Teanga

A Linked Data based Natural Language Processing Platform

Housam Ziad, John P. McCrae, Paul Buitelaar
Agenda

NLP Services
Research Problems
Related Work
Teanga Features
Linked Data
Teanga Components and Technologies
Design and Implementation
Demo

Quiz and winner announcement: www.teanga.io/quiz

How to say “Teanga”
www.youtu.be/lupjm7ub7js
NLP Services

Term extraction
Taxonomy extraction
Machine translation
Sentiment analysis
Suggestion mining
Summarisation
Entity recognition
Text classification
...

Example Use Cases

Using entity recognition to extract the names of diseases or drugs from a healthcare database.

Using sentiment analysis to extract emotions from social media websites for the purpose of social science studies.

...
The Problems

NLP services frequently have one or more of the following problems:

- Focused on a single task
- Early technological readiness level
- Cannot be installed or run easily
- No modular design
- Limited to a single programming language
Related Work - Existing NLP Platforms

GATE: a system to support research and development of language processing software. (Cunningham, 2002)

UIMA: Unstructured Information Management Architecture (Ferrucci and Lally, 2004).

Language Grid: An Infrastructure for Intercultural Collaboration (Ishida, 2006)

WebLicht: environment for automatic annotation of text corpora. Linguistic tools (Hinrichs et al., 2010)

LAPPS Grid (Ide et al., 2015) and its Galaxy front-end (Ide et al., 2016)
Related Work - Some Issues

GATE
- “The visual interface is complex and somewhat non-standard.” (Cunningham, 2002)
- Intended for NLP experts

UIMA
- Only serves as middleware architecture - no interface to process data
- Complex installing and setting up of development environment
- Intended for NLP experts

Existing platforms often run on a desktop machine or rely on a platform-specific program, e.g. Eclipse plugins.
Interoperability of existing platforms guaranteed only by service developers following specific standards or guidelines.
Teanga

Features

- Containerisation
- Easy to install
- Multiple services
- Easy to use
- Building pipelines
Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.
Easy to install

By using containerisation, the platform can be installed by only pulling and running the container.
Multiple services

The ability to use only one interface for multiple services.
Easy to use

Teanga has a simple drag and drop interface.
Users can use Teanga to build a pipeline of services.
## Workflows

**Task:** Extract Suggestions from multilingual customer reviews.  
**Input:** Customer reviews in English (EN), German (DE), and Spanish (ES).

<table>
<thead>
<tr>
<th>Service</th>
<th>IN</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linked Data Profiling</td>
<td>LOD Docs</td>
<td>Selected LD Datasets</td>
</tr>
<tr>
<td>Machine Translation</td>
<td>DE, ES Docs</td>
<td>JSON LD EN Docs</td>
</tr>
<tr>
<td>Suggestion Mining</td>
<td>En Docs</td>
<td>JSON LD Docs</td>
</tr>
<tr>
<td>Entity Linking</td>
<td>SLD, JSON LD</td>
<td>JSON LD Docs</td>
</tr>
<tr>
<td>Data Visualisation</td>
<td>JSON LD, JSON LD</td>
<td>XLS, D3JS</td>
</tr>
</tbody>
</table>
Linked Data

(Berners-Lee, 2006) gives four key requirements for data to be considered linked data:

• Use URIs to identify all types of data items
• Make the URIs accessible globally by using HTTP URIs.
• If someone accessed one of the above mentioned URIs, provide useful structured information in RDF.
• Provide links among different data items by including RDF links that point to other URIs.
Linked Data in Teanga
Using JSON-LD in Teanga

We have chosen JSON-LD as the main output format for Teanga as:

1. It’s easy to use with all programming languages and environments, especially for Web browsers
2. Provides deep semantics based on RDF and other Semantic Web technologies
3. Introduces few overheads to the encoding of the data.
JSON-LD Example

{
  "@context": "https://json-ld.org-contexts/person.jsonld",
  "@id": "http://dbpedia.org/resource/John_Lennon",
  "name": "John Lennon",
  "born": "1940-10-09",
  "spouse": "http://dbpedia.org/resource/Cynthia_Lennon"
}
Technologies Used in Teanga

- Node.js
- MongoDB
- Docker
- Bootstrap
- AngularJS
Teanga Components

1. User Interaction
   - User
   - File Upload Service
   - MongoDB

2. Service Lookup
   - Docker Service

3. Teanga Startup
   - Service Lookup
Adding a Service to Teanga

Service Developer
- Write Teanga compatible JSON-LD file to describe the service
- Expose file when containerizing the service

Teanga
- Scans exposed file for compatibility
- Copies JSON-LD to “schema” directory
- Adds service to service list.
Service Description File

Teanga can host unlimited number of services, for that it needs a service description file to interact with services.

JSON-LD format
Describes input params and data types
Describes output params and data types
Error Control

Teanga can handle the following errors:
Showing the service error message, if returned.
Showing a service failure message instead of crashing the system.
Showing an error message that the service is returning an empty message.

We’re still working on improving error handling with our continuous experiments with adding services.
Demo Time

www.teanga.io/demo
Questions

I ❤ Teanga
Bibliographical References