PROJECTS

It is time to form your teams (ideally groups of two… bigger groups allowed only after special request, groups of one allowed but strongly discouraged). I'd like a 1 page, 10 pt font note from you by March 19th describing your project. Use the following rubric for your writeup.

Hypothesis/Goal: Your project should begin with a hypothesis statement or a goal. Example hypothesis statements are: "using dependency parser based features substantially impacts a review's sentiment", "design of sentiment mining system that outputs various aspects and associated polarity and its strength"

Datasets: Each NLP project requires some input dataset(s). It is important to identify what dataset(s) you would work on. Ideally, the dataset is downloadable today or is very easy to create. Our timeline does not allow a massive data curation effort, except if generating the data itself is the goal (unlikely). So, my request is that you should be able to acquire/curate the data by end of this month, if not much before. That way you will have enough time to work on the project.

Literature/Code Search: After you have identified the dataset and the goal of the project, it is important to do some basic literature search to check what all has been done. Since this is a course project it is ok if you implement someone else's paper. Personally, I'd much rather prefer if you started with a state of the art system and made some (even if small) modifications to improve the system. The good news is that this is relatively easy to do since a lot of papers release their code. So, a simple strategy is to find some recent paper (relevant conferences are ACL, NAACL, EMNLP), download/ask for their codes and replicate their reported results. If you can do this asap, then you can focus the rest of the time on improving those results. This could even lead to a research publication down the line.

Evaluation: Finally, you need to clearly specify how you will evaluate your hypothesis/system. What will be the experiment? What will be the metric? If there is any annotation, where will gold labels come from, etc.

Demo: The expectation of every project is that you will have a demo accessible over the Web. This is good for you since you can show it to your prospective employers. It is good for me since I can brag about how smart IITD-NLP students are. There will be extra credit for a fully working demo integrated with our demo page (www.cse.iitd.ac.in/nlpdemo) before the end of grading

Project Ideas

Word Vector Embeddings: Deep learning for NLP has become super-popular. Google (and many others) have released their word vector embeddings. https://code.google.com/p/word2vec/

1. See if you can improve the quality of embeddings by adding additional insights such as in papers like Wordnet http://www.manaalfaruqui.com/papers/naacl15-retrofitting.pdf or Open IE http://www.cse.iitd.ac.in/~mausam/papers/acl15.pdf or context http://allennlp.org/elmo.
2. Lots of important issues with word embeddings have been listed on this blog. This exposes many interesting problems with word embeddings. Fix one? http://ruder.io/word-embeddings-2017/
3. Learn word vector embeddings on a new genre such as Twitter. Use existing Twitter methods (such as lexical normalization that makes 2moro → tomorrow) in the process of learning embeddings.
4. Brown clusters give a hierarchical feature representation that can be used in sequence labeling/classification. Word vectors don't give a dense distributed vector which can also be used in sequence labeling. Explore the value of both kinds of features. Compare them and contrast them for several problems.

MultiModal NLP: Recently because of neural networks being a general representation, there is new emphasis on image captioning, video captioning and likes. Check out latest papers on the topic such as http://aclweb.org/anthology/D17-1098 and see if you can improve the existing ideas.

Social NLP: Developing tools on Twitter/Facebook are always fun and exciting. Some possible ideas for processing tweets below:

5. Lexical Tweet Normalization: Automatically rewriting tweet spellings into grammatical spellings. Example: 4ward → forward; 2moro → tomorrow; …
6. Grammatical Tweet Normalization: Automatically rewrite tweet into a grammatical sentence.
7. Tweet Information Extraction: Develop Open IE (https://github.com/dair-iitd/OpenIE-standalone) for tweets.
8. Tweet-based Timelines Identify the salient sub-events in a sporting (or other popular events) based on amount of traffic that gets generated.
9. Tweet-based Summarization: Given a product or an entity of interest, create a summary page identifying (i) trending topics on the entity, (ii) representative tweets, (iii) sentiment distributions, etc.
10. Tweet Sentiment Mining++: Given a new entity, identify the various aspects of the entity, find the average sentiment along that aspect and give representative tweets. (E.g., iPhone will have aspects as 'battery life', 'screen size', etc).
11. NLP for Friends: Analyze status updates of friends and 'Likes' of friends to create a user model of each friend. Identify the kind of posts your friends like and dislike(?). Predict a friend's like behavior.
12. Interpreting Social Networks: Learn what is common between friends. Learn what is common in communities (or cliques).
13. Combine Twitter dataset with some other dataset to learn about correlations. For example, connect twitter stream with cricinfo stream. Or twitter stream with soccer commentary. Or

twitter stream with prime minister's address to the nation. Depending upon which datasets you connect you might be able to accomplish different artifacts.

Domain-dependent NLP: Most of NLP pipeline has been trained on New York Times corpus, and therefore works great on news but not on other domains. Explore retraining parts of NLP pipeline (POS tagger, NP chunker, Named entity recognizer, syntactic parser, coreference resolution, etc) on a very different domain. Different domains enable other interesting end tasks – for e.g., app recommendations based on query word, prediction of success of a movie based on the script, etc. Example domains include:

- Sports stream
- Scientific documents (research papers in NLP, PubMed abstracts)
- Hiking trails in national parks
- Food websites discussing recipes
- Ehow.com style websites that provide instructions to complete some task
- E-education discussion boards on Coursera or other MOOC portals
- Movie scripts
- Comics
- Android/iPhone app descriptions
- …

Information Extraction: Mining information from textual corpora. For example,

- mining definitions of scientific terms from scientific corpora (e.g., what is expectation maximization),
- mining temporal information alongside common facts (when did a fact begin and when it ended, e.g., marriage or employment),
- mining lists of instances for a given type (e.g., German authors born in 1800s)
- mining the list of diseases and appropriate medicines
- We recently had a paper on mining comparisons between entities. Preprint copy available here: http://www.cse.iitd.ac.in/~mausam/courses/col772/spring2016/papers/naacl16.pdf. The code is available if needed. Improve the quality of comparisons to incorporate attribute-value comparisons. You could alternatively apply this idea in a new domain, liking mining comparisons between two cellphones, two politicians, etc.
- We recently released our latest version of Open IE system. https://github.com/dair-iitd/OpenIE-standalone. Test this on generic datasets and improve the quality of results.
- You can use the mined information to detect trends over time, for example, which concepts were more active in which era You can use existing softwares to start information extraction process. Example: http://cs.nyu.edu/grishman/jet/jet.html, http://reverb.cs.washington.edu/, http://nlp.stanford.edu/software/index.shtml.

Event Extraction: Another kind of information extraction tries to extract events. For example, court case, there will be a judge, a defendant, a prosecutor, two lawyers, etc. Can you extract instances of a court case and identify each entity and its role? There are many such types of events. Extracting them can be non-trivial.

Knowledge-bases: Several papers have been written on KB completion and inference tasks. A discussion to get you started is here: https://arxiv.org/abs/1706.00637 and a latest state of the art is here: http://proceedings.mlr.press/v48/trouillon16.pdf. Improve one of the top models OR combine several existing models to get best of all world or robustness in results. You may also study fine grained typing of named entities (e.g., albums, bands, movies, etc) using the new dataset: https://arxiv.org/abs/1711.05795.

Question Answering/Dialog/Machine Comprehension: There is huge recent working in these areas such as on SQUAD dataset (https://rajpurkar.github.io/SQuAD-explorer/), Maluuba datasets (https://datasets.maluuba.com/), Facebook datasets (https://research.fb.com/downloads/babi/) and likes. With each dataset, there are associated state of the art papers. A natural approach will be to start from one such paper and attempt to do better.

Summarization: Summarization is an important problem with the goal of reducing information overload. We have a good but older paper and software called GFLOW (http://knowitall.cs.washington.edu/gflow/).

- Evaluate the value of GFLOW for a new domain of interest, for example, sports stories, stock market news reports, scientific documents, legal proceedings, product reviews, newsgroup threads, etc. If GFLOW does not work, modify it to improve performance.
- You can also generate a comparison summary based on our NAACL work (see above in IE section).

Other NLP problems: Build neural models for understanding compound nouns (e.g., the latent relationship between ice and cream in ice-cream; or between plastic and mug in plastic mug; tea cup; etc). Or build neural models for understanding difficult prepositions (e.g., of). For example, differentiating between the hidden relationships expressed by 'of' in "Bay of Bengal" versus "Indian Institute of Technology" versus "University of Massachusetts", etc.

- Simple papers evaluating CNNs and RNNs for text classification have been written. See https://arxiv.org/pdf/1702.01923.pdf. Similar papers can be attempted for models like bidirectional LSTMs, Transformer and others.

Named Entity Recognizer: While NER systems for Person, Organization and Location are very common, they do not exist for many other types. Can you train an NER to recognize softwares, or band names or movies or course names, etc? How about creating a named entity recognizer for Twitter or another genre like legal documents?

Part of Speech Tagger: While English POS tagging has a very high performance you could consider retraining a POS tagger for tweets, or fb status updates. You could work with different languages, e.g., Hindi or Marathi.

Text Categorization: Last, but not the least, there are myriads of opportunities for categorizing a piece of text. For example, you could try to categorize if a given tweet is factual or subjective? You could try to predict if a movie will be a hit or a flop. You could predict whether a specific stock price will go up or down based on current news. You could categorize what category a given tweet belongs to. And so on…

Other ideas in open competitions: Every year many shared tasks are organized in which various systems compete around the world. That will give you more ideas for current relevant projects. There are also open competitions in NLP. These include:

- SemEval Tasks: http://alt.qcri.org/semeval2018/ (or older competitions)
- CONLL Shared Task on Multilingual Parsing (or older competitions): http://universaldependencies.org/conll18/
- TACKBP competitions on Knowledge Base Population: http://www.nist.gov/tac/
- Kaggle competitions on NLP
- Answering 8th grade science questions: http://allenai.org/data.html
- Winograd Schema Challenge: http://commonsensereasoning.org/winograd.html
- …

These are only a few ideas. You can explore many other ideas around coreference resolution, commonsense knowledge extraction, semantic role labeling, compositional semantics, parsing, etc. You can also look at some other shared tasks.