Final Project Discussion

Adam Meyers
New York University
Outline

• Goals
• Requirements
• Types of Projects
• Examples
Goals

• Some Version that works, even if more basic than intended
• Clear Description
  – Including what worked and what didn't work
• Impact Beyond this Class
  – Step towards Conference submission
  – Distribute-able Software or Other Resource
    • Open source or Commercial
    • Need not be final version
  – Step towards further research
    • Specifications, pilot studies, etc.
    • Preliminary research for later project
Project Time Line

• Project Proposal
  – Due November 14, 2017

• Project Talk:
  – Single student projects
    • 3 minutes + 1 minute per questions
    • 3 slides summarizing recommended
  – Multi-student projects
    • 4 minutes + 1 minutes per question
    • 4 slides recommended
  – Presented on December 5, 2017 and December 7, 2017
    • Maximum Time = 38 1-student talks * 4 min = 144 min
    • Estimate = (18 1-student talks * 4 min) + (10 multi-student talks * 5 min) = 122 minutes
    • 2 * 75 minute class = 150 minutes
    • At least 6 minutes, but probably 28 minutes maximum leeway

• Final Project Report Due
  – December 14, 2017
Types of Final Projects

- **Programming Project** – a working program, evaluation and a write-up
- **Resource** (lexicon, annotated corpus, etc.) – specifications, evaluation and write-up
- **Survey Paper** – Try to describe the state of the art for attempted solutions to a particular NLP problem or subfield.
Format

• All projects should include a written paper of approximately 5-10 pages, not including references.
  – The page length is less important than the content
  – You may hand in other materials in addition to the paper
• Use the ACL 2016 style guidelines (suggested, not required)
  – Call for papers: http://acl2016.org/index.php?article_id=9
  – Formatting Instructions:
    • http://acl2016.org/files/acl2016.zip
  – Use instructions for final copy, not submission (blind review)
  – Latex style files for the text and the bibliography are provided
    • This makes formatting easy if you use latex
    • Typical for many areas of CS, including NLP
    • https://www.overleaf.com/—Web-based interface that compiles latex in real-time
  – No MS Word template, but .pdf file explains guidelines
Collaboration

• Types of Collaboration
  – A team project
  – Set of projects that are components of a larger system
    • e.g., pipeline architecture
  – A shared task with different systems
    • 1 group could create the task via annotation and specs
      – That group could also conduct the evaluation
      – They could create your shared task and advertise on the forum
    • Several groups could make systems that run on this data
  – Ensemble system combining different approaches
  – Trading tasks, e.g., annotating for each other's tasks
  – Tasks that are part of a pipeline, e.g.,
    • Student/Group 1 does NE extraction
    • Student/Group 2 does Relation extraction using those Nes
    • Student/Group 3 does coreference using NEs
      – relations extracted by Group 2 important feature for coreference system
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
  – input $\rightarrow$ system$_1$ $|$ system$_2$ $|$ $\ldots$ $|$ system$_{N-1}$ $\rightarrow$ output
Example of How to Work Together

- Student$_1$ and Student$_2$ invent a new task: automatic classification of recipes
  - Recipe = appetizer, dessert; main course; lunch; dinner; breakfast;
  - Ingredients = main ingredient; minor ingredient; garnish; other
- Student$_1$:
  - 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$_2$:
  - 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
Proposal (due 11/14/2017)

• A couple pages showing that:
  – You have begun working on your project
  – It 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
  – Programming Project: Your algorithm and how you plan to implement and test it. Include both a simple version that you are confident you can complete before the deadline and a more elaborate version that you want to implement if you have time.

• Resources
  – 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)

  – Survey paper: what you plan to read and how you expect it will address the problem statement

• If this a multi-person project, please indicate the roles each member will play
  – Makes grading easier

• If this is a single-person project, please indicate if you would consider teaming up with somebody who is working in a similar area

• Indicate any other forms of collaboration you are considering
How to Get Started

• Find an area that interests you and read 1 or 2 academic papers in that area

• A possible goal is to produce a paper similar to one of the ones that you read
  – I don't expect a conference-grade paper in 1 month of work – the beginnings of 1 would be nice
  – However, this can still give you an idea of the flavor of paper that I am looking for
    • Type of evaluation
    • Sort of references
    • Etc.
Final Submission: Programming Project

• A working program
  – If your program works, but you cannot share it that is OK, but please indicate the issue (e.g., it is owned by your employer).
  – **Implement at least one component yourself**, though including other systems & libraries is expected

• Sample input and Output
  – training/development/test data and output

• A written description:
  – Some NLP references
  – Purpose of program
  – If you incorporate libraries/other programs, be clear about the part of the project that you implemented yourself
  – Description of the algorithms used
  – Description of baseline(s)
  – Evaluation
Baseline and Full Systems for Programming Projects

• It is useful to compare your results to baselines, systems used for purposes of comparison.

• A (low) 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).

• Higher baseline systems may be proof-of-concept systems, very basic implementations of what you are doing before adding bells & whistles.

• 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 you have more exciting things you would like to try.
Sample Programming Projects

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

• Information Extraction
  – Named Entities, Time Expressions, Relations, Events

• Implement Existing NLP Algorithm

• Document Classification
  – Gather features/statistics of documents segmented into classes
    • Genre, Style, Topic, Source, Viewpoint, …
  – Use to Automatically sort an unsegmented set of documents

• Automatic Summarization

• Extend Homework Assignment to Cover New Ground
Final Submission for Annotation Project

• Annotation Specifications
• Annotation Output
• Write-up including:
  – Comparison with previous work
  – Summary of specifications
  – Annotation and evaluation procedures
  – Evaluation Results
Annotation Projects

• Write Specifications & Annotate Documents
  – Find related work and distinguish your approach
  – Find one annotator in addition to yourself, so it is possible to evaluate
    • InterAnnotator Agreement
      – Kappa = \( \frac{\text{Percent (Actual Agreement)} - \text{Prob (Chance Agreement)}}{1 - \text{Prob (Chance Agreement)}} \)
    • Score (precision, recall, f-measure) against a gold standard
      – Multiply annotate and adjudicate to create sample
  – 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)
Sample Annotation Projects

• Apply 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
Baseline for Annotation Projects

• As an initial experiment, annotate a simpler version of your task to see if you can get reasonable agreement.

• Try Collapsing distinctions in your annotation for purposes of evaluation
  – Example: 2 sense of the adverb *logically*
    • *Logically, they should be able to do it.* [VIEWPOINT]
    • *They should be able to solve the problem logically* [MANNER]
  – If collapsing these sense for evaluation purposes, gives you a better agreement score, you might report both results.

• Try Only annotating a subclass of your data or part of your specs
  – Example1: You are annotating instances of attacks reported in blogs, along with arguments of the attacks (attacker and victim)
    • Initially only annotate sentences containing a limited number of attack words (e.g., *attack, harm, kill, ...*)
  – Example2: You are annotating attack and movement events in news
    • Initially, only annotate *attack* events
Crowd Sourced Annotation: Getting Good Results

- Detailed Specifications (true for any annotation)
- Simplify Task (not expert annotators)
- Use a qualification test (keep out trolls)
- Do short tests of your task
  - Fix task until you are satisfied with results
  - Then run on larger amount of data
- Multiply annotate data and merge, e.g.,
  - 30 groups of 25 examples = 750
  - Each group of 25 is annotated 3 times = 2250
  - 10 annotators each annotate 225 examples (+ 25 qualifying examples)
  - Sample merging strategy: voting
    - If binary classification, always can pick one answer
Final Submission for a Lexicon Project

• The Lexicon Created or Acquired
• Code for any programming components
• If manually created, similar components as with annotation project
• If automatically acquired, similar components as programming project
• Write-up including comparison to previous work and evaluation.
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][A-Za-z]*'`, especially when applied to a very large corpus (e.g., via a webcrawler)
  – 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
    • Perhaps with a very primitive program (think proof of concept)
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, and other social media
  – Technical domains
    • Science, Technology, Business, Law, Trades
    • Games, Hobbies, Collectables, ...
Survey Papers

• **Goal**: characterize the benefits/drawbacks of different techniques and to say something about the state of the art or use research to propose and motivate ambitious project.

• **Research Areas**: Machine Translation, Summarization, Question Answering, Sentiment Analysis, Information Extraction, Reference Resolution, Predicate Argument Structure, …

• **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.

• I can recommend some articles/books to get you started
Other Project Ideas

• Example: An Evaluation Paper
  – Propose a system for evaluation and implement it.
  – Evaluate the output of several open source systems using your measure.
  – Compare your measure to previous work

• Any idea that you come up for a project should have most of the following components:
  – A way of succeeding if your original idea turns out to be overly ambitious
  – A method for evaluating your results
  – Citing previous work
  – Major components that are part of Natural Language Processing
Use and Cite Previous Work

• Read and cite previous papers
• Run existing NLP programs to create input to your system
  – You could contribute one piece to a larger pipeline
• Use corpora, annotation and lexicons created by others as input, training/test/dev data, etc.
• Examples:
  – Parsers or treebanks for annotating pronoun coreference
  – POS tagger, pre-tagged data, existing lexicons
    • Annotation or lexicon of finer grained classes
    • Example: Subclasses of adjectives or adverbs.
The Wall Street Journal Penn Treebank

• Upenn: Trees, POS tags, Noun Groups
  – Some available in HW, more possible

• BBN:
  – NE – Inline annotation – you can convert it to one token per line or use it as is
    • NYUClasses Resources: BBN-NE.tgz
  – Coreference – marks pronoun coreference – there is probably a bit of corpus preparation to make this work
    • Can be made available
Downloadable Tasks with Annotated Corpora (for Testing and/or Training)

- Corpora for Drug-Drug Interaction
  - [http://labda.inf.uc3m.es/DrugDDI/DrugDDI.html](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](http://www.comp.lancs.ac.uk/linguistics/crater/spanishfiles.html)
Some Downloadable Corpora

• The Open American National Corpus
  – http://www.anc.org/OANC/
  – 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)
  – http://help.sentiment140.com/for-students/

• A list of influential corpora in NLP:
  – http://www.lancaster.ac.uk/staff/xiaoz/papers/corpus%20survey.htm
Some Lexicons

• COMLEX – available through NYUClasses Resources
  – COMNOM.tgz (most uptodate version)
• 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

- Machine Learning Packages, e.g.,
  - OpenNLP (https://opennlp.apache.org/)
    - used for HW6
  - http://textminingonline.com/tag/maxent-classifier

- Some advanced software may be challenging to install/use
  - Moses (open source, state-of-the-art MT system)
    - http://www.statmt.org/moses/
  - Parsers
    - http://www.isi.edu/publications/licensed-sw/fanseparser/

- I may be able to help you find a particular type of software
Other Sources for Final Projects

• It is possible to extend a homework assignment into a final project
  – Generalize, Add techniques, etc.
  – Example: POS tagging for Chinese (LDC2007T36.tgz on NYUClasses)

• Website with ML tasks: www.kaggle.com
  – Example: Predict Rotten Tomatoes scores: negative, somewhat negative, neutral, somewhat positive, positive
    https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews

• Shared Tasks – CONLL, MUC, ACE, …
  – Read about the task you are interested first
  – Download if it is open source
  – I can get you access to some (but not all) of the licensed data
Other LDC Corpora

• [https://catalog.ldc.upenn.edu/](https://catalog.ldc.upenn.edu/)
• Go to LDC catalog
  – For the first time, you can obtain anything in the LDC catalog through the NYU Library
  – Before the Library took over, I could provide access to only a subset of the material
  – Since this is a new process, I suggest investigating it earlier, rather than later, so I can help it go smoothly
  – Contact at the library:
    • Scott A Collard
    • Email: scott.collard at nyu.edu
NYUClasses Resources: Available for Download

• ACE Corpus
  – ACE-events.tgz and plain-ACE-txt.tgz

• Named Entity tags on Penn Treebank
  – BBN-NE.tgz

• Chinese Penn Treebank
  – LDC-2007T36.tgz

• COMLEX Syntax
  – COMNOM.tgz

• More info if you download and need clarification
Web of Law Corpus

- Text versions of 64K supreme court decisions, along with processed versions of the text (sentence split, POS tagged, and several other types of analysis)
- Part of my current research
- There are students currently working on this. Preliminary Information is provided at these links:
  - Files: https://nlp.cs.nyu.edu/meyers/web_of_law.html
  - Programs: https://github.com/AdamMeyers/Web-of-Law
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
• Note that most publications have multiple authors, i.e., collaboration may help
Please Ask

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

• Counts as a Homework Assignment
• http://cs.nyu.edu/courses/fall17/CSCI-UA.0480-006/homework7.html