The Computer and Natural Language (Ling 445/515)
Computer-Aided Language Learning

Markus Dickinson
Dept. of Linguistics, Indiana
Autumn 2010
Some common computer uses

- Computers are widely used in support of foreign language teaching (FLT). For example, they
  - provide access to foreign language newspapers, radio, and TV programs through the world-wide web
  - connect language learners with native speakers through email/chat
  - support multimedia presentations providing an audio-visual foreign language context
  - enable the learner to search for real-life examples in electronic corpora

- Essentially, such computer usage helps language learners experience a foreign language and culture in a more direct, real-life fashion.
Language Tutoring Systems

Overarching question: How computers can help provide foreign language learners with experiences that are:

- richer,
- more personalized, and
- more effective?
First Language Acquisition

Second language learning is very different from **first language acquisition**

- Babies & young children need no instruction
- Researchers disagree on how much of language learning ability
  - is **innate**, i.e., a biological endowment
  - **emerges** from experience, i.e., a rich social and physical environment.
First Language Acquisition

Stages of First Language Acquisition

Typical stages of first language acquisition:

- Babies play with making sounds and around six months, typical babies begin to **babble**
  - use sequences of consonants and vowels, e.g., *bababa*
- Quickly start learning words by first birthday
- Form simple 2-word utterances by the time they turn 2
- Voice & understand complex sentences by 3
- Continue acquiring words & complex language structures over next 9 or 10 years
  - Some structures, e.g., passives, added relatively late

Essentially the same pattern across all languages and cultures (with some individual variation)

A child can be a **native speaker** of multiple first languages, acquiring each of them without explicit instruction
Second Language Acquisition

Awareness of language forms

Adults do not automatically acquire a second language

- Even after living in a foreign country for a long time, listening to & talking in a foreign language there

- Research since the 90s has shown that awareness of language forms and rules is important for an adult learner to successfully acquire a foreign language.
  - e.g., the use of the articles *the* and *a* in English is difficult to learn
    - especially for those whose native language does not make use of articles (Chinese, Russian, etc.)
  - Mastery requires awareness of categories such as mass nouns (e.g., *rice*) & generics (e.g., *milk* in *I like to drink milk*)

CALL can provide an opportunity to enhance awareness of a language’s rules
Needs of second language learners

- The time a student can spend with an instructor/tutor typically is very limited.
- In consequence, work on form and grammar is often deemphasized and confined to homework so that the time with the instructor can be used for purely communicative activities.
- The downside is that the learner has relatively few opportunities to gain awareness of forms and rules and receive individual feedback on errors.

Recent British National Student Survey found:

- “students are notably less positive about assessment and feedback on their assignments than about other aspects of their learning experience”
An opportunity for CALL

- The situation seems like an excellent opportunity for developing Computer-Aided Language Learning (CALL) tools to
  - provide individual feedback on learner errors and
  - foster learner awareness of relevant language forms and categories.

- But existing CALL systems which offer exercises
  - typically are limited to uncontextualized multiple choice, point-and-click, or simple form filling, and
  - feedback usually is limited to yes/no or letter-by-letter matching of the string with a pre-stored answer.
From CALL to ICALL

- Linguistic modeling is needed to improve on this situation, e.g.:
  - tokenization: identify words
  - morphological analysis: identify/interpret morphemes
  - syntactic analysis: identify selection, government and agreement relations and word order requirements
  - formal pragmatic analysis: identify coreference relations, information structure partitioning, . . .

- Computational tools identifying such linguistic properties need to be integrated into CALL systems to obtain language-aware “Intelligent” CALL (ICALL).
  - Tools must be extended/written to permit and diagnose errors made by language learners.

CALL is a big business: 106 million Euro (about $120 million) spent on CALL products in Europe in 1994, and US market is twice as big. (Nerbonne 2003)
Expectations

How successful we think CALL systems are depends upon our expectations and upon the expectations of the students.

- Unlike classroom situations, where the instructor can be assumed to be right, ICALL systems are prone to errors:
  - Parsers return wrong analyses: students could be told they are right when they are wrong, and vice versa
  - Parsers are incomplete: no analysis may be returned
- Students have to understand up front that this is not a human:
  - Perhaps obvious, but affects, for example, the image of a tutor you might have on screen
CALL Systems

Having discussed what we need to know when learning a language, we can now focus on:

- Basic uses of computers for CALL
- Early CALL Systems
- Intelligent CALL (ICALL)
  - Straightforward uses of NLP for CALL
  - Authentic Text ICALL Systems
  - Parser-Based ICALL Systems
Basic uses of computers for CALL

Lots of general possibilities for using a computer to learn:

- multimedia presentations
- online dictionaries with fast access
- extensive databases of information
- digital audio files
- digital videos of people speaking in L2
  - Digital advantages: easy playback, easy isolation of problematic spots, etc.
- interactive games & puzzles
- exercises for students to complete

The last two examples potentially require sophisticated natural language processing
Computers can explicitly store knowledge about words or grammar necessary to complete a specific exercise.

1. Fred lives _____ Mill Street, doesn’t he?
   - in
   - on
   - at

2. My father was born _____ Christmas Eve.
   - at
   - on
   - in

3. Come here _____ once! I need your help right now!
   - at
   - on
   - in

(Source: http://www.eslcafe.com/quiz/prep3.html)

**multiple choice exercises** work well for practicing or testing specific choices of forms or meanings:

- include so-called **distractors** as incorrect choices.
CALL systems

Fill-in-the-blank

Putting questions on the web or another computer-based platform makes it possible to provide immediate feedback.

Other possible exercises include:

- pull-down menus listing the choices
- **fill-in-the-blank (FIB)** texts: a word in a sentence is erased & the learner must type in the missing word
  - also referred to as **cloze** exercises
  - often include a **fallback case** to respond to any unexpected input
    - i.e., **canned text responses**
Early CALL systems

- **Frame-based systems** “match student answers with a set of correct and incorrect answers stored in a frame”

These systems differ in their strategies for selecting questions, but they rely on preset questions and answers

- In principle, could be used with NLP techniques

Many also feature a dynamic **sequencing of instruction**
Linear systems

A **linear system** does the following:

1. poses a question
2. accepts an answer
3. informs the student as to whether or not the answer was correct

⇒ Regardless of the correctness of the answer, linear systems proceed to the next question.

▶ **Branching systems** select a question of appropriate difficulty based on performance thus far
Problems with frame-based systems

Frame-based systems are fairly simple and generally do not involve much linguistic knowledge

- There is no deep understanding of question domain
- They generally only match answers with questions, but language use is more varied
  - Could be more than one correct answer, as with a translation task
  - Fill-in-the-blank exercises (pointed out by Trude Heift):
    1. Today is November 10. What date is tomorrow? Tomorrow is ____.
       a. The eleventh. [Judged correct]
       b. November 11. [Judged incorrect]
- There is not much tailoring to particular student needs
Intelligent CALL (ICALL) focuses on using linguistics and natural language processing to make CALL better.

- Some basic NLP uses for CALL (Nerbonne 2003):
  - Concordancers
  - Text alignment
  - Speech recognition and synthesis
  - Morphological processing (for vocabulary help)

- ICALL can also involve integrating authentic text into exercises, usually for more advanced learners

- ICALL involves using a parser (syntactic processor) to handle real learner input and deals with issues like:
  - Diagnosing and accounting for user errors
  - Modeling the system on particular (kinds of) users
  - Presenting useful feedback to the user
Concordancers

- Take a text and create a **concordance** = display of words in context.
- Concordancers help learners understand how a given word is used.
  - For example, is the word *data* in English singular or plural?

```
contract to supply voice and data communications within the Tunnel in
giving control over how much data is sent over the network
humanists to fit their special data to the software, rather
27 mm. But these data are for fourth-year crabs.
```
Language awareness

Making generalizations

What happens when teachers must specify all options for answering an exercise?

(2) Today is November 5. What date is tomorrow?
   Tomorrow is _______________________.

Possible correct answers (among others):

   d. 11/06
   (3) a. Nov., the 6th
       b. the sixth
       c. November, the sixth
       e. 06.11.
       f. 6. Nov.

- Many different ways to misspell any of these options
- Many different possible incorrect answers

⇒ We need linguistic generalizations

- **named entity recognition** = identify special expressions, e.g., dates, addresses, person/company

- Authentic Text ICALL
  - Language awareness
  - Named entity recognition
    - Identify special expressions, e.g., dates, addresses, person/company

- Language awareness
  - Making generalizations
  - What happens when teachers must specify all options for answering an exercise?

- Authentic Text ICALL
  - Language awareness
  - Named entity recognition
    - Identify special expressions, e.g., dates, addresses, person/company
Language awareness

Semantic generalizations

More broadly: refer to classes instead of individual strings
  ▶ Consider fill-in-the-blank exercise modeled on a German exercise in Trude Heift’s E-Tutor system:

(4) John works in New York City, but his family lives in Boston. On the weekend, he drives home. Fortunately, John has a new _________________.

Different options for correctly filling in this blank:
  ▶ **synonyms**: words which mean the same thing, at least in certain contexts: e.g., *car* & *vehicle*
  ▶ Other **lexical semantic relations** between words:
    ▶ **hyponymy**: using a more specific term (**hyponym**), e.g., *pick-up*, *SUV*, or *hybrid car*
    ▶ the more general term *car* is the **hypernym**

Specifying all related words as options in the frame of a FIB exercise would involve a lot of work
  ▶ Would have to be repeated for every new exercise
Language awareness
Morphological generalizations

Additionally, a single word in a language can show up in different forms.

- e.g., **citation form** or **lemma** of *bring* *into* *bring*
  - Also realized as *bringing*, *brought*, *bring*, or *brings*
  - The different word forms and their function are investigated in **morphology**

- Other languages feature richer inventories of forms
  - e.g., 6 forms for one of the verbs meaning *to be* in Spanish: *soy*, *eres*, *es*, *somos*, *sois*, *son*
  - Plus over a dozen other tenses and moods

We would need to spell out the many different forms for each exercise in a CALL system
Language awareness

Syntactic generalizations

Consider exercises where learner can enter multiple words
  ▶ the various word order possibilities result in additional, systematic variation
  ▶ syntax identifies different word order possibilities & the forms words have to appear in

(5) John, the radio is much too loud. Please __________________________!

(6) a. turn down the radio.
    b. turn the radio down.

Many non-English languages allow freer word order
  ▶ capturing all possible word orders is infeasible
Linguistic generalizations can compactly specify the expected correct or incorrect answers
  ▶ NLP tools can be used for this
Adding linguistic analysis

Tokenization

To get lemmas (or anything else), we need to find the words (or tokens)

- A text is simply a very long list of letters
- **tokenization** (or **word segmentation**) = task of finding tokens in a text

Why is this challenging?

1. **Covering ambiguity**: two or more characters may be combined to form one word or not
   - Writing systems of many languages do not use spaces between words, e.g., 要害 in Chinese:
     - Option #1: segment as two words of one character each, meaning *will hurt*
     - Option #2: segment it as a single word of two characters, meaning *vitals*
   - Context determines the segmentation
2. **Overlapping ambiguity**: a given character may either combine with the previous or with the next word

- 布什在谈话中指出 (ex. from Xiaofei Lu)
- Meaning changes depending on which word the second to last character 指 quantities part of

\[
\begin{align*}
\ast \text{布什在谈话中指出} & \quad \text{Bush at talk middle-finger out} \\
\text{布什在谈话中指出} & \quad \text{Bush at talk middle point-out} \\
\text{‘Bush pointed out in his talk’}
\end{align*}
\]

- NB: in Chinese, only the second segmentation option is grammatical
Adding linguistic analysis

Tokenization (3)

Even for English, spaces are not exact:

- e.g., *inasmuch as* and *insofar as*, *in spite of* should be single tokens

1. **Compound nouns** such as *flu shot*:
   
   (7) a. I got my flu shot yesterday.

   b. I got my salary yesterday.

2. **Contractions**: e.g., *I’m*, *cannot*, or *gonna*
   
   ▶ They should likely be treated on a par with *I am*, *can not*, and *going to*

Automatic tokenizers typically have long lists of known words & abbreviations, plus finite-state rules for subregularities
Morphological analysis

- **Lemmatization** = extract the lemma, or stem, of a word. (e.g. lemma of *corpora* is *corpus*.)

- **Morphological generation** = generate different forms of a word based on its lemma and part of speech

These processes are used to:

- help provide drill material for learners
- facilitate dictionary lookup (which can be very difficult otherwise for highly-inflected languages)

GLOSSER, for example, is a system that uses morphological processing to speed up dictionary look-up (100 times faster) (Nerbonne 2003)
Adding linguistic analysis

POS tagging

With tokens identified, we can obtain the general classes of words we want, such as part-of-speech (POS) classes

- e.g., to support **meta-linguistic feedback** messages such as “The sentence you entered is missing a verb.”

Parts of-speech are labels for classes of words which behave alike ... in 3 different ways:

1. **Distribution**: linear order with respect to other tokens, i.e., the slot a word appears in.
   - e.g., for *John gave him ____ ball.*:
     - slot between *him* & *ball* is distributional slot of a determiner such as *the* or *a*
   - For automatic POS taggers, distributional information encoded as statistics about how likely POS (*n*-gram) sequences are to occur
     - One thus needs a corpus already annotated with POS tags (**gold-standard annotation**) to train
     - general field of **supervised machine learning**
Adding linguistic analysis

POS tagging (2)

2. Lexical stem lookup
   - Unambiguous part-of-speech (POS): e.g., *claustrophobic* is only an adjective
   - Ambiguous POS: e.g., *can*
     - auxiliary: *The baby* *can* *walk*.
     - full verb: *I* *can* *tuna* *for* *a* *living*.
     - a noun: *Pass* *me* *that* *can* *of* *beer*, *please*!
   - Words not in the lexicon: a big problem for computers

3. Morphology the form of words
   - Markings (e.g., *suffixes* added to stem endings) encode information only appropriate for particular POS
     - e.g., the *-ed* indicated past tense
   - Inflectional suffixes: information such as tense or agreement (e.g., *-s* on third person singular verbs)
   - Derivational affixes (e.g., *-er* turns verbs into nouns: *walk* – *walker*).
     - Automatic POS-taggers use *suffix analysis* as a fallback step
     - If a word has not been seen before, *suffix analysis* determines the most likely POS
Adding linguistic analysis

POS tagging (3)

Complication: **interlanguage** written by students acquiring a second language

Consider these sentences written by Spanish learners of English (from the NOCE corpus):

(8) a. ... to be **choiced** for a job ...
   b. RED helped him **during** he was in the prison.

- **choiced**:
  - distributionally appears in a verbal slot
  - morphologically carries verbal inflection (*'-ed'*)
  - lexically the stem *choice* is a noun (or adjective)

- **during**:
  - morphologically is a preposition
  - distributionally a conjunction

POS tagging for learner language need to be extended to take into account such potentially mismatching evidence.
Parser-Based (I)CALL

Parser-Based ICALL systems generally fall along the following lines:

- System presents the learner with an exercise
- Learner inputs an answer, possibly with errors, i.e., a potentially **ill-formed** sentence
- The parser processes this sentence
  - Identifying where, if at all, it was incorrect
  - The nature of the error
- Feedback is then presented to the student

We’ll look at three different example systems:

- e-Tutor (German Tutor): Heift & Nicholson
- BANZAI/Robo-Sensei: Nagata
- TAGARELA: Amaral & Meurers
e-Tutor (German Tutor)

e-Tutor (Heift & Nicholson 2001) is used at Simon Fraser University to teach German to students; it is:

- general, i.e., allows for any native language (L1)
- able to capture different kinds of errors . . . because the exercises are very constrained

Student input is put through the following modules and stops with feedback when the first error is encountered

1. String match: if the input matches a pre-defined correct answer, we know it’s good.
   - Prevents time-consuming analysis for perfect answers
2. Punctuation check: is any punctuation missing?
More on system architecture

3. Spell check: run an off-the-shelf spell checker on the input and get the **lemmas**
   - Idea: eliminate the really basic errors.
   - Problem: sometimes a “misspelled” word is a sign of lack of grammatical competence, e.g. *runned*

4. Example check: are the right words being used?

5. Missing word check: are any words missing?

6. Extra word check: are any words added?
   - These 3 steps (example, missing word, and extra word checks) all are based on the notion that the exercise has *pre-defined* all the acceptable words
More on system architecture (cont.)

7. Word order check: match the user word order with the correct word order

8. Grammar check
   ▶ This is the most complicated part of the process, the one which requires linguistic knowledge (syntax)
   ▶ About 60% of errors make it to this stage.

9. Catch-all: just in case everything else fails

Note:
   ▶ Heift’s system works so well because the exercises themselves are constrained, as we will see
   ▶ The approach is very modular = each check is an independent program
Kinds of exercises

Here are some example exercises from Heift’s system, outlined in http://zif.spz.tu-darmstadt.de/jg-06-2/beitrags/heift2.htm

- Dictation
- Build a Phrase
- Which Word is Different
- Word Order Practice
- Fill-in-the-Blank
- Build a Sentence
Student hears a sentence in German and types it in. They are told if they are correct, and if not, why.
Dictation (cont.)

Pros:
- Input is very constrained.
- Very useful to be able to practice listening by oneself.
- Won’t take up class time.

Cons:
- Requires multimedia resources & may take awhile to prepare.
- Not very contextualized.

Such good and bad points could be given for all of the following exercises.
Build a Phrase

Build up a complete phrase (e.g., a noun phrase) based on a given picture—in other words, provide your own vocab.

![Image of croissants with German text: Guten Tag, Trude! Schreiben Sie das Substantiv mit Artikel. Übung 2 von 10. die Croissants Prüfen Lösung Weiter >>. Prima!](image-url)
Which Word is Different

Given 4 words, pick the one which differs from the others.

Guten Tag, Trude!
Was passt nicht? Klicken Sie das Wort und dann "PRÜFEN".

Übung 5 von 10
Sommer Uhr
Frühling Winter

Gut gemacht!
Take all the given words and arrange them into a sentence.

Guten Tag, Trude!
Ziehen Sie die folgenden Wörter in das Textfeld.

Übung 5 von 15

einem oder Wohnung? einer Haus

Wohnst du in

Prüfen
Lösung
Neu laden
Weiter >>
Fill-in-the-Blank

Give the learner a lemma or choice of lemmas, and they have to fill in the blank.

Guten Tag, Trude!

Schreiben Sie die fehlenden Wörter.

Übung 3 von 10

Fumiko wohnt in München.

Toll!
Build a Sentence

Use all the given words (lemmas) and create a grammatical German sentence.

Advanced learner output here: “There is an error in gender with the subject.”
Robo-Sensei is proprietary software developed for teaching introductory Japanese courses

▶ Exercises cover a range of Japanese constructions
  ▶ Extensive metalinguistic feedback is given to the learner
  ▶ Noriko Nagata has shown, to some extent, that more difficult constructions are better served with this kind of feedback

▶ Hand-written rules cover Japanese morphology and syntax patterns
  ▶ Comparison is done between the analysis of a learner’s input and a target analysis, in order to spot the error(s) and provide effective feedback
TAGARELA is a system for individualized instruction of Portuguese at Ohio State

▶ It features standard exercises, as found in foreign language workbooks
▶ NLP processing is used to detect spelling, morphological, syntactic, and semantic errors
▶ A student model is kept to track performance and to choose appropriate feedback
  ▶ An instruction model allows the instructor to state what is important
TAGARELA system overview

Expert Model
- Segmentation model of language
  - Tokenizer
- Non-word spell check
- Shallow Parser
- Deep Processing
- Lexical look-up/Morpho analysis
  - Lexicon

Content modules
- correct answer
- required words
- required stems
- WN module

INPUT
- Form Analysis
  - Content Analysis

Analysis Manager

Instruction Model
- Activity Info
  - Exercise requirements
  - Feedback requirements
- Error Info
  - Error Taxonomy (E.T.)
  - E.T. and feedback

Student Model
- Student info
  - Name, gender, level, etc
- Grammatical competence
  - Performance by knowledge units
- Strategic competence
  - Performance by task environment

Output
- Feedback Generation
- Feedback Manager (pedagogical modules)
  - Error Filtering
  - Ranking
- Feedback Manager (pedagogical modules)
  - Error Filtering
  - Student analysis
  - Feedback selection

ANNOTATED INPUT

Non-word spell check
Segmentation model of language
Shallow Parser
Deep Processing
Lexical look-up/Morpho analysis
Lexicon
Content modules
- correct answer
- required words
- required stems
- WN module

Instruction Model

Feedback Manager (pedagogical modules)
- Error Filtering
- Ranking

Student Model
- Student info
  - Name, gender, level, etc
- Grammatical competence
  - Performance by knowledge units
- Strategic competence
  - Performance by task environment

Annotation Input
Demand-driven architecture

Different from the e-Tutor, TAGARELA works in a **demand-driven** fashion; the analysis manager:

- receives input from the student
- gather the necessary information from:
  - instruction model
  - student model
- decides on the best processing strategy
  - which NLP modules to call
  - in which order (as opposed to linearly)
- calls NLP modules to process input, producing an input annotated with linguistic properties
- hands the annotated input to the feedback manager
Sources of information for CALL systems

1. Language/linguistic properties
2. Exercise information
3. Information about the learner
Modeling the learner

**Learner modeling** generally includes two types of information:

1. Learner properties which are more or less permanent
   - e.g., gender, native language, learning style preferences –

2. Dynamic record of learner performance so far: whether a learner successfully used particular words or structures

Both types of information are relevant for feedback

- e.g., native language (L1) of a learner influences words & constructions used & mistakes made
  - Positive and negative **L1-transfer**
  - Negative transfer: many native speakers of languages such as Chinese or Czech, find *the* & *a* difficult
    - L1s do not include articles of the kind found in English
    - Tutoring system should provide feedback on article misuse for learners with such native languages
    - For other users, may be a minor typo
Modeling the learner

Obtaining learner information

How do we obtain dynamic record of learner performance?

- The system needs to draw **inferences** from the learner’s interaction with the system.
  - Need to abstract to general linguistic properties & classes which a learner answer provides evidence for
    - e.g., whether a learner answer contained a finite verb, provided evidence for subject-verb agreement, etc.
  - After seeing answers with instances of a particular property, we can infer that the learner has mastered it
    - e.g., deprioritize feedback on it in the future
- Models may help **sequence teaching material**
  - e.g., by guiding the learner to additional material on concepts not yet mastered

An intelligent tutoring system needs to take into account learner strategies

- e.g., avoidance of structures the learner is unsure about
The parser-based systems we’ve just seen have to deal with errors in learner input:

▶ How do we adapt our technologies to find errors?
▶ Do we tailor the system to a particular kind of learner?
▶ Do we tailor the system for an individual learner?
▶ What is the exact error?
▶ How many errors are there?
How do we adapt technology to find errors?

Our parsers, morphological analyzers, and so on, are made to handle well-formed input.

- Use so-called **mal-rules** = rules which are added to your grammar that handle error cases.
  - e.g., A singular noun and a plural verb are allowed to combine, but it is marked as an error.
  - \( S_{\text{error}} \rightarrow \text{NP}_s \text{g VP}_p \)
- Modify your technology: a parser can be reworked to handle ill-formed input.
  - Parsers normally just “die” when handling bad input.
  - e.g., I’ll parse *John are big*, but I’ll tell you that I didn’t like it and where it went wrong.
Do we tailor the system for an individual?

Do we keep track of a student model = what level each student is at, for a given task?

- Allows us to say which grammatical points need more work for a given individual.
- Allows us to give different feedback based on the learner’s abilities.
- Make sure the learner knows the terminology presented in the feedback, e.g., for the input *John are big*.
  - Beginner: “*John* is a singular subject and *are* is a plural verb.”
  - Intermediate: “There is a subject-verb agreement error.”
  - Advanced: “There is an error.”
Constraining the domain

Systems like the e-Tutor work well because they constrain what it is that the students can talk about.

- Input is fairly free, but students select from a pool of vocab and grammatical structures.
- Students are fairly introductory, so no exercise is too complex.
  - Compare teaching people subject-verb agreement vs. teaching them counterfactual clauses in English.
- All the pre-processing steps (before the grammar check) allow the grammar checker to assume well-formed input.
Authentic Text ICALL attempts to connect learners to appropriate naturally-occurring texts

- Allows students to find examples in target language related to their interests
- Allows for more exploration and something akin to “immersion”

There are challenges in:

- Extracting appropriate exercises from the real texts
- Finding level-appropriate texts
The WERTi System
Working with English Real-Texts: An Intelligent Workbook for English

WERTi is an “intelligent automatic workbook, providing an unlimited number of activities designed to foster awareness of English grammatical forms and functions”

▶ Learners select a topic which fits their interests
▶ Webpages are returned, which learners interact to learn about, e.g., prepositions
    ▶ Learners can choose to see prepositions in color; click on them; or fill in blanks

Crucially, the exercises are **generated** on the fly
▶ Pre-existing NLP technology (e.g., a POS tagger) is used to spot the relevant categories
The REAP Project
Reader-Specific Lexical Practice for Improved Reading Comprehension

In the REAP system:

▶ Teachers have target vocabulary items
▶ REAP finds appropriate texts for learners, based on their individual profile
  ▶ Learners get individualized vocabulary practice from authentic web texts

There are several challenges in extracting text for reading

▶ Each extracted text is analyzed for its “syntactic features, readability, length, and the occurrence of target vocabulary”
▶ Information retrieval and statistical NLP techniques are used to find appropriate texts