A Large-Scale System for Annotating and Querying Quotations in News Feeds

Jisheng Liang
Evri Inc.
Seattle, WA, United States
jisheng@evri.com

Navdeep Dhillon
Evri Inc.
Seattle, WA, United States
deep@evri.com

Krzysztof Koperski
Evri Inc.
Seattle, WA, United States
kris@evri.com

ABSTRACT

In this paper, we describe a system that automatically extracts quotations from news feeds, and allows efficient retrieval of the semantically annotated quotes. APIs for real-time querying of over 10 million quotes extracted from recent news feeds are publicly available. In addition, each day we add around 60 thousand new quotes extracted from around 50 thousand news articles or blogs. We apply computational linguistic techniques such as coreference resolution, entity recognition and disambiguation to improve both precision and recall of the quote detection. We support faceted search on both speakers and entities mentioned in the quotes.

1. INTRODUCTION

We demonstrate a faceted search system for querying semantically annotated quotations, and show its advantage over the bag-of-keyword approach. We use natural language processing techniques to extract quotations from text (i.e. news articles, blogs, etc.), including identifying the speaker and the quote, as well as the context of the quote being made. We apply entity recognition to identify named entities and concepts mentioned in the quotations. We store each extracted quotation, together with its associated properties, in an inverted index that enables efficient and flexible retrieval. The system supports search for quotes in many different ways, in the form of

What did <speaker> say about <subject>?

where <speaker> and <subject> are specified by users. The <speaker> parameter could be specified as a specific entity (e.g. Obama), a facet (a type or category of entities, e.g. politician, basketball player), or anyone. And <subject> can be specified as combination of keywords, entities, and/or facets. For example,

- What did Obama say about global warming?
- What are people saying about Obama?
- What are people saying about Obama and global warming?
- What are [actors] or [movie directors] saying about Oscars?
- What did Obama say about any [basketball players] or [basketball teams], given he likes to play basketball?

Here, the term facet refers to some aspect of an entity discoverable from an ontology or taxonomy of entities. Facets typically are more finely granular characteristics of entities, often times used to categorize entities. For example, particular entities may belong to different categories such as football players, movies, cities, companies, websites, etc. In this paper, we use square brackets to indicate facets (e.g. [Football_Player], [Website]).

The facets could be organized into a hierarchical taxonomy, based on relations such as is-a, member-of, etc. Therefore, a facet can be represented as a taxonomic path, e.g. [Person/Sports/Athlete/Football_Player]. At search time, we have the flexibility to support query on a facet (e.g. [Football_Player]), or any of its parent nodes in the taxonomic path (e.g. [Athlete]), or any subpath (e.g. [Person/Sports]). We have built a large entity repository, in which the entities are associated with types and facets, as well as many other properties.

1.1 Related Work

Google's InQuotes[1] feature allows users to search for quotes made by a small selected set of politicians. Users can type in any keywords in the search box, and quotes containing the keywords would be returned. They do not allow search on the speaker itself other than from the selected set.

Daylife returns recent quotes made by a given speaker (e.g. quotes by President Obama[2]). However, they do not allow users to search for quotes on a particular subject.

Pouliquen et. al.[7] present a system called NewsExplorer[3]. The system identifies quotations from live news feeds, together with the person who made the quotation and the persons mentioned in the quotation. For each person in the system's database, the most recent quotations from and about the person are listed on this person's dedicated information page.

On the quote extraction part, the prior approaches extract quotations based on regular expression pattern matching. They do not apply linguistic and semantic analysis like we do (e.g. coreference resolution, entity disambiguation, etc.), in order to reliably identify the speaker, as well as entities and concepts being mentioned in the quotations. On the search of quotations, the prior approaches use traditional keyword search. They do not support semantic search of entities and facets as we do.

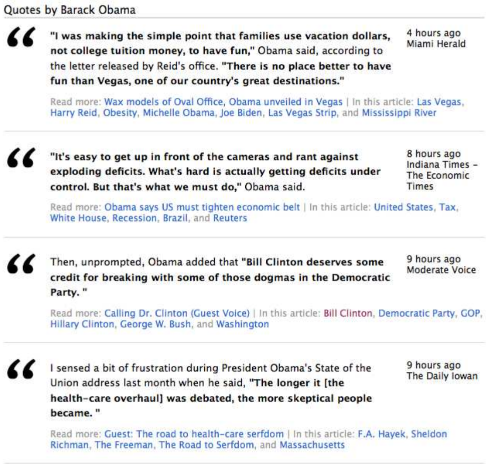
Copyright is held by the author/owner(s).
WWW2010, April 26-30, 2010, Raleigh, North Carolina.
.

[1] http://labs.google.com/inquotes
[2] http://www.daylife.com/topic/Barack_Obama/quotes
[3] http://press.jrc.it/NewsExplorer/

2. DEMONSTRATION

2.1 UI examples

Examples of the quotes we retrieve from live news feeds can be found at `http://www.evri.com`. User can type in a query in the search box, select any one of the returned entities, and land on an entity profile page. If the entity is a person type, we show retrieved quotes about the person, followed by quotes made by the person. For example, on the profile page of President Obama, we display the recent quotes about Barack Obama, as well as recent quotes made by him (See Figure 1).



Quotes by Barack Obama

"I was making the simple point that families use vacation dollars, not college tuition money, to have fun," Obama said, according to the letter released by Reid's office. **"There is no place better to have fun than Vegas, one of our country's great destinations."**

4 hours ago
Miami Herald

Read more: Wax models of Oval Office, Obama unveiled in Vegas | In this article: Las Vegas, Harry Reid, Obesity, Michelle Obama, Joe Biden, Las Vegas Strip, and Mississippi River

"It's easy to get up in front of the cameras and rant against exploding deficits. What's hard is actually getting deficits under control. But that's what we must do," Obama said.

8 hours ago
Indiana Times – The Economic Times

Read more: Obama says US must tighten economic belt | In this article: United States, Tax, White House, Recession, Brazil, and Reuters

Then, unprompted, Obama added that **"Bill Clinton deserves some credit for breaking with some of those dogmas in the Democratic Party. "**

9 hours ago
Moderate Voice

Read more: Calling Dr. Clinton (Guest Voice) | In this article: Bill Clinton, Democratic Party, GOP, Hillary Clinton, George W. Bush, and Washington

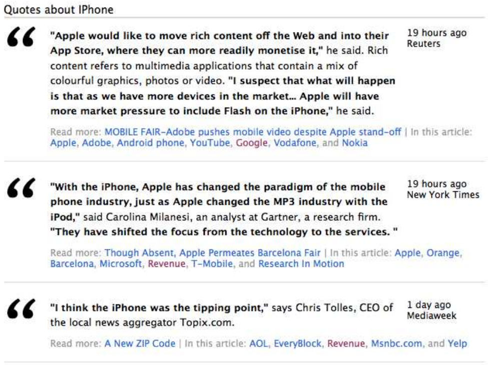
I sensed a bit of frustration during President Obama's State of the Union address last month when he said, **"The longer it [the health-care overhaul] was debated, the more skeptical people became. "**

9 hours ago
The Daily Iowan

Read more: Guest: The road to health-care serfdom | In this article: F.A. Hayek, Sheldon Richman, The Freeman, The Road to Serfdom, and Massachusetts

Figure 1: Quotes by President Obama

If the entity is not a person, we show quotes about the entity, e.g. quotes about iPhone (See Figure 2).



Quotes about IPhone

"Apple would like to move rich content off the Web and into their App Store, where they can more readily monetise it," he said. Rich content refers to multimedia applications that contain a mix of colourful graphics, photos or video. **"I suspect that what will happen is that as we have more devices in the market... Apple will have more market pressure to include Flash on the iPhone,"** he said.

19 hours ago
Reuters

Read more: MOBILE FAIR-Adobe pushes mobile video despite Apple stand-off | In this article: Apple, Adobe, Android phone, YouTube, Google, Vodafone, and Nokia

"With the iPhone, Apple has changed the paradigm of the mobile phone industry, just as Apple changed the MP3 industry with the iPod," said Carolina Milanesi, an analyst at Gartner, a research firm. **"They have shifted the focus from the technology to the services. "**

19 hours ago
New York Times

Read more: Though Absent, Apple Permeates Barcelona Fair | In this article: Apple, Orange, Barcelona, Microsoft, Revenue, T-Mobile, and Research In Motion

"I think the iPhone was the tipping point," says Chris Tolles, CEO of the local news aggregator Topix.com.

1 day ago
Mediaweek

Read more: A New ZIP Code | In this article: AOL, EveryBlock, Revenue, Msnbc.com, and Yelp

Figure 2: Quotes about iPhone

In our system, at the time of writing, we have a total of 1.6 million entities[4], with about 0.67 million as person names[5]. Besides the person type, the other major types are organization, location, product, event, concept, substance, and organism. The entities are further categorized into more than 500 facets. In addition, we have an automated process in place that detects trendy entities/concepts, and bring them into our system on an hourly basis. We surface latest quotes made by or about each of those entities.

To handle anything that is not yet included as an entity in our system, we also provide quotes about any keywords or phrases (e.g. "semantic search", "WWW 2010").

2.2 Public APIs

Our portal only demonstrates a subset of the available quotation search capability. We expose the full capability via a set of public APIs (see documentation at `http://www.evri.com/developer/rest#API-GetQuotations`). Below are a number of examples of the API usage. The retrieved quotes can be returned as either XML or JSON format[6].

- quotes by a person, `http://api.evri.com/v1/quotes?speaker=/person/barack-obama-0x16f69`
- quotes about a particular entity, e.g. iPhone `http://api.evri.com/v1/quotes/about?entityURI=/product/iphone-0x4d735`
- quotes by any entities of certain facet, e.g. quotes made by basketball players, `http://api.evri.com/v1/quotes?speaker=facet/basketball_player`
- quotes about any entities of certain facet, e.g. quotes about college football teams, `http://api.evri.com/v1/quotes/about?facet=college_football_team`
- quotes about any keyword or phrase, `http://api.evri.com/v1/quotes/about?phrase=hoyas`
- quotes made by a person about any entities of a particular facet, e.g. quotes made by David Letterman, a talk show host, about any politicians, `http://api.evri.com/v1/quotes/about?facet=politician&speaker=/person/david-letterman-0x1b480`
- quotes made by any entities of a particular facet about something, e.g. quotes made by any football players about the Super Bowl, `http://api.evri.com/v1/quotes/about?phrase=super%20bowl&speaker=facet/football_player`

3. SYSTEM OVERVIEW

The quotation extraction and search system has three components:

- Quotation extraction and attribution from text documents;
- Indexing of the extracted quotations and attributions in an inverted index;
- Efficient and flexible search of the indexed quotations.

This system is built on top of our existing infrastructure for indexing of syntactic and semantic annotation of text [4, 5, 6]. We extended the framework to support indexing and querying of quotations.

This is also related to our work on entity recognition and disambiguation [3]. By identifying entities in text, and linking them to the corresponding entries in our entity repository, we are able to support retrieval of quotes by or about entities, as well as by certain types or facets of the entities.

[4] for latest numbers, refer to API call: `http://api.evri.com/v1/entities`

[5] `http://api.evri.com/v1/entities?type=person`

[6] `http://www.json.org/`

3.1 Quotation Extraction and Attribution

We achieve highly accurate extraction of quotations by applying linguistic analysis such as sentence parsing, named entity recognition and disambiguation, and coreference resolution.

3.1.1 Linguistic and Semantic Analysis

For each text document, the processing includes the following steps:

Sentence splitting Split the document into paragraphs, and paragraphs into sentences.

Parsing For each sentence, apply linguistic parsing to assign part-of-speech tags (e.g. nouns, verbs), perform lexical analysis (e.g. detecting phrases), and determine grammatical roles (e.g. subjects, verbs, objects).

Entity recognition Apply named entity recognition to identify entities and concepts appearing in the text.

Coreference resolution Link multiple mentions of the same entity across the whole document, including resolving pronoun coreference (e.g. "he", "him", "she"), aliases and abbreviations (e.g. "Bill Gates" referred to as "Gates", "General Mills" as "GM", "Alaska Airlines" as "Alaska"), and definite-noun anaphora (e.g. "the president", "the coach"). The coreference resolution step is very important to determining quotation attributions, because very often the speaker's full name is not provided for a given quote. Instead, the writer typically uses pronouns ("he said"), partial names ("said Gates"), or definite nouns ("the president said"). Similarly, in quotations, entities are often mentioned as aliases or pronoun anaphora. Applying coreference resolution would help identify such mentions, that otherwise would be missed by keyword matching techniques.

Below is an example of coreferencce resolution. As the result, the quote is attributed to President Obama.

> I sensed a bit of frustration during President Obama's State of the Union address last month when he said, "The longer it [the health-care overhaul] was debated, the more skeptical people became."

Facet tagging To each entity, we assign its type and facet categories. For example, we tag entity "Michael Jackson" with type 'person' and facet [musician].

Disambiguation Apply entity disambiguation such that each mention of an entity is linked to an entry in our repository of entities. As the result, different mentions of an entity are all marked with a unique identifier. During search, we support search of entities by their unique identifiers, instead of using ambiguous keywords. For example, we are able to distinguish between Will Smith the actor and Will Smith the American football player who plays for the New Orleans Saints.

3.1.2 Quotation Detection

The above step marks up each sentence with syntactic and semantic annotations, which are then leveraged for extracting quotations.

1. For each verb detected in a sentence, we check if the verb belongs to a pre-determined list of verbs that can be potentially used to indicate a quotation (e.g. acknowledge, add, argue, caution, say, suggest, urge, etc.);
2. Check the occurrences and positions of quotation marks in the sentence, as well as it surrounding sentences - a long quote could span more than one sentence.
3. Determine quotation candidates based on combination of the above two factors, i.e. if there is a quotation verb and there are quotation marks nearby, we have higher confidence there is a quotation contained in this piece of text.

3.1.3 Attribution and Collapsing

We collapse each detected quotation into a triple of (speaker, verb, quote):

Speaker: the main subject of the verb, as well as its modifier, such as title and affiliation of the speaker (e.g. given "said Microsoft chairman Bill Gates ...", we recognize "Bill Gates" as the speaker, with "Microsoft" and "chairman" as the modifiers)

Verb: quotation verb. In addition, we store the prepositional modifiers of the verb. The modifiers usually provide context of the quote being made (e.g. given "said Bill Gates in the Microsoft shareholder meeting in Seattle", the modifiers are "in Seattle" and "in the Microsoft shareholder meeting")

Quote: actual quote within the beginning and ending quotation marks. Note that a quote could span multiple sentences. We search for starting and ending quotation marks from the neighboring sentences, and determine the proper quote boundaries. Then, we store all the segments of the same quote here.

In addition, we associate each triple with metadata tags from the document itself, i.e. URL, title, author, publication date, etc.

3.2 Indexing

Each extracted quotation and attribution is stored as a triple in our inverted index structure of subject-action-object triples (See Figure 3). For the underlying information retrieval capability, we use a typical Vector Space Model (VSM) based system, i.e. Apache Lucene[7]. By indexing syntactic and semantic annotations of text data using such a keyword search engine, we are able to provide a highly scalable, fast semantic seach capability[6]. In Lucene's index, each document contains one or more named fields. Each field corresponds to a piece of data that is either queried against or retrieved from the index during search[2]. In our case, each triple is treated as a document, while the elements of the triple are represented as fields.

- Subject field: store the speaker entity name, the entity's ID, and the entity's facets.
- Subject-modifier field: store modifiers of the speaker
- Action field: store the verb

[7] `http://lucene.apache.org`

- Action-modifier field: store context modifiers of the quotation, with entities marked up.
- Object field: store the actual quote, marked up with the entities recognized in the quote.

During processing of documents, we index the subject-action-object relations extracted from all the sentences, not just from quotes. The quotation triples are distinguished from other triples by a flag isQuotation. During search, only quotation triples will be retrieved when the isQuotation flag is set in the query.

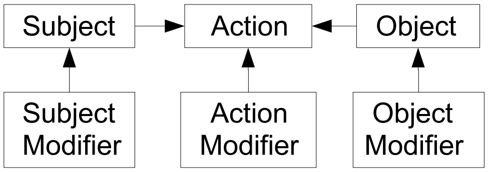


Figure 3: Subject-action-object triples

For entities identified both within and outside of the quote, we index not only the entity names, but also their unique identifiers and assigned categories (i.e. facets). Therefore, during search, we support search for quotes by or about entities by their names, as well as by their IDs or categories (i.e. find quotes made by any college football coach, or find quotes about any hybrid cars).

The subject-modifier field would support search for quotes made by speakers of certain properties, e.g. "Did anyone from Microsoft say anything about iPhone?"

Similarly, the action-modifier field supports searching for quotes within a particular context, e.g. "What did Obama say about global warming during his trip to China?".

Example 1:

> Nash said, "I would love to meet him, obviously, and to play hoops with the President would be kind of fun."

This quote is made by Steve Nash, an NBA basketball player, about President Obama. We are able to link Nash to his full name appearing earlier in the text. Through coreference resolution, we recognize "him" and "the president" refer to President Obama. We assign facet [basketball player] to Steve Nash. Therefore, when user queries about any comments made by any basketball players (or any sports athletes) regarding President Obama, this quote would be returned as one of the results.

Speaker field Entity name = Steve Nash
Entity ID = 0x49c26
Facet = [Basketball player]

Action Verb = said, isQuotation

Quote Keywords: love, meet, Barack Obama, obviously, play, hoops, president, fun
Entity 1:
Name = Barack Obama
ID = 0x49c26
Facets = [Politician], [Country leader]

Example 2:

> "They might think they've got a pretty good jump shot or a pretty good flow, but our kids can't all aspire to be LeBron or Lil Wayne," Obama said.

We recognize LeBron as LeBron James, the NBA basketball player, and Lil Wayne as a musician. The pronoun "they" is linked to "children" in the previous sentence. When users search for Obama's quotes regarding any basketball player or musician, this quote would be returned.

Speaker Entity Name = Barack Obama
Entity ID = 0x16f69
Facets = [Politician], [Country Leader]

Action Verb = said, isQuotation

Quotation Keywords: children, think, get, pretty, good, jump shot, flow, kids, aspire, LeBron James, Lil Wayne
Entity 1:
Entity Name = LeBron James
Entity ID = 0x49c85
Facet = [Basketball player]
Entity 2:
Name = Lil Wayne
ID = 0x15393
Facet = [Musician]

3.3 Search

This section describes the retrieval of the indexed quotes and presentation of the results.

3.3.1 Query

This system allows users to search for quotations in many different ways, by specifying quotes by certain speakers and/or about certain topics, as well as the categories (facets) of speakers and topics. The parameter <speaker> can be specified as:

- an entity, by its unique identifier or simply its name
- a facet, e.g. [football player]
- or unspecified, i.e. quotes by any person

Furthermore, the speaker field can be constrained by some modifiers, e.g. "What did any [Football player] from the University of Notre Dame say?" where Notre Dame is the modifier.

The <about> parameter can be specified as:

- an entity, by its unique identifier or name
- an entity facet, e.g. [movies], [hybrid cars]
- any keywords
- or unspecified, i.e. quotes about anything

We also support Boolean combinations (i.e. AND, OR) of multiple topics. For example:

- Speaker = Obama; Quote = China AND "global warming"

- Speaker = Peyton Manning; Quote = [football team] OR [football coach]
- Speaker = [actor]; Quote = Oscars AND any [movie]

The targeted entity does not have to be in the quote. For example, the quote from Bill Gates: "Oh my God, Microsoft didn't aim high enough." was about iPhone. We support search of such quotes by allowing the target entity as either within the quote boundary or as a context modifier.

- Speaker = Bill Gates; (Quote = iPhone) OR (Modifier = iPhone)

3.3.2 Result Presentation

For a given query, the result returned is a ranked list of quotation objects, each containing the following information:

- actual quote; the starting and ending positions of the quote are marked;
- quote attribution - speaker name and its modifiers;
- context - surrounding text outside the quote;
- document metadata, i.e. URL, document title, publication date, publisher name, the top entities we identified in the document text, etc.

Sometimes, the quote is very long such that we need to extract a snippet of a specified length that best matches the query request. During processing and indexing, we have identified the entities in each sentence, as well as their positions within the sentence. Given a query request on a particular subject (specified as entity or keyword), we determine the snippet that has most occurrences of the subject entity/keyword.

3.3.3 Result Aggregation and Ranking

Sometimes, what was said by a speaker could be quoted in multiple different articles. When retrieving quotes, we apply an aggregation process to detect duplicate quotes by computing the similarity distance between each pair of quotes.

The quotes are then ranked by a combination of the following factors:

1. Matching score between query and indexed quotes;
2. Publication date. We prefer quotes with fresher date;
3. Number of duplicates. Usually, interesting or significant quotes are repeated more often;
4. Credibility of the source, i.e. articles from major newspapers have higher credibility than less known blogs.

Users can choose to sort the results by their default rank or purely by date.

4. CONCLUSIONS AND FUTURE WORK

We demonstrated adapting the standard IR technologies (i.e. keyword queries matched against bag-of-words document representation) to semantically tagged natural text. By indexing the annotated quotations, we enable users to search for quotes made by a particular person or a category of speakers. Users can also search for quotes about an entity or a category of entities.

We have made APIs publicly available for querying over 10 million quotes extracted from news feeds of recent months. In addition, each day we add around 60 thousand new quotes extracted from around 50 thousand news articles or blogs. Based on our test on a set of 150,000 randomly sampled entities, the uncached query execution time has an average of 109 milliseconds with median at 54 milliseconds. Note that the queries are executed against an index that contains not only quotes, but all relationships extracted from every article text. The indexed document size is close to half billion.

As future work, we will investigate the relevance re-ranking of the retrieve quotes, aiming for surfacing more timely and authoritative quotes about a given topic, as well as the novelty and diversity of the results. In addition, during indexing, we could identify the most important and interesting quote to highlight for each article - automatically selecting the so called pull quotes. Finally, we will explore applying sentiment analysis[8] or opinion mining [1] to the extracted quotations.

5. ACKNOWLEDGMENTS

The authors would like to thank Andy Skalet for his initial implementation of the public APIs, and Jonathan Reichhold for his valuable suggestion on improving quote search performance.

6. REFERENCES

[1] A. Balahur, R. Steinberger, E. van der Goot, B. Pouliquen, and M. Kabadjov. Opinion mining on newspaper quotations. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, 2009.

[2] O. Gospodneti and E. Hatcher. *Lucene in Action*. Manning Publications, 2004.

[3] J. Liang, K. Koperski, N. Dhillon, C. Tusk, and S. Bhatti. *NLP-based Entity Recognition and Disambiguation*. US Patent Application 20090144609, 2009.

[4] J. Liang, K. Koperski, T. Nguyen, and G. Marchisio. Extracting statistical data frames from text. *ACM SIGKDD Explorations*, 7(1):67–75, June 2005.

[5] G. Marchisio, N. Dhillon, C. Tusk, K. Koperski, J. Liang, T. Nguyen, D. White, and L. Pochman. A case study in natural language based web search. In *Natural Language Processing and Text Mining*. Springer-Verlag, 2006.

[6] G. Marchisio, K. Koperski, J. Liang, T. Nguyen, C. Tusk, N. Dhillon, L. Pochman, and M. Brown. *Method and System for Extending Keyword Searching to Syntactically and Semantically Annotated Data*. US Patent 7,526,425, 2009.

[7] B. Pouliquen, R. Steinberger, and C. Best. Automatic detection of quotations in multilingual news. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2007*, 2007.

[8] `http://www.evri.com/developer/rest#API-GetSentimentInformation`