Interim Report

Introduction to NLP

Team Cheese Maggi:

Palash Sharma (2019101082),

Aryan Jain (2019101056),

Arth Raj (2019101094)

---

1. Work Done:

A. Problem Statement Understanding:

- The first step of any research project is to understand to the fullest, the core of the problem we set out to solve.
- This was achieved by diving deep into the field of dataset annotation and the various difficulties involved in manual annotation and the accuracies associated to it.
- This also formed the preliminary literature review.

B. Literature Review:

- We went through several research papers to understand properly how the NLP world has progressed on the problems of annotation. Some of the papers we referred are:

- **Named Entity Recognition without Labelled Data: A Weak Supervision Approach:** This presents a simple but powerful approach to learn NER models in the absence of labelled data through weak supervision. The approach relies on a broad spectrum of labelling functions to automatically annotate texts from the target domain.

- **Snorkel: rapid training data creation with weak supervision:** They

present a flexible interface layer for writing labeling functions based on their experience over the past years collaborating with companies, agencies, and research labs.

C. Named Entity Recognition Model using Snorkel:

- Building on the understand develop from the previous steps, we went on to implement NER models using snorkel in an iterative manner:
- Build a labelling function to tag a specific kind of entity
- Analyse the performance of the labelling functions together
- Improve the logic for labelling used in the labelling functions
- Go to first step until results are satisfactory

2. Comparison of Work done against timeline:

1. **Dataset understanding and topic research**: We read through several research papers and online links to have a deep understanding of the topic of what we are building and why we are building. The understanding is summarized below:
    - There used to always remains a problem of manual data labelling in order to use the data for train and testing purposes. The datasets can be huge and the task of manual labelling can be cumbersome often or not even practical sometimes.
    - To get over this problem, weak supervision systems are built which can programmatically label millions of data points using heuristics, rules-of-thumb, existing databases, ontologies, etc. at once. Snorkeling is one such technique.
    - Snorkel is a system that facilitates the process of building and managing training datasets without manual labelling. The first component of a Snorkel pipeline includes labelling functions, which are designed to be weak heuristic functions that predict a label given unlabeled data.
    - Now, the data quality needs to be assessed in order to make it to work under a particular domain. Named Entity Recognition (NER) performance often degrades rapidly when applied to target domains that differ from the texts observed during training. When in-domain labelled data is available,

transfer learning techniques can be used to adapt existing NER models to the target domain.

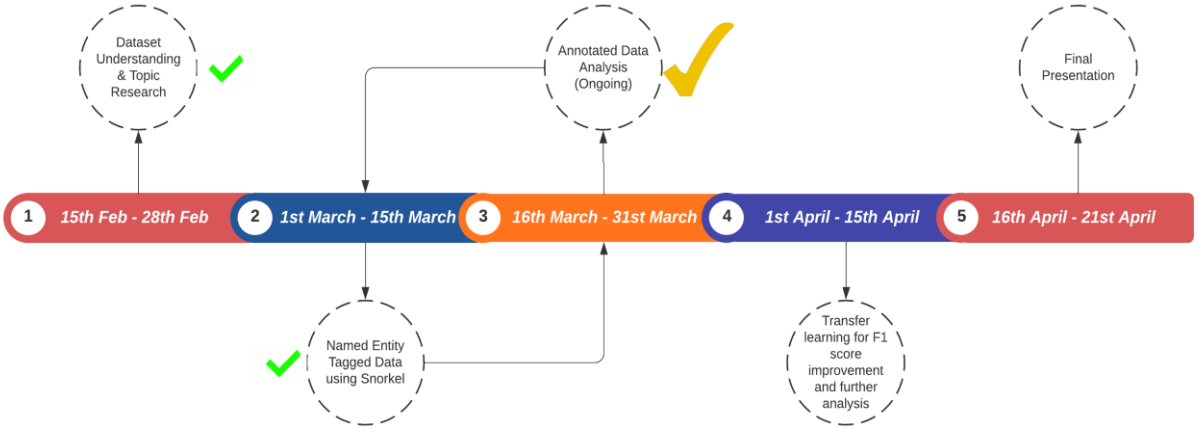
- So, Transfer Learning will be done using bootstrapping for F1 improvement.

2. **Named Entity tagged data using Snorkel**: Wrote labelling functions (Snorkel heuristics) of the following categories:

    - Hard-coded heuristics: usually regular expressions (regexes)
    - Syntactics: for instance, Spacy's dependency trees
    - Distant supervision: external knowledge bases
    - Noisy manual labels: crowdsourcing
    - External models: other models with useful signals

    After writing LFs (Labelling Functions), Snorkel will train a *Label Model* that takes advantage of conflicts between all LFs to estimate their accuracy. By looking at how often the labeling functions agree or disagree with one another, we learn estimated accuracies for each supervision source (e.g., an LF that all the other LFs tend to agree with will have a high learned accuracy, whereas an LF that seems to be disagreeing with all the others whenever they vote on the same example will have a low learned accuracy). And by combining the votes of all the labeling functions (weighted by their estimated accuracies), we're able to assign each example a fuzzy "noise-aware" label (between 0 and 1) instead of a hard label (either 0 or 1). Then, when labeling a new data point, each LF will cast a vote: positive, negative, or abstain. Based on those votes and the LF accuracy estimates, the Label Model was programmatically able to assign probabilistic labels to millions of data points. Finally, the goal is to train a classifier that can generalize beyond our LFs.

3. Work Plan:



So, work is almost done is at par with the timeline.