

Workshop 1: Project planning

Christopher Potts

CS 244U: Natural language understanding
Feb 5



Three workshops

- Today: Workshop 1: Project planning
Feb 14: Lit review due [[link](#)]
- Feb 21: Workshop 2: Evaluating your model
Feb 28: Project milestone due [[link](#)]
- Mar 7: Workshop 3: Writing up and presenting your work
Mar 12, 14: Four-minute in-class presentations [[link](#)]
Mar 20, 3:15 pm: Final project due [[link](#)]

(Policy on submitting related final projects to multiple classes [[link](#)])

Final project types and expectations

Research papers

These are papers where you attempted some new research idea. This doesn't have to be publishable research; it's totally great to do a replication of a result you read about.

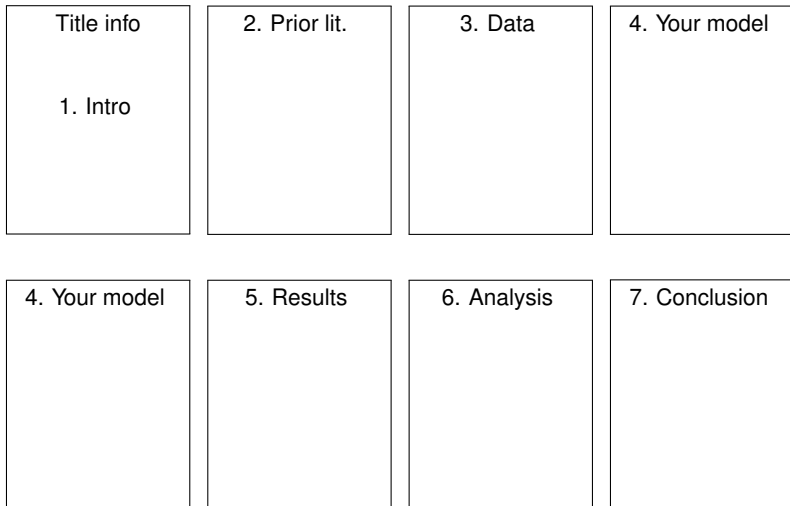
Implementation papers

These are papers where you code up a version of someone else's algorithm just to learn the details of the algorithm, or do a big semantic data labeling project.

For more on expected components and expected results:
<http://www.stanford.edu/class/cs224u/index.html#projects>

The outline of a typical NLP paper

Eight two-column pages plus 1-2 pages for references. Here are the typical components (section lengths will vary):



Goals for today

- Get you thinking concretely about what you want to accomplish.
- Identify productive steps you can take even if you're still deciding on a topic or approach
- Try to help you avoid common pitfalls for projects.

Inspiration

It's nice if you do a great job and earn an A on your final project, but let's think bigger:

- Many important and influential ideas, insights, and algorithms began as class projects.
- Getting the best research-oriented jobs will likely involve giving a job talk. Your project can be the basis for one.
- You can help out the scientific community by supplying data, code, and results (including things that didn't work!).

Inspiring past projects

<https://www.stanford.edu/class/cs224u/restricted/past-final-projects/>

- Semantic role labeling
- Unsupervised relation extraction
- Solving standardized test problems
- Humor detection
- Biomedical NER
- Sentiment analysis in political contexts
- Learning narrative schemas
- Supervised and unsupervised compositional semantics
- ...

Plan for today

Overview

Lit review

Getting data

Annotating data

Crowdsourcing

Project set-up

Project development cycle

Conclusion

Lit review

General requirements

- A short 6-page single-spaced paper summarizing and synthesizing several papers on the area of your final project.
- Groups of one should review 5 papers, groups of two should review 7 papers, and groups of three should review 9.
- Preferably fuel for the final project, but graded on its own terms.

Tips on major things to include

- General problem/task definition
- Concise summaries of the articles
- Compare and contrast (most important)
- Future work

More details at the homepage [[direct link](#)]

Our hopes

- The lit review research suggests baselines and approaches.
- The lit review helps us understand your project goals.
- We'll be able to suggest additional things to read.
- The prose itself can be modified for inclusion in your paper.

Resources

The relevant fields are extremely well-organized when it comes to collecting their papers and making them accessible:

- ACL Anthology: <http://www.aclweb.org/anthology-new/>
- ACL Anthology Searchbench: <http://aclasb.dfki.de/>
- ACM Digital Library: <http://dl.acm.org/>
- arXiv: <http://arxiv.org/>
- Google Scholar: <http://scholar.google.com/>

Strategies

- The course homepage is a starting place!
- Trust the community (to an extent): frequently cited papers are likely to be worth knowing about.
- Consult textbooks for tips on how ideas relate to each other.
- Until you get a core set of lit review papers:
 - 1 Do a keyword search at ACL Anthology.
 - 2 Download the top papers that seem relevant.
 - 3 Skim the introductions and prior lit. sections, looking for papers that appear often.
 - 4 Download those papers.
 - 5 Return to step 3.

Start your lit review now!

In just five (5!) minutes, see how many related papers you can download:

- 1 Do a keyword search at ACL Anthology.
- 2 Download the top papers that seem relevant.
- 3 Skim the introductions and prior lit. sections, looking for papers that appear often.
- 4 Download those papers.
- 5 Return to step 3.

Bonus points for downloading the most papers worth looking at!!!

Getting data

If you're lucky, there is already a corpus out there that is ideal for your project.

Large repositories

Linguistic Data Consortium: <http://www.ldc.upenn.edu/>

- Very large and diverse archive.
- Especially rich in annotated data.
- Corpora are typically very expensive (but see the next slide).

InfoChimps: <http://www.infochimps.com/>

- For-profit data provider
- Lots of free and useful word-lists
- Links to publicly available data (census data, maps, etc.)

Stanford Linguistics corpus collection

- We subscribe to the LDC and so have most of their data sets:
<http://linguistics.stanford.edu/department-resources/corpora/inventory/>
- To get access, follow the instructions at this page:
<http://linguistics.stanford.edu/department-resources/corpora/get-access/>
- When you write to the corpus TA, cc the course staff address (the one you use for submitting work). Don't forget this step!
- Write from your Stanford address. That will help the corpus TA figure out who you are and how to grant you access.

Twitter API

- <https://dev.twitter.com/>
- As of this writing, the following command will stream a random sample of current tweets into a local file `mytweets.json`:

```
curl http://stream.twitter.com/1/statuses/sample.json  
-uUSER:PASS
```

where `USER` is your Twitter username and `PASS` your password.
- I think this will deliver ≈ 7 million tweets/day.
- But Twitter data requires *extensive* pre-processing. Tips:
 - Filter heuristically by language (don't rely only on "lang" field).
 - Filter spam based on tweet structure (spam warnings: too many hashtags, too many usernames, too many links)
 - Handle retweets in a way that makes sense given your goals.

Other APIs

- Kiva (micro-loans): <http://build.kiva.org/>
- eBay: <http://developer.ebay.com/common/api/>
- Yelp: <http://www.yelp.com/developers/documentation/v2/overview>
- Stack Exchange: <http://api.stackexchange.com/>

Scraping

- Link structure is often regular (reflecting a database structure).
- If you figure out the structure, you can often get lots of data!
- Once you have local copies of the pages:
 - Beautiful Soup (Python) is a powerful tool for parsing DOM structure.
 - Readability offers an API for extracting text from webpages.
- Be a good citizen! Don't get yourself (or your apartment, dorm, school) banned from the site.
- Beware highly restrictive, legally scary site policies! You don't want to run afoul of an aggressive, unfeeling, politically ambitious US Attorney.
- For more on crawler etiquette, see Manning et al. 2009 (<http://nlp.stanford.edu/IR-book/>).

A few special NLU data sets (open Web)

- Wikipedia data dumps:
http://en.wikipedia.org/wiki/Wikipedia:Database_download
- Stack Exchange data dumps:
<http://www.clearbits.net/torrents/2076-aug-2012>
- Switchboard Dialog Act Corpus:
<http://www.stanford.edu/~jurafsky/ws97/>
- Pranav Anand & co. (<http://people.ucsc.edu/~panand/data.php>)
 - Internet Argument Corpus
 - Annotated political TV ads
 - Focus of negation corpus
 - Persuasion corpus (blogs)
- Data I've made available as part of other courses and projects:
 - My data/code page:
<http://www.stanford.edu/~cgpotts/computation.html>
 - Extracting social meaning and sentiment:
<http://nasslli2012.christopherpotts.net>
 - Computational pragmatics
<http://compprag.christopherpotts.net>
- The Cards dialogue corpus:
<http://cardscorpus.christopherpotts.net>

A few special NLU data sets (on AFS)

Get access from the corpus TA, as described earlier:

- Nate Chambers' de-duped and dependency parsed NYT section of Gigaword: `/afs/ir/data/linguistic-data/GigawordNYT`
- Some of my own data sets on Stanford AFS (get access from the corpus TA, as described earlier):
 - `/afs/ir/data/linguistic-data/mnt/mnt4/PottsCorpora`
`README.txt`, `Twitter.tgz`, `imdb-english-combined.tgz`,
`opentable-english-processed.zip`
 - `/afs/ir/data/linguistic-data/mnt/mnt9/PottsCorpora`
`opposingviews`, `product-reviews`, `weblogs`
- Twitter data collected and organized by Moritz!
`/afs/ir.stanford.edu/data/linguistic-data/mnt/mnt3/TwitterTopics/`

Is there existing data for your project?

- 1 In just five (5!) minutes, see if you can find data for your project (or a topic you're interested in).
- 2 The above links should get you started, but search engines might take you where you want to go as well.
- 3 If you can't find data for your project, then crowdsource your woes by sharing them with the class when we reconvene.

Annotating data

- Suppose you can't find data for your project. Then you might consider annotating your own data, for training and/or assessment.
- This section briefly discusses such projects.
- We're not especially encouraging about having you launch an annotation project right now, at least not if your project depends on it.
- In the next section, we look at crowdsourcing, which is less risky (but more limited in what it can accomplish).

Setting up an annotation project

- Annotate a subset of the data yourself. This will reveal challenges and sources of ambiguity.
- Writing a detailed annotation manual will save you time in the long run, even if it delays the start of annotation.
- Consider a training phase for annotators, following by discussion.
- Consider whether your annotators should be allowed to collaborate and/or resolve differences among themselves.
- brat rapid annotation tool: <http://brat.nlplab.org>

Assessment

- Kappa is the standard measure of inter-annotator agreement in NLP. It works only where there are exactly two annotators and all of them did the same annotations.
- Fleiss kappa is suitable for situations in which there are multiple annotators, and there is no presumption that they all did the same examples.
- Both kinds of kappa assume the labels are unordered. Thus, they will be harsh/conservative for situations in which the categories are ordered.
- The central motivation behind the kappa measures is that they take into account the level of (dis)agreement that we can expect to see by chance. Measures like “percentage choosing the same category” do not include such a correction.

Sources of uncertainty

- Ambiguity and vagueness are part of what make natural languages powerful and flexible.
- However, this ensures that there will be uncertainty about which label to assign to certain examples.
- Annotators might speak different dialects, resulting in differing intuitions and, in turn, different label choices.
- Such variation will be systematic and thus perhaps detectable.
- Some annotators are better than others.

Pitfalls


- Annotations projects almost never succeed on the first attempt. This is why we don't really encourage you to start one now for the sake of your project.
- (Crowdsourcing situations are an exception to this, not because they succeed right way, but rather because they might take just a day from start to finish.)
- Annotation is time-consuming and expensive where experts are involved.
- Annotation is frustrating and taxing where the task is filled with uncertainty. Uncertainty is much harder to deal with than a simple challenge.

Example annotation results

	A	B	C	D	E	entropy	maj
ex1	1	1	0	0	1	0.67	1
ex2	0	-1	-1	-1	-1	0.72	-1
ex3	0	-1	-1	-1	-1	0.72	-1
ex4	-1	-1	-1	-1	-1	0	-1
ex5	1	1	-1	-1	-1	0.97	-1
ex6	0	1	1	1	-1	1.37	1
ex7	0	0	0	1	0	0.72	0
ex8	1	1	1	1	1	0	1
ex9	1	-1	-1	-1	1	0.97	-1
ex10	-1	1	1	-1	1	0.97	1
Deviation from maj.	6	1	1	3	2		
Mean Euc. dist.	3.56	3.04	2.87	3.19	3.38		
Mean correlation	0.43	0.72	0.70	0.66	0.58		

Fleiss kappa calculation on the HW 8 data

	A	B	C	D	E
ex1	0	0	1	0	0
ex2	-1	1	-1	1	-1
ex3	1	-1	-1	1	1
ex4	0	1	0	1	1
ex5	1	0	1	1	-1
ex6	-1	-1	-1	-1	-1
ex7	-1	0	-1	-1	-1
ex8	-1	0	-1	-1	-1
ex9	1	1	1	1	1
ex10	-1	1	-1	-1	1



	-1	0	1
ex1	0	4	1
ex2	3	0	2
ex3	2	0	3
ex4	0	2	3
ex5	1	1	3
ex6	5	0	0
ex7	4	1	0
ex8	4	1	0
ex9	0	0	5
ex10	3	0	2

Category	κ	s.e.	$z = \frac{\kappa}{\text{s.e.}}$	p
-1	0.39	0.34	1.16	0.25
0	0.25	0.25	1.04	0.30
1	0.28	0.31	0.89	0.37
Overall	0.32	0.10	3.24	0.001

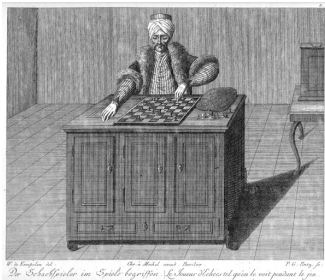
For details, Fleiss 1971.

Crowdsourcing

- You need new annotations.
- Your annotations can be done by non-experts.
- Perhaps you need *a ton* of annotations.
- Crowdsourcing might provide what you need, if you go about it with care.

The historical Mechanical Turk

Advertised as a chess-playing machine, but actually just a large box containing a human expert chess player.



From http://en.wikipedia.org/wiki/The_Turk

So Amazon's choice of "Mechanical Turk" to name its crowdsourcing platform is appropriate: **humans just like you are doing the tasks, so treat them as you would treat someone doing a favor for you.**

Outlets

There are many crowdsourcing platforms. The following are the ones that I have experience with:

- Amazon's Mechanical Turk: <https://www.mturk.com/>
- Crowdfunder: <http://crowdfunder.com/>
(handles quality control)
- oDesk: <https://www.odesk.com> (for expert work)

Outlets

There are many crowdsourcing platforms. The following are the ones that I have experience with:

- Amazon's Mechanical Turk: <https://www.mturk.com/>
- Crowdfunder: <http://crowdfunder.com/>
(handles quality control)
- oDesk: <https://www.odesk.com> (for expert work)

Currently, the biggest platforms are now where people are working for virtual currency inside of games. Rather than being paid a few cents per task to working on AMT, it is just as likely that someone is being paid right now in virtual seeds within an online farming game.

(Munro and Tily 2011:2–3)

Papers

- Munro and Tily (2011): history of crowdsourcing for language technologies, along with assessment of the methods.
- Crowd Scientist, a collection of slideshows highlighting diverse uses of crowdsourcing:
<http://www.crowdscientist.com/workshop/>
- 2010 NAACL workshop on crowdsourcing:
<http://aclweb.org/anthology-new/W/W10/#0700>
- Snow et al. (2008): early and influential crowdsourcing paper, finding that crowdsourcing requires more annotators to reach the level of experts but that this can still be dramatically more economical.
- Hsueh et al. (2009): strategies for managing the various sources of uncertainty in crowdsourced annotation projects.

Setting up a crowdsourcing project on MTurk

Examples

① **Mark promised Ellen to take out the trash.**

Which of the following two options better paraphrases the sentence?

- a. Mark made Ellen promise that she would take out the trash.
- b. Mark promised Ellen that he would take out the trash.

② **The owl is easy to see.**

Which of the following two options better paraphrases the sentence?

- a. It is easy to see the owl.
- b. The owl sees easily.

Setting up a crowdsourcing project on MTurk

Item, Target, Question, Response1, Response2

1, "Mark promised Ellen to take out the trash", "Which of the following two options better paraphrases the sentence?", "Mark made Ellen promise that she would take out the trash.", "Mark promised Ellen that he would take out the trash."

2, "The owl is easy to see", "Which of the following two options better paraphrases the sentence?", "It is easy to see the owl.", "The owl sees easily."



Mind-bending English sentences

Read the following short (somewhat unusual) sentence and choose the interpretation that seems most appropriate.

\$Target

\$Question

- \$Response1
- \$Response2

For the test items, there is no right answer, and you will not be judged according to your responses for those. However, there are interspersed, unidentified non-test items for which there are correct answers. We will be watching to see how you perform on these. These items are easy: if you read carefully, you are sure to get these right!

Setting up a crowdsourcing project on MTurk

Item, Target, Question, Response1, Response2

1, "Mark promised Ellen to take out the trash", "Which of the following two options better paraphrases the sentence?", "Mark made Ellen promise that she would take out the trash.", "Mark promised Ellen that he would take out the trash."

2, "The owl is easy to see", "Which of the following two options better paraphrases the sentence?", "It is easy to see the owl.", "The owl sees easily."

```

<h3>Mind-bending English sentences</h3>
<p>Read the following short (somewhat unusual) sentence and choose the interpretation that seems most appropriate.</p>
<p><b>$Target</b></p>
<p><i>$Question</i></p>
<table cellspacing="4" cellpadding="0" border="0">
  <tbody>
    <tr>
      <td valign="center"><input type="radio" value="$Response1" name="Response1" /></td>
      <td><span class="answertext">$Response1</span></td>
    </tr>
    <tr>
      <td valign="center"><input type="radio" value="$Response1" name="Response2" /></td>
      <td><span class="answertext">$Response2</span></td>
    </tr>
  </tbody>
</table>
<p>For the test items, there is no right answer, and you will not be judged according to your responses for those. However, there are interspersed, unidentified non-test items for which there are correct answers. We will be watching to see how you perform on these. These items are easy: if you read carefully, you are sure to get these right!</p>
  
```

Setting up a crowdsourcing project on MTurk

Edit Project

Specify the properties that are common for all of the HITs created using this project.

1 Enter Properties

2 Design Layout

3 Preview and Finish

Project Name: This name is not displayed to Workers.

Describe your HIT to Workers

Title

Describe the task to Workers. Be as specific as possible, e.g. "answer a survey about movies", instead of "short survey", so Workers know what to expect.

Description

Give more detail about this task. This gives Workers a bit more information before they decide to view your HIT.

Keywords

Provide keywords that will help Workers search for your HITs.

This project may contain potentially explicit or offensive content, for example, nudity. ([See details](#))

Setting up a crowdsourcing project on MTurk

Setting up your HIT

Reward per assignment
Tip: Consider how long it will take a Worker to complete each task. A 30 second task that pays \$0.05 is a \$6.00 hourly wage.

Number of assignments per HIT
How many unique Workers do you want to work on each HIT?

Time allotted per assignment **Minutes** ▾
Maximum time a Worker has to work on a single task. Be generous so that Workers are not rushed.

HIT expires in **Days** ▾
Maximum time your HIT will be available to Workers on Mechanical Turk.

Results are automatically approved in **Days** ▾
After this time, all unreviewed work is approved and Workers are paid.

[Advanced »](#)

Setting up a crowdsourcing project on MTurk

Advanced

[Worker requirements](#) ◀

Worker requirements:

Customize Worker Requirements...

Specify ALL the qualifications Workers must meet to work on your HITs:

Masters	◻	remove
HIT Approval Rate (%) for all Requesters' HITs	◻ greater than or equal to ◻ 95 ◻	remove
Number of HITs Approved	◻ greater than or equal to ◻ 1000 ◻	remove
Location	◻ is ◻ UNITED STATES ◻	remove

(up to 5)

Only Workers who qualify to do my HITs can preview my HITs.

Yes
 No

The “Master” qualification is expensive and puts you out of touch with many good workers. Better to design an explicit qualification task or include gold standard items to weed out trouble-makers.

On being a good requestor: before you run the task

- Log-in as a Worker at least once and check out what other Requesters are asking people to do. It can be pretty shocking. Vow to provide more interesting (less cynical) tasks.
- Strive to pay a fair wage. The ethics of pricing are very challenging.
 - If you pay too little, you might be exploiting people.
 - If you pay too much, your task might be coercive, especially if you allow workers from countries with a dramatically lower standard of living than yours.
- Check that your instructions make sense to non-specialists and that it is clear what you are asking people to do.
- Set the “Time allotted” to be higher than you expect, so that people who get distracted in the middle of the work aren’t penalized. This is especially important for long or intricate tasks.

On being a good requestor: while your HIT is live

- Join Turker Nation and advertise your HIT there in the “Everyone else” section. (Perhaps introduce yourself under “Requester introductions” first.)
- There is evidence that, at least in the U. S., it helps to mention that you’re a scientist. Many people Turk for the feeling of shared enterprise.
- Monitor the Turker Nation thread and respond to workers’ questions and concerns.
- Monitor the email account connected with your Turker account. People can send questions there.

On being a good requestor: when your results are all in

- Err on the side of approving people's work. It can be very hard to distinguish miscreants from people who were confused by something you said.
- Approve work in a timely fashion.
- If you screwed up, you absolutely have to pay for all the work that was done. Requesters who violate this tenet quickly find it very hard to get work done.
- MTurk is reputation based; it takes a long time for workers to “work off” a rejection.

On being a good requestor: when your results are all in

+ Post New Thread





















Threads 1 to 20 of 1147

Page 1 of 58 1 2 3 11 51 ... Last

Forum: Requesters Hall of Fame/Shame-RATINGS ONLY

This area is for rating Requesters - feel free to post your experience. This is NOT the area for discussing Requesters, that is Everyone Else.

⋮

 StanfordPragLab Started by pewi, 11-22-2011 04:00 PM	Replies: 6 Views: 797	erod429 01-31-2013, 09:55 PM 
 Research Project Started by SummerSerenityRelic, 08-20-2012 01:17 PM 1 2	Replies: 11 Views: 332	SileMarie 01-31-2013, 07:07 PM 
 KaBOOM! Started by jamie9100, 01-31-2013 05:53 PM	Replies: 0 Views: 35	jamie9100 01-31-2013, 05:53 PM 
 raINC (Adult HITS) - good requester, crap pay Started by TyTurker, 11-10-2012 12:23 AM	Replies: 6 Views: 176	taintturk 01-30-2013, 04:59 PM 
 NetMSi / Aric M / TMI9 / Nick Scott (adult hits) Started by Gulfcoaster, 01-19-2012 01:11 AM 1 2 3 ... 5	Replies: 49 Views: 2,575	lolabear 01-29-2013, 06:30 PM 
 p9rp9r - Requester P9R Started by dbrew101, 01-16-2012 01:04 AM 1 2 3 ... 4	Replies: 38 Views: 1,282	rgarner13 01-29-2013, 09:26 AM 
 Checkout 51 - rejections for no reason Started by SileMarie, 01-28-2013 01:56 PM	Replies: 0 Views: 34	SileMarie 01-28-2013, 01:56 PM 
 ProductRnR Started by taintturk, 08-16-2011 02:41 PM 1 2 3 ... 11	Replies: 103 Views: 3,413	Timothyj 01-28-2013, 11:13 AM 
 Adrian Chira - good requester, crap pay Started by ciocci, 01-27-2013 05:35 PM	Replies: 0 Views: 38	ciocci 01-27-2013, 05:35 PM 
 Project Endor Started by gabby354, 01-27-2013 05:03 PM	Replies: 0 Views: 24	gabby354 01-27-2013, 05:03 PM 

Other tips for getting good results

- Consider launching your task during the workday in the area you are trying to get workers from. My impression is that this results in the best work.
- Munro and Tily (2011): researchers might suffer less from scammers for two reasons:
 - Their jobs are typically too small for it to be worth the trouble to write a script to automate responses to it.

(If you need a massive number of responses, run in small batches.)
 - Their jobs are quirrier, making it harder to write such scripts.

(Avoid using the sample templates MTurk provides.)

Where it works and where it probably won't

- One hears that crowdsourcing is just for quick, simple tasks.
- This has not been my experience. We have had people complete long questionnaires involving hard judgments.
- To collect **the Cards corpus**, we used MTurk simply to recruit players to play a collaborative two-person game.
- If you post challenging tasks, you have to pay well.
- There are limitations, though:
 - If the task requires any training, it has to be quick and easy (e.g., learning what your labels are supposed to mean).
 - You can't depend on technical knowledge.
 - If your task is highly ambiguous, you need to reassure workers and tolerate more noise than usual.

Competence questions

- I encourage you to pay essentially everyone who does your task, blocking only obvious scammers.
- However, you'll still want to have some questions that are easy and that you know the answer to, so you can heuristically spot sources of bad data.
- Crowdfunder will require you to provide such “gold standard” annotations.

Project set-up

Now that you've got your data set more or less finalized, you can get started on the experiments.

Data

- It will pay to get your data into an easy-to-use form and write general code for reading it.
- If your data-set is really large, considering putting it in a database or indexing it, so that you don't lose a lot of development time iterating through it.

Additional annotations/structure

- If there's a chance that you might need additional structure — POS tags, named-entity tags, etc. — consider adding it now.
- The Stanford NLP group has released lots of software for doing this.
- The code takes the form of Java libraries and can also typically be used from the command-line:

<http://www-nlp.stanford.edu/software/index.shtml>

- Check out CoreNLP in particular — amazing!

Train/dev/test splits

- Set aside some data now as a test set.
- Don't look at the test set until your project is complete except for filling in the final results table and writing the error analysis.
- If you can afford it, create a development set as well, for interim assessments.
- You can also do random train/test splits on your training data and cross-validation on the training data as well.
- Deciding on a train/dev/test split is subtle and fraught issue. The right decisions here will depend on what you are trying to accomplish.
- (Some of you will be lucky enough to have a train/(dev)/test split defined for you, say, because the data were used in a bake-off. In that case, use that split so that you get the most precise comparisons with existing systems.)

Optimal train/test split?

What's the best way to divide up the following corpus:

Movie	Genre	Review count
Jaws	Action	250
Alien	Sci-Fi	50
Aliens	Sci-Fi	40
Wall-E	Sci-Fi	150
Big	Comedy	50
Ran	Drama	200

Optimal train/test split?

What's the best way to divide up the following corpus:

Movie	Genre	Review count
Jaws	Action	250
Alien	Sci-Fi	50
Aliens	Sci-Fi	40
Wall-E	Sci-Fi	150
Big	Comedy	50
Ran	Drama	200

Answer: depends on what you're doing!

Conceptualizing your task

Table 1. The three components of learning algorithms.

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Write your own or off the shelf?

There is great value in implementing algorithms yourself, but it is labor intensive and could seriously delay your project. Thus, we advise using existing tools where possible for this project:

- Stanford Classifier (Java): <http://nlp.stanford.edu/software/classifier.shtml>
- Stanford Topic Modeling Toolbox (Scala): <http://nlp.stanford.edu/software/tmt/tmt-0.4/>
- MALLET (Java): <http://mallet.cs.umass.edu/>
- FACTORIE (Scala): <http://factorie.cs.umass.edu/>
- LingPipe (Java): <http://alias-i.com/lingpipe/>
- NLTK (Python): <http://nltk.org/>
- Gensim (Python): <http://radimrehurek.com/gensim/>
- GATE (Java): <http://gate.ac.uk/>
- scikits.learn (Python): <http://scikit-learn.org/>
- Lucene (Java): <http://lucene.apache.org/core/>

Benchmarks

How will you know when you've succeeded?

- 1 *Weak baselines*: random, most frequent class
- 2 *Strong baselines* (and the desirability thereof): existing models and/or models that have a good chance of doing well on your data
- 3 *Upper bounds*: oracle experiments, human agreement (non-trivial; human performance is rarely 100%!)

Project development cycle

Your project is set up. Now the fun begins!

Development methodology

- 1 Construct a tiny toy data set for use in system development
- 2 Iterative development:
 - a. Get a baseline system running on real data ASAP.
 - b. Implement an evaluation — ideally, an automatic one, but could be more informal if necessary.
 - c. Hill-climb on your objective function, using human intelligence.
 - d. Feature engineering cycle: add features \Rightarrow eval on development data \Rightarrow error analysis \Rightarrow generalizations about errors \Rightarrow brainstorming \Rightarrow add features
- 3 Research as an “anytime” algorithm: have some results to show at every stage
- 4 Consider devising multiple, complementary models and combining their results (via max/min/mean/sum, voting, meta-classifier, ...).
- 5 Grid search in parameter space:
 - can be useful when parameters are few and train+test is fast
 - easy to implement
 - informal machine learning

Focus on feature engineering

- Finding informative features matters more than choice of classification algorithm.

Domingos (2012:84): “At the end of the day, some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used.”

- Do error analysis and let errors suggest new features!
- Look for clever ways to exploit new data sources.
- Consider ways to combine multiple sources of information.

Evaluation techniques

Understanding your system's performance:

- Confusion matrices to spot problem areas and overlooked oddities.
- Check performance on your training data to spot problems with over-fitting.
- Visualization to make multiple formal and informal comparisons and identify overlooked relationships.
 - t-SNE for 2d visualization of high-dimensional data:
<http://homepage.tudelft.nl/19j49/t-SNE.html>
 - Gephi: <http://gephi.org/>
 - Visualization tools from Jeff Heer's group:
<http://hci.stanford.edu/jheer/>

(Evaluation will be covered more fully in workshop 2.)

What are you learning?

Above all else, your project should teach us something new.

- It's very nice if you achieve state-of-the-art performance.
- So-called null results are also valuable.
- Even if you don't beat all competitors, there are likely to be aspects of your system that are useful and informative.
- In other words, the worse-case scenario should be that your error analysis is the most valuable part of your paper.

Conclusion

- Lit review (due Feb 14)
- Getting data (with luck, you can work with existing data)
- Annotating data (difficult, time-consuming)
- Crowdsourcing (less difficult and time-consuming; still requires care)
- Project set-up (lay the groundwork soon)
- Project development cycle (rapid iteration, learning as you go)

References I

- Domingos, Pedro. 2012. A few useful things to know about machine learning. *Communications of ACM* 55(10):78–87.
- Fleiss, Joseph I. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76(5):378–382.
- Hsueh, Pei-Yun; Prem Melville; and Vikas Sindhwani. 2009. Data quality from crowdsourcing: A study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, 27–35. Boulder, Colorado: Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W09-1904>.
- Manning, Christopher D.; Prabhakar Raghavan; and Hinrich Schütze. 2009. *An Introduction to Information Retrieval*. Cambridge University Press.
- Munro, Rob and Harry J. Tily. 2011. The start of the art: An introduction to crowdsourcing technologies for language and cognition studies. Ms., Stanford University and MIT, URL http://www.crowdscientist.com/wp-content/uploads/2011/08/start_of_the_art.pdf.

References II

Snow, Rion; Brendan O'Connor; Daniel Jurafsky; and Andrew Y. Ng. 2008.
Cheap and fast — but is it good? Evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 254–263. ACL.