Writing Tips

What belongs in each section of a technical report for a project like this? Here are some tips to help you structure your project and your report.

Introduction

Because this project was assigned to you, you may find it difficult to write about broader context or even the research question. The solution is to choose your corpus carefully. Although you will spend the most research time developing the CFG, the most important result is what kind of sentences you can parse or generate automatically with your CFG. Choose a corpus where it would be interesting or useful to generate new sentences or to learn more about the style. If I chose *The Cat in the Hat* by Doctor Seuss, for example, I might describe my broader context and research question like this: 'Doctor Seuss is a well loved children's author who is famous for his distinctive writing style. How simple is the grammar that he uses for children, and is it so simple that we could write stories in the Seuss style semi-automatically?' If I chose a selection of abstracts from the research papers of a particular author, my broader context and research question might look like this: 'Dr Yksomch has written many papers on the Bantu language family. Her most important findings are in the abstracts we selected. If we wanted to build a question-and-answer system for facts about Bantu languages, how easy would it be to auto-generate responses based on her abstracts?'

There will not be time in this project to investigate earlier findings thoroughly: the textbook already has a summary of the most important findings. You may, however, find useful tips for building your CFG by looking at literature on CFGs or treebanks for the language of your corpus and using those CFGs or treebanks as a starting point. If you are working in English, for example, you may wish to read about the tags in the Penn Treebank style (e.g., http://www.clips.uantwerpen.be/pages/mbsp-tags, or §5.2 in the textbook); in Dutch, you could use the Spoken Dutch Corpus (http://lands.let.ru.nl/cgn/doc_Dutch/topics/version_1.0/annot/syntax/syn_prot.pdf). Wikipedia has a list of popular treebanks in other languages (http://en.wikipedia.org/wiki/Treebank).

You should not pay for any of these treebanks! Many of them have free documentation available, and you can also search for a research papers that have used them in order to get a sense of their style.

Method

The most important part of your Method section will be a description of your corpus. Where does it come from? How long is it in words and in sentences? What is the average word length? What is the size of the vocabulary (number of unique words)? What pre-processing did you use before building your grammar (e.g., converting everything to lowercase or removing punctuation marks)?

After settling on a corpus, most students will follow a similar procedure. You will want to make a manual annotation (syntax tree) for each sentence in your corpus. (We will learn how to do this in the coming weeks.) As you make your annotations, you may come across sentences that are too difficult to annotate. You will rewrite them in as simpler form so that you can make a syntax tree-annotation. Once you have annotations for every (possibly simplified) sentence, you will use all of the patterns in your annotations to construct a CFG.

Finally, in order to evaluate your grammar, you will (1) confirm that it can parse every sentence in your (simplified) corpus correctly and (2) use the grammar to generate 50 random sentences. You will check these sentences to see if their syntax is correct and if they make semantic sense.

After the first exam (so not for your first draft), you will learn about possible techniques to improve the performance of your grammar.

Results

Your results will include your grammar itself, any simplifications you made, and your analysis of the syntax and semantics of your random sentences. After you try to improve your grammar, your results will also include your new grammar and your analysis of the syntax and semantics of your new random sentences. You may wish to use a well-designed table to present these findings.

Discussion

The goal of this assignment is **not** to achieve a perfect grammar – that would an enormous amount of time and not be interesting to read. The goal is to use your experiment to explain (1) what is simple and what is complex about the grammar of your corpus and (2) how well that grammar can generalise to generate new sentences automatically. Is there anything surprising about the results that teaches us something about the corpus, or are they exactly what one should have expected?

The discussion should also include your explanation of where and *why* CFGs work well for the corpus – and where they do not. CFGs are a popular but imperfect technique, and it is often easier to describe and understand their advantages and limitations using a specific experiment like this project.

You may also wish to discuss other improvements that could help your system perform better in the future. We will learn about many of these in the second half of the course, e.g., semantic attachments. What do you think would be *most* helpful for your system? Why?

References

As the PAV website recommends, the ACM style is fine. You will need to cite the source of your corpus and any references you used to help you make your syntax-tree annotations.

Appendices

Depending on how complex your grammar becomes, you may choose to include the full grammar only in an appendix and write a shorter summary or excerpt of it in the results. Discuss with your PAV leader what the best choice for your project is. The goal is always to make it as easy as possible for a reader to understand what you did and possibly build on it in his or her own research.