CS-5340/6340 Project Description, Fall 2021

The project for this course will be to design and build an information extraction (IE) system for the domain of corporate acquisition events.[1] You can work in a 2-person team or as a solo team, depending on your preference. Your team's program should read short news stories about corporate acquisition events and extract several pieces of information. A sample news story is shown below:

Text 379

> Four Seasons Hotels said it and VMS Realty Partners of Chicago have agreed to purchase the Santa Barbara Biltmore Hotel from Marriott Corp for an undisclosed amount.
> It said the venture will rename the hotel the Four Seasons Biltmore at Santa Barbara and invest over 13 mln dlrs in improvements on the 228-room property.
> Reuter

Each news story will describe exactly one relevant corporate event. For each story, your IE system will have to fill out an *event template* with the 8 "slots" defined below. The TEXT field will contain the story's identifier and the other 7 slots should contain information automatically extracted from the story. Most stories will not contain all of this information, in which case only a subset of these slots should be filled. Some stories may have only a couple slots filled, while others may have most or even all of these slots filled.

TEXT: filename identifier
ACQUIRED: entities that were acquired
ACQBUS: the business focus of the acquired entities
ACQLOC: the location of the acquired entities
DLRAMT: the amount paid for the acquired entities
PURCHASER: entities that purchased the acquired entities
SELLER: entities that sold the acquired entities
STATUS: status description of the acquisition event

You will be provided with a collection of corporate acquisition news stories as well as "gold" *answer key templates*, which you can use to help design and develop your system.[2] The output of your system will be scored against these answer keys.

[1] The data set may also include some related events, such as corporate mergers. But we will refer to the domain as "corporate acquisitions" for the sake of simplicity.

[2] The gold answer key templates were produced by another research group many years ago. The "correct" answers were manually identified by human annotators. Data created by people is never perfect and you may disagree with a few answers, but we believe that it is a high quality data set overall and will not make any changes to it to maintain the stability of the data.

Template formatting is as follows:

- Every template MUST have at least 8 rows corresponding to the 8 slots defined earlier. *The slots must be printed in exactly the order shown.*
- If no information can (or should) be extracted for a slot, then you should print three dashes (- - -) to indicate that the slot is empty.
- Every non-empty answer should be printed in double quotes (e.g., “this is an acceptable answer string”).
- Every slot except the TEXT field can potentially have more than one answer. Each distinct answer should be printed on a separate line with the same slot field name. For example, if someone acquires 3 companies, then your output template should contain 3 rows for the slot type ACQUIRED. The order in which you print these answers does not matter (but you should print them in adjacent rows).
- The answer key templates will sometimes contain a disjunction of acceptable answers for a slot, with the disjuncts separated by a slash. For example, an answer key might list “IBM” / “IBM Corp”, indicating that the strings “IBM” and “IBM Corp” are both acceptable answers. **IMPORTANT: Your system should not use slashes in its output templates! Each slot should be filled by a single extracted string in your output.**

As an example, the answer key template for Text 379 is shown below:

Answer Key Template

TEXT: 379
ACQUIRED: “Biltmore Hotel” / “Santa Barbara Biltmore Hotel”
ACQBUS: - - -
ACQLOC: “Santa Barbara”
DLRAMT: “undisclosed amount”
PURCHASER: “Four Seasons Hotels”
PURCHASER: “VMS Realty Partners”
SELLER: “Marriott Corp”
STATUS: “agreed to purchase”

This answer key template indicates that the Biltmore Hotel should be extracted as the ACQUIRED entity, and that the strings “Biltmore Hotel” and “Santa Barbara Biltmore Hotel” are both acceptable answers. So if your system extracts either one of those strings, its output will be scored as correct.

The ACQBUS slot is empty in the answer key template, indicating that nothing should be extracted for that slot.

The ACQLOC, DLRAMT, SELLER, and STATUS slots each have a single answer that should be extracted. Note that the DLRAMT slot should be filled by a phrase, rather than a monetary amount, in this particular story.

The story mentions that two entities purchased the Biltmore Hotel, so both entities need to be extracted as PURCHASERs. Each one is printed on a separate line, with the same PURCHASER: slot field name. The order they are listed does not matter.

Input

Your IE system should accept a single input file as a command-line argument, which will list the texts to be processed. We should be able to run your program like this if you use python:

```
python3 extract.py <doclist>
```

If you use Java, you should invoke Java similarly and be sure to accept the same argument on the command-line.

The doclist file will contain a list of full pathnames for the files to be processed, one per line. For example, a doclist file might look like this:

```
/home/kermit/docsA/10
/home/kermit/docsA/XYZ
/home/kermit/docsB/story.txt
```

Each pathname should be split into a path (everything up to and including the rightmost slash /) and the filename itself (the string after the rightmost slash). For example, the pathnames in the doclist above should be split into:

```
PATH = /home/kermit/docsA/   FILENAME = 10
PATH = /home/kermit/docsA/   FILENAME = XYZ
PATH = /home/kermit/docsB/   FILENAME = story.txt
```

The filename should be treated as a text's identifier when creating its output template. You can assume that the filenames will be unique.

Output

As output, your IE system should produce a set of output templates, one template per story. Your system should always produce a template for a story and the TEXT: slot should always be filled.

Your system should print the output templates to a single file that has the same name as the input file but with an added extension of ".templates". For example, if the input file is called "doclistX" then the output file should be named "doclistX.templates". **The output templates should correspond to the stories in the doclist input file in exactly the same order.** Print a blank line between the templates for different stories.

For example, if the input file lists three stories with the identifiers 10, 22, and 43 (listed in that order), then the output file should contain exactly three templates, where the first one corresponds to text 10, the second one corresponds to text 22, and the third one corresponds to text 43.

The Data Sets

You will be given three sets of data at different points in the project.

Development Set: approximately 400 stories and answer keys

Test Set #1: 100 stories and answer keys

Test Set #2: 100 stories and answer keys

Project Phases

The project will involve three phases:

Development Phase: A **Development Set** is available on CANVAS for you to use when creating your IE system. You may use these stories and the answer keys in any way that you wish.

In addition, we will give you the scoring program that we will use to evaluate your IE system. You can use this scoring program to assess the performance of your system yourself as you experiment with different ideas. The arguments that it takes are described at the beginning of the file.

Midpoint Evaluation: There will be a midpoint evaluation of everyone's IE systems. Each team will submit the source code for their IE system and we will evaluate each system on a new data set called **Test Set #1**. The purpose of this evaluation is to make sure that every team is making progress on creating a IE system and to allow everyone to see how other teams are performing at the (roughly) halfway point.

Once the midpoint evaluation is over, we will release Test Set #1 so that you can improve the performance of your system on those stories.

Final Evaluation: For the final evaluation, each team will submit the source code for their IE system. We will run your IE system on a new data set called **Test Set #2**.

Evaluation

The performance of each IE system will be evaluated using the **F-measure** statistic, which combines *recall* and *precision* in a single metric.

Recall (R): the number of correct items extracted by your system divided by the number of items in the answer template. An extracted string is correct only if it exactly matches one of the strings in the answer template.

Precision (P): the number of correct items extracted by your system divided by the total number of items extracted by your system.

F-measure: $\mathbf{F(R, P)} = \frac{2 \times P \times R}{P + R}$

This formula tries to find a good balance between recall and precision. (It is the harmonic mean of recall and precision.) *The final performance of each system will be judged based on its F-measure score.*

The scoring program that we will use to evaluate your IE systems is available on CANVAS, so you will know exactly how we will be computing the scores. We encourage you to use it during system development as well, to track the performance of your system as you make changes and try to improve it.

IMPORTANT: Robustness will also matter during the project evaluations! You should make NO assumptions about the formatting of the stories that your system must process. The news stories that you will be given are from a real-world text collection put together by other researchers. We do not know how they were originally harvested, and you may notice that some of them look a bit strange formatting-wise, probably because meta-information was automatically removed. This is a valuable lesson for real-world text analysis – documents can be sloppy and unpredictably formatted.

Please try to make your systems as robust as possible so that they will not crash when processing stories! Points will be deducted for systems that crash during the midpoint or final evaluations. Also, your system will not be able to extract information from a story if it crashes, so your recall score will suffer as well. Please take some time to run many documents through your system to test its robustness! You should be able to run it on ANY news article, so you can even try downloading some yourselves and see how your system works.

Schedule

October 20: Fill in the Team Request Form on Canvas.

November 10: Midpoint evaluation systems due.

November 30: Final evaluation systems due.

December 6, 8: Project presentations (during class).

December 10: Final project slides and posters due.

Grading

Each project will be graded according to the following criteria:

- 35% of the grade will be based on the performance of your IE system on Test Set #1 during the midpoint evaluation.
- 60% of the grade will be based on the performance of your IE system on Test Set #2 during the final evaluation.

- 5% of the grade will be based on your project presentation, which will be an in-class presentation for the top-performing teams or a poster presentation for the remaining teams. All teams will also be required to submit their slides or posters for grading.

To determine the grades for the midpoint and final evaluations, teams will be ranked based on the performance of their system relative to the other IE systems. The teams will then be clustered (manually) so that teams whose systems produced similar scores will get similar grades. It is fine to share ideas with other teams (but not code!), and to compare your system's performance with other teams. But if your team is doing almost exactly the same thing as many other teams, chances are your system will end up in the middle of the rankings. To distinguish yourself and stand apart in the rankings, we encourage teams to try different things!

IMPORTANT: I will also ask each team to document the specific contributions of each team member to the team's final IE system. This will allow me to adjust individual grades in case some people work a lot harder than others. If all teammates contribute roughly equally to the project (as I expect in most cases), then they will all get the same project grade. But if one teammate contributes very little to the project, then they will get a lower grade than the person who put in substantially more time and effort.

The final grading will be based on how well your system does relative to other team's systems, but it is <u>not</u> the case that the highest ranked system will get an automatic 'A' or that the lowest ranked system will get an automatic 'E'. If every team produces a system that works well, then I will be happy to give everyone an 'A' on the project! If, at the other extreme, no one generates a system that works at all, then I would have to give every team a failing grade. I hope that the competitive spirit will energize everyone to work hard and produce interesting and effective IE systems so that I can give many teams a high grade!

External Software & Data

You may use external software packages and data resources for your project, as long as the following criteria are met:

- **You may NOT use any external software that performs event extraction or a subtask that is specific to this domain or problem!** If we discover that your submitted system uses any external system or code that violates this condition, then you will be disqualified and get a zero for the project.

- You must fully acknowledge in your final presentations ALL of the external software and data resources that you used in your system.

- We must be able to run all of your software (your own and external resources) on the linux-based CADE machines. This means either including the software in your code submission, or installing it in your own CADE directory and giving us full permission to access it from there. Feel free to discuss options with the TAs.

- You MAY use external NLP software that performs general-purpose NLP functionality, including tokenizers, sentence splitters, part-of-speech taggers, syntactic parsers, general-purpose named entity recognizers, coreference resolvers, and general semantic dictionaries such as WordNet. **If you are uncertain about whether a specific resource is acceptable to use, please ask the instructor!**

Machine Learning

You do not need to use machine learning (ML) for this project. But you are welcome to do so if you wish. If you choose to use ML:

- You MAY use general-purpose external ML software packages, such as scikit-learn.
- You may NOT use any ML models that have been previously trained for event extraction or a subtask that is specific to this domain or problem!
- You MAY use the Development Set as training data for both the Midpoint and Final Evaluations. You may use Test Set #1 as additional training data for the Final Evaluation.
- If you create your own ML model for event extraction you must train it yourself using ONLY the cs5340/6340 data provided on Canvas. **You may NOT use any additional sources of training data.** If you submit an ML model that has been trained with external data, your IE system will be disqualified and you will get a zero for the project. The reason is to level the playing field for all teams in the class, so that everyone is using exactly the same data set to build their IE systems.
- You can use pre-trained general-purpose embedding vectors (such as word2vec or GloVe) and pre-trained language models (such as BERT) if you wish. “General-purpose” means that it was trained using a broad coverage text collection that is not specific to any domain. If, however, you were somehow to find a model that had been pre-trained specifically on news articles about corporate events, that would be prohibited.

CAVEAT AND ENCOURAGEMENT

Building an effective event extraction system is hard! Information extraction is not a solved problem in NLP, so you shouldn’t expect your IE systems to produce super-high scores! Just try to design a IE system as best you can to perform reasonably well on the cs5340/6340 data. I encourage everyone to have fun with this project and experiment with lots of ideas. I hope to see many different types of systems and approaches.

This project will give you exposure to a cutting edge research area, understanding of an important application area for NLP, the experience of building a real NLP system, and the opportunity to explore your creative side!