Assignment 2 deadline pushed to Tuesday noon

Error in Assignment 2:

Use “Pikachu”, “Charizard” and “Charmander”
(as opposed to “pikachu”, “charizard”, “charmander”)

Using NLTK n-grams is ok, but I think you could implement it on your own.
STRUCTURE OF THIS LECTURE

1. Why sentence representations?
2. Multi-task Learning
3. Training Sent. Representations
4. Contextualized Embeddings
We can create a vector or sequence of vectors from a sentence

this is an example

this is an example

Obligatory Quote!

“You can’t cram the meaning of a whole %&!*ing sentence into a single $&!*ing vector!”

— Ray Mooney
GOAL FOR TODAY

Briefly Introduce tasks, datasets and methods

Introduce different training objectives

Talk about multitask/transfer learning
TASKS USING SENTENCE REPRESENTATIONS
WHERE WOULD WE NEED/USE SENTENCE REPRESENTATIONS?

Sentence Classification
Paraphrase Identification
Semantic Similarity
Entailment
Retrieval
SENTENCE CLASSIFICATION

Classify sentences according to various traits

Topic, sentiment, subjectivity/objectivity, etc.

I hate this movie

I love this movie
In addition to standard tags, each constituent tagged with a sentiment value.
Identify whether A and B mean the same thing

Charles O. Prince, 53, was named as Mr. Weill’s successor.

Mr. Weill’s longtime confidant, Charles O. Prince, 53, was named as his successor.

• **Note:** exactly the same thing is too restrictive, so use a loose sense of similarity
SEMANTIC SIMILARITY/RELATEDNESS (MARELLI ET AL. 2014)

Do two sentences mean something similar?

<table>
<thead>
<tr>
<th>Relatedness score</th>
<th>Example</th>
</tr>
</thead>
</table>
| 1.6               | A: “A man is jumping into an empty pool”  
B: “There is no biker jumping in the air” |
| 2.9               | A: “Two children are lying in the snow and are making snow angels”  
B: “Two angels are making snow on the lying children” |
| 3.6               | A: “The young boys are playing outdoors and the man is smiling nearby”  
B: “There is no boy playing outdoors and there is no man smiling” |
| 4.9               | A: “A person in a black jacket is doing tricks on a motorbike”  
B: “A man in a black jacket is doing tricks on a motorbike” |

• Like paraphrase identification, but with shades of gray.
Entailment: if A is true, then B is true (c.f. paraphrase, where opposite is also true)

The woman bought a sandwich for lunch
→ The woman bought lunch

Contradiction: if A is true, then B is not true

The woman bought a sandwich for lunch
→ The woman did not buy a sandwich

Neutral: cannot say either of the above

The woman bought a sandwich for lunch
→ The woman bought a sandwich for dinner
MODEL FOR SENTENCE PAIR PROCESSING

Calculate vector representation

Feed vector representation into classifier

How do we get such a representation?

this is an example

this is another example

classifier → yes/no
MULTI-TASK LEARNING OVERVIEW
Multi-task learning is a general term for training on multiple tasks.

Transfer learning is a type of multi-task learning where we only really care about one of the tasks.

Domain adaptation is a type of transfer learning, where the output is the same, but we want to handle different topics or genres, etc.
In NLP, there are a plethora of tasks, each requiring different varieties of data

**Only text:** e.g. language modeling

**Naturally occurring data:** e.g. machine translation

**Hand-labeled data:** e.g. most analysis tasks

And each in many languages, many domains!
RULE OF THUMB 1: MULTITASK TO INCREASE DATA

Perform multi-tasking when one of your two tasks has many fewer data

General domain $\rightarrow$ specific domain
(e.g. web text $\rightarrow$ medical text)

High-resourced language $\rightarrow$ low-resourced language
(e.g. English $\rightarrow$ Telugu)

Plain text $\rightarrow$ labeled text
(e.g. LM $\rightarrow$ parser)
RULE OF THUMB 2: TASK RELATEDNESS

Perform multi-tasking when your tasks are related

e.g. predicting eye gaze and summarization (Klerke et al. 2016)
STANDARD MULTI-TASK LEARNING

Train representations to do well on multiple tasks at once

- In general, as simple as randomly choosing minibatch from one of multiple tasks
- Many many examples, starting with Collobert and Weston (2011)
PRE-TRAINING

First train on one task, then train on another

- This is an example
  - Encoder
  - Translation

  Initialize

- This is an example
  - Encoder
  - Tagging

- Widely used in word embeddings (Turian et al. 2010)
- Also pre-training sentence encoders or contextualized word representations (Dai et al. 2015, Melamud et al. 2016)
THINKING ABOUT MULTI-TASKING, AND PRE-TRAINED REPRESENTATIONS

Many methods have names like SkipThought, ParaNMT, CoVe, ELMo, BERT along with pre-trained models
These often refer to a combination of

**Model:** The underlying neural network architecture

**Training Objective:** What objective is used to pre-train

**Data:** What data the authors chose to use to train the model

Remember that these are often conflated (and don't need to be)!
For any model, we can always use an end-to-end training objective

**Problem:** paucity of training data

**Problem:** weak feedback from end of sentence only for text classification, etc.

Often better to pre-train sentence embeddings on other task, then use or fine tune on target task
TRAINING SENTENCE REPRESENTATIONS
GENERAL MODEL OVERVIEW

I hate this movie

lookup lookup lookup lookup

some complicated function to extract combination features

scores

softmax

probs
Model: LSTM
Objective: Language modeling objective
Data: Classification data itself, or Amazon reviews

• **Downstream:** On text classification, initialize weights and continue training
CONTEXTUALIZED WORD REPRESENTATIONS
Instead of one vector per sentence, one vector per word!

How to train this representation?
BI-DIRECTIONAL LANGUAGE MODELING OBJECTIVE (ELMO; PETERS ET AL. 2018)

- **Model**: Multi-layer bi-directional LSTM
- **Objective**: Predict the next word left->right, next word right->left independently
- **Data**: 1B word benchmark LM dataset

**Downstream**: Finetune the weights of the linear combination of layers on the downstream task
Like ELMo, uses bidirectional context, but with transformer model as base (+ tricks for efficient training)

- **Model**: Multi-layer self-attention. Input sentence or pair, w/ [CLS] token, subword representation

- **Objective**: Masked word prediction + next-sentence prediction

- **Data**: BooksCorpus + English Wikipedia
SELF-ATTENTION

From http://jalammar.github.io/illustrated-transformer/
SELF-ATTENTION

From http://jalammar.github.io/illustrated-transformer/
SELF-ATTENTION

From http://jalammar.github.io/illustrated-transformer/
## SELF-ATTENTION

### Diagram

<table>
<thead>
<tr>
<th>Input</th>
<th>Thinking</th>
<th>Machines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding</td>
<td>$x_1$</td>
<td>$x_2$</td>
</tr>
<tr>
<td>Queries</td>
<td>$q_1$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>Keys</td>
<td>$k_1$</td>
<td>$k_2$</td>
</tr>
<tr>
<td>Values</td>
<td>$v_1$</td>
<td>$v_2$</td>
</tr>
<tr>
<td>Score</td>
<td>$q_1 \cdot k_1 = 112$</td>
<td>$q_1 \cdot k_2 = 96$</td>
</tr>
<tr>
<td>Divide by 8 ($\sqrt{d_k}$)</td>
<td>14</td>
<td>12</td>
</tr>
<tr>
<td>Softmax</td>
<td>0.88</td>
<td>0.12</td>
</tr>
</tbody>
</table>

---

From [http://jalammar.github.io/illustrated-transformer/](http://jalammar.github.io/illustrated-transformer/)
# SELF-ATTENTION

<table>
<thead>
<tr>
<th>Input</th>
<th>Thinking</th>
<th>Machines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding</td>
<td>$x_1$</td>
<td>$x_2$</td>
</tr>
<tr>
<td>Queries</td>
<td>$q_1$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>Keys</td>
<td>$k_1$</td>
<td>$k_2$</td>
</tr>
<tr>
<td>Values</td>
<td>$v_1$</td>
<td>$v_2$</td>
</tr>
<tr>
<td>Score</td>
<td>$q_1 \cdot k_1 = 112$</td>
<td>$q_1 \cdot k_2 = 96$</td>
</tr>
<tr>
<td>Divide by $8 \cdot \sqrt{d_k}$</td>
<td>14</td>
<td>12</td>
</tr>
<tr>
<td>Softmax</td>
<td>0.88</td>
<td>0.12</td>
</tr>
<tr>
<td>Softmax X Value</td>
<td>$v_1$</td>
<td>$v_2$</td>
</tr>
<tr>
<td>Sum</td>
<td>$z_1$</td>
<td>$z_2$</td>
</tr>
</tbody>
</table>

From [http://jalammar.github.io/illustrated-transformer/](http://jalammar.github.io/illustrated-transformer/)
SELF-ATTENTION

\[
\begin{align*}
X \times W^Q &= Q \\
X \times W^K &= K \\
X \times W^V &= V
\end{align*}
\]

\[
\text{softmax} \left( \frac{Q \times K^T}{\sqrt{d_k}} \right) = Z
\]

From http://jalammar.github.io/illustrated-transformer/
1. predict a masked word

80%: substitute input word with [MASK]

10%: substitute input word with random word

10%: no change

Like context2vec, but better suited for multi-layer self attention
1. classify two sentences as consecutive or not:

50% of training data (from OpenBooks) is "consecutive"

\[
\text{Input} = [\text{CLS}] \text{ the man [MASK] to the store [SEP]}
\]

\[
\text{penguin [MASK] are flight [SEP] less birds [SEP]}
\]

\[
\text{Label} = \text{NotNext}
\]

\[
\text{Input} = [\text{CLS}] \text{ the man went to [MASK] store [SEP]}
\]

\[
\text{he bought a gallon [MASK] milk [SEP]}
\]

\[
\text{Label} = \text{IsNext}
\]
USING BERT WITH PRE-TRAINING/FINETUNING

Use the pre-trained model as the first “layer” of the final model, then train on the desired task.
Using BERT with Pre-Training/Finetuning

Use the pre-trained model as the first "layer" of the final model, then train on the desired task.
Use the pre-trained model as the first "layer" of the final model, then train on the desired task.
USING BERT WITH PRE-TRAINING/FINETUNING

Use the pre-trained model as the first "layer" of the final model, then train on the desired task
Use the pre-trained model as the first “layer” of the final model, then train on the desired task.
USING BERT FOR REPRESENTATIONS

Use the pre-trained model to obtain contextualized word representations for the input

[visualization from The Illustrated BERT: https://jalammar.github.io/illustrated-bert/]
WHICH METHOD IS BETTER?
WHICH MODEL?

Not very extensive comparison...

Wieting et al. (2015) find that simple word averaging is more robust out-of-domain

Devlin et al. (2018) compare unidirectional and bi-directional transformer, but no comparison to LSTM like ELMo (for performance reasons?)
WHICH TRAINING OBJECTIVE?

Not very extensive comparison...

Zhang and Bowman (2018) control for training data, and find that bi-directional LM seems better than MT encoder

Devlin et al. (2018) find next-sentence prediction objective good compliment to LM objective
Not very extensive comparison...

Zhang and Bowman (2018) find that more data is probably better, but results preliminary.

Data with context is probably essential.
SOME RECENT IMPROVEMENTS
VARIOUS MONOLINGUAL BERTS

French: FlauBERT, CamemBERT

BERTje, ALBERTO, BETO, KoBERT, FinBERT, Bangla-BERT, German, Chinese, Russian, Japanese, etc

web-scale scraped corpora:
https://oscar-corpus.com/
MBERT

BERT trained on more than 100 languages

Really good starting point, but also issues for low-resource languages, e.g. over-segmentation
ROBERTA

Original BERT model was under-trained

Better trained, more data, and more robust model
XLM AND XLM-R

BERT problem: each sample in a single language

Combine MLM with Translation LM
Part-of-speech and Part-of-speech tagging