LG 467 Computers in Linguistics

[1-2021] Topic 2: Text Normalization

Sakol Suethanapornkul

This lecture is based on Dickinson et al. (2013)

Computers in LT

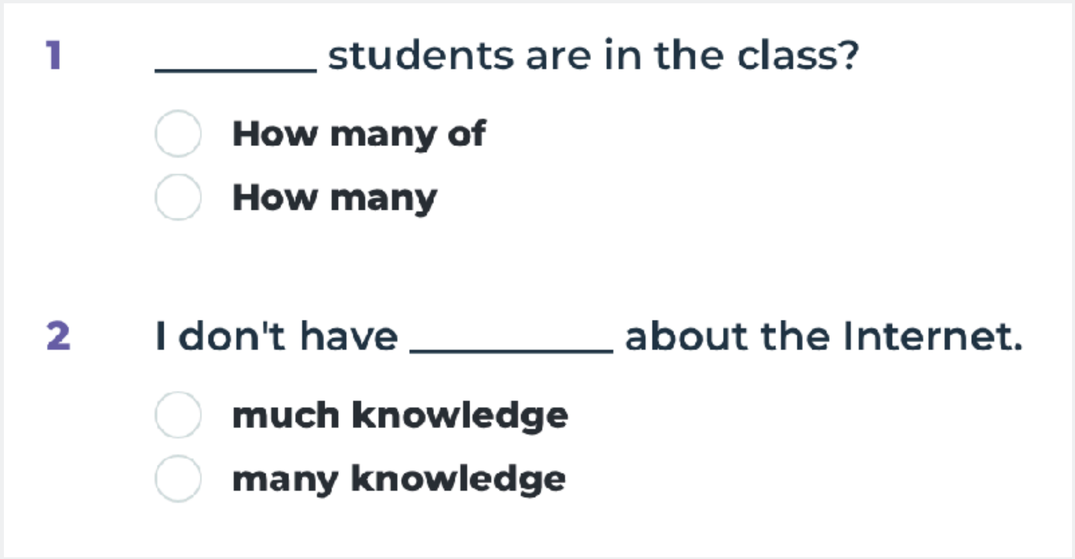
Computer-Assisted Language Learning (CALL) tools can provide support for language learning outside class

But existing CALL systems (in many language programs) which offer language exercises

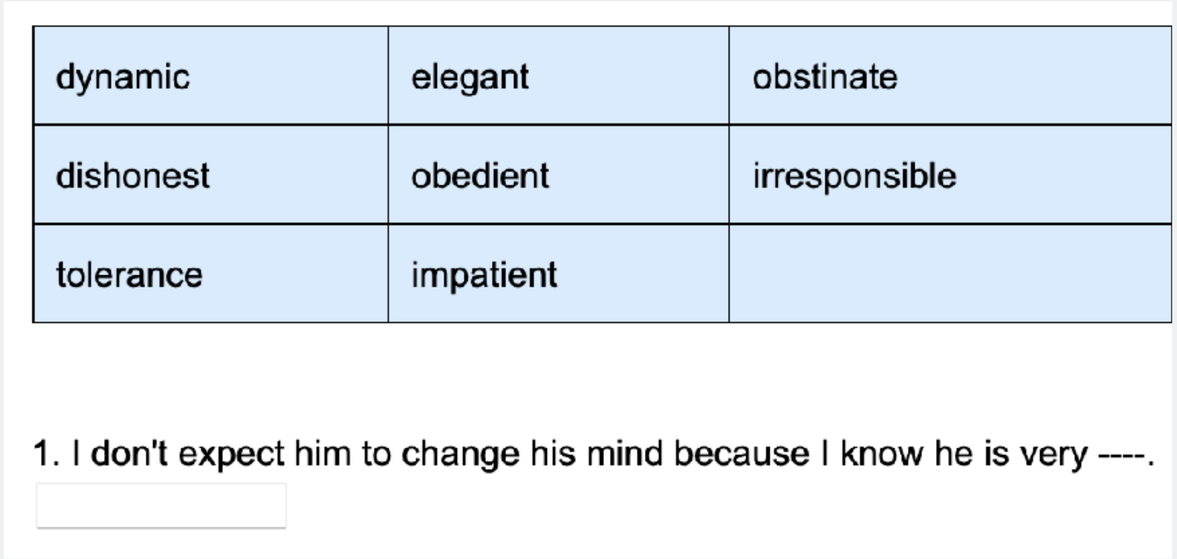
- typically are limited to decontextualized multiple choice, point-and-click, or simple form filling, and
- feedback usually is limited to yes/no or letter-by-letter matching of the string with a pre-stored answer

Computers in LT

Example: multiple-choice and fill-in-the-blank exercises



Source: Dave's ESL Cafe website



dynamic	elegant	obstinate
dishonest	obedient	irresponsible
tolerance	impatient	

1. I don't expect him to change his mind because I know he is very ----.

Source: GrammarBank website

Computers in LT

A better system requires linguistic knowledge & generalizations

Exhibit A: Today is November 5. What date is tomorrow?
Tomorrow is _____.

Possible correct answers:

06/11, 11/06 Nov., the 6th, the sixth, November 6…

Named Entity Recognition (NER) = identify special expressions, e.g., dates, addresses, names

Computers in LT

A better system requires linguistic knowledge & generalizations

Exhibit B: John works in New York, but his family lives in Boston. On the weekend, he drives home. Fortunately, John has a new _____.

Possible correct answers: **Synonym** (*car* and *vehicle*), **hyponym** (*SUV, pick-up, hybrid car*) or **hypernym** (*car*)

Computers in LT

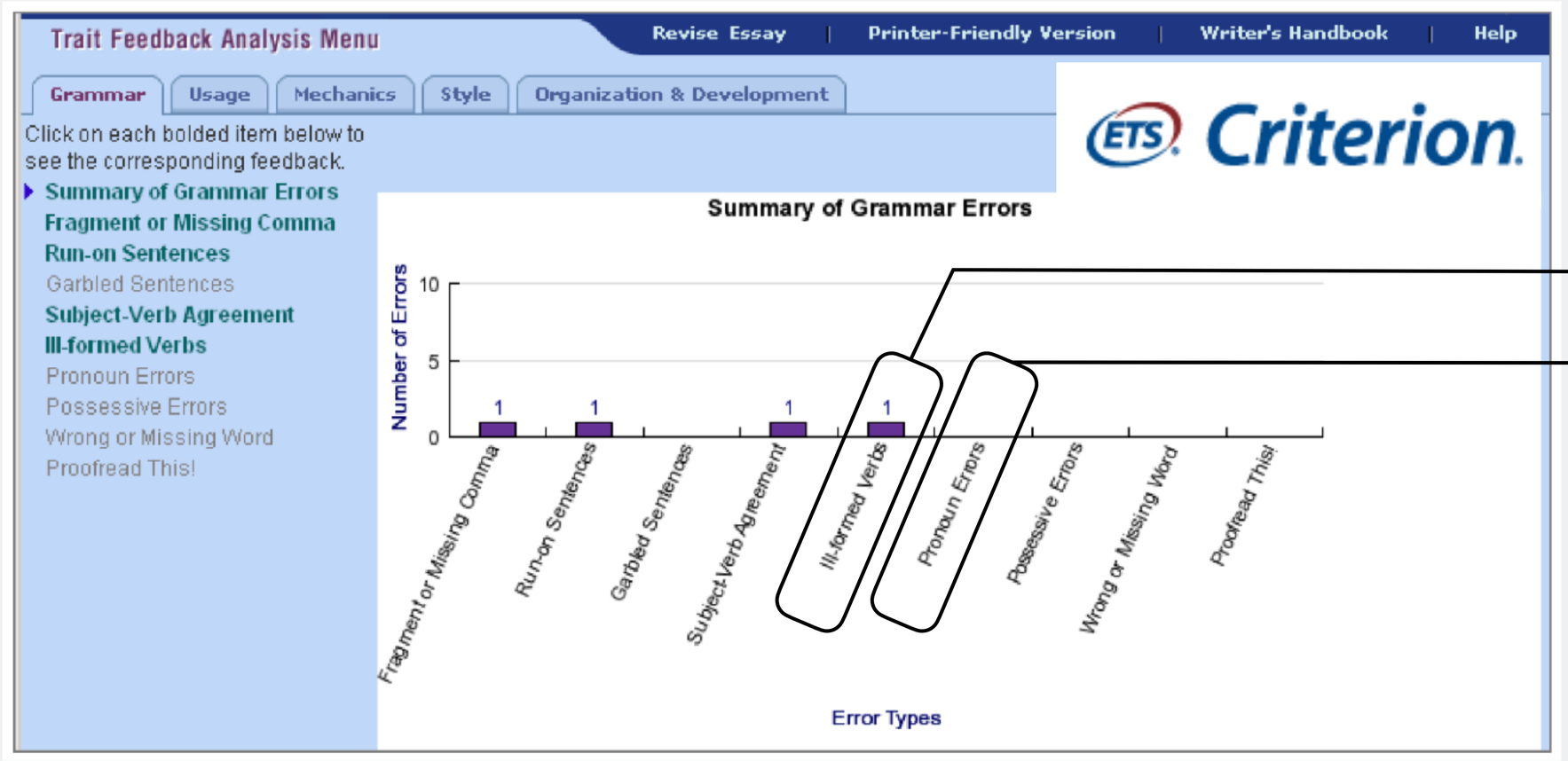
A better system requires linguistic knowledge & generalizations

Exhibit C: John _____ in New York, but his family _____ in Boston.

A single word can show up in different forms. A **lemma** *to live* can be realized as *live, lives, lived, living*

Computers in LT

ICALL systems that are aware of language exist:

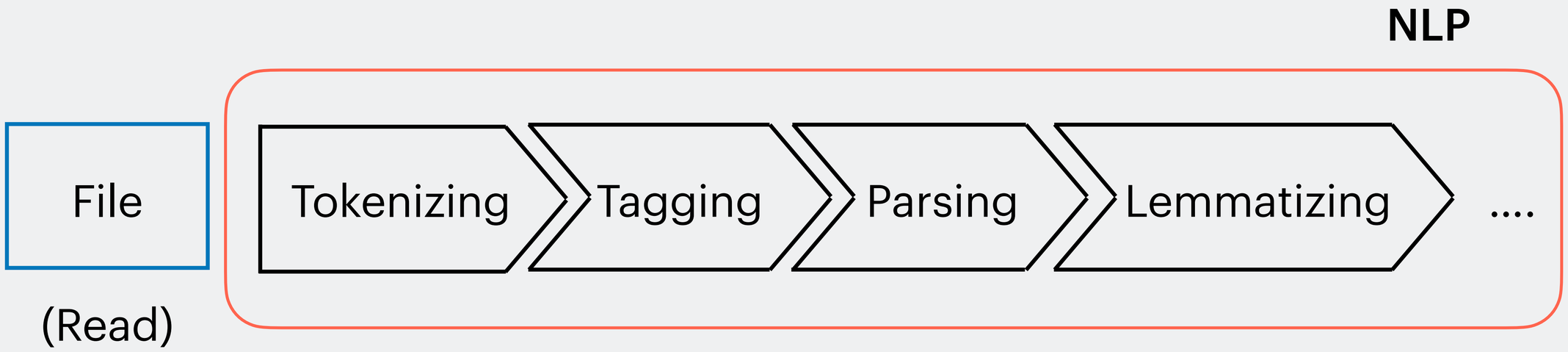


Source: ETS Criterion review

Text Normalization

NLP pipeline

- A processing pipeline in NLP:



Tokenization

- A text is simply a very long sequence of letters
- One of the first steps in dealing with a text in NLP is to divide it into minimal units (**tokens**)
  - **Tokenization** (or word segmentation) is finding tokens in a text

Tokenization

- Why is this challenging?
  1. **Covering ambiguity**: Two or more characters may be combined to form one word
     - Writing systems of many languages don't use spaces between words

     มีนา #1: two words, meaning owning a rice paddy

     #2: one word, meaning March
- Context determines the segmentation

Tokenization

- Why is this challenging?

  2. **Overlapping ambiguity**: A character may either combine with the previous or with the next word

#1	พ่อ	กิน	ข้าวเช้า	
	dad	eat	breakfast	
#2	พ่อ	กิน	ข้าว	เช้า
	dad	eat	rice	early

Tokenization

- Even in English, tokenization is a non-trivial problem
- Spaces are not exact:

1. Compound nouns such as *flu shot*
   a. I got my *flu shot* yesterday.
   b. I got my *salary* yesterday.
2. Phrases such as *inasmuch as*, *insofar as*, and *in spite of*

Tokenization

- Even in English, tokenization is a non-trivial problem
- Spaces are not exact:

3. Contractions such as *I'm*, *cannot*, *can't* or *gonna*
    - They should likely be treated on a par with *I am, can not, and going to*
    - If *not* is a word, should *n't* be one too? (Answer: Yes!)
    - In NLP, English has a modal verb *wo* (*I wo n't do it*)

Tokenization

- Automatic tokenizers (e.g., NLTK) typically have long lists of known words and abbreviations, plus (finite-state) rules for subregularities

Our plan next week!

- Tokenization in practice
  - a string method `.split()`
  - Python list
  - NLTK (`from nltk.tokenize import word_tokenize`)
- Lemmatization & stemming (time permitting!)