DOI 10.7603/s40742-015-0001-6

Automatic and Semi-Automatic Test Generation for Introductory Linguistics Courses Using Natural Language Processing Resources and Text Corpora

Peter Wood

Received 21 May 2015 Accepted 22 Jun 2015

Abstract **- This paper describes a collection of Natural Language Processing (NLP) modules which automatically generate exercises for introductory courses on structural linguistics and English grammar at a Canadian University.**

While there is a growing demand for electronic exercises, online testing tools, and self contained linguistics and grammar courses, the exercises and tests offered on companion websites for popular textbooks consist largely of multiple choice type questions.

The modules create exercises to practice and test part-of-speech identification, morphological analysis of complex words, and the analysis of sentences into phrase structure trees. They are part of an infrastructure capable of delivering instructional material, exercises for for self assessment, and online testing tools for courses which either use blended instruction or are taught exclusively online.

Modules which are work in progress will be briefly discussed in the final section of this paper.

Keywords: Natural Language Processing, Online Testing, Computational Linguistics, Teaching Linguistics, Corpus Linguistics

I. Introduction

In the domain of Computer Assisted Language Learning (CALL) [1], and more specifically in the area of Intelligent Computer Assisted Language Learning (ICALL) [2], much research and development has been dedicated over the last 20 years to turn computers into intelligent language tutors, in the sense that they are meant to provide meaningful input, exercises and feedback geared, ideally, to the specific needs of the individual student.

This paper discusses the creation of meaningful exercises related to natural language. However, they are not geared toward learners of a second language, but to beginning students of linguistics. The need to develop the online resources outlined in this paper arose from the requirement to teach introductory linguistics and grammar classes to large numbers of students with a variety of different backgrounds in linguistics, language learning, and an overall level of general education with very limited resources in terms of instructors, tutorial leaders, and class time.

In CALL, descriptions of institutional settings similar to these are often considered valid reasons which justify investments into CALL technologies. And, as developments in CALL, ICALL, and Automatic Essay Scoring (AES) show, these investments have started to pay off [3].

Considering the availability of Natural Language Processing (NLP) based resources for linguistics courses, it might come as a surprise to learn that such resources are basically non-existent. A look at the online resources that many text book publishers offer as companion material for their traditional text books shows that the additional practice and exercise sections on the pertinent websites are largely limited to multiple choice questions, true false questions, or fill in the gap exercises. Questions asking students to complete more complex tasks such as analyzing the structure of a compound are not machine gradable and require a human corrector. This fact makes more complex exercises unsuitable for inclusion in large scale testing due to limited human resources. This is especially surprising since many NLP resources can be readily exploited to be used for the creation of meaningful exercises for many different areas of linguistics. The aim of this paper is to demonstrate just this.

II. Remarks on the Courses

The course for which the individual modules were originally developed is an introductory linguistics course delivered over one term. It covers the basic principles of structuralist phonetics, phonology, morphology, morphology, syntax and semantics.

While this course serves as a foundations course for students enrolled in a linguistics BA program, it is also popular among students from a wide variety of other disciplines who are taking it to fulfill a requirement in the Humanities, including a large number of foreign students with limited English language skills.

Given the heterogeneous make-up of classes, it is not surprising to find that many students do not have a background in the formal analysis of language, and require practice to acquire the basic skills necessary to successfully complete the course.

DOI: 10.5176/2345-7163_3.1.62

Published online: 12 December 2015

The modules presented here address common problems in morphology and syntax: identifying word classes, analyzing words and sentences into their constituents.

We have found that students enrolled in a beginner level grammar course also benefit from using some of the exercises, given that classes here have a similar make-up and topics covered do overlap in some respect.

The modules started out as small projects developed for a NLP course taught to 4th year linguistics students, demonstrating common NLP tasks. Over the last three years they have been integrated gradually into the infrastructure used to deliver the courses online.

III. Remarks on the Infrastructure

The modules presented below form part of a larger infrastructure that is used to deliver most of our linguistics courses. At its core is a web interface based on Twitter Bootstrap which aims to be easy to use and providing a common user experience across different browsers and devices. Using the interface, users are able to access course materials, lecture notes and slides, as well as online exercises for self testing and graded quizzes. Quizzes consist of traditional multiple choice questions and exercises generated with the NLP modules.

The web interface is programmed using the Angular.js Javascript framework. It is delivered by a Node.js server which makes static resources available and serves dynamic web content by handling client side Ajax requests. It is used to interface with the NLP modules to deliver exercises, to check answers, and to provide feedback messages. The server is connected to a Mongodb database system which stores course resources, quiz questions, persists student data such as answers to questions and quiz results, and logs.

IV. Creating Part-Of-Speech Identification Exercises

Both introductory courses aim at teaching students to identify the parts of speech of the words in an English sentence, using the standard part-of-speech categories, such as noun, verb, adjective, etc.

As most other modules described here, the POS test module uses the Natural Language Toolkit (NLTK) [4], a Python library which provides a natural language processing infrastructure, including tokenizers, POS-taggers, chunkers, parsers, various machine learning algorithms used for statistical parsing and tagging, and a collection of corpora.

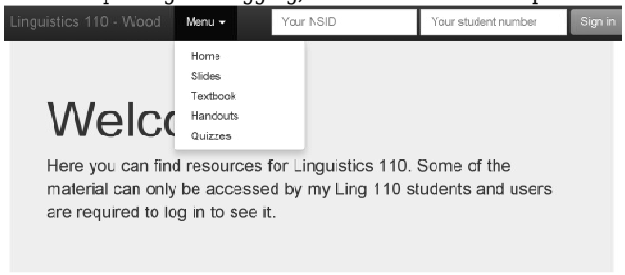


Fig. 1: User Interface

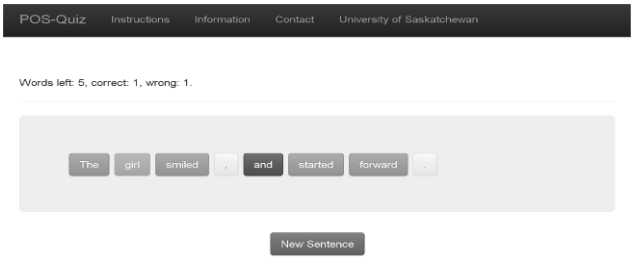


Fig. 2: Part of Speech Quiz - Practice Mode

Part of speech exercises can be created using sentences from the POS-tagged Brown corpus [5], which is part of the NLTK.

The Brown corpus was originally tagged using the CLAWS tag set. It provides a high level of granularity and is important for many NLP tasks. However, this high degree of granularity is unsuitable for pedagogical purposes. Therefore, we use the so called simplified tag set which - after some modifications - maps neatly to the traditional word classes introduced in many grammar books and introductory linguistics texts:

- noun
- pronoun
- adjective
- adverb
- conjunction
- determiner
- preposition
- verb

Verbs are subclassified into participles, past tense verbs, modal verbs, and others (present tense and bare infinitive forms of non-modal verbs).

Apart from the Brown corpus, we are also using sentences from manually assembled corpus. The custom corpus does not aim to be balanced, but instead is intended to provide contemporary Canadian texts and contains web texts from the web sites of Canadian news papers, tv stations, and academic institutions. To tag it, we use a Bayesian part of speech tagger trained on the tagged Brown corpus. It performs at roughly 98\% precision using the simplified tag set. This corpus is used to correct the bias of American spelling and 1960's English introduced by using the Brown corpus as our only source of input and is intended to replace the Brown corpus entirely once it is sufficiently large.

V. Dealing with Mis-Tagged Words

Our custom corpus of Canadian English, as well as the tagged version of the Brown corpus included with the NLTK contains sentences with tagging errors. We estimate that one out of every seven or eight sentences contains at least one tagging error. While they do not seem to cause major problems for students who practice part-of-speech identification using the ungraded exercises, they may have a negative effect on test

scores if they are used in automatically generated exercises for the graded quizzes. In order to provide a mechanism to exclude sentences with faulty tags from quizzes the following system was created:

Table 1: Internal Representation of Sentences

token	tok	tag	wrong
1	She	PRO	0
2	likes	V	0
3	Picasso	PrN	0
4	.	PCT	0

- After a sentences has been tagged, it is recorded in a non-relational database. The sentence is stored as an array of token objects. token objects have the format outlined in table 1. For every token in the sentence, we create a triplet consisting of the token itself(tok), the tag assigned by the POS-tagger (tag), and an extra field called wrong,initialized to 0. In addition, we store one more field per sentence, called completed and initialize this to 0, as well.
- The data base contains three collections. One collection is called: newEntries. Tagged sentences are initially added to this collection.
- In addition to this collection, there are two additional collections: verified} and testing.
- When users start the part of speech exercise, five sentences each are picked at random from the newEntries collection and from the testing collection.
- If a sentence is completed (the user has tried to identify the part of speech for each token) completed for the sentence is incremented.
- If completed is zero, the sentence remains in newEntries.
- Otherwise, the wrong counters for the tokens misidentified are incremented, and the sentence is moved to testing if completed is less than 10.
- If the completed counter reaches 10 and none of the wrong counts exceeds 3, the sentence is moved to verified, otherwise it is removed from the database altogether (actually, we are saving it in another database for future use to increase the tagger's accuracy).

This heuristic prevents any sentence from being used in graded quizzes if at least 4 out of 10 users misidentified the token. This allows for "noise": tokens which are tagged correctly but misidentified

erroneously by some students. So far, the heuristic has performed reasonably well and the number of sentences with tagging errors in the `verified` collection used for graded tests seems to be low judging by student feedback on quiz results.

VI. Creating Morphological Analysis Exercises

The inflectional morphology of English is, arguably, not very complex compared to other languages [6]. Its derivational morphology,however, is very rich.

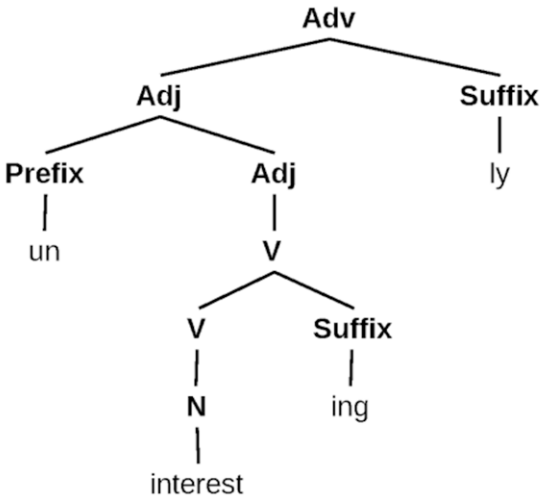
One of the objectives of an introductory linguistics course is to teach students how to analyse the inner structure of complex words systematically. For example, students should learn to analyze a word such as "uninterestingly" into its individual parts and be able to infer what word formation processes are at work at specific stages during the construction of the complex form.

Although the word is extremely rare - it occurs only five times in the Corpus of Contemporary American English [7] - it is the result of productive word formation processes.

Figure 3 provides a representation of the word's inner structure. It can be interpreted the following way:

- The noun interest is turned into a verb via conversion.
- /-ing/ attaches to the verb to form the present participle.
- The present participle is converted to an adjective.
- /un-/ attaches to the adjectival base, no change of word class.
- /-ly/ is added to the end of the adjective to turn it into an adverb.

Fig. 3: Representation of the Internal Structure of "uninterestingly"



While morphological analysis using NLP is well researched [8, 9] and commercial tools for the analysis of words' internal structure perform robustly [10] and have been available for some years now, it is interesting to note that none of the well known open source NLP libraries include tools to perform deep morphological analysis on complex English words.

NLTK provides stemmers and lemmatizers for English, but these do not perform a deep morphologic analysis. In most NLP tasks the internal structure of words is of minor concern. Morphological analysis is an intermediate step, largely

performed to help in the identification of the part-of-speech, as well as inflectional affixes which determine the syntactic structure of the sentence and therefore provide important information exploited by syntactic parsers.

For our project, the creation of morphological analysis exercises is currently semi-automatic. We started out using routines to find morphologically "interesting" words in the NLTK corpus collection using parameters such as word length and the presence of certain derivational affixes as heuristics. The output of these routes was then manually analyzed and stored in labeled bracketing format in the database. This format serves as the basis to create NLTK `tree` objects. The NLTK API can then be used to to perform various processing tasks on the graphs, such as exploring and modifying their inner structure, and creating tree diagrams similar to the one in figure 3 dynamically.

One important desideratum for the analysis is that each node in the graph representing the word structure actually represents a possible and - ideally - attested form. In figure 3 for example, we claim that /un-/ attaches to the adjectival base /interesting/. The graph implicitly implies that a noun, verb, or adjective /uninterest/ does not exist, or, at least, would be very rare, as it does not follow regular word formation patterns. Such claims can be tested by looking up possible forms in a large enough corpus and by comparing individual frequencies. This provides an empirical basis for analyses were more than one structure is theoretically possible. In these cases the frequency counts serve as a tie breaker, and preference is given to the analysis with the higher frequency counts of word forms hypothesized at intermediate levels.

We are gradually improving the performance of morphological analyzer based on finite state machines which is currently able to perform automatic analysis for a fragment of English derivational processes and which automatically checks for the frequency counts of possible intermediate forms in the NLTK corpora in order to make branching decisions in the manner outlined in the previous paragraph.

In the exercises created by the module, students are provided with the word already split up into its individual parts. It is their task to combine them in the right order, label nodes correctly, and by so doing reconstruct the graph from the bottom up.

VII. Creating Syntactic Analysis Exercises

When it comes to syntax, it seems that many textbooks try to provide students with a basic understanding of Chomskian syntax along the lines of the Government and Binding framework [11], while excluding more recent developments in theoretical syntax, even in the Chomskian tradition like Minimalism [12], see for example [13].

While this approach has its merits - it certainly teaches students syntactic theory that has been influential for almost any contemporary syntactic theory, it can be argued that it also has some disadvantages:

- teaches outdated syntactic theories.
- The level of abstraction often proves fairly difficult for beginners.
- These theories do not easily integrate with contemporary theories that are used in more advanced courses.

Our approach is to provide a non-derivational, non-transformational account of syntax where tree structures are flat, binary branching is not a necessary requirement, and deeply embedded structures commonly seen in X-bar theory are avoided. S forms the highest node in a sentence, and the existence of abstract phrases such as IP, TP, and CP is not assumed. In this regard, our syntactic analysis is similar to the structural analysis in the Simpler Syntax framework [14], but we are not introducing any of the other tiers assumed in that framework, such as the CS tier. This is possible because we are not considering complex sentences, inversion or other phenomena which have to be explained in some way.

Simpler Syntax takes a lexicalist perspective and accounts for many phenomena on the word level. The form of sentences is then determined by interaction of the CS tier with the grammatical function tier, and others. This way, Simpler Syntax, like other modern frameworks, does not require transformations to rearrange syntactic structures.

Apart from minimizing the complexity of syntactic analysis which has a pedagogical benefit, this has the advantage that the syntactic model can easily be extended with lexical, or semantic components and integrates easily with many non-derivational approaches, such as Simpler Syntax, Head-Driven Phrase Structure Grammar [15], or Construction Grammar [16] that students will be introduced to in advanced courses.

We are using the `treebank` corpus from the NLTK, which comprises 10\% of the parsed sentences from the PENN tree bank [17]. While this provides us with just over 3,900 sentences, not all of them are suitable because of the texts which were used for the creation of the corpus. The PENN tree bank fragment we are using comprises sentences collected from the Wall Street Journal. Consequently, many sentences contained in the corpus are complex, containing coordinated and subordinated clauses, as well as complex phrases using adverbials, reduced clauses, participial constructions, etc. which are unsuitable for beginning students.

We want to constrain the set of sentences used for the creation of exercises to simple sentences consisting only of one independent clause, have no subordinate clauses, and which display the word order of canonical English sentences.

Parse trees of the PENN treebank have a flat structure, and by filtering out complex sentences, we are left with trees which are fairly close to the tree structures used in Simpler Syntax. Differences exist, of course.

For example, the PENN trees make modals and auxiliaries daughters of the highest level VP, while Simpler Syntax makes them daughters of S and sisters to the highest level VP.

Simpler Syntax uses a Tense node, which is also a daughter of S, and which accounts for tense marking on the modal, auxiliary, or the main verb via affix hopping, We consider this

to be beyond the scope of an introductory linguistics course, we deviate from this analysis and follow the PENN treebank analysis.

Tree structures are represented by a labeled bracketing scheme in the PENN treebank. As already mentioned, the NLTK uses a `tree` object to represent the structures. The `tree` API exposes the structure of the sentence as a list of lists. Each list represents the structure of one node. Developing an algorithm to find sentences suitable for inclusion in exercises, therefore, is straightforward:

- We recursively descent into every list.
- If we encounter a node label indicating an embedded clause, question, indirect question (S-BAR, etc.) the algorithm enters a fail state and we move on to the next sentence.
- We modify node names, so that the names used reflect those generally used in introductory linguistics courses: NP, PP, VP, etc.)
- We keep track of the overall count of nodes and levels of embeddedness, so that we are able to exclude sentences which are potentially too complex based on these criteria.

The resulting tree representations can be stored in a similar fashion as the representations of complex words, and the creation of exercises works analogously. Students are given the sentence and are asked to recreate the sentence structure from the bottom up.

VIII. Integration

Exercises created with the three modules described above are available to students at any time. By completing the exercises they can practice doing real linguistic analyses at a level appropriate for their stage of proficiency.

It also helps them to get accustomed to the test format and to working with the user interface in the graded quizzes. As mentioned already, quizzes consist of a mix of multiple choice questions and some of the automatically created exercises.

Students are required to take four quizzes:

- Phonetics and Phonology: 30 multiple choice questions and 10 IPA transcription exercises. For the latter, students are provided with a word in standard orthography and are required to transcribe it. IPA symbols are displayed in the user interface and can be clicked to facilitate the entry of the symbols. Student responses are evaluated by simple pattern matching (multiple possible correct answers are checked were appropriate). Partial credit is given for every correct symbol.
- Morphology: 30 multiple choice questions, 5 POS exercises, and 5 morphological analysis exercises. Partial credit is given for correctly identified POS-tags, or nodes in the morphological analysis task.
- Syntax: 30 multiple choice questions, 10 syntactic analysis exercises. Partial credit is given for every correctly identified and labeled node.
- Semantics: 50 multiple choice questions.

In order to account for the fact that the complexity of the exercises individual users will be given will vary, the mark for the NLP based exercises is calculated on the basis of the ratio of correct and incorrect responses.

IX. Work in Progress

While work on the individual modules is ongoing as outlined above, we are also trying to create more modules for other areas of linguistics.

Examples include:

- PSR evaluator: The idea behind this module is to provide students with a simple vocabulary and a number of sentences that a phrase structure grammar for this set of words should license. Their task is to write a context free phrase structure grammar which is evaluated by a chart parser and students will receive partial credit for every correct sentence their grammar licenses.
- Semantic networks. This exercise is based on the Wordnet corpus which is part of the NLTK. The module retrieves a synsent for a random word and displays it to the user. The task is to recreate the relations between the member of the synset correctly, which requires users to mark words as hypernyms, hyponyms, and holonyms

We are also working on a module that allows students to create mini feature grammars to be used in upper year courses on Simpler Syntax, HPSG, or Construction Grammar.

X. Conclusion

We have demonstrated that NLP resources can be used to build applications for students in undergraduate linguistics courses fairly easily.

There are almost no comparable resources available, as far as we know (at least none that do not require users to have a considerate level of computer literacy, as for example the Grammix system by Stefan 18, and are therefore unsuitable to be used in general beginner courses). And we hope that both developers and text book publishers will become aware of this gap and address it.

References

[1] P. Hubbard (Ed.), Computer Assisted Language Learning, volumes 1–4. New York: Routledge, 2009.

[2] T. Heift and M.Schulze, Parsers and pedagogues. Errors and intelligence in computer-assisted language learning. New York: Routledge, 2007.

[3] N. Garrett., "Computer-assisted language learning trends and issues revisited: Integrating innovation," in The Modern Language Journal, 93(s1), 2009, pp. 719–740.

[4] S. Bird, E. Klein, and E. Loper, Natural language processing with Python. Sebastopol(CA): O'Reilly, 2009..

[5] W. N. Francis and H. Kucera, Frequency analysis of English usage. Lexicon and grammar. Boston: Houghton Mifflin, 1982.

[6] M. Aronoff and K. Fudeman, What is morphology? 2nd ed. Oxford: Wiley, 2010.

[7] M. Davies, "The 385+ million word corpus of contemporary American English (1990–2008+): Design, architecture, and linguistic insights," in International Journal of Corpus Linguistics, 14(2), 2009, pp.159–190.

[8] D. Jurafsky and J. H. Martin, Speech and language processing. An introduction to natural language processing, computational linguistics, and speech recognition, 2nd ed. Upper Saddle River(NJ): Prentice Hall, 2010.

[9] B. Roark and R. W. Sproat, Computational approaches to morphology and syntax. Oxford: Oxford University Press, 2007.

[10] K. R. Beesley and L. Karttunen, Finite-state morphology: Xerox tools and techniques. Stanford: CSLI, 2003.

[11] N. Chomsky, Lectures on government and binding. Dordrecht(Netherlands): Foris, 1981.

[12] N. Chomsky, A minimalist program. Cambridge(MA): MIT Press, 1995.

[13] W. O'Grady and J. Archibald, Contemporary linguistic analysis. 7th ed. London: Pearson, 2011.

[14] P. W. Culicover and R. Jackendoff., Simpler syntax. Oxford: Oxford University Press, 2005.

[15] C. Pollard and I. A. Sag, Head-driven phrase structure grammar. Chicago: University of Chicago Press, 1994.

[16] A. E. Goldberg, Constructions. A construction grammar approach to argument structure. Chicago: University of Chicago Press, 1995.

[17] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: The Penn treebank," in Computational Linguistics, 19(2), 1993, pp. 313–330.

[18] S. Müller, Head-Driven Phrase Structure Grammar: Eine Einführung. 3rd ed. Tübingen(Germany): Stauffenburg Verlag, 2013.

AUTHOR'S PROFILE

Dr. Peter Wood received his PhD in German Studies from the University of Waterloo, ON Canada in 2010. He teaches Linguistics, Computational Linguistics, and ESL courses at the University of Saskatchewan. Dr. Wood's areas of interest are computational linguistics, corpus linguistics, computer assisted language learning, second language acquisition, and empirical linguistics.