Project Ideas for CS224N, 2014-15

NLP and ML researchers at Stanford were approached for NLP related project ideas appropriate for CS224N. Their proposals are listed in this document, in no significant order (the order of submission time). You can also comment on this document if you like.

Feel free to contact the person who proposed each project for more information / advice. Their emails can be found below the project title. You are encouraged to set up a meeting to discuss more about the projects that you are interested in.

In addition to the descriptions in this document, some project proposals came with extra slides and they can be found here:

<http://web.stanford.edu/class/cs224n/handouts/cs224nprojectideas.pdf>

Past projects can be found here:

<http://nlp.stanford.edu/courses/cs224n/>

and the 2013 projects can be found here:

Of course, you do not have to restrict your search scope to these projects. You are free to work on any NLP related project that interests you, though we do need to see a proposal to make sure it is reasonable for the class. Have fun.

[Projects](#h.imuq7omqyd10)

[How does sentence embedding accuracy scale with sentence size and embedding size](#h.11g34h7k49r1)

[Chat bot with real NLP](#h.gfd5xqngk7sr)

[Learning distributed representations for variable-length texts](#h.jog9148s4n8g)

[Using global context in joint neural language model](#h.jxlmkgxfwx90)

[Nonsense Detector](#h.rmu9vq7s85tb)

[Sublexical compositionality in semantic parsing](#h.cru9uty1hxfy)

[Automatically improving dependency annotation](#h.zi1ovvjteof3)

[ML-based tokenizer](#h.pzrag0jpxl15)

[ML based grammar correction](#h.ll6cfy9x5a3g)

[Net Neutrality Language Analysis](#h.9ujs2o7kephf)

[Analyzing/Visualizing/Summarizing (Yelp) Restaurant Reviews with Neural Nets](#h.lscgl6jjcoq6)

# Projects

## How does sentence embedding accuracy scale with sentence size and embedding size

Sam Bowman [sbowman@stanford.edu](mailto:sbowman@stanford.edu)

Fixed length vector representations of sentences – generated from neural network models like recursive NNs or LSTMs – are fairly widely used in NLP (most recently in some groundbreaking translation work). Intuitively, vector representations with only one or two floats cannot accurately distinguish a significant number of natural language sentences. However, for larger vectors (hundreds dimensions) and reasonable sentence lengths (dozens of words), essentially all of the information in the sentence is captured by the vector.[1] I don't believe that there are any clear results out there on the relationship between the size of the embedding and the degree to which it can accurately represent a text of a given length.

This project would involve building a relatively simple RNN or LSTM model to embed sentences as vectors and then unfold them back into sentences [2], and try to evaluate how long of sentences can be recovered verbatim from different lengths of embedding vector. Error analysis could include evaluating what types of sentences (genres, syntactic structures, ...) tend to be recovered more accurately.

[1] For example, see: http://arxiv.org/pdf/1409.0473.pdf

[2] For recursive NNs: <http://papers.nips.cc/paper/4204-dynamic-pooling-and-unfolding-recursive-autoencoders-for-paraphrase-detection.pdf>

**Recommended Background**: Some skill in MATLAB could help

## 

## 

## Chat bot with real NLP

Gabor Angeli [angeli@stanford.edu](mailto:angeli@stanford.edu)

I've always found it somewhat strange that there's little to no NLP research [anymore] on making chat bots. The result is that most seem to blindly parrot other user's input, or revert to inane general conversation.

The project proposal is to combine a number of common NLP techniques and systems to create a more ""intelligent"" chat bot. For instance:

\* Learn common conversation patters from subtitles / scripts (e.g., http://www.tvsubtitles.net/tvshow-32-1.html) -- I suspect this would be the real ""technical"" part of the project.

\* Add in SemPre (https://github.com/percyliang/sempre) for freebase factoid Q/A

\* Make a few cute custom grammars for SemPre (e.g., for arithmetic; for questions like ""who created you?"", etc.)

\* Throw in a simple sentiment detector, and tailor comments appropriately.

\* And more!

**Recommended Background**: Interest/experience in pragmatics

## 

## 

## Learning distributed representations for variable-length texts

Thang Luong [lmthang@stanford.edu](mailto:lmthang@stanford.edu)

In this project, we will work on a recently popular research topic in NLP that is to learn distributed (vector) representations not just for words, but for longer text spans (phrases, sentences, paragraphs, and documents). Learning such representations has been proven to be much better than bag-of-word models and achieved state-of-the-art representations in many tasks as shown in this paper http://cs.stanford.edu/~quocle/paragraph\_vector.pdf.

However, such method has a disadvantage that it requires an inference step for new text spans. We'd like to experiment with several functions to compute representations for a text span from its individual words and train in a similar fashion as the reference paper.

## Using global context in joint neural language model

Thang Luong [lmthang@stanford.edu](mailto:lmthang@stanford.edu)

In this project, we examine neural network language model (NNLM) in the context of machine translation in which the NNLM is jointly conditioned on not only the target words but also source words. Such type of joint NNLM has improved MT quality significantly as shown in this paper http://acl2014.org/acl2014/P14-1/pdf/P14-1129.pdf. However, only one part of the source sentence is considered at a time for each prediction of the target words. We'd like to examine if using the entire source sentence can help improve the model similar in spirit to this work <http://www.socher.org/index.php/Main/ImprovingWordRepresentationsViaGlobalContextAndMultipleWordPrototypes>

**Recommended Background**: Python, machine learning experience.

Notes:

(a) Software: develop from my Python/Theano software.

(b) My offer: I will help you learn Theano (a popular deep learning library) and contribute ideas/advices on techniques and training data.

(c) My expectation: eventually, it's great to push for a paper submission, so I expect you to take have time to work on the project seriously and early. We should meet regularly, say once per week or every two weeks.

## Nonsense Detector

Gabor Angeli [angeli@stanford.edu](mailto:angeli@stanford.edu)

Classify whether a sentence is a ""well-formed sentence."" In particular, the internet is full of a bunch of nonsense that we regularly treat as not-nonsense. This includes sentence fragments, headlines, YouTube comments, etc. The end result is that including these sentences tends to introduce noise in downstream tasks.

The project would be to create a classifier to try to filter out these ""bad"" sentences -- for example, by looking for uncommon POS tag sequences, lack of punctuation (too much punctuation?), etc."

## Sublexical compositionality in semantic parsing

Jonathan Berant [joberant@stanford.edu](mailto:joberant@stanford.edu)

Abstraction level in language differs from the one in KBs. Specifically, some words in language correspond to compositional logical forms on the KB side. For example, ""actress"" is presented in the knowledge-base as the intersection of actors and females, ""grandfather"" is represented as the intersection of males and people whose children have children, and ""relative"" is the union of ""parent"", ""child"", ""sibling"" and so on.

Semantic parsing on large KBs involves creating lexicons that provide a noisy mapping from text to the KB. Usually, the mapping is restricted to non-compositional logical forms. The goal of the project is to create a mapping for more compositional words such as ""actress"", ""relative"", ""president"" and so on.

## Automatically improving dependency annotation

Natalia Silveira [natalias@stanford.edu](mailto:natalias@stanford.edu)

Quality annotations are crucial for many NLP applications, but annotating large amounts of data consistently and correctly is very difficult. A team in the NLP group annotated about 250k words of web data with syntactic dependencies, and now we're looking for ways to automatically improve the quality of these annotations, mainly by identifying possible errors. The project would to implement a classifier that detects annotation errors using features of the annotation, based on existing literature. This will allow students to learn a lot about dependency syntax, and understand some of the challenges involved in creating useful NLP resources -- as well as contribute to the NLP community, since this will be a widely distributed dataset.

**Recommended Background**: Interest in the more linguistic aspects of NLP

## ML-based tokenizer

Christopher Manning [manning@stanford.edu](mailto:manning@stanford.edu)

The Stanford NLP tools use a hand-written deterministic FSM tokenizer. It's actually pretty good, but it can't consider word forms that are natural to humans, and so it doesn't get right things like when two words are runtogether and nor can it decide when a parenthesis followed by a colon is intended punctuation (which sometimes happens): versus when it is a backwards frown face, which also sometimes happens. ): Could a machine learning classifier work better? It could be a sequence model done as a CRF like Chinese word segmentation, but could well do better using longer range context like a word-level language model or certain kinds of recurrent neural net. There has been some prior work on this using somewhat unsupervised means (search punkt).

## 

## 

## ML based grammar correction

Sida Wang sidaw@stanford.edu

Writing in English can be especially hard for people who learned it as a second language. Existing checkers in popular document processing programs do not catch many of these common mistakes. For example:

This essay will [discuss about → discuss] whether a carrier should tell his relatives or not.

The benefits of disclosing genetic risk information [outweighs → out- weigh] the costs.

It is obvious to see that [internet → the internet] saves people time and also connects people globally.

Medical technology during that time [is → was] not advanced enough to cure him.

A good amount of data is collected on this from the CoNLL 2013, 2014 shared tasks

<http://www.comp.nus.edu.sg/~nlp/conll13st.html>

and

<http://www.comp.nus.edu.sg/~nlp/conll14st.html>

So now is a good time to look at some of the top systems and see if similar or better results can be obtained without extensive feature engineering but with a automatic feature generation / feature selection procedure or a deep learning procedure.

## Net Neutrality Language Analysis

Dan Jurafsky [jurafsky@stanford.edu](mailto:jurafsky@stanford.edu)

The FCC has given us all the comments that citizens made to the FCC about the topic of net neutrality, whether the internet should be a common carrier, etc. The task is to do unsupervised clustering, topic modeling, and such tasks to see if we can learn something about what the gist of people's comments are.

## 

## 

## Analyzing/Visualizing/Summarizing (Yelp) Restaurant Reviews with Neural Nets

Andrej Karpathy [karpathy@cs.stanford.edu](mailto:karpathy@cs.stanford.edu)

Have you ever found yourself scrolling through pages and pages of Yelp reviews for a restaurant, trying to figure out what is special about this place, or what people agree on? In this project you will use Yelp dataset API to automatically create meta-reviews that analyze all reviews for a restaurant and highlight/summarize or visualize the most informative or interesting bits.

**Recommended Background**: Calculus, CS229