Resources for intelligent computer-assisted language learning (ICALL)

Elena Volodina, Research Engineer
Språkbanken (Swedish Language Bank), Dpt of Swedish Language
elena.volodina@svenka.gu.se

Lecture plan

- Computer-Assisted Language Learning (CALL) versus Intelligent CALL
  - Definitions and short historical overview
- Language Learning (L2) – short introduction
  - L2 main steering document (CEFR)
  - L2 skills, proficiency levels and L2 activities
- NLP for Language Learning (L2)
  - Supportive NLP components
  - Demos
- Designing an ICALL application – methodology and problems
  - Defining aims: end-user perspective first
  - Reuse of NLP components versus creating new ones
  - Standardization
  - Architecture principles
  - Evaluation
- ICALL platform development at Språkbanken: Lärka

Computer-Assisted Language Learning

- Drill-and Kill era
- Multimedia and graphics
- Authoring tools as a way out of CALL determinism
- Web-based materials, item banks --> need for standards
- Criticism

Intelligent Computer-Assisted Language Learning

- AI-based CALL
- Intelligent tutoring systems as surrogate teachers
- NLP-based CALL
  - Annotation-based CALL
  - Parser-based CALL
  - Any NLP component as intelligence in the application

NLP-based CALL

- characterized by use of NLP (Natural Language Processing) tools and resources in the area of CALL

Advantages

- ensures linguistic analysis of the input data through use of NLP tools and/or annotated resources
- adds generative power of applying the same analysis model to different (authentic) language samples over and over again, e.g. for generating exercises or detect errors in text production
- enables reuse of NLP tools and resources for practical use in language learning and
  - relieves teachers of monotonous tasks that can be modeled by computers
  - supports self-learning for students where it is feasible and motivated
  - popularizes NLP among CALL end-users

What is necessary?

- available reliable NLP tools/algorithms, e.g.:
  - sentence segmenters, tokenizers, pos-taggers, lemmatizers, syntactic parsers, error parsers, spell-checkers, etc.
- available reliable annotated resources, e.g.:
  - corpora, lexicons, learner-oriented word lists, etc.

Where do we get them?

- re-use existing reliable NLP tools and resources
- develop/create lacking ones
Language Learning Framework

- CEFR – Common European Framework of References for assessment of language proficiency (Council of Europe 2001)

- The CEFR is a document which describes in a comprehensive manner i) the competences necessary for communication, ii) the related knowledge and skills and iii) the situations and domains of communication as well as provides guidelines.

NLP components in support of individual language skills

<table>
<thead>
<tr>
<th>L-ge skill</th>
<th>NLP components... Ideas?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vocabulary</td>
<td></td>
</tr>
<tr>
<td>Grammar</td>
<td></td>
</tr>
<tr>
<td>Pronunciation</td>
<td></td>
</tr>
<tr>
<td>Reading</td>
<td></td>
</tr>
<tr>
<td>Writing</td>
<td></td>
</tr>
<tr>
<td>Listening</td>
<td></td>
</tr>
<tr>
<td>Speaking</td>
<td></td>
</tr>
</tbody>
</table>

Vocabulary

- Spelling, morphology, pronunciation, exercises (translation-based, semantic-based, sentence-based, text-based), testing

- Potential NLP components:
  - Learner lists (freq-based, level-based, text-based, etc.)
  - Spell-checkers
  - Lexicons
  - Morphological analyzers
  - PoS-taggers
  - Lemmatizers
  - Corpora – general, domain-specific, learner, written, spoken, etc. (collecting, annotating, using)

- Demos:
  - Multidict: http://www2.smo.uhi.ac.uk/multidict/

Vocabulary – research questions

- Vocabulary scope per CEFR level. How to identify? How many words per level? Which ones?
- Go by frequency... On what texts? Where to get texts? Copyright restrictions?
- Go by domain... Again – which words for which level? Manual work? Intuitions?
- Multiple choice: selection of distractors. Feasible for automatic approaches?
- Automatic selection of sentences for training... Procedures for testing sentence for complexity per level. Problems.
- Semantic disambiguation of polysemous words: lexeme rather than lemma... What do we need for that? (to fire)
- ......

Grammar

- Recognition, use, exercises, testing

- Potential NLP components:
  - Morpho-syntactic description (corpus annotation & tagging)
  - Syntactic parsing (dependency relations)
  - Tree visualization
  - Error parsing
  - Formal grammars (phrase grammar, context-free grammar, etc.)
  - Corpora & corpus search applications

- Demos:
  - Grammarly: http://www.grammarly.com/
  - VISL: http://beta.visl.sdu.dk/
Grammar – research questions

- Grammar scope per CEFR level. Which grammar phenomena?
- Go by frequency... On what texts? Where to get texts? Copyright restrictions?
- Manual work? Intuitions?
- Multiple choice: selection of distractors. Feasible for automatic approaches?
- Acceptable alternative forms – how to solve the problem?
- Automatic selection of sentences for training... Procedures for testing sentence for complexity per level.
- ......

Reading

- Understanding, question asking and answering, grammar+vocabulary
- Potential NLP components:
  - Corpora (+ annotation)
  - Automatic text selection
  - Information retrieval
  - Readability assessment (text & sentences)
  - Text-to-Speech synthesis
  - Question generation
  - Semantic disambiguation
- Demos:
  - http://www2.smo.uhi.ac.uk/clilstore/
  - http://www.let.rug.nl/glosser/Glosser/
  - https://the.sketchengine.co.uk/
  - http://sifnos.sfs.uni-tuebingen.de/WERTi/index.jsp

Writing

- Essay writing, letter writing, free responses
- Potential NLP components:
  - Error parsing
  - Error detection
  - Error annotation
  - Spell checking
  - Lexicons
  - Specific corpora
  - Assessment
  - Feedback generation
- Demos:
  - https://www.ets.org/Media/Products/Criterion/tour2/critloader.html

Pronunciation, Listening, Speaking

- Recognition in speech, use in speech, training exercises
- Potential NLP components:
  - Lexicons with recordings of the words
  - Text-to-Speech synthesis
  - Speech recognition
  - Dialogue-based systems
- Demos:
  - http://imtranslator.net/translate-and-speak/

Developing ICALL applications

- End-user needs versus technological solutions.
  - Technology-driven or pedagogically driven?
  - NLP community versus L2 teachers – how aware are they of each other?
- Reliability of NLP components. Some linguistic teasers.
- Designing an ICALL application – methodology and problems
- Standardization efforts (GATE, etc)
- Architecture principles
- Evaluation
- ICALL platform development at Språkbanken: Lärka, its scope and future

End-user needs versus technological solutions.

- Technology-driven or pedagogically driven?
- Expectations of technology vs what it can perform
- NLP community versus L2 teachers
  - Are they aware of each other?
  - Communication problems? Different cultures? Misunderstandings?
  - Linguists = technophobes and keep to their practices?
  - IT developers = arrogant and ignorant of pedagogy and language learning?
Use of NLP components in CALL: Criticism

- Not reliable; cannot promise 100% correctness.

Linguistic teaser 1

- Where should a sentence boundary go?

"What have you done to your breast?" she asked.

- </s>"What have you done to your breast?</s> " she asked. </s>
- </s>"What have you done to your breast?" she asked.</s>

Linguistic teaser 2

2. Warnings in style with "Look out for the stairs!", "Watch your head!" are seen everywhere.

- </s>Warnings in style with "Look out for the stairs!"</s> , "Watch your head!"</s> " are seen everywhere.</s>
- </s>Warnings in style with "Look out for the stairs!", "Watch your head!" are seen everywhere.</s>

Linguistic teaser 3

She moved a bit

she = pronoun
moved = verb
a = determiner
bit = noun

- </s>"What have you done to your breast?</s> " she asked.</s>

Reusing NLP components

- 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

- What are the strategies for making use of them?
  - Rewrite in another programming language
  - Find chunks of similar code and build upon it using open-source initiatives (e.g. Free Software Foundation: http://www.fsf.org)
  - Any other ideas?

Reusing NLP components, 2

- Communication between the tools and resources
  - Standardization initiatives for corpora (EAGLES, TEI)
  - Standards for e-learning (IMS Global Learning Consortium, SCORM, ADL)
  - Standardization for NLP tools (GATE, NLTK) – still bound to programming language
  - Service-Oriented Architecture
  - Web services

- Possible solution - reuse without rewriting the code. Suggestion: SOA – Service Oriented Architecture.

Reusing NLP components, 3

- Service Oriented Architecture (SOA) principles:
  - Modular services that can be reused 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 recombined

- Web services as an implementation technology
  - Wrapper around a program defining a port of access to it
  - Can reuse 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
Evaluation

- Group vs individual
- Qualitative vs quantitative
- Student vs teacher opinions
- Pedagogical effect (teacher vs student perspectives)
- Reliability
- Time effectiveness
- Retaining rate
- etc.

System Architecture for ICALL

- Lärka (Eng. Lark) – LÄR språket via KorpusAnalyse:
  - 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, web-service based
  - Underlying corpora: SUC2, Talbanken, LäsBart
  - Underlying lexicons: Saldo, Wikipedia, Wiktionary, ev. Lexin, etc.
  - Underlying learner lists: Kelly list, Base Vocabulary pool, Lexin domain
    vocabulary lists, Swadesh list, Academic word list, etc.
  - Scope: exercise types for linguists and for vocabulary training
  - Modes: self-study and test
  - Feedback: in terms of correct/incorrect and lexicon/encyclopedia entries

Lärka flow (architecture)

Lärka: Syntactic relations exercise

Lärka: POS exercises
Lärka, near future

- Expand exercise scope:
  - (already in pipeline): gap cloze, wordbank
  - (potentially): morphological paradigm, semantic closeness, yes-no diagnostic test, spelling, etc...
  - (potentially, corpus-based): naming grammar features (past, present, etc); shuffling word-order by syntactic groups
  - word-building (compounding, affixation)

- Learner lists/lexical database: vocabulary organized by domain (FrameNet, Lexin), CEFR-level (Kelly, BaseVoc), and word class; dynamically added translations and corpus examples by internet users

- Enrich encyclopedia feedback

- Syntactic tree for sentence-based exercises and eventually exercises based on syntactic trees

Lärka, distant future, if ever

- Readability analysis (for texts and sentences)
- "Hit-ex" - rating corpus search hits (tests are ongoing)
- Half-automatic mode for exercise generation (feeding the system with the user choices/lists, etc.)
- Editable "mode" of exercise generation – proofreading and modifying automatically created items; saving the items into a database
- Error typology and analysis of written texts
- etc.

Thank you!
Questions?