Final Project Discussion

Adam Meyers
New York University
Summary

- Project Timeline
- Project Format
- Details/Examples for Different Project Types
  - Linguistic Resource Projects: Annotation, Lexicons, ...
  - Programming Projects
  - Research Paper
  - Other
- Note: I will give mostly English examples, but you can use any language for your data: systems that process other languages, resources for other languages, etc. are all appropriate.
Project Time Line

• Project Proposal
  – Due April 11

• Project Talk:
  – 3 minutes + 1 minute per questions
    • 3 slides summarizing recommended
  – Presented on Apr 27, May 2 or 4
    • Only 2 days if there are a lot of multi-student projects
    • Currently 54 students * 4 minutes = 216 minutes
    • 3 * 75 minute class = 225 minutes

• Final Project Report Due
  – May 9
Format

• Use the ACL style information either for Word or latex for ACL 2015
  – Go to the call for papers:
    • http://acl2015.org/call_for_papers.html
    • Scroll down to “Format” and use the appropriate style files for latex or MS Word
  – Please include your name on the paper and do not try to make it anonymous as the ACL instructions indicate (i.e., you are not submitting the paper for blind review)
• The final papers should be approximately 6-10 pages, not including references.
  – The page length is less important than the content
  – You may hand in other materials in addition to the paper
Proposal (due Apr 11)

- A couple pages showing that your paper is plausible
- Problem Statement or Introduction
  - Indication about what motivates the work you are describing: the research question you are investigating, the purpose of the algorithm you are implementing, etc.
- Strategy for Solving the Problem
  - **Annotation Project:** initial specifications, a small amount of annotation (e.g., 1 page), and a plan on how to achieve a modest amount of high-quality annotation.
  - **Lexicon Project:** initial specifications, a small number of lexical entries, a word list or method for deriving word list, method for deriving dictionary (manual, automatic, semi-automatic)
  - **Programming Project:** Description of an existing algorithm and how you plan to implement it and test it.
  - **Research paper:** what you plan to read and how you expect it will address the problem statement
Baseline and Full Systems

• When describing the results it is often useful to compare these results to some baseline, a system used for purposes of comparison.

• The baseline system can represent an obvious method that any system should beat
  – For example, for POS taggers, choose most frequent tag for each word based on training corpus, and choose NN for all unknown words.
  • It is possible that such a system could get 75% accuracy (suppose 50% of tokens have only 1 possible tag and another 25% have one tag that makes up at least 50% of the instances).

• Another (higher) baseline system could be also a proof-of-concept system, the very basic implementation of what you are doing before adding all the bells and whistles.

• Similar to a program, annotation guidelines may include distinctions that are more or less difficult tags to annotate consistently. A baseline system may collapse difficult distinctions. For example, suppose you are tagging adverbs and your system distinguishes the 2 instances of the adverb logically listed below. It may make sense to evaluate annotator agreement with and without this distinction.
  
  – Logically, they should be able to do it. [VIEWPOINT]
  – They should be able to solve the problem logically [MANNER]

• It is a good idea to mention one or more baselines in your proposal. It is a good idea to guarantee that you can produce something simple, even if your more advanced ideas interest you more.
Linguistic Annotation Projects

• Write Specifications & Annotate Documents
  – Find related work and distinguish your approach
  – Find one annotator in addition to yourself, so that it is possible to measure inter-annotator agreement
    • Multiply annotate and adjudicate sample for evaluation purposes
    • Or multiply annotate and adjudicate for all annotation
  – Possible deals:
    • annotate for each other
    • Programmer uses annotation for program project

• Design and Implement Annotation Project using Amazon Turk
  – Figure out a way to use very simple annotation
  – Design a task for Amazon Turk and Run it
  – May cost you a little money ($50 buys a lot of annotation)
  – If anyone is interested in this route, I can provide more info
Sample Annotation Projects

• Apply one a known type of Annotation (NE, POS, Chunking, semantic role labeling) to a new domain of text: web data, technical data, a new language, etc.

• Develop specifications and annotate new classes of NEs, Relations, or Events

• Develop specifications and annotate an interesting phenomenon:
  – quantifier scope
  – sentiment (your version)
  – Idiomatic expressions
Lexicon Projects

• Motivation
  – What would your lexicon help achieve?
  – How would you test this?

• Strategy for Constructing Lexicon
  – Word List:
    • I can provide large lists of English lemmas (and morphological variants)
    • Lists can be derived from corpora, sorted by frequency
  – Automatic Methods:
    • Even heuristics that are 50% accurate can save time, e.g., for finding a list of place names, this pattern could help: `grep -E 'going to [A-Z]'`
  – Lexicographer Interfaces (may be programs online)
    • Bad idea to have lexicographers type in features
  – Write up and Test Specifications: similar to annotation
  – Evaluation
    • Consistency (like annotation)
    • Demonstrate that information in the lexicon can be used for some application
      – Demonstrate that it would case the correct prediction most of the time.
Existing & Future Lexicons

• Existing Lexicons and Databases
  – Comlex Syntax: Syntactic & Semantic
  – Nomlex-Plus, ADJADV: Paraphrase
  – WordNet: Word Sense
  – CIA FactBook: Gazetteer

• New Genres Where Lexicons Could Help: Twitter, etc.
Sample Programming Projects

• Sequence Labeling System
  – Strategies: Rule based, HMM or Other
  – Types: POS tagger, Chunker, NE tagger, Time Expression, ...

• Phrase Structure Generation Program
  – Change the system to make more plausible sentences
  – Generate Poetry

• Implementing Existing NLP Algorithm, even if not covered in class
Survey Papers

• If you select an area, I can recommend some articles/books to get you started
• Research Areas: Machine Translation, Summarization, Question Answering, Sentiment Analysis, Information Extraction, Reference Resolution, Predicate Argument Structure, …
• Your proposal should include an abstract, a description of some preliminary work that you have read, a list of articles that you intend to read and an enumeration of your research goals.
Resources Available

• In addition to publications, there are other resources available to help with your research.
  – Corpora
  – Lexicons
  – Programs

• If you are creating a resource or program, you could possibly use these resources as pre-processors.
  – If you are classifying adjectives, a POS tagger may help you find adjectives
  – A parser would be a good pre-processor to a system doing pronoun coreference
The Wall Street Journal Penn Treebank

• I could make this available in several forms
• Upenn: Trees, POS tags, Noun Groups
• BBN:
  – NE – Inline annotation – you can convert it to one token per line or use it as is
  – Coreference – marks pronoun coreference – there is probably a bit of corpus preparation to make this work
• There are some licensing issues, so I have to distribute these in a careful way.
Downloadable Tasks with Annotated Corpora (for Testing and/or Training)

• Corpora for Drug-Drug Interaction
  – http://labda.inf.uc3m.es/DrugDDI/DrugDDI.html

• WePS – searching for entities on the Web

• Spanish Corpus with POS tags
  – http://www.comp.lancs.ac.uk/linguistics/crater/spanishfiles.html
Some Downloadable Corpora

• The Open American National Corpus
  – A variety of different types of data
  – A limited amount of manually annotated data
  – Automatically annotated data from various programs
  – Most annotation is offset annotation

• The Singapore SMS corpus

• Wikipedia XML

• Tweet Corpus (for sentiment)
Some Lexicons

• COMLEX – licensing issues, but I could make it available
• NOMLEX and related dictionaries
    • “Those Other Dictionaries” describe the dictionaries
      – “directory linked here” brings you to directory of dictionaries and other resources
    • Everything as one archive file (Nombank 1.0 Release)
• WordNet: http://wordnet.princeton.edu/
• FrameNet: https://framenet.icsi.berkeley.edu/fndrupal/
• CMU Pronunciation Dictionary: http://www.speech.cs.cmu.edu/cgi-bin/cmudict
• Subjectivity Lexicon (and sentiment corpus)
  – http://mpqa.cs.pitt.edu/
• CIA factbook:
Tools and Packages

• You can use output of NLP tools, as input to your system, provided you make your contribution clear.

• Examples:
  – NLTK
  – Machine Learning Packages (as per talks)
  – Other tools you download
Homeworks/Lectures as Sources for Final Projects

• It is possible to extend a homework assignment into a final project
• Generalize, Add techniques, etc.
• I will start including ideas of how to extend particular HW assignments
• Even if there is no assignment, I will begin providing data or information about how to do an assignment based on a particular lecture.
Consider Turning Your Paper into a Conference Paper

• It is Possible that a Really Good Paper could form the basis of a Conference Paper
• If applicable, I will help you find a good venue: conference or workshop
• A published paper can be a major factor in getting into a good graduate program, particularly a PhD program
Group Project: Coordinating Project with other Students

• Observation: Most published research papers have multiple authors
• It is possible to do a more ambitious project if you team up
• For grading purposes, I ask that you provide a clear statement as part of the write-up about who did what
Considerations for Group Projects

• Divide labor based on individual strengths

• Code writing
  – Version control
  – Write separate components, but agree on input and output specifications
  – If different programming languages, consider using file I/O

• Possibly choose different pieces of a pipeline architecture
  – \( \text{input} \rightarrow \text{system}_1 \mid \text{system}_2 \mid \ldots \mid \text{system}_{N-1} \rightarrow \text{output} \)
Example of How to Work Together

- Student\textsubscript{1} and Student\textsubscript{2} invent a new task: automatic classification of recipes
  - Recipe = appetizer, dessert; main course; lunch; dinner; breakfast;
  - Ingredients = main ingredient; minor ingredient; garnish; other
- Student\textsubscript{1}:
  - Writes annotation guidelines
  - Writes up an entry program
  - Annotates data 1 time
  - Adjudicates multiply annotated data
  - Choosing some features for final ML system
  - Designs and runs evaluation on system
- Student\textsubscript{2}:
  - Sets up corpus: acquires corpus, formats corpus, divides into train, dev and test
  - Writes baseline system for task
  - Annotates data 1 time
  - Coordinates tests of algorithms in Machine Learning Package
  - Chooses some features for final system
Please Ask

• If you need help finding resources or citations relevant to your project, please send me email.
Final Project Proposal

- Due April 11