Université de Montréal

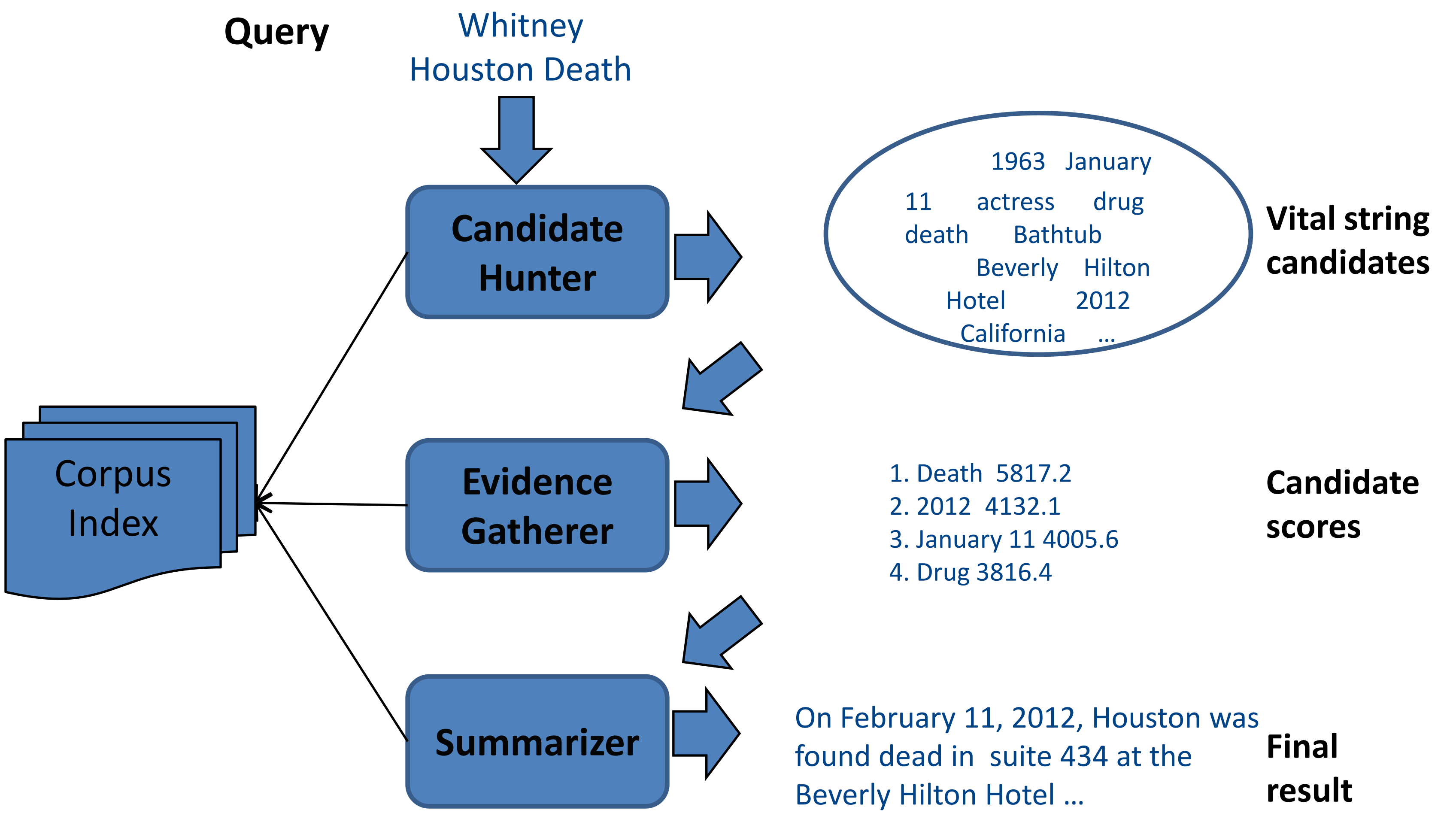
Hunter Gatherer: UDEM at 1CLICK-2

Pablo Duboue, Jing He, Jian-Yun Nie
DIRO, Université de Montréal, Canada

1CLICK-2

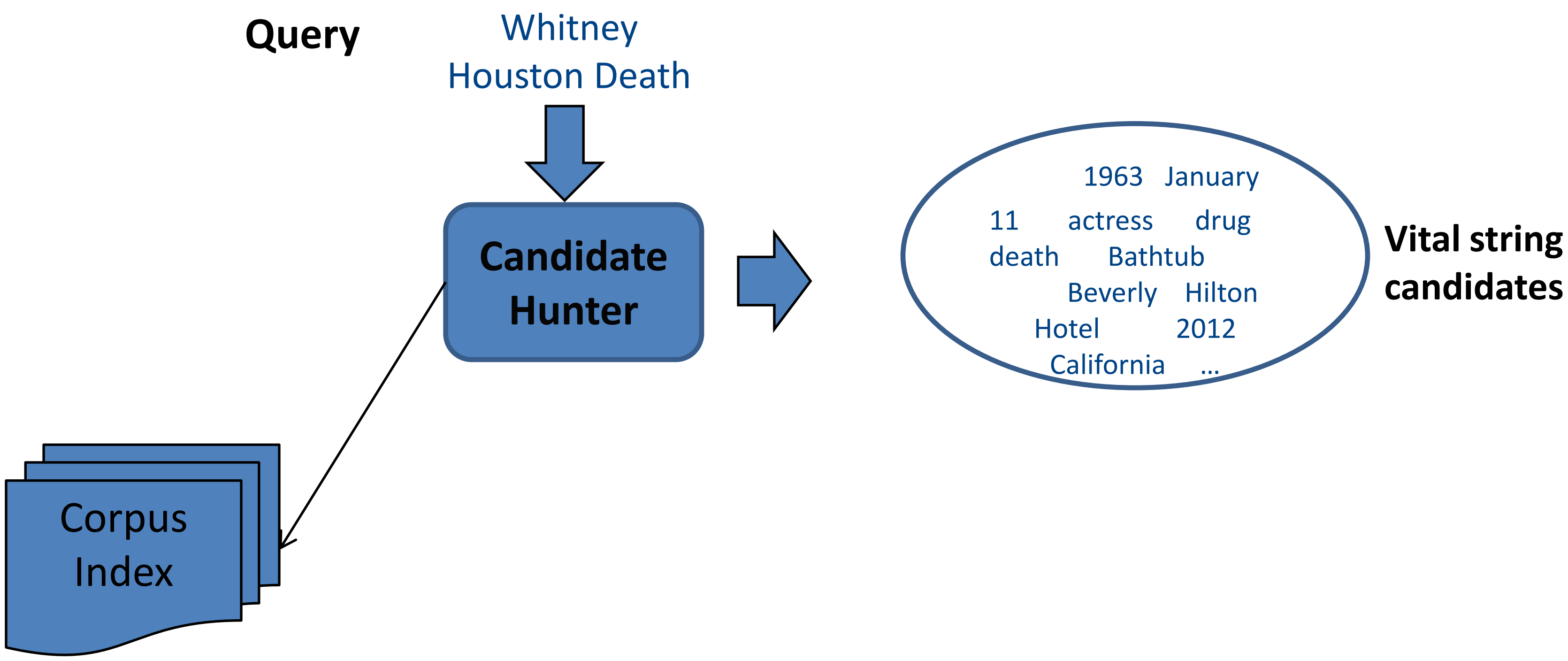
- Task
  - Return short text containing relevant information for queries
- Problems
  - Retrieving relevant information
  - Organizing relevant information

Our Framework



* The candidate hunter and gatherer is inspired by DeepQA framework

Candidate Hunter



Candidate Hunter

- Assumption
  - Relevant Information can be covered in top ranked passages
- Method
  - Passage Retrieval (Main Search)
  - Identifying Candidates

Candidate Hunter: Main Search

- Parse Query String
  - Named Entity Recognition

Whitney Houston death → "Whitney Houston" "death"

- Build Indri Query for Passage Retrieval

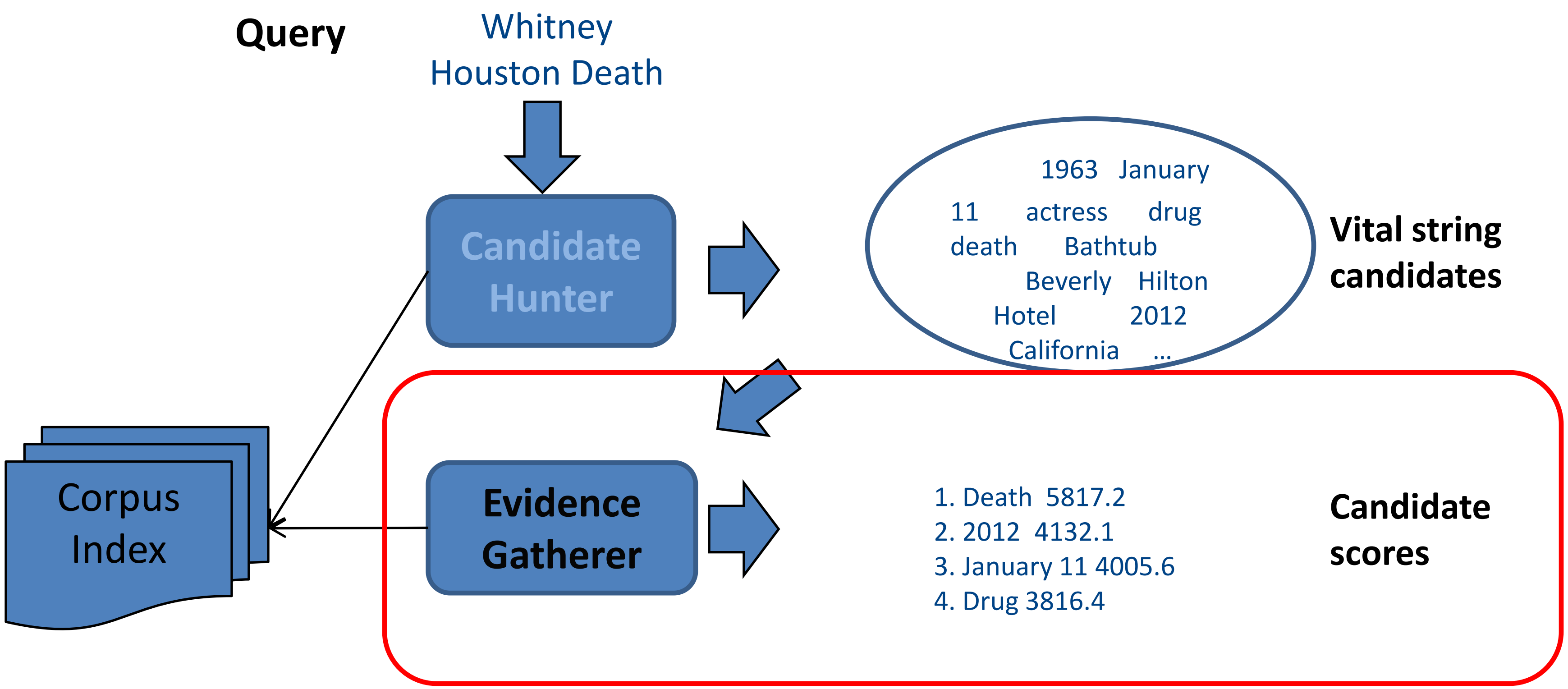
"Whitney Houston" "death" → #combine[passage120:50](#1(Whitney Houston) death)

- Retrieving Top K passages

Candidate Hunter: Identifying Candidates

- Selecting Candidates from Top K Passages
  - Terms
  - Named Entities
  - Pattern-based Candidates
    - Important attributes (birthday for a person, area for a country, etc.)
    - Information Extractor
      - Training Data: Wikipedia infobox – Wikipedia article
      - Model: CRF

Our Framework



Evidence Gatherer

- Assumption
  - More evidence about query + candidate → more relevant candidate
- Method
  - Passage retrieval to gather evidences
  - Combining evidences to estimate relevance

Evidence Gatherer: Evidence Search

- Building Evidence Gathering Query
  - Original Query + Candidate
  - Candidate Types

Text	Query	Phrase Type
A B	`#1(A B)`	named entity
A B	`#combine(0.5 #1(A B) 0.5 #combine(A B))`	pattern phrases

Whitney Houston death

Beverly Hilton Hotel

#combine[passage120:50](
#1(Whitney Houston)
#1(Beverly Hilton Hotel)
death)

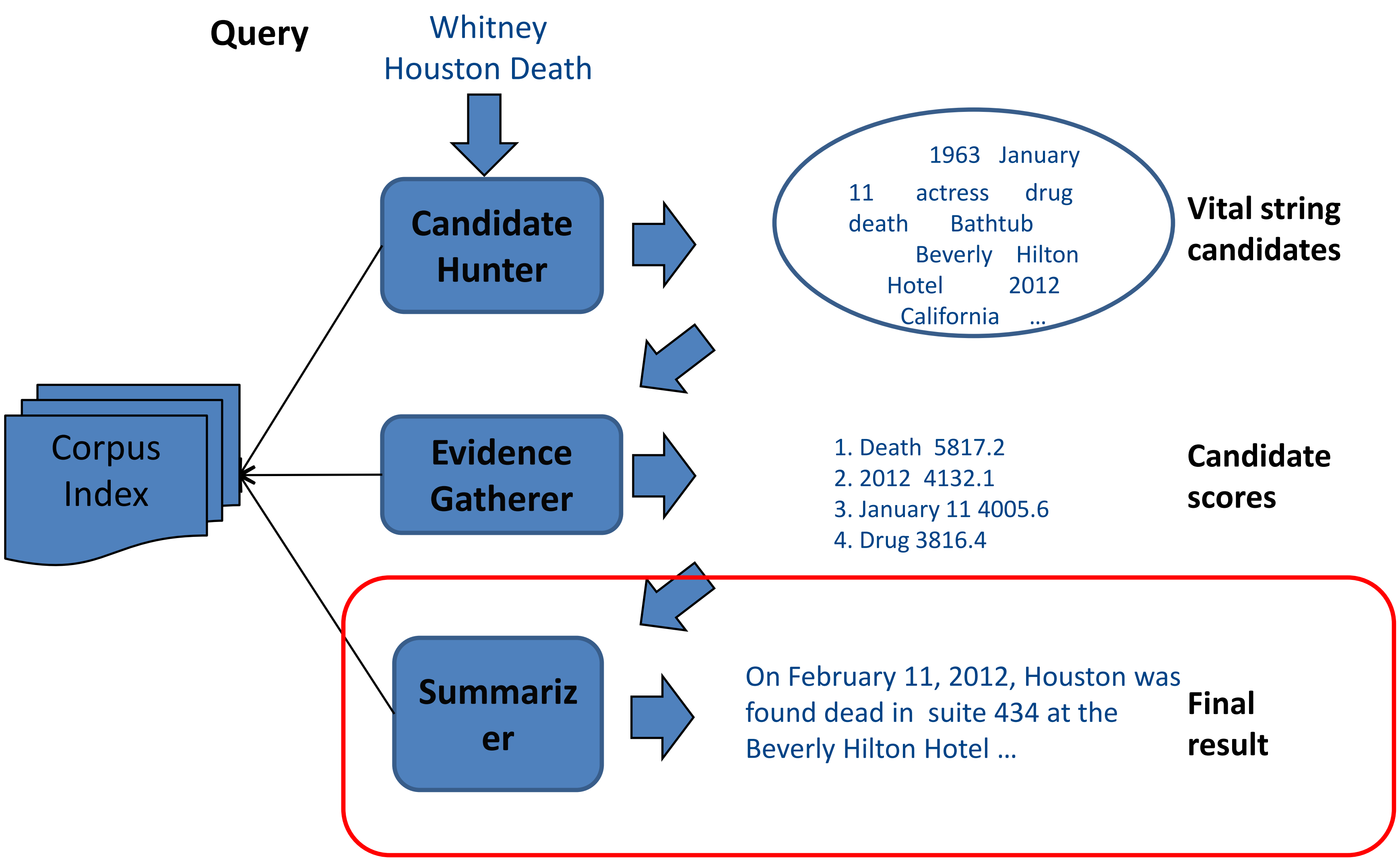
Evidence Gatherer: Evidence Search

- Heuristic Formula

$$R(q,u) = \lambda_1 \cdot \sum_{p \in MS, u \in p} \underbrace{(R(q,p) + \alpha)}_{\text{Main search (MS)}} + \lambda_2 \cdot \sum_{p \in ES} \underbrace{(R(q,p) + \beta)}_{\text{Evidence search (ES)}}$$

- Learn to Rank
  - Training Data: 60 Wikipedia articles
    - Query: title
    - Good Candidates: candidates from first passage of the Wikipedia article
    - Search Database: Clueweb09B
  - Model: GBDT

Summarizer: Organizing Relevant Information



Summarizer: Organizing Relevant Information

- Problem
  - Fit relevant information into limited length of text
- Method
  - Maximum Marginal Relevance (MMR)
  - Integer Linear Programming (ILP)

Summarizer: MMR

- Greedy algorithm to select sentences iteratively

$$s^* = \arg\max_s [\underbrace{\lambda \cdot R(Q,s)}_{\text{Relevance Function}} - (1-\lambda) \cdot \underbrace{\max_{s' \in S} sim(s,s')}_{\text{Redundancy Function}}]$$

- Relevance Function
  - Score sum of candidates in a sentences
- Redundancy Function
  - Jaccard similarity of bigrams

Summarizer: ILP

- Global Optimization Problem
  - Define
    - Sentence length (vector): *l*
    - Candidate score (vector): *w*
    - Candidates contained in sentences (matrix): M
    - Which sentences are selected (vector): *s*
    - Which candidates are selected (vector): *e*
  - Problem

$$\max e^{\mathrm{T}} w$$

$$\text{s.t.}\ 1)\ l^{\mathrm{T}} s \le k;$$

$$2)\ \mathbf{M} s \ge e$$

Submissions

	Hunter	Gatherer	Summarizer
Run 1.	Term, NE	Heuristics formula	MMR
Run 2.	Same as Run1 as a mobile run		
Run 3.	Term, NE, Pattern Info	Learnt scorer	MMR
Run 4.	Term, NE	Heuristics formula	ILP

Desktop Mandatory Results

RUN	Category								
	All	ACTOR	ATHLE	ARTIST	POLIT	FACIL	GEO	DEFIN	QA
Run 1	0.047	0.040	0.028	0.039	0.037	0.060	0.025	0.066	0.068
Run 3	0.050	0.058	0.016	0.038	0.086	0.058	0.016	0.077	0.053
Run 4	0.080	0.068	0.084	0.074	0.025	0.079	0.062	0.076	0.146
MAX	0.080	0.068	0.084	0.074	0.086	0.083	0.080	0.088	0.146
MIN	0.047	0.040	0.016	0.018	0.025	0.005	0.016	0.055	0.053
AVRG	0.059	0.053	0.034	0.032	0.049	0.070	0.044	0.067	0.096
MEDIAN	0.055	0.053	0.028	0.027	0.039	0.076	0.035	0.066	0.089

- Better performance of Run 4 (with ILP)
  - → It's important to organize relevant information intelligently

* MAX, MIN, AVRG, MEDIAN for all Desktop Mandatory Results

Desktop Mandatory Results

RUN	Category								
	All	ACTOR	ATHLE	ARTIST	POLIT	FACIL	GEO	DEFIN	QA
Run 1	0.047	0.040	0.028	0.039	0.037	0.060	0.025	0.066	0.068
Run 3	0.050	0.058	0.016	0.038	0.086	0.058	0.016	0.077	0.053
Run 4	0.080	0.068	0.084	0.074	0.025	0.079	0.062	0.076	0.146
MAX	0.080	0.068	0.084	0.074	0.086	0.083	0.080	0.088	0.146
MIN	0.047	0.040	0.016	0.018	0.025	0.005	0.016	0.055	0.053
AVRG	0.059	0.053	0.034	0.032	0.049	0.070	0.044	0.067	0.096
MEDIAN	0.055	0.053	0.028	0.027	0.039	0.076	0.035	0.066	0.089

- It performs well for QA queries
  - → Naturally, the DeepQA framework helps QA queries

Desktop Mandatory Results

RUN	Category								
	All	ACTOR	ATHLE	ARTIST	POLIT	FACIL	GEO	DEFIN	QA
Run 1	0.047	0.040	0.028	0.039	0.037	0.060	0.025	0.066	0.068
Run 3	0.050	0.058	0.016	0.038	0.086	0.058	0.016	0.077	0.053
Run 4	0.080	0.068	0.084	0.074	0.025	0.079	0.062	0.076	0.146
MAX	0.080	0.068	0.084	0.074	0.086	0.083	0.080	0.088	0.146
MIN	0.047	0.040	0.016	0.018	0.025	0.005	0.016	0.055	0.053
AVRG	0.059	0.053	0.034	0.032	0.049	0.070	0.044	0.067	0.096
MEDIAN	0.055	0.053	0.028	0.027	0.039	0.076	0.035	0.066	0.089

- Minor improvement from Run 3 compared to Run 1
  - Some improvement is from person type queries

Existing Problems and Future Works

- Spam
  →Filtering methods
- Candidate Selection
  →Unsupervised parsing for chunk detection
- Summarization Granularity
  - Sentences are too long
    → Sentence compression
  - Only part of sentence is relevant
    → Breaking multi-clause sentences by text simplification