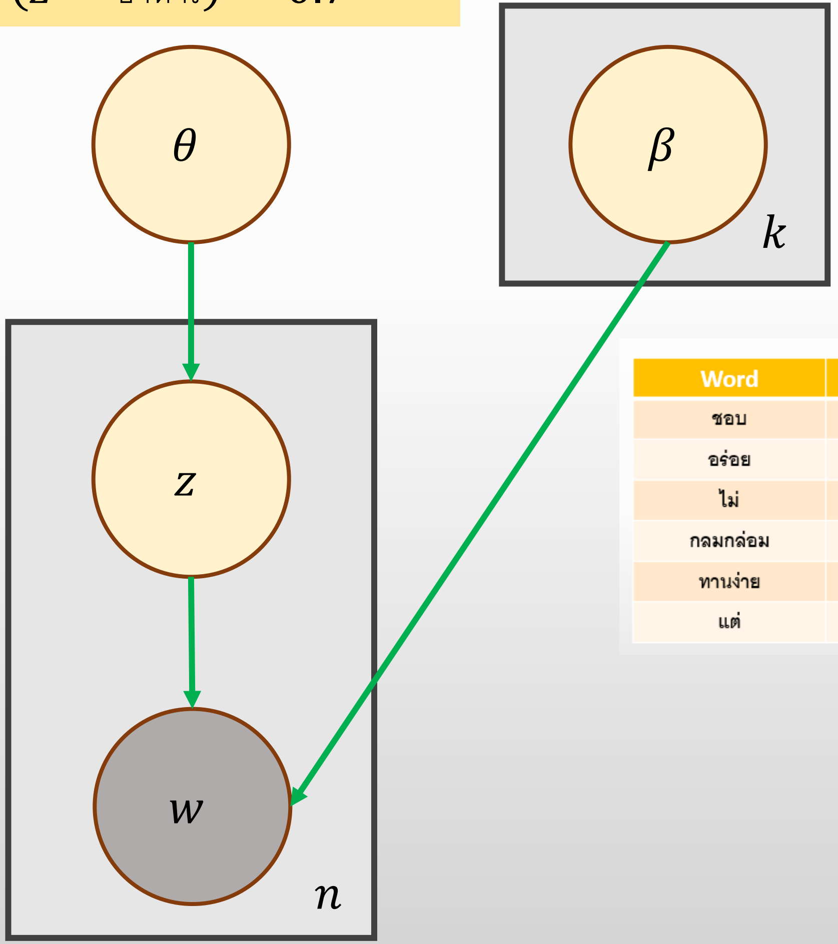


Plate notation in Graphical model (cont.)



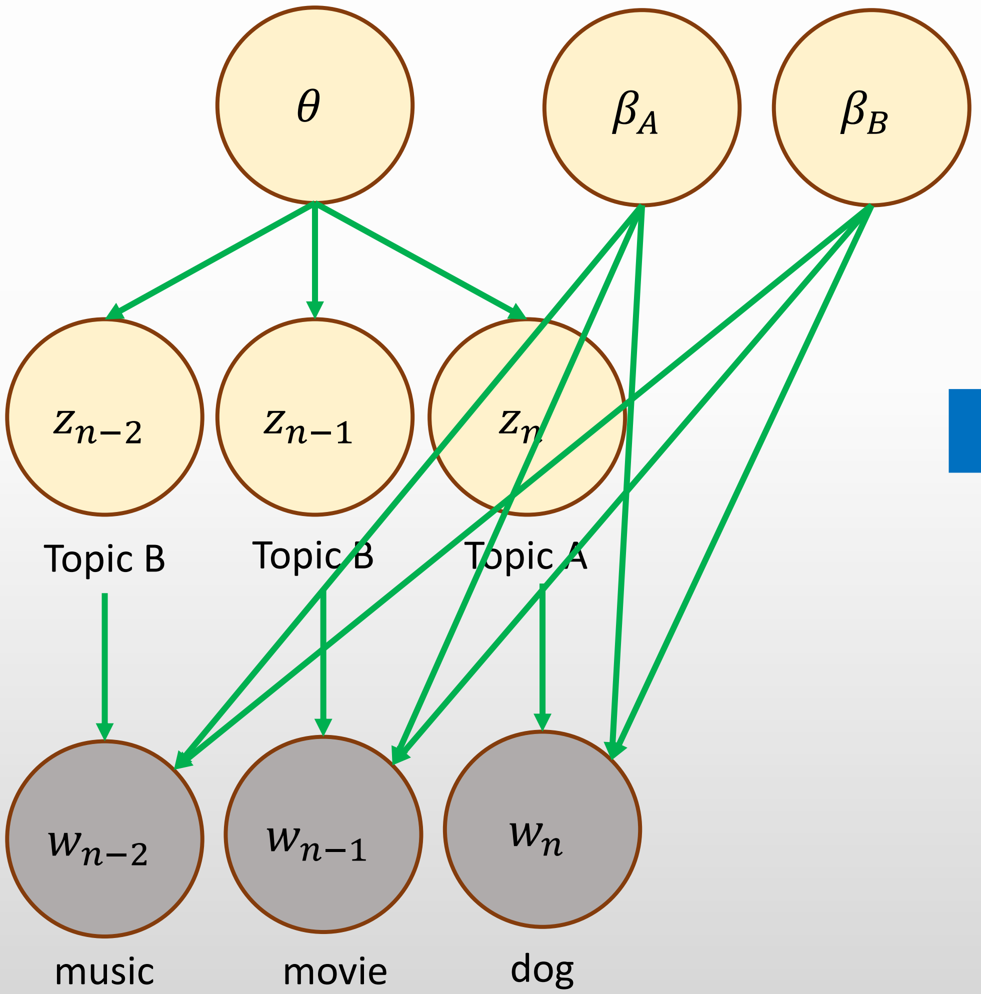
$$P(z = \text{บรรยากาศ}) = 0.3$$
$$P(z = \text{อาหาร}) = 0.7$$

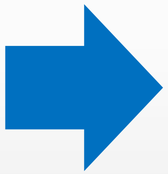


Word	Class = บรรยากาศ	Class = อาหาร
ชอบ	0.3	0.05
อร่อย	0.3	0.05
ไม่	0.05	0.6
กลมกล่อม	0.25	0.1
ทานง่าย	0.05	0.1
แต่	0.05	0.1

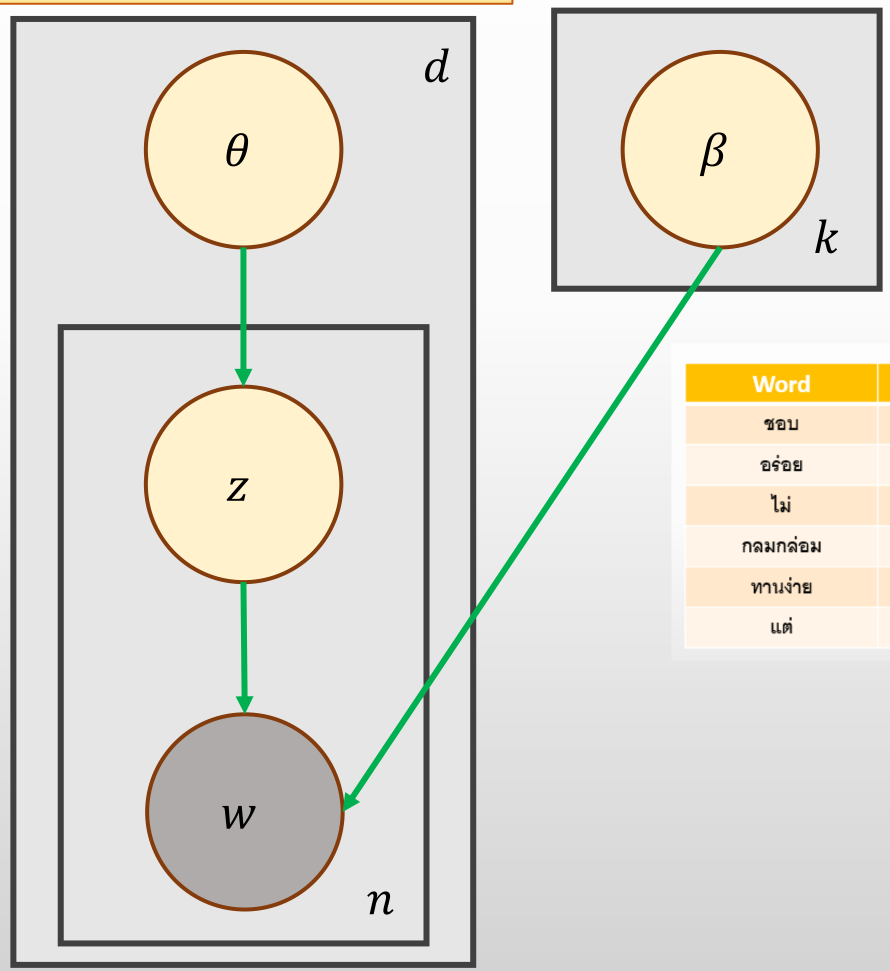
$P(w|z)$

Plate notation in Graphical model (cont.)





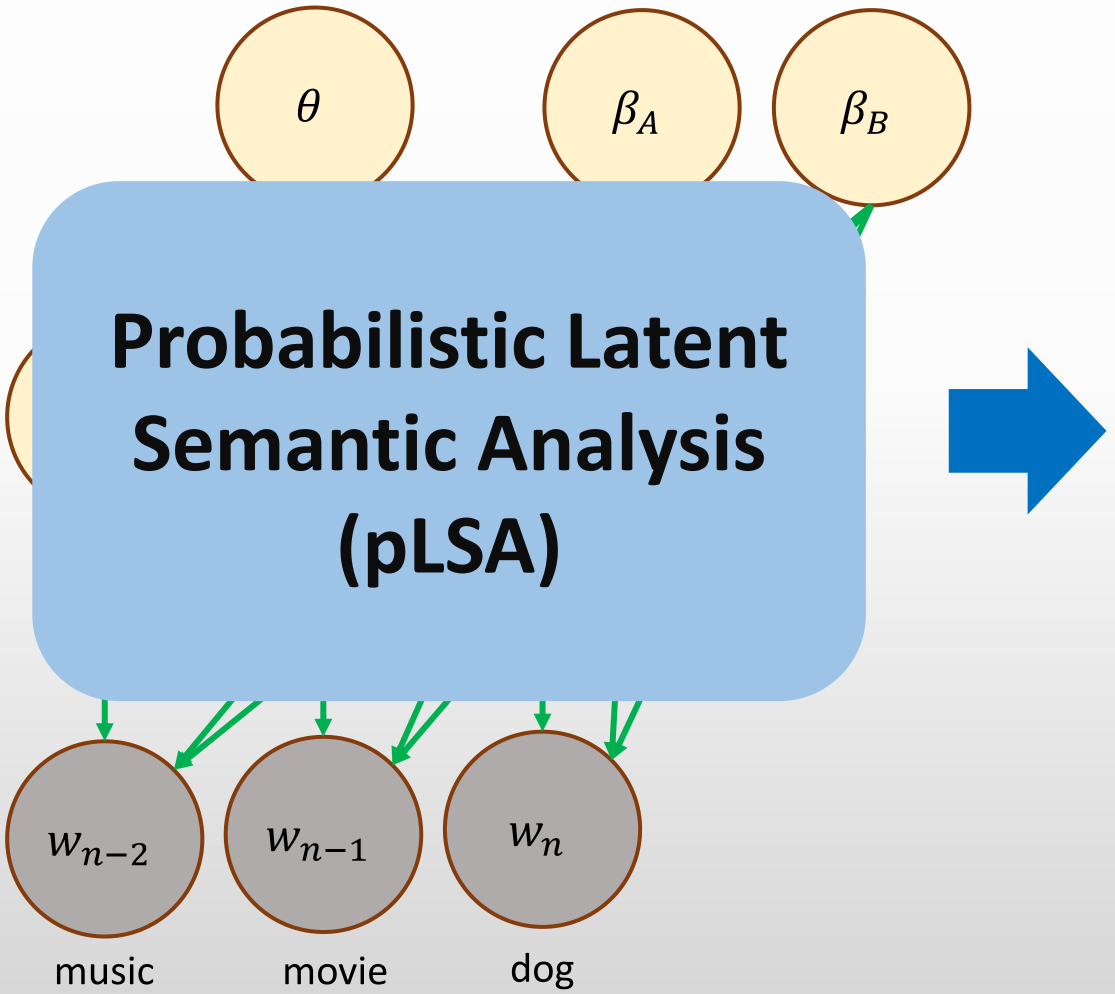
$P(z = \text{บรรยากาศ}) = 0.3$
$P(z = \text{อาหาร}) = 0.7$



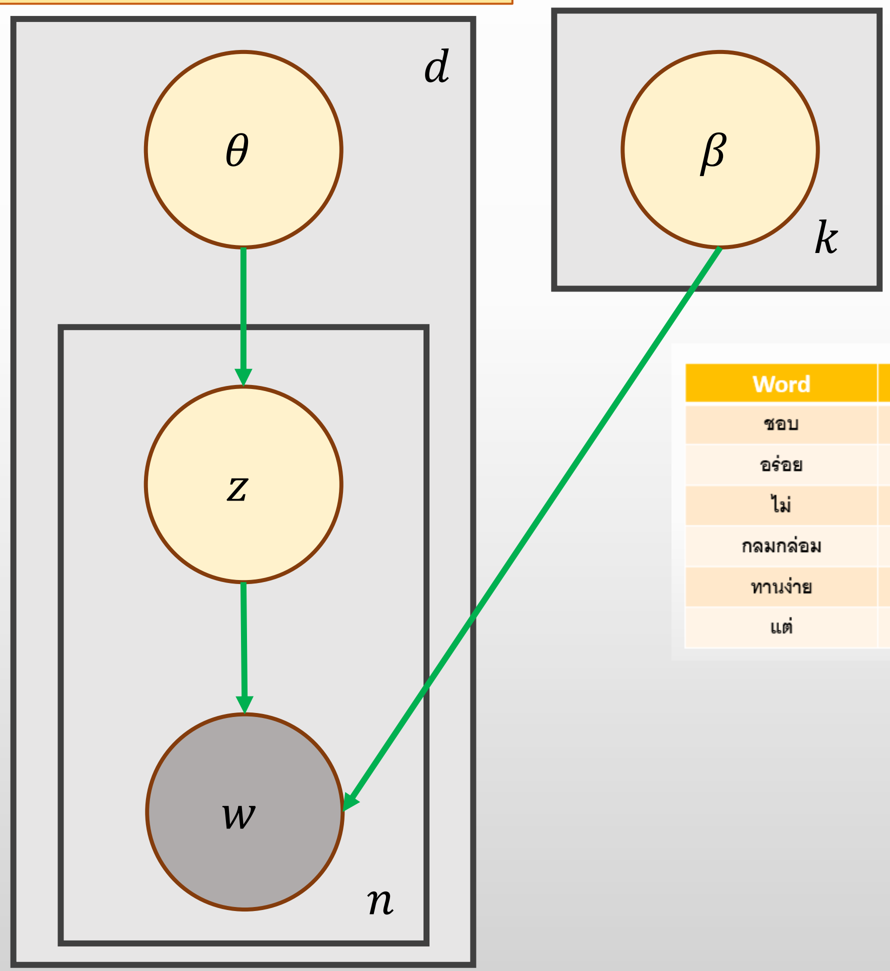
Word	Class = บรรยากาศ	Class = อาหาร
ชอบ	0.3	0.05
อร่อย	0.3	0.05
ไม่	0.05	0.6
กลมกล่อม	0.25	0.1
ทานง่าย	0.05	0.1
แต่	0.05	0.1

$P(w|z)$

Probabilistic Latent Semantic Analysis (pLSA)



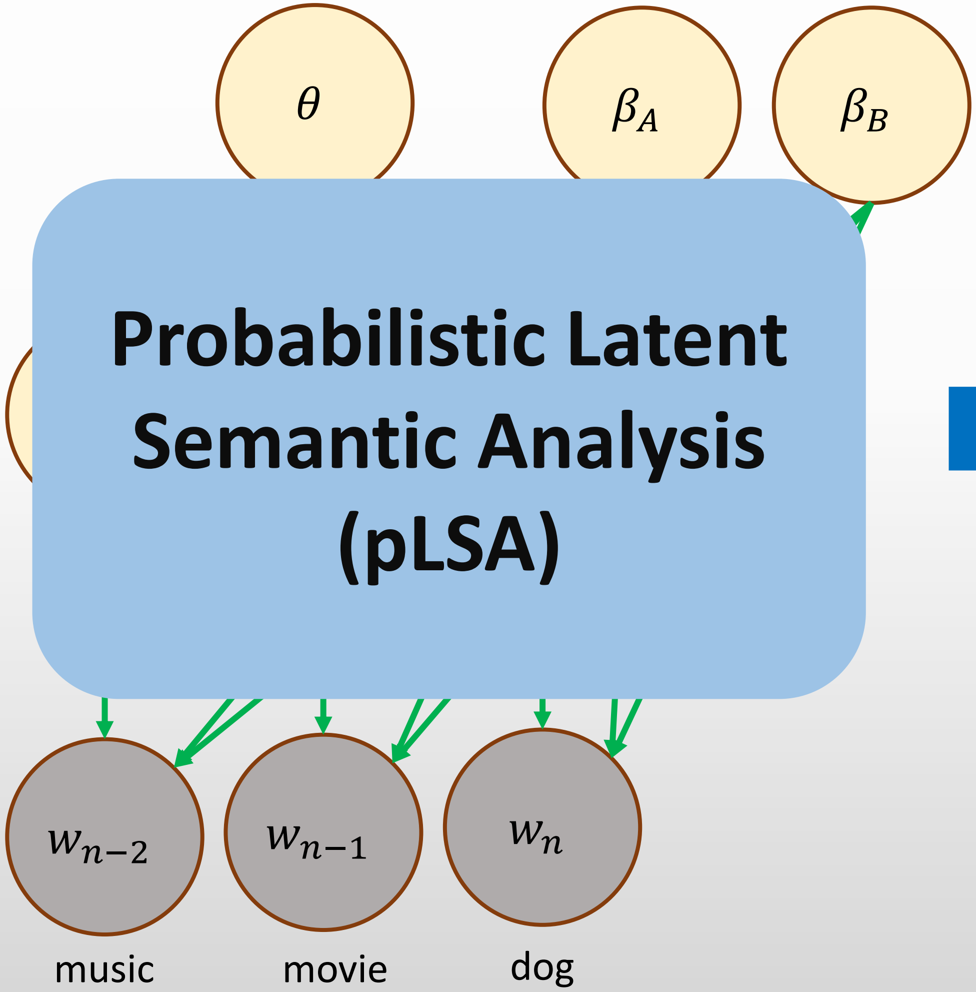
$P(z = \text{บรรยากาศ}) = 0.3$
$P(z = \text{อาหาร}) = 0.7$



Word	Class = บรรยากาศ	Class = อาหาร
ชอบ	0.3	0.05
อร่อย	0.3	0.05
ไม่	0.05	0.6
กลมกล่อม	0.25	0.1
ทานง่าย	0.05	0.1
แต่	0.05	0.1

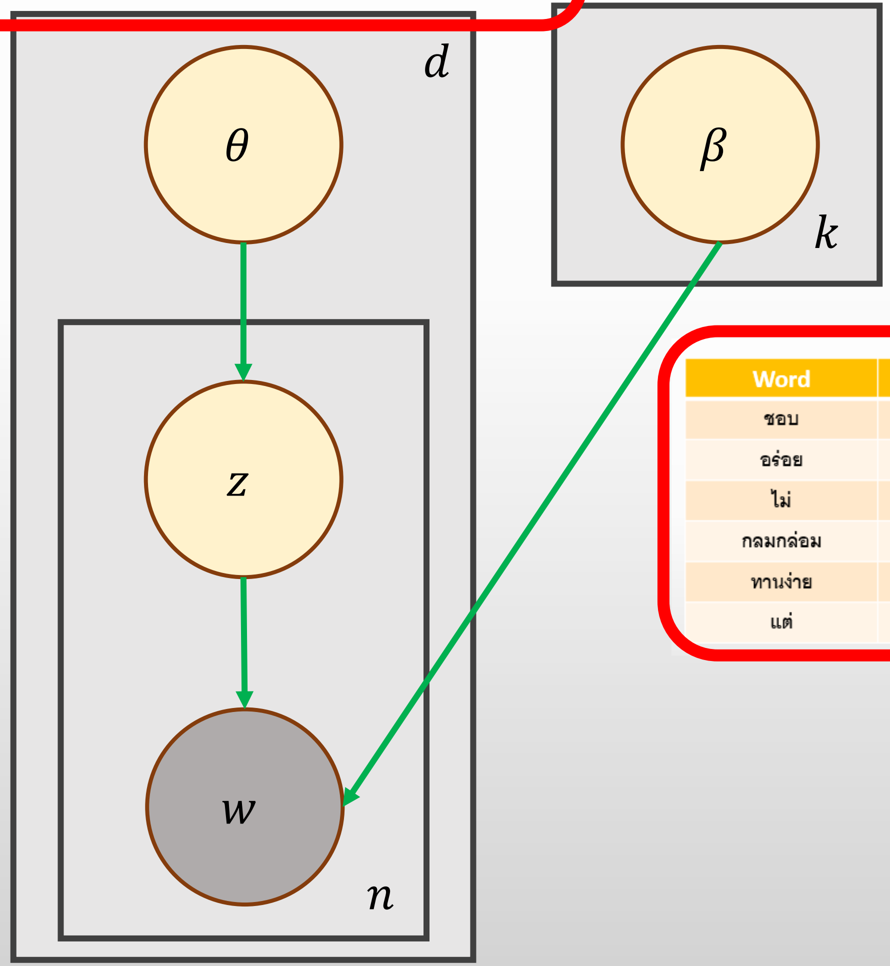
$P(w|z)$

Probabilistic Latent Semantic Analysis (pLSA) (cont.)



$P(z = \text{บรรยากาศ}) = 0.3$
$P(z = \text{อาหาร}) = 0.7$

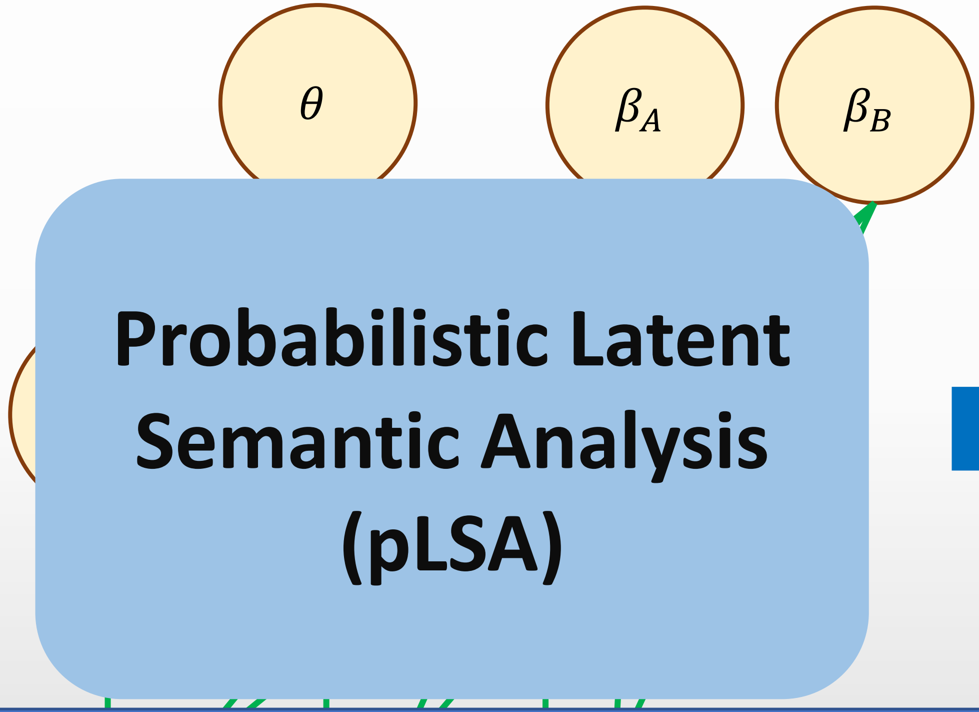
How to find θ and β?



Word	Class = บรรยากาศ	Class = อาหาร
ชอบ	0.3	0.05
อร่อย	0.3	0.05
ไม่	0.05	0.6
กลมกล่อม	0.25	0.1
ทานง่าย	0.05	0.1
แต่	0.05	0.1

$P(w|z)$

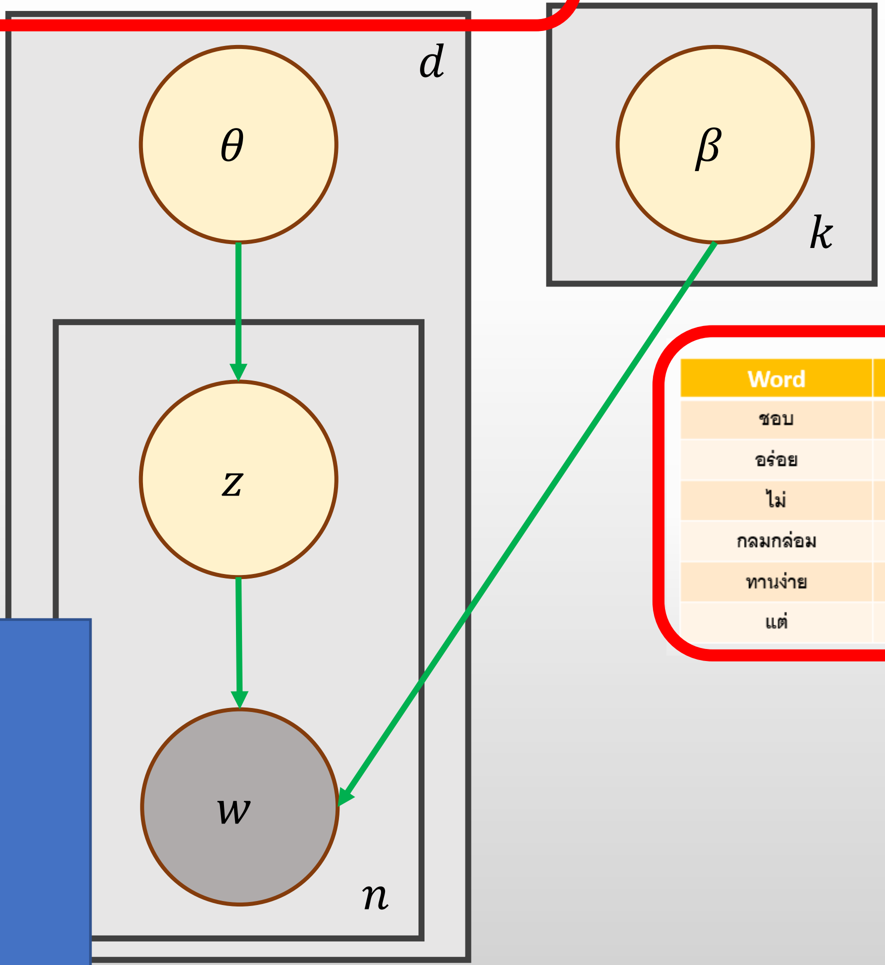
Probabilistic Latent Semantic Analysis (pLSA) (cont.)



Probabilistic Latent Semantic Analysis (pLSA)

$P(z = \text{บรรยากาศ}) = 0.3$
$P(z = \text{อาหาร}) = 0.7$

How to find θ and β?



Word	Class = บรรยากาศ	Class = อาหาร
ชอบ	0.3	0.05
อร่อย	0.3	0.05
ไม่	0.05	0.6
กลมกล่อม	0.25	0.1
ทานง่าย	0.05	0.1
แต่	0.05	0.1

$P(w|z)$

- Supervised learning
  - $\rightarrow P(\text{ชอบ}|\text{บรรยากาศ}) = \dfrac{count(\text{ชอบ,บรรยากาศ})}{count(\text{บรรยากาศ})}$
  - $\rightarrow P_1(\text{บรรยากาศ}) = \dfrac{count(\text{บรรยากาศ})}{count\ (all\ word\ in\ doc1)}$
- Unsupervised learning (like word2vec, Skip-thought, etc.)???

Expectation maximization (EM)

- A method to iteratively maximize the likelihood of a model on training data
  - Initialize θ, β
  - Expectation step (E-step): guess latent variables from model parameters (get soft counts)
  - Maximization step (M-step): re-estimate model parameters from latent variables (counts)
  - Update θ, β
  - Repeat E and M step until satisfied (Likelihood of the whole training set using the model does not change much)

Expectation maximization (EM): E-step

- Find an estimate for the latent variable given parameters θ and β
- Iterative algorithm: assume distribution of θ and β
- Try to find $P(z_{di}|w_{di}, \theta, \beta)$

Parameter θ,β

Topic of word i document d

Topic of word i document d

Expectation maximization (EM): E-step (cont.)

- Find an estimate for the latent variable given parameters θ and β
- Iterative algorithm: assume distribution of θ and β
- Try to find $P(z_{di}|w_{di}, \theta, \beta)$

Parameter θ,β

Topic of word i document d

Topic of word i document d

$$P(X|Y) = \frac{P(X \text{ and } Y)}{P(Y)}$$

$$P(z_{di}|w_{di}, \theta, \beta) = \frac{P(z_{di}, w_{di}, \theta, \beta)}{P(w_{di}, \theta, \beta)} = \frac{P(z_{di}, w_{di}, \theta, \beta)}{\sum_{z'}^{k} P(z'_{di}, w_{di}, \theta, \beta)} = \frac{\theta_{z|d}\beta_{w|z}}{\sum_{z'}^{k} \theta_{z'|d}\beta_{w|z'}}$$

k: # of topic

Expectation maximization (EM): E-step (cont.)

- Find an estimate for the latent variable given parameters θ and β
- Iterative algorithm: assume distribution of θ and β
- Try to find $P(z_{di}|w_{di}, \theta, \beta)$

Parameter θ,β

Topic of word i document d

Topic of word i document d

$$P(X|Y) = \frac{P(X \text{ and } Y)}{P(Y)}$$

$$P(z_{di}|w_{di}, \theta, \beta) = \frac{P(z_{di}, w_{di}, \theta, \beta)}{P(w_{di}, \theta, \beta)} = \frac{P(z_{di}, w_{di}, \theta, \beta)}{\sum_{z'}^{k} P(z'_{di}, w_{di}, \theta, \beta)} = \frac{\theta_{z|d}\beta_{w|z}}{\sum_{z'}^{k} \theta_{z'|d}\beta_{w|z'}}$$

k: # of topic

Example: P(Word is from topic A| word is cat from document 1)

Expectation maximization (EM): M-step

- Instead of real counts by $P(z_{di})$ as the topic label
  - Example:
  - $P(Cat|A) = \frac{count(Cat,A)}{count(A)} = \frac{\sum_{d'}^{d} P(z_{d'i} = A|w_{d'i} = cat, \theta, \beta)}{\sum_{d'}^{d} \sum_{w'}^{n} P\left(z_{d'i} = A\middle|w'_{d'i}, \theta, \beta\right)}$
  - $P_1(A) = \frac{count(A)}{count(all\ words\ in\ doc1)} = \sum_{w'}^{n} P\left(z_{d=1,i} = A\middle|w_{d=1,i}, \theta, \beta\right)$

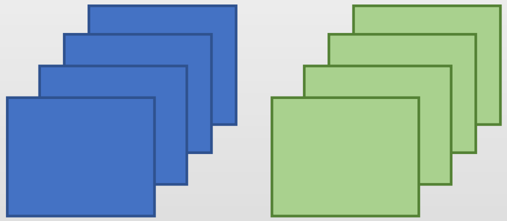
Expectation maximization (EM): M-step (cont.)

- Instead of real counts by $P(z_{di})$ as the topic label
  - Example:
  - $P(Cat|A) = \frac{count(Cat,A)}{count(A)} = \frac{\sum_{d'}^{d} P(z_{d'i} = A|w_{d'i} = cat, \theta, \beta)}{\sum_{d'}^{d} \sum_{w'}^{n} P(z_{d'i} = A|w_{d'i}, \theta, \beta)}$
  - $P_1(A) = \frac{count(A)}{count(all\ words\ in\ doc1)} = \sum_{w'}^{n} P\left(z_{d=1,i} = A \middle| w'_{d=1,i}, \theta, \beta\right)$

pLSA

- Automatically learn document representation based on the learned topics.
- Nothing that ties all document together.
- A document from a document collection should be have topic distributions that are similar.

Solution → Latent Dirichlet Allocation (LDA)

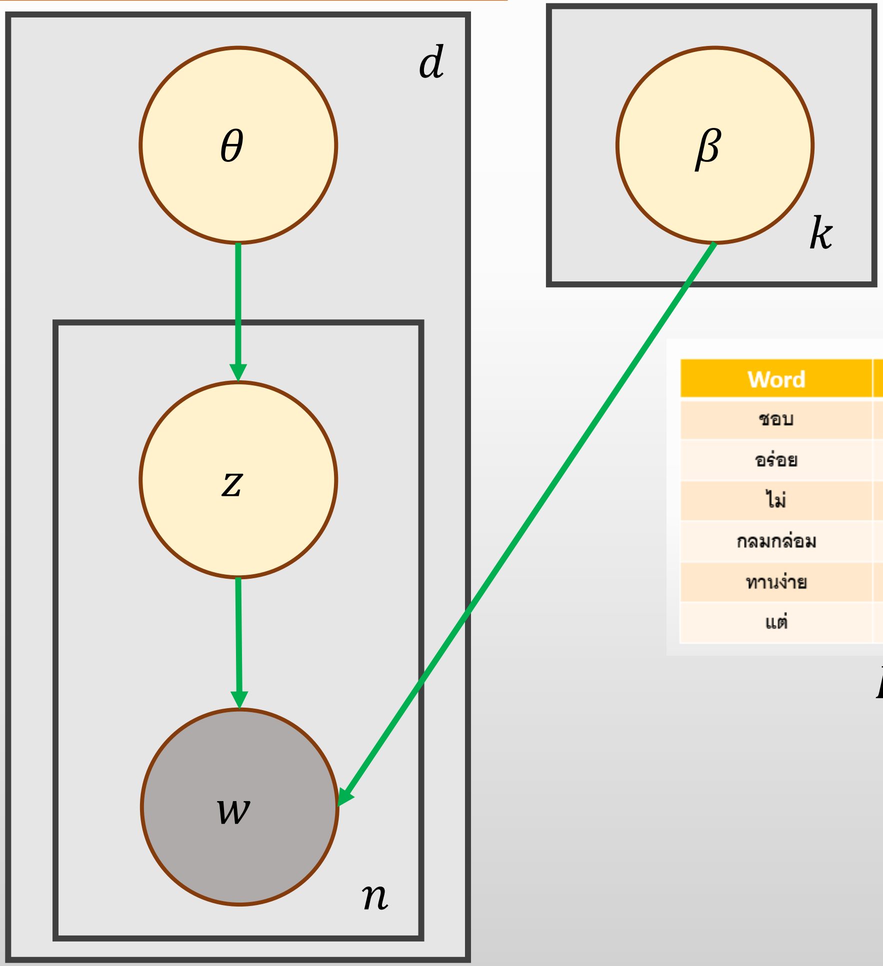


Pantip Wikipedia

Latent Dirichlet Allocation (LDA)

- General pLSA

$$P(z = \text{บรรยากาศ}) = 0.3$$
$$P(z = \text{อาหาร}) = 0.7$$



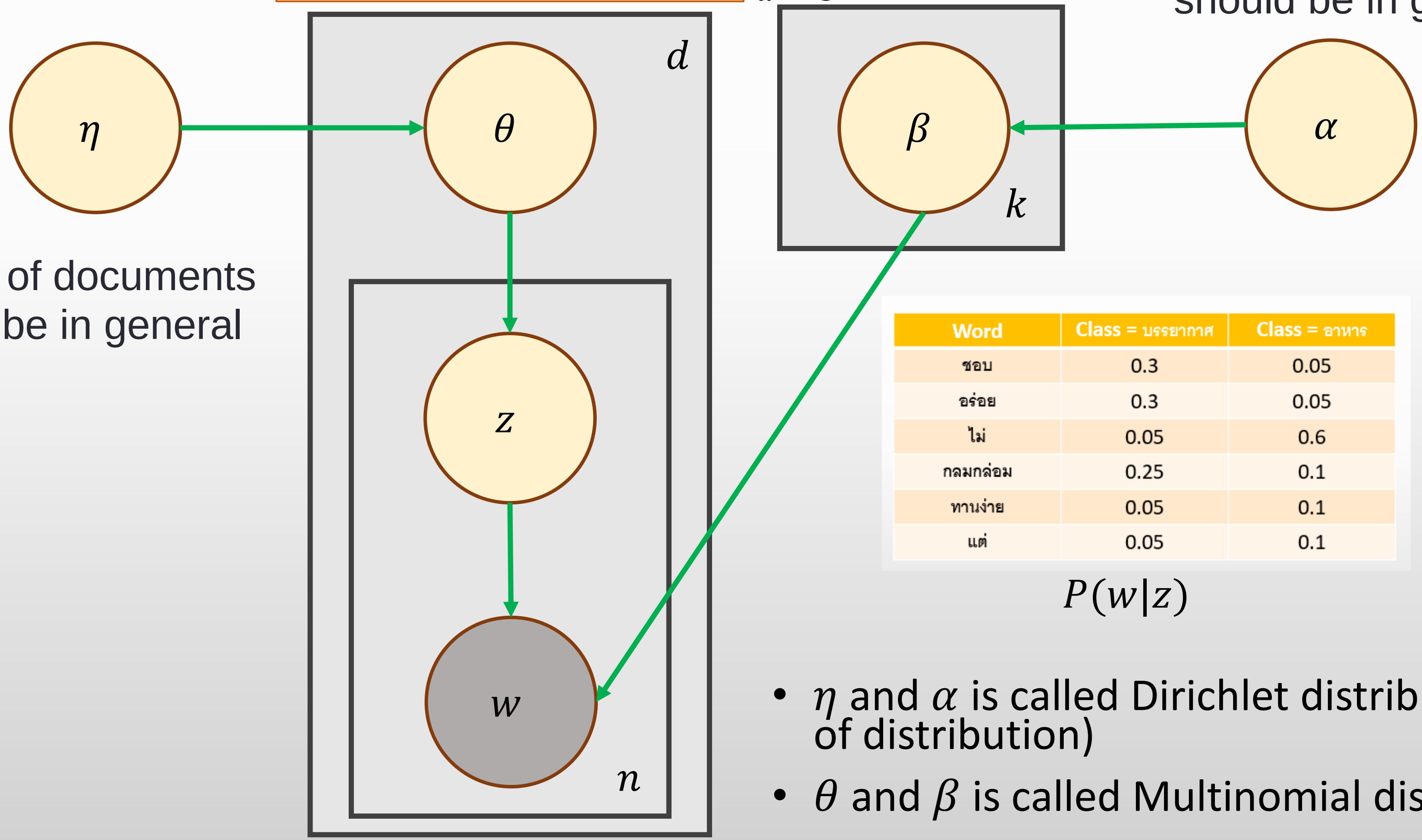
Word	Class = บรรยากาศ	Class = อาหาร
ชอบ	0.3	0.05
อร่อย	0.3	0.05
ไม่	0.05	0.6
กลมกล่อม	0.25	0.1
ทานง่าย	0.05	0.1
แต่	0.05	0.1

$P(w|z)$

Latent Dirichlet Allocation (LDA) (cont.)

- pLSA with LDA

$P(z = \text{บรรยากาศ}) = 0.3$
$P(z = \text{อาหาร}) = 0.7$



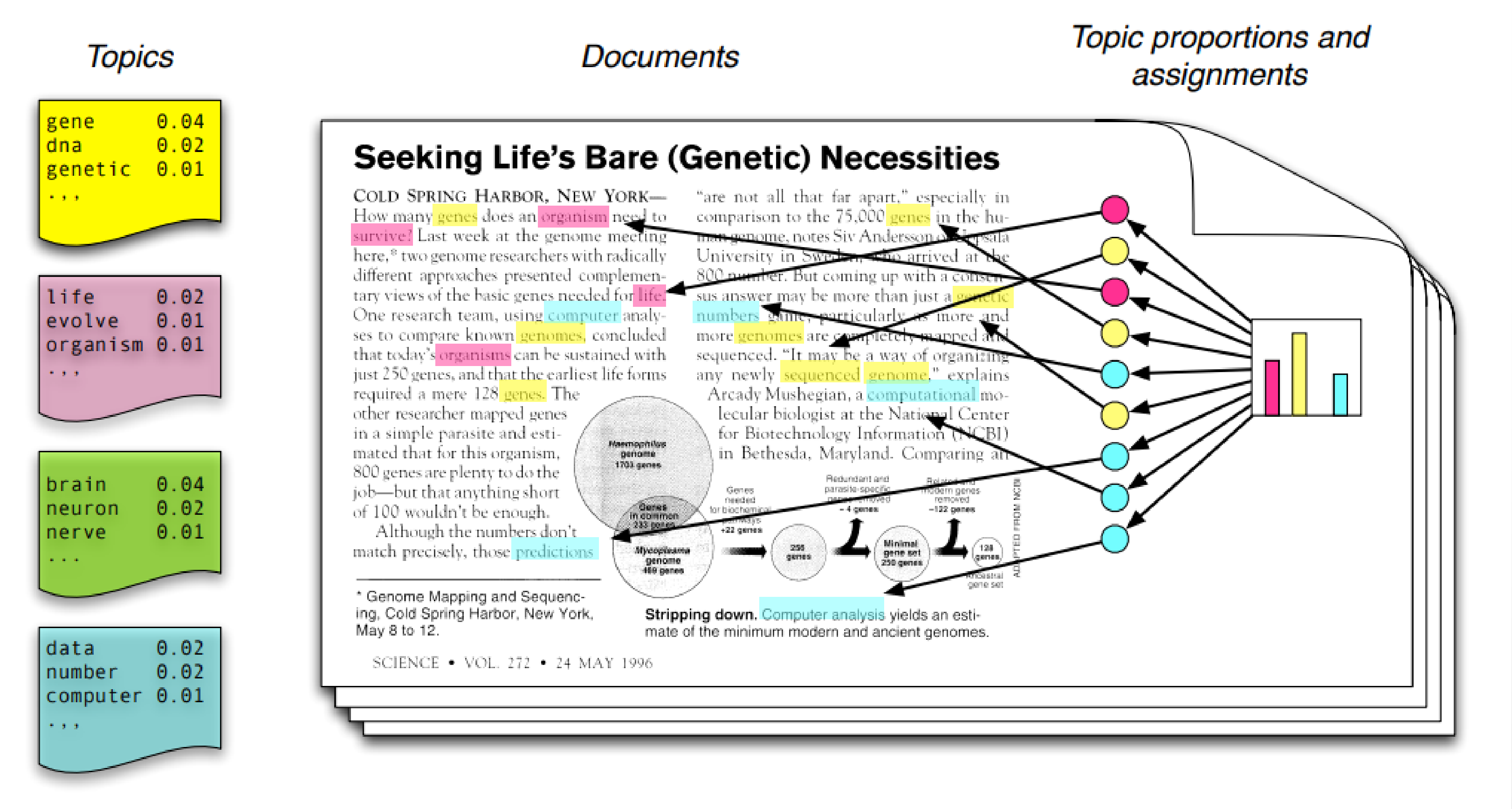
Word	Class = บรรยากาศ	Class = อาหาร
ชอบ	0.3	0.05
อร่อย	0.3	0.05
ไม่	0.05	0.6
กลมกล่อม	0.25	0.1
ทานง่าย	0.05	0.1
แต่	0.05	0.1

$P(w|z)$

- η and α is called Dirichlet distribution (Distribution of distribution)
- θ and β is called Multinomial distribution

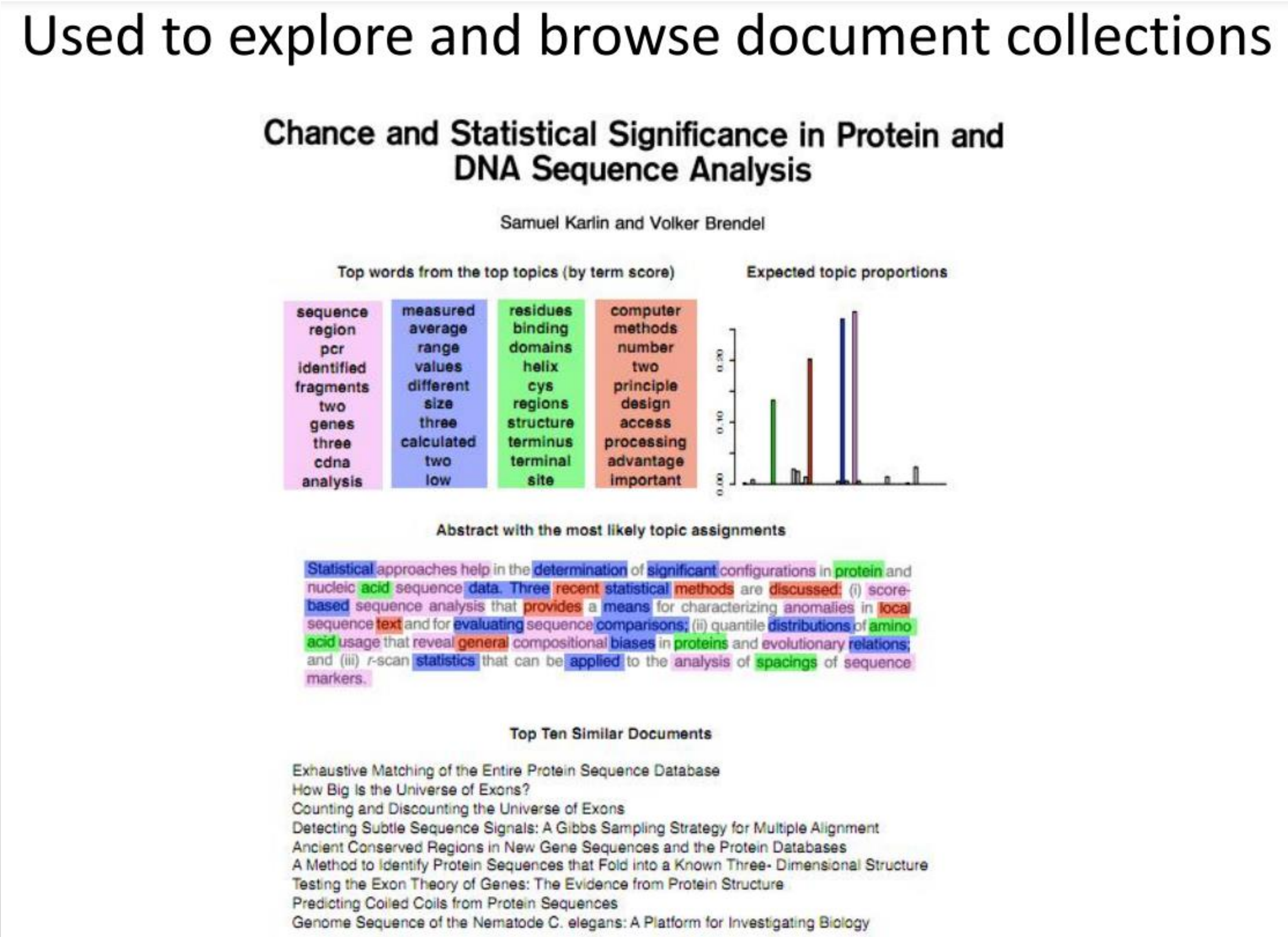
LDA application

- Automatically learns topics
- Give the word distribution of each topic
- Easy for interpretability
- Requires number of topic
- Requires user to make sense of the learned topics



Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.
https://www.eecis.udel.edu/~shatkay/Course/papers/UIntrotoTopicModelsBlei2011-5.pdf

LDA application (cont.)



Used to explore and browse document collections

Chance and Statistical Significance in Protein and DNA Sequence Analysis

Samuel Karlin and Volker Brendel

Top words from the top topics (by term score)

sequence	measured	residues	computer
region	average	binding	methods
pcr	range	domains	number
identified	values	helix	two
fragments	different	cys	principle
two	size	regions	design
genes	three	structure	access
three	calculated	terminus	processing
cdna	two	terminal	advantage
analysis	low	site	important

Expected topic proportions

Abstract with the most likely topic assignments

Statistical approaches help in the determination of significant configurations in protein and nucleic acid sequence data. Three recent statistical methods are discussed: (i) score-based sequence analysis that provides a means for characterizing anomalies in local sequence text and for evaluating sequence comparisons; (ii) quantile distributions of amino acid usage that reveal general compositional biases in proteins and evolutionary relations; and (iii) r-scan statistics that can be applied to the analysis of spacings of sequence markers.

Top Ten Similar Documents

Exhaustive Matching of the Entire Protein Sequence Database
How Big Is the Universe of Exons?
Counting and Discounting the Universe of Exons
Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment
Ancient Conserved Regions in New Gene Sequences and the Protein Databases
A Method to Identify Protein Sequences that Fold into a Known Three- Dimensional Structure
Testing the Exon Theory of Genes: The Evidence from Protein Structure
Predicting Coiled Coils from Protein Sequences
Genome Sequence of the Nematode C. elegans: A Platform for Investigating Biology

http://jialu.info/technical_notes/Topic_Model.pdf

LDA application (cont.)

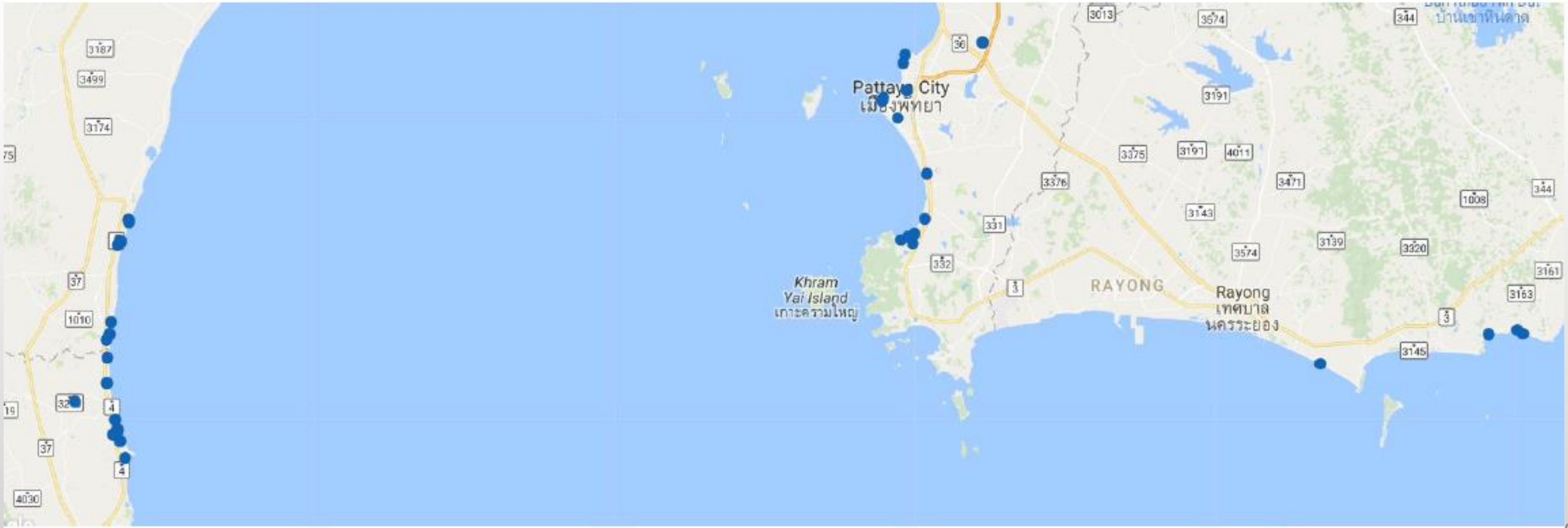
- Project: Chula x HOME dot TECH

คอนโดหรูสไตล์อังกฤษ แห่งแรกในเขาใหญ่ ที่ติด ถ.ธนะรัชต์ มากที่สุด 1ห้องนอน 1 ห้องน้ำ 1 ห้องนั่งเล่นพร้อมห้องครัวแยกเป็นสัดส่วนคอนโดหรูสไตล์อังกฤษ แห่งแรกในเขาใหญ่ ที่ติด ถ.ธนะรัชต์ มากที่สุด 1 ห้องนอน 1 ห้องน้ำ 1 ห้องนั่งเล่น พร้อมห้องครัวแยกเป็นสัดส่วน

LDA application (cont.)

- Project: Chula x HOME dot TECH

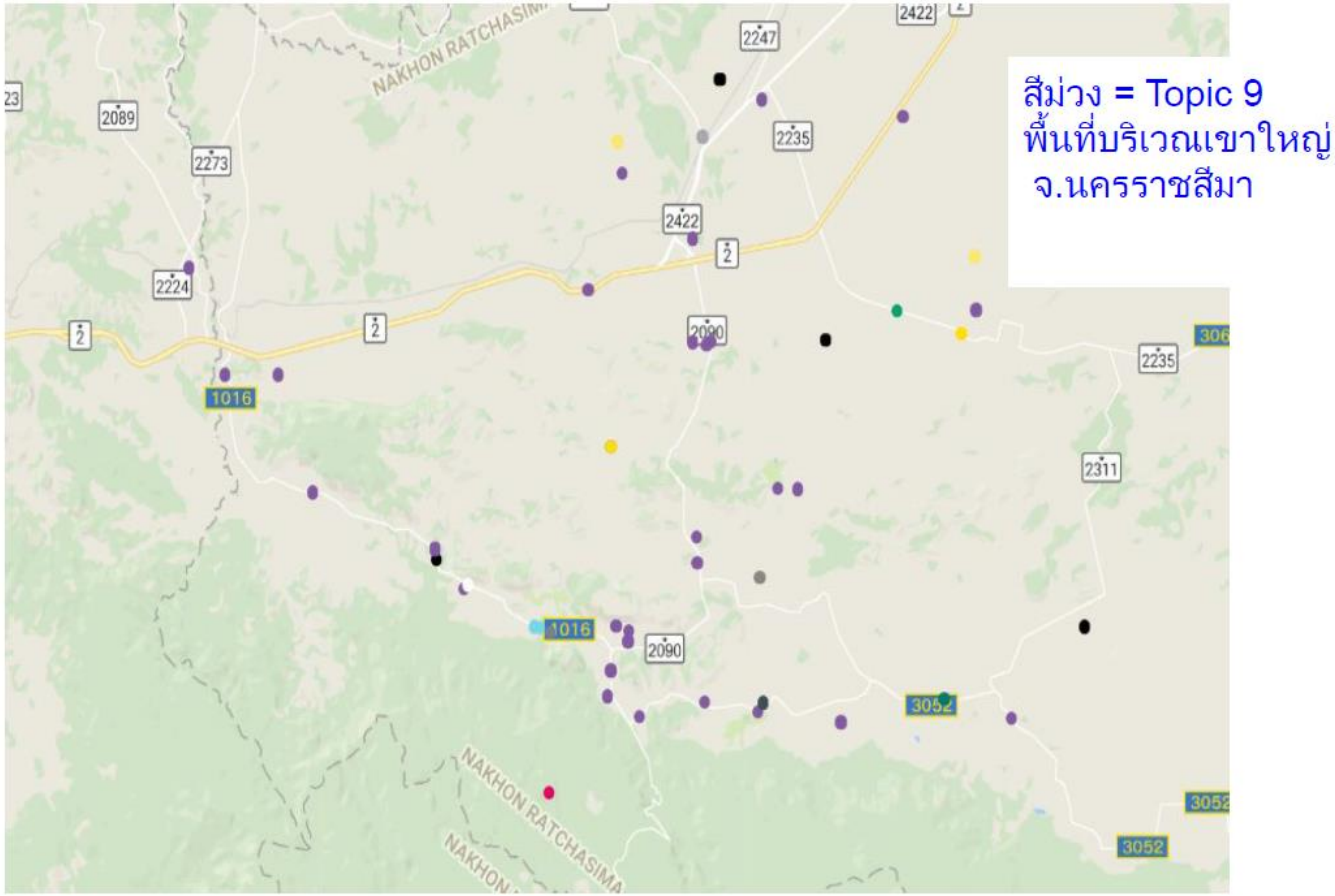
Topic 28
0.068*"วิว" + 0.058*"ทะเล" + 0.038*"คอนโด" + 0.029*"หัว" + 0.027*"คอนโดมิเนียม" + 0.025*"มองเห็น" + 0.023*"ทัศนียภาพ" + 0.022*"ชายหาด"



LDA application (cont.)

- Project: Chula x HOME dot TECH

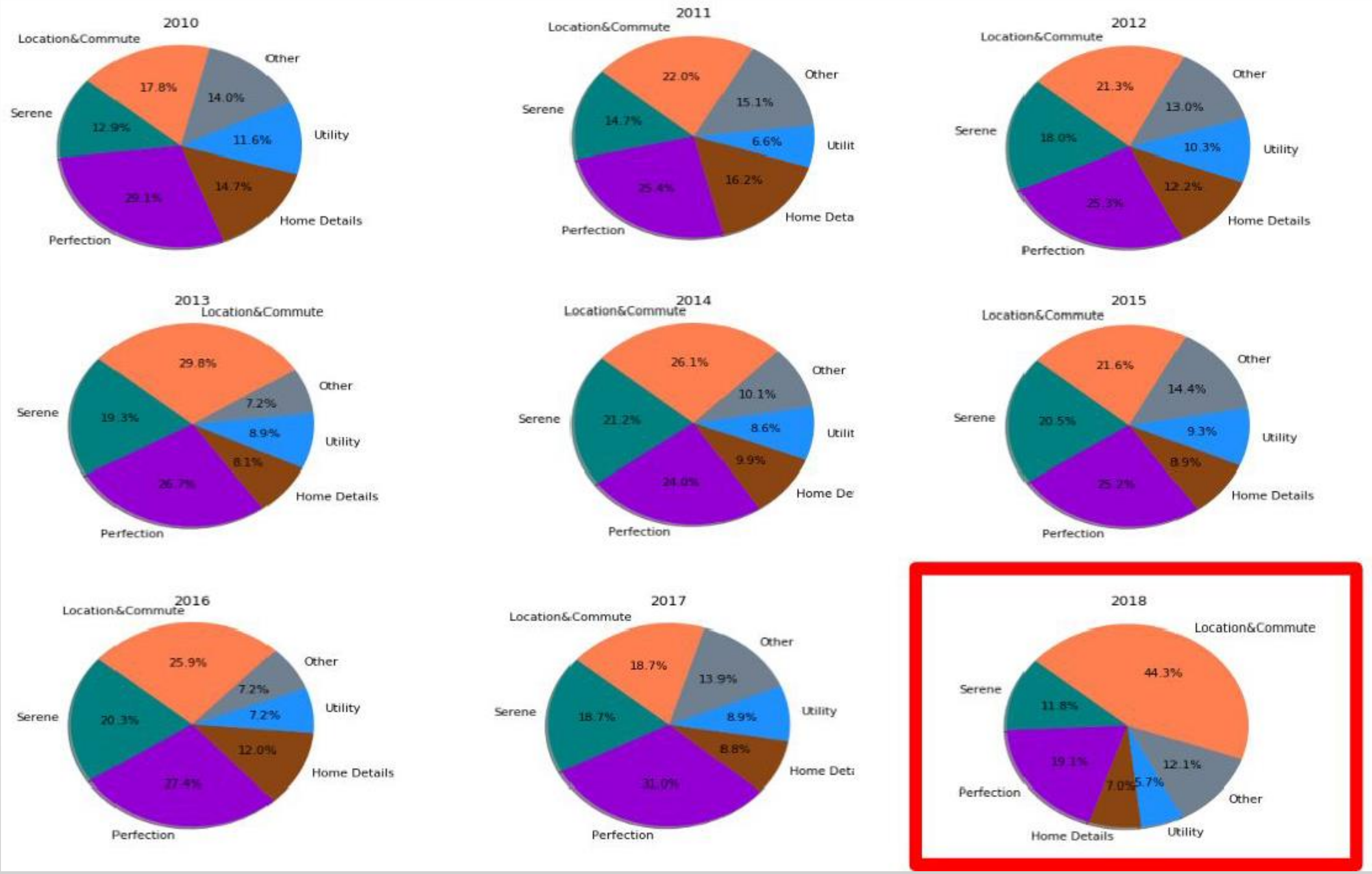
```
Topic 9
0.071*"ธรรมชาติ" + 0.031*"บรรยากาศ" + 0.028*"ร่มรื่น" + 0.027*"บ้าน" + 0.025*"ท่ามกลาง" + 0.025*"สวน" + 0.025*"สัมผัส" + 0.
021*"พื้นที่"
```



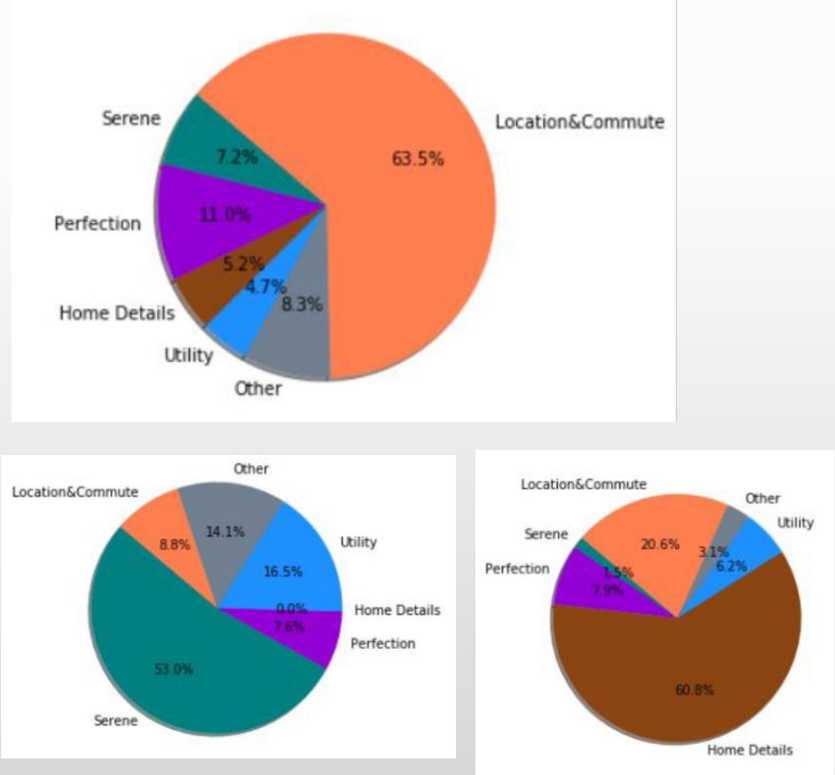
LDA application (cont.)

- Project: Chula x HOME dot TECH

Trend analysis

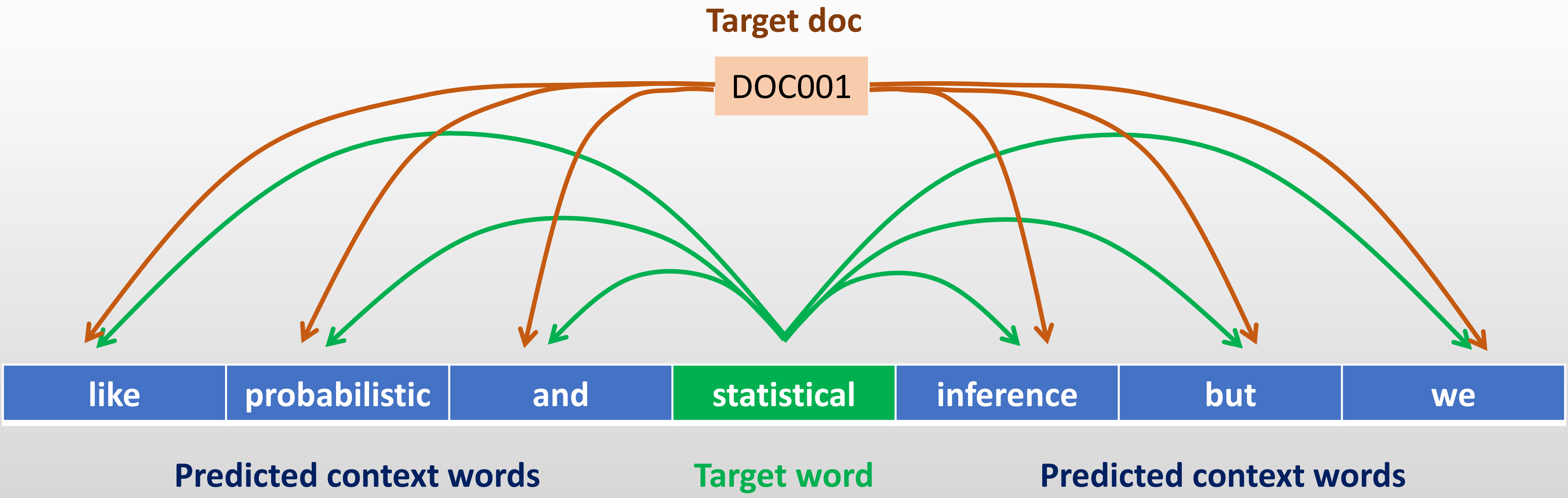


Niche of each project



LDA with deep learning

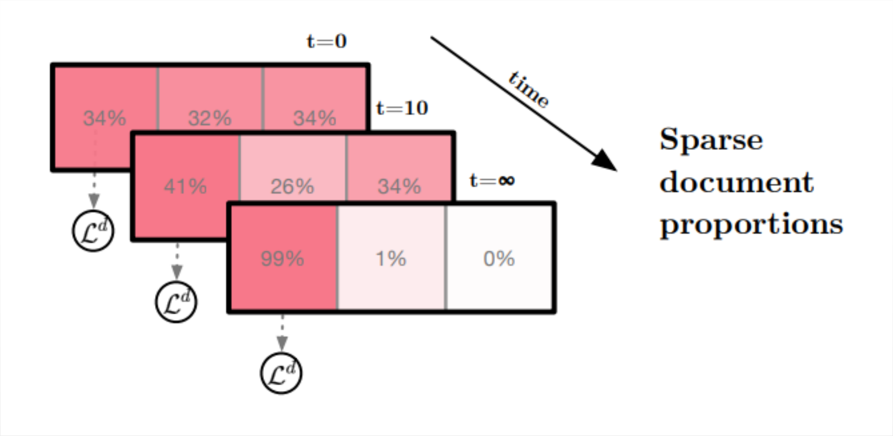
- Modified network structure and loss function to include LDA traits
  - Use like a neural network (just like how we use crf in neural networks)
  - LDA2vec (global information)



LDA with deep learning (cont.)

Doc2Vec

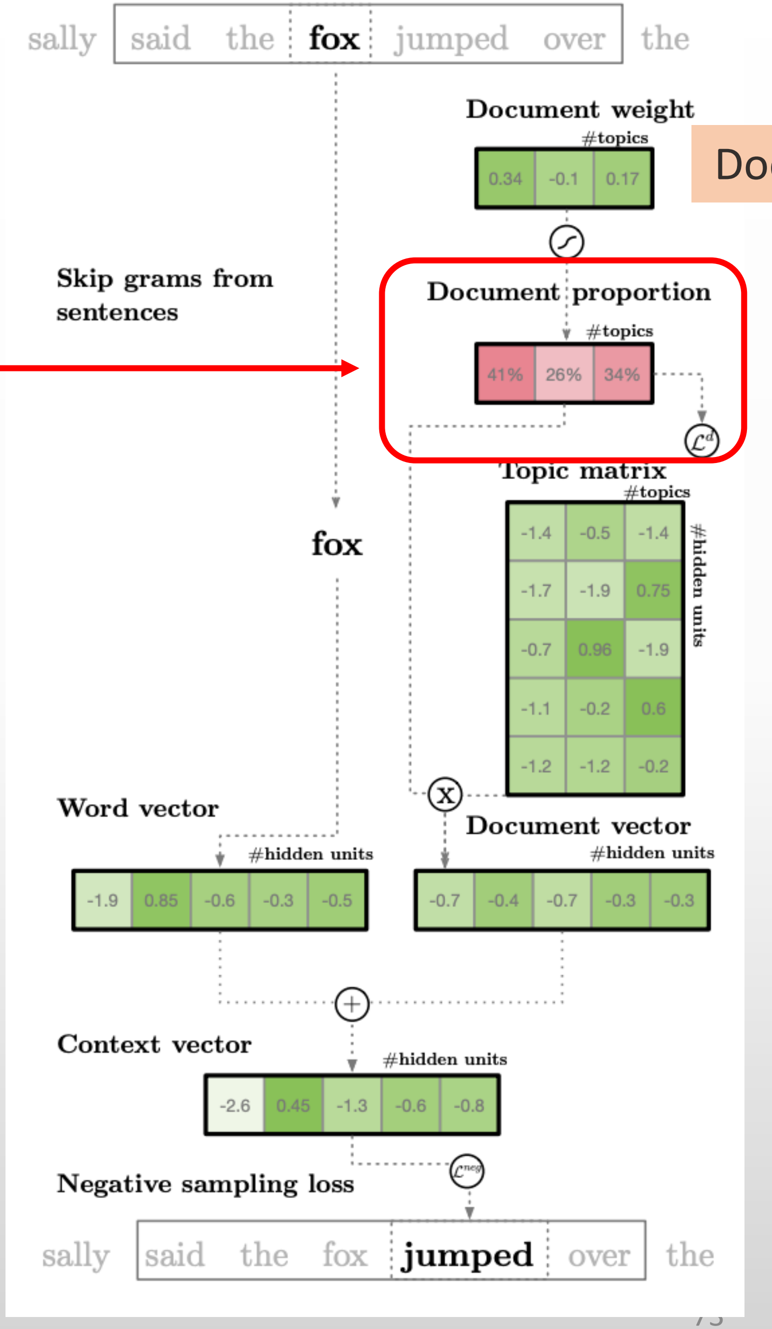
Add a loss term to include the Dirichlet loss which prefers sparse topics



$$\mathcal{L}^d = \lambda \Sigma_{jk} \left(\alpha - 1\right) \log p_{jk}$$

$$\alpha = n^{-1}$$

n is number of topics



Moody, C. E. (2016). Mixing dirichlet topic models and word embeddings to make lda2vec. *arXiv preprint arXiv:1605.02019*.

https://arxiv.org/pdf/1605.02019v1.pdf

Demo: Naïve Bayes for text Classification

https://drive.google.com/file/d/1fBBM-ILOf5_IwxD4pLIyT-616GTP_d6b/view?usp=share_link