An Analysis and Visualization Tool for Case Study Learning of Linguistic Concepts

Cecilia Ovesdotter Alm, Benjamin S. Meyers, Emily Prud'hommeaux
Rochester Institute of Technology
{coagla,bsm9339,emilypx}@rit.edu

Abstract

We present an educational tool that integrates computational linguistics resources for use in non-technical undergraduate language science courses. By using the tool in conjunction with evidence-driven pedagogical case studies, we strive to provide opportunities for students to gain an understanding of linguistic concepts and analysis through the lens of realistic problems in feasible ways. Case studies tend to be used in legal, business, and health education contexts, but less in the teaching and learning of linguistics. The approach introduced also has potential to encourage students across training backgrounds to continue on to computational language analysis coursework.

1 Introduction

The computational linguistics community makes available software resources for performing structural and meaning-related linguistic analysis on language input. While these tools and models are commonly used in research contexts and have long been used in computational linguistics instruction (Meurers et al., 2002; Baldridge and Erk, 2008), they also have a role to play for enhancing non-computational pedagogy in linguistics.

We present an educational innovation that aims to provide students in undergraduate language science classes with case-based active learning opportunities by enabling them to actively confront linguistic concepts and methods encountered in textbooks and class discussions (or as supplementary materials to stimulate learning) in hands-on tasks that emphasize the applied nature of the study and practice of language science. Computational linguistics software resources tend to be designed for tech-savvy users and expect knowledge acquired in computational, or computational linguistics, curriculum or similar contexts. They often require some understanding of technical details about computer programming and computational linguistics principles and methods. While there are web-based linguistic corpus resources, the interaction potential for users tends to be limited to functionalities such as key-words-in-context searches or look-up of relatively simple structural patterns, as opposed to providing students across majors with the ability to conduct, assess, and critique analyses built on models developed in the computational linguistics community.

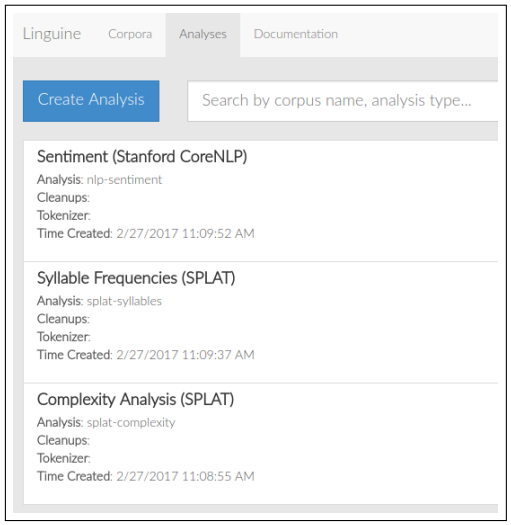

Figure 1: Linguine's tab-based interface showing the Analyses tab with three completed analyses.

Linguine [1] is a web-based tool with a user-friendly interface (Figure 1) tailored to educational use in language science coursework. [2] It draws on natural language processing resources in the open domain in order to provide users with the capabilities to study a range of linguistic structural and semantic patterns in written language input.

[1] Demo: tinyurl.com/ritlinguine [2] github.com/ritlinguine

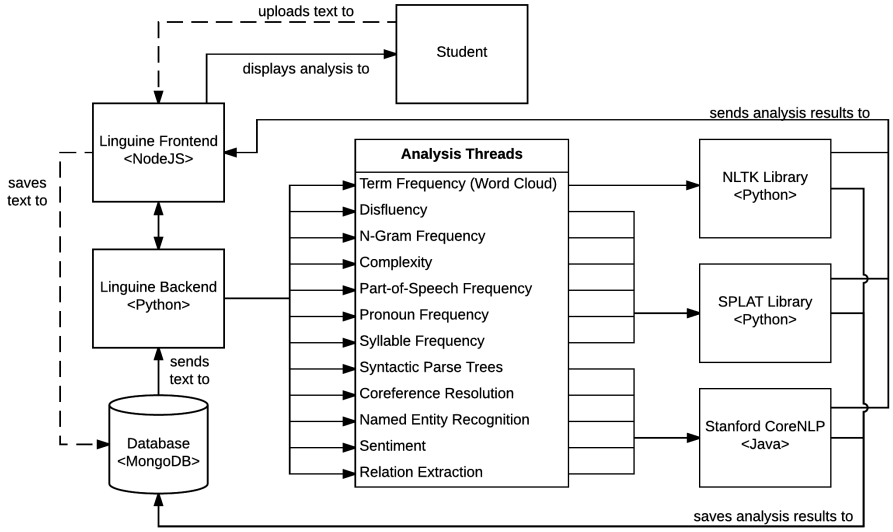

Figure 2: Linguine framework overview. Dotted lines represent text upload. Solid lines represent analysis generation. NodeJS forwards the users's analysis request to Python, which locates the text in the database, performs preprocessing by analysis, and sends the processed text to the appropriate framework for analysis. Generated analyses are stored for later use. NodeJS generates the appropriate visualization.

The motivation for Linguine includes (1) making computationally driven language analysis accessible to non-computing majors in undergraduate linguistics courses; (2) enabling practical, formative opportunities for linguistically informed analysis; and (3) guiding students to perform user-friendly analyses and presenting intuitive visualizations of automatically processed results to make them more readily interpretable. Thus, Linguine contrasts with resources such as the CLARINO Language Analysis Portal (Lapponi et al., 2013) in its pedagogical purpose and its pairing of analyses with visualizations. Linguine also enables a broader set of language analysis functionalities.

This work's main innovation is the applied pairing of this tool with case studies as active learning vehicles in language science. The case studies task students with seeking evidence-based solutions to and recommendations for linguistically grounded real-world problems. They also aim to train students in oral and written communication about their analysis, recommendations, and critical observations. While the case study method is a recognized learning tool in domains such as business and law, it is rather unusual to see it used extensively in the linguistics curriculum. This approach thus differentiates itself from standard ways of teaching linguistics concepts and analysis. Linguine's functionalities and visualizations support and facilitate the situational problem solving and hands-on critical thinking that case studies enable. Adopting the case study method further seeks to nurture student experiences in utilizing linguistics to address and reason over relatable problems in society. Students can also increase their understanding of the limitations and potential utility of language technologies.

In this paper, we describe Linguine and how it contributes to using case study pedagogy in language science. We also report on instructor observations in conjunction with student surveys to provide insights into the utility of the case study model with the Linguine system.

2 Learning Linguistics with Linguine

Linguine is a web application designed for an educational purpose. It provides an easy-to-use interface, allowing interaction with preloaded default or custom-uploaded plain texts for performing language-based analysis. Figure 1 shows the interface for selecting the resulting analysis. For analysis functionalities, Linguine leverages widely available resources for performing natural language processing, including NLTK (Bird et al., 2009), Stanford CoreNLP (Manning et al., 2014), and SPLAT [3], along with the web technologies, NodeJS [4] and d3 [5]. Aspects that set Linguine apart are its focus on enabling class activities and ac-

[3] http://splat-library.org
[4] https://nodejs.org/en/about/
[5] https://d3js.org/

tive learning and its ability to transform machine-processed results into intuitive visualizations. Visualizations depend on the analysis performed and include sentence-by-sentence display of syntactic trees, data summary tables, tool-tips displaying sequential annotations, and colorful markup in text. Users can inspect results in a structured representation within the tool. Results can also be downloaded in JSON format for off-line analysis by users with the necessary background.

Figure 2 shows the data flow between Linguine's components. The architecture is composed of a Python server that interacts with a NodeJS server and a MongoDB [6] database. These components run as system services on a RHEL7 virtual machine. The Python server receives analysis requests from the NodeJS server. It obtains the relevant text from the database and performs preprocessing operations. Preprocessed text is passed to a queue for analysis. Analyses are carried out in parallel using the expected resources. Analysis time depends on the size of the text and the type of analysis. At present, Linguine is an English-focused environment, aimed at language science coursework offered by computational linguists faculty in an English department. However, its framework enables incorporating additional resources for text-based analyses. For example, models trained on other languages, or potentially other forms of unstructured data, could be integrated with adapted visualizations.

The SPLAT library computes statistics on n-grams, part-of-speech tags, syllables, and disfluencies. SPLAT also calculates linguistic complexity measures, including content and idea densities, Flesch readability, Flesch-Kincaid grade level, and type-token ratio. Examples of visualizations produced by Linguine using SPLAT functionality are shown in Figures 3, 4 (left), and 6 (right). Using output from Stanford CoreNLP, Linguine incorporates analysis options requiring sophisticated modeling, including syntactic trees with sentiment labels (Figure 5) and named entity recognition (Figure 6, left). Analyses are saved to the database, allowing users to return to visualizations without having to reprocess their analysis.

The integration of technologies required managing asynchronous communications between Python and Linguine's other subsystems. Data transfer within Linguine is handled by Tornado [7], a Python framework that transfers information as HTTP requests. At present, the tool is envisioned for use by a 25-person class. Detailed analysis of resource utilization for different user group sizes is left for future work.

[6] https://www.mongodb.com/

[7] http://www.tornadoweb.org/en/stable/

Q	This case study activity...
1	was engaging.
2	had clear instructions.
3	was related to the course material.
4	involved a reasonable time commitment.
5	was a valuable learning experience.
6	enhanced my understanding of linguistic concepts.
7	reinforced theoretical concepts from class with an application.
8	let me use linguistic approaches to problem solving.
9	had me engaged in critical thinking.
10	involved a useful reporting experience.
Q	**Using the provided web tools and input...**
11	was straightforward.
12	went hand-in-hand with the case instructions.
13	enhanced my thinking about the case resolution plan.
14	was interesting.
15	was a good learning experience.

Table 1: Satisfaction agreement statements.

3 Case Studies

We have so far developed three case studies for two course contexts: an overview course of language science fundamentals and an English language history course. Combining case studies with the functionality of Linguine provides students with fictional yet realistic scenarios to solve using active inquiry with linguistic data. Each case study comprises a set of design elements, following a template that enables task clarity and effective design and prototyping:

- The case description includes a narrative that sets up a problem and provides background about the data needed to perform language-based analysis in Linguine for gathering evidence. This includes step-by-step analysis instructions with questions to answer, as well as guidelines for presenting work, preparing written reporting, and completing a quiz, in addition to an evaluation rubric.

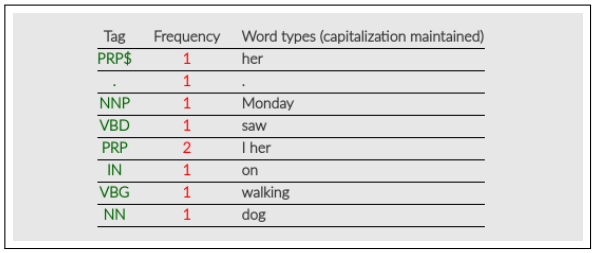

Figure 3: Visualizations: *I saw her walking her dog on Monday.* POS (left); pronoun frequencies (right).

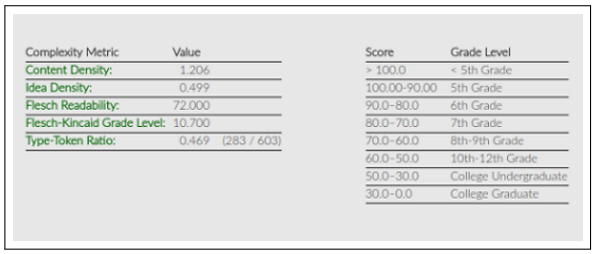

Figure 4: Complexity metrics (left) and term frequencies (right) for a Sir Arthur Conan Doyle excerpt.

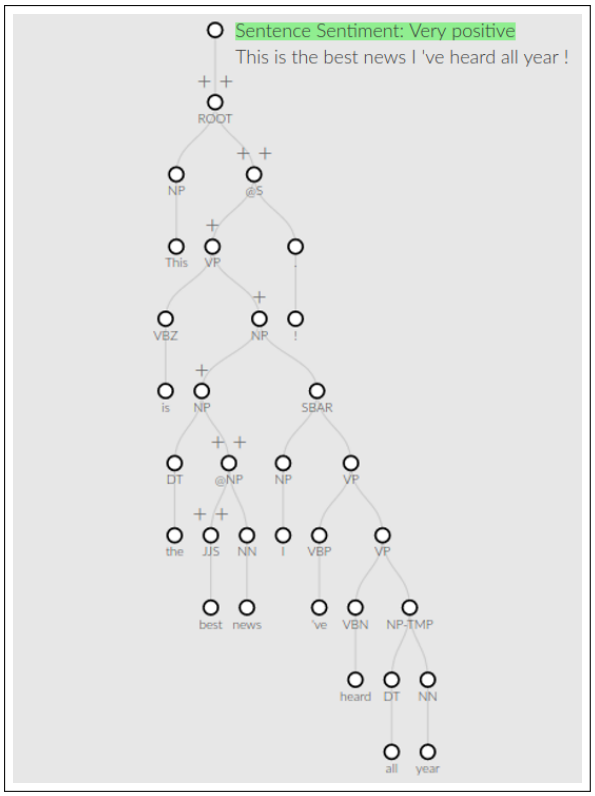

Figure 5: Syntactic tree with sentiment labels for *This is the best news I've heard all year!*

- Data that has been selected and prepared for out-of-class and in-class analysis.
- Two readings that students can consult for reasoning about a case: an applied, broad-audience reading vs. an academic reading.

The case studies are expanded with a development and teaching guide that outlines the motivation for text selection, details preprocessing on the texts, and includes expected answers to their questions. The initially developed case studies are:

- *The Language of Dementia*: Students analyze a selection of picture descriptions from the DementiaBank corpus (Becker et al., 1994) with the goal of assisting a medical researcher in identifying linguistic markers of Alzheimer's disease. Readings include Szatloczki et al. (2015); Goldstein et al. (2010).
- *Historical Varieties of English*: Students examine excerpts of literature across time periods to assist school teachers in choosing grade-appropriate readings for their classes. Readings include, e.g., Perera (1980).
- *Formality in Business Communications*: In roles as analysts for a training agency, students use email data (Klimt and Yang, 2004; Pavlick and Tetreault, 2016, added later) to critically envision guidelines for workplace communications. Readings include Pavlick and Tetreault (2016); Lebowitz (2015).

4 Results of Case Study Exploration

Students in an introductory linguistics course worked with Linguine in assigned teams first on *The Language of Dementia* and a few weeks later on *Formality in Business Communications*. For both cases, students used Linguine and engaged

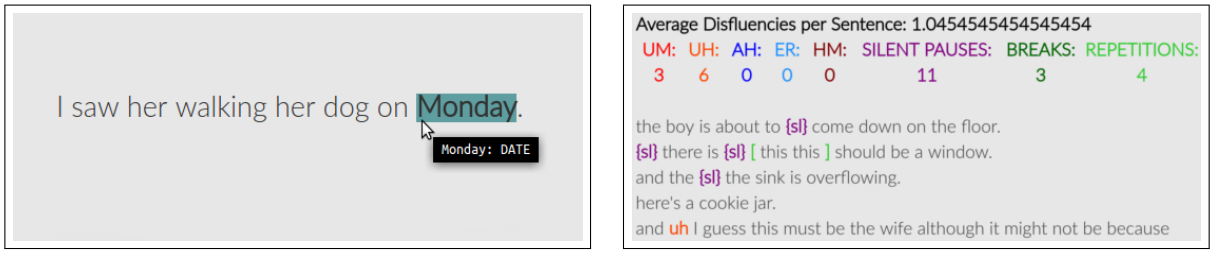

Figure 6: Visualizing named entity recognition (left) and disfluencies in speech transcription (right).

with provided data in and out of the classroom. Student teams reported results orally to classmates in short presentations and in write-ups. The third case study in the English language history class was completed individually by fewer students.[8] Students were instructed to complete the case components and then answer an anonymous case satisfaction survey (Tables 1- 2). This provided a chance to self-report on the learning experience.

Instructor observations of class interactions suggest that several pedagogical benefits emerged. First, students engaged in increased critical thinking about analysis, data, and methods in class. Second, the reporting exercise nurtured co-learning, as students could observe how others approach a problem and choose to visualize, summarize, and present results. Third, the case approach offered a structured framework for teamwork.

Figure 7 (p. 6) shows that a majority of students evaluated the activity positively on all measures, agreeing that the experience was engaging, educational, interesting, and that it stimulated critical thinking and learning (Q1, Q5, Q9, Q14, Q15). Students reported that the activity and tool were clear and straightforward (Q2, Q11, Q12). Crucially, most students felt the experience was relevant to and practiced class material, enhanced their understanding of linguistics, and engaged them in such problem solving (Q3, Q6, Q7, Q8).

Students were also given the opportunity to provide qualitative feedback about their experience; examples are in Table 2 (p. 6). Nearly half the students reported that they particularly enjoyed learning about practical, real-world applications of linguistics. They recognized the links between the concepts seen in class and the case studies. Students found the texts in the second case study (from email data) to be particularly entertaining, and they appreciated the open-ended nature of that case study compared with the first case study.

[8] A co-author served as course instructor.

The negative comments centered on three issues: (1) the modest amount of data provided for analysis; (2) the constraints of Linguine (e.g., allows downloading results as JSON, not csv); and (3) the reporting experience, which a few students found repetitive in the first case study. This feedback has been valuable for continuing to enhance Linguine and case-based instructional materials.

5 Conclusion

We presented the integration of the pedagogical web application Linguine and case studies using this tool in courses. We will continue to explore the system and its educational use and effectiveness in parallel. Developing a systematic process for preparing new case studies is left for future work. Planned expansions of the system and materials include increased focus on processing meaning, and expanding the analysis potential for transcribed speech to enable further pedagogical bridging of analyzing spoken and written language data in language science case studies.

Acknowledgments

The work was partially supported by an RIT PLIG grant. Many thanks to three senior project teams: ProNouns, Rigatoni, and Pastafarians helped develop Linguine's core functionality and interface.

References

Jason Baldridge and Katrin Erk. 2008. Teaching computational linguistics to a large, diverse student body: Courses, tools, and interdepartmental interaction. In *WS on Issues in Teaching CL*. pages 1–9.

James T Becker, François Boiler, Oscar L Lopez, Judith Saxton, and Karen L McGonigle. 1994. The natural history of Alzheimer's disease: Description of study cohort and accuracy of diagnosis. *Archives of Neurology* 51(6):585–594.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: Analyzing text with the natural language toolkit*. O'Reilly.

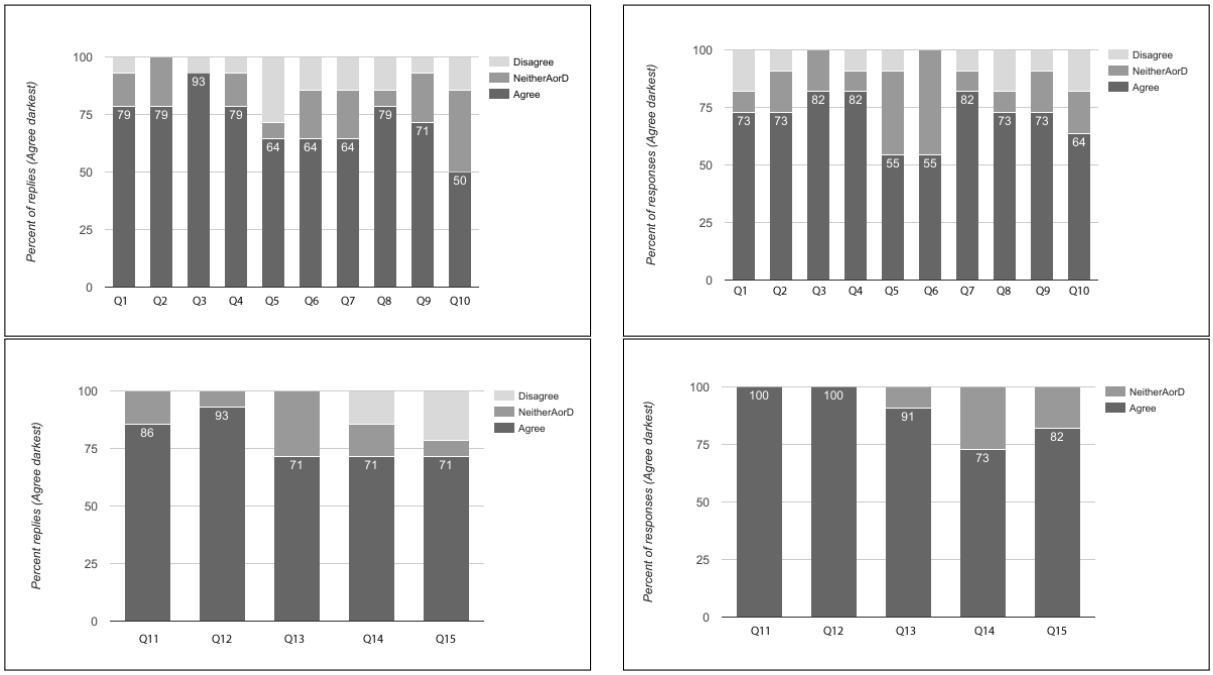

Figure 7: Percent replies for two cases in a class (N = 14 vs. 11 respondents). Top: Feedback on case study activities 1 and 2 shows that users mostly concur on statements in Table 1. Bottom: Feedback on the tool and its input suggest positive user experiences, increasing as tool familiarity grew.

Applications	• I learned that there are real world applications for what we learned in class. • got to work with real data on solving a real problem • I was able to come up with something that could prove to be useful in my life.
Engagement	• Interesting scenario, engaging • I really enjoyed using linguine, it's a great tool. • Keep the online tool, super fun.
Drawbacks	• n-grams could be downloadable as a csv for additional processing. • I wish there was a way to get quantitative data about the parse trees.
Presentations	• awkward having all of the groups present essentially the same information • the reporting part (in class) could use some modifications.

Table 2: Example open-ended feedback from the satisfaction survey.

Jessica Goldstein, Maggie Starbard, and Soren Wheeler (Prod). 2010. Agatha Christie and nuns tell a tale of Alzheimer's. *NPR* [June 1].

Bryan Klimt and Yiming Yang. 2004. The Enron corpus: A new dataset for email classification research. In *Europ Conf on Machine Learn*. pages 217–226.

Emanuele Lapponi, Erik Velldal, Nikolay A. Vazov, and Stephan Oepen. 2013. Towards large-scale language analysis in the cloud. In *WS on Nordic Lang. Research Infra. at NODALIDA*. pages 1–10.

Shana Lebowitz. 2015. Here's exactly how to write an email to your CEO. *Business Insid.* [Nov. 13].

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL Systems Demonstrations*. pages 55–60.

W Detmar Meurers, Gerald Penn, and Frank Richter. 2002. A web-based instructional platform for constraint-based grammar formalisms and parsing. In *WS on Effective Tools and Methodologies for Teaching NLP and CL*. pages 19–26.

Ellie Pavlick and Joel Tetreault. 2016. An empirical analysis of formality in online communication. *TACL* 4:61–74.

Katharine Perera. 1980. The assessment of linguistic difficulty in reading material. *Ed R* 32(2):151–161.

Greta Szatloczki, Ildiko Hoffmann, Veronika Vincze, Janos Kalman, and Magdolna Pakaski. 2015. Speaking in Alzheimer's disease, is that an early sign? Importance of changes in language abilities in Alzheimer's disease. *Front in Aging Neurosci.* 7:195.