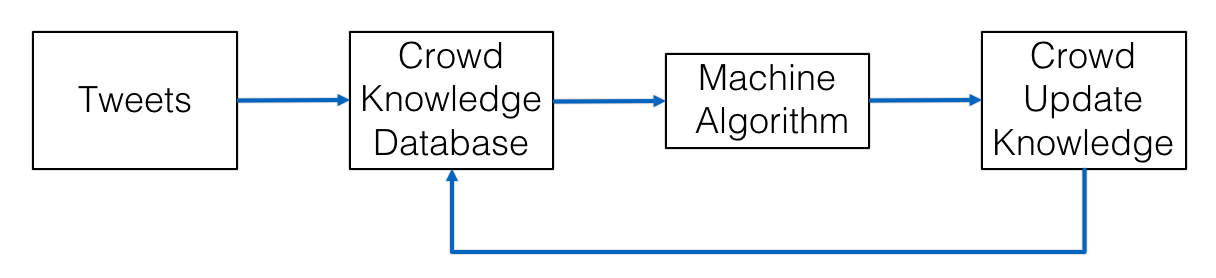
Design

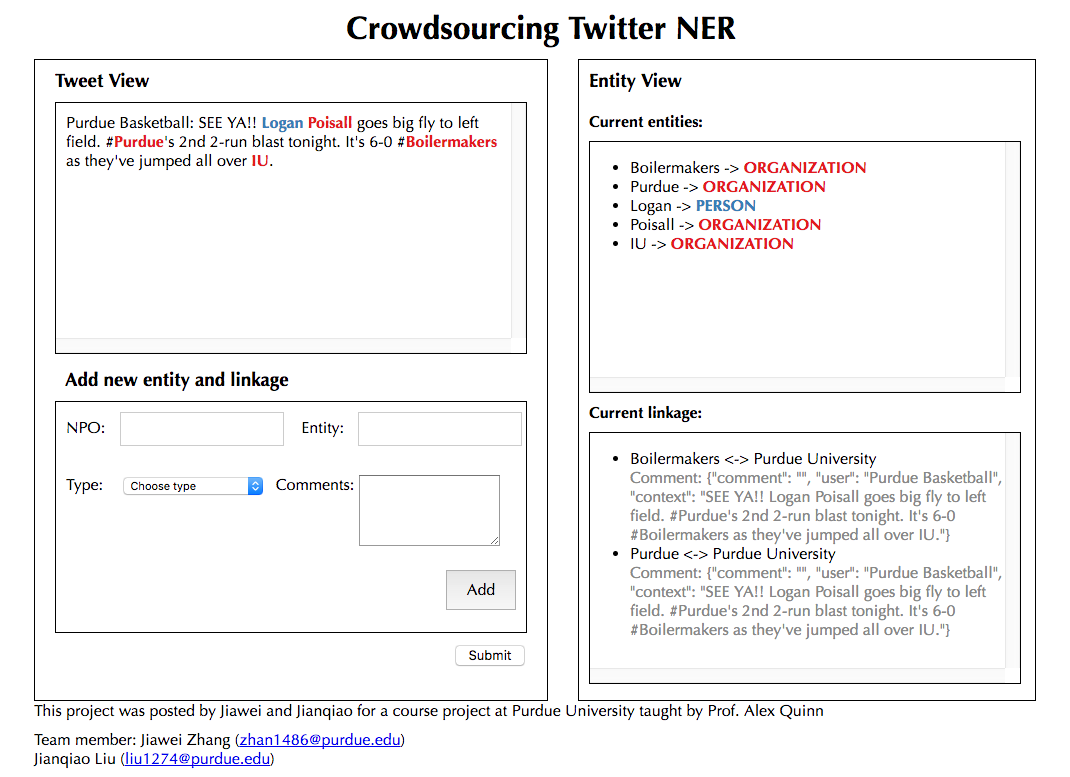
To address the obscure words problem, we expect a perfect natural language processing (NLP) algorithm that can automatically partition the sentences, extract the tokens, analyze the context and tag the tokens with necessary descriptions. However, in practice, we find that event the state-of-the-art natural language processing algorithms are far away from satisfactory. On one hand, NLP algorithms perform well at “word” level extraction, but fail at understanding and capture words (combinations) that people use for convention or amusement. For example, the NLP would treat the “Hoosiers” as an organization name in the tweet “Indiana fans celebrating the Hoosiers' most recent national championship”. Actually here the “Hoosiers” just means the Indiana state. On the other hand, there are many words created in our culture. Even the normal words we used for a long history may be given new meanings. For example, “The Unburnt Queen of the Andals, Queen of Meereen, Khaleesi of the Great Grass Sea, Breaker of Chains, Mother of Dragons.” It looks like all words here are just normal use, but people who haven’t read/watched the Game of the Throne can hardly understand these words’ real meaning. All these five phrases are the title of Daenerys Targaryen, a key role in the novel/movie.

Based on our discovery, we design a crowd sourcing approach to effectively recognize entities in tweets. Our system can not only help people with no background knowledge understand the context more easily, but also help machine learning algorithms learn more effectively. We design our NER is three steps:

1. Query the user tweets with our crowd knowledge database, and mark all the entities that have already been recognized
2. Use standard NLTK (Natural Language Toolkit 3.0) to process the tweets. In this process, we ignore the entities marked in the first. And due to the time limitation, we only consider three categories: the person, the organization and the location.
3. Ask the crowd sourcing to correct the recognition results from previous steps and add entities that have not been recognized. The crowd modified result will be updated into knowledge database.



Implementation



Our target is to recognize obscure words in tweets as meaningful entities and link them to some other entities that are easier to understand/track. We organize the whole task with two separate database tables: the NPO table and ETY table.

NPO table -- Unique, real record for each natural person/organization/location/others, just like one's legal name. E.g. "Donald Trump". Each record in NPO would store the name, class, description and all records in ETY that are sourced from it. We call each record in NPO table a "SRC".

ETY table -- Any entity detected by machine/crowd from tweets, like one's nickname. Usually the mapping from record in ETY to record in NPO is a multiple-to-one mapping. E.g. "45", "DT" and "Orange Julius" could all exist in ETY table, and their "source" point to "Donald Trump" in NPO table. The "Donald Trump" may and may not exist in ETY (depends whether it appear in tweet). We call each record in ETY table a "DES".

When the crowd knowledge database detects an entity, the back-end will query the ETY table, and send the context and comment when the link is built to the front end.

When the crowd find a new entity, or correct an existing entity from tweets, they need to specify a NPO name for the entity. If the NPO name could be searched from our NPO table, we will create a new DES in ETY table, and add it into the “destination” of that NPO SRC. Otherwise, we need to create both a DES in ETY table and a SRC in NPO table and link them together. In both conditions, we prefer the workers to add some comments about why they link them and will show the comments when the link is searched in the future.

We strongly recommend the worker to use the most widely known name as SRC's name. The SRC's name is the unique keyword we define a person/place/object in our system. For current step, we can not avoid the duplication in NPO by machine, but we may potentially use crowd source to do this in the future. The description of SRC could be a link to a Wikipedia webpage, or a piece of words written by the user.