Ling185B: Computational Linguistics 2
Spring 2019

Instructor  Tim Hunter  
3103K Campbell Hall  
timhunter@ucla.edu

Prerequisites  Ling185A

Classes  Mon & Wed, 4:00pm – 5:50pm, Bunche A152

Office Hours  Tue, 1:00pm – 2:00pm; or by appointment

Course overview

The aim of this course is to build upon the material from Ling185A in ways that make connections to grammatical systems of the sort that are introduced in theoretical linguistics classes. The first part of the course introduces the necessary computational background, largely revisiting topics from Ling185A in more detail. This serves as preparation for the second part, where we will look in detail at formalized versions of minimalist grammars: grammars that are both in line with core principles of contemporary linguistic theory and precise enough to be manipulated computationally.

Schedule

This schedule, particularly the latter parts, is somewhat tentative and may be adjusted according to the progress of class discussions and student interests.

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Possible readings</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intro and review: FSAs and CFGs</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Chart parsing and grammar intersection</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Probabilistic grammars: parametrizations, mathematical properties</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Information-theoretic complexity metrics (ICMs)</td>
<td>Hale (2016)</td>
</tr>
<tr>
<td>6</td>
<td>Multi-component expressions; MCFGs; MGs with string-tuples</td>
<td>Clark (2014), Stabler (2004)</td>
</tr>
<tr>
<td>7</td>
<td>MGs as finite-state tree grammars</td>
<td>Kobele et al. (2007)</td>
</tr>
<tr>
<td>8</td>
<td>Probabilistic MGs and ICMs</td>
<td>Hunter and Dyer (2013)</td>
</tr>
<tr>
<td>9</td>
<td>Extra topics 1 (see below)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Extra topics 2 (see below)</td>
<td></td>
</tr>
</tbody>
</table>

Possible extra topics for the last two weeks of the quarter include:

- Subregular phonology (Heinz, 2018)
- Semantics via continuation-passing (Barker, 2002)
• Relating syntactic generalizations about long-distance dependencies (e.g. movement) to the finite-state property of MG derivation trees (Graf, 2018a,b)
• Transition-based parsing for MGs: top-down (Stabler, 2013) and left-corner (Stanojevic and Stabler, 2018)
• Empirical case studies in human sentence-processing, compared with ICMs (Hale, 2006; Yun et al., 2015)

Requirements and Grading

• Four homework assignments (60%). These will involve a combination of pencil-and-paper mathematical exercises and programming tasks. Tentative due dates in Weeks 3, 5, 7 and 9.
• Participation in discussion (10%). This means coming to class having thought about any relevant readings and homework assignments, asking questions when something is unclear, and engaging with other students’ questions and comments. This may also include participation in arranged online discussions in between class meetings.
• Final project (30%). A small project on a topic of your choice related to the course material. This may be completed either individually or in pairs. It may include no programming at all or be largely a programming exercise (or anywhere in between), but if it has a significant programming component it must be accompanied by a short written report explaining the background and motivation and describing your implementation. Due at the end of exam week (6/14/2019).

<table>
<thead>
<tr>
<th>Letter grades</th>
</tr>
</thead>
</table>

Learning outcomes

• Knowledge outcomes:
  – Students should know what finite-state grammars, context-free grammars and multiple context-free grammars are, and know the limitations of each.
  – Students should understand the relationship between grammatical structure, recursion and dynamic programming.
  – Students should understand the relationship between probabilistic and non-probabilistic grammars.
• Skills outcomes:
  – Students should be able to read descriptions of grammatical systems in traditional mathematical notation and write corresponding programs using recursion and/or dynamic programming.
  – Students should be able to relate analyses expressed in the standard terminology and notation of contemporary linguistic theory to appropriate formalized implementations.
• Attitudes and values outcomes:
  – Students should come to appreciate the kind of understanding of the human mind that can come from trying to express its workings in a completely formalized system.
Behavioral outcomes:

- By combining the skills outcomes listed above, students should be able to construct programmed implementations of linguistic analyses and use these to facilitate the empirical testing of these analyses.

References


