Subtopic Ranking Based on Hierarchical Headings

Tomohiro Manabe and Keishi Tajima

Graduate School of Informatics, Kyoto Univ.

{manabe@dl.kuis, tajima@i}.kyoto-u.ac.jp

What are subtopics?

- We focus on a topic given as a keyword query
- A *subtopic* of a given keyword query is:
  Another keyword query that specializes and/or disambiguates the search intent of the given query

harry potter	Search		office	Search

✓ harry potter movie
✗ harry potter hp

✓ office workplace
✗ office office

Sakai, T., Dou, Z., Yamamoto, T., Liu, Y., Zhang, M., and Song, R. (2013). Overview of the NTCIR-10 INTENT-2 task. In NTCIR.

Why are subtopics important?

Subtopics are useful for

- Query suggestion/completion
- Search result diversification
    - By including a few pages for each subtopic in the search result

Our Problem: Subtopic Ranking

- Query suggestion/completion
  - Which subtopic should be suggested?
- Search result diversification
  - Which subtopic should be included in the search results?

Subtopic Ranking Problem

Sorting subtopics by their *intent probabilities*

(the probability that the user intends that subtopic)

Our Idea: Hierarchical Headings are useful

We use *hierarchical heading structure* in documents

It consists of:

- Nested logical *blocks*
- Each block has its own *heading*
  - A heading describes its own and descendant blocks

Assumption 1:

Hierarchical headings represent hierarchical topics

Example Document

Programming

All about computer programming skills.

Schools

Top schools for computer …

Courses

Specifically, the most famous …

Degrees

Some schools award degrees …

Jobs

Programming skills are required …

Programming

- Programming schools
  - Programming school courses
  - Programming school degrees
- Programming jobs

Programming

All about computer programming skills.

> *Schools*
>
> Top schools for computer …
>
> > *Courses*
> >
> > Specifically, the most famous …
>
> > *Degrees*
> >
> > Some schools award degrees …

> *Jobs*
>
> Programming skills are required …

E.g. Schools block contains more letters and descendant blocks than Jobs block

- Authors must have assumed the readers need more information on "Schools"
- It suggests that "Schools" have higher intent probability

Assumption 2:
<u>Subtopics with more contents are more important</u>

Overview of our Assumptions and Methods

Our assumptions are:

- Hierarchical headings represent hierarchical topics
- Topics with more contents is more important

Our subtopic ranking method:

1. Score blocks based on their content quantity
2. Score subtopics by integrating the scores of blocks *matching* the subtopics
3. Rank the subtopics based on their scores

Matching between Subtopics and Blocks

A subtopic *matches* a block iff:

All words in the subtopic appear either in the headings of the block or of its ancestor blocks

Before comparing, we perform basic preprocessing

- Tokenization
- Stop word filtering
- Stemming

Example of Matching

Programming

All about computer programming skills.

Schools

Top schools for computer …

Courses

Specifically, the most famous …

Degrees

Some schools award degrees …

Jobs

Programming skills are required …

Subtopic “programming schools” matches block “schools” in this document.

NOTE: if a topic matches a block, its descendant blocks also match it, but we only consider top-most matching blocks

Overview of our Methods

1. Score blocks based on their content quantity
   We compare 4 block-scoring methods
2. Score subtopics by integrating scores of blocks *matching* the subtopics
   We compare 4 integration methods
3. Rank the subtopics based on their scores
   We compare 2 ranking methods

total: 4x4x2=32 methods

Overview of our Methods

Our subtopic ranking methods:

1. Score blocks based on their content quantity

   We compare 4 block-scoring methods

2. Score subtopics by integrating scores of blocks *matching* the subtopics

   We compare 4 integration methods

3. Rank the subtopics based on their scores

   We compare 2 ranking methods

1. Scoring Blocks Based on Content Quantity

We compare four block-scoring methods:

1-A. Length scoring

1-B. Log-scale scoring

1-C. Bottom-up scoring

1-D. Top-down scoring

1-A. Length Scoring

Idea: Block with more text is more important

Score a block by the number of letters in it

- Including those in descendant blocks

Programming 3,000 letters
All about computer programming skills.

Schools 2,500 letters
Top schools for computer …

Courses 1,600 letters
Specifically, the most famous …

Degrees 400 letters
Some schools award degrees …

Jobs 440 letters
Programming skills are required …

1-B. Log-Scale Scoring

Idea: Importance of block is not linearly proportional to its content quantity

Score a block by logarithm of the numbers of letters in it

Programming log(3k) ≈ 3.5
All about computer programming skills.

Schools log(2,500) ≈ 3.4
Top schools for computer …

Courses log(1,600) ≈ 3.2
Specifically, the most famous …

Degrees log(400) ≈ 2.6
Some schools award degrees …

Jobs log(440) ≈ 2.6
Programming skills are required …

1-C. Bottom-up Scoring

Idea: Importance of some topics are independent from text length

- e.g. telephone number

Score a block by the number of blocks in it (including itself)

Programming 1+3+1=5
All about computer programming skills.

> *Schools* 1+1+1=3
> Top schools for computer …
>
> > *Courses* 1
> > Specifically, the most famous …
>
> > *Degrees* 1
> > Some schools award degrees …

> *Jobs* 1
> Programming skills are required …

1-D. Top-down Scoring

Idea: Authors often divide a block into child blocks that have the equal importance

$$\text{score} = \frac{\text{parent's score}}{|\text{sibling}| + 1}$$

Programming 1
All about computer programming skills.

Schools 1 / (2 + 1) = 1/3
Top schools for computer …

Courses (1/3) / (2 + 1) = 1/9
Specifically, the most famous …

Degrees (1/3) / (2 + 1) = 1/9
Some schools award degrees …

Jobs 1 / (2 + 1) = 1/3
Programming skills are required …

Overview of our Methods

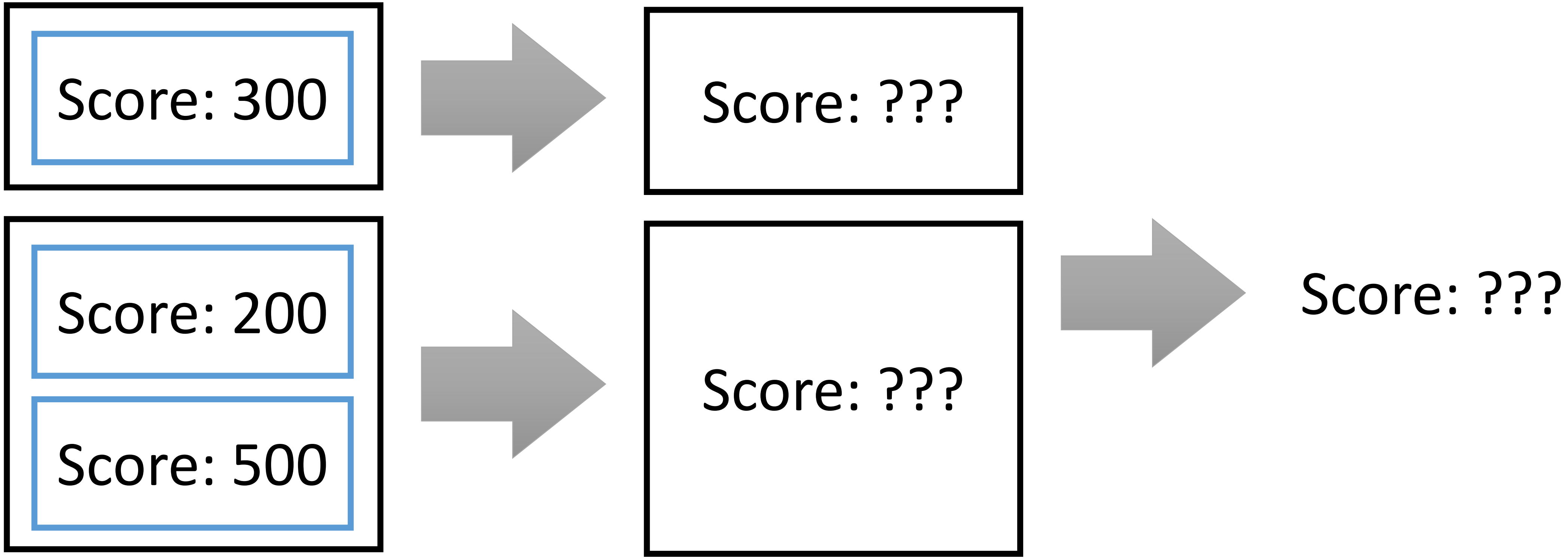
Our subtopic ranking methods:

1. Score blocks based on their content quantity
   We compare 4 block-scoring methods
2. Score subtopics by integrating scores of blocks *matching* the subtopics
   We compare 4 integration methods
3. Rank the subtopics based on their scores
   We compare 2 ranking methods

2. Score Subtopics by Integrating Scores of Matching Blocks

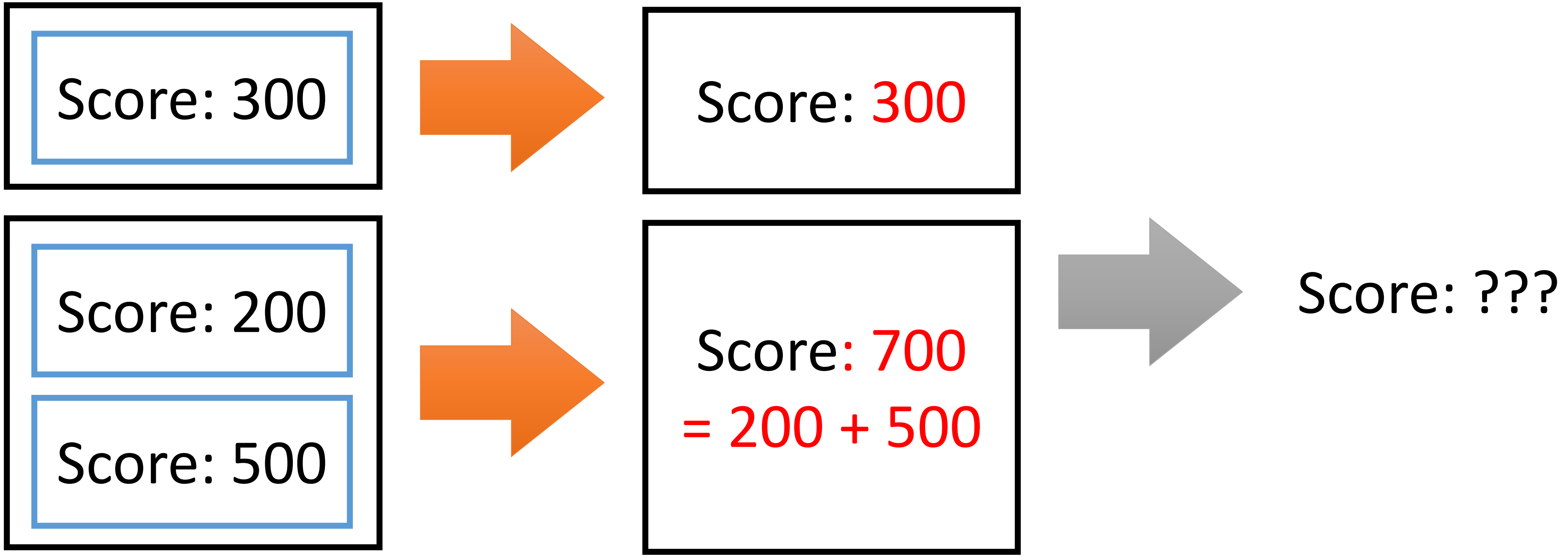
2-1. Integrate the block scores into document scores

2-2. Integrate the document scores into the final score



2-1. Integrate Block Scores into Document Score

- Simply sum up the scores of all matching blocks in each document



2-2. Integrate Document Scores into the Final Score

We compare four integration methods:

2-2-a. Simple Summation

2-2-b. Per-Document Normalization

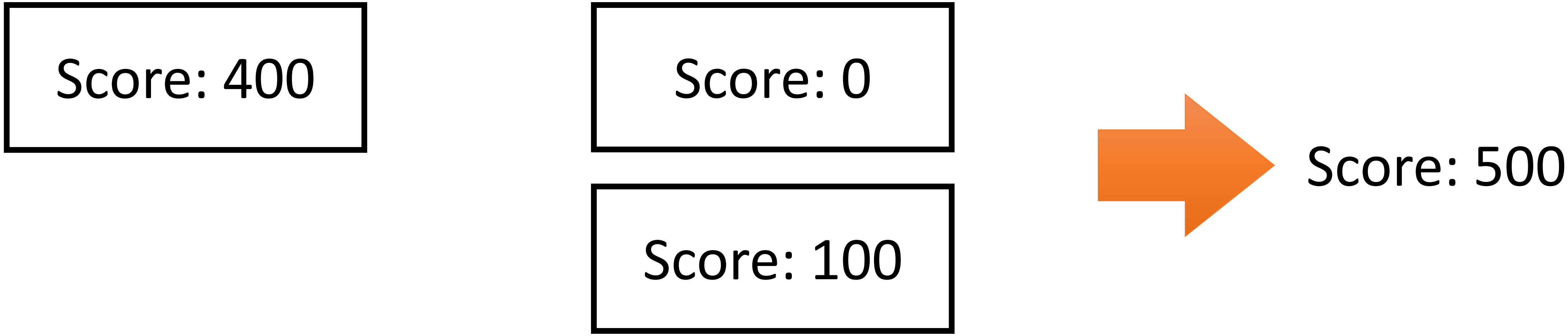
2-2-c. Per-Domain Normalization

2-2-d. Hybrid Normalization

2-2-a. Simple Summation

Simply sum up scores of multiple documents

- The score of a subtopic is content quantity in whole corpus



2-2-b. Per-Document Normalization

- In summation method, documents with more contents have bigger influence on scores
- However, each document may be equally important

Divide scores by the scores of the root block of document

Score:
400 / 500

Score:
0 / 900

Score:
100 / 100

Score: 1.8

2-2-c. Per-Domain Normalization

- We can also consider per-domain normalization

Divide total score of matching blocks in a domain by the total score of root blocks in the domain

http://abc.com/

Score: 400 / 500

Score:
400 /500

http://def.com/

Score: (100+0) / (900 + 100)

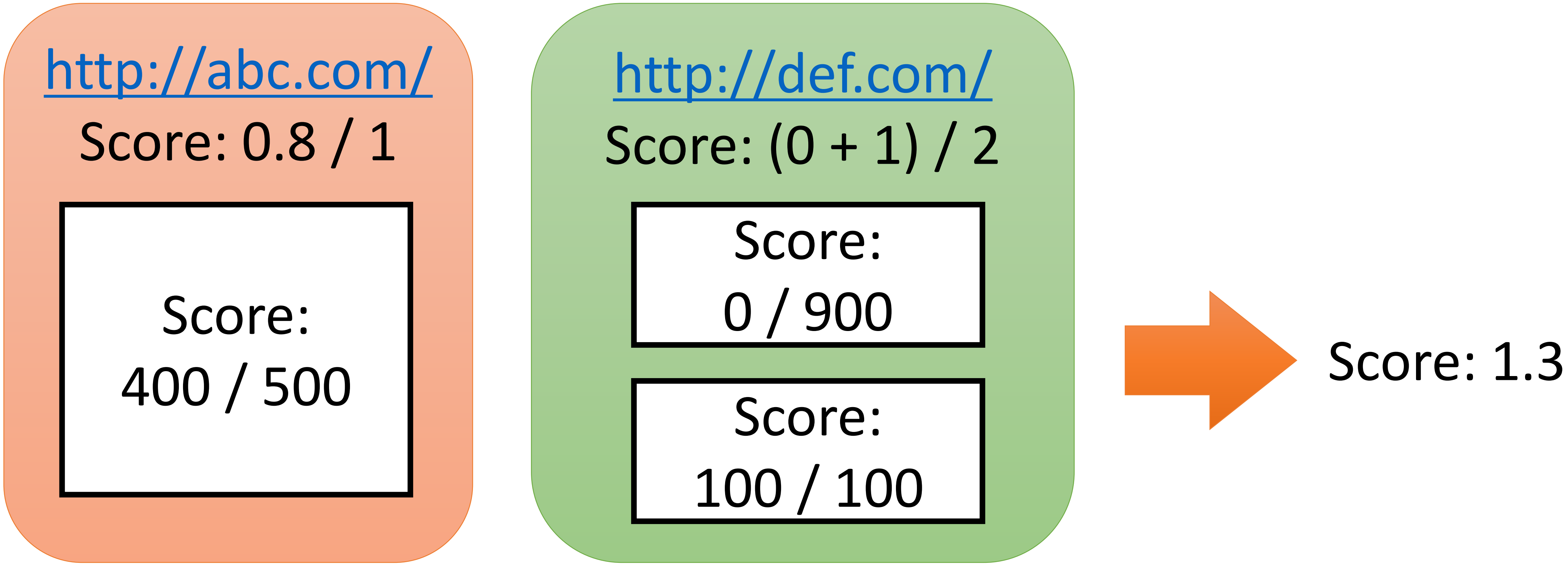
Score:
0 / 900

Score:
100 / 100

Score: 0.9

2-2-d. Hybrid Normalization

Apply both page-based and domain-based normalization



Overview of our Methods

Our subtopic ranking methods:

1. Score blocks based on their content quantity

   We compare 4 block-scoring methods

2. Score subtopics by integrating scores of blocks *matching* the subtopics

   We compare 4 integration methods

3. Rank the subtopics based on their scores

   We compare 2 ranking methods

3. Rank The Subtopics based on Their Scores

We compare 2 ranking methods:

3-A. Simple Ranking Method

3-B. Diversified Ranking Method

3-A. Simple Ranking Method

- Simply sort subtopics by their scores

Example Subtopics	Score
Programming Schools	2,500
Programming School Courses	1,600
Programming Jobs	440

Programming 3,000 letters

All about computer programming skills.

Schools 2,500 letters

Top schools for computer …

Courses 1,600 letters

Specifically, the most famous …

Degrees 400 letters

Some schools award degrees …

Jobs 440 letters

Programming skills are required …

3-B. Diversified Ranking Method

- As search result diversification is an important application, we also want <u>diversified ranking of subtopics</u>
- Basic idea is:
  - If a block matches an already-ranked subtopic,
    the topic of the block is already included in the ranking
  - So even if the block also matches some lower-ranked subtopics, the block should not contribute to their scores

3-B. Diversified Ranking Method

Each time a subtopic is ranked, all blocks matching the subtopic is removed

Example Subtopics	Score
Programming Schools	2,500
Programming School Courses	~~1,600~~ 0
Programming Jobs	440

Programming 3,000 letters

All about computer programming skills.

Schools 2,500 letters

Top schools for computer …

Courses 1,600 letters

Specifically, the most famous …

Degrees 400 letters

Some schools award degrees …

Jobs 440 letters

Programming skills are required …

Evaluation

We compared:

- Three baselines
- Our 4*4*2=32 proposed methods

Block Scoring

- Length
- Log-scale
- Bottom-up
- Top-down

Integration

- Summation
- Per-Page
- Per-Domain
- Hybrid

Ranking

- Simple
- Diversified

Data Set

Data set used in NTCIR-10 INTENT-2

- Fifty keyword queries (i.e., topics)
- Baseline subtopic rankings for them
  - Snapshots of query completion results by Google, Yahoo!
  - Merged and dictionary-sorted query completion or suggestion results of three commercial search engines
- Known subtopics of each query and their intent probabilities (probability that the user intends that subtopic)

Sakai, T., Dou, Z., Yamamoto, T., Liu, Y., Zhang, M., and Song, R. (2013). Overview of the NTCIR-10 INTENT-2 task. In NTCIR.

Evaluation Methodology

- We extract hierarchical headings (i.e., subtopics) from documents in baseline rankings for TREC 2012 Web (131-837 web pages for each query)
  - Hierarchical headings were extracted by our previously proposed method [Manabe, Tajima, VLDB2015]
- Calculate the scores of the extracted subtopics
- Re-rank baseline subtopic rankings
- Evaluate top-10 subtopics

Evaluation Measures

I-rec: $\dfrac{|\text{Actual subtopics in the ranking}|}{|\text{All actual subtopics}|}$

- Measures recall and diversity of subtopics in rankings

D-nDCG is like nDCG for document rankings

- The more actual subtopics at higher ranks, D-nDCG score of the ranking gets higher

D#-nDCG: Mean of I-rec and D-nDCG

Sakai, T., Dou, Z., Yamamoto, T., Liu, Y., Zhang, M., and Song, R. (2013). Overview of the NTCIR-10 INTENT-2 task. In NTCIR.

Comparison with Google (I-rec@10 = 0.3841)

Scoring	Integration	Ranking	D-nDCG@10
Log-scale	Domain	Uniform	.4502
Log-scale	Combi.	Uniform	.4501
Log-scale	Domain	Diversified	.4487
Log-scale	Combi.	Diversified	.4485
Bottom-up	Page	Diversified	.4479
Baseline (Google query completion)			.3735

Comparison with Yahoo! (I-rec@10 = 0.3815)

Scoring	Integration	Ranking	D-nDCG@10
Log-scale	Page	Diversified	.4617
Bottom-up	Domain	Diversified	.4609
Log-scale	Page	Uniform	.4608
Log-scale	Summation	Diversified	.4601
Length	Domain	Diversified	.4587
Baseline (Yahoo! query completion)			.3829

Comparison with merged and dictionary-sorted subtopics

Scoring	Integration	Ranking	I-rec@10	D-nDCG@10	D#-nDCG@10
Log-scale	Summation	Uniform	.4009	.3997	.4003
Log-scale	Page	Uniform	.3986	.3981	.3984
Length	Summation	Uniform	.3974	.3945	.3959
Log-scale	Combi.	Uniform	.3956	.3921	.3939
Log-scale	Domain	Uniform	.3956	.3913	.3934
Baseline (Merged, dictionary-sort)			.3310	.3066	.3188

Comparison with Google (I-rec@10 = 0.3841)

Scoring	Integration	Ranking	D-nDCG@10
Log-scale	Domain	Uniform	.4502
Log-scale	Combi.	Uniform	.4501
Log-scale	Domain	Diversified	.4487
Log-scale	Combi.	Diversified	.4485
Bottom-up	Page	Diversified	[illegible]
Baseline (Google query completion)			.3735

Log-scale/Page/Diversified .4470

Comparison with Yahoo! (I-rec@10 = 0.3815)

Scoring	Integration	Ranking	D-nDCG@10
Log-scale	**Page**	**Diversified**	**.4617**
Bottom-up	Domain	Diversified	.4609
Log-scale	Page	Uniform	.4608
Log-scale	Summation	Diversified	.4601
Length	Domain	Diversified	.4587
Baseline (Yahoo! query completion)			.3829

Comparison with merged and dictionary-sorted subtopics

Scoring	Integration	Ranking	I-rec@10	D-nDCG@10	D#-nDCG@10
Log-scale	Summation	Uniform	.4009	.3997	.4003
Log-scale	Page	Uniform	.3986	.3981	.3984
Length	Summation	Uniform	.3974	.3945	.3959
Log-scale	Combi.	Uniform	.3956	.3921	.3939
Log-scale	Domain	Uniform	[illegible]	[illegible]	[illegible]
Baseline (Merged, dictionary-sort)			.3310	.3066	.3188

Log-scale/Page/Diversified .3840 .3695 .3768

Comparison with Google (I-rec@10 = 0.3841)

Scoring	Integration	Ranking	D-nDCG@10
Log-scale	Domain	Uniform	.4502
Log-scale	Combi.	Uniform	.4501
Log-scale	Domain	Diversified	.4487
Log-scale	Combi.	Diversified	.4485
Bottom-up	Page	Diversified	.4479
Baseline (Google query completion)			.3735

Comparison with Yahoo! (I-rec@10 = 0.3815)

Scoring	Integration	Ranking	D-nDCG@10
Log-scale	Page	Diversified	.4617
Bottom-up	Domain	Diversified	.4609
Log-scale	Page	Uniform	.4608
Log-scale	Summation	Diversified	.4601
Length	Domain	Diversified	.4587
Baseline (Yahoo! query completion)			.3829

Comparison with merged and dictionary-sorted subtopics

Scoring	Integration	Ranking	I-rec@10	D-nDCG@10	D#-nDCG@10
Log-scale	Summation	Uniform	.4009	.3997	.4003
Log-scale	Page	Uniform	.3986	.3981	.3984
Length	Summation	Uniform	.3974	.3945	.3959
Log-scale	Combi.	Uniform	.3956	.3921	.3939
Log-scale	Domain	Uniform	.3956	.3913	.3934
Baseline (Merged, dictionary-sort)			.3310	.3066	.3188

Comparison with Google (I-rec@10 = 0.3841)

Scoring	Integration	Ranking	D-nDCG@10
Log-scale	Domain	Uniform	.4502
Log-scale	Combi.	Uniform	.4501
Log-scale	Domain	Diversified	.4487
Log-scale	Combi.	Diversified	.4485
Bottom-up	Page	Diversified	.4479
Baseline (Google query completion)			.3735

Comparison with Yahoo! (I-rec@10 = 0.3815)

Scoring	Integration	Ranking	D-nDCG@10
Log-scale	Page	Diversified	.4617
Bottom-up	Domain	Diversified	.4609
Log-scale	Page	Uniform	.4608
Log-scale	Summation	Diversified	.4601
Length	Domain	Diversified	.4587
Baseline (Yahoo! query completion)			.3829

Comparison with merged and dictionary-sorted subtopics

Scoring	Integration	Ranking	I-rec@10	D-nDCG@10	D#-nDCG@10
Log-scale	Summation	Uniform	.4009	.3997	.4003
Log-scale	Page	Uniform	.3986	.3981	.3984
Length	Summation	Uniform	.3974	.3945	.3959
Log-scale	Combi.	Uniform	.3956	.3921	.3939
Log-scale	Domain	Uniform	.3956	.3913	.3934
Baseline (Merged, dictionary-sort)			.3310	.3066	.3188

Comparison with Google (I-rec@10 = 0.3841)

Scoring	Integration	Ranking	D-nDCG@10
Log-scale	Domain	Uniform	.4502
Log-scale	Combi.	Uniform	.4501
Log-scale	Domain	Diversified	.4487
Log-scale	Combi.	Diversified	.4485
Bottom-up	Page	Diversified	.4479
Baseline (Google query completion)			.3735

Comparison with Yahoo! (I-rec@10 = 0.3815)

Scoring	Integration	Ranking	D-nDCG@10
Log-scale	Page	Diversified	.4617
Bottom-up	Domain	Diversified	.4609
Log-scale	Page	Uniform	.4608
Log-scale	Summation	Diversified	.4601
Length	Domain	Diversified	.4587
Baseline (Yahoo! query completion)			.3829

Comparison with merged and dictionary-sorted subtopics

Scoring	Integration	Ranking	I-rec@10	D-nDCG@10	D#-nDCG@10
Log-scale	Summation	Uniform	.4009	.3997	.4003
Log-scale	Page	Uniform	.3986	.3981	.3984
Length	Summation	Uniform	.3974	.3945	.3959
Log-scale	Combi.	Uniform	.3956	.3921	.3939
Log-scale	Domain	Uniform	.3956	.3913	.3934
Baseline (Merged, dictionary-sort)			.3310	.3066	.3188

Conclusion

Our ideas

- Hierarchical headings represent topic structure
- Length of contents for each topic ≈ importance of the topic

Our methods

- Rank subtopics based on scores of blocks whose hierarchical headings match the subtopics

Our evaluation results indicated

- Our methods improved baseline rankings
- Log-scale scoring seems effective
- No difference among our score integration methods
- Our diversified ranking method was not effective