Query-Chain Focused Summarization

1. Introduction

Purpose

- This paper relates to text summarization.
  - Existing task:
    (i) Generic Multi Document Summarization
    (ii) Update Summarization
    (iii) Query-Focused Summarization etc.

⇩ In contrast...

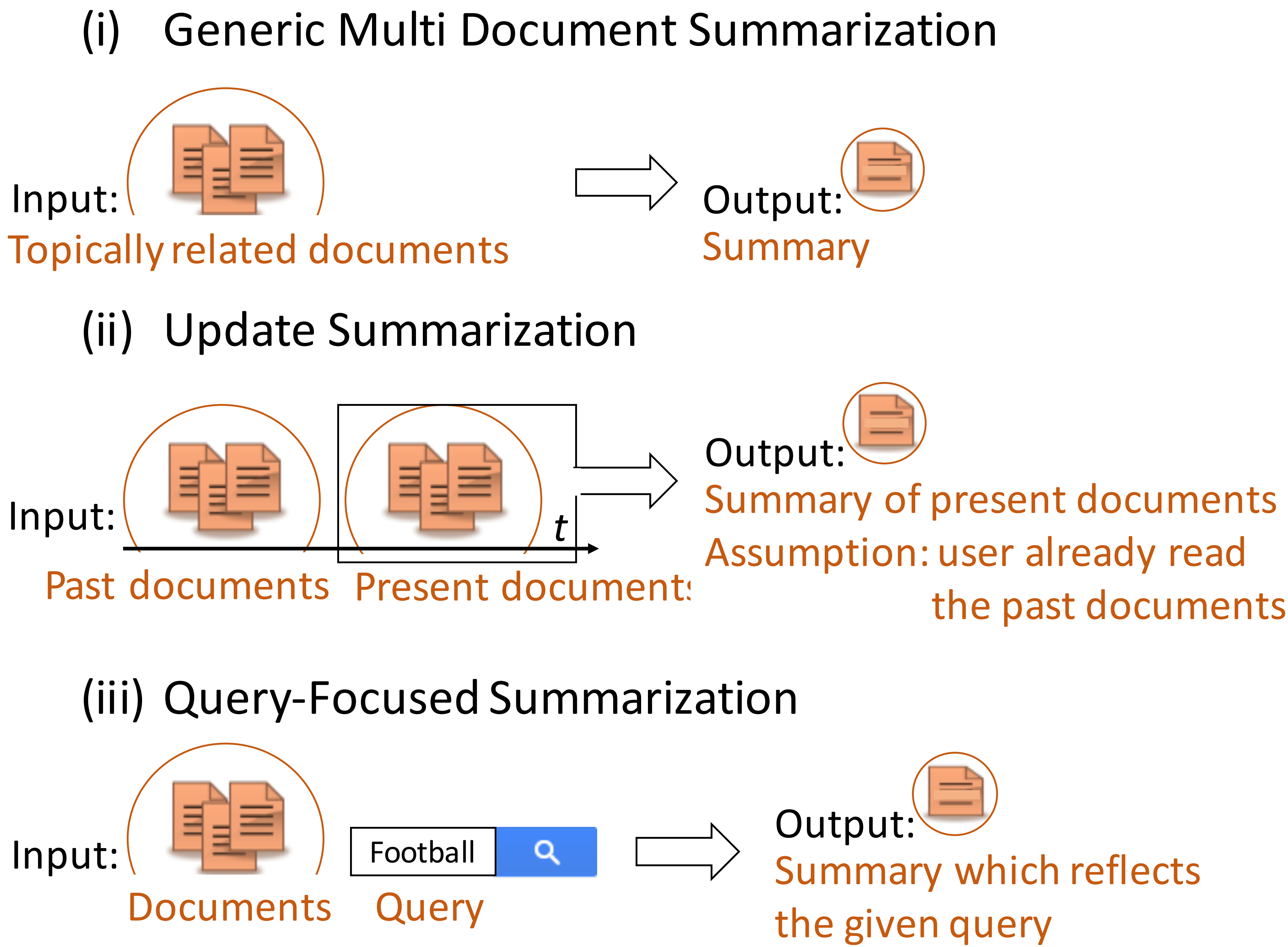
- This paper proposes new task, Query-Chain Focused Summarization to improve exploratory search

Purpose

- This paper focuses on text summarization.
    - Existing task:
      (i) Generic Multi Document Summarization
      (ii) Update Summarization
      (iii) Query-Focused Summarization etc.

⇩

- This paper proposes new task, Query-Chain Focused Summarization to improve exploratory search


(i) Generic Multi Document Summarization
Input:
Topically related documents
Output:
Summary
(ii) Update Summarization
Input:
t
Past documents
Present documents
Output:
Summary of present documents
Assumption: user already read
the past documents
(iii) Query-Focused Summarization
Input:
Football
Documents
Query
Output:
Summary which reflects
the given query

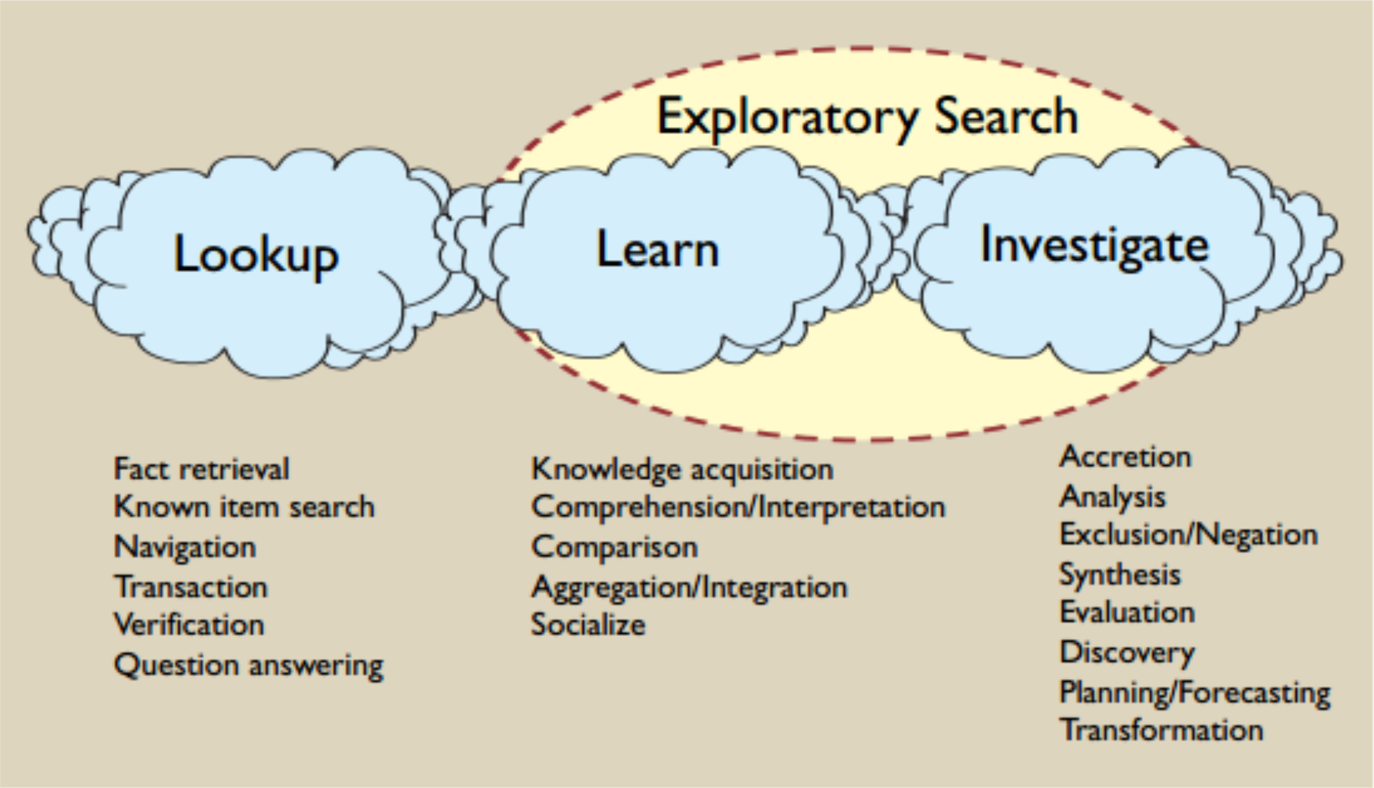
Purpose

- This paper focuses on text summarization.
  - Existing task:
    (i) Generic Multi Document Summarization
    (ii) Update Summarization
    (iii) Query-Focused Summarization etc.

- This paper proposes new task, Query-Chain Focused Summarization to improve exploratory search

Preliminary
— Exploratory Search —

- Search to learn or investigate a new topic



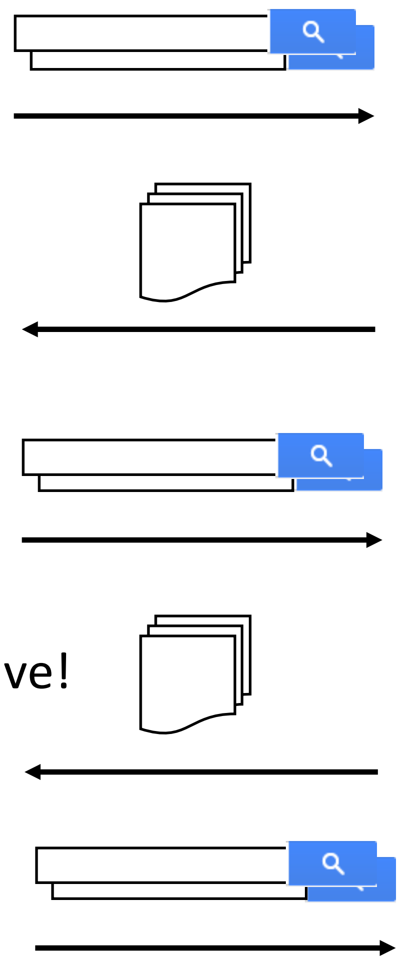
Preliminary
— Exploratory Search: example —

I want to go on a trip to France.
put up at a resort villa
in the countryside.

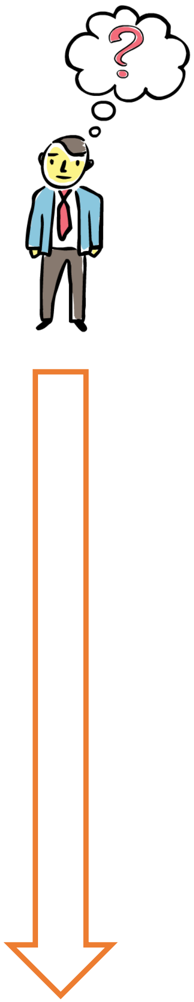
There are several resort villa
in Provence....

Provence is nice!
I plan to put up at a resort villa
in Provence!

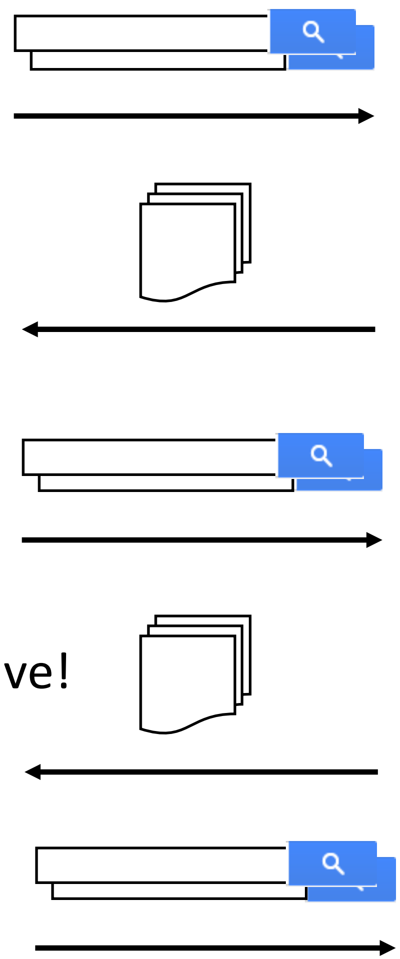
No!! The hotel bill is too expensive!
Search another town...



Preliminary
― Exploratory Search: example ―



I want to go on a trip to France.
put up at a resort villa
in the countryside.



There are several resort villa
in Provence....

Provence is nice!
I plan to put up at a resort villa
in Provence!

zoom in

No!! The hotel bill is too expensive!
Search another town...

zoom out

Purpose

- This paper focuses on text summarization.
  - Existing task:

    (i) Generic Multi Document Summarization

    (ii) Update Summarization

    (iii) Query-Focused Summarization etc.

⇩

- This paper proposes new task, Query-Chain Focused Summarization to improve exploratory search

Query-Chain Focused Summarization: Definition

- Define:

For each query in an exploratory search session,
we aim to extract a summary
that answers the information need of the user,
while not repeating information
already provided in previous steps.

- ____ is similar to Query-Focused Summarization.
- ____ is siimilar to Update Summarization.

Contribution

- The definition of a new summarization task: QCFS
- Construct novel Dataset for QCFS:
  - Novel dataset of Query-Sets with matching manual summarizations in the consumer health domain
- Adapt well-known multi-document algorithms to the task
- Propose a new algorithms to address the task of QCFS, based on a new LDA topic model variant

2. Query-Chain Focused Summarization

Query-Chain Focused Summarization: Formalization

- Formalization:

Given:

ordered chain of queries Q

a set of documents D

For each query $q_i \in Q$, a summary S_i is generated from D answering q_i

under the assumption that the user has already read the summaries $S_{0:i-1}$ for queries $q_{0:i-1}$

- This paper focuses on the *zoom in* aspect of the exploratory search process.

Query-Chain Focused Summarization: example

- A typical example of query chain in the consumer health domain:
  - query chain:

    causes of asthma
    → *asthma and allergy*
    → *asthma and mold allergy*

  - Reference set D:

    documents relevant to the domain of Asthma

  - Task:
    - Generating one summary of D as an answer to each query
    - The successive answers do not repeat information already provided in a previous answer

4. Dataset Collection

Dataset Collection

- Build in Consumer Health domain
  - providing medical information ranging from layman and up to expert information
- Dataset is composed of
  - Query Chains
    - Manually selected from PubMed query logs
  - Document Set
    - Manually selected from various sites to contain relevant information about the queries
  - Manual Summaries
    - Created for each query some were created within the context of the query chain and some weren’t

Dataset Collection

- Build in Consumer Health domain
  - providing medical information ranging from layman and up to expert information
- Dataset is composed of
  - Query Chains
    - Manually selected from PubMed query logs
  - Document Set
    - Manually selected from various sites to contain relevant information about the queries
  - Manual Summaries
    - Created for each query some were created within the context of the query chain and some weren't

Dataset Collections
—Query Chains—

- Using PubMed query logs
- Extract laymen queries relating to four topics:

1. Asthma	3. Obesity
2. Lung Cancer	4. Alzheimer's disease

<u>Procedure</u>

1. Extract a single day query log
2. Extract sessions which contained the terms from the query log
3. Sessions containing search tags (such as "[Author]") were removed
4. The sessions were then manually examined

Dataset Collection
—Example of Query Chain—

- Only *zoom in* query chains of length 3 at most

Asthma:

Asthma causes→ asthma allergy→ asthma mold allergy;

Asthma treatment→asthma medication→corticosteroids;

Exercise induced asthma→ exercise for asthmatic;

Atopic dermatitis→ atopic dermatitis medications→ atopic dermatitis side effects;

Atopic dermatitis→ atopic dermatitis children→ atopic dermatitis treatment;

Atopic dermatitis → atopic dermatitis exercise activity → atopic dermatitis treatment;

Figure 1: Queries Used to Construct Dataset

Dataset Collection

- Build in Consumer Health domain
  - providing medical information ranging from layman and up to expert information
- Dataset is composed of
  - Query Chains
    - Manually selected from PubMed query logs
  - Document Set
    - Manually selected from various sites to contain relevant information about the queries
  - Manual Summaries
    - Created for each query some were created within the context of the query chain and some weren’t

Dataset Collections —Document Set—

- Using Wikipedia, WebMD, NHS
  - well-known and reliable consumer health websites

- Asked medial experts to construct four document collections about the four topics
  - Each document provide general information relevant to the queries

Dataset Collection

- Build in Consumer Health domain
  - providing medical information ranging from layman and up to expert information
- Dataset is composed of
  - Query Chains
    - Manually selected from PubMed query logs
  - Document Set
    - Manually selected from various sites to contain relevant information about the queries
  - Manual Summaries
    - Created for each query some were created within the context of the query chain and some weren't

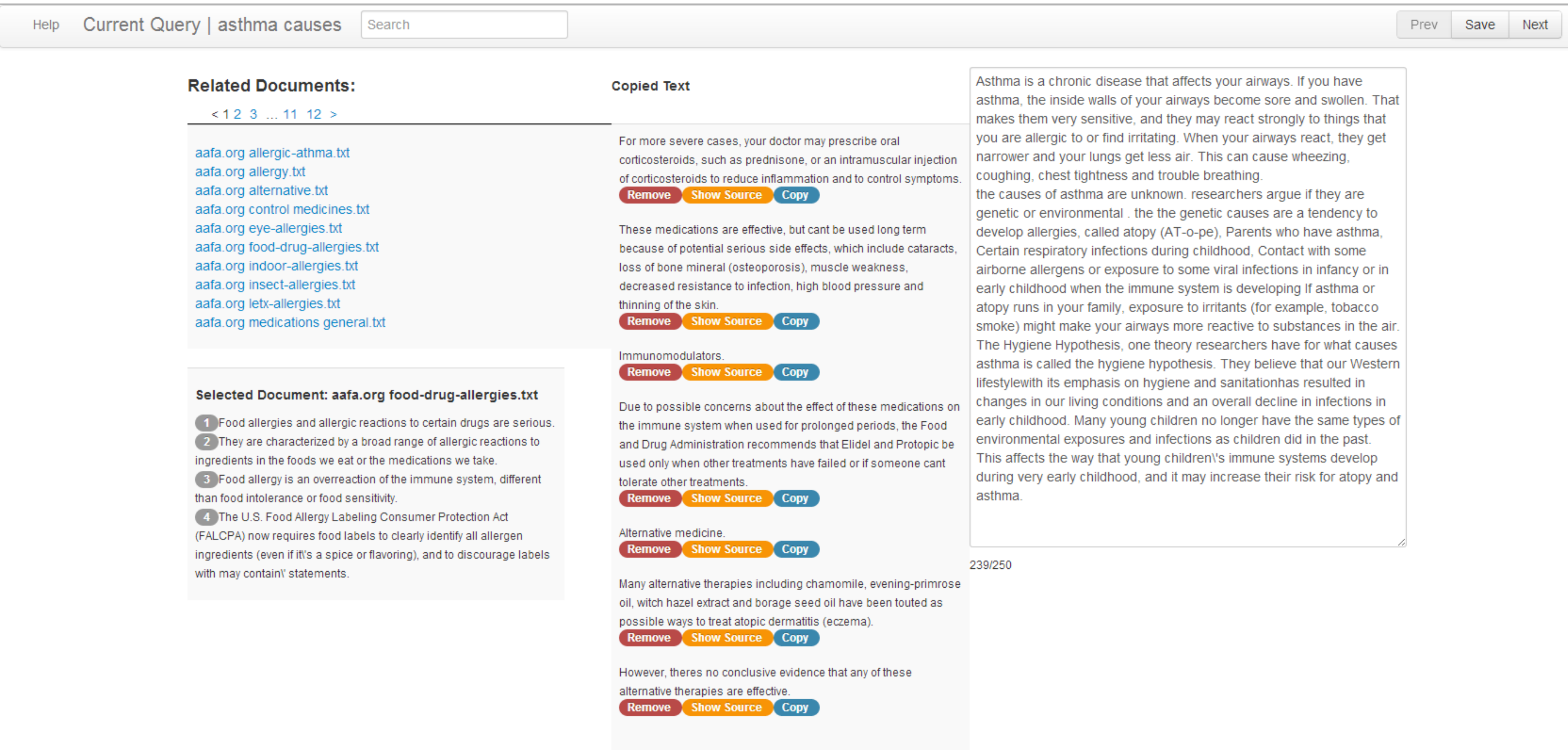
Dataset Collections
ー Manual Summaries ー

- Asked medical students to manually produce summaries of these four document collections for each query-chain

<u>Instruction</u>

1. Construct a text of up to 250 words that provides a good answer to each query in the chain
2. Assume that the person reading the summaries is familiar with the previous summaries in the chain
   ー avoid redundancy
3. Not shown the next steps in the chain

Interface for annotators



Statistics on the collected dataset

Document sets	# Docs	# Sentences	#Tokens / Unique
Asthma	125	1,924	19,662 / 2,284
Lung-Cancer	135	1,450	17,842 / 2,228
Obesity	289	1,615	21,561 / 2,907
Alzheimer's Disease	191	1,163	14,813 / 2,508

Queries	# Sessions	# Sentences	#Tokens / Unique
Asthma	5	15	36 / 14
Lung-Cancer	6	18	71 / 25
Obesity	6	17	45 / 29
Alzheimer's Disease	4	12	33 / 16

Manual Summaries	# Docs	# Sentences	#Tokens / Unique
Asthma	45	543	6,349 / 1,011
Lung-Cancer	54	669	8,287 / 1,130
Obesity	51	538	7,079 / 1,270
Alzheimer's Disease	36	385	5,031 / 966

Verifying the dataset

- The summaries for advanced (second or third) query should contain updated information

Verify:

- Asked additional annotators to create manual summaries for advanced queries without ever seeing the previous queries

- Compare the mean ROUGE score
  [With-Context]:
  each manual summary
  vs. all other summaries about the same query
  [Without-Context]:
  The mean ROUGE scores of the additionally created summaries

Verifying the dataset
ーROUGE scoreー

[With-Context]
ROUGE-1 = 0.52, ROUGE-2 = 0.22, ROUGE-SU4 = 0.13

[Without-Context]
ROUGE-1 = 0.40, ROUGE-2 = 0.22, ROUGE-SU$ = 0.01

Except for the ROUGE-2, results showed statistically significant difference with 95% confidence interval

5. Algorithms

Algorithms

- Baselines (adapted existing methods to QCFS)
    - Focused KLSum
    - KL-Chain-Update
    - ChainSum

- New Algorithm for QCFS
    - Adapted LexRank

Algorithms

- Baselines (adapted existing methods to QCFS)
  - Focused KLSum
  - KL-Chain-Update
  - ChainSum
- New Algorithm for QCFS
  - Adapted LexRank

Focused KLSum ー Overview ー

- A variation of KLSum which is adapted to Query-Focused Summarization
  - Originally, KLSum is a method for generic multi-document summarization
  - KLSum tries to minimize the KL-divergence between the summary and document set unigram distribution
- Focused KLSum used KLSum on the 10 documents with best TF/IDF matches to the query

Focused KLSum
—Algorithm—

Given a query q,

1. Select a focused subset of documents from D: $D(q)$
2. Search $S^* = argmin_{|S|<L} KL(P_{D(q)}||P_S)$,
   where $KL(P||Q) = \sum_w \log\left(\frac{P(w)}{Q(w)}\right) P(w)$

   — This search is performed by greedy manner

   — $D(q)$ is determined by selecting the top-10 documents in D ranked by $TF \times IDF$ scores to the query

<u>Notation</u>

S: candidate summary

$P_{D(q)}$: unigram distribution of $D(q)$

P_S: unigram distribution of candidate summary

L: maximum length of summary

Algorithms

- Baselines (adapted existing methods to QCFS)
  - Focused KLSum
  - KL-Chain-Update
  - ChainSum

- New Algorithm for QCFS
  - Adapted LexRank

KL-Chain-Updates —Overview—

- A variation of KLSum that answers a query chain $(q_1, q_2, \dots, q_i, q_{i+1})$
- Try to minimize the KL-divergence of the summary and the top 10 TF/IDF retrieved documents for query q_{i+1}
- Select sentences for q_{i+1} assuming the smoothed distribution of the previous summaries $(q_1, \dots, q_i)$ is already part of the summary (eliminates redundancy)

KL-Chain-Updates —unigram distribution—

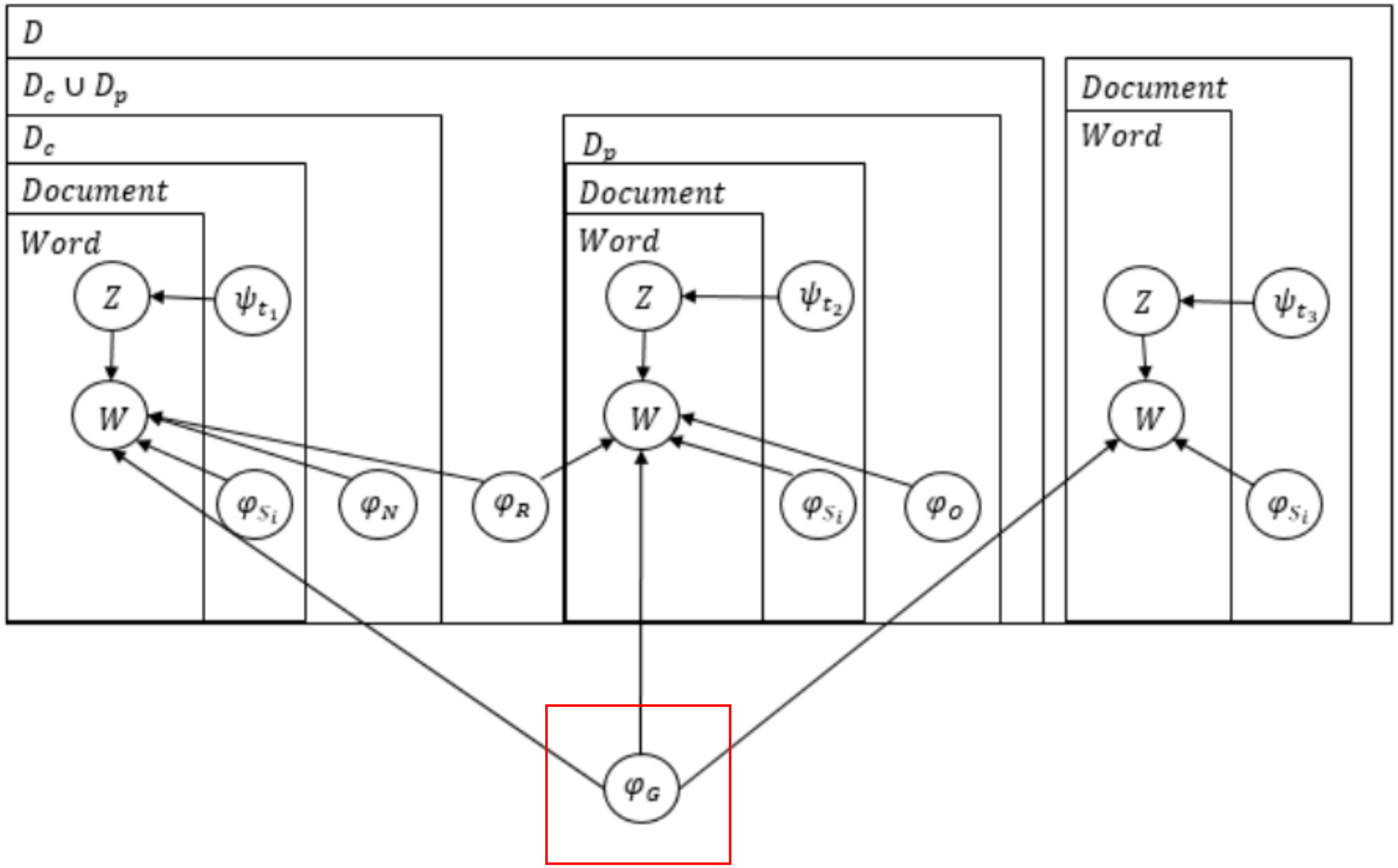
Unigram Distribution of Word w	
KLSum	KL-Chain-Updates
$\frac{Count(w, CurrentSum)}{Length(CurrentSum)}$	$\frac{Count(w, CurrentSum) + \frac{Count(w, PreviousSum)}{Smoothing\ Factor}}{Length(CurrentSum) + \frac{Length(PreviousSum \cap CurrentSum)}{Smoothing\ Factor}}$

Algorithms

- Baselines (adapted existing methods to QCFS)
  - Focused KLSum
  - KL-Chain-Update
  - ChainSum
- New Algorithm for QCFS
  - Adapted LexRank

ChainSum
—Overview—

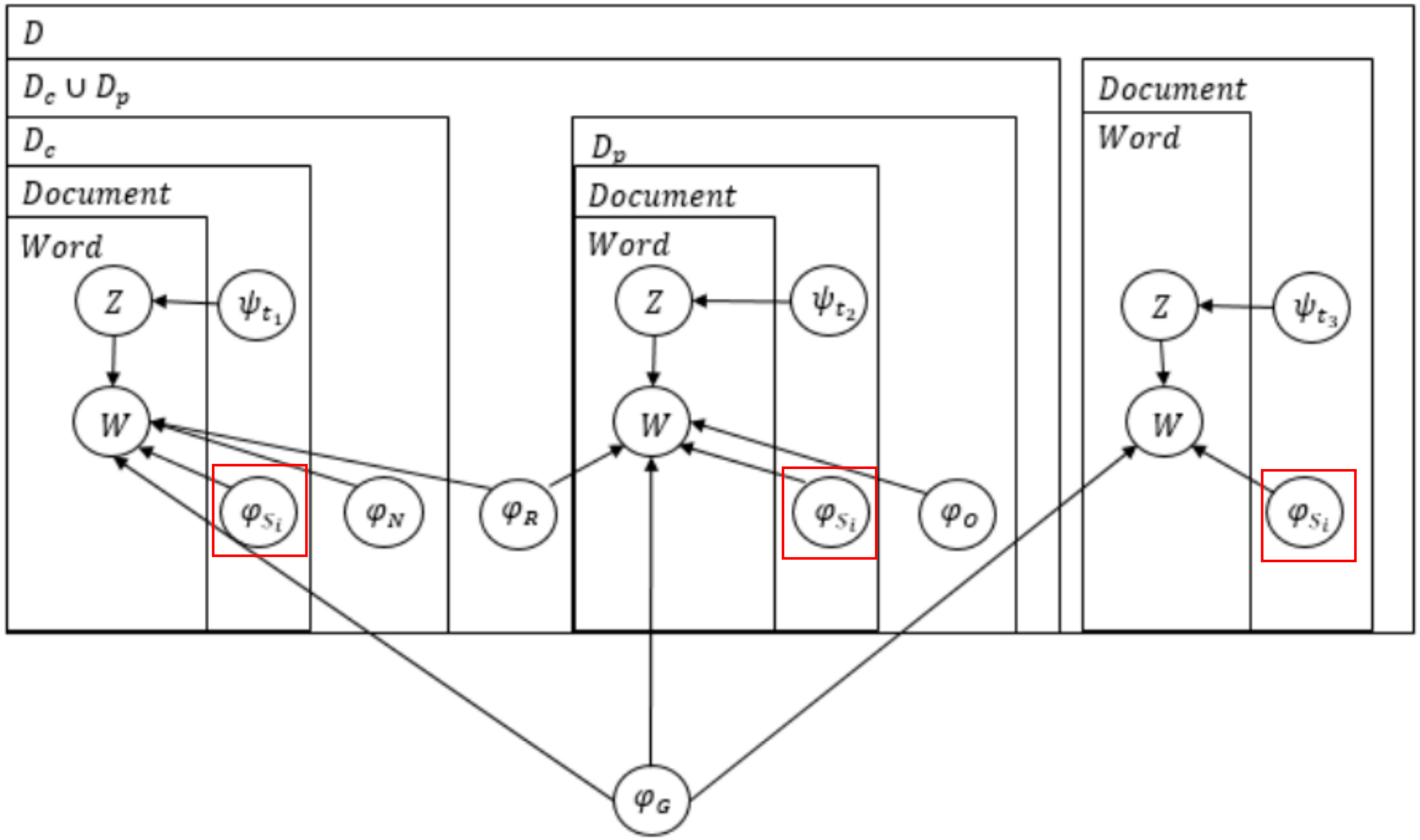
- Adaptation of TopicSum to the QCFS task
- Develop a novel topic model
  - to identify words that are associated to the current query and not shared with the previous queries
- For each query in a chain
  - Consider the documents D_c: “good answers” to the query
  - Also consider the documents D_P: documents used to answer the previous steps of the chain
- Assumption:
  - D_c and D_P are observable
  - But in their implementation, they select these subsets by ranking the documents for the query based on $TF \times IDF$ similarity



1. G is the *general words* topic,

—capture stop words, and non-topic specific vocabulary

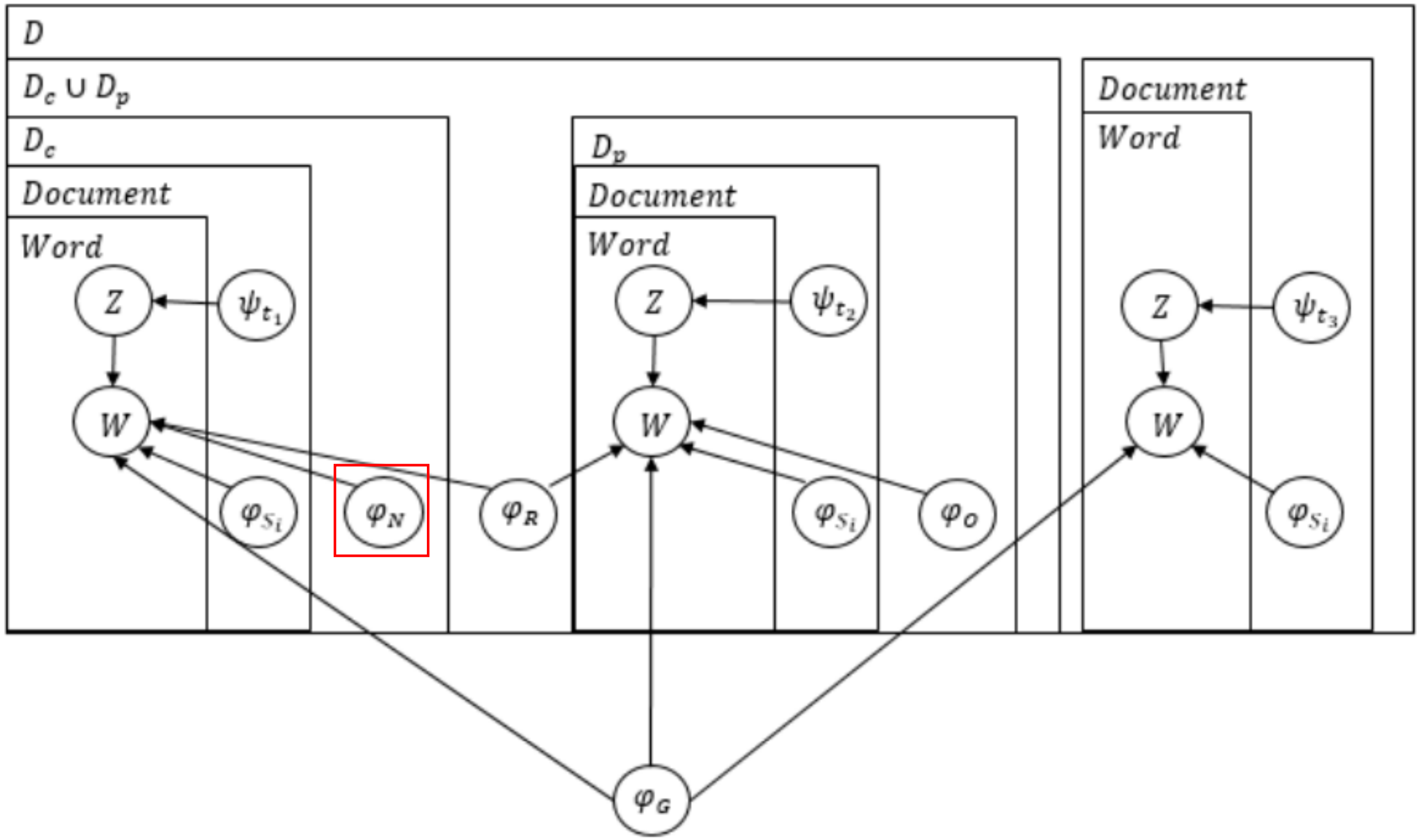
φ_G is drawn for all the documents from $Dirichlet(V, \lambda_G)$



2. S_i is the *document specific* topic,

—represents words, which are local for a specific document

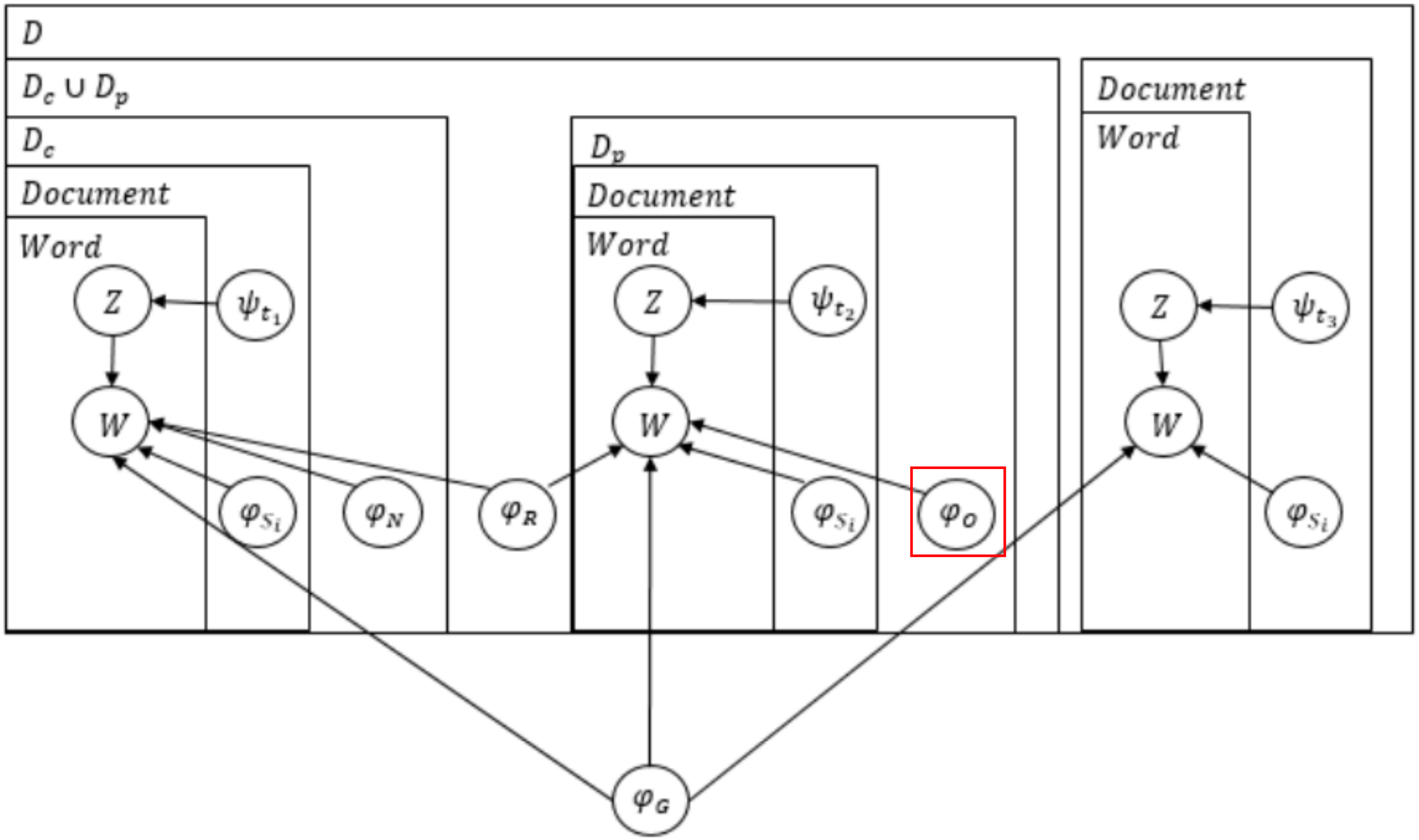
φ_{S_i} is drawn for each document from $Dirichlet(V, \lambda_{S_i})$



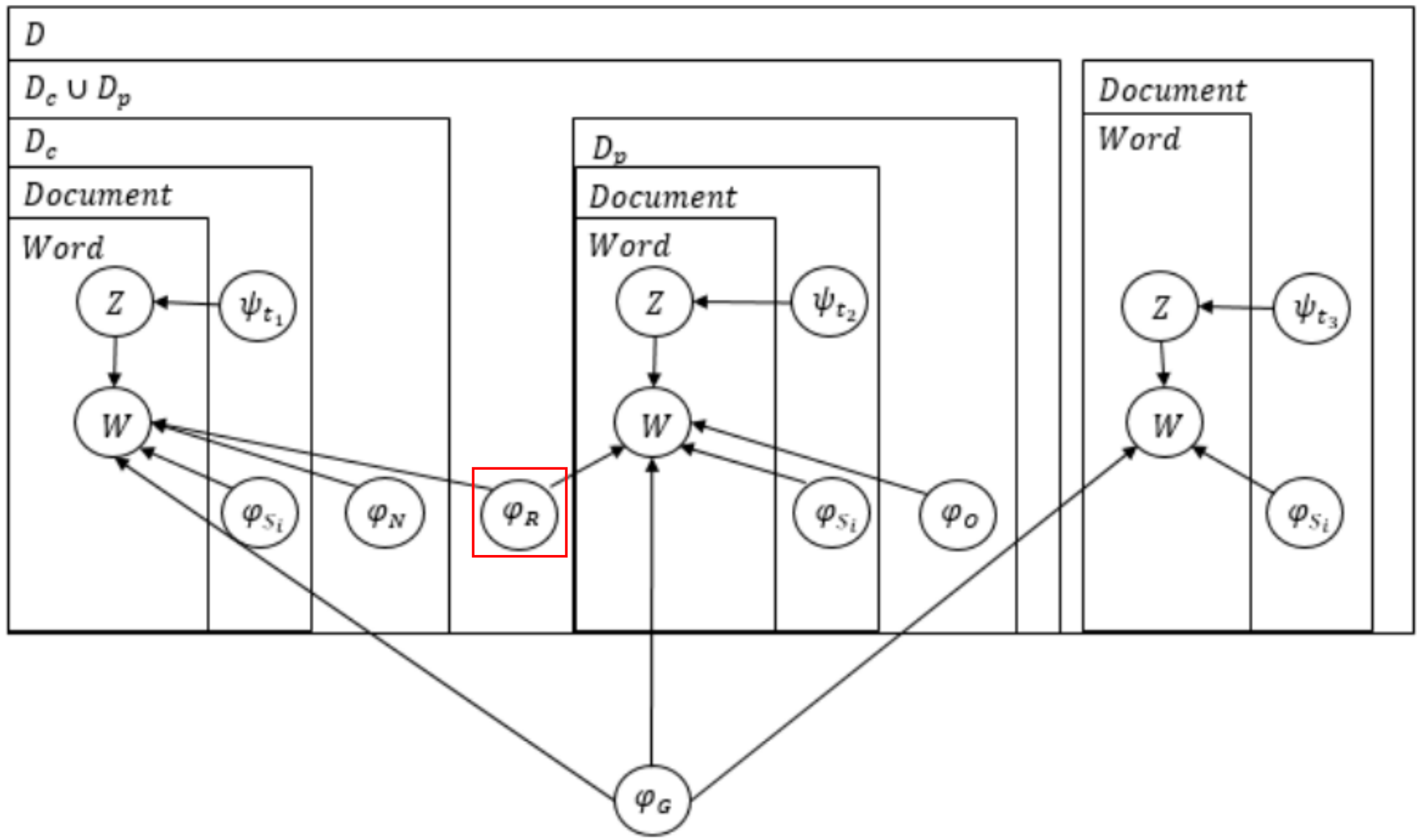
3. N is the *new content* topic,

—capture words that are characteristic for D_c

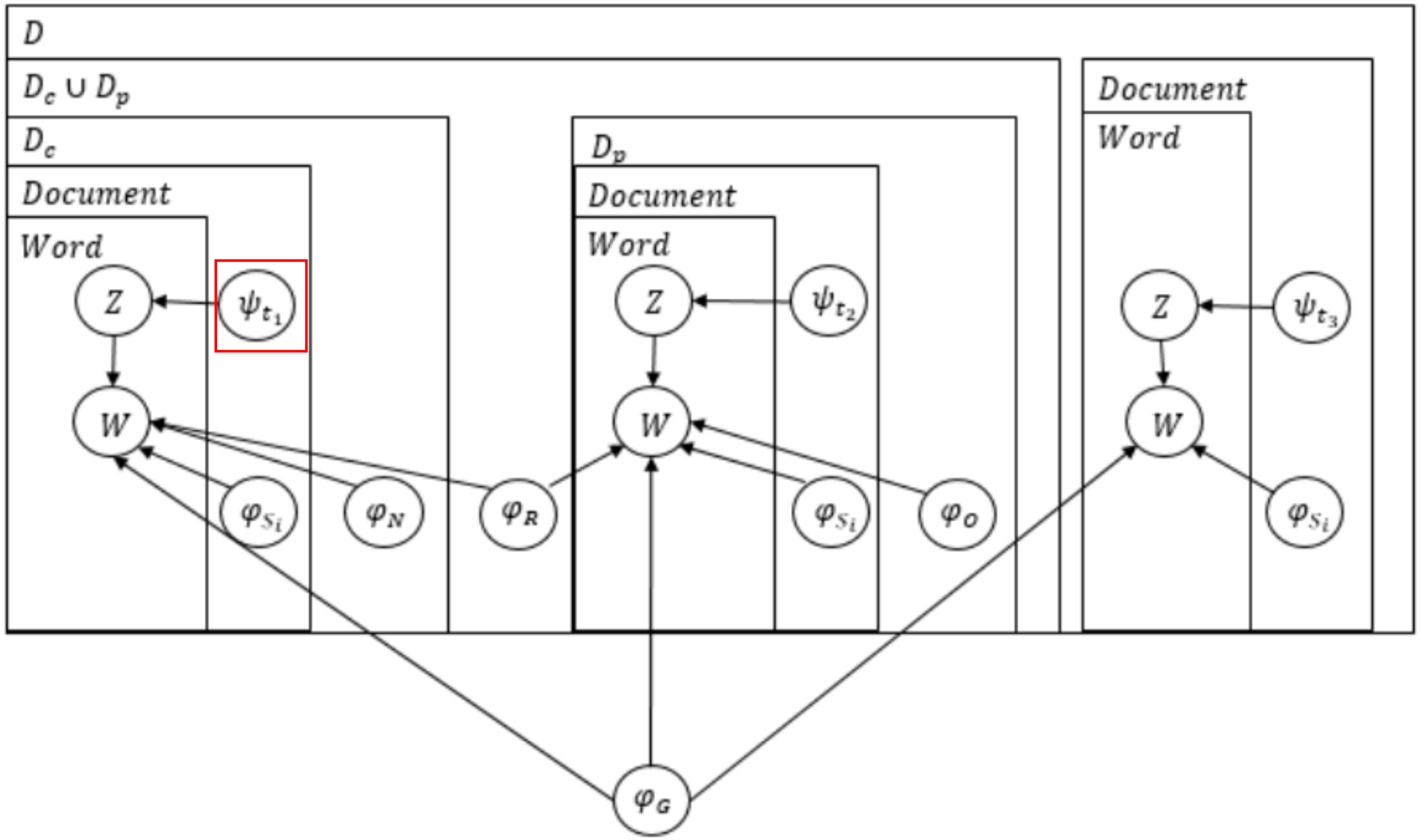
φ_N is drawn for all the documents in D_c from $Dirichlet(V, \lambda_N)$



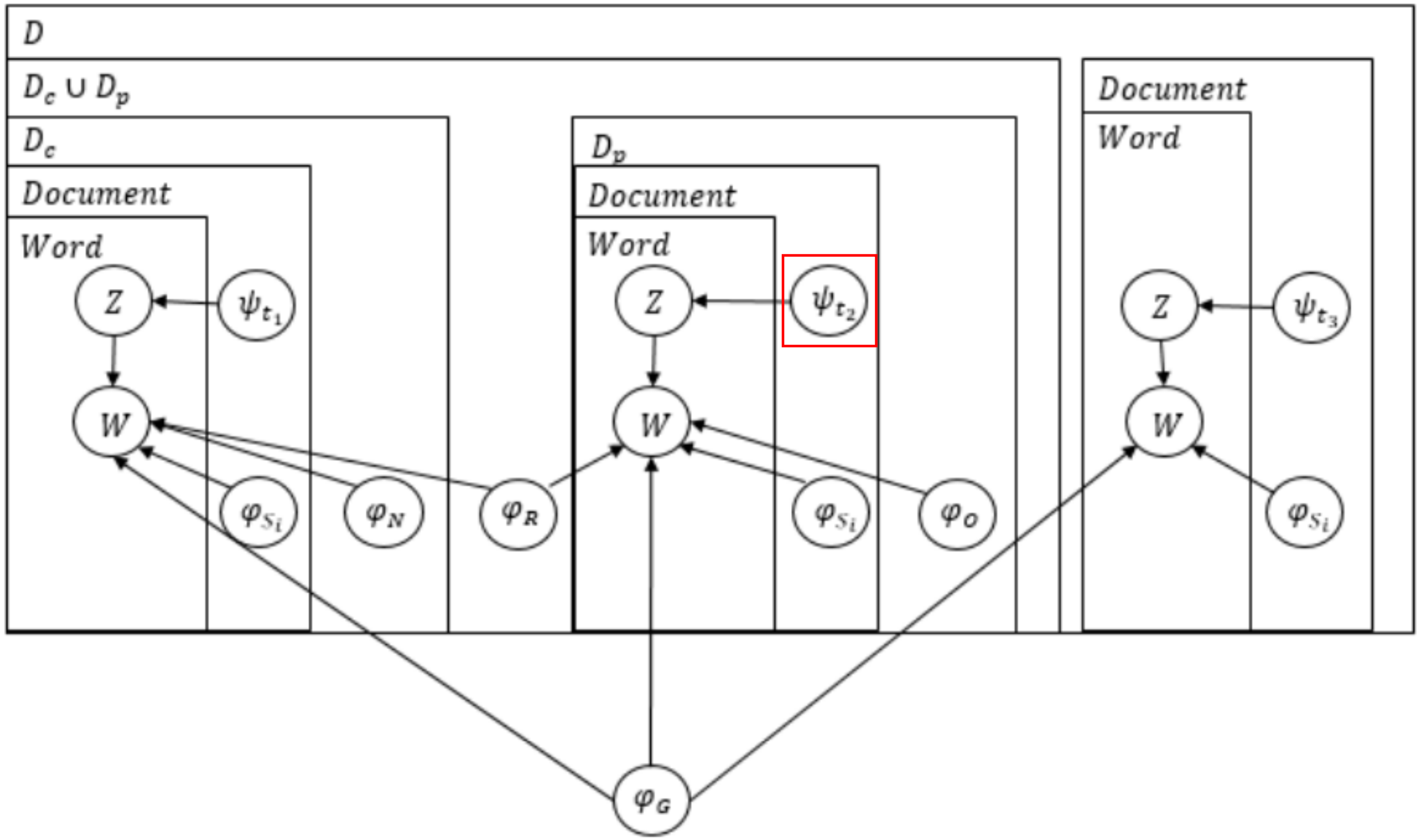
4. O captures *old content* from D_P,
φ_O is drawn for all the documents in D_P from $Dirichlet(V, \lambda_O)$



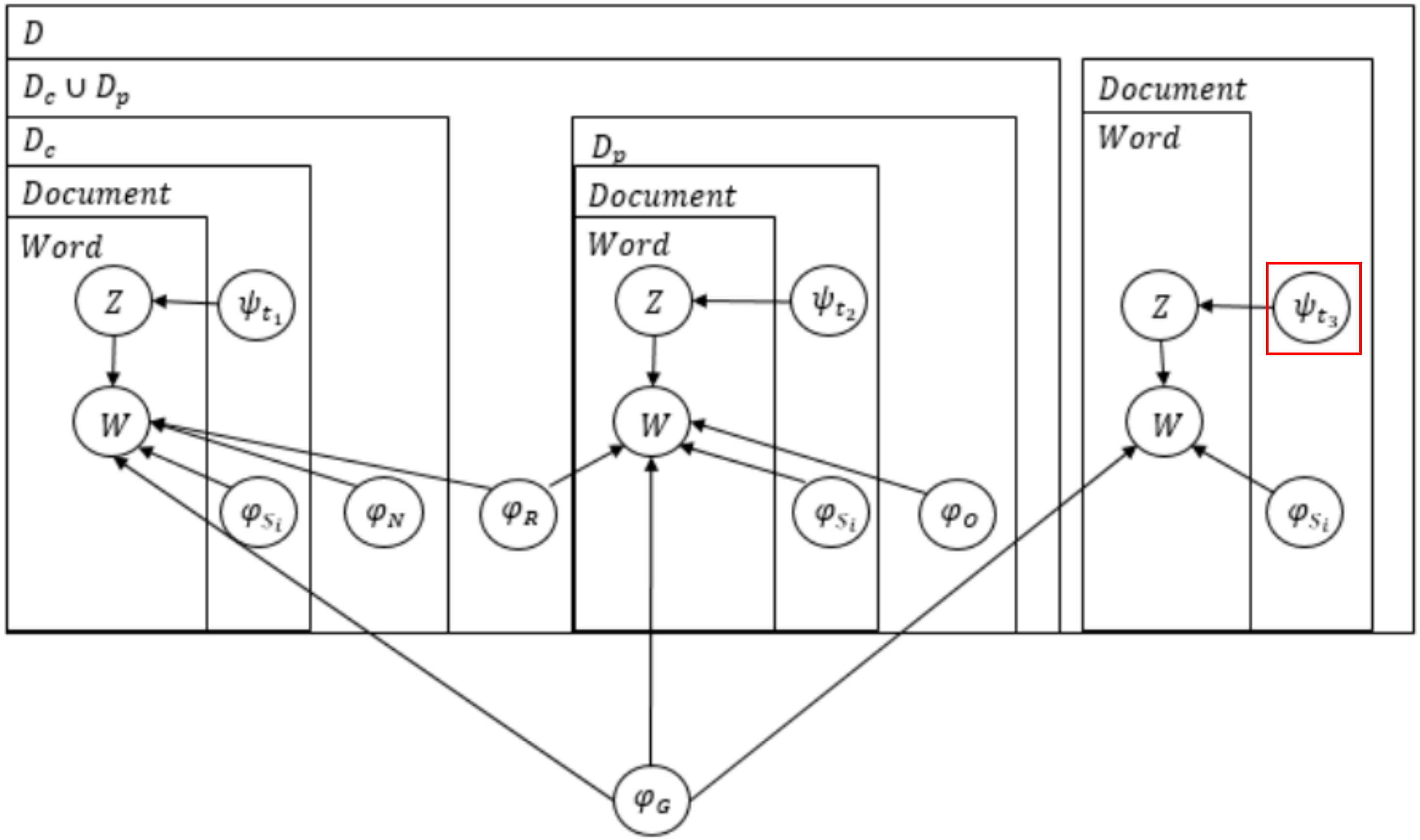
5. R captures *redundant information* between D_c and D_P,
φ_R is drawn for all the documents in $D_P \cup D_c$ from $Dirichlet(V, \lambda_R$



6. For documents from D_c draw from the distribution ψ_{t_1} over topics (G, N, R, S_i) from a Dirichlet prior with pseudo-counts (10.0, 15.0, 15.0, 1.0).
   For each word in the document, we draw a topic Z from ψ_{t_1}, and a word W from the topic indicated by Z.



7. For documents from D_P, draw from the distribution ψ_{t_2} over topics (G, O, R, S_i) from a Dirichlet prior with pseudo-counts (10.0, 15.0, 15.0, 1.0).
The words are drawn in the same manner as in t_1.



8. For documents from $D \setminus (D_C \cup D_P)$, draw from the distribution ψ_{t_3} over topics (G, S_i) from a Dirichlet prior with pseudo-counts (10.0, 1.0).
The words are drawn in the same manner as in t_1.

ChainSum —Procedure—

- Apply KLSum only on words that are assigned to the *new content* topic after the topic model is applied to the current query

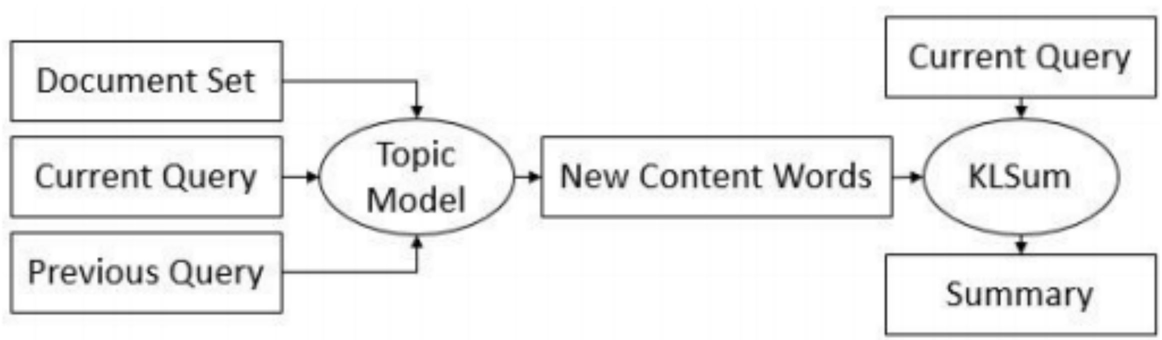


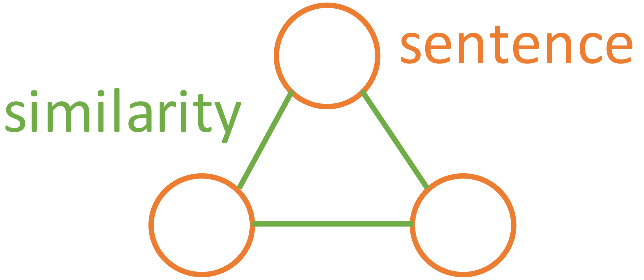
Figure 4 ChainSum Architecture

Algorithms

- Baselines (adapted existing methods to QCFS)
  - Focused KLSum
  - KL-Chain-Update
  - ChainSum
- New Algorithm for QCFS
  - Adapted LexRank

LexRank
—Overview—

- A stochastic graph-based method for computing the importance of sentences
  - Original algorithm does not address update or query-focused variants
- A summary is created by extracting top ranking sentences



1. Create a graph where nodes represent the sentences from the text
   – weighted edges represent the cosine similarity of each sentence's $TF \times IDF$ vectors
2. PageRank is run to rank sentences

Adapted LexRank —Overview—

- A variation of LexRank which is adapted to QCFS
- For the adaptation,
  - Extend the sentence representation scheme to capture semantic information
  - Refine the model of sentence’s similarity so that it captures query answering instead of centrality
  - Adapt to update of query

Adapted LexRank
―Overview―

- A variation of LexRank which is adapted to QCFS
- For the adaptation,
  - Extend the sentence representation scheme
    to capture semantic information
  - Refine the model of sentence's similarity
    so that it captures query answering instead of centrality
  - Adapt to update of query

Adapted LexRank ー sentence representation scheme ー

- Original LexRank only deal with lexical information for sentence representation
- Extend the sentence representation scheme to capture semantic information

  ⇨ Using Wikipedia terms and UMLS terms

- How to detect Wikipedia and UMLS terms?

  ⇨ Wikifier, HealthTermFinder

Adapted LexRank
ー sentence representation scheme ー

- edges scoring : $Score(U,V)$
  - use the sum of Lexical Semantic Similarity (LSS) functions on:
    - Lexical terms, Wikipedia terms, UMLS terms
  - $Score(U,V) = LSS_{lexical}(U,V) + a * LSS_{wiki}(U,V) + b * LSS_{UMLS}(U,V)$

where $$LSS(S_1, S_2) = \frac{\sum_i max_j \left(\frac{Sim\left(w_i^1, w_j^2\right)}{Sim\left(w_i^1, w_j^1\right)}\right) \times IDF\,(W_i^1)}{\sum_i IDF\,(w_i^1)}$$

- For lexical terms Sim is identity function
- For Wikipedia terms Sim is calculated by Wikiminer
- For UMLS terms Sim is calculated by Ted Pedersen UMLS similarity function

Adapted LexRank
—similarity example—

“Asthma is a common chronic inflammatory disease.”

Term Extraction

['Asthma', 'Inflammation']

“inhaler commonly used for long-term control.”

Term Extraction

['Inhaler']

Sim(‘Asthma’,’Inhaler’) = 0.725798946382
Sim(‘'Inflammation’,’Inhaler’) = 0.559059383727

Adapted LexRank —Overview—

- A variation of LexRank which is adapted to QCFS
- For the adaptation,
  - Extend the sentence representation scheme to capture semantic information
  - Refine the model of sentence’s similarity so that it captures query answering instead of centrality
  - Adapt to update of query

Adapted LexRank
ー Refine the Model ー

- To handle the query, added to the LexRank graph:
  - a new node representing the query
  - all the required edges
- Change the page rank algorithm to achieve similarity
  - In PageRank, the damping factor jumps to a random node in the graph

⇨ Allowed the damping factor to only jump back to the query node

Simulate the probability of reaching a sentence when starting a random walk at the query

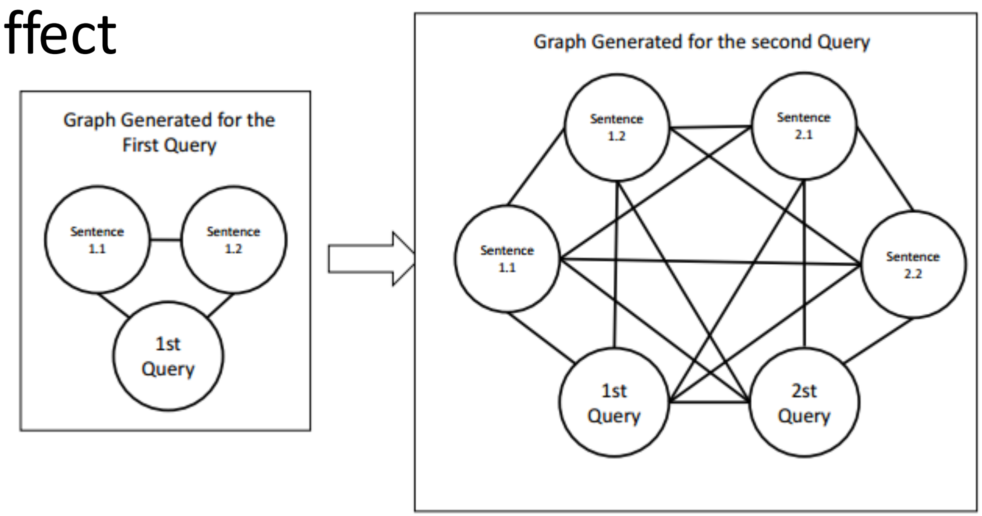
Adapted LexRank —Overview—

- A variation of LexRank which is adapted to QCFS
- For the adaptation,
  - Extend the sentence representation scheme to capture semantic information
  - Refine the model of sentence's similarity so that it captures query answering instead of centrality
  - Adapt to update of query

Adapted LexRank
ー Adapt to update of query ー

- To adapt the query updates, the model is incorporated into the following changes:
  - Did not create a new graph
  - Merged the graph from the previous query and with the new query and sentences from the top N documents fetched

Enable a drill down effect to the summary



Adapte LexRank
ー Avoid Redundancy ー

- After ranking, select only sentences that are:
  - different from sentences that are selected for the current summary
  - also different from sentences that are selected for the previous summaries in the session

6. Evaluation

Evaluation Dataset

- Using dataset created for QCFS
- Added semantictags:
  - 10% of the tokens had Wikipedia annotations
  - 33% had a UMLS annotation

Results
—Automatic Evaluation—

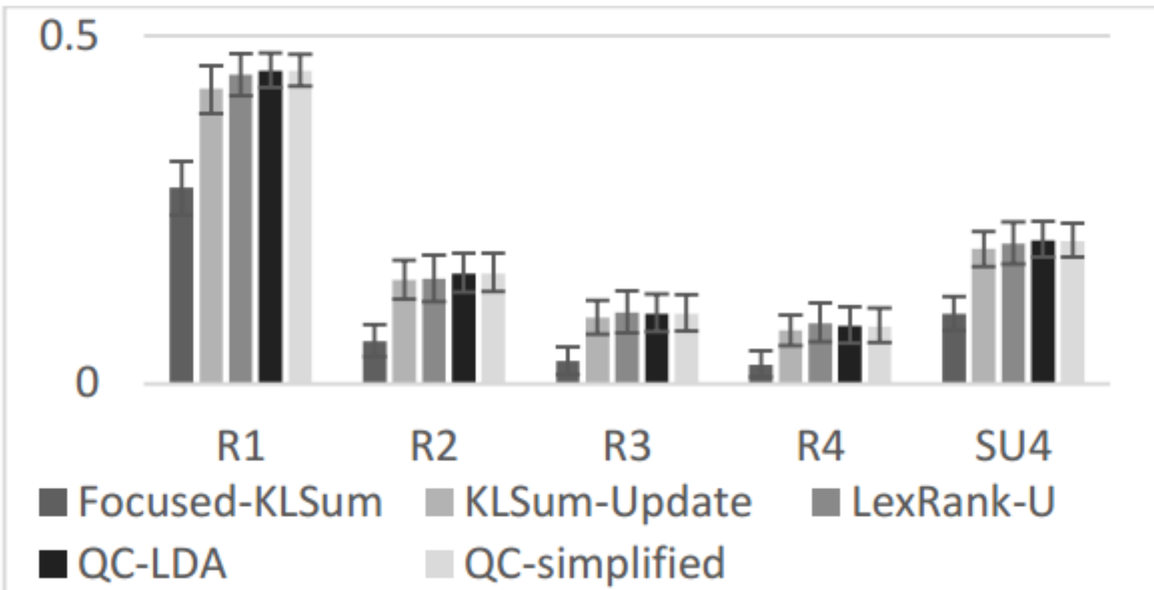


Figure 5: ROUGE Recall Scores (with stemming and stop-words)

Conclusions

- Presented new summarization task: QCFS
- Construct a novel dataset for QCFS containing human summaries
- Four methods were evaluated for the task. The baseline methods based on KLSum show a significant improvement when penalizing redundancy with the previous summarization.

Future Work

- This paper only concentrated on *zoom in* query chains
    - *Zoom out* or *switch topic* were left to future work
- Attempt to derive a task-specific evaluation metric that exploits the structure of the chains better assess relevance, redundancy and contrast