Name Date Place Extraction in unstructured text

Automatically scan machine-readable text to locate name, date, and place information

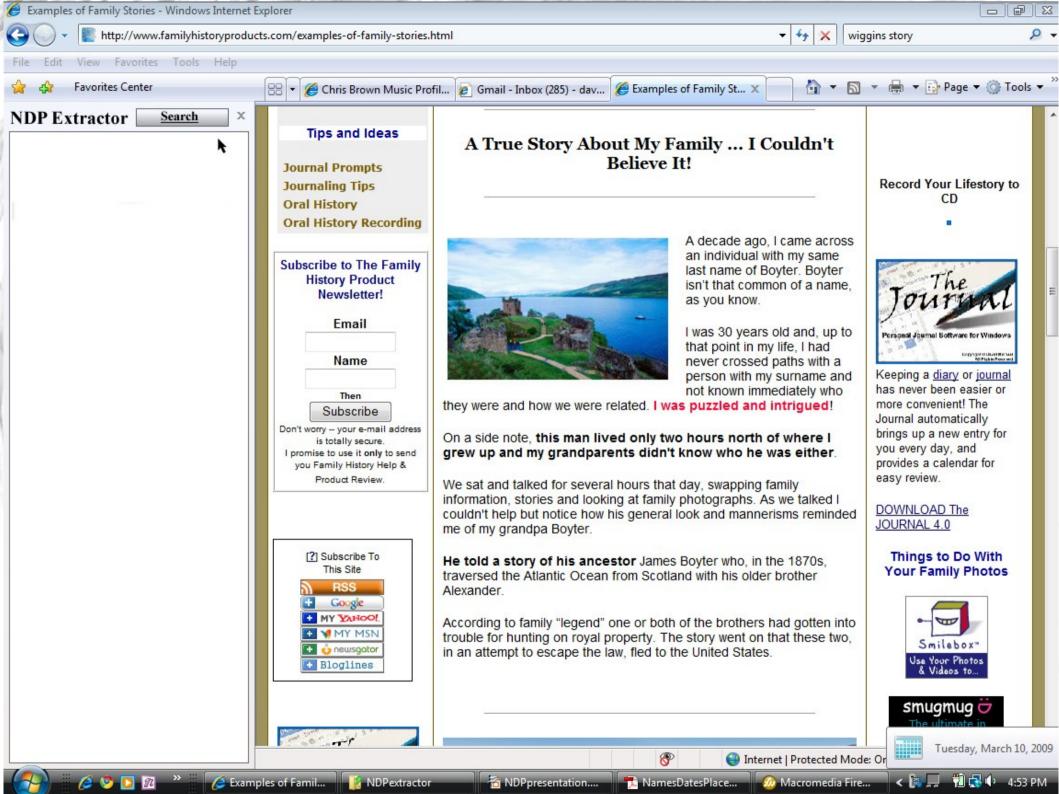
The Problem

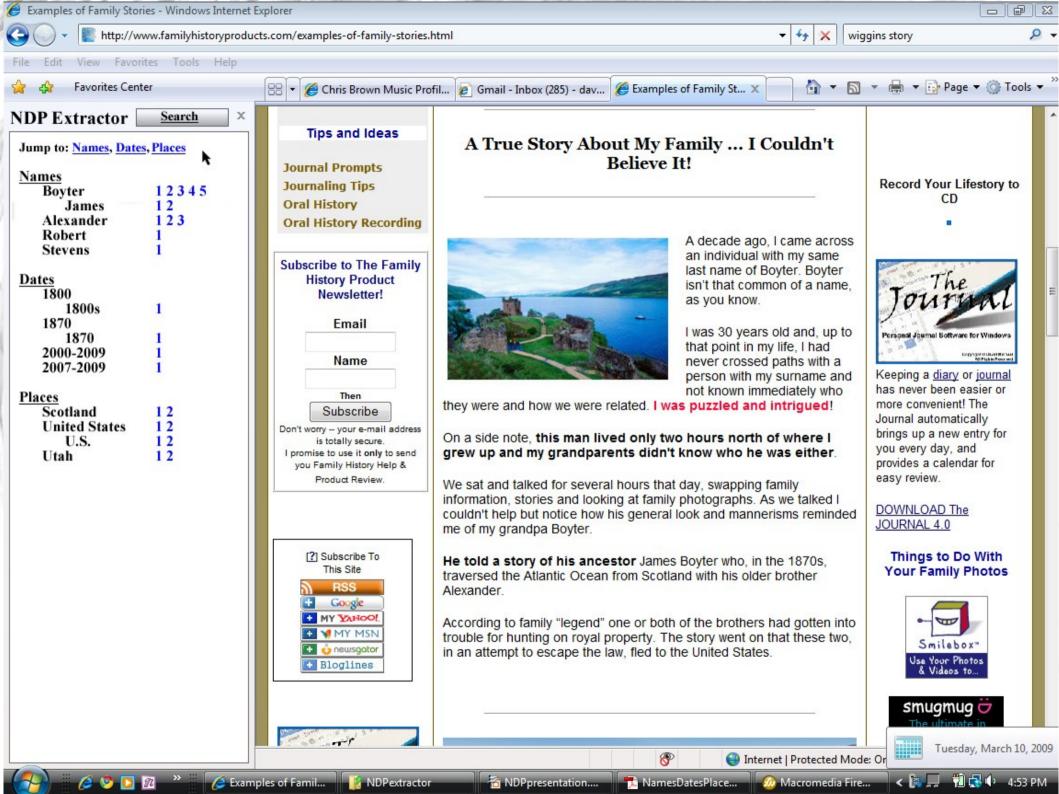
It's difficult to:

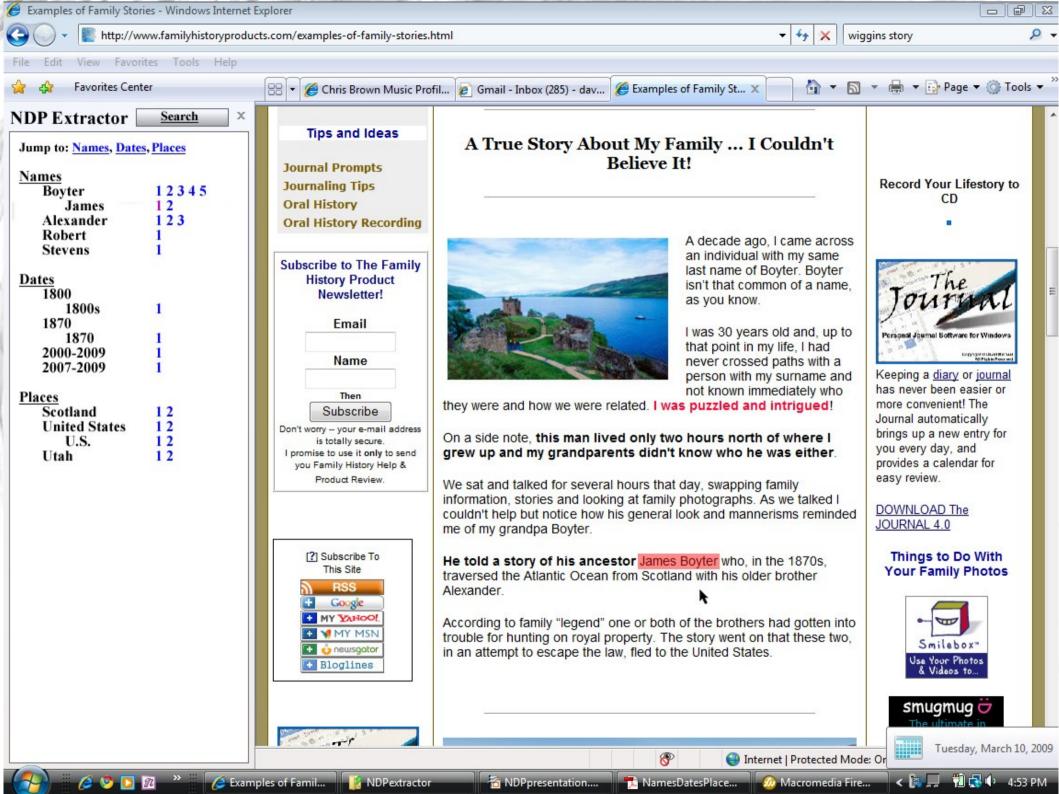
- Find pertinent information in long documents
- Make accurate queries for unknown entities
- Make queries that compensate for all variations
 - (spelling, alternate names, format)

Our Proposal

Create a tool that will find all the locations of names, dates, and places within a document.







How we plan to do it

Four step Algorithm

- 1. Convert the content to plain text.
- 2. Convert the text from a sequence of characters to a sequence of categorized tokens.
- 3. Identify the complete names, dates, and places with a lexical analyzer. (combine tokens)
- 4. Format the results.

Convert to plain text

<Cities on a Saturday are often such interesting places: full of people, full of cars, full of the hustle and bustle of modern life. And Leicester is no exception. I was born there so I can speak from personal experience. But something was different last Saturday. There were more people, more cars and much more hustle and bustle than I had ever seen or heard before.

|�d gone into town with my mates that Saturday - as we always do. We caught the same No. 149 bus from Oadby � that�s a small town south of Leicester. Nothing unusual in that. The journey was as predictable as ever � l�m so used to it. I can�t even remember getting on the bus; but I can certainly remember getting off�

Cities on a Saturday are often such interesting places: full of people, full of cars, full of the hustle and bustle of modern life. And Leicester is no exception. I was born there so I can speak from personal experience. But something was different last Saturday. There were more people, more cars and much more hustle and bustle than I had ever seen or heard before.

Id gone into town with my mates that Saturday - as we always do. We caught the same No. 149 bus from Oadby thats a small town south of Leicester. Nothing unusual in that. The journey was as predictable as ever Im so used to it. I cant even remember getting on the bus; but I can certainly remember getting off

Tokenize and Categorize

- Divide the text into organizable pieces
 - Tokenize the input on white space and punctuation
- <u>Identify strings of characters as simple tokens</u> classified as parts of names, dates, or places
 - Use a Name Authority to determine parts of names
 - Use a Place Authority to determine parts of places
 - Use research done by Robert Lyon to identify dates

Lexically analyze

Place → city PlacesFollowingCity CountyStateCountry StateCountry country

PlacesFollowingCity \rightarrow PossibleComma CountyOrStateOrCountry | ϵ CountyOrStateOrCountry → CountyStateCountry | StateCountry | country → county County Designator Places Following County CountyStateCountry

StateCountry

Country

→ Left Paren County OrParish Right Paren | Country OrParish **CountyDesignator**

CountyOrParish → countyWord | parishWord

PlacesFollowingCounty \rightarrow PossibleComma StateOrCountry | ϵ

StateOrCountry → StateCountry | country

 \rightarrow state StateDesignator PlacesFollowingState StateCountry → Left Paren stateWord Right Paren | stateWord **StateDesignator**

→ PossibleComma country **PlacesFollowingState**

PossibleComma $\rightarrow Comma \mid \varepsilon$

Create completed name, date, and place results by combining our categorized tokens using these regular grammars

Name → title SimpleName | SimpleName

→ NameWithGivenNames | LastName **SimpleName**

NameWithGivenNames → GivenNames LastNameAndSuffix

LastNameAndSuffix \rightarrow LastName GenerationId | ϵ

→ givenName OptionalPeriod RemainingGivenNames **GivenNames**

RemainingGivenNames \rightarrow GivenNames | ε

→ surname RemainingSurnames

→ dash surname | SurnameList

→ surname SurnameList | ε

 \rightarrow suffix $\mid \varepsilon \mid$

GenerationId OptionalPeriod \rightarrow period | ε

LastName

SurnameList

RemainingSurnames

Date Identification

September 1, 1997

1 September 1997

Sept. 1, 1997

Sept 1, 1997

Sept 1, '97

Sept 1

September 1997

09/01/1997

September 1st 1997

1st of September 1997

after Sept 1, 1997

- Original

- Alternative ordering

- Month abbreviation

- Alternate punctuation

- Year abbreviation

- Assumed year

- No day of the month

- Numeric format

- Ordinal day of the month

- Internal preposition

- Altering preposition

[Lyon2000] Lyon, Robert W., *Identification of temporal phrases in natural language*, Masters Thesis, Brigham Young University. Dept. of Computer Science, 2000

Format results

Jump to: Names, Dates, Places

Names

Boyter 12345
James 12
Alexander 123
Robert 1
Stevens 1

Dates

1800 1800s 1 1870 1870 1 2000-2009 1 2007-2009 1

Places

 Scotland
 1 2

 United States
 1 2

 U.S.
 1 2

 Utah
 1 2







Summer '09

- Recruit BYU CS students for capstone
- Further research and design of the project
- Find/Develop solutions for name and place authority requirements

Fall Semester '09

 Implement CS598R capstone project to develop the NDPextractor

December '09

Finish CS598R capstone project

Questions?

