AnHai's Group

Search this site

Home
People
Research
Publications

Current Projects
Data Cleaning & Integration
Data Science
Magellan
py_entitymatching
py_stringsimjoin
py_stringmatching
activelearn
py_labeler

Useful Stuff
Data Repository
How to Create Packages

Past Projects
Knowledge Bases/Graphs
Crowdsourcing
Schema/Ontology Matching
Silicon Valley Time Off

Courses
CS 638 DS
Fall 2016
CS 838 DS
Spring 2018
CS 564

Outreach & Funding
Walmart Labs
NSF IIS-Medium 2016

Miscellaneous
Wisconsin DB Group

Courses > CS 838 Spring 2018 (Also Known as CS 839) > Project Description for CS 839 Spring 2018 >

CS 839 Spring 2018, Project Stage 1

Due date to be decided.

In this project stage your team will perform information extraction (IE) from natural text documents, using a supervised learning approach. Here are the steps to follow:

- You must collect at least 300 text documents from which you will extract mentions of ONE entity type (e.g., person names, locations, organizations, etc.). These documents must contain well-formed sentences (such as those in news articles).
- In fact, collect a bit more than 300, say 330 or so, because during the project stage you will do some cleaning and may decide to remove some documents. So you need a bit more than 300 to ensure you still have 300+ for the various steps of this project stage. These documents must contain only plain text in English.
- Decide on ONE entity type (e.g., person names, locations, organizations, etc.) that you will extract from these documents. NOTE: you will extract mentions of an entity type (e.g., some kind of noun). You are not supposed to extract adjectives (e.g., great, happy, excellent) nor other stuff such as color, size, weight, etc.
- Now go through the documents and mark up the mentions of this entity type. For simplicity, let's assume that you are extracting person names. Then mark up all person names in the documents. Once you have marked up the mentions, you should have at least 800 mentions (over 300 documents). If not, discard documents with too few mentions and get some more documents. Try to avoid skew (such as a few documents contain most of the mentions).
- Let this set of documents be B. Split it into a set I of 200 documents and a set J of the remaining 100 documents (after you have randomized the order of the documents). The set I will be used for development (dev set), and the set J will be used for reporting the accuracy of your extractor (the test set).
- Your goal is to develop an extractor that achieves at least precision of 90% or higher and as high recall as possible, but at least 60% in recall.

- **The learning step:** you will start out by building the best possible learning-based extractor:
    - Perform cross validation (CV) on the set I to select the best classifier. You must consider at least the following classifiers: decision tree, random forest, support vector machine, linear regression, and logistic regression. You must use the scikit-learn package for this CV purpose.
    - Let the best classifier found as above be M. Then debug M using the same set I (in the class we have discussed or will discuss how to do this: you need to split I into two sets P and Q, train M on P, apply M to label examples in Q, then identify and debug the false positive/negative examples; if you try to improve recall then pay attention to the false negative examples).
    - Once you have debugged M, you need to redo CV to see if another classifier may happen to be more accurate this time. Then you may need to debug that classifier. And so on. (I will discuss how to do this in detail in the class and will provide class note.)
    - Finally, suppose you have not been able to do anything else to improve the classifier accuracy. Let the resulting classifier be X. You can apply X to the set-aside test set J to see if its accuracy already meets the requirement (90% P and 60% R). If yes, you can stop here. Otherwise, you should now try adding rules.
- **The rule-based postprocessing step:** at this point you will try to add rules in a post-processing step in order to further improve the accuracy of the classifier X.
    - You will again use the set I for this purpose. Again, split I into P and Q. Train X on P and apply X to Q. Again, analyze the false pos/neg examples in Q and see what rules you can add to fix those.
    - Do this until you feel the overall accuracy of X is already meeting the requirement, or you are out of ideas to try, or out of effort.
- Let Y be the final combination of X and the postprocessing rules. Then you apply Y to the set-aside test set J and report Y's accuracy on that. You MUST NOT go back and try to debug again on I, in order to further improve accuracy on J (this can lead to overfitting on J, we discussed this in the class).
- Note: Whenever you want to compute the accuracy of a classifier X on the dev set I, you can perform a CV on I to compute the accuracy. This accuracy will overfit a bit the set I.

What to Submit?

IMPORTANT: Please read and follow the submission instruction below carefully. Points will be taken off for failure to follow instructions.

On your team's project page, by the deadline:

- provide link to a directory that stores the 300+ documents. Do not zip all the documents into a single file. Instead, we should be able to browse the documents in that directory, i.e., each document must be in a file, and that file should be browsable.
    - Within the above directory, provide a README file that tells us the entity type that you have marked up in the documents, and give us information on where to find these markups. For example, if you have marked up person names, you can say something like: person names are marked up in the documents using <person>...</> tags.
- provide link to a browsable directory that stores all documents in set I.
- provide link to a browsable directory that stores all documents in set J.
- provide link to a directory that stores all of your code (this directory must also be browsable).
- provide link to a compressed file that stores all of the above directories.
- **provide link to a pdf document that contains at least the following:**
    - the names of all team members.
    - the entity type that you have decided to extract, give a few examples of mentions of this entity type.
    - the total number of mentions that you have marked up.
    - the number of documents in set I, the number of mentions in set I.
    - the number of documents in set J, the number of mentions in set J.
    - the type of the classifier that you selected after performing cross validation on set I *the first time*, and the precision, recall, F1 of this classifier (on set I). This classifier is referred to as classifier M in the description above.
    - the type of the classifier that you have finally settled on *before* the rule-based postprocessing step, and the precision, recall, F1 of this classifier (on set J). This classifier is referred to as classifier X in the description above.
    - if you have done any rule-based post-processing, then give examples of rules that you have used, and describe where can we find all the rules (e.g., is it in the code directory somewhere?).
    - Report the precision, recall, F1 of classifier Y (see description above) on set J. This is the final classifier

(plus rule-based post-processing if you have done any).

- If you have not reached precision of at least 90% and recall of at least 60%, provide a discussion on why, and what else can you possibly do to improve the accuracy.
- Provide any other information that you would like.

Misc Information

Check out MadDSI for tools that can help you. Please email us if you find some tools that can help and are not listed here.

Comments

You do not have permission to add comments.

Sign in | Recent Site Activity | Report Abuse | Print Page | Powered By Google Sites