Carnegie Mellon University
Language
Technologies
Institute

11-411 Natural Language Processing

Course Project

David R. Mortensen

January 24, 2023

Language Technologies Institute

Learning Objectives

By the end of this lectures, students should be able to

- Describe what the course project is about and what the major milestones are.
- Locate relevant scholarly literature and assess whether it is relevant to the project.
- Describe three approaches to question generation and state advantages and disadvantages of each.

Overview of the Project

What is the Project about?

Build a Question/Answer System

- Given a Wikipedia article
    - Generate *n* “good” questions.
- Given a Wikipedia article
    - Answer *n* questions generated from that article.

What is a Good Question?

Pittsburgh is a city in the Commonwealth of Pennsylvania in the United States, and is the county seat of Allegheny County. The Combined Statistical Area (CSA) population of 2,659,937 is the largest in both the Ohio Valley and Appalachia, the second-largest in Pennsylvania after Philadelphia and the 20th-largest in the U.S. Located at the confluence of the Allegheny and Monongahela rivers, which form the Ohio River, Pittsburgh is known as both “the Steel City” for its more than 300 steel-related businesses, and as the “City of Bridges” for its 446 bridges. The city features 30 skyscrapers, two inclines, a pre-revolutionary fortification and the Point State Park at the confluence of the rivers. The city developed as a vital link of the Atlantic coast and Midwest. The mineral-rich Allegheny Mountains made the area coveted by the French and British empires, Virginia, Whiskey Rebels, and Civil War raiders.

- Is Pittsburgh a country?
- Where is Pittsburgh?
- What is the population of Pittsburgh?
- What is Pittsburgh’s nickname?
- How many steel-related businesses are there in Pittsburgh?

What is a Bad Question?

Pittsburgh is a city in the Commonwealth of Pennsylvania in the United States, and is the county seat of Allegheny County. The Combined Statistical Area (CSA) population of 2,659,937 is the largest in both the Ohio Valley and Appalachia, the second-largest in Pennsylvania after Philadelphia and the 20th-largest in the U.S. Located at the confluence of the Allegheny and Monongahela rivers, which form the Ohio River, Pittsburgh is known as both "the Steel City" for its more than 300 steel-related businesses, and as the "City of Bridges" for its 446 bridges. The city features 30 skyscrapers, two inclines, a pre-revolutionary fortification and the Point State Park at the confluence of the rivers. The city developed as a vital link of the Atlantic coast and Midwest. The mineral-rich Allegheny Mountains made the area coveted by the French and British empires, Virginia, Whiskey Rebels, and Civil War raiders.

- What is the capital of France?
- What is the 30th letter of this article?
- What are Pittsburgh's three rivers of?
- Who coveted Allegeny Mountains?

How to Find Questions?

A template-based approach (`baseline`):

- *X* is *Y* → What is *X*? (what/why/who/when)
- The *X* verbs *Y* → What does the *X* verb?
- Number-based questions
  - What is the *X* of *Y*?
  - How many *X*'s does *Y* verb?
    - How many skyscrapers does Pittsburgh feature?

What Makes a Good Answer?

Pittsburgh is a city in the Commonwealth of Pennsylvania in the United States, and is the county seat of Allegheny County.

What is Pittsburgh?

*****	a city
****	a city in the Commonwealth of Pennsylvania in the United States
***	city
**	a city in the Commonwealth of Pennsylvania in the United States, and is the county seat of Allegheny County.
*	the Commonwealth

How to Answer Questions?

A template-based approach (**`baseline`**$_1$):

- What is Y? Look for X is Y
- “Extractive” answers
  - Subset of the text
  - Maybe with small changes

How to Answer Questions?

An information retrieval approach (`baseline`$_2$):

- Classify the query (question) according to type (polar question [yes/no], wh-question [who/what/where/where/when/how], etc.)
- Represent the query as a vector
- Represent the passages in the document as vectors
- Find the passage with the greatest similarity to the query
- Process the passage (give the query and its class), yielding an answer

How to Answer Questions

An approach using a pretrained language model (*baseline*$_3$):

- Feed a model that has been TRAINED to perform language modeling and FINE-TUNED to identify answer spans input like the following:

  `[CLS] Question [SEP] Context [SEP]`

  Where

  - `[CLS]` marks the beginning of the input
  - `[SEP]` separates or terminates the different parts of the input (the question and the context)
- Get back TOKEN INDICES for the beginning and end of the answer

Homework 1: Baseline$_3$

- For your first homework assignment, you will implement a baseline QA system with **Hugging Face Transformers**
- Hugging Face is a company built around an AI community and open source tools and models
- From Hugging Face, you can download pre-trained models for many NLP applications
- The assignment might look scary
  - Lots of scaffolding
  - Lots of documentation
  - Enthusiastic TAs

Evaluation of Questions and Answers

- Human Evaluation of Questions
  - Fluency—is the question grammatically well-formed and idiomatic?
  - Reasonableness—can the question be answered by a human, given the document?
- Human Evaluation of Answers
  - Fluency—is the answer grammatically well-formed and idiomatic?
  - Reasonableness—is the answer correct, given the question and the document?
  - Conciseness—is the answer free of extraneous content?

Question Generation

- **Input** text of a Wikipedia article and an integer *n*
- **Output** *n* distinct questions about the article. They should be
  - fluent
  - “reasonable”
- `ask article.txt n`

Question Generation Evaluation

- We choose *n* (you don't know in advance)
- We will use your program to generate questions on
  - Some of the articles you had access to
  - Similar articles (same domains)
  - A completely different type of topic (still Wikipedia)
  - **All articles will consist of plain text with no HTML or wikitext markup**
- Each question will be evaluated
  - How fluent? (40%)
  - How reasonable? (40%)
  - How difficult to answer? (20%)
- **Systems will be tested on an equal number of wh-questions ("factoid question") and polar questions ("yes/no-questions").**

Question Answering

- **Input** text of a Wikipedia article and list of questions about the article
- **Output** The answers to the questions They should be
  - fluent (30%)
  - reasonable ("intelligent" or "human-like" in their accuracy) (50%)
  - concise (20%)
- `answer article.txt questionstxt n`

Question Answering Evaluation

- We will feed your system questions based on the output of the question generation systems
    - High fluency
    - High reasonableness
    - 30% easy, 40% medium, 30% hard
- Each answer will be evaluated
    - How fluent? (30%)
    - How correct? (50%)
    - How concise? (20%)

Teams

- Students are to organize themselves into groups of 4 (with up to 3 groups of 5 allowed only where mathematically necessary)
- All members in the group are expected to contribute to the project **equally**
- Individual groups decide who does what in the execution of the project
- Everyone on your team shares a grade out of 52 but the instructors reserves the right to adjust this based on peer survey results

Important Check-Point Dates

COMPONENT	POINTS	PERCENTAGE	DUE DATE
Literature Search	2	3.85%	Jan 26
Proposal	4	7.69%	Feb 07
Progress Report	7	13.46%	Feb 21
QA Prototype	8	15.38%	Mar 21
QG Prototype	8	15.38%	Mar 30
Final System	11	21.15%	Apr 18
Final Report	12	23.08%	Apr 28
Project Total	**52**	**100.00%**	

The Project Package

The project handout includes a directory `data` which includes articles from four Wikipedia topics (four sets). Each article is a `txt` file. The first homework assignment will produce annotations for all of these articles.

Literature Search

You are to identify and describe (in one paragraph each) 4 articles or papers relevant to your project. Two papers must be about question generation and two must be about question answering. Places to look:

- ACL Anthology (`https://aclanthology.org/`). The most important resource for finding articles relating to natural language processing and computational linguistics
- Semantic Scholar (`https://www.semanticscholar.org/`). A useful, high quality, search tool for scholarly literature. Less noisy than Google Scholar (`https://scholar.google.com/`).
- arXiv (`https://arxiv.org/`). Indispensable, but also full of garbage. *Caveat lector*!

Initial Plan (2 pages max)

Your initial report needs to contain a discussion of how you are going to approach the project at a high level and how you are going to divide your effort up. Some details you can address are

- Is there going to be a relationship between the asking and answering components of your system or are you going to implement them independently?
- What tools are you going to use?
- How are you going to share code data inside your team?
- How are you going to coordinate development inside the team?
- What technical approaches are you going to take? **Be sure to provide two system diagrams**, one for QG and one for QA.

Progress Report Video (3 minutes max)

- What is your concrete plan now? You must provide a system diagram for both systems.
- What have you achieved so far? You must have started coding at this point.
- Any fundamental issues and how you want to address them?

2 page written supplement is allowed but not required.

Working QA System and QG System

- By **Mar 21**, you must submit a prototype question answering system as a Docker file (your `answer` program)
- By **Mar 30**, you must submit a prototype question generation system as a Docker file (your `ask` program)
- By **Apr 18**, you must submit a Docker file that can build your final system (`ask` and `answer` programs)

Final Report (4 pages)

By **Apr 28**, you must hand in a polished scientific paper describing and discussing your methodology and results.

- a high-level discussion and analysis of your approach to solving the problem, your system architecture, etc. and that showcases the novelty and technical soundness of your approach (e.g., your experiments)
- **description of the experiments you carried out to evaluate your programs and the results**
- sufficient detail for someone else who took this class to approximately replicate your system

We want to see that you have used what you have learned in class to build your system. Your paper must correctly cite prior work and any tools you used. The paper should be formatted using the ACL template and should take the form of an ACL short paper.

Experiments

- Your paper must report experiments and their results
- These experiments must compare the performance (in one or more dimensions, including the evaluation criteria) of your systems to baselines (one or more fof QA, one or more for QG)
- The baselines must be reasonable (taken from published papers or preprints and representative of systems in use)

On Pretrained Models

- You are allowed to use pretrained models in your project (QA and QG).
- **You are not allowed to make API calls to external servers.**
- If you use a pretrained model for QA or QG, you must find the original paper employing that model for the task and reimplement their approach[1] and use it as a baseline in your experiments
- You must show that your system is better is some respect than the previously existing system (performance on a dataset, efficiency, interpretability, etc.)

[1] If they make their code available, you may use it.

Question Generation

Question Generation as a Task

- **Input:** a document
- **Output:** a series of questions that can be answered by a human, given the document

Three Approaches to Question Generation

There are many approaches to generation. Here are three basic ones:

- Template-based matching and generation
- Syntactic transformations
- Encoder-decoder question generation

To understand these approaches, take the statement “Andrew Carnegie founded the Carnegie Institute of Technology.”

Template-Based Matching and Generation

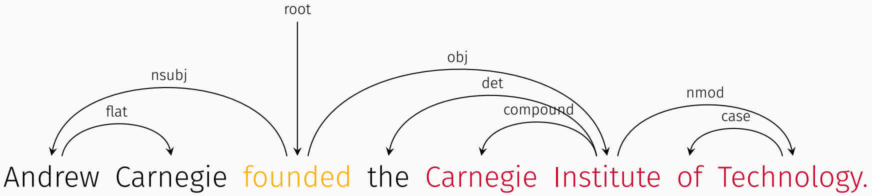
Sentence, with part of speech and named entity tags (Module 5):

Andrew	Carnegie	founded	the	Carnegie	Institute	of	Technology.
PN	PN	V	Det	PN	PN	P	PN
PER				ORG			

Template: if you match `PER V x`, return `who V x` where `x` is a variable matching any number of TOKENS (words).

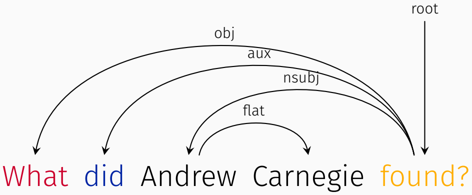
Syntactic Transformations

A dependency parse (Module 7) of the sentence:



Andrew Carnegie founded the Carnegie Institute of Technology.

can be deterministically transformed into a question:



What did Andrew Carnegie found?

Encoder-Decoder Question Generation

A neural model with an encoder-decoder architecture can also be trained to generate questions from passages.

- **Option 1:** Train on passage-question pairs
- **Option 2:** Fine-tune a pretrained model on passage-question pairs

Requires a considerable volume of training or development data but does not rely on hand-crafted templates or transformations.

Questions?