The "Smart" Social Media Engine

Discover Actionable Data in Web Posts and in Social Media
– Using Advanced Conversational Technology

Identify Consumer Sentiment and Reader Reactions with Multilevel Language Analysis — Semantics, Sequence Packages, and Conversation Patterns — Integrated with Structured Data

Linguistic Technology Systems
POC: Amy Neustein, Ph.D.
Founder and CEO
amy.neustein@verizon.net
201-224-5096

Lead Software Architect: Nathaniel Christen

A Business Case for a "Smart" Social Media Engine

Features of a "Smart" Engine

- **Marks conversational content as available data so that it can be converted to actionable data by marketing and research departments.**
- **Finds the hidden attributes of products (as well as implicit reader sentiments useful to pollsters and others) buried in conversational threads, adding to the corpora of structured data extracted from web postings.**

How Smart Engine Prepares for Web 3.0

- Ensures that content is available in multiple formats, including XML, RDFa (RDF in Attributes, a kind of Semantically Annotated HTML), and new representation languages suitable for analyzing Semantic Web data. As "Web 3.0" evolves, we are diligently exploring academic and technology projects whose mission is to change how we define web content. These new representations will better serve algorithms and tools designed to find patterns within Web 3.0 and the Semantic Web.
- We allow anyone to contribute code libraries matching those formats. That is, instead of having to parse XML or RDFa, libraries will allow researchers to grab blog data as an object or data value in their favorite programming language.[a]

[a] These libraries could also be provided by the community (including bloggers themselves), either open-source or licenced.

Sequence Package Analysis

Combining Natural Language Processing and Conversation Analysis:
for Next Generation Language and Conversation Tools

What Does Sequence Package Analysis (SPA) Do?

- Provides an Annotation System for Unconstrained, Open-ended Digital Text
- Provides an Annotation System for Online Text that has Few Keywords

How does SPA Work?

- Using a corpus of annotated training data comprised of a unique set of feature extractions, SPA provides the algorithms for machines to understand the richness of human language with its attendant ambiguities, convolutions, and ellipses.
- The algorithms are based on detecting sequence packages: a series of related context-free grammatical units and associated prosodic/punctuation features, discretely packaged as a sequence of online conversational interaction, for which there is a corresponding list of interchangeable terminals: words, phrases, or a whole utterance.
- By marking sequence package boundaries and specifying package properties, SPA gives the software downstream the contextual indicia — the precise location points in the flow of interactive online dialog, signifying the different conversational activities of the web and social media post — needed to interpret the rest of the data stream reliably.

The Utility of SPA for Social Media Analysis

SPA Grammar and Database Technology

- Pinpointing Emotions/Sentiments in Social Media and other Web Posts by matching feature extractions — drawn from an SPA-designed table of parsing structures defining each sequence package — against the wide range of sentiments found in digital posts.
- SPA-designed grammars can identify complicated emotions that build up in posts, by recognizing the incremental design of complex grammatical components from more elemental units. For example, a "very angry complaint" is an accretion of more elemental SPA-defined parsing structures, such as assertions, exaggerations, and declarations.[a]
- The "Smart" Social Media Engine stores content in a database customized to support Sequence Package Analysis. Analyses on this content are therefore projected to be especially fast and accurate.
- The "Conversational Query Language" (C-QL) is a new language for databases with extensive natural language data (please see next slide).

[a] This is similar to the way a BNF table is used to incrementally denote syntactic parts of natural language grammars.

C-QL Information

Conversational Query Language

A high level query language that integrates **Sequence Package Analysis** algorithms with **Structured Information** (who/what/when/where)

Using C-QL

- **C-QL-F is a Formal Query Language. C-QL Queries find signatures of**
  - Conversation patterns → often unconscious; detected by using Sequence Package Analysis
    (example ≡ subtle insertion of "negative" restaurant attributes
    in a thread of very positive review postings.)
  - Natural Language patterns (example ≡ word meanings, parts of speech)
  - Structured Data patterns
    (annotating Object-Oriented data models to connect data fields with words and phrases)
- **C-QL-F extends existing technology**
  - Includes a modern implementation of "Stack-Based Query Language" (SBQL)
    (for searching complex Application and Object-Oriented data)
  - Partial implementation of Functor Query Lanuage
    (a mathematical framework used in Categorical Informatics)
  - Secure Access Layer based on the E programming language for maximum security
- **C-QL-N is a way to write C-QL-F queries using Natural Language**
  (extensible to support multiple languages and alphabets)
- **"Query as Conversation": An interface designed to show and review query results in conversational ways, modeling user/data interaction via Sequence Package Analysis**

The C-QL Database Stack

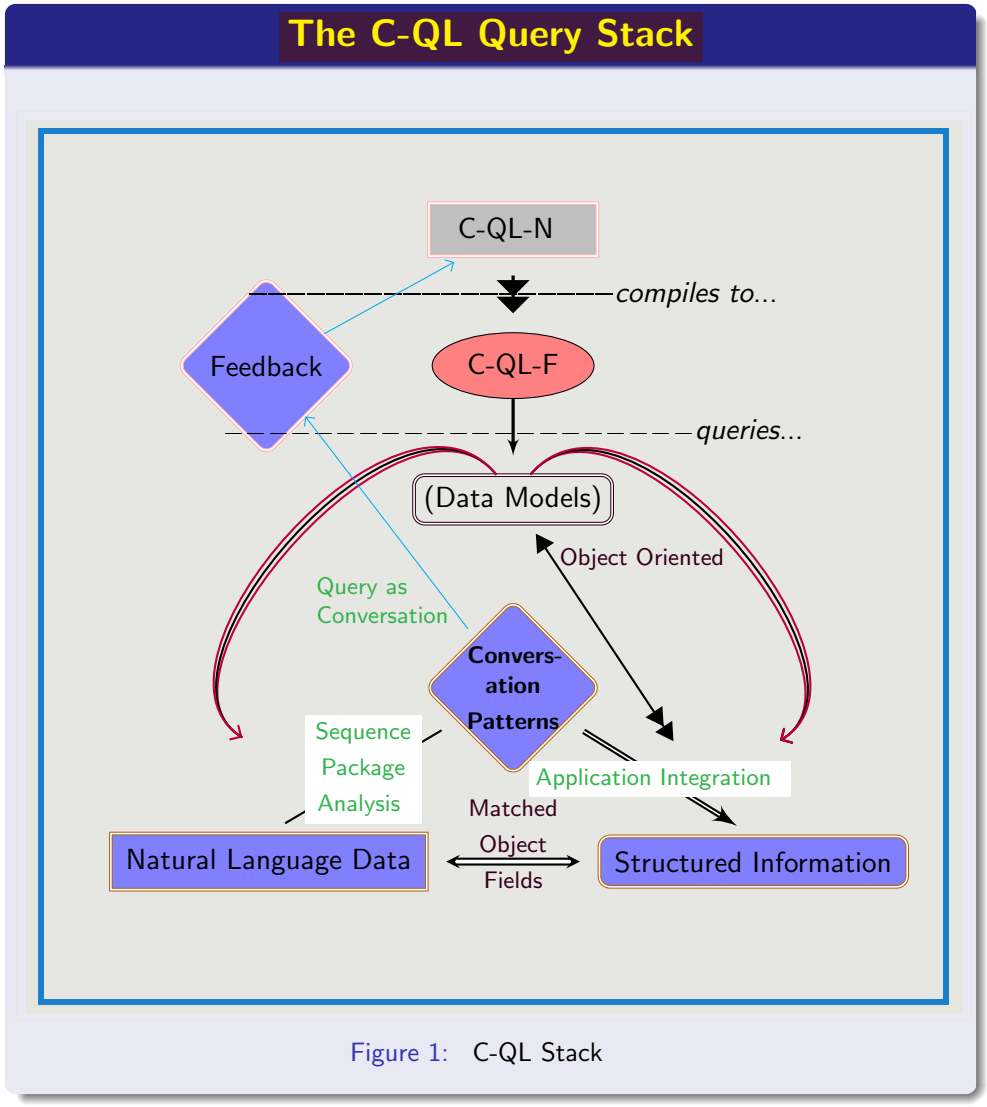


Figure 1: C-QL Stack

Conversational Query Language

Figure 1 (left) outlines major components for storing C-QL-aware data and executing C-QL Queries. Note:

- Components modeled as Natural Language are shown as Rectangles (with straight corners).
- Components modeled as Conversations are shown as Diamonds.
- Components modeled as Structured Data are shown as rectangles with rounded corners.

- **C-QL-F: A Formal Query Language integrating Language, Conversation, and Structured Data.**
- **C-QL-N: A way to write C-QL-F queries using Natural Language.**
- **"Query as Conversation": A way to use C-QL Databases for a more streamlined UX (User Experience), inspired by Natural Language.**

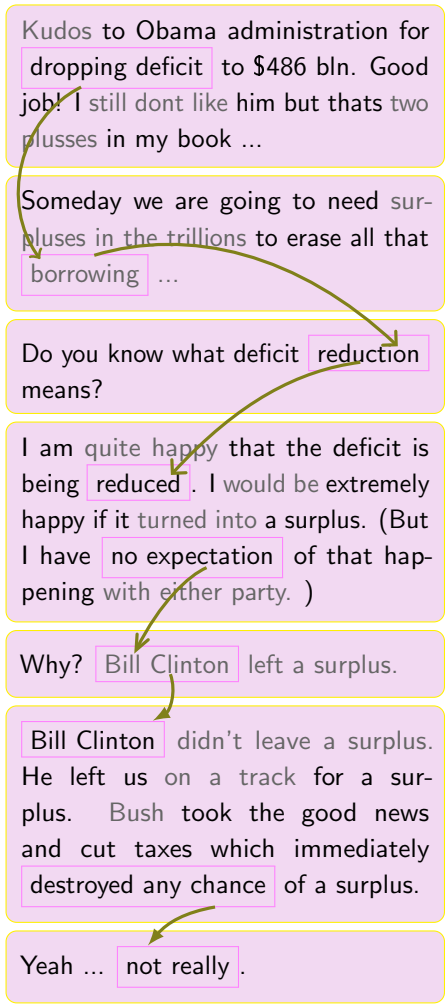
Finding Conversation Threads

Conversational Indicators of Sentiment

Writers' opinions are a complex mixture of conversing with each other, and responding to a post or news article which starts a new conversation. Different writers add different perspectives, often introducing new ideas to complement, not directly challenge, what others are saying. This give-and-take provides opinions which do not line up rigidly into "pro" and "con".

A dialog initiated by a CNN news piece, sampled at left, showed posts which supported Obama Administration policies without necessarily endorsing the president personally. The posts contained some very interesting conversations that illustrated different levels and kinds of support or opposition to Obama.

A visual graph demonstrating this spectrum, showing Obama Administration and Personal support as two distinct axes, is reproduced from an interactive 3d display and shown on the next slide. This analysis identified "hot threads" which spurred further dialog (colored red in the display).



Conflicting Opinions? *Analysis*: Contrasting sentiment, Obama personally vs. adminstration policies

Less Positive. *Analysis*: Responds to the first post by arguing that deficit *reduction* is not enough

Sarcastic challenge to prior writer's credibility. *Analysis*: implicitly supports the original post

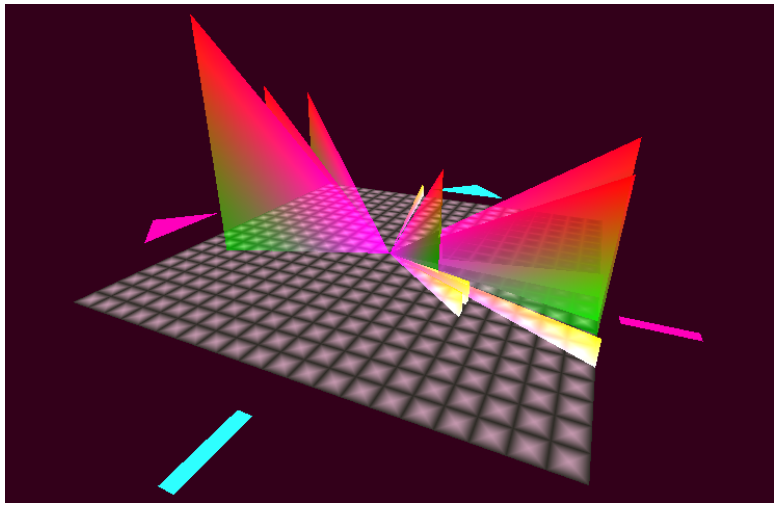
Supports original comment, with reservations. *Analysis*: This writer implies that while the Obama Administration policies have not eliminated all problems, they have made the reality better than it would be otherwise.

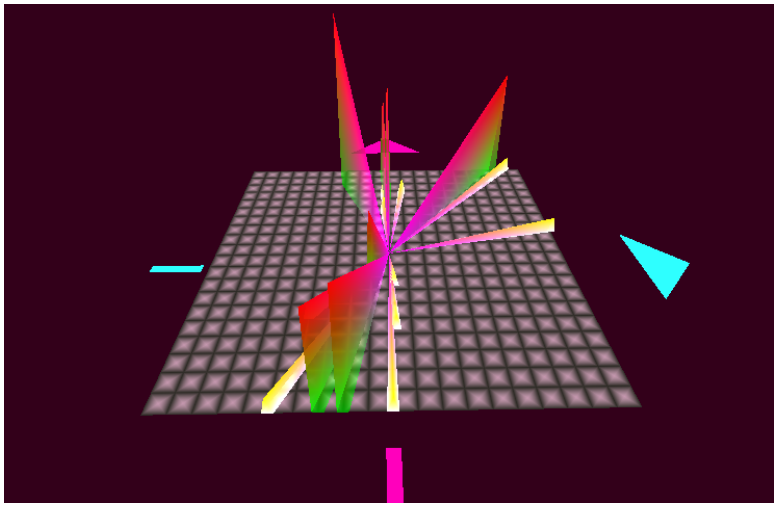
These posts generally support the first comment but from different angles. *Analysis*: Responding directly to the then most recent comment, they add perspective rather than a direct challenge.

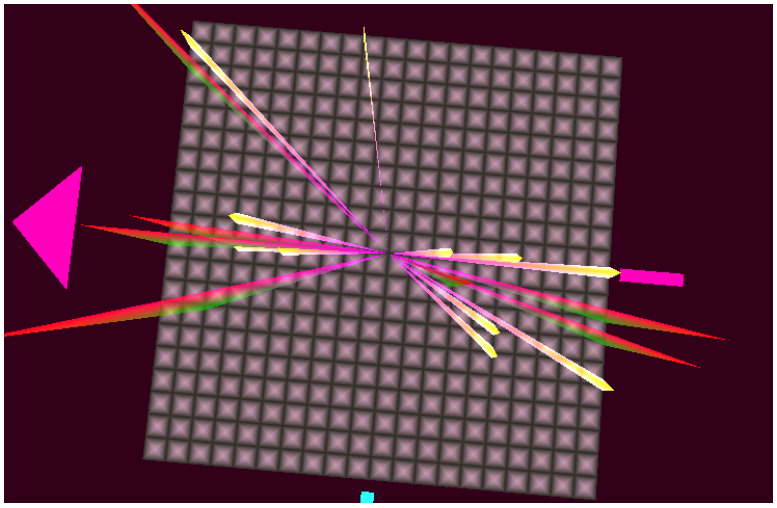
Sarcastic, direct challenge. *Analysis*: Ellipses and idiomatic "not really" highlight the challenge, and align this writer in opposition to the original post, which (despite having added extra perspective) the previous comment implicitly endorsed.

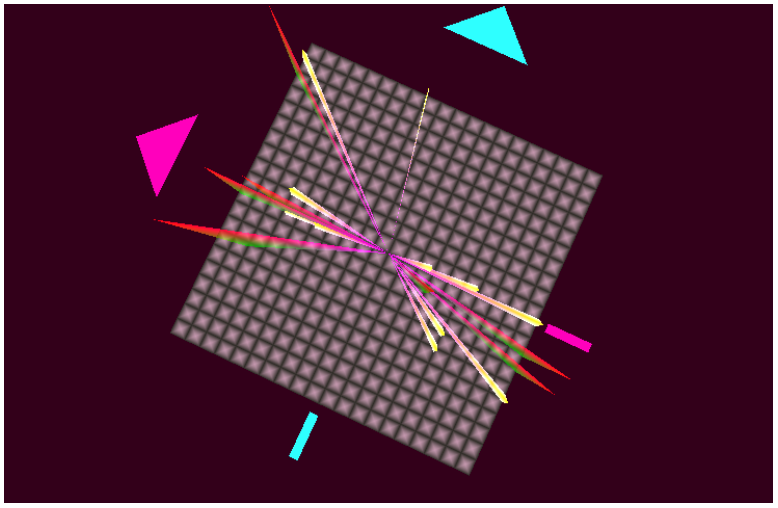
Visualizing Patterns

Readily available conversation data makes it easy for applications to integrate such data, including providing useful, innovative visual representations. These graphics show a comparison of posts, some excerpted on the previous slide, reflecting different sentiment about Obama personally or policy-wise. Axes show distribution in multiple directions — not every Obama supporter fully endorses his policies, nor vice-versa — and also how forcefully these opinions are espoused (measuring this from different markers in language and dialogic interaction). The figure at top-left shows a centered view, while others highlight patterns in distribution. The **magenta**-colored axis shows strength of endorsement or criticism of Obama Administration policies, while the **cyan**-colored axis applies to Obama personally.









Software Tools for Implementing the "Smart" Engine

Including both Open-Source and Proprietary tool sets — which can be used or licensed in applications and web sites:

Open-Source Exchange Tools

If one prefers a standard format for storing or presenting text-related and/or web content, to lay a foundation for Conversation Analysis, one can use open-source libraries to read, create, and distribute text in **Conversational Standard Text Data (CSTD)** format. This format is optimized for observing dialogic and data-integration patterns, whether these are manually annotated or identified by computer. CSTD annotation can capture "basic" concepts (like names and places) and conventional Semantic Web data, as well as more complex conversation patterns. While time-consuming, manually adding dialog annotations can be useful in some settings, like closely studying opinions voiced about specific positions taken by political candidates. Even without annotations for dialog, supporting CSTD format can be useful for sharing data through text. In addition, CSTD markup can be integrated with visual languages for conversations, like the new CAVE icon system. SPA Tools will create CSTD, so users licensing SPA software as well as those developing hand-annotated CSTD can share text and observations.

Commercial "Smart" Engine Tools

Customized for Enterprise and Research on top of content generated by the "Smart" Social Media Engine and Open-Source tools

Sequence Package Analysis Tools

Large corpora can be made available for conversation analysis by using SPA Grammars to find sequence packages in texts, including social media, blogs, and online comments. SPA Tools use sophisticated algorithms and machine learning to find both large-scale and fine-grained patterns: from word and sentence boundaries to large-scale dialogic interactions and "Big Data". Linguistic Technology Systems will develop custom software and grammars targeted to content — words, phrases, ideas, and user actions — most relevant to the enterprise.

Language and Data Integration Tools

Running and supporting SPA requires powerful Natural Language and Data Management tools, such as multi-language lexical scanners and identifiers at word, phrase, and sentence level, and solutions for interconnecting NLP and Database software. These solutions can also be personalized for one's technological and operational needs.

Deliverables

We have designed a two-track development strategy with parallel implementation of (1) A "Smart" Social Media Engine and (2) the C-QL query language and algorithms. Specifically, while we implement and mature our C-QL technology, we emphasize (1) Mining product/service reviews for critical business intelligence feedback for marketers and developers; (2) Assessing reader comments for public opinion feedback for pollsters, campaign organizers and policy analysts; and (3) Developing a conversational search of web postings on mobile devices, by bringing first-hand user experiences — the "big data" of web postings — directly to the consumer (more than 50% of web search is now done on a mobile device).

Six Quarters starting mid-2015

2015 – 3rd & 4th Quarters

- Establish a User Experience model and overall navigation and page layout for the "Smart" engine.
- Integration Framework for *Natural Language* and *Structured Information* Layers on C-QL aware databases.
- Completion of "Smart" Engine database backends and testing using a command-line interface.
- Extending the Integration Framework to interrelate *Conversation Pattern Observations* with *Actionable Structured Information* based on their shared *Natural Language Semantic Context*.

2016 – 1st & 2nd Quarters

- Completion of "Smart" Engine web front ends and User Experience models for mobile platforms.
- Formal Specification, Parsers, and Intermediate Representation for C-QL-F as a formal query language.
- Prototype mobile front ends, with summaries of comment threads, product reviews, and news stories, useful with smaller screen-size and distracting environments (like walking, commuting, or sitting in a restaurant).

2016 – 3rd & 4th Quarters

- Full Implementation of C-QL-F, database engines, and analytic suites for content on the "Smart" Social engine.
- Finalize extensions of the analytic framework to social networks beyond the "Smart" Social engine.
- Implementation of C-QL-N and the "Queries as Conversation" User Experience model and interface pattern.