Deep Structured Semantic Model (DSSM) for Web Search using Clickthrough Data

Overview of Task

- Task : Ranking documents with respect a query using clickthrough information
- Input: {Query (Q), set of clickthrough documents for that query}
- Output: Documents ranked in order of their relevance with respect to a query.

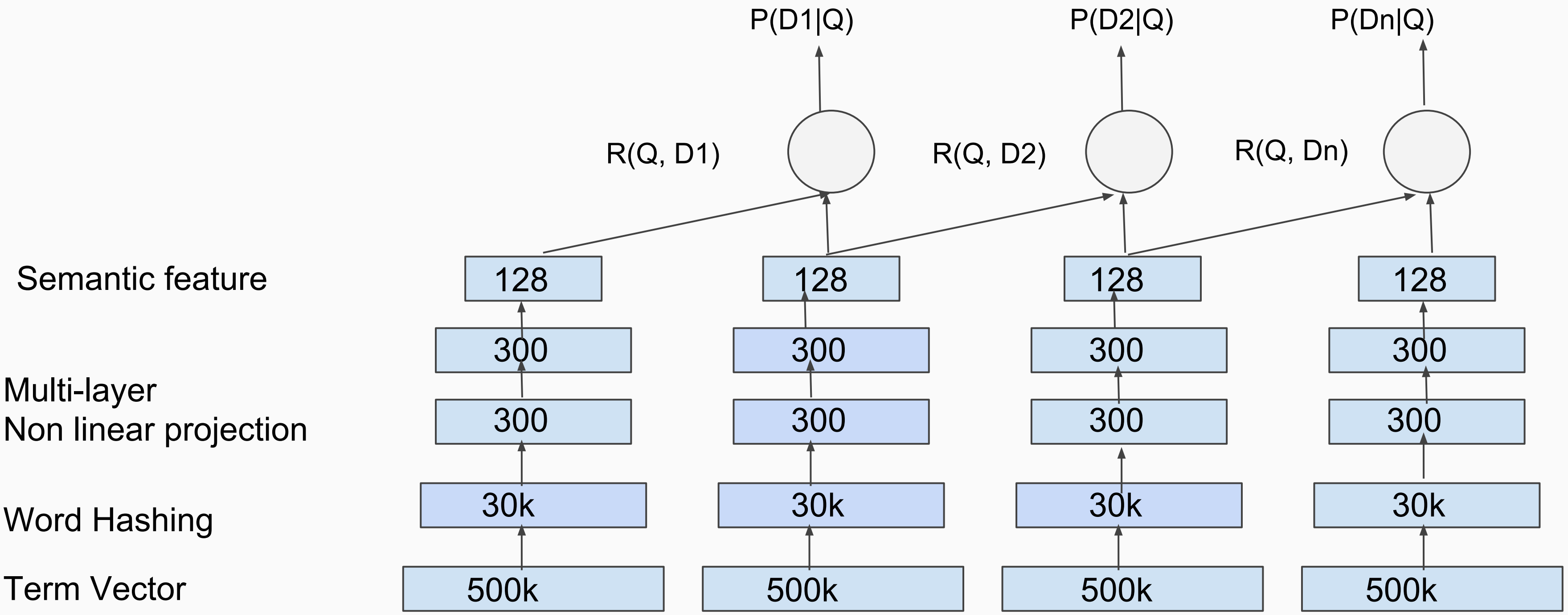
Why Semantic Representation ?

- In modern search engines, web document retrieval is mainly done by lexical matching between query and the web documents.

- Fact and concepts often expressed using different vocabularies and language styles.

  I like playing cricket
  I enjoy playing cricket

DSSM Model



P(D1|Q)
P(D2|Q)
P(Dn|Q)
R(Q, D1)
R(Q, D2)
R(Q, Dn)
Semantic feature
128
128
128
128
300
300
300
300
Multi-layer
Non linear projection
300
300
300
300
Word Hashing
30k
30k
30k
30k
Term Vector
500k
500k
500k
500k

Word Hashing

Words are represented using letter n-grams(tri-grams).

E.g: good.

Add delimiters(#good#).

Break into letter tri- grams (#go,goo,ood,od#)

Drawback of letter n-gram representation : Collision

How to identify Collision?

The hash codes need to be stored in an array.

The hash code of the 3kth n-gram in the document is stored at array index k. Then, the array need to be scanned for recurring hash codes.

A recurring code could indicate a collision, if the two occurrences of the code derived from different n-grams, or a non-collision, if the two occurrences derived from different occurrences of the same n-gram.

Properties and Observations of Word Hashing

- It maps morphological variations of same word to the point that are close to each other in letter n-gram space.
- It overcomes the problem of representation of unseen words.

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

Table 1: Word hashing token size and collision numbers as a function of the vocabulary size and the type of letter ngrams.

Learning The DSSM

Clickthrough logs contain:

- List of queries
- Corresponding clicked documents

Aim:To maximize the conditional likelihood of the clicked documents given the queries

Learning The DSSM

- Compute posterior probability of a document given a query.

$$P(D|Q) = \frac{\exp\left(\gamma R(Q,D)\right)}{\sum_{D' \in \mathbf{D}} \exp\left(\gamma R(Q,D')\right)}$$

Where γ is smoothing factor of softmax function.

D:set of candidate documents, D+ = clicked doc, D- = Unclicked Doc

Learning The DSSM

- Minimize loss function:

$$L(\Lambda) = -\log \prod_{(Q,D^+)} P(D^+|Q)$$

Where Λ denotes parameter set of DNN.

Model is trained using gradient based numerical optimization algorithms.

Implementation details

- 3 hidden layers
  - Hashing layer of about 30k nodes.
  - Two layers with 300 nodes.
  - Output layer of 120 dimensions.
- Dataset
  - 16510 english queries.
  - Each query has 15 documents, with relevance score for each document.

Evaluation

For evaluation, NDCG is used.

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p},$$

Where $\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$ and $\mathrm{IDCG_p} = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)}$

rel_i is available on a 0-4 scale.

Note: Only query-title pairs are used.

Results

Comparison against various models

TF-IDF	0.462
BM25	0.455
WTM	0.478
LSA	0.455
PLSA	0.456
DAE	0.459
BLTM-PR	0.480
DPM	0.479
DNN	0.486
L-WH linear	0.495
L-WH Non-lin	0.494
L-WH DNN	0.498

Few related points

- C-DSSM: An extension using convolutional neural networks.
- Became popular as Sent2Vec and did not just limit to web search.