Short Text Understanding Through Lexical-Semantic Analysis

Wen Hua [§1], Zhongyuan Wang [†2], Haixun Wang [#3], Kai Zheng [*4], Xiaofang Zhou [*5]

[§] *School of Information, Renmin University of China, Beijing, China*
[1] huawenr@ruc.edu.cn

[†] *Microsoft Research, Beijing, China*
[2] zhy.wang@microsoft.com

[#] *Google Research, U.S.A.*
[3] haixun@google.com

[*] *School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, Australia*
[4] kevinz@itee.uq.edu.au [5] zxf@itee.uq.edu.au

What?

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

wanna watch eagles band

↓

<u>watch</u> <u>eagles</u> <u>band</u>

↓

<u>watch</u>$_{\text{[verb]}}$ <u>eagles</u>$_{\text{[entity]}}$ <u>band</u>$_{\text{[concept]}}$

↓

<u>watch</u>$_{\text{[verb]}}$ <u>eagles</u>$_{\text{[entity](band)}}$ <u>band</u>$_{\text{[concept]}}$

Why?

- Applications
  - Web search, microblogging, ads matching, etc.
- Challenges
  - Incorrect syntax
  - Limited content
  - More **ambiguous**

Example 1 (Ambiguity in Text Segmentation):

- "april in paris lyrics" vs. "vacation april in paris"
- "book hotel california" vs. "hotel california eagles"

Example 2 (Ambiguity in Type Detection):

- "pink$_{[e](singer)}$ songs" vs. "pink$_{[adj]}$ shoes"
- "watch$_{[v]}$ free movie" vs. "watch$_{[c]}$ omega"

Example 3 (Ambiguity in Concept Labeling):

- "hotel california eagles$_{[e](band)}$" vs. "jaguar$_{[e](brand)}$ cars"

How?

- Traditional NLP approaches fail
  - Only lexical features
- Humans succeed
  - Semantic knowledge
- This work
  - Use **lexical-semantic** knowledge provided by a well-known semantic network for short text understanding

Outline

- Preliminaries
- Methods
  - Text Segmentation
  - Type Detection
  - Concept Labeling
- Experiments
- Conclusion

Preliminaries – Notations

	Definition	Example
s	short text	book hotel california
p	segmentation	{<u>book</u> <u>hotel</u> <u>california</u>}
t	term	hotel,california,hotel california
$\bar{t}$	typed-term	$\text{book}_{[v]}$,$\text{book}_{[c]}$,$\text{book}_{[e]}$
$\bar{t}.r$	type	v,adj,att,c,e
$\bar{t}.\vec{c}$	concept vector	(theme park,company,park...)
$\bar{t}.\vec{C}$	concept cluster vector	({theme park,park},{company}...)

Preliminaries – Probase

- Is-A Network
  - Instance -> concept

$$\bar{t}.\vec{C} = \begin{cases} \emptyset & \bar{t}.r \in \{v, adj, att\} \\ (< C, 1 > | \bar{t} \in C) & \bar{t}.r = c \\ (< C_i, W_i > | i = 1, ..., N) & \bar{t}.r = e \end{cases}$$

- Co-occurrence Network
  - Node -> typed-term
  - Edge -> co-occurrence
    - Weight -> strength of relatedness

$$w(\bar{x}, \bar{y}) = \frac{f(\bar{x}, \bar{y})}{\sum_{\bar{z}} f(\bar{x}, \bar{z})} \cdot \log \frac{N}{N_{nei(\bar{y})}}$$

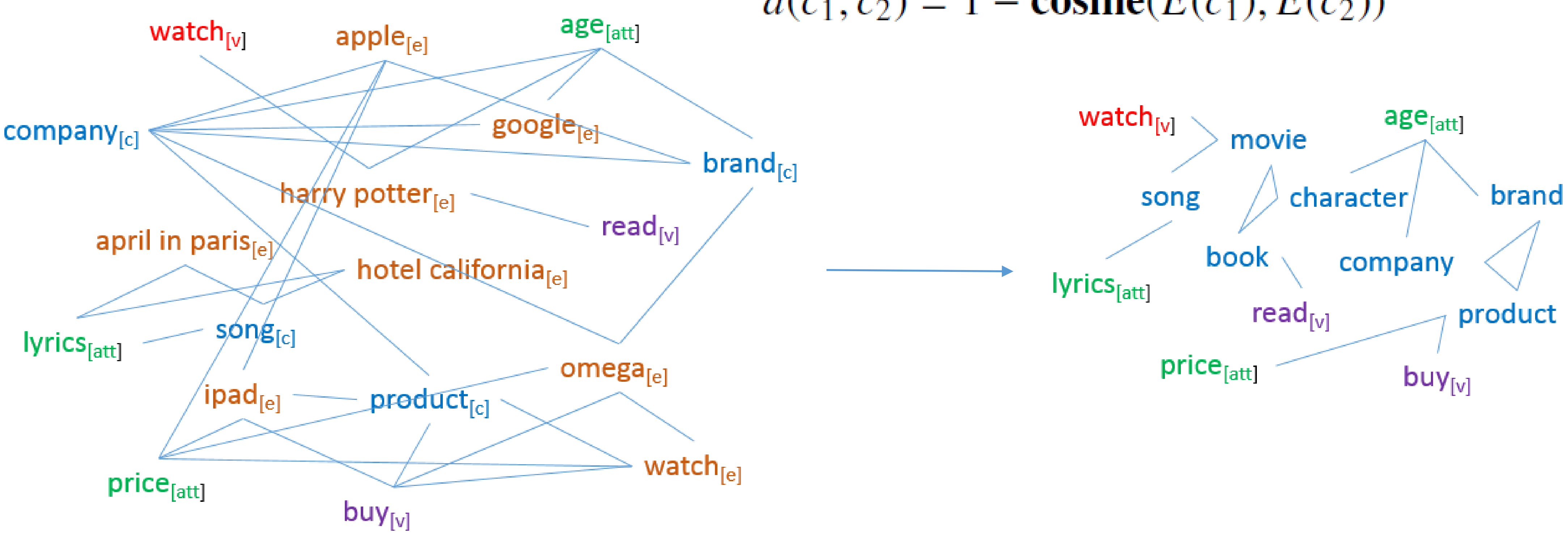
$$f(\bar{x}, \bar{y}) = \sum_{s} f_s(\bar{x}, \bar{y})$$

$$f_s(\bar{x}, \bar{y}) = n_s \cdot e^{-dist_s(\bar{x}, \bar{y})}$$

Preliminaries – Probase (Cont.)

- Compress co-occurrence network
  - K-Mediods

$$d(c_1, c_2) = 1 - \textbf{cosine}(E(c_1), E(c_2))$$



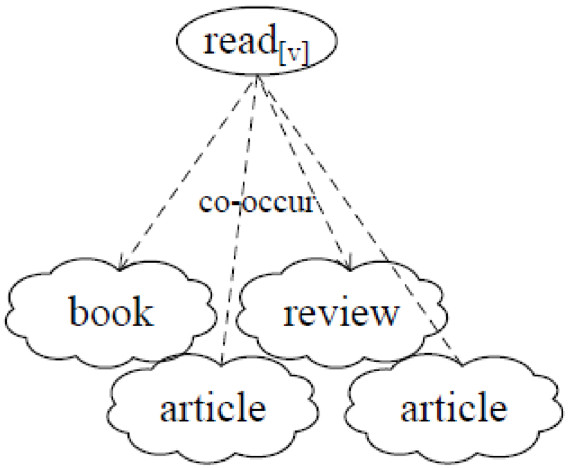
Preliminaries – Probase (Cont.)

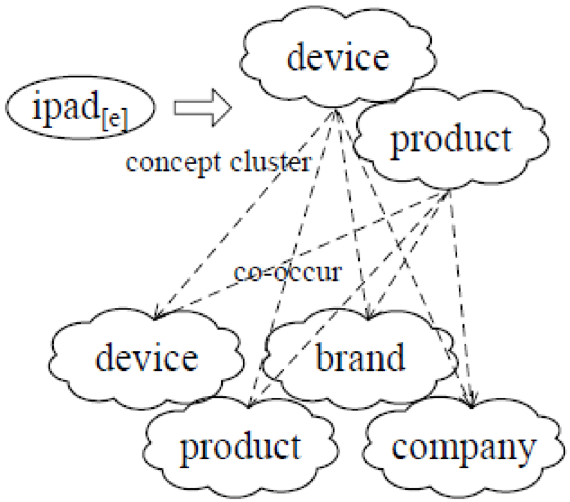
- Scoring Semantic Coherence
  - Affinity Score (AS) -> measure semantic coherence between typed-terms

$$S(\bar{x}, \bar{y}) = \mathbf{max}(S_{sim}(\bar{x}, \bar{y}), S_{co}(\bar{x}, \bar{y}))$$

$$S_{sim}(\bar{x}, \bar{y}) = \mathbf{cosine}(\bar{x}.\vec{C}, \bar{y}.\vec{C})$$

$$S_{co}(\bar{x}, \bar{y}) = \mathbf{cosine}(\vec{C}_{co(\bar{x})}, \bar{y}.\vec{C})$$

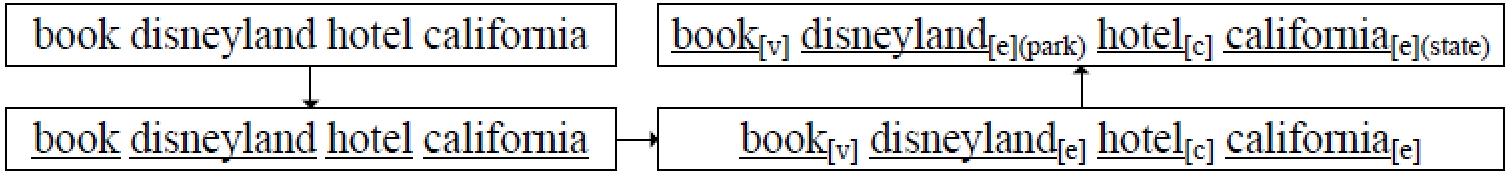




Preliminaries – Problem Definition

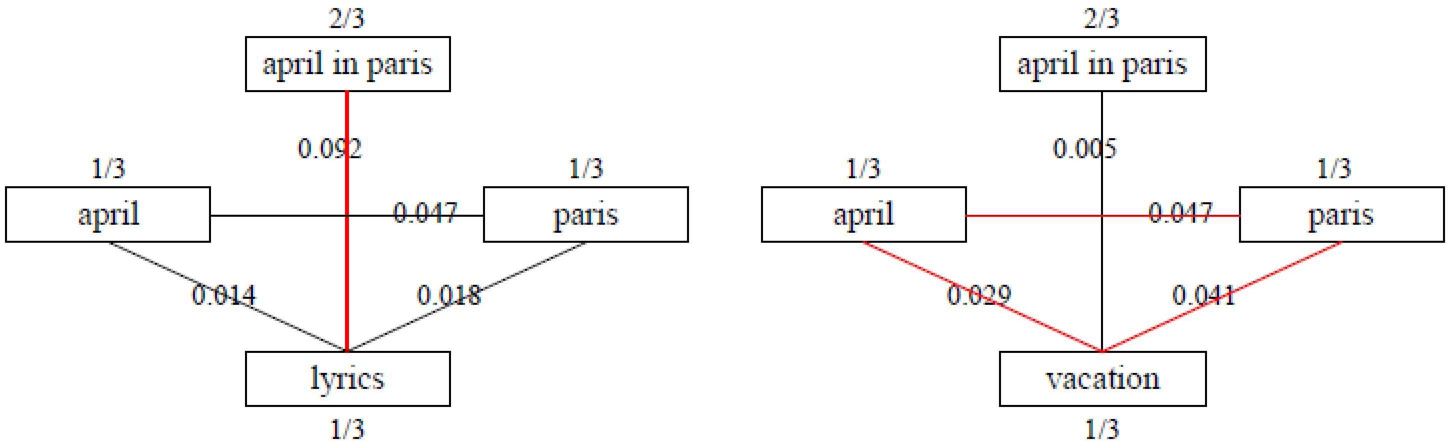
Definition 6 (Short Text Understanding): For a short text s in natural language, generate a semantic interpretation of s, which is represented as a sequence of typed-terms, namely $\bar{s} = \{\bar{t}_i | i = 1, ..., l\}$.

1. **Text Segmentation**. Given a short text s, find the best segmentation p^*.
2. **Type Detection**. For term t, find the best typed-term $\bar{t}^*$ in the context.
3. **Instance Disambiguation**. For any instance $\bar{t}$ with possible senses (concept clusters) $\vec{C} = (C_1, C_2, ..., C_N)$, rank the senses with regard to the context.



Text Segmentation

- Good segmentation
  - Mutual Exclusion
  - Mutual Reinforcement
- Build Term Graph (TG)
  - Node -> candidate term
    - Weight -> coverage
  - Edge -> not mutually exclusive
    - Weight -> strength of mutual reinforcement



$$w(x, y) = \max(\epsilon, \max_{i,j} S(\bar{x}_i, \bar{y}_j))$$

Text Segmentation (Cont.)

- Finding the best segmentation
  - Retrieving a Maximal Clique with the largest average edge weight from the TG
- Brute Force Algorithm
  - NP-hard with exponential time complexity
- Randomized algorithm
  - Approximate solution with polynomial time complexity

Text Segmentation (Cont.)

Algorithm 1 Maximal Clique by Monte Carlo (MaxCMC)

Input:
$G = (V, E)$; $W(E) = \{w(e) | e \in E\}$

Output:
$G' = (V', E')$; $s(G')$

$V' = \emptyset$; $E' = \emptyset$
while $E \neq \emptyset$ **do**
  randomly select $e = (u, v)$ from E with probability proportional to its weight
  $V' = V' \cup \{u, v\}$; $E' = E' \cup \{e\}$
  $V = V - \{u, v\}$; $E = E - \{e\}$
  for each $t \in V$ **do**
    if $e' = (u, t) \notin E$ or $e' = (v, t) \notin E$ **then**
      $V = V - \{t\}$
      remove edges linked to t from E: $E = E - \{e' = (t, *)\}$
    end if
  end for
end while
calculate average edge weight: $s(G') = \frac{\sum_{e \in E'} w(e)}{|E'|}$

Algorithm 2 Chunking by Maximal Clique (CMaxC)

Input:
$G = (V, E)$; $W(E) = \{w(e) | e \in E\}$
number of times to run Algorithm 1: k

Output:
$G'_{best} = (V'_{best}, E'_{best})$

$s_{max} = 0$
for $i = 1; i \leq k; i++$ **do**
  run Algorithm 1 with ¡$G'_i = (V'_i, E'_i)$,$s(G'_i)$¿ as output
  if $s(G'_i) > s_{max}$ **then**
    $G'_{best} = G'_i$; $s_{max} = s(G'_i)$
  end if
end for

Type Detection

- The preferred result of type detection
  - Considering traditional lexical features
    - Singleton Score (SS)
  - Semantically coherent
    - Affinity Score (AS)

$$S_{sg}(\bar{x}) = \begin{cases} 1+\theta & \bar{x}.r = pos(\bar{x}) \\ 1 & \text{otherwise} \end{cases}$$

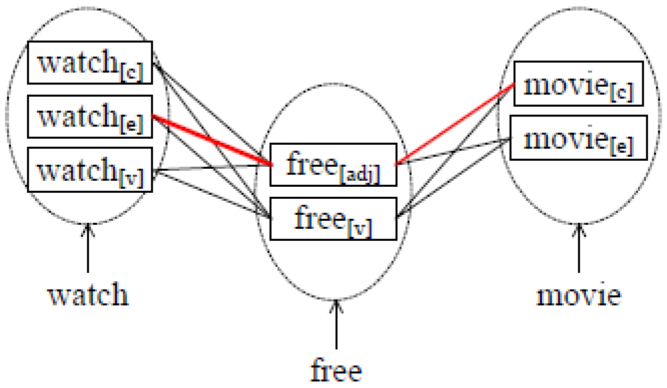
- Graph
  - Node -> typed term
  - Edge
    - Adjacent terms -> Chain Model (CM)
    - Cross-term -> Pairwise Model (PM)
    - Weight

$$w(\bar{x}, \bar{y}) = S_{sg}(\bar{x}) \cdot S(\bar{x}, \bar{y}) \cdot S_{sg}(\bar{y})$$

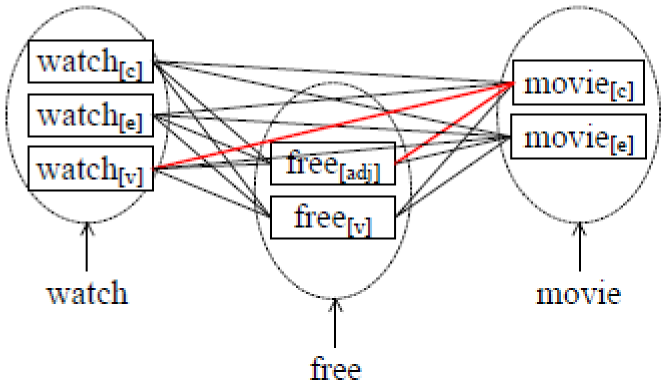
Type Detection (Cont.)

- Chain Model (CM)
  - Maximizes the total weight of the resulting sub-graph

- Pairwise Model (PM)
  - Maximum Spanning Tree (MST) of the resulting sub-graph has the largest weight.



(a) type detection result of "watch free movie using the *Chain Model* is $\{watch_{[e]}, free_{[adj]}, movie_{[c]}\}$.



(b) type detection result of "watch free movie using the *Pairwise Model* is $\{watch_{[v]}, free_{[adj]}, movie_{[c]}\}$.

Concept Labeling

- Appropriate concept clusters
  - Re-ranking concept clusters of the target instance based on context information in a short text
- Weighted-Vote approach
  - The most related term to help with disambiguation
    - Comparing weights of edges connecting to the target instance

$$\bar{x}.W_i' = V_{self}(C_i) \cdot V_{context}(C_i)$$

The original weight of concept cluster Ci

The weight of Ci in the most related term's co-occur concept cluster vector

Experiments

- Benchmark
- Data
  - Manually picked 11 terms that have ambiguity
  - Randomly selected 1100 queries (100 queries for each term)
  - 400 queries without any restriction
  - Removed 22 queries containing only one word
  - Altogether 1478 queries
- Evaluation
  - 5 disjoint parts
  - 15 colleagues to label them (3 for each part)

[27] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen. Short text conceptualization using a probabilistic knowledgebase. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2330–2336. AAAI Press, 2011.

[16] D. Kim, H. Wang, and A. Oh. Context-dependent conceptualization. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, pages 2654–2661. AAAI Press, 2013.

Experiments (Cont.)

- Effectiveness of Text Segmentation
  - Longest-Cover
  - MaxCBF (Maximal Clique by Brute Force)
  - MaxCMC (Maximal Clique by Monte Carlo)

TABLE II
ACCURACY OF TEXT SEGMENTATION.

	Longest-Cover	MaxCBF	MaxCMC
accuracy	0.954	**0.984**	0.979

Experiments (Cont.)

- Effectiveness of Type Detection
- Method
  - Stanford Tagger (ST)
  - Chain Model (CM)
  - Pairwise Model (PM)
- Level
  - Lexical (v, adj)
  - Semantic (attr, c, e)
  - Term
  - Query

TABLE III

ACCURACY OF TYPE DETECTION.

	ST	CM	PM
lexical-level	0.865	0.967	**0.978**
semantic-level	0.944	0.969	**0.973**
term-level	0.932	0.968	**0.974**
query-level	0.876	0.955	**0.967**

Experiments (Cont.)

- Effectiveness of Short Text Understanding
- Method
  - Song ([27])
  - Kim ([16])
  - This work
- Level
  - Term (whose top-1 concept cluster is correct)
  - Query (whose instances are all correct)

TABLE IV
ACCURACY OF SHORT TEXT UNDERSTANDING.

	Song	Kim	Our Approach
term-level	0.694	0.701	**0.943**
query-level	0.525	0.526	**0.890**

Experiments (Cont.)

- Efficiency of Short Text Understanding

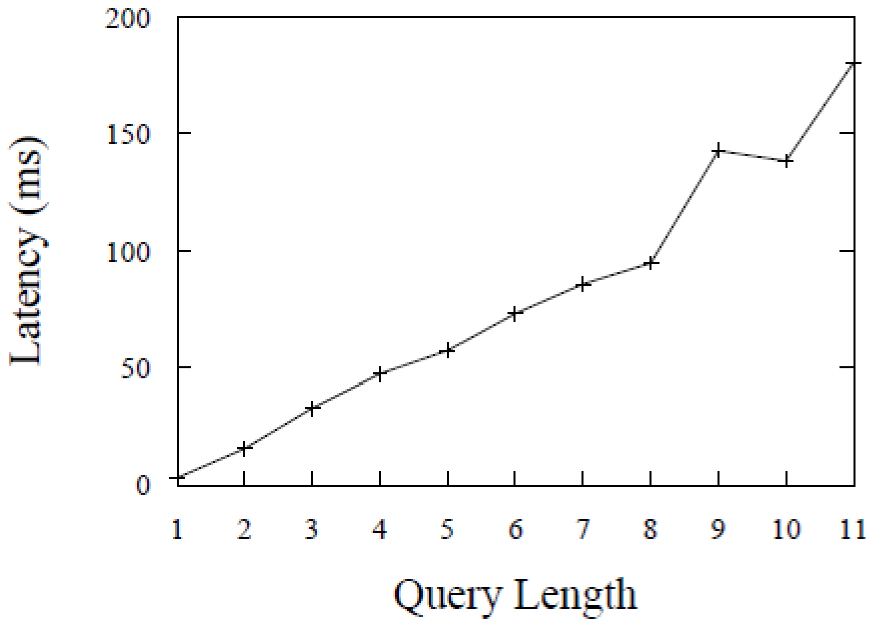


Fig. 8. Average time requirement of short text understanding when length (number of words) increases.

Conclusion

- Propose a generalized framework to understand short texts effectively and efficiently
- Three steps of short text understanding, namely text segmentation, type detection, and concept labeling are actually related with each other
    - A better framework for short text understanding should be one with feedbacks

Thanks!