Short Text Understanding Through Lexical-Semantic Analysis*

Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, Xiaofang Zhou

Introduction

- Short Text Understanding = Semantic Labeling
  - **Text Segmentation** – divide text into a sequence of terms in vocabulary
  - **Type Detection** – determine the best type of each term
  - **Concept Labeling** – infer the best concept of each entity within context

- Applications
  - Calculate semantic similarity between short texts
  - Identify interest
  - Query recommendation/clustering/classification

- Challenges
  - Limited Content: query < 5 words and tweet < 140 characters
  - Incorrect Syntax: “microsoft office download free”
  - Segmentation Ambiguity: “april in paris lyrics / vacation”
  - Type Ambiguity: “pink shoes / songs”
  - Entity Ambiguity: “watch harry potter” vs. “read harry potter”

Knowledge-Intensive Approaches

- Find the best segmentation from a set of candidate terms contained in a pre-defined machine readable vocabulary
  - best – topically coherent
- Mutual Exclusion & Mutual Reinforcement
- Build a Candidate Term Graph (CTG)
  - Best segmentation = sub-graph in CTG which: 1) Is a complete graph (clique); 2) Has 100% word coverage; Has largest average edge weight
  - Theorem: finding a clique with 100% word coverage is equivalent to retrieving a Maximal Clique from the original CTG.
  - Best segmentation = Maximal Clique with largest average edge weight
  - NP-hard -> Approximation algorithm based on Monte Carlo

Framework

- Maximize total score of consecutive terms
- Verify generalizability, we randomly sampled 400 queries containing these terms commonly used to illustrate ambiguity and randomly sampled 11*100 queries

Experiment

- Find the best concept of each entity within context
  - Filtering/re-rank of the original concept cluster vector
- The final score of each concept cluster is a combination of its original score and the support from other terms

Concept Labeling

- Infer the best concept of each entity within context
  - Filtering/re-rank of the original concept cluster vector
  - The final score of each concept cluster is a combination of its original score and the support from other terms

Type Detection

- Determine the best type of each term in a segmentation of a short text
  - Verbs, adjectives, attributes, concepts, entities ...
- Chain Model - Consider relatedness between consecutive terms; Maximize total score of consecutive terms
- Pairwise Model - Most related terms might not always be adjacent; Find the best type for each term so that the Maximum Spanning Tree of the resulting sub-graph between typed-terms has the largest weight

Co-occurrence Network

- What knowledge is required for short text understanding?
  - Knowledge about *vocabulary*: verbs, adjectives, attributes, concepts, entities
  - Knowledge about *entity-concept* relation: “harry potter” is a book, a movie, a character...
  - Knowledge about *semantic relatedness*: “harry potter” as a book is related with “read” / “harry potter” as a movie is related with “watch” / “harry potter” as a character is related with “age” ...

- Construct *co-occurrence network*
  - A single term with different types co-occurs with different context. Build co-occurrence network between typed-terms.
  - Two typed-terms are related if they often co-occur in a sentence within short distance;
  - Vague typed-terms (“item”, “object”) or typed-terms that co-occur with almost every other typed-term are meaningless in modeling semantic relatedness.

- Compress co-occurrence network
  - Reduce cardinality
  - Improve inference accuracy

Experiments

- Benchmark and evaluation methods
  - To verify the effectiveness of disambiguation, we chose 11 terms commonly used to illustrate ambiguity and randomly sampled 11*100 queries containing these terms: “april in paris”, “hotel california”, “watch”, “book”, “pink”, “blue”, “orange”, “population”, “birthday”, “apple”, “fox”
  - To verify generalizability, we randomly sampled 400 queries
- Experimental results
  - Improve the accuracy of short text understanding over state-of-the-art approaches by up to 30%
  - Understand most of the short texts within 50ms in average

* This work was partially done at Microsoft Research Asia