Learning to rank using lexical and query information such as a word sense and distributional representation

Research goal
1. Build an effective ranking model with heterogeneous features
2. Employ semantic information as novel and promising features
3. Explore trends obtained by traditional/machine learning techniques

Motivation
• Some studies argue NLP techniques are useful for effective IR
  - POS/word sense/distributed representation (DR)
  - Systematic knowledge is not acquired
• The neural network model is often applied
  - Used for term weighting including query expansion
  - More accurate, but not a breakthrough so far
• Problems caused by queries’ characteristics
  - Short length, Random word order
  > Neither extract sufficient contexts nor construct an accurate language model
Enrich a query (keyword) with latent semantic information in terms of lexical and query levels

Lexical semantic info.
• Term as a symbol (in IR community)
  - A combination of terms identifies a topic of a query
  > Inactive to disambiguate a word sense
  - Query expansion based on similarity/co-occurrence between terms
  > Word semantics are not considered → Not always perform well
Expected to take account of semantic information
• DR (*2vec) is widely employed
  - Granularity: term, phrase/query, sentence, paragraph, doc

Query semantic info.
• Query task identification
  - Web queries are classified into 20 tasks [1]
    > Simple information need: Weather, stock price
    > Complex information need: QA, definition
  Domain dependent model is more effective
• Query intent
  - A specific interpretation of an ambiguous query or an aspect of a faceted query
    > Query: “real Madrid” -> Intents: “Players”, “History”

Preliminary experiment
• Distributed representation
  - Original doc2vec (300-D) decreases effectiveness
    > Ranking differs from classification or prediction
  - Similarity with a query (scalar value) performs better
• Word sense
  - Improve precision at top-ranked results
  - Filtering also improves effectiveness
• Overall
  - Performance of LTR is unstable
    > L2R does not always outperform a traditional retrieval model
    > # of retrieved doc (top, overall). Query (difficulty, # of relevant data), learning algorithm

Proposed framework
• Two-layered learning to rank
  - Construct much better model to improve insufficient learning
  - Employ more orthogonal features to alleviate correlations