Towards a system architecture for ICALL

Based on NLP component re-use

Elena Volodina*, Lars Borin*, Hrafn Loftsson**, Birna Arnbjörnsdóttir***, Guðmundur Örn Leifsson****

* Centre for Language Technology, Språkbanken, University of Gothenburg, Sweden
** School of Computer Science, Reykjavik University, Iceland
*** School of Humanities, University of Iceland
**** School of Engineering and Natural Sciences, University of Iceland, Iceland

ICALL: Intelligent Computer-Assisted Language Learning

Intelligence is ensured through the use of NLP tools and/or AI techniques

Systems Architecture for ICALL Funded by NordPlus Sprog

Partners: Reykjavik University, University of Gothenburg, University of Iceland
General aim: Encourage and facilitate the use of NLP tools and resources in CALL

Practical aims:
- Design a system architecture for re-use of existing reliable NLP tools/resources in CALL
- Implement applications for testing architecture

Principles for architecture and applications:
- Open-source, re-use, language independent
- Easy to adapt to new tasks
- Plug-and-play basis, modularity

NLP-based CALL:
- Is characterized by use of NLP tools/resources
- Ensures linguistic analysis of the input data through use of NLP tools/annotated resources
- Adds generality, i.e., coping with the same analysis model to different (authentic) language samples

Re-using NLP components

Most NLP components are:
- Monolithic and inflexible; need to be individually adapted to every new application
- Not readily available as the rights are held by individuals or institutions all over the world
- Physically located in different places
- Not interoperable via standardized interfaces

Strategies for making use of them:
- Rewrite in the target programming language
- Find chunks of similar code and build upon it
- Using open-source initiatives, e.g., http://www.tsf.org
- Standardize communication between the tools and resources, e.g., initiatives for corpora (EAGLES, TEL, etc.); for e-learning (MS Global Learning Consortium, SCORM, etc.); for NLP tools (GATE, NLTK, Apache UIMA – still bound to programming languages (Java, Python))

SOA & web services: an approach to NLP component re-use

Service Oriented Architecture (SOA) principles:
- Modular services that can be re-used by others
- Communication layer with a well-defined interface for sending a request and getting a response
- Standardized data output format
- Well-documented interface and its service
- Services loosely coupled and can be re-combined

Web services as an implementation technology:
- Wrapper around a program making it accessible worldwide
- Can re-use other web services, databases, resources, etc.
- Access over Internet; the original software can still be residing on its original server
- Standardization initiative: trying to attract software and resource owners to provide web services

The Icelandic work

NLP and ICALL for Icelandic:
- IceNLP: Open source collection of tools for processing and analyzing Icelandic texts. Contains, e.g., a tokenizer, an unknown word guesser, a PoS tagger, and a shallow parser.
- Currently, no ICALL application exists for Icelandic
- ICALICd Online (IOL) is a CALL application (web course) with almost 90,000 registered users

ICALL platform:
- Being developed for supporting ICALL systems
- Connects various pre-existing NLP tools from IceNLP
- Uses the Text Corpus Format (TCF) for communication of information between components
- Individual components can be accessed through a web service

Output from the platform in TCF:

```plaintext
<text>Enn er gððr kennarli</text>
<tokens>
	<Token ID="t3">Henn</Token>
	<Token ID="t2">er</Token>
	<Token ID="t1">gððr</Token>
	<Token ID="t4">kennarli</Token>
</tokens>
</tokens>
</Postags tagset="tfd">

The Icelandic word

The adjective "gððr" (‘good’) does not agree in gender with the following noun "kennari" (‘teacher’) in the noun phrase “gððr kennari”.

First evaluation results

<table>
<thead>
<tr>
<th>Error type</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Errors in noun phrases</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>Errors between subjects and verb complements</td>
<td>100.00%</td>
<td>87.50%</td>
</tr>
<tr>
<td>Errors between subjects and verbs</td>
<td>42.99%</td>
<td>42.99%</td>
</tr>
<tr>
<td>Incorrect case selection of web objects</td>
<td>100.00%</td>
<td>50.00%</td>
</tr>
<tr>
<td>All types</td>
<td>76.00%</td>
<td>76.00%</td>
</tr>
</tbody>
</table>

In general students found the system helpful for error detection and that it aided them in their writing.

The Swedish work

Lärka (Eng. Lark) – LAR språket via KorpusAnalyser

- ICALL platform, at the moment consisting of an exercise generator; eventually other learner-related activities, e.g., rating corpus hits, manipulating vocabulary lists, performing readability analysis, etc.

Characteristics:
- Web-based, modular, plug-and-play principle, SOA-based with web-service implementation
- Underlying corpora: SUC2, Talbanken, LåsBart
- Underlying lexicons: Saldo, Wikipedia, Wiktionary, Lexin
- Underlying word lists: Kelly list, Base, Vocabulary pool, Lexin domain lists, Swadesh list, Academic word list, etc.
- Scope: exercise types for linguists and for L2 learners
- Modes: self-study and test

Feedback: in terms of correct/incorrect and lexicon & encyclopedia entries

Flow of control (Lärka’s architecture)

- Output from Lärka’s web service in JSON format

Lärka, future

- Expand exercise scope, e.g.: gap cloze, wordbank; morphological paradigm, semantic closeness, yes-no diagnostic test, spelling, naming grammatical features; yes-no order by syntactic group, word-building;
- Add learner lists/lexical database
- Enrich encyclopedia feedback
- Visualize syntactic tree for sentence-based exercises and eventually add exercises based on syntactic trees;
- "hit-ex" – rating corpus search hits (tests are ongoing)

Readability analysis
- Half-automatic mode for exercise generation (feeding the system with the user choices/lists, etc.)
- Editable “mode” of exercise production – proofreading and modifying automatically created items; saving the items into a database
- Error typology and analysis of written texts etc.

Writing support: http://nlp.cs.ru.is/rithjalp

Lärka: http://sprakbanken.gu.se/larka