Supervised Retrieval Models & Some Tips

Kuan-Yu Chen (陳冠宇)

2020/12/25 @ TR-313, NTUST

The Research Topic of Final Project

1/8

Presentation-1

M10915100 郭O威、D10907005 陳O宏、M10915034 黃O泓、M10915066 盧O恆
Using Intelligent method for Fraud SMS/Email hunting and detection

M10915103 邱O儒、M10915006 廖O宏、M10915046 陳O穎、M10915092 林O哲

M10915095 薛O翔、M10915050 林O瑜 、M10909112 石O安、M10909120 樊O驊
Term Weighting for Feature Extraction on Twitter: A Comparison Between BM25 and TF-IDF

M10909211 李O妍、M10909118 蔡O真 、M10909109 陳O炫、M10909114 李O凱

80847002S 羅O宏、80847001S 顏O成、60947058S 曹O升、60947012S 王O偉

M10815111 謝O耀、M10815112 鄭O哲、M10915017 林O箴

M10915012 黃O愷、M10915036 王O崴、M10915082 張O哲
將資訊檢索技術應用於聊天機器人開發

M10915045 施O宏、M10915031 鄭O謙、M10915080 羅O程

M10915019 顏O庭、M10915013 王O翔、M10815103 陳O揚

M10915201 陳O凡、M10915097 朱O亞、M10815064 侯O林

M10915028 陳O勳、M10815036 王O德、M10815048 張O銘

1/15

Presentation-2

B10615013 李O鎧、B10615024 李O宗、B10615026 溫O勳、B10615043 何O峻

B10615022 姜O昀、B10615034 黃O翰、B10615036 黃O銘、B10615056 黃O翔

B10615047 陳O緯、B10615017 林O叡、B10615023 楊O安、B10615039 高O雲

B10632026 吳O瑄、M10907505 游O臨、M10915010 盧O函

B10615046 柯O豪、B10615045 陳O富、B10601002 廖O捷

M10802131 李O宇、M10802130 陳O瑋

M10915060 林O崴、B10430302 許O森、M10815090 曾O筑、M10915002 許O樂
An Extended Vector Space Model for XML Information Retrieval

M10815013 陳O妮、M10815074 張O綸

M10915027 石O峰、B10615033 王O禎

B10630024 劉O奇、B10630040 吳O宏

HW6 - BERT

#	Team Name	Notebook	Team Members	Score	Entries	Last
1	BERT			0.48011	3	4d
2	as			0.46754	23	14h
	Rerank with BM25 and BERT			0.45084		
3	乖乖切驗證集找alpha不香ㄇ			0.45084	1	9d
4	Wecan Huang			0.44878	25	16h
5	Ke-Han Lu			0.44080	6	15h
6	pon pon shit			0.43234	5	9h
7	B10615034_黃柏翰			0.42322	17	7h
8	80847001s_顏必成			0.39530	5	17h
	BM25			0.39136		
9	Winnie the Pooh			0.39136	5	10h
10	怎麼過baseline			0.39135	2	17h
11	i want graduated			0.38994	5	20h
12	終於最後惹			0.38925	3	2d
13	TESTING			0.38093	6	9h
	Rerank with BERT only			0.30248		
	Random Documents			0.00007		

NN-based Language Models



Language Representations (2013~)

Neural Network Language Models (2001~)

Continuous Language Models

Continuous Language Models (2007~2009)

Topic Models (1997~2003)


Topic Models

Query Language Models (2001~2006)


Word-Regularity Models

Discriminative Language Models (2000~2011)

Word-Regularity Models (~1997)




2000 2002 2004 2006 2008 2010 2012 2014 2016

Significant Innovations in LMs.

- A goal of statistical language modeling is to learn the joint probability function of sequences of words in a language

$$P(w_1, w_2, \dots, w_T)$$

- A statistical model of language can be represented by the conditional probability of the next word given all the previous ones (**chain rule**)

$$P(w_1, w_2, \dots, w_T) = \prod_{t=1}^{T} P(w_t | w_1, w_2, \dots, w_{t-1})$$

$$\approx \prod_{t=1}^{T} P(w_t | w_{t-n+1}, \dots, w_{t-1})$$

  - Such statistical language models have already been found useful in many technological applications involving natural language

Significant Innovations in LMs..

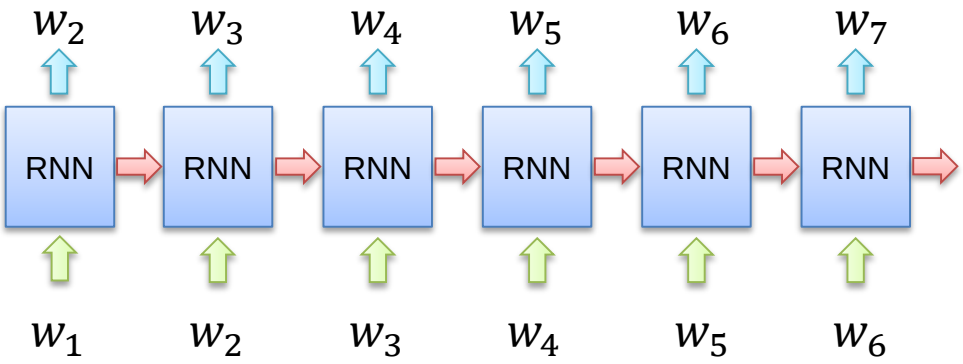
- Long-span information can be integrated by using topic models
  - The historical words $w_1, w_2, \ldots w_{t-1}$ can be treated as a document H_1^{t-1}!
  - Follow the bag-of-words assumption

$$P(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} P(w_t | w_1, w_2, \ldots, w_{t-1})$$
$$= \prod_{t=1}^{T} P\left(w_t | H_1^{t-1}\right)$$
$$= \prod_{t=1}^{T} \left(\sum_{k=1}^{K} P(w_i | T_k) P\left(T_k | H_1^{t-1}\right) \right)$$

T. Mikolov, *et al.*, "Recurrent Neural Network based Language Model," in *Proc. of INTERSPEECH*, 2010.

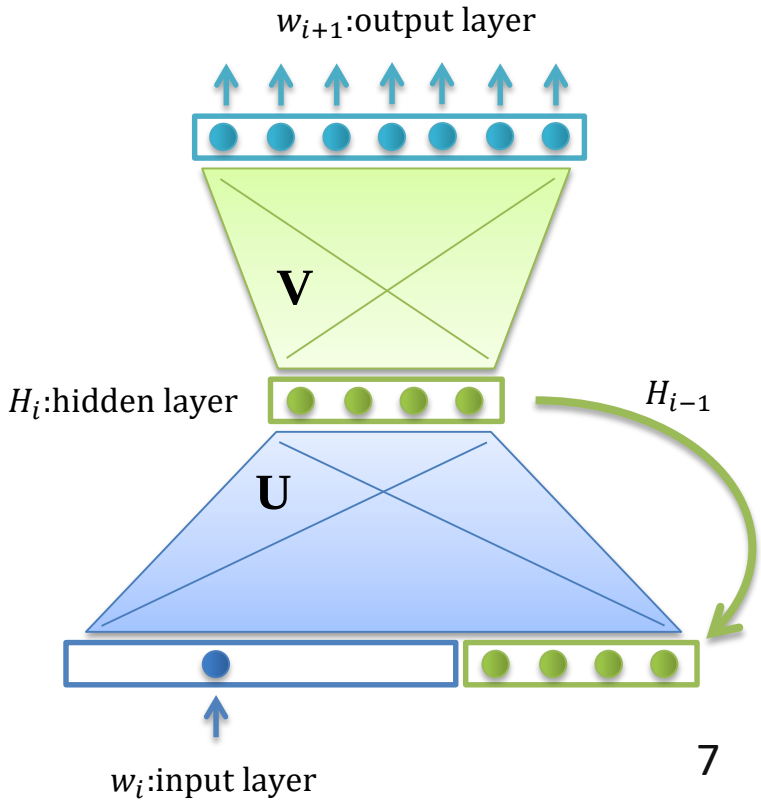
Significant Innovations in LMs…

- When the story comes to the field of deep learning, the recurrent network is the best choice to model the long-span information
  - $RNN(w_t)$ interpretes $RNN(w_t|w_{t-1}, H_1^{t-2})$



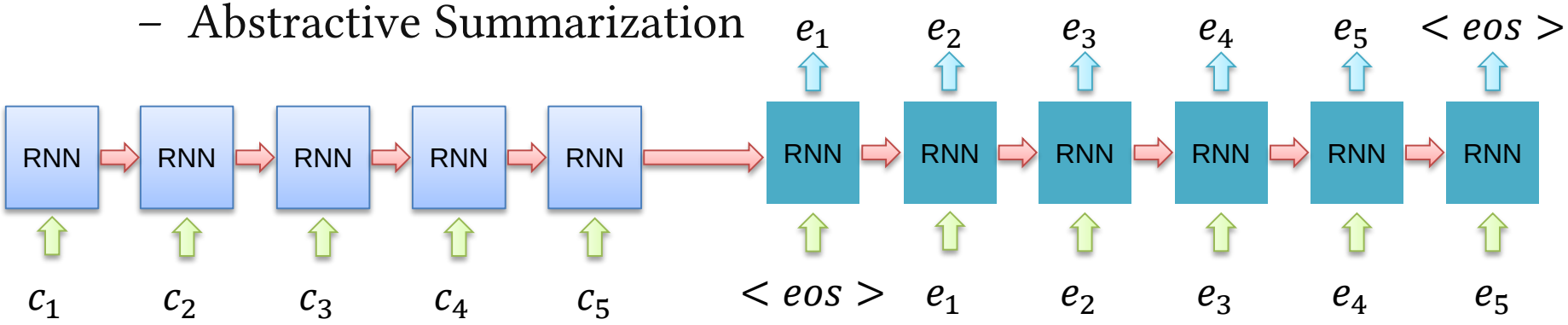
$$P(w_1, w_2, \dots, w_T) = \prod_{t=1}^{T} P(w_t|w_1, w_2, \dots, w_{t-1})$$

$$= \prod_{t=1}^{T} RNN(w_t)$$



Significant Innovations in LMs…

- Based on the recurrent networks, sequence-to-sequence learning becomes a popular research subject
  - Machine Translation
  - Speech Recognition
  - Abstractive Summarization



- RNN is good, but it is not power enough to model the long-span information

$$P(e_1, e_2, \dots, e_T) = \prod_{t=1}^{T} P\left(e_t | \{e_1, e_2, \dots, e_{t-1}\}, \{c_1, c_2, \dots, c_{T'}\}\right)$$

Significant Innovations in LMs…

- Attention mechanism is proposed to the machine translation task
  - The word "attention" is only three times in the paper
  - The objective of the translation model is to estimate a conditional probability of the upcoming word

$$P(e_1, e_2, \ldots, e_T) = \prod_{t=1}^{T} P\left(e_t | \{e_1, e_2, \ldots, e_{t-1}\}, \{c_1, c_2, \ldots, c_{T'}\}\right)$$

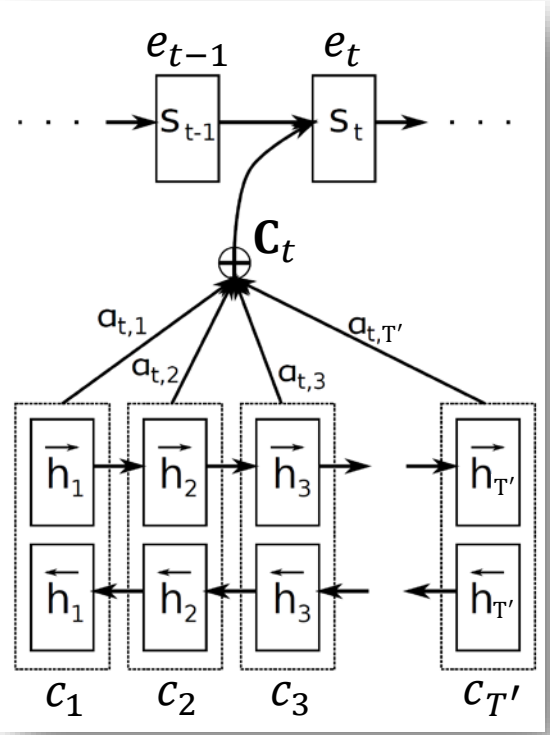
$$= \prod_{t=1}^{T} P(e_t | s_t, e_{t-1}, \mathbf{C}_t)$$

$$s_t = f(s_{t-1}, e_{t-1}, \mathbf{C}_t)$$

$$\mathbf{C}_t = \sum_{i=1}^{T'} \alpha_{ti}\, h_i = \sum_{i=1}^{T'} \frac{\exp(e_{ti})}{\sum_{j=1}^{T'} \exp(e_{tj})} h_i$$

$\delta(\cdot,\cdot)$ is the key!

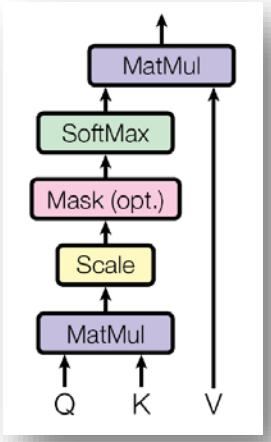
$$= \sum_{i=1}^{T'} \frac{\exp(\delta(s_{t-1}, h_i))}{\sum_{j=1}^{T'} \exp(\delta(s_{t-1}, h_j))} h_i$$

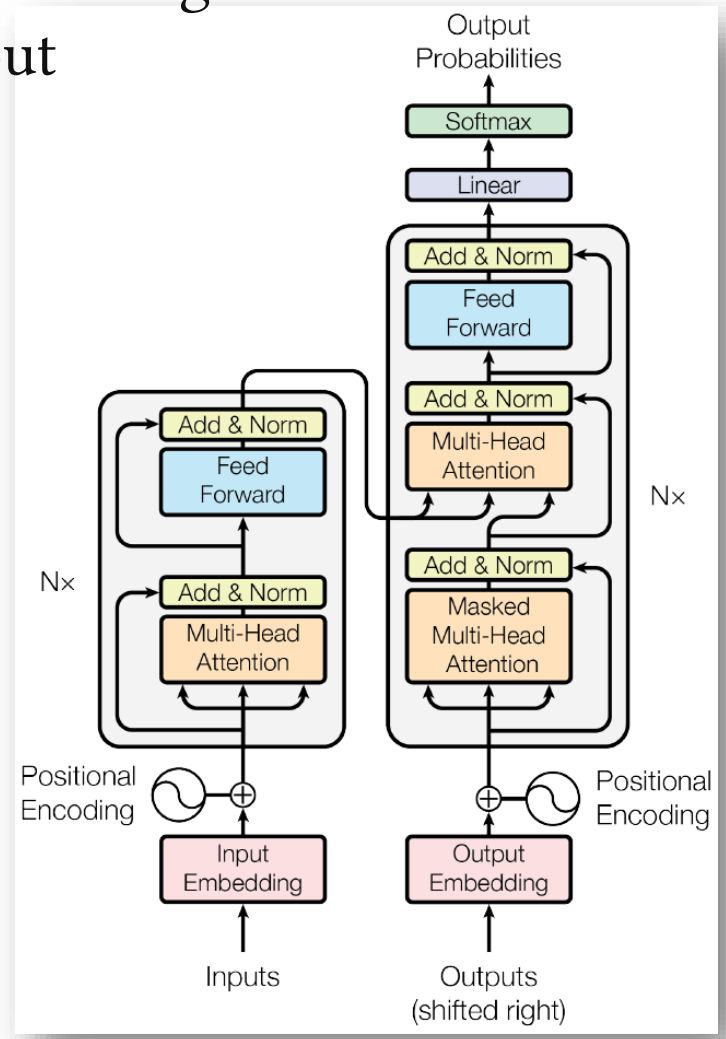


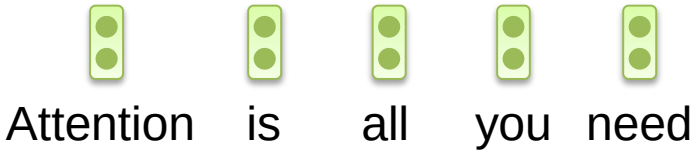
A. Vaswani, *et al*., "Attention is All You Need," in *Proc. of NIPS*, 2017.

Significant Innovations in LMs….

- The transformer eschews recurrence and instead relies entirely on an attention mechanism to draw global dependencies between input and output
  - RNN is very slow
  - Word pair information is enough!
    - Self-attention is a key

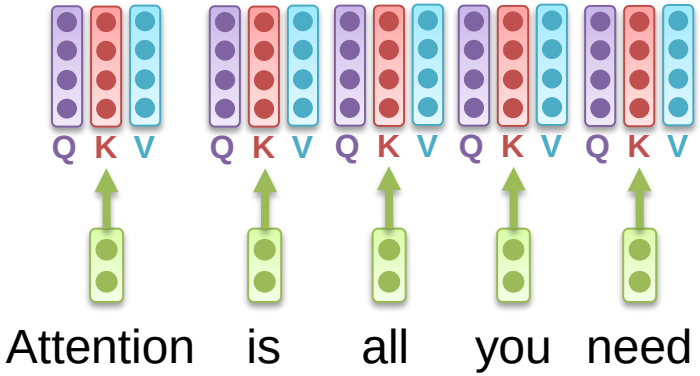


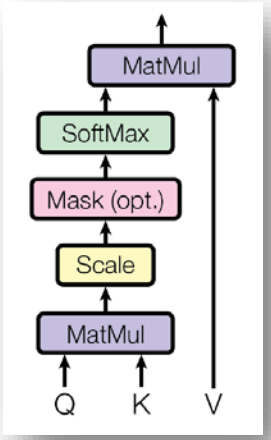


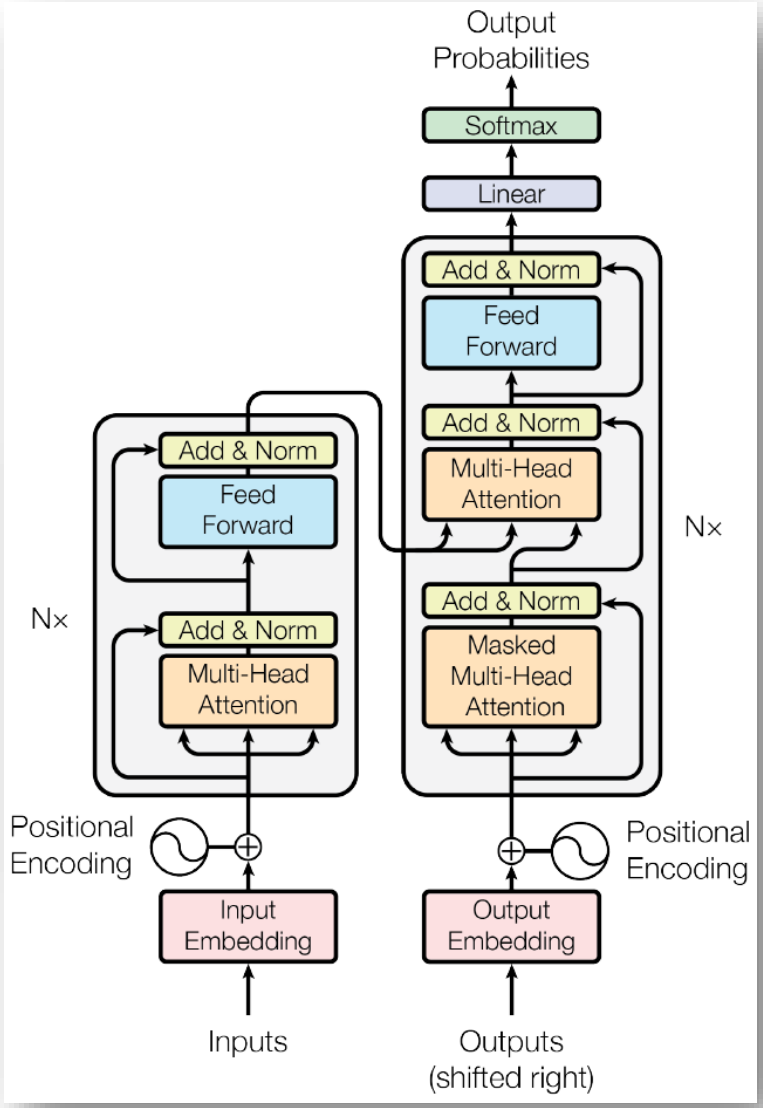


Significant Innovations in LMs….

- Word pair information is enough!
  - The “transformer” unit is proposed
    - Attention is all you need
    - Self-attention is the key

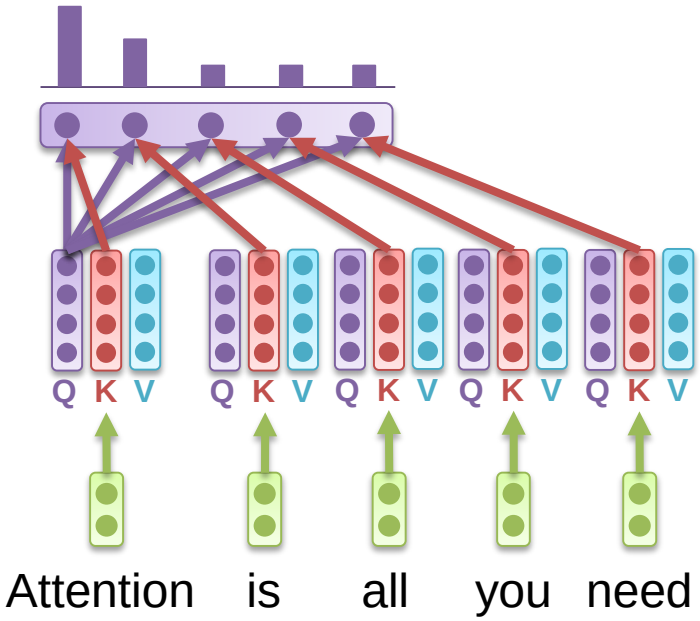


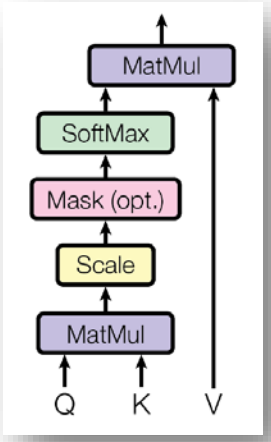


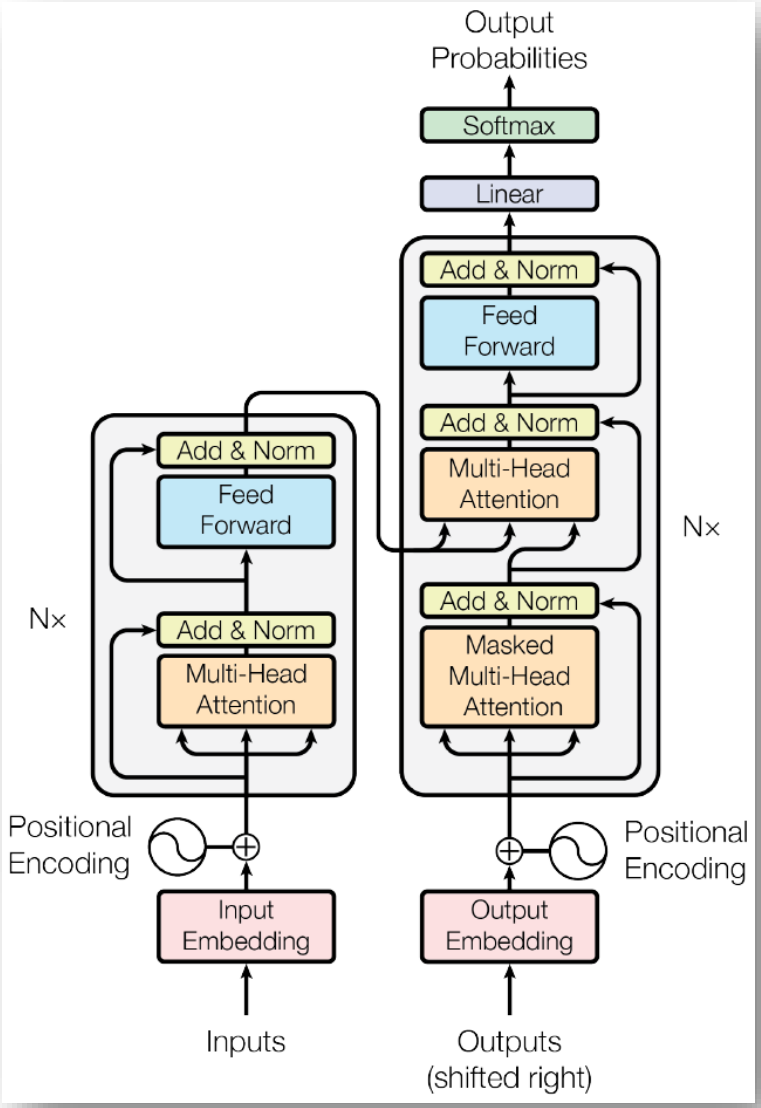


Significant Innovations in LMs….

- Word pair information is enough!
  - The “transformer” unit is proposed
    - Attention is all you need
    - Self-attention is the key

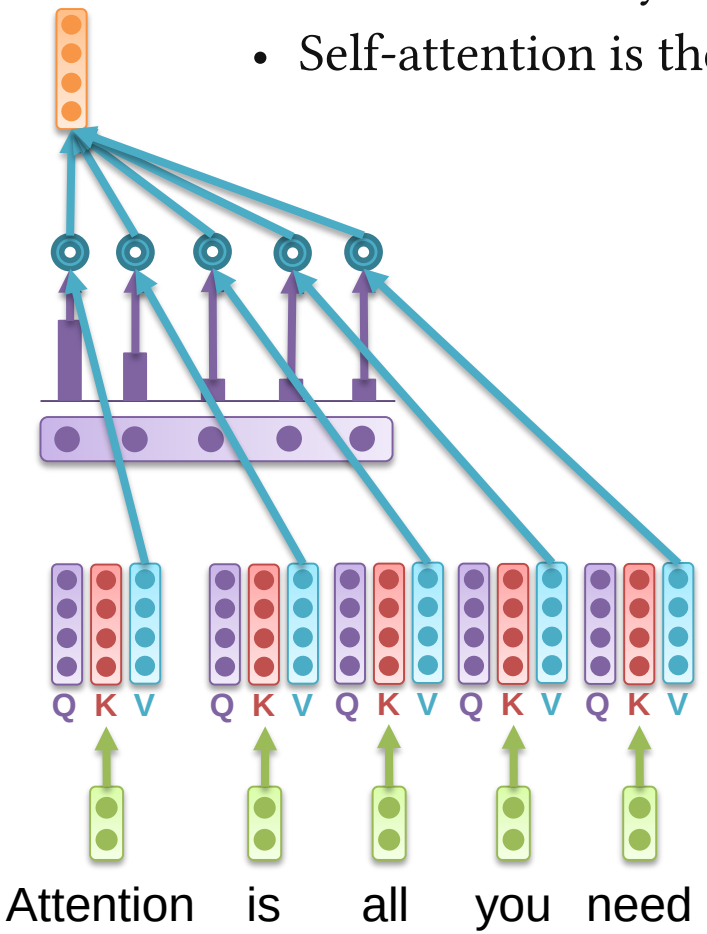


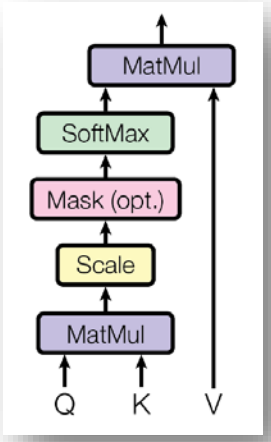


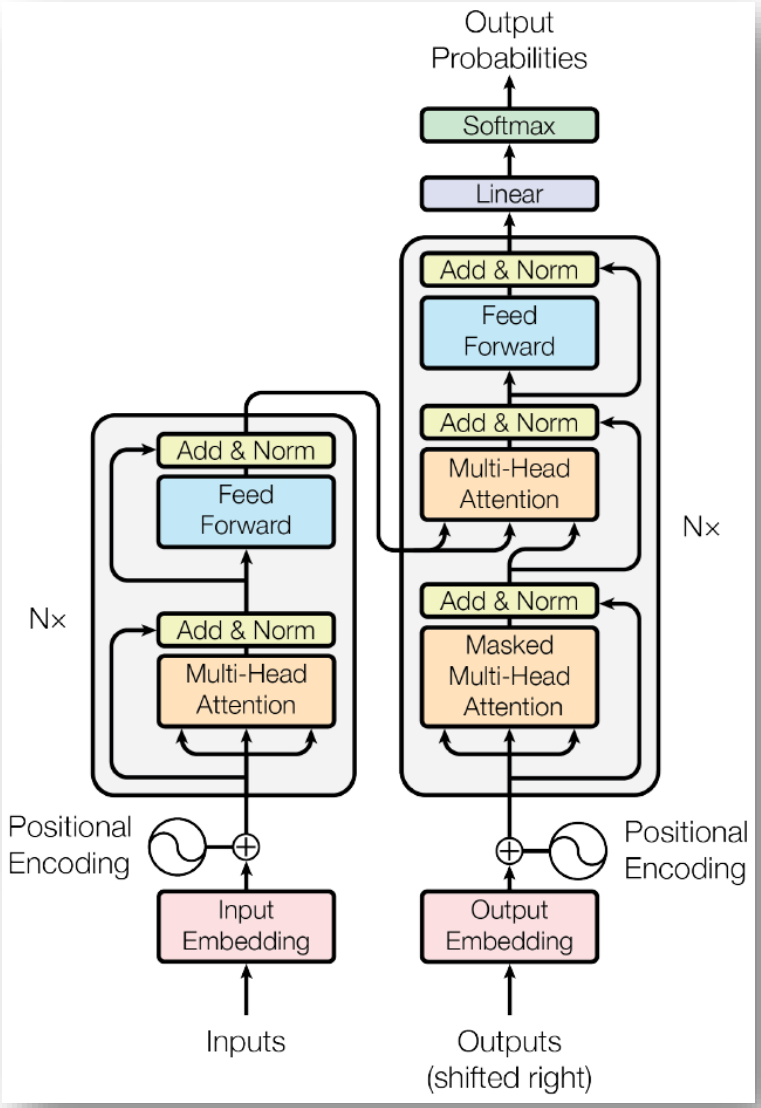


Significant Innovations in LMs….

- Word pair information is enough!
  - The “transformer” unit is proposed
    - Attention is all you need
    - Self-attention is the key





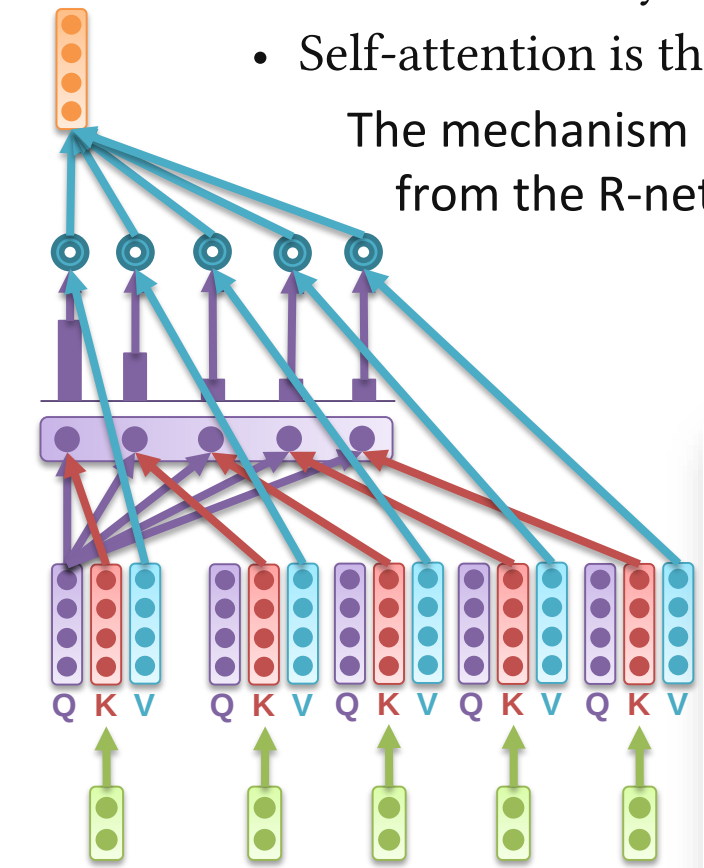


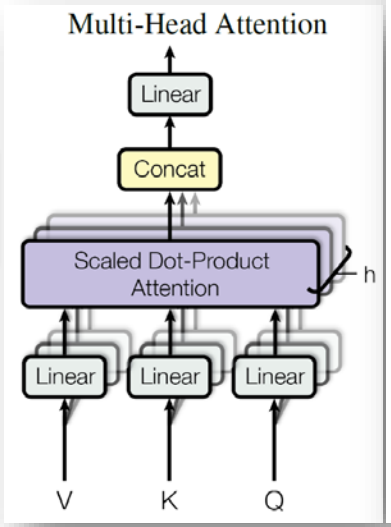
Microsoft Research Asia, "R-NET: MACHINE READING COMPREHENSION WITH SELF-MATCHING NETWORKS," in *Proc. of ACL*, 2017.

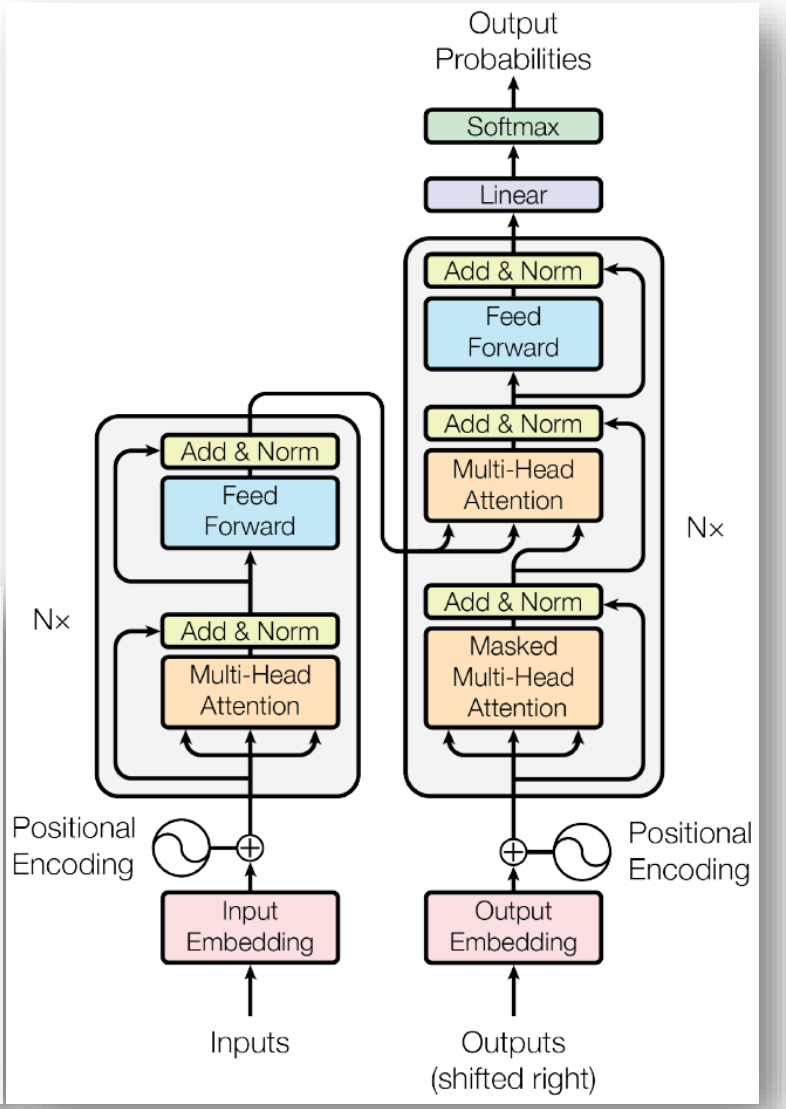
Significant Innovations in LMs….

- Word pair information is enough!
  - The "transformer" unit is proposed
    - Attention is all you need
    - Self-attention is the key

The mechanism may inspire from the R-net

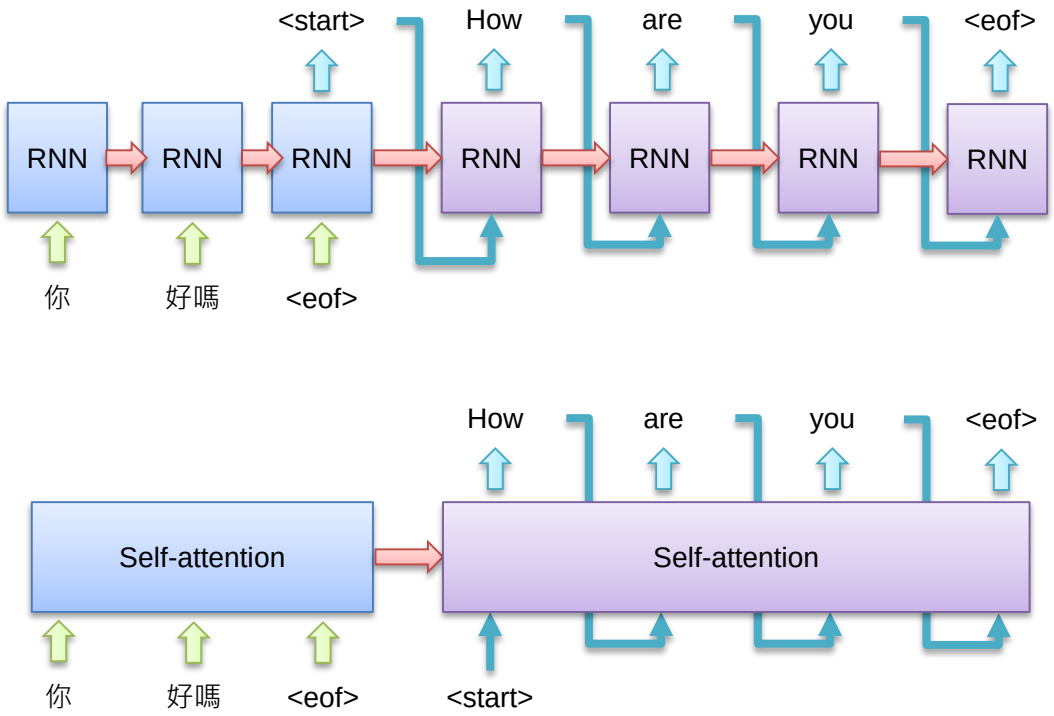


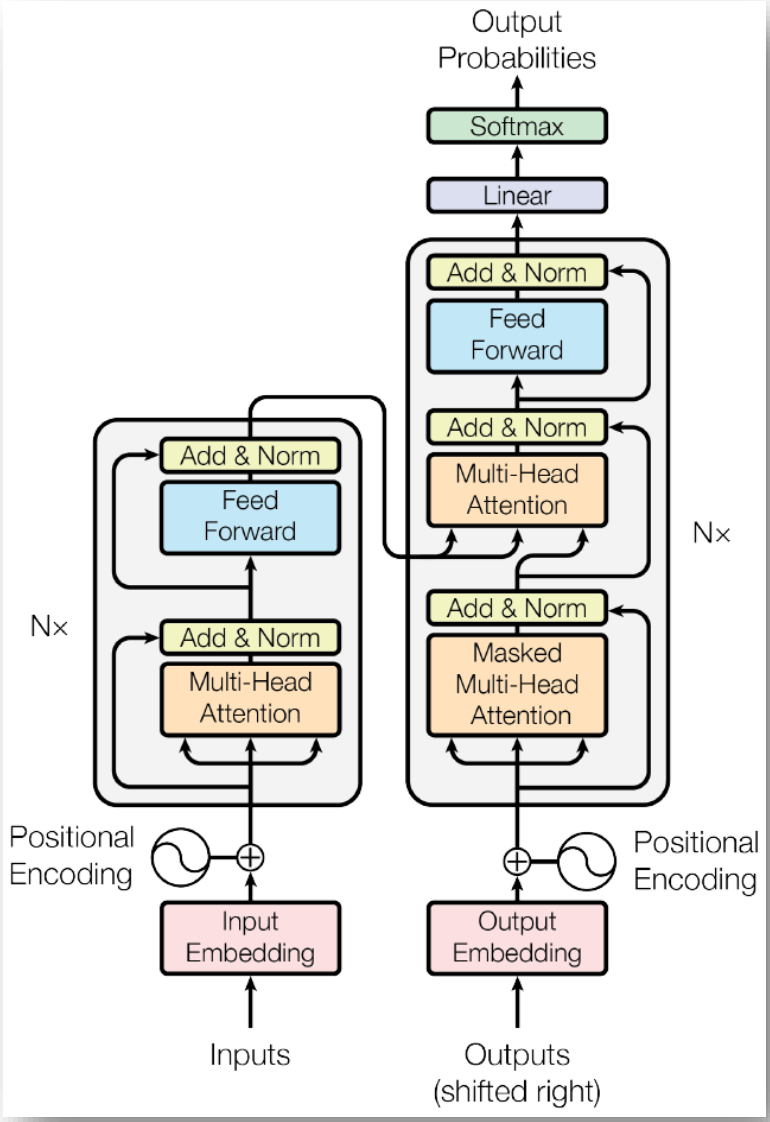




Significant Innovations in LMs….

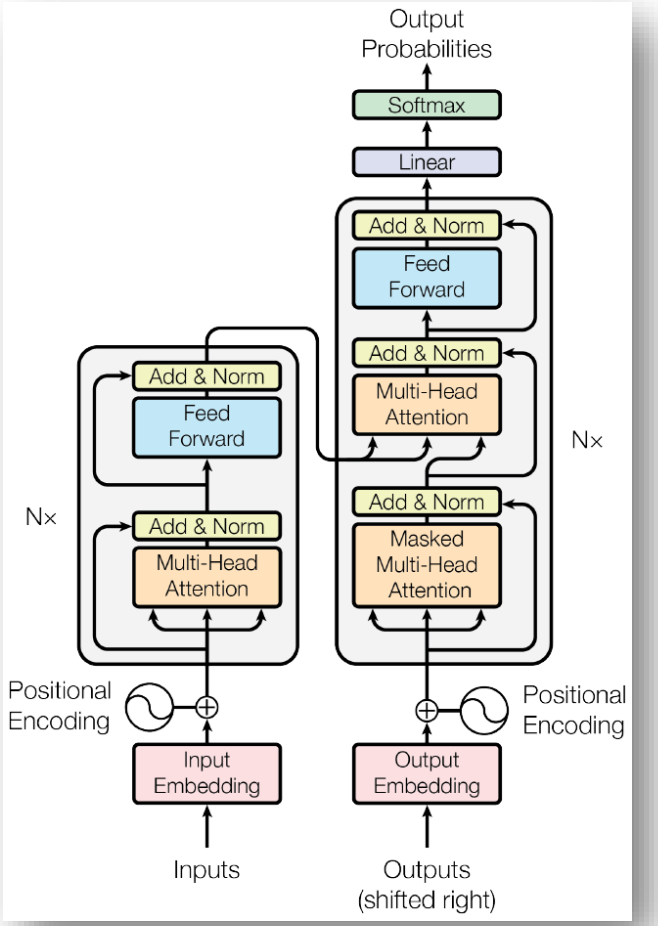
- Word pair information is enough!
  - The “transformer” unit is proposed
    - Attention is all you need





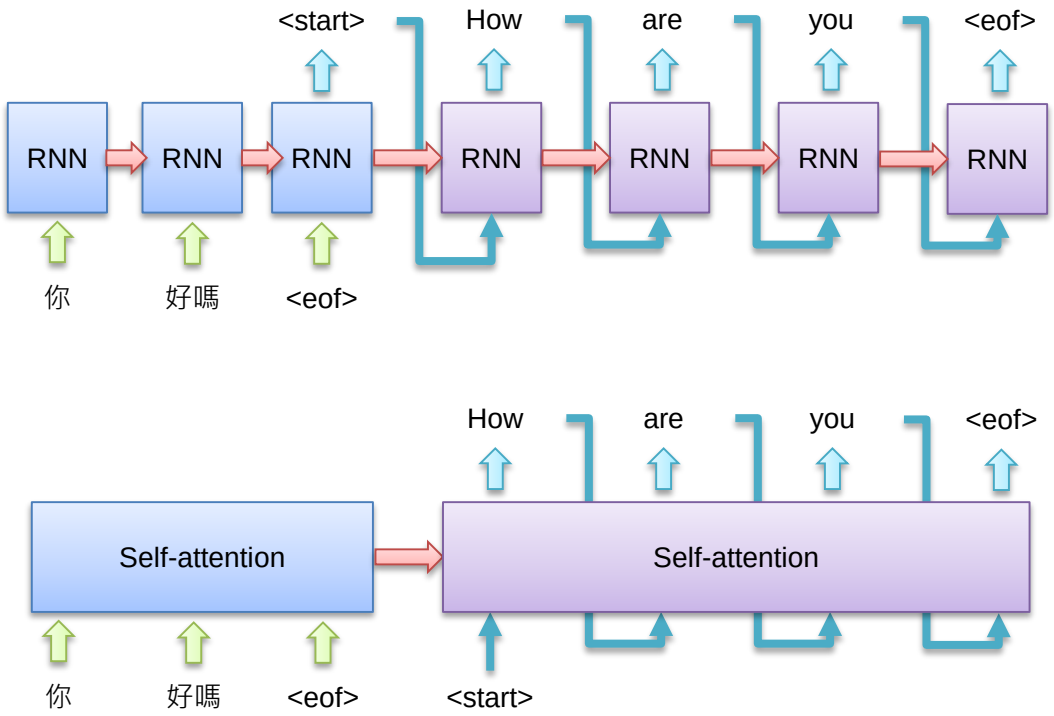
https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

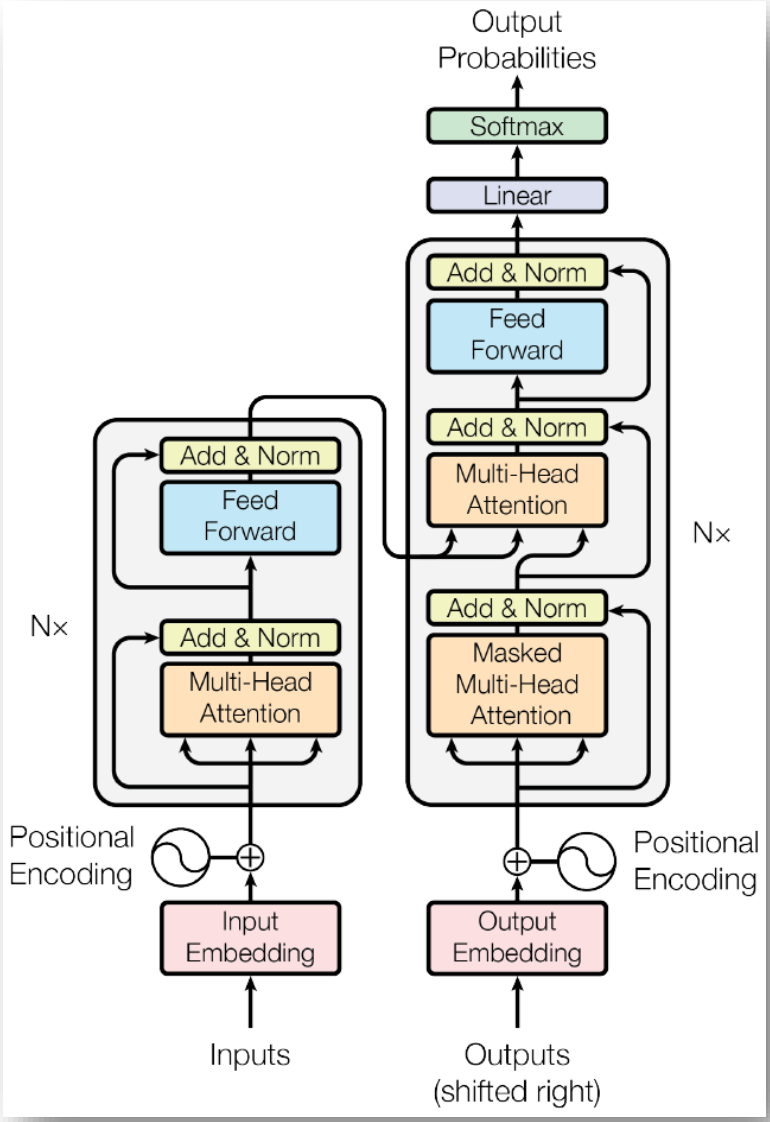
Significant Innovations in LMs….



Significant Innovations in LMs….

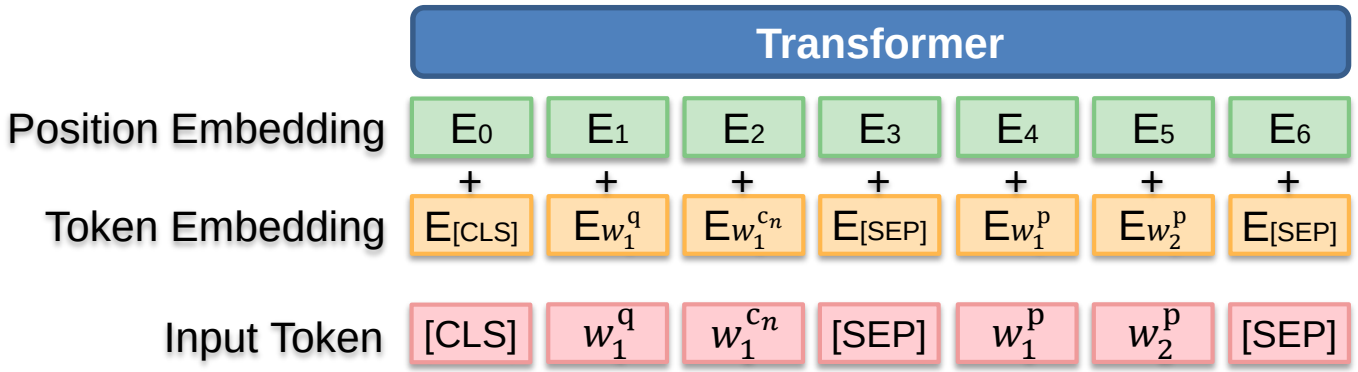
- Word pair information is enough!
  - The “transformer” unit is proposed
    - Ordering is encoded by a **positional embedding**



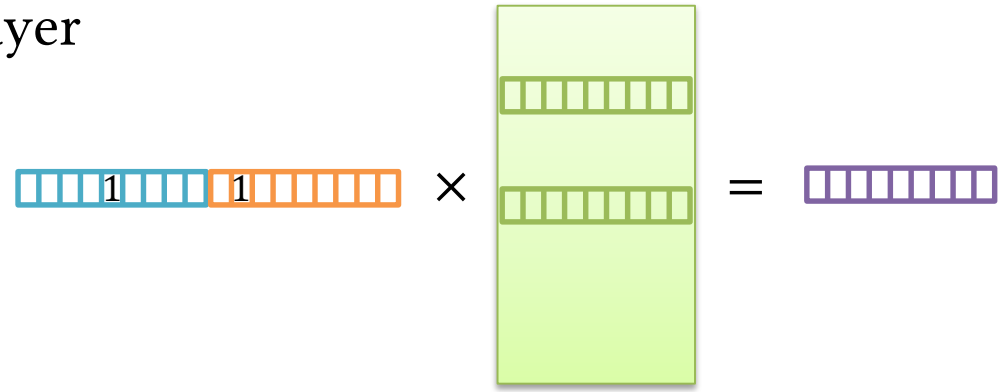


Significant Innovations in LMs….

- Word pair information is enough!
  - The "transformer" unit is proposed
    - Ordering is encoded by a **positional embedding**



  - Similar to concatenate two one-hot vector and go through a dense layer



R. Sennrich *et al*., “Neural Machine Translation of Rare Words with Subword Units,” in *Proc. of ACL*, 2016.

Wordpiece.

- Without stemming step, BERT uses wordpieces to re-represent given sentences
  - BPE: Byte-Pair Encoding

unit	frequency
lower	2
new	6
widest	3
low	5

unit	frequency
l	**7**
o	**7**
w	**16**
e	**11**
n	**6**
r	2
i	3
d	3
s	3
t	3

unit	frequency
l	**7**
o	**7**
w	**16**
e	**11**
n	**6**
r	2
i	3
d	3
s	3
t	3
lo	**7**
ow	**7**
we	2
wi	3
er	2
ew	**6**
es	3
ne	**6**

Wordpiece..

- If the size of lexicon is set to 9
  - l, o, w, e, n, lo, ow, ew, ne
- By using the BPE units, the words become:
  - lower = _lo w e
  - new = _ne w
  - widest = _w e
  - low = _lo w
- The advantage is the size of the lexicon can be pre-defined!

unit	frequency
l	**7**
o	**7**
w	**16**
e	**11**
n	**6**
r	2
i	3
d	3
s	3
t	3
lo	**7**
ow	**7**
we	2
wi	3
er	2
ew	**6**
es	3
ne	**6**

https://nlp.h-its.org/bpemb/en/

Wordpiece…

Vocab size	enwiki sample
original	the diets of the wealthy were rich in sugars, which promoted periodontal disease. despite the flattering physiques portrayed on tomb walls, the overwe the first project proposed and worked on was a period film to be called ''dora-heita'', but when this was deemed too expensive, attention shifted to ' neutralization with a base weaker than the acid results in a weakly acidic salt. an example is the weakly acidic ammonium chloride, which is produced
1000	_the _d iet s _of _the _we al th y _were _r ich _in _su g ars , _which _p rom ot ed _per i od on t al _dis e ase . _des p ite _the _fl at ter ing _ph ys i qu es _port ra y ed _on _to m b _w all s , _the _over we _the _first _pro ject _pro pos ed _and _work ed _on _was _a _per i od _film _to _be _called _'' d or a - he it a '', _but _when _this _was _de em ed _to o _exp ens ive , _att ent ion _sh if ted _to _' _ne ut ral iz ation _with _a _b ase _we ak er _than _the _ac id _res ult s _in _a _we ak ly _ac id ic _s al t . _an _ex amp le _is _the _we ak ly _ac id ic _am m on ium _ch l or ide , _which _is _prod uced
3000	_the _d iet s _of _the _we alth y _were _rich _in _sug ars , _which _promot ed _period on tal _dis e ase . _despite _the _fl at ter ing _phys i qu es _port ray ed _on _tom b _wall s , _the _over we _the _first _project _pro posed _and _worked _on _was _a _period _film _to _be _called _'' d or a - he ita '', _but _when _this _was _de em ed _too _exp ensive , _att ention _sh if ted _to _' _ne ut ral ization _with _a _base _we aker _than _the _ac id _result s _in _a _we ak ly _ac id ic _sal t . _an _example _is _the _we ak ly _ac id ic _am mon ium _ch lor ide , _which _is _produced
5000	_the _d iet s _of _the _we alth y _were _rich _in _sug ars , _which _promot ed _period on tal _disease . _despite _the _flat ter ing _phys iques _portray ed _on _tom b _wall s , _the _over we _the _first _project _proposed _and _worked _on _was _a _period _film _to _be _called _'' d ora - he ita '', _but _when _this _was _de em ed _too _exp ensive , _attention _sh if ted _to _' _ne ut ral ization _with _a _base _we aker _than _the _acid _results _in _a _we ak ly _acid ic _sal t . _an _example _is _the _we ak ly _acid ic _am mon ium _ch lor ide , _which _is _produced

https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html

Text Normalization

- Text pre-processing is a very important step, which includes
  - Lowercasing
  - Stemming
  - Lemmatization
  - Stopword Removal
  - Normalization
  - Noise Removal
  - …

https://tartarus.org/martin/PorterStemmer/

Stemming

- In linguistic morphology and information retrieval, stemming is the process of reducing inflected words to their word stem, base or root form
  - cats -> cat
  - stemmer -> stem
  - **Porter** and **snowball** algorithms are two representatives

- For practical work, the new Snowball stemmer is recommended
  - The Porter stemmer is appropriate to IR research

	original_word	stemmed_words
0	connect	connect
1	connected	connect
2	connection	connect
3	connections	connect
4	connects	connect

	original_word	stemmed_word
0	trouble	troubl
1	troubled	troubl
2	troubles	troubl
3	troublemsome	troublemsom

Lemmatization

- Lemmatization is very similar to stemming
  - The goal is to remove inflections and map a word to its root form

- Compared to the stemming, the only difference is that lemmatization actually transforms words to the actual root
  - "better" would map to "good"

- It may use a dictionary such as WordNet for mappings or some special rule-based approaches

	original_word	lemmatized_word
0	trouble	trouble
1	troubling	trouble
2	troubled	trouble
3	troubles	trouble

	original_word	lemmatized_word
0	goose	goose
1	geese	goose

Stop Words

- Stop words are words which are filtered out before or after processing of natural language data (i.e., documents & queries)
  - English Stop Words List
    - I a about an are as at be by com for from how in is it of on or that the this to was what when where who will with the www
  - Chinese Stop Words List
    - 的一不在人有是为以于上他而后之来及了因下可到由这与也此但并个其已无小我们起最再今去好只又或很亦某把那你乃它吧被比别趁当从到得

Normalization

- Text normalization is important for **noisy** texts such as social media comments, text messages and comments to blog posts
  - Abbreviations (縮寫)
  - Misspellings
  - Out-of-vocabulary words

Raw	Normalized
2moro 2mrrw 2morrow 2mrw tomrw	tomorrow
b4	before
otw	on the way
:) :-) ;-)	smile

Noise Removal

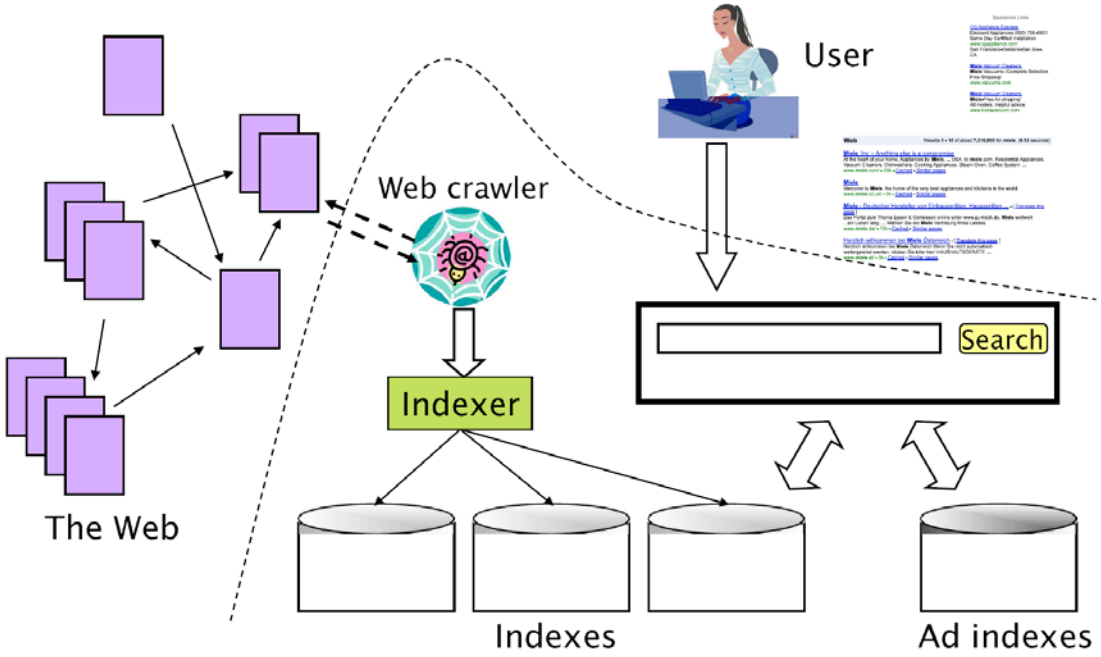
- Noise removal is about removing characters digits and pieces of text that can interfere with your text analysis
  - Noise removal is one of the most essential text preprocessing steps

- It is highly domain dependent
  - For example, in Tweets, noise could be all **special characters except hashtags** as it signifies concepts that can characterize a Tweet

The Representative Engine – Google

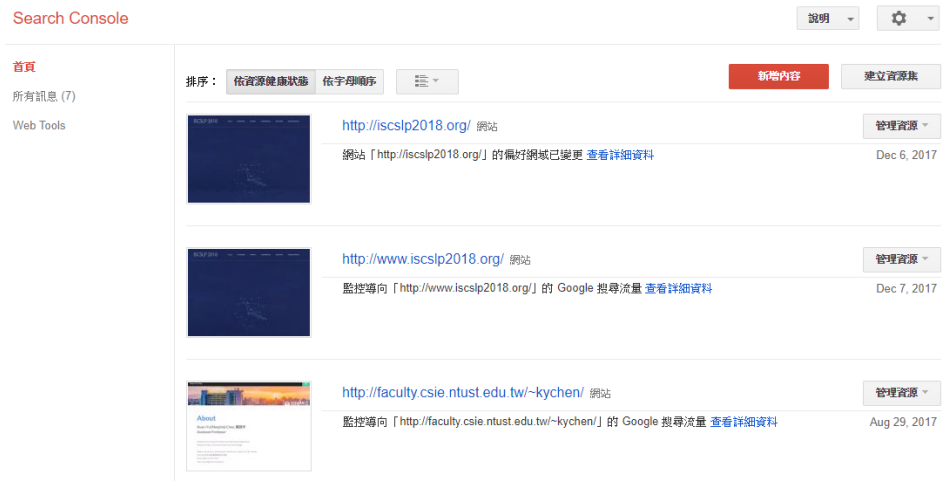
- For a typical query, there are thousands, even millions, of webpages with potentially relevant information

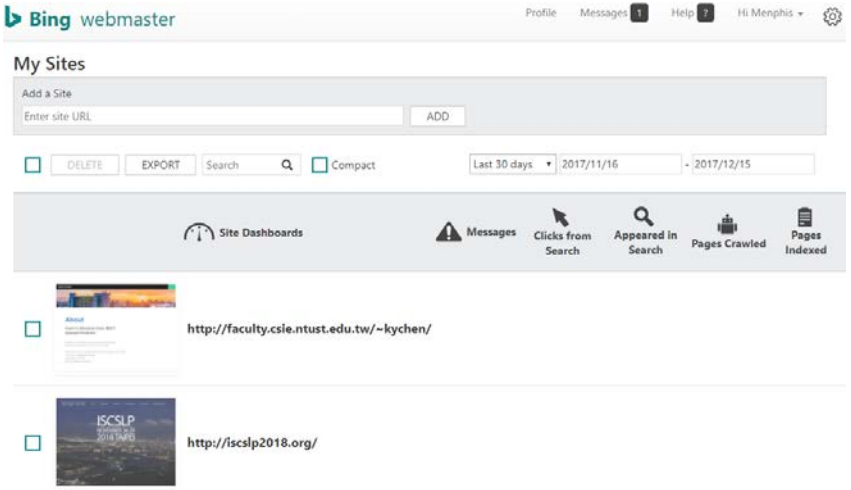
- How does Google figure out what to show in your search results?
  - The journey starts before you even type your search



Crawling

- The web is like an ever-growing library with billions of books and no central filing system
  - Crawlers look at webpages and follow links on those pages
  - They go from link to link and bring data about those webpages back to Google's servers
    - Google: Search Console
    - Bing: webmaster





Indexing.

- Google contains hundreds of billions of webpages and is well over 100,000,000 gigabytes in size

- When Google indexs a web page, they add it to the entries for all of the words it contains
  - Inverted Table!

Indexing..

Doc 1
I did enact Julius Caesar: I was killed i' the Capitol; Brutus killed me.

Doc 2
So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious:

term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

⟹

term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

⟹

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
I	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Indexing…

Doc 1
I did enact Julius Caesar: I was killed i' the Capitol; Brutus killed me.

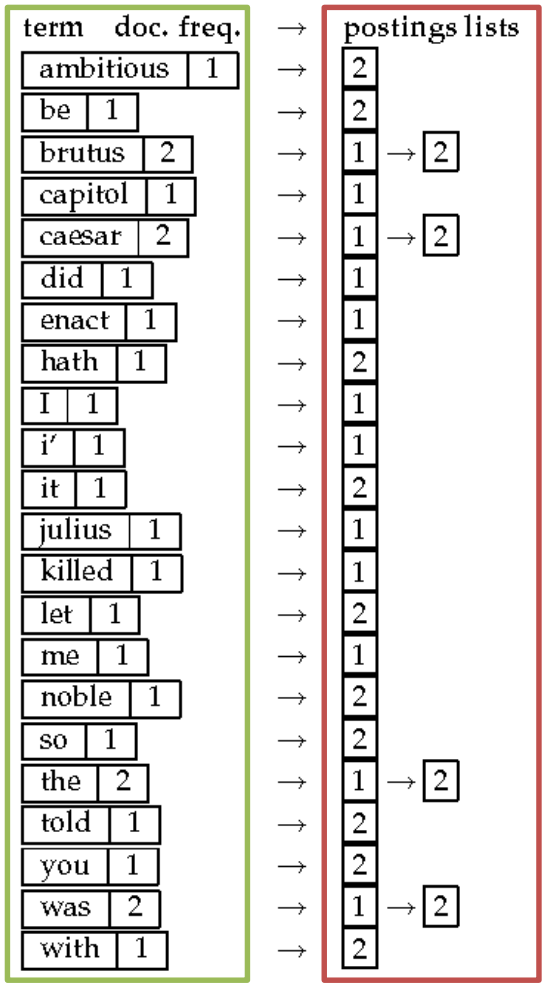
Doc 2
So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious:

term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

⟹

term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

⟹

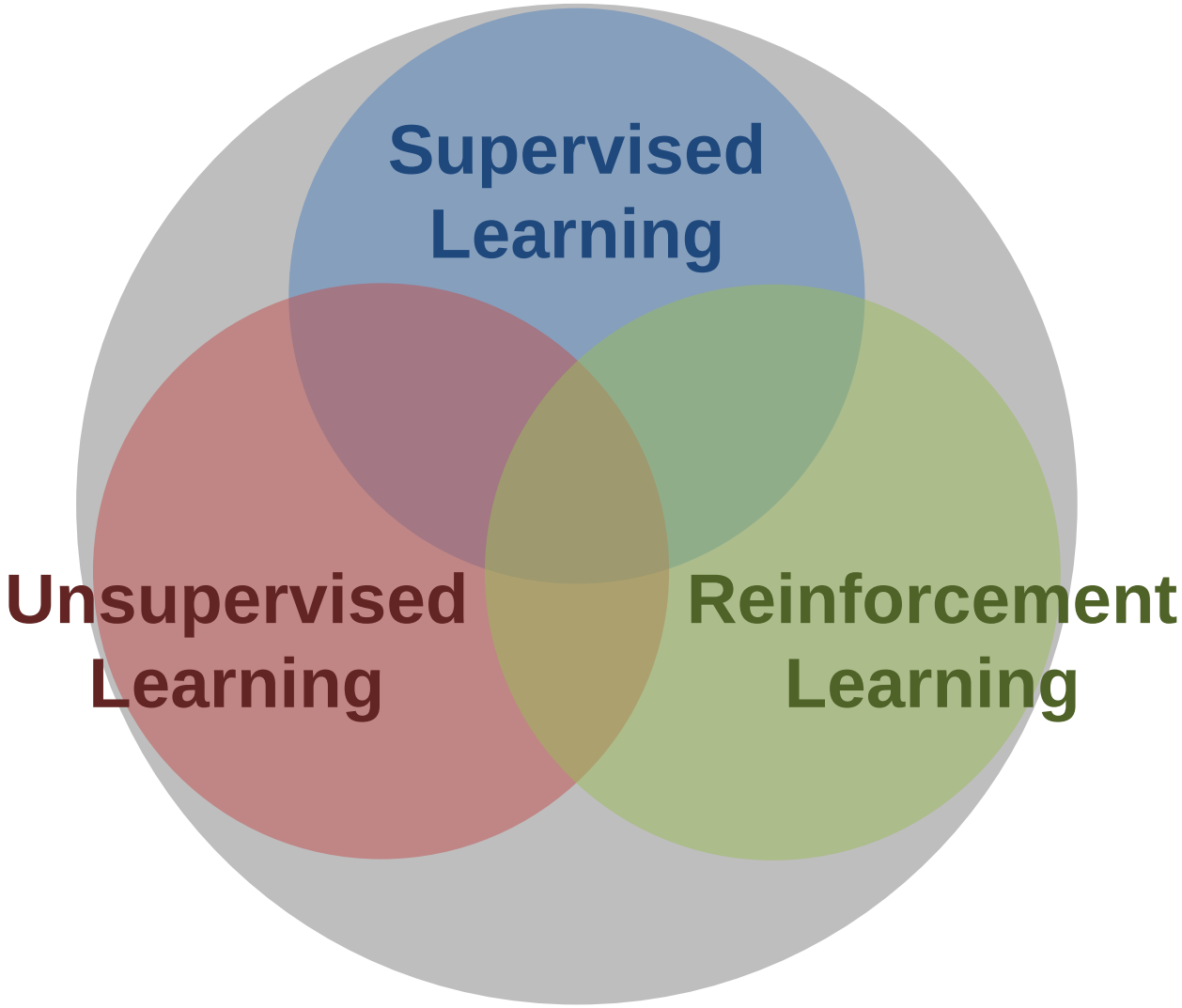


Dictionary (in Memory)

Postings (in HDD)

Search & Presenting

Learning


Supervised Learning
Unsupervised Learning
Reinforcement Learning

Problems with Conventional Models

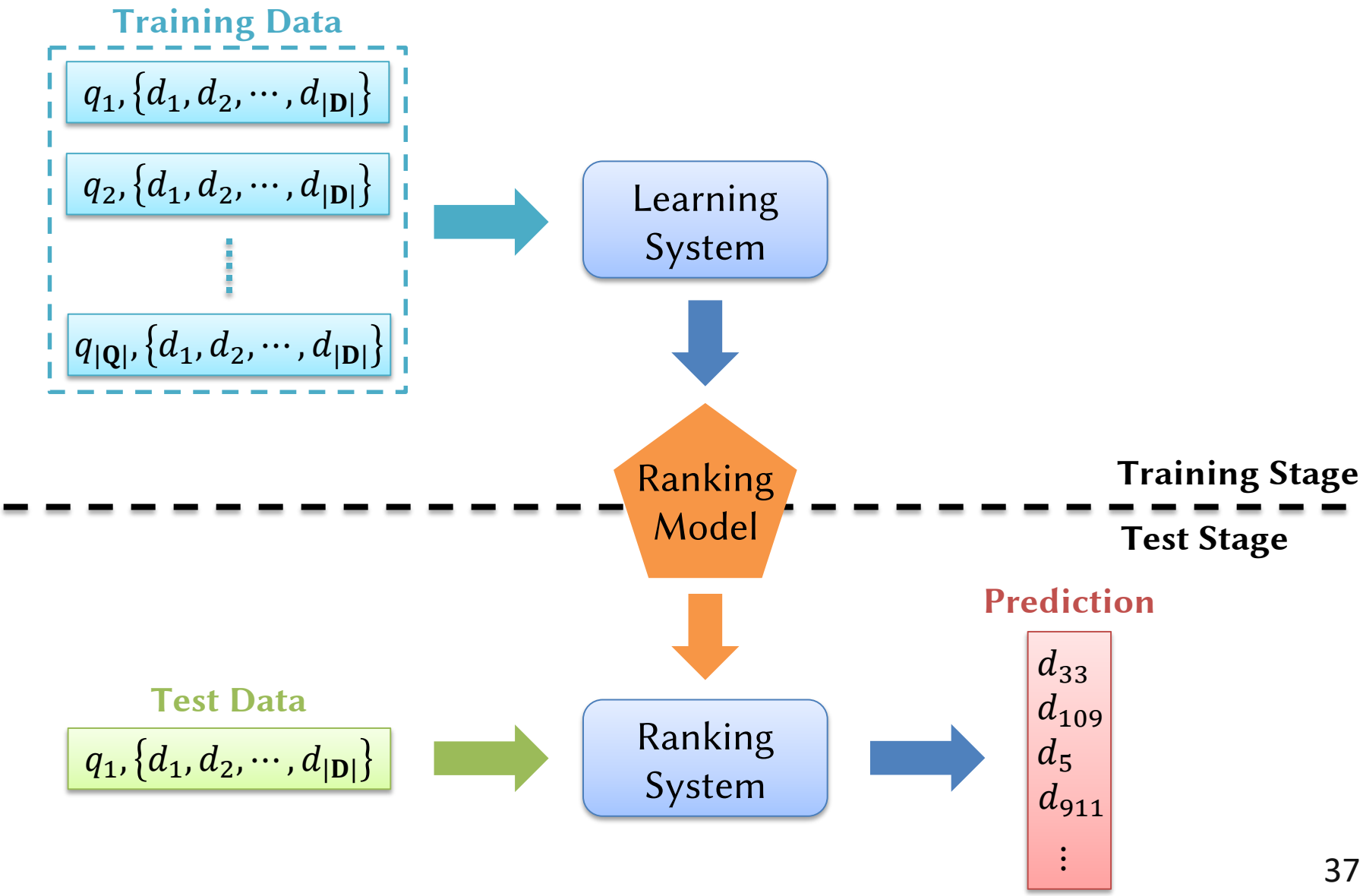
- Parameters tuning is usually difficult
  - Over-fitting vs. Under-fitting
- Non-trivial to combine several methods to be a robust one
- Naïve strategies based on simple features
  - Term-Frequency
  - Inverse-Document Frequency

Learning for Ranking

- Machine learning is an effective way
  - To automatically tune parameters
  - To combine multiple features/models easily
  - To avoid over-fitting by using regularizations

- Learning for ranking (or the well-known **learning to rank** techniques) means that use machine learning technologies to solve the problem of ranking

The General Flowchart



Support Vector Machine

- Features

	Feature		
1	$\sum_{q_i \in q \cap d} \log(c(q_i, d)+1)$	2	$\sum_{q_i \in q \cap d} \log(\frac{\lvert C \rvert}{c(q_i, C)}+1)$
3	$\sum_{q_i \in q \cap d} \log(idf(q_i))$	4	$\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\lvert d \rvert}+1)$
5	$\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\lvert d \rvert} \cdot idf(q_i)+1)$	6	$\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\lvert d \rvert} \cdot \frac{\lvert C \rvert}{c(q_i, C)}+1)$
7	$\log(BM25\ score)$		

- Models
  - For a given query q, if d_i is a relevant document and d_j is an irrelevant document
    - SVM

$$\begin{cases} f_{SVM}(q, d_i) = 1 \\ f_{SVM}(q, d_j) = 0 \end{cases}$$

    - Ranking SVM

$$f_{R-SVM}(q, d_i) > f_{R-SVM}(q, d_j)$$

Simple Regression

- Features

	FEATURE		
1	$\sum_{q_i \in q \cap d} \log(c(q_i, d)+1)$	2	$\sum_{q_i \in q \cap d} \log(\frac{\|C\|}{c(q_i, C)}+1)$
3	$\sum_{q_i \in q \cap d} \log(idf(q_i))$	4	$\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\|d\|}+1)$
5	$\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\|d\|} \cdot idf(q_i)+1)$	6	$\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\|d\|} \cdot \frac{\|C\|}{c(q_i, C)}+1)$
7	$\log(BM25\ score)$		

- Regard relevance degree as real number, and use regression method to learn the ranking function
  - SVR and Neural Networks

$$L(f; q, d) = \left| f(q, d) - l_{q,d} \right|^2$$

Supervised Topic Model – 1

- For PLSA, the training objective is defined to maximize the total log-likelihood of a given training collection
  - The model parameters are $P(d_j)$, $P(w_i|T_k)$, and $P(T_k|d_j)$

$$\mathcal{L} = \sum_{w_i \in V} \sum_{d_j \in \mathbf{D}} c(w_i, d_j) log P(w_i, d_j)$$

$$= \sum_{w_i \in V} \sum_{d_j \in \mathbf{D}} c(w_i, d_j) log \left(P(d_j) \sum_{k=1}^{K} P(w_i|T_k) P(T_k|d_j) \right)$$

Supervised Topic Model – 2

- In the context of retrieval, the ultimate goal is to rank relevant documents in front of the non-relevant documents
  - The conventional PLSA can be used to "encode" document
  - The label data can be used to fine-tune a "decoder"
    - The objective is to maximize the total query likelihood generated by its relevant documents
    - The topic distribution for each document $P(T_k|d)$ is fixed and can be obtained by encoder!

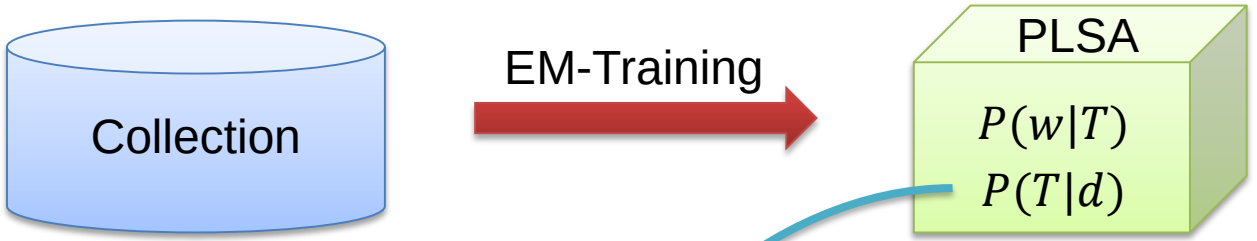
$$\mathcal{L} = \sum_{w_i \in V} \sum_{q \in \mathbf{Q}} \sum_{d_j \in R_q} c(w_i, q) log P(w_i, d_j)$$

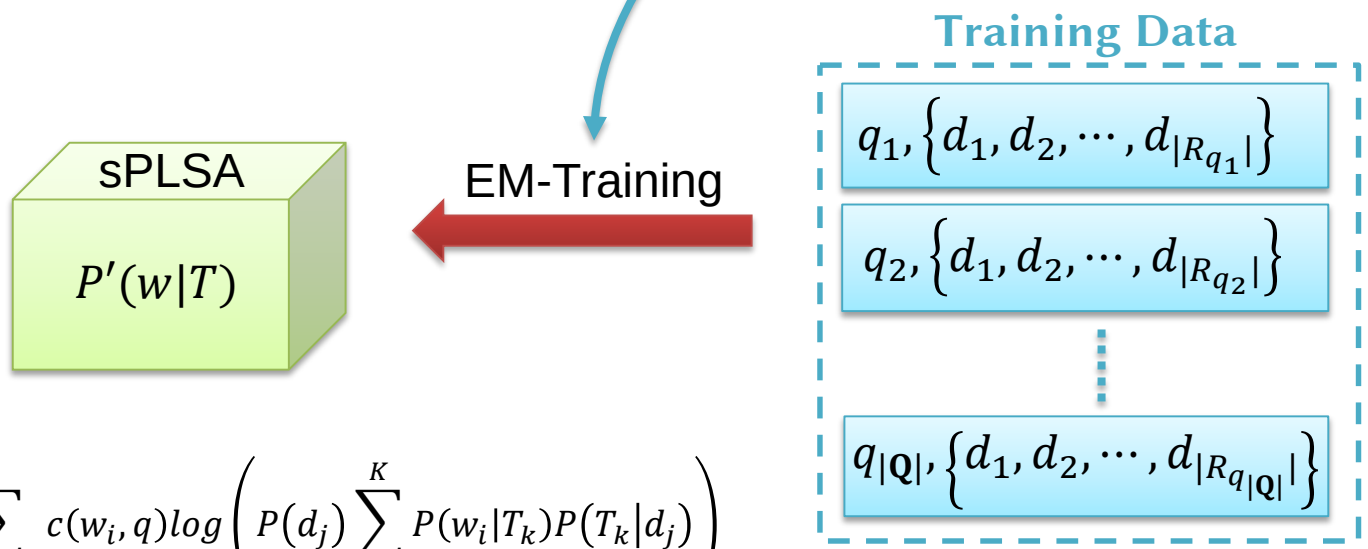
$$= \sum_{w_i \in V} \sum_{q \in \mathbf{Q}} \sum_{d_j \in R_q} c(w_i, q) log \left(P(d_j) \sum_{k=1}^{K} P(w_i|T_k) P(T_k|d_j) \right)$$

$$\propto \prod_{q \in \mathbf{Q}} \prod_{d_j \in R_q} P(q|d_j)$$

Supervised Topic Model – 3

- For Training

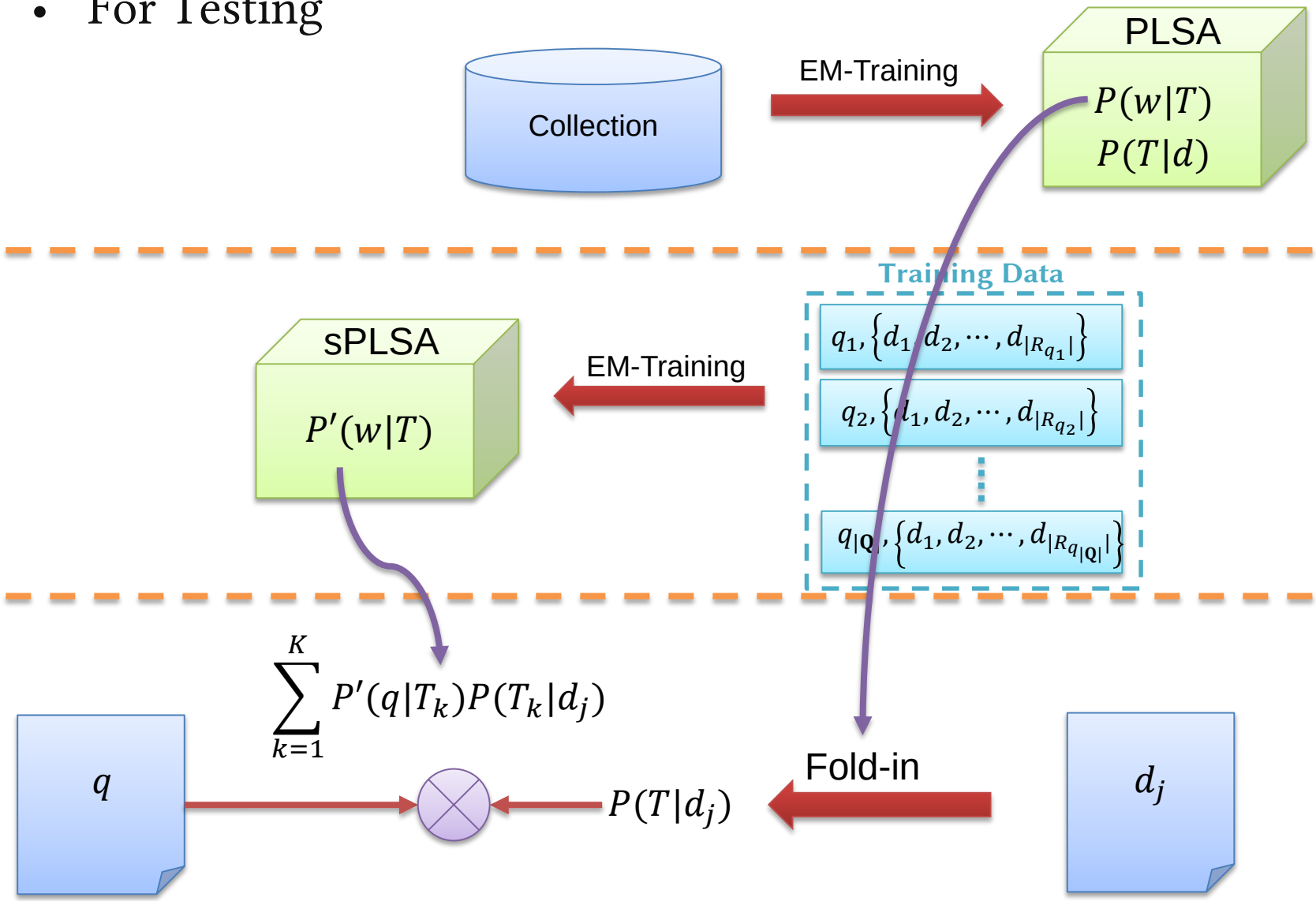




$$\mathcal{L} = \sum_{w_i \in V} \sum_{q \in \mathbf{Q}} \sum_{d_j \in R_q} c(w_i, q) log \left(P(d_j) \sum_{k=1}^{K} P(w_i|T_k) P(T_k|d_j) \right)$$

Supervised Topic Model – 4

- For Testing



Triplet-based Metric Learning – 1

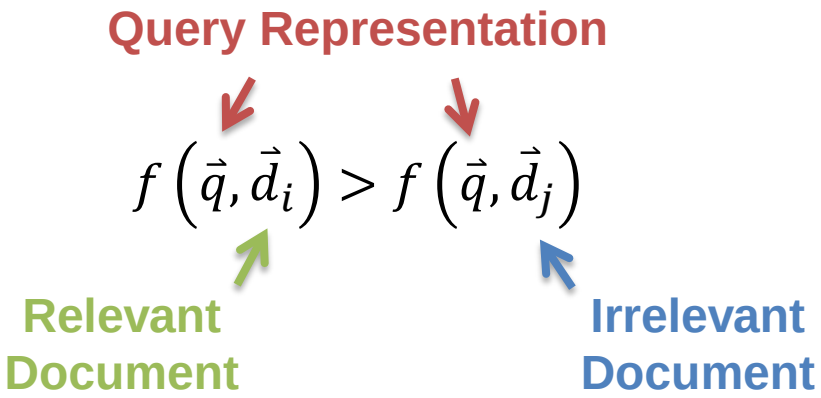
- Although the cosine similarity measure intuitively can be used to determine the relevance degree
  - Such framework ignores the inter-dimensional correlation between two representations
  - The **Triplet Learning Modeling**
    - For a given query q, if d_i is a relevant document and d_j is an irrelevant document

$$\vec{q} = \sum_{w \in q} \frac{c(w,q)}{|q|} v_w \qquad \vec{d}_i = \sum_{w \in d_i} \frac{c(w,d_i)}{|d_i|} v_w \qquad \vec{d}_j = \sum_{w \in d_j} \frac{c(w,d_j)}{|d_j|} v_w$$

G. Chechik, *et al.*, "Large Scale Online Learning of Image Similarity Through Ranking," *Journal of Machine Learning Research*, vol. 11, pp. 1109-1135, 2010.

Triplet-based Metric Learning – 2

- Without loss of generality, the goal is to learn a similarity function that assigns higher similarity scores to relevant documents than irrelevant documents:



  - The parametric ranking function has a bi-linear form:

$$f\left(\vec{q},\vec{d}_i\right) \equiv \vec{q}^T \mathbf{M} \vec{d}_i$$

where $\mathbf{M}$ is a square parametric matrix

Triplet-based Metric Learning – 3

- Followed by the **Passive-Aggressive learning algorithm**, we aim to derive a similarity function such that all triplets obey:

$$f\left(\vec{q}, \vec{d}_i\right) > f\left(\vec{q}, \vec{d}_j\right) + \delta$$

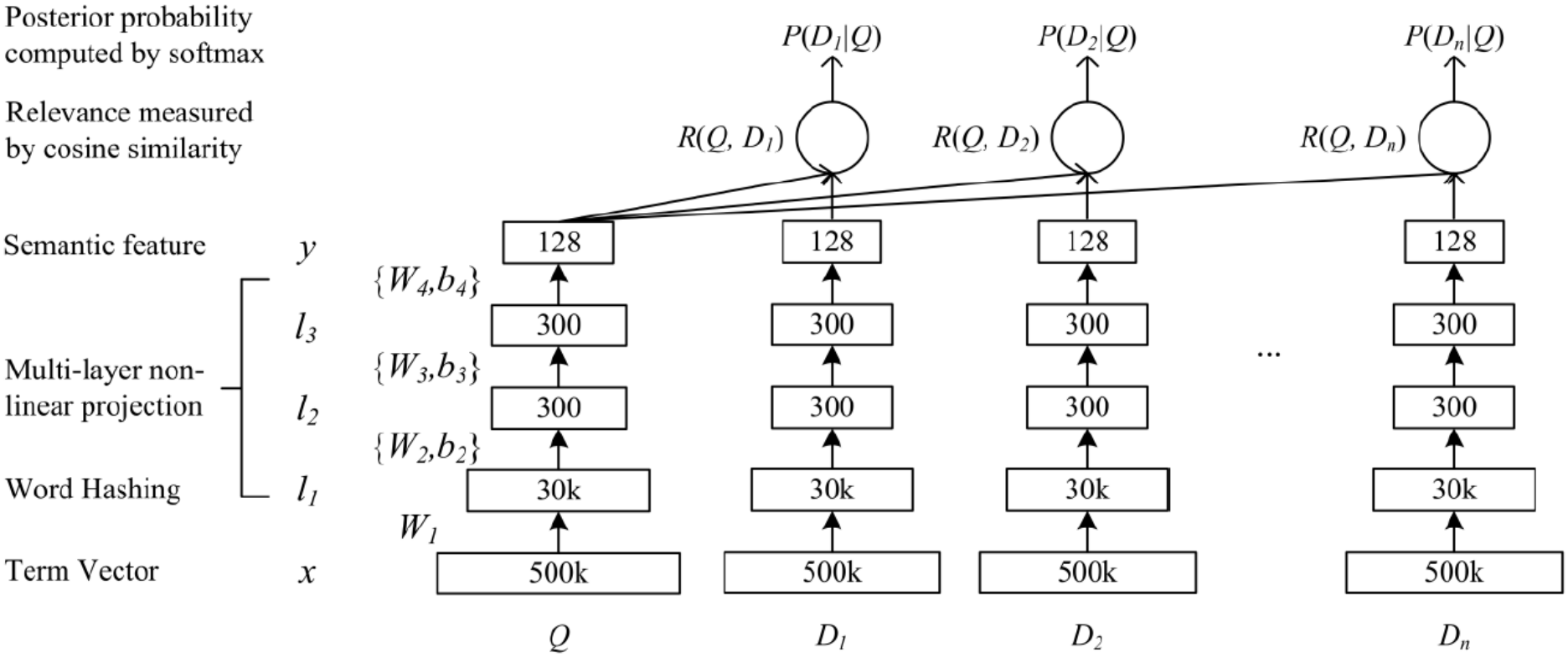
- Finally, a hinge loss function can be defined for the triplet learning:

$$loss\left(\vec{q}, \vec{d}_i, \vec{d}_j\right) = max\left\{0, \delta - f\left(\vec{q}, \vec{d}_i\right) + f\left(\vec{q}, \vec{d}_j\right)\right\}$$

  - By applying the sequential learning algorithm iteratively over triplets, a solution **M** can be derived

P.-S. Huang, *et al.*, "Learning deep structured semantic models for web search using clickthrough data," in *Proc. of CIKM*, 2013.

Deep Structured Semantic Model (DSSM)

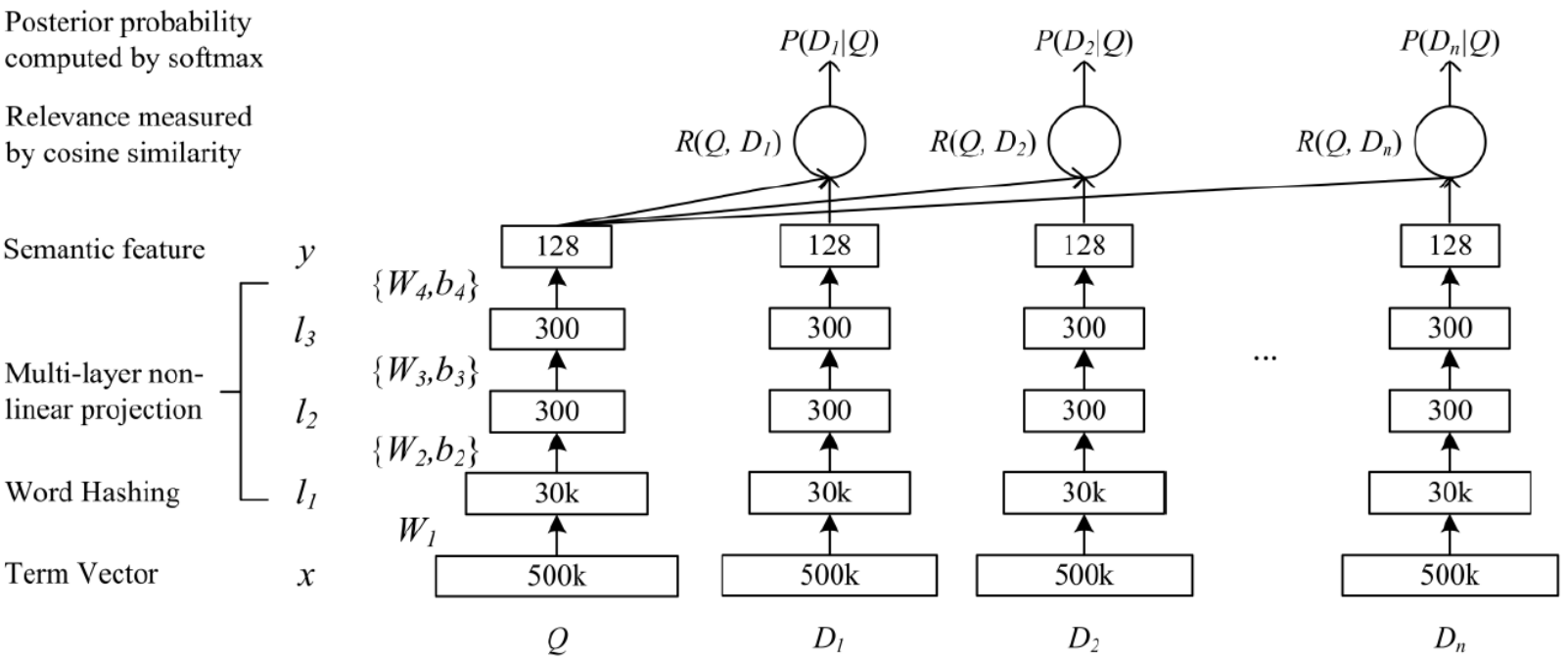


	Letter-Bigram		Letter-Trigram	
Word Size	**Token Size**	**Collision**	**Token Size**	**Collision**
40k	1107	18	10306	2
500k	1607	1192	30621	22

#good# => [#go, goo, ood, od#]

P.-S. Huang, *et al.*, "Learning deep structured semantic models for web search using clickthrough data," in *Proc. of CIKM*, 2013.

DSSM



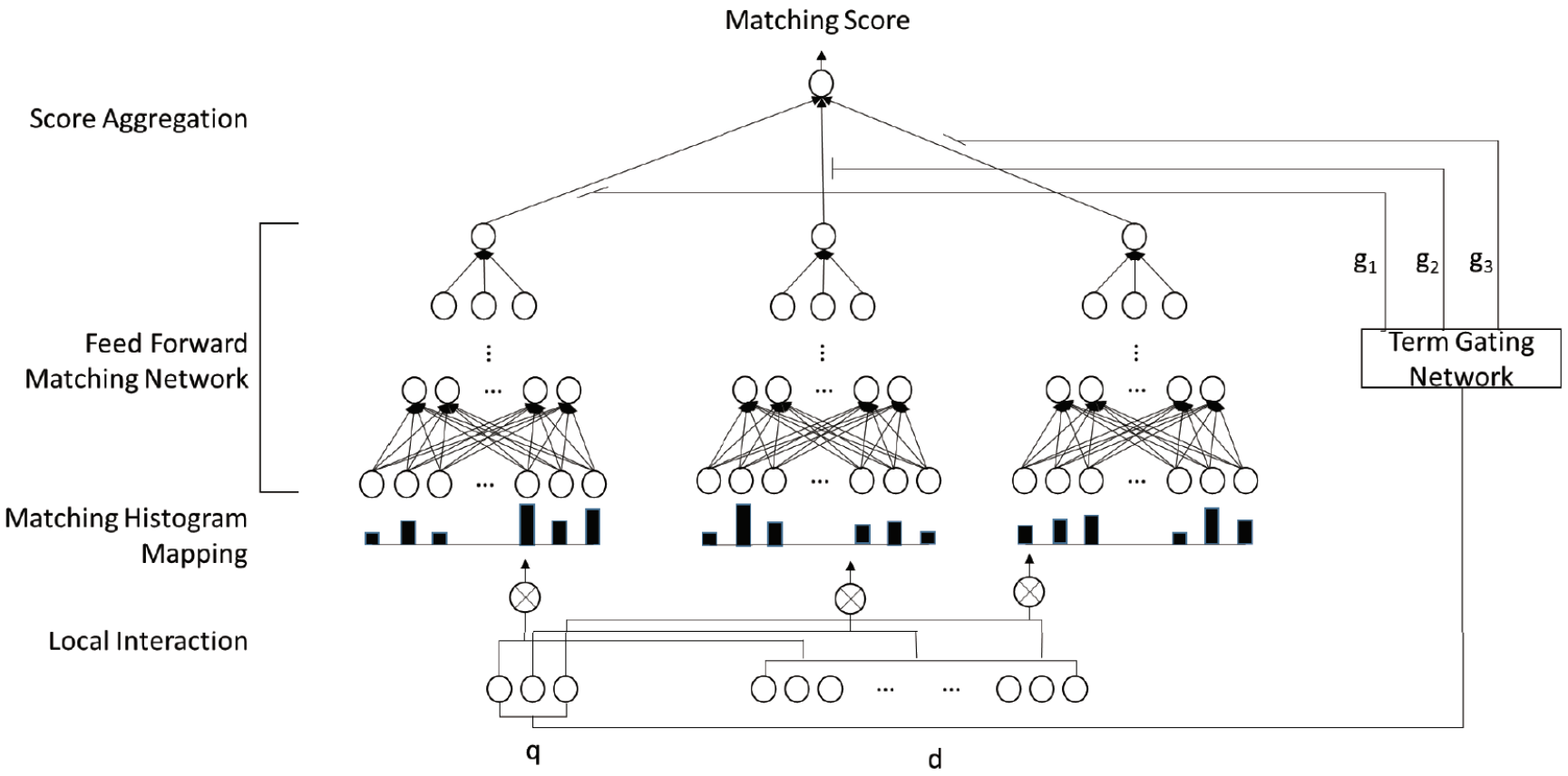
$$R(q,d) = \cos(\vec{q}, \vec{d})$$

$$P(d|q) = \frac{\exp(R(q,d))}{\sum_{d'} \exp(R(q,d'))}$$

$$L = \prod_{d \in R_q} P(d|q)$$

J.-F. Guo, *et al*., "A deep relevance matching model for ad-hoc retrieval," in *Proc. of CIKM*, 2016.

Deep Relevance Matching Model



Query: "car to go"
Document: "car, rent, truck, bump, injunction, runway"
Five Bins: {[-1,-0.5), [-0.5,0), [0,0.5), [0.5,1), [1,1]}
Local Interaction for "car": (1, 0.2, 0.7, 0.3, -0.1, 0.1)
Matching Histogram for "car": [0, 1, 3, 1, 1]

Point-wise, Pair-wise and List-wise

- Pointwise approaches concentrate on a **single document** at a time in the loss function
  - SVM, SVR, Supervised Topic Model, DRMM
- Pairwise approaches focus on a **pair of documents** at a time in the loss function
  - Triplet Learning, DSSM
- Listwise approaches directly look at the entire **list of documents** and try to come up with the optimal ordering for the set of documents

NN-based Language Models



Language Representations (2013~)

Neural Network Language Models (2001~)

Continuous Language Models

Continuous Language Models (2007~2009)

Topic Models (1997~2003)


Topic Models

Query Language Models (2001~2006)


Word-Regularity Models

Discriminative Language Models (2000~2011)




Word-Regularity Models (~1997)

2000 2002 2004 2006 2008 2010 2012 2014 2016

Questions?

kychen@mail.ntust.edu.tw