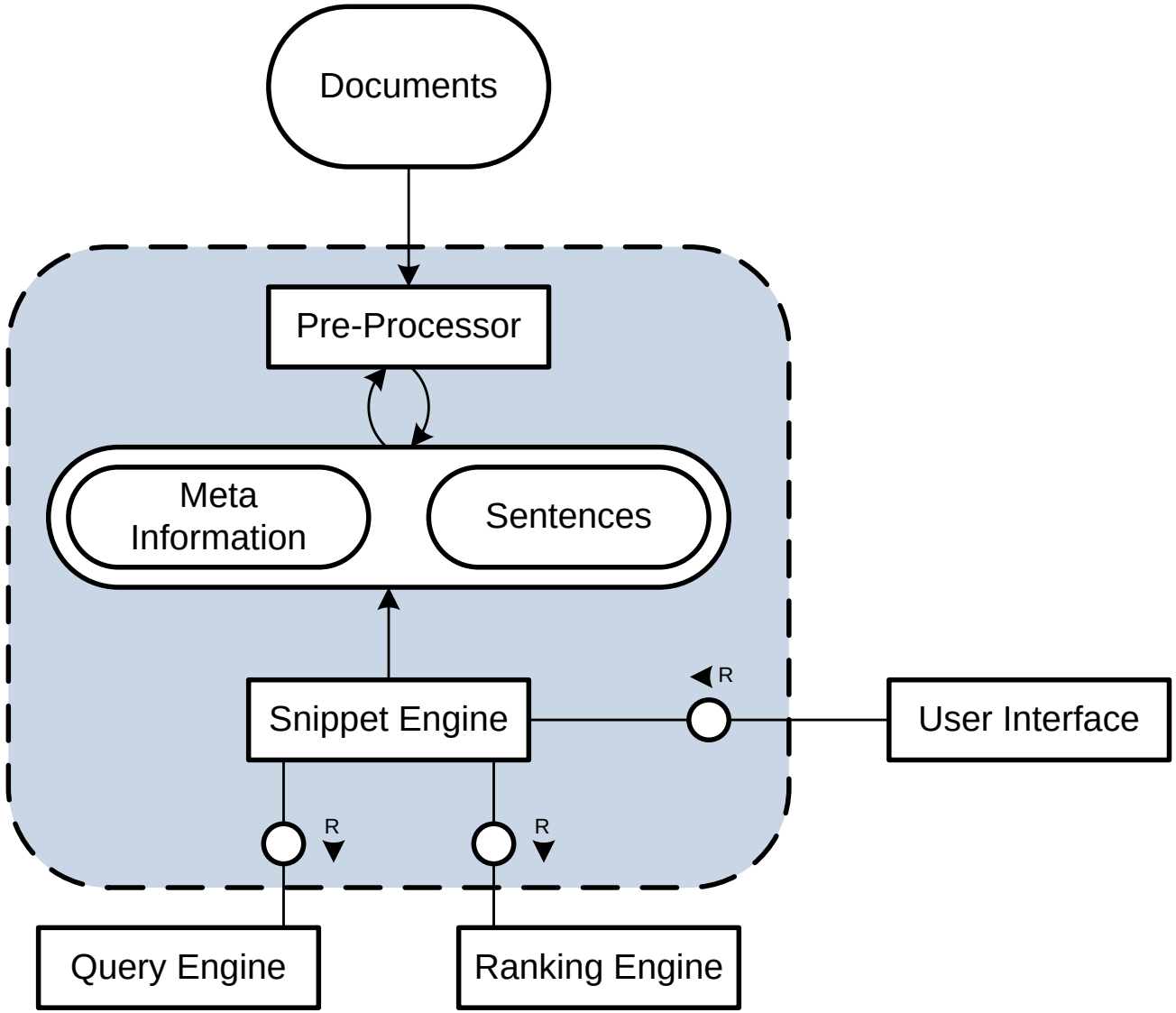
HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Snippet Generation for WikiSearch

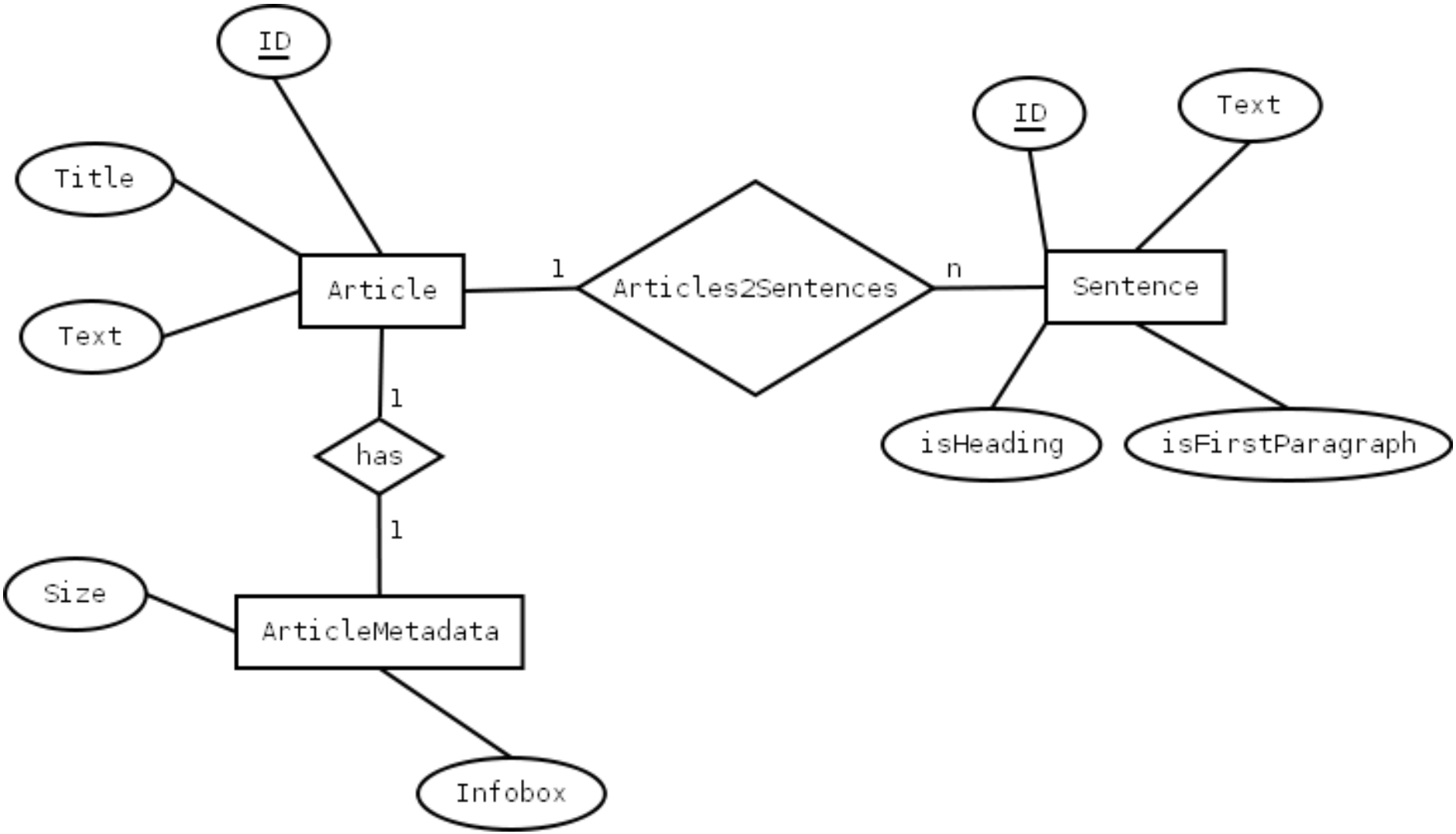
Felix Geller,
Robert Krahn,
Alexander Krasnogolowy

July 21, 2009

Architecture



Database Schema



Demo

Hoover, Alabama
The city is divided into two battalions. There are eight **engine** companies, three quints, three ALS rescue/ambulances, and two battalion chiefs [4.08]... All **engine** companies are staffed with a minimum of three, with at least two being firefighter/paramedics [4.0]... BE&K **Engineering** - 302 [3.88]...

Official Name:	Hoover, Alabama
Settlement Type:	City

kheed Vega, a single **engine** plane that carried four passengers
nental system, as well as competitors flights, to assure tight quality
adopted to Continentals network [4.08]... Continental Airlines, Inc.

efighters. The CFD utilizes three **Engines,** a Ladder truck, a Heavy
h in Bergen County, New Jersey, United States [2.5]... As of the
tion was 5,917 [2.0]...

bathyscaphe Trieste, oceanographic ship Mizar and other ships,
r, some 8,400 feet (2560 m) below the surface, in six
he sail, sonar dome, bow section, **engineering** spaces section,
[4.08]... In later life, McCoole was sure he would have delayed

Features

- WikiMedia markup parser
- Sentence normalization separated from query processing
- Flexible ranking algorithm [Turpin et al., 2007]
- Account for WikiMedia markup specific features
- 42 unit tests
- LOC: 575 (parser) + 392 (front-end) = 967
  (`cat *java | grep \; | grep -v  // | wc -l`)

Lessons learned

- WikiMedia markup is not *that* simple
- Best to process it in multiple passes
- Sentence normalization should be handled earlier in search engine architecture
- Unicode is fun

Ranking Algorithm


```
1   if (isHeading)
2     rank+= HEADING_WEIGHT;
3   else
4     rank+= (1.0F / sentence.getInParagraphPosition());
5
6   if (isInFirstParagraph)
7     rank+= FIRST_PARAGRAPH_WEIGHT;
8
9   rank+= queryTermsCount * REPEATED_QT_WEIGHT;
10  rank+= distinctQueryTermsCount * DISTINCT_QT_WEIGHT;
11  rank+= highestConsecutiveCount * CONSECUTIVE_QT_WEIGHT;
```

Ranking Example

- Sentence: "Sun Microsystems has released the Niagara and Niagara 2 chips, both of which feature an eight-core design."

```
1   static final float HEADING_WEIGHT = 0.5F;                    // added
2   static final float FIRST_PARAGRAPH_WEIGHT = 1.5F;            // higher than heading!
3   static final float REPEATED_QUERY_TERMS_WEIGHT = 1.25F;      // multiplied
4   static final float DISTINCT_QUERY_TERMS_WEIGHT = 1.5F;       // multiplied
5   static final float CONSECUTIVE_QUERY_TERMS_WEIGHT = 1.0F; // multiplied
6   static final float RANK_THRESHOLD = 1.5F + (1.0F * DISTINCT_QUERY_TERMS_WEIGHT);
7   static final int SENTENCES_COUNT_THRESHOLD = 3;
8
9   float rank = 0.0F;
10  if (isHeading)
11      rank+= HEADING_WEIGHT;
12  else
13      rank += (1.0F / sentence.getInParagraphPosition());
14  if (isInFirstParagraph)
15      rank+= FIRST_PARAGRAPH_WEIGHT;
16  // rank = 1
17  rank += matchedQueryTermsCount * REPEATED_QUERY_TERMS_WEIGHT;
18  // rank = 1 + 2*1.25 = 3.5
19  rank += matchedDistinctQueryTermsCount * DISTINCT_QUERY_TERMS_WEIGHT;
20  // rank = 3.5 + 2*1.5 = 6.5
21  rank += highestConsecutiveCount * CONSECUTIVE_QUERY_TERMS_WEIGHT;
22  // rank = 6.5 + 2*1 = 8.5
```

Parser Dispatch


```
1   while (!atEnd()) {
2     step();
3     if (parseHeading()) continue;
4     if (parseProcessingAction()) continue;
5     if (parseInfobox()) continue;
6     if (parseTemplate()) continue;
7     if (parseTag()) continue;
8     if (parseEscapedTag()) continue;
9     if (parseFileLink()) continue;
10    if (parseEmbeddedLink()) continue;
11    if (parseLink()) continue;
12    if (parseList()) continue;
13    if (parseDots()) continue;
14    if (parseParagraphEnd()) continue;
15    if (parseSentenceEnd(WikiPart.SENTENCE)) continue;
16    if (parseUnimportantChars()) continue;
17    if (parseSentenceChar()) continue;
18    throw new Error("Not recognized input");
19  }
```

References

Turpin, A., Tsegay, Y., Hawking, D., and Williams, H. (2007).
Fast generation of result snippets in web search.
In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 23–27.