

NLP: Weakly Supervised Learning

AMIT DHOMNE

1 Weakly-Supervised Learning

- In the early days of NLP, all systems used hand-crafted, rule-based models built by experts.
- In the 1990s, the field shifted to statistical methods where models were learned from annotated data.
- Once researchers got good at supervised learning, they tried unsupervised learning (learning from no annotation).
- Unsupervised learning generally produces very poor results.
- But it's usually possible to be *some kind* of annotation.
- A better way forward is to assume that you are able to obtain at least some tiny amount of possibly low-quality data, and to try to learn from that.

2 Low-Resource POS Tagging

- Type (tag dictionary) vs Token (full sentence) annotation
 - Type annotation is fast, but annotations are out of context
 - Token annotation is slower, but gives sequence information as context
- Garrette and Baldridge (2013) found that *type* annotations are better.
 - Sequence context information can be recovered with semi-supervised learning algorithms.
 - Getting *some* (even out-of-context) information on more high-frequency words is better.
- However, there are lots of ways that the token annotation gathering could have been improved
 - Randomly order the sentence before giving them to the annotator. Adjacent sentences tend to be on the same topics, meaning that they will share vocabulary. Shuffling sentences gives a wider array of words.
 - Strategically order the sentences to maximize word coverage.
 - Actively choose the next best sentence *during annotation*

3 Active Learning

- Instead of collecting all annotations before training, iteratively choose the next item to be annotated based on the model's confidence given the annotations thus far.
- Useful when you need to get the best performance possible with only a limited amount of time for annotation; make the most of your time!
- A basic active learning pattern:
 1. Assume a corpus of annotatable items
 2. Choose an initial N items for annotation
 3. Train a model on everything annotated so far
 4. Run the model on all not-yet-annotated items
 5. Select the N lowest confidence item for annotation
 6. Repeat from Step 3 until you run out of time
- One simple way of judging confidence with, for example, a classifier is to use the model to compute the likelihood of each possible class. If some class has a much higher probability than the others, then the model has high confidence for that item. If, instead, no class has particularly high probability, then the model has low confidence.
- Annotating low confidence items is better because they are likely to be the most informative. If an item is high confidence, then the model is already sure of the answer and, therefore, won't learn anything from having that item annotated.
 - Though you'd probably still want to take other things into consideration like overlap with other items. If your system has low confidence because the item has a lot of totally unique words, then annotating it probably won't inform other items very much.
- The goal of active learning is to shift the learning curve to the left. That is, we want to reach the target accuracy with less annotation time than we would otherwise.
- Decisions the active learner can make:
 - Which item to have annotated?
 - What kind of annotation (eg, type vs token)? Some kinds of annotation are more costly than others; don't do a costly annotation if a cheaper one will suffice.
 - Who should annotate it? Different annotators have different skills. For example, don't waste an expert annotator's time on a simple annotation that a non-expert could do.
- Active learning experiments have typically been bad
 - Most experiments treat each annotation item as a single "unit" of effort.
 - But not all items are equally easy to annotate. E.g. long sentences take longer than short sentences.
 - Choosing all the complex items instead of the simple items might mean high accuracy is reached with fewer annotations, but if a complex item takes significantly longer, then the comparison is meaningless.

- A real evaluation needs to measure actual annotation time/cost.
- To choose the best item/annotation/annotator combo, the system must model the potential benefit of the annotation, the effort/time/cost of the annotation, and potentially the computational effort involved.

4 Graph Fragment Language

- A fast, flexible scheme for annotating dependency trees/graphs
- Allows partial annotations (leaving part of the tree unspecified), which is useful:
 - when the right analysis is unclear
 - when the right analysis is so obvious that the annotation contributes nothing
 - when the annotator needs to just focus on one kind of analysis
- Certain portions of the analysis may require an expert, while others may be doable by a non-expert.

5 Multi-Domain/Language

- If you have annotated data in one domain/language, you try to connect annotated data to unannotated data and propagate the information
- If you have sentences in multiple languages, and a good parser in one of the languages, you can use it to guide a parser for the other languages.