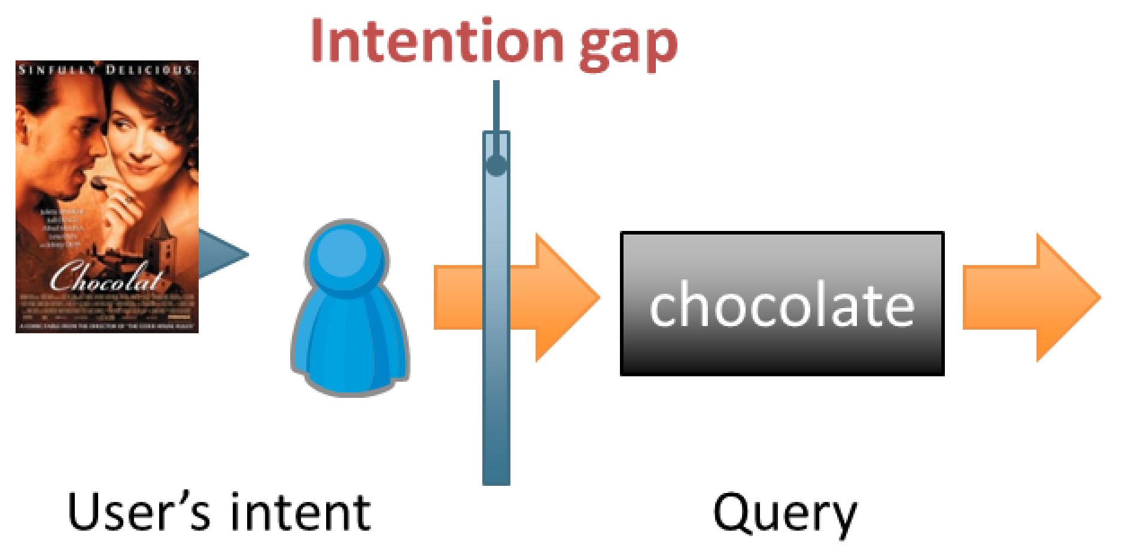
The KLE's Subtopic Mining System for the NTCIR-10 INTENT-2 Task

Se-Jong Kim and Jong-Hyeok Lee
Pohang University of Science and Technology (POSTECH), Korea

Introduction

Ambiguous/broad queries: Some users do not choose appropriate words for a web search, and others omit specific terms needed to clarify search intents, because it is not easy for users to express their search intents explicitly through keywords. This intention gap between users' search intents and queries results in queries which are ambiguous and broad.

Subtopic mining: A subtopic of a given query is a query that specifies and disambiguates the search intent of the original query. Subtopic mining returns a ranked list of subtopics in terms of the relevance to the query, popularity and diversity of subtopics.



rank	subtopic	score
1	chocolate recipe	0.25
2	valentine chocolate	0.22
3	white chocolate	0.15
4	movie chocolate	0.10

Subtopic mining

Subtopic Extraction

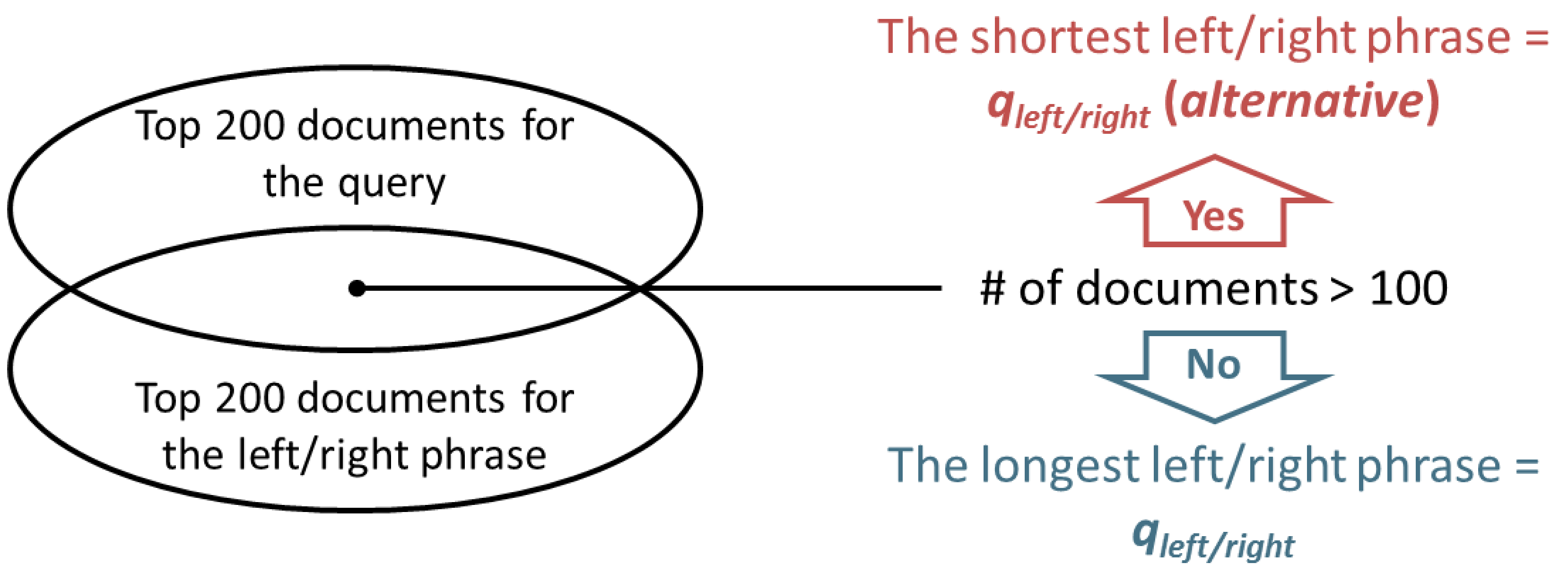
Step 1. Creation of simple patterns: We assumed that a subtopic consists of the original query and one or more noun phrases that specify the query. From this assumption, we created simple patterns to extract candidate strings. q_{left}/q_{right} was one of the left/right phrases of the original query. Each original query had only one q_{left} and one q_{right} which were alternative or not.

P1: ((adjective)$^?$(noun)$^+$(non-noun)*)$^?$**(query)**((non-noun)*(adjective)$^?$(noun)$^+$)$^?$

P2: ((adjective)$^?$(noun)$^+$(non-noun)*)$^?$$\underline{(q_{left})\text{(word)}^*(q_{right})}$((non-noun)*(adjective)$^?$(noun)$^+$)$^?$

P3: $\underline{(q_{right})}$(non-noun)*(adjective)$^?$(noun)$^+$

P4: (adjective)$^?$(noun)$^+$(non-noun)*$\underline{(q_{left})}$

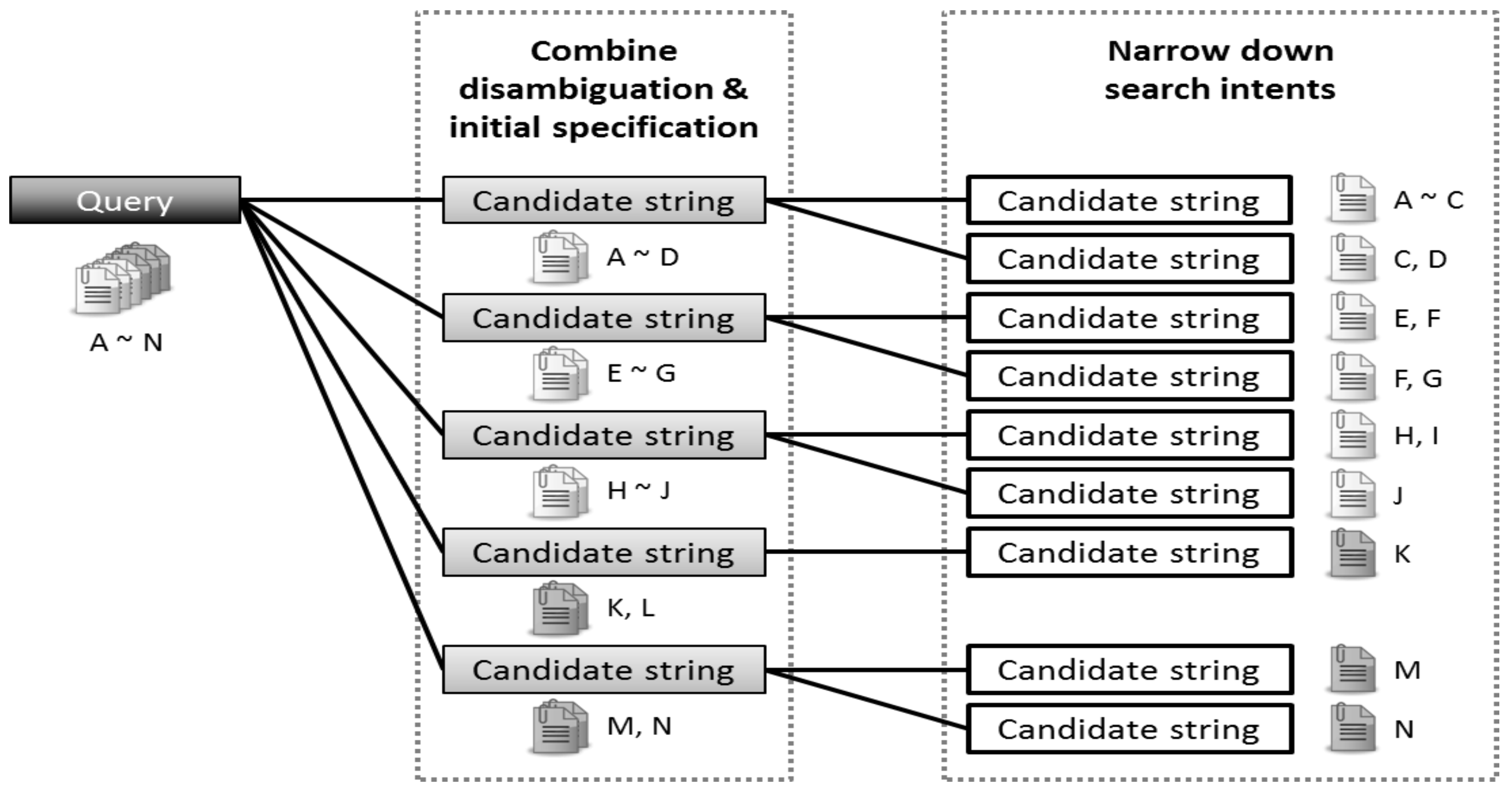

Top 200 documents for the query
Top 200 documents for the left/right phrase
The shortest left/right phrase = $q_{left/right}$ (alternative)
Yes
# of documents > 100
No
The longest left/right phrase = $q_{left/right}$

Step 2. Extraction of candidate strings: We generated several documents in which the *i*-th item of each official query suggestion appeared 11 - *i* times. We found phrases using the simple patterns from these and the top 1,000 relevant web documents for a given query. We replaced the parts of phrases corresponding to the underlined patterns with the original query.

Step 3. Filtering of candidate strings: s_{np} was a set of lemmas of noun phrases at the start or end of each candidate string. If s_{np}s of candidate strings were identical, we merged the frequency information of these candidate strings, and selected the most frequent and concise candidate string among these.

Subtopic Ranking

Step 1. Construction of the hierarchical structure of subtopics (*sts*): We used sets (clusters) of documents containing each candidate string and cluster measure.



$$CE(st,P) = -\sum_{st' \in ST, st' \neq st} \frac{|D(st,P) \cap D(st',P)|}{|D(st,P)|} \cdot log \frac{|D(st,P) \cap D(st',P)|}{|D(st,P)|}$$

P : the set of the top 200 relevant documents for the query, or documents containing the parent of *st*
ST : the set of unselected candidate strings that appear in at least two documents in *P*
D(*st*, *P*) : the set (cluster) of documents containing *st* in *P*

Step 2. Estimation of popularities of subtopics:

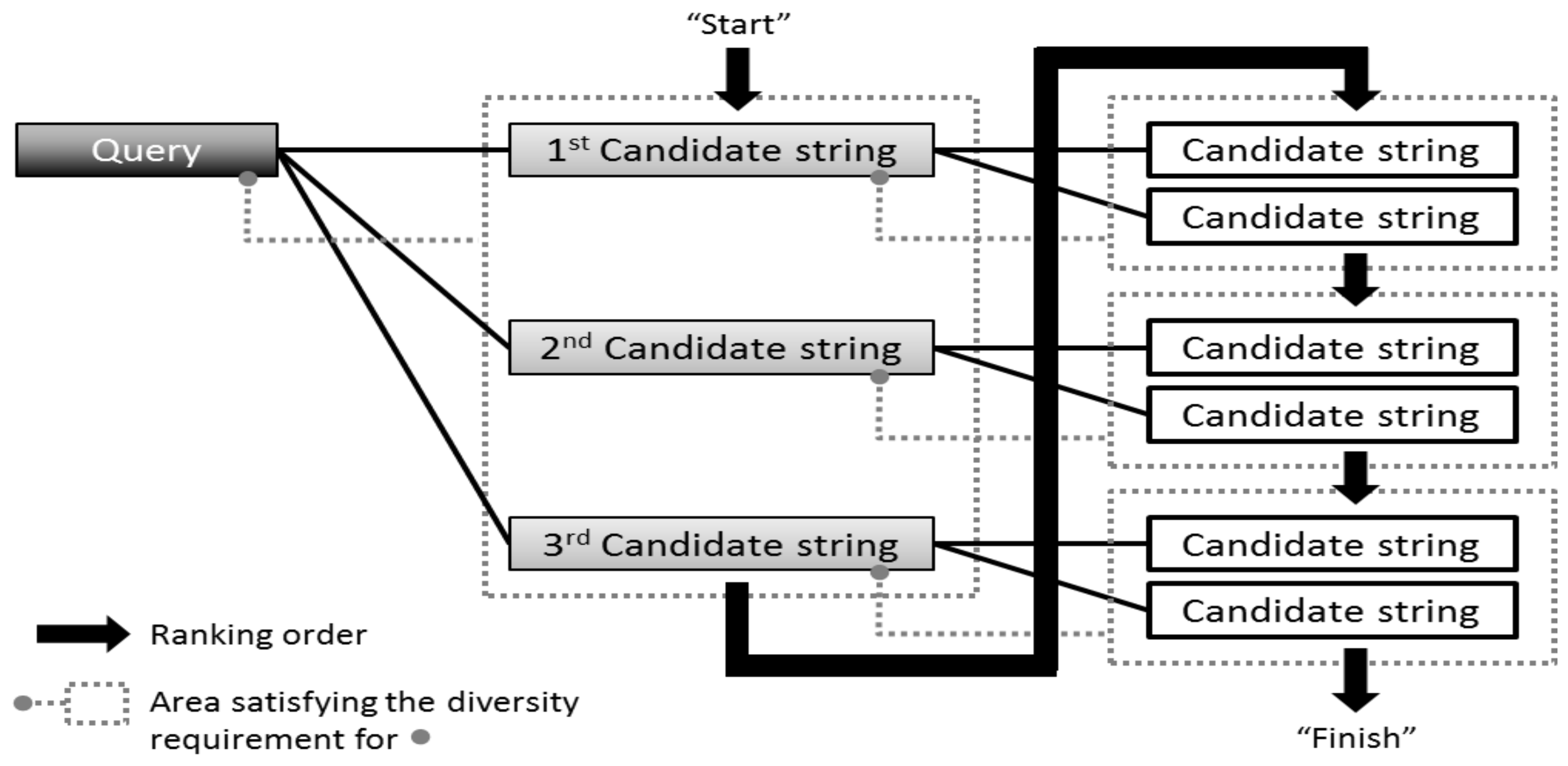
$$DC(st) = \sum_{doc \in (HR_{st} \cap HR_{query})} DocScore(doc)$$

$$CTFIDF(st) = freq(st, R_{query}) \cdot log \frac{|R_{query}|}{|D(st, R_{query})|}$$

$$Score(st) = \frac{DC(st)}{average\ of\ DCs} + \frac{CTFIDF(st)}{average\ of\ CTFIDFs}$$

HR_{st} : the set of the top 200 relevant documents for *st*
HR_{query} : the set of the top 200 relevant documents for the query
DocScore(*doc*) : the ranking score of *doc* for the query
R_{query} : the set of the top 1,000 relevant documents for the query
$freq(st,R_{query})$: the frequency of *st* in R_{query}

Step 3. Ranking of subtopics: We ranked candidate strings by popularities according to the ranking order.



Results

We used the given English(E)/Japanese(J) web document collection(doc) and the official query suggestions(qs).
(1: doc, *DC* / 2: doc, *Score* / 3: doc, qs, *DC* / 4: doc, qs, *Score*)

Run	Mean I-rec@10	Mean D-nDCG@10	Mean D#-nDCG@10
KLE-S-E-1A	0.3529	0.3540	0.3535
KLE-S-E-2A	0.4292	0.4159	0.4225
KLE-S-E-3A	0.3676	0.3661	0.3668
KLE-S-E-4A	**0.4457**	**0.4401**	**0.4429**
KLE-S-J-1B	**0.2607**	0.2656	**0.2632**
KLE-S-J-2B	0.2034	0.1667	0.1851
KLE-S-J-3B	0.2529	**0.2726**	0.2628
KLE-S-J-4B	0.2146	0.1687	0.1917