

Question Answering Using XML-Tagged Documents

Ken Litkowski
CL Research
ken@clres.com
<http://www.clres.com>
<http://www.clres.com/trec11/index.html>

CL Research XML QA System

- Full text processing of TREC top 20 documents
 - ▶ Sentence splitting, tokenization, full sentence parsing
 - ▶ Unitizing the text into “discourse entities” (generally noun phrases), verbs, and prepositions (semantic relations)
- Discourse processing of parse output sentence by sentence
 - ▶ Segmentation of discourse “events” using discourse and parse output markers
 - ▶ Update of anaphora resolution data structures
 - ▶ Analysis of each discourse entity
- Creation of XML tagged version of text
- Question answering
 - ▶ Analyzing question into search statement (XPath expression)
 - ▶ Automatic selection of XML tree nodes as answers, using XML Analyzer

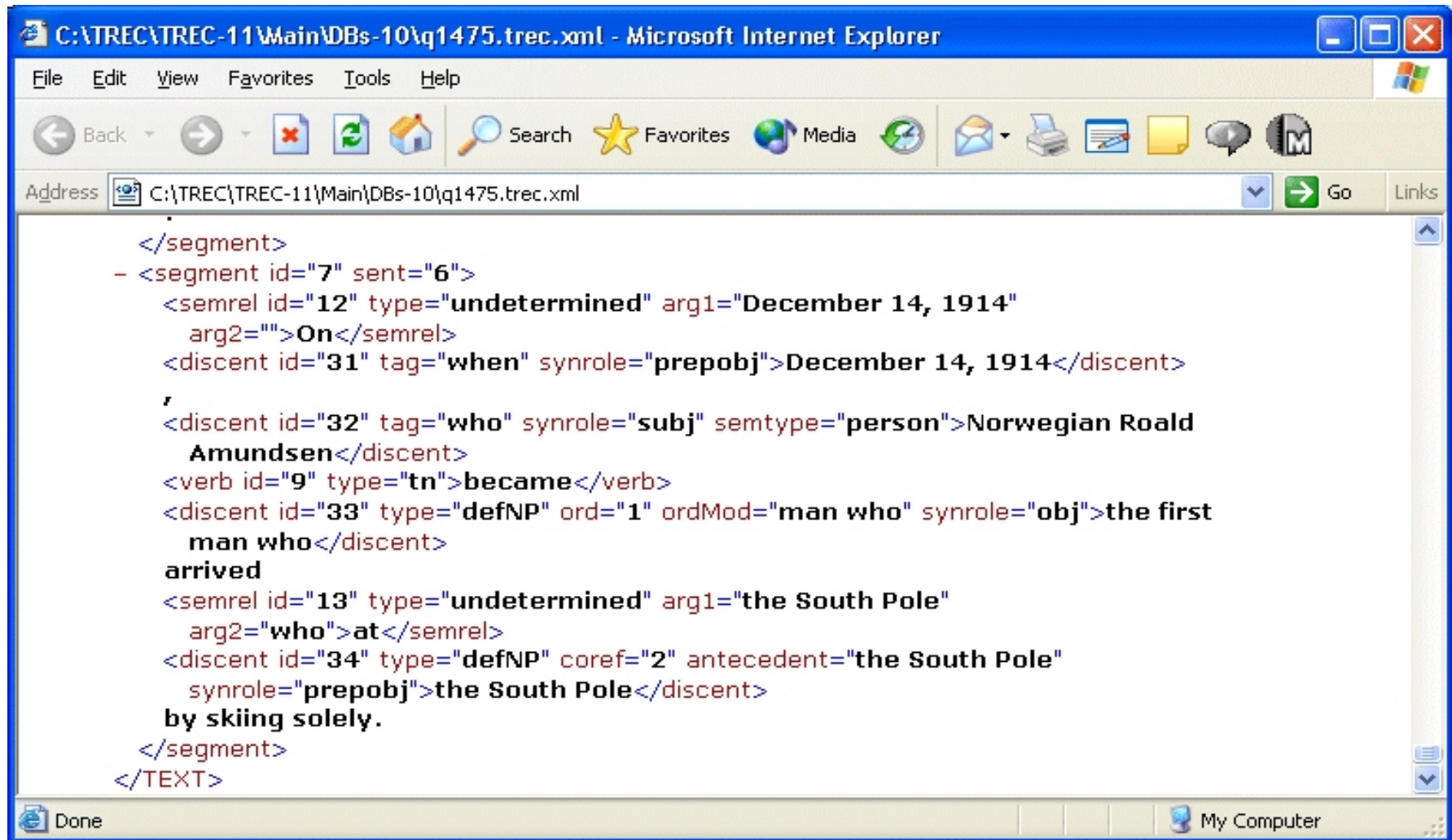
Analysis of Discourse Entities

- Capturing numbers, adjective sequences, possessives (subjected to the anaphora resolution module), genitive determiners (made into separate discourse entities), leading noun sequences, ordinals, and time phrases
- Coreference, anaphora, and definite NP identification and antecedent resolution
- Semantic typing (WordNet or integrated Macquarie machine-readable dictionary)
- Syntactic and semantic relation analysis (including preposition disambiguation) among discourse entities

XML-Tagging the Text

- Begin with list of “events”, discourse entities, verbs, and prepositions,
 - ▶ each with unique identifier and segment number
 - ▶ syntactic, semantic, and relational attributes appropriate to each type
 - Segment: subsegments, parent (if a subsegment), type
 - Discourse entity: syntactic role (subject, object, prepositional object), syntactic characteristics (number, gender, and person), type (anaphor, definite or indefinite), semantic type (such as person, location, or organization), coreferent (if any), number or an ordinal (if present), antecedent (if appropriate), and a tag indicating the type of question it may answer (such as who, when, where, how many, and how much)
 - Verb: subcategorization type, arguments, base form, and grammatical role (for adjectival use)
 - Preposition: type of semantic relation, and arguments (prepositional object and attachment point)
- Creating XML-tagged text (with attributes and values from lists)
 - ▶ Begin with untagged text and proceeding by sentence position
 - ▶ Add beginning **segment** tag
 - ▶ Tagging discourse entities (**discent**), verbs (**verb**), prepositions (**semrel**), and subsegments (**segment**)

CL RESEARCH XML Tagging



The screenshot shows a Microsoft Internet Explorer window displaying XML markup for a text segment. The address bar shows the file path: C:\TREC\TREC-11\Main\DBs-10\q1475.trec.xml. The XML content is as follows:

```
</segment>
- <segment id="7" sent="6">
  <semrel id="12" type="undetermined" arg1="December 14, 1914"
    arg2="">On</semrel>
  <discent id="31" tag="when" synrole="prepopbj">December 14, 1914</discent>
  ,
  <discent id="32" tag="who" synrole="subj" semtype="person">Norwegian Roald
    Amundsen</discent>
  <verb id="9" type="tn">became</verb>
  <discent id="33" type="defNP" ord="1" ordMod="man who" synrole="obj">the first
    man who</discent>
  arrived
  <semrel id="13" type="undetermined" arg1="the South Pole"
    arg2="who">at</semrel>
  <discent id="34" type="defNP" coref="2" antecedent="the South Pole"
    synrole="prepopbj">the South Pole</discent>
  by skiing solely.
</segment>
</TEXT>
```

CL RESEARCH XML Analyzer

XML Document: C:\TREC\TREC-11\Main\DBs-10\q1475.trec.xml

XML Elements: DOCNO

Analysis Unit: DOCNO

XPath Selector: //segment[contains(.,'first') and contai...

Analysis Units:

- XIE19980102.0127
- NYT19991206.0114
- APW20000414.0178
- NYT19981103.0190
- XIE19981221.0153

discent	Document	Sentence
Norwegian Roald Amundsen	XIE19981221.0153, 6	On December 14, 1914, Norwegian Roald Amundsen became the first man who arrived at the South Pole by skiing solely.

Attr. Name | Attr. Value

id	32
tag	who
synrole	subj
semtype	person

Number of Elements: 1

Save Results to File

//segment[contains(.,'first') and contains(.,'South') and contains(.,'Pole')]/discent[@semtype='person']

Creation of XPath Expressions

- Question is answered by selecting node(s) from the XML tree using an XPath expression
 - ▶ Expression is very similar to an ordinary query posed to a search engine
 - All Boolean analogs are available
 - Can be a simple string search (regular expressions in next version of XPath Language)
 - ▶ XML Tree structure, with attributes and values on nodes, allows search for “structure” in text
- Hand-crafted for TREC-11 to develop patterns for use as basis for automation
 - ▶ Consists of **segment** searches to match strings in the question to identify potential sentences
 - ▶ Further localization to discourse entities (**discent**) nodes, using attributes and values, to obtain actual answers
 - ▶ In some cases, more complex examination of surrounding nodes (discourse entities, verbs, and prepositions) to navigate the document tree

What Examples

- (1514) What is Canada's most populous city?
 - //segment[contains(.,'most populous')]/discent[contains(.,'most populous')]/following-sibling::semrel[.='of']/following-sibling::discent[position() = 1] **Toronto**
- (1516) What does CPR stand for?
 - //segment[contains(.,'CPR') and (contains(.,'(') or contains(.,'or'))]/discent[contains(.,'CPR')]/following-sibling::discent[position() = 1 and starts-with(.,'c')] **cardiopulmonary resuscitation**
- (1525) What university did Thomas Jefferson found?
 - //segment[(contains(.,'university') or contains(.,'University')) and contains(.,'Jefferson') and contains(.,'found')]/discent[contains(.,'University')] **the University of Virginia**
- (1532) What is the literacy rate in Cuba?
 - //segment[contains(.,'literacy') and contains(.,'rate') and contains(.,'Cuba')]/discent[@tag='howmany' or @tag='howmuch'] **96 percent**
- (1544) What is the most populated country in the world?
 - //segment[contains(.,'most') and contains(.,'world') and contains(.,'populous')]/discent[contains(.,'populous')]/preceding-sibling::verb[.='is']/preceding-sibling::discent[position()=1] **China**

When Examples

- (1502) What year was President Kennedy killed?
 - //segment[(contains(.,'kill') or contains(.,'assassinate') or contains(.,'murder')) and contains(.,'Kennedy') and contains(.,'resident')]//verb[contains(.,'kill') or contains(.,'assassinate') or contains(.,'murder')]//following-sibling::discent[(@tag='when' or @tag='num') and starts-with(.,'1')] **1963**
- (1518) What year did Marco Polo travel to Asia?
 - //segment[contains(.,'Marco Polo') and contains(.,'travel') and (contains(.,'China') or contains(.,'Japan'))]//discent[@tag='when'] **more than 700 years ago, 1275**
- (1555) When was the Tet offensive in Vietnam?
 - //segment[contains(.,'Tet') and contains(.,'offensive')]//discent[@tag='when' or @tag='num'] **1968**
- (1564) When did Led Zeppelin appear on BBC?
 - //segment[contains(.,'Zeppelin') and contains(.,'BBC')]//discent[@tag='when' or @tag='num' or @tag='howmany'] **1969**
- (1569) When did the Vietnam War end?
 - //segment[contains(.,'Vietnam War') and contains(.,'end')]//discent[@tag='when'] **25 years ago**

Where Examples

- (1498) What school did Emmitt Smith go to?
 - //segment[contains(.,'Emmitt') and contains(.,'Smith') and contains(.,'chool')]/discent[@tag='where' and contains(.,'chool')] **Escambia High School**
- (1500) Where is Georgetown University?
 - //segment[contains(.,'Georgetown University')]/discent[@tag='where' and not(contains(.,'Georgetown'))] **Washington**
- (1519) Where was Hans Christian Anderson born?
 - //segment[contains(.,'Hans') and contains(.,'Christian') and contains(.,'Anderson')]/discent[@tag='where' or @synrole='gendet'] **Denmark**
- (1528) Where did Kublai Khan live?
 - //segment[contains(.,'Kublai Khan')]/discent[@tag='where' or @tag='location'] **Beijing**
- (1542) Where is Hill Air Force Base?
 - //segment[contains(.,'Hill') and contains(.,'Base')]/discent[@tag='where' or @semtpe='location'] **Utah**

TREC-11 XML QA Results

Comparison with Official CL Research Results

**Table 2. Mean Reciprocal Ranks
for 75 Question Sample**

Sample	CWS
Official (first answer only)	0.160
XML-based (first answer only)	0.800
Uofficial (top 5 answers)	0.243
XML-based (top 5 answers)	0.828

Table 1. Confidence-Weighted Scores for Question Samples

Sample	CWS
Official (100)	0.192
XML-based (100)	0.816
Official (75)	0.266
XML-based (75)	0.869

Macquarie Tagging Project

- Macmillan Encyclopedia of Australia's Aboriginal Peoples
 - ▶ 548 articles, intended for online research use by middle school students
 - ▶ support for question answering and links to interactive map component of MacquarieNet
- Serves as the basis for CL Research's XML tagging
 - ▶ Existing SGML-tagged text parsed and processed to add discourse analysis XML tagging (700k --> 3.2 MB)
 - ▶ XML file post-processed to add gazetteer coordinates (latitude and longitude) for discourse entities with @tag='where' and capitalized
- Using "Australia's Heritage" (hand-tagged with coordinates) as benchmark
 - ▶ 564 chapters, 2633 pages, 13870 tagged places
 - ▶ Based on 5% sample, recall of 0.648 and precision of 0.802
- Use of XML Analyzer with benchmark allows for quick focus on problem areas

Discourse Marker Project

- Objective: Develop algorithms for assigning temporal relations among discourse “events”
- Method
 - ▶ Select sentences (e.g., beginning with ‘After’) from XML-tagged corpora (e.g., RST corpus or TREC documents) using XML Analyzer
 - ▶ Process sentences with POS-tagger to produce XML compliant output
 - ▶ Use XML Analyzer to look for patterns in POS-tagged output
 - ▶ Develop algorithms for assigning event-relation attributes
- XML Analyzer Planned Extension
 - ▶ Implement “stylesheet” capability (XSLT) for automatic assignment of event-relation attributes to XML file, enabling retention and augmentation of XML-compliant POS file

XML-Tagging Research Issues

- Amount of analysis necessary (for QA or other applications)
 - ▶ Parsing: use full parse output as initial tags?
 - ▶ Word-sense disambiguation: determine amount of knowledge representation in each sense?
 - ▶ Anaphora resolution: examine initial tagging to review antecedents?
 - ▶ Discourse analysis: incorporate text mining and lexical chaining?
 - ▶ Semantic relations: further post-processing of results to establish arguments?
 - ▶ Results for QA: factoid questions do not seem to require elaborate analysis
- Tagging alternatives
 - ▶ What should be the tags, attributes, and values (e.g., for inclusion in XML-aware indexing)
 - ▶ How much semantic typing (incorporation of “lexical” resources)
 - ▶ How to group information: what sentence parts should be grouped together and which modifiers should be separated or put into attributes of a discourse entity

Potential Applications of XML Tagging

- Examination of linguistic phenomena
 - ▶ XML Analyzer has been extended to handle arbitrary tagged data
 - ▶ Part-of-speech taggers, chunkers, word-sense taggers, and discourse taggers
- Information extraction
 - ▶ XML Analyzer immediately capable, depends only on processing of text to include worthwhile attributes
- Text mining
 - ▶ XML Analyzer can examine patterns and can be extended to look for particular kinds of relations
- Novelty
 - ▶ Low-level XML analysis will facilitate identification of novel information
- Text summarization
 - ▶ XML nodes can be extracted and used to build summaries

Conclusions

- Documents XML-tagged with syntactic, semantic, relational, and structural characteristics provide ready access to factoids supporting fact-based question answering
- Use of XML-tagged documents in QA suggests their potential usefulness in other text processing applications such information extraction, text mining, novelty detection, and text summarization