Outline of a Benchmark for Publishing

First TUC meeting

Barcelona 19/20th November 2012

Overview

- Semantic publishing
- Characteristics of the use-case
- What should be measured?
- Outline of the benchmark

Semantic Publishing 1

- More powerful method for exploiting content
- Based on annotations that associate content (assets) with concepts and instances in ontologies (hence semantic annotation)
- Annotations can be generated in many ways, but text-processing is common (even for photos and video)

Semantic Publishing 2

- Semantic search of content metadata is powerful, especially when combined with ranking, geo-tagging and full-text search, e.g.
  - Find stories about X
  - No? Find stories that mention X
  - No? Find stories located close to X
  - No? Find stories about anything in the same class as X, located within 20 miles of X or about something that sounds like X

Semantic Publishing 3

- Inference makes many things possible, e.g.
  - with a suitable ontology and lightweight inference, a query for stories about sports people or their families will return a story about Posh Spice

Semantic Publishing 4

- Templates for aggregating content can be developed, semantic search is used to fill in the gaps, e.g.
  - Most recent story about the player at the top, or if nothing about him then about his team. Failing that, use any story that mentions his country.
  - Insert photo about the player (or his team, or his family or use a video, etc, etc)

Semantic Publishing 5

- RDF database with inference used to manage metadata
- Vast majority of operations are for searching content for dynamic aggregation
- Metadata updated in real-time as new content is added/processed
- When text-analytics used to generate metadata then occasional reload of gazetteers

Characteristics of the use-case

- RDF database that stores:
  - Large reference datasets
  - Domain specific ontologies
  - Metadata
- Constantly high query loads
- Varying update rates (mostly new metadata for assets, but some amount of retraction)

Characteristics of the use-case

- RDF database requirements:
  - Inference (rdfs + owl:TransitiveProperty, owl:sameAs, owl:equivalentClass,...) either forward or backward chaining
  - Geo-spatial constraints
  - Some form of ranking (structural interrogation)
  - Full-text search, e.g. Lucene
  - Provenance

What should be measured?

- Concurrent query and update performance with required level of inference
  - Probably the hardest facet of the use-case
  - Accuracy or (eventual) consistency?
  - Deliberately make it hard for forward and/or backward reasoning strategies (if it is useful)
- Query during (bulk) loading
- ACID – requirement or something to measure?

What should be measured?

- Full-text search queries (precision/recall)
- Geo?
- Ranking?
- Data curation tasks?

What should be measured?

- Enterprise functions:
  - Effect of backup on performance
  - Effect of failure (fail-over) on performance
  - How to measure resilience?
- Question:
  - When is a feature (e.g. geo) a requirement for compliance with the benchmark, an optional feature (yes/no) or something that can be measured?

Outline of the benchmark

- Define ontology
- Specify reference datasets or generate data?
- Parameterised query and update mixes executed concurrently from clients, e.g.
  - 20 aggregation agents (query only)
  - 5 annotation agents (modify metadata)

Outline of the benchmark

- Cost metric?
- Single output metric
  - something like (X.qph + Y.uph + Z.....) /cost
- 'Enterprise features'
  - Which features?
  - How to measure the performance (drop) of a backup?

Questions/suggestions?