TagCurate: Crowdsourcing the Verification of Biomedical Annotations to Mobile Users

Bahar Sateli Sebastien Luong René Witte

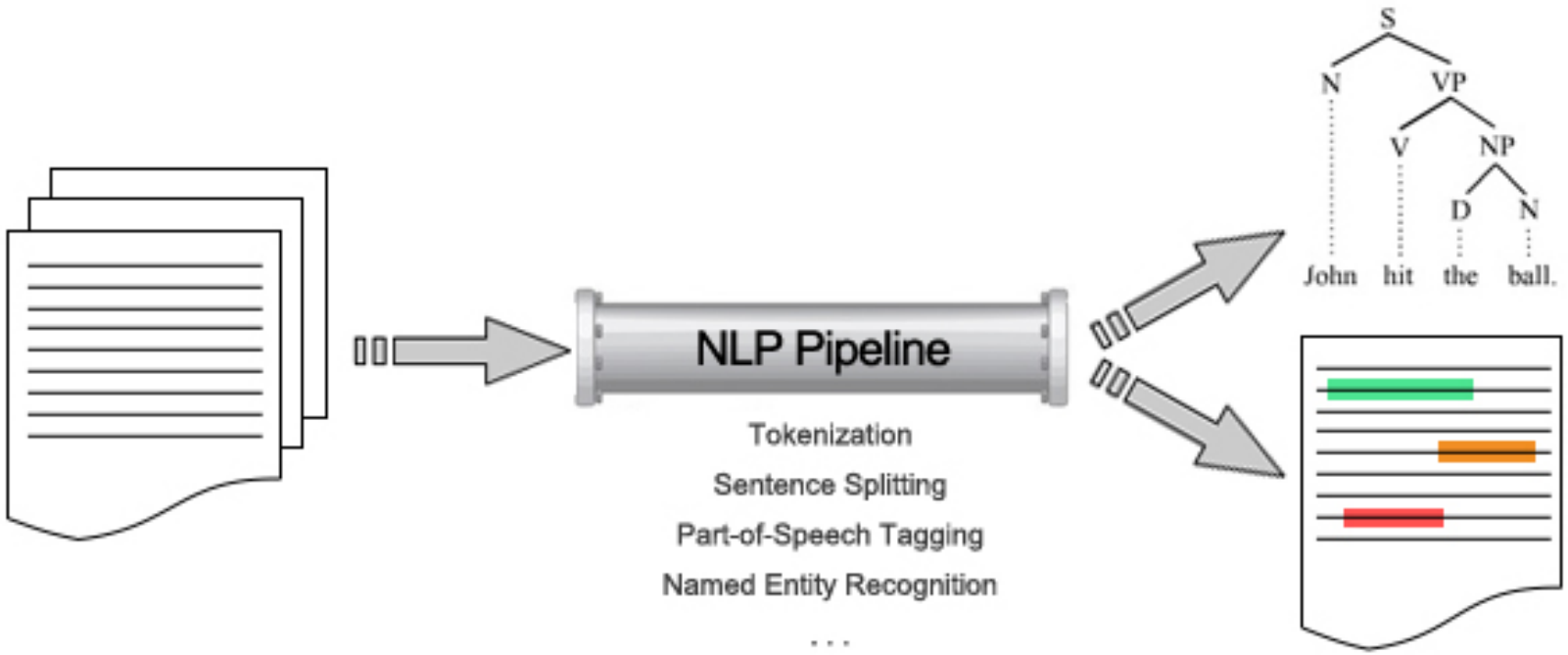
Semantic Software Lab
Department of Computer Science and Software Engineering
Concordia University, Montréal, Canada

NETTAB 2013

Natural Language Processing (NLP)

Definition

A branch of Artificial Intelligence that uses various techniques to process content written in a natural language, e.g., English or German.

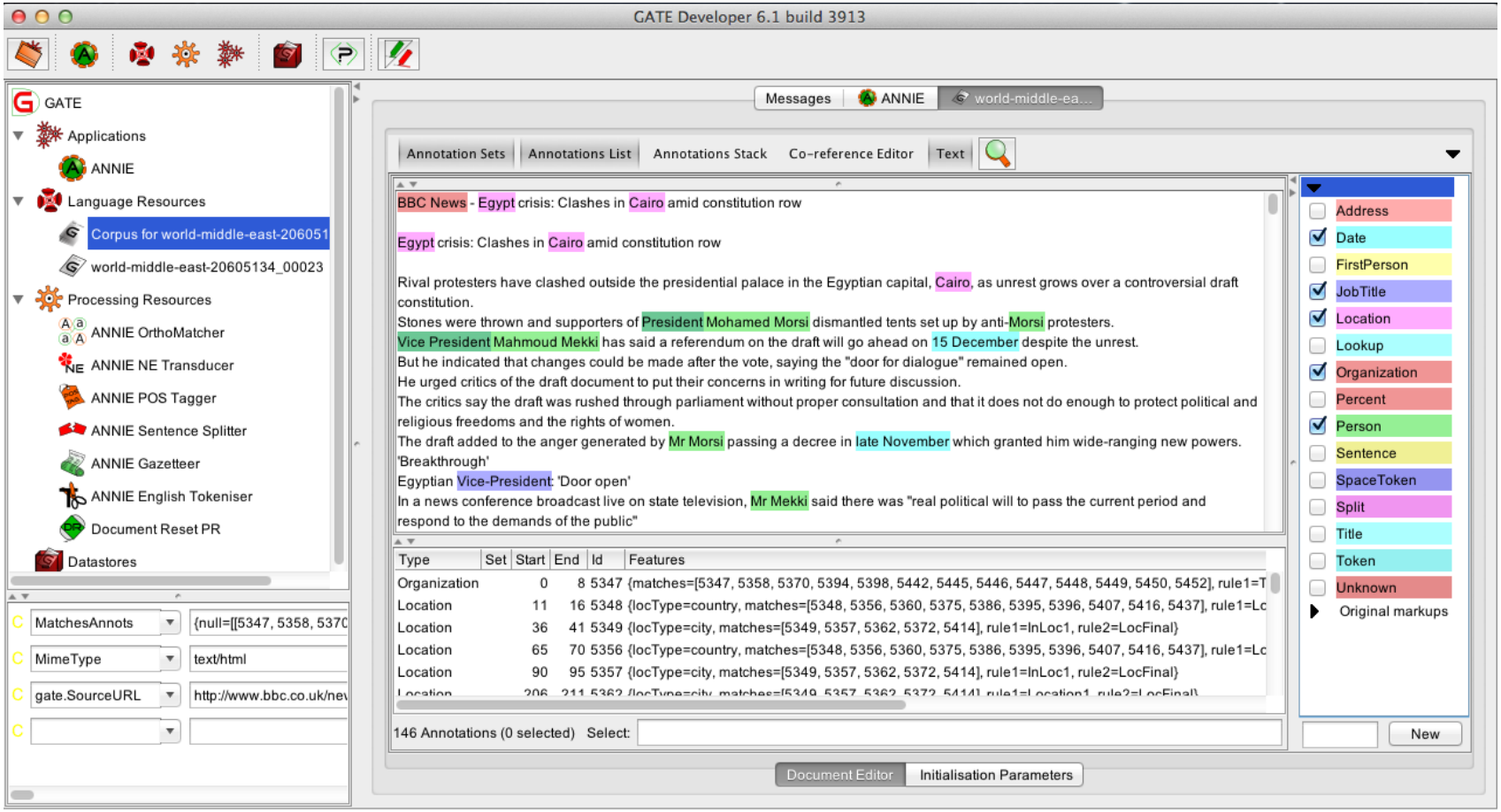


Bottleneck: Gold Standard Corpora

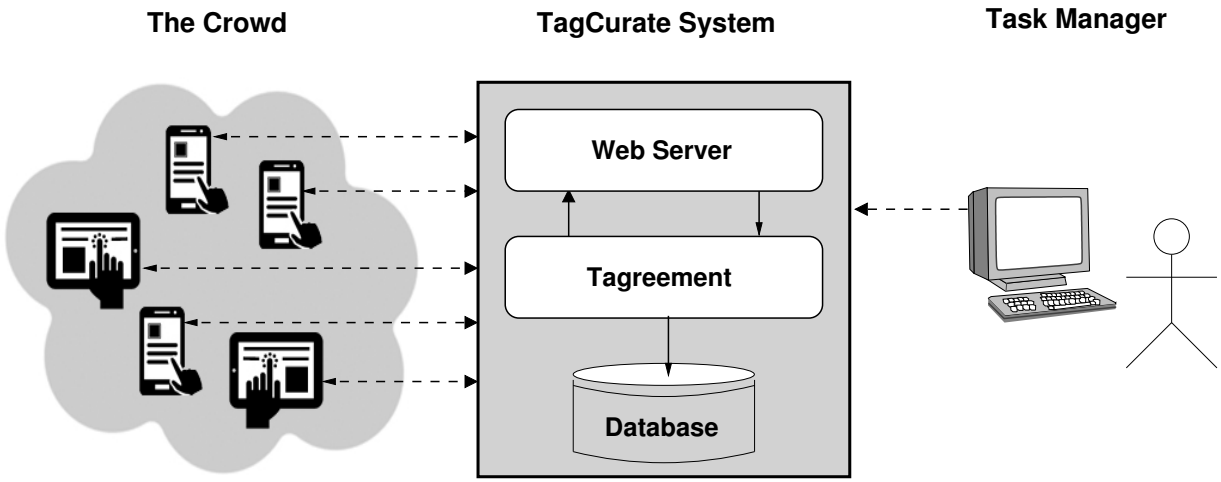
Manually annotated documents required for training & testing NLP pipelines (especially for machine learning components).

Can we 'crowdsource' some of this work to mobile users?

Challenge: Current Web-based annotation frameworks (e.g., GATE Teamware) not designed for mobile use



System Architecture



Client-Server Model

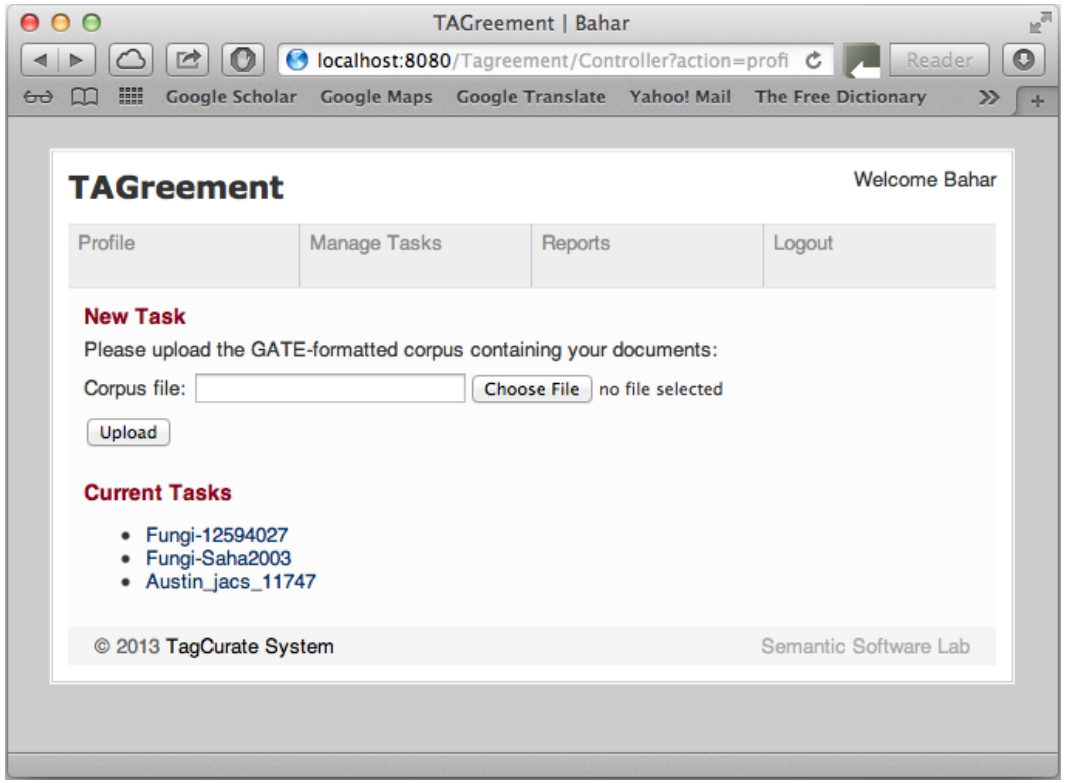
- RESTful communication over HTTP
- *Tagreement* component is responsible for managing the crowdsourcing as well as measuring (dis)agreements

User Groups

- **Task Managers**, define verification tasks using the web-based interface
  - e.g., *NLP pipeline developers, literature curators, . . .*
- **The Crowd**, verify (biomedical) annotations using the Android app
  - i.e., *Virtually anyone with access to an Android-enabled device*

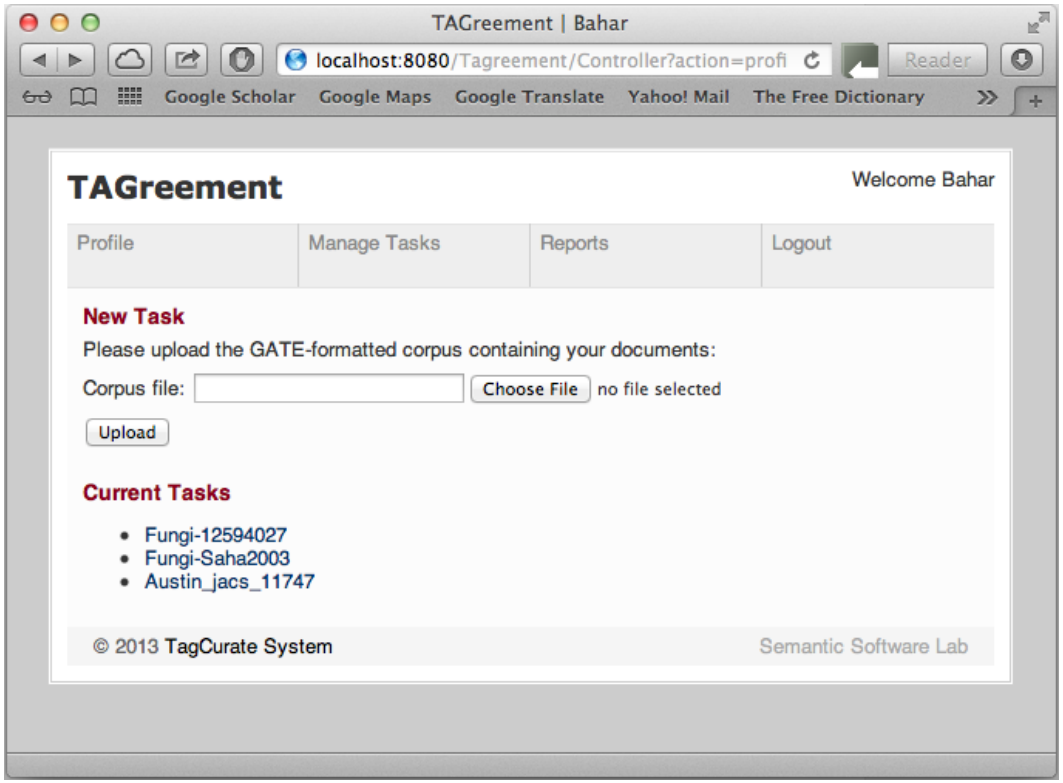
Tagreement Web-based Interface

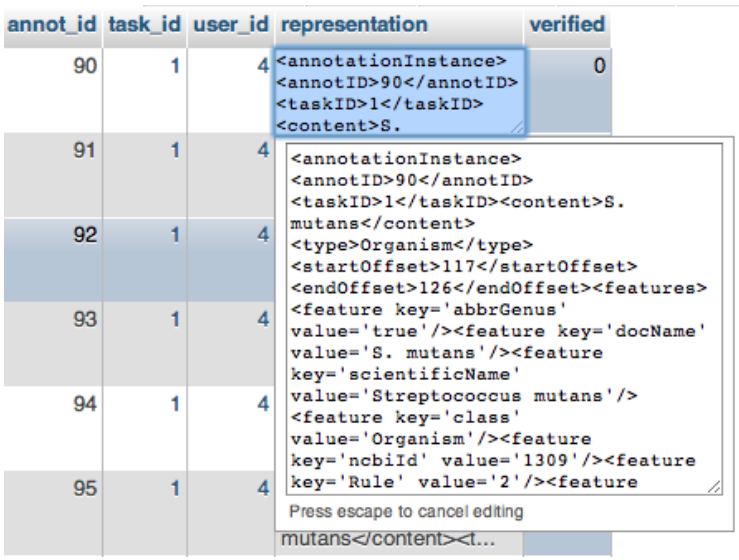
- *Task Managers* can define and supervise crowdsourcing tasks
- Currently, only accepts GATE-formatted corpora



Tagreement Web-based Interface

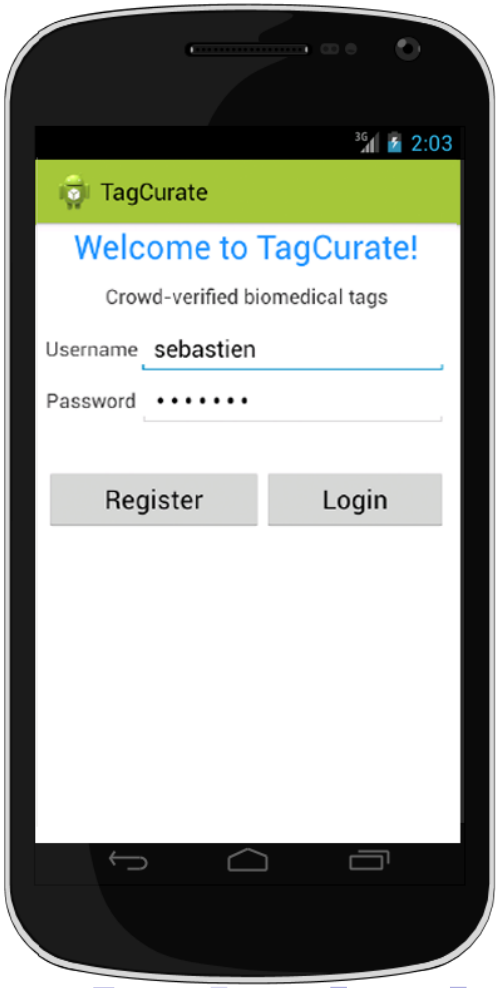
- *Task Managers* can define and supervise crowdsourcing tasks
- Currently, only accepts GATE-formatted corpora
- Stores an internal representation of each tag for distributed verification





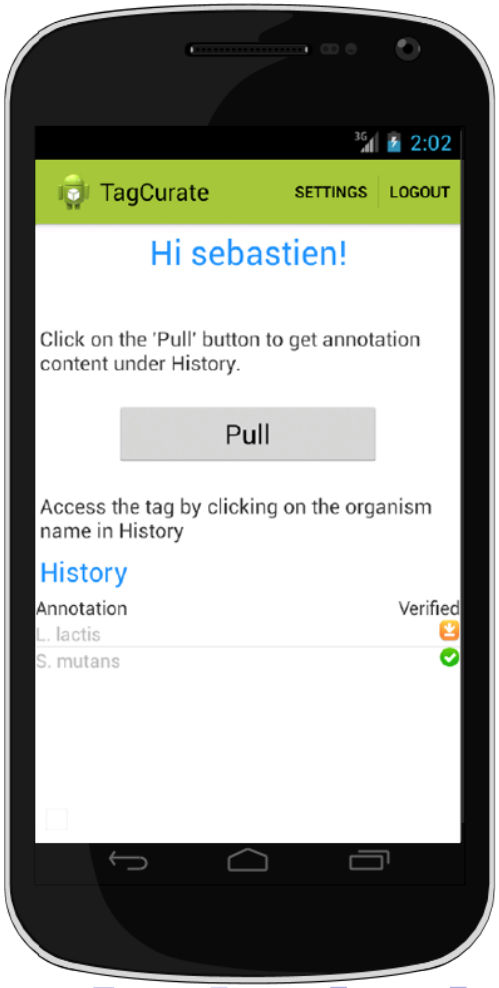
TagCurate Android App

- Developed based on the latest Android distribution (Jelly Bean version 4.3)
- Responsive design for phones and tablets



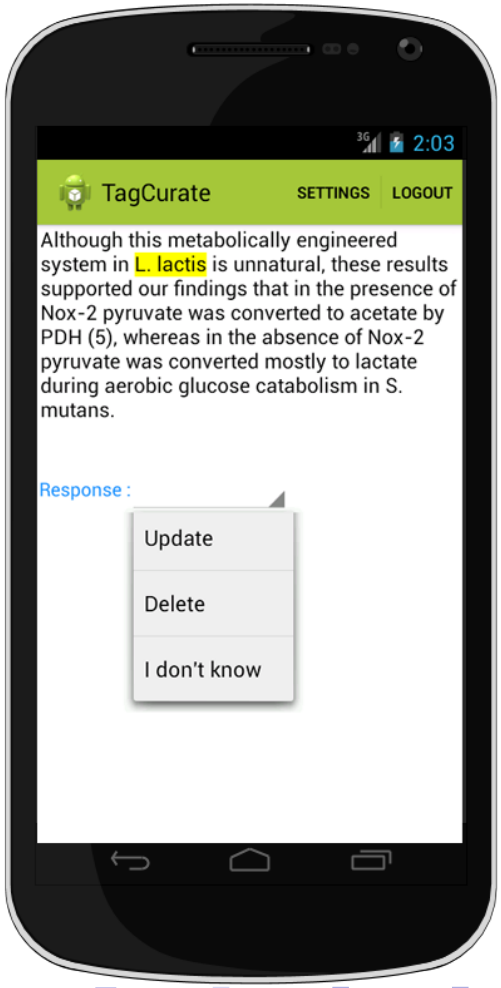
TagCurate Android App

- Developed based on the latest Android distribution (Jelly Bean version 4.3)
- Responsive design for phones and tablets
- Users authenticate themselves on the server
- Users *pull* tags from server
- Temporary storage of verification history



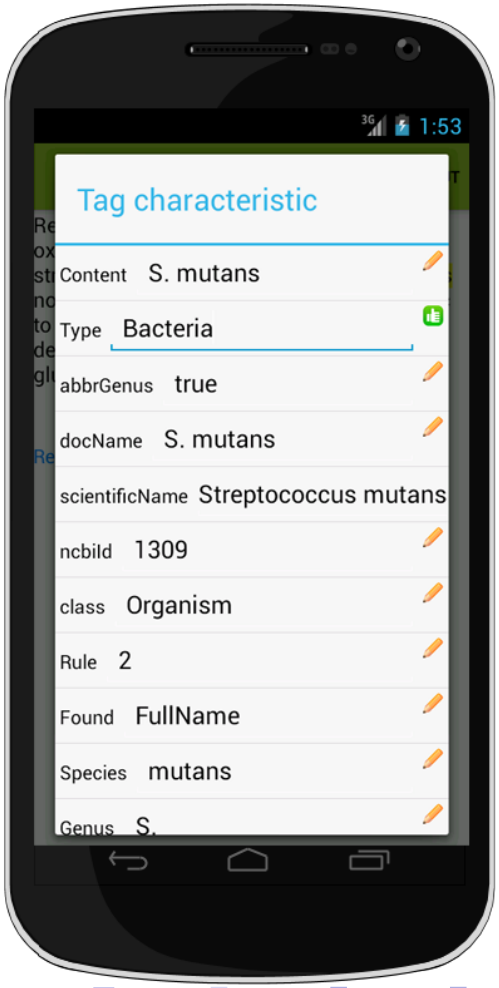
TagCurate Android App

- Developed based on the latest Android distribution (Jelly Bean version 4.3)
- Responsive design for phones and tablets
- Users authenticate themselves on the server
- Users *pull* tags from server
- Temporary storage of verification history
- View tags in context
- Verify whether a tag is a case of:
  - True Positive (correct)
  - False Positive (spurious)



TagCurate Android App

- Developed based on the latest Android distribution (Jelly Bean version 4.3)
- Responsive design for phones and tablets
- Users authenticate themselves on the server
- Users *pull* tags from server
- Temporary storage of verification history
- View tags in context
- Verify whether a tag is a case of:
  - True Positive (correct)
  - False Positive (spurious)
- Modify tags features
  - Pairs of $<$ key, value $>$
  - Modifications reflect in the tag representation



What about the missing tags?

Manual Annotation

Users select a text span and assign type and features to the generated tag.

Pros

- Human-generated tags usually have a higher quality

Cons

- Difficult task on devices with small screen
- Difficult to achieve an adequate inter-annotator agreement
- Requires well-established annotation guidelines

Automatic Annotation

Users invoke domain-specific text mining pipelines that generate various tags from text.

Pros

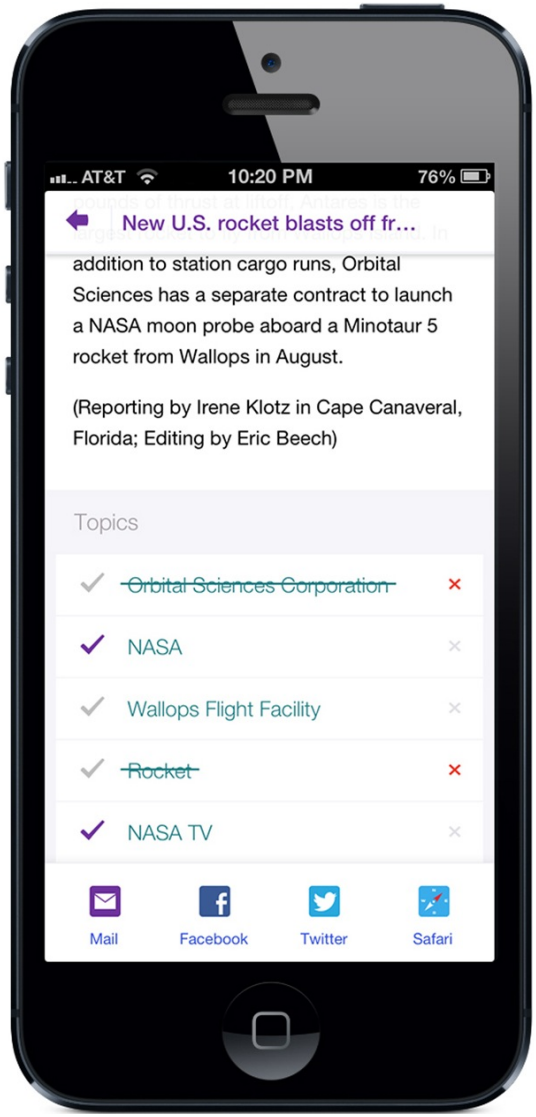
- Reuse existing text mining pipelines

Cons

- Text mining techniques are resource-intensive

Mobile Applications of NLP

- **Automatic Summarization**
  - Condensed version of document(s)
  - Various types: Generic, Focused, Update
  - e.g., Summly



(Image Courtesy of Yahoo!)

Mobile Applications of NLP

- **Automatic Summarization**
  - Condensed version of document(s)
  - Various types: Generic, Focused, Update
  - e.g., Summly
- **Question Answering**
  - Answering *factual* questions
  - e.g., Apple's Siri App

Mobile Applications of NLP

- **Automatic Summarization**
  - Condensed version of document(s)
  - Various types: Generic, Focused, Update
  - e.g., Summly
- **Question Answering**
  - Answering *factual* questions
  - e.g., Apple's Siri App
- **Information Extraction (IE)**
  - Identifying instances of specific classes
    e.g., Persons, Organization, Events, etc.

Mobile Applications of NLP

- **Automatic Summarization**
  - Condensed version of document(s)
  - Various types: Generic, Focused, Update
  - e.g., Summly
- **Question Answering**
  - Answering *factual* questions
  - e.g., Apple's Siri App
- **Information Extraction (IE)**
  - Identifying instances of specific classes
    e.g., Persons, Organization, Events, etc.
- **Content Development**
  - Combining other NLP services
  - Generate new or complementary content

Mobile Applications of NLP

- **Automatic Summarization**
  - Condensed version of document(s)
  - Various types: Generic, Focused, Update
  - e.g., Summly
- **Question Answering**
  - Answering *factual* questions
  - e.g., Apple's Siri App
- **Information Extraction (IE)**
  - Identifying instances of specific classes
    e.g., Persons, Organization, Events, etc.
- **Content Development**
  - Combining other NLP services
  - Generate new or complementary content
- **Other domain-specific services**
  - e-Health, e-Learning, etc.

Mobile Natural Language Processing

What we know

- Numerous mobile applications can benefit from NLP support
- Robust, open-source NLP frameworks are already available
- However, NLP analysis is a very resource-intensive task!

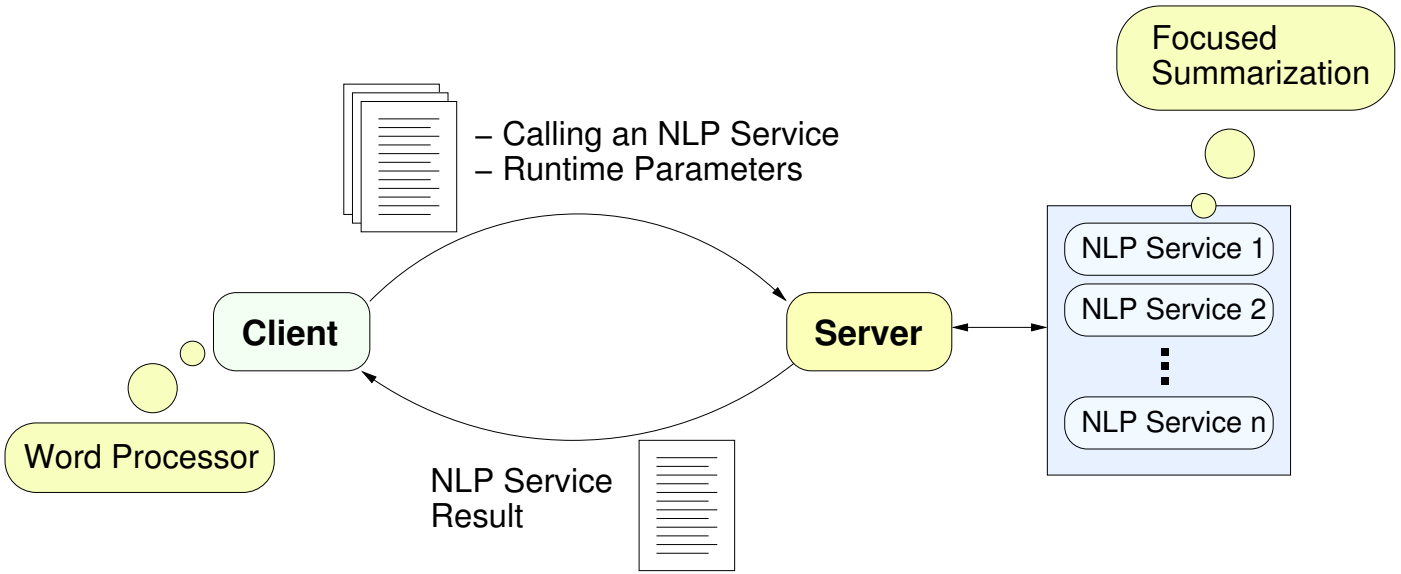
Semantic Assistants Android-NLP Integration

- Novel Android-NLP integration approach
- Provides Separation of Concerns
  - NLP developer does not need to know Android
  - Android app developer does not need to know NLP
- Android *library* for NLP service execution, rather than multiple apps
- Enable users to benefit from complex NLP services in their tasks

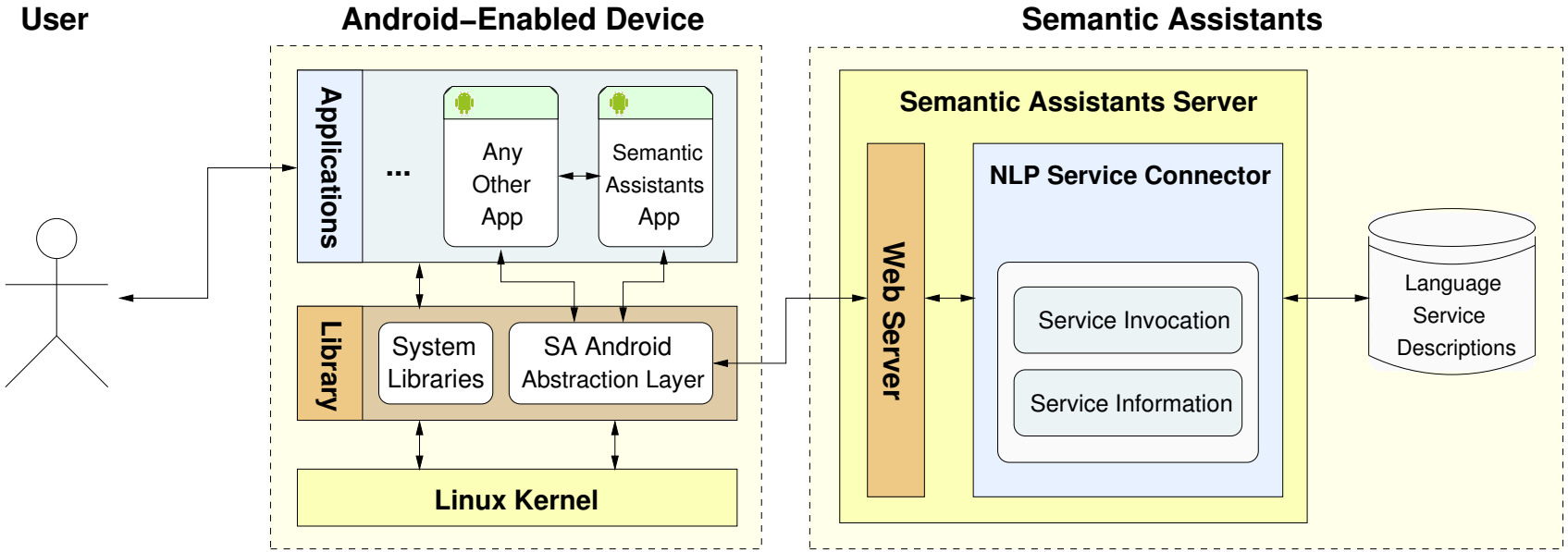
[B. Sateli, G. Cook, R. Witte, *"Smarter Mobile Apps through Integrated Natural Language Processing Services"*, MobiWIS 2013]

Semantic Assistants Framework

- Existing open-source (AGPL3) service-oriented architecture
- Brokers NLP pipelines as standard W3C Web services
- Avoids context-switching of user to external text mining applications
- Brings NLP analysis directly to various applications via plug-ins



Semantic Assistants NLP Intents



- Client-Server Model
  - Client is an Android app
  - Server-side component is the Semantic Assistants server
- RESTful communication over HTTP(S)
- Handles various NLP service result formats
  - *Annotation*, e.g., a person name in text
  - *Document*, e.g., summary of a long webpage
  - *Files*, e.g., an HTML document

Developing NLP Android Apps

Separation of Concerns

Android Developer

- Identify the NLP task
- Extend the SA intents by choosing a unique package name for this new service
- Embed the SA Android library in a new Android app
- Invoke the intent in app using the library

NLP Developer

- Develop the concrete NLP pipeline
- Deploy the pipeline on a SA server

Summary and Outlook

Summary

- Distribute annotation jobs to large user groups
- Expert annotators can focus on quality control and difficult cases
- Easily bring NLP pipelines to (Android) mobile apps

Ongoing work

- TagCurate app facelift
- Expanding the user profiles
- Finding incentives and introducing social aspects
- Add annotation capabilities (both manual and semi-automatic)

Find out more...

- Twitter: @SemSoft
- Web: `http://www.semanticsoftware.info/`