Machine Learning for NLP

- Until the early 1990’s, NLP systems were built manually with hand-crafted dictionaries and rules.
- Manually creating systems is time-consuming and requires linguistic expertise. Humans are especially prone to errors of omission.
- As large electronic text corpora became increasingly available, researchers began using machine learning techniques to automatically build NLP systems. Today, the vast majority of NLP systems use machine learning!
- ML-based NLP systems are generally more robust (less fragile) and have good coverage, when sufficiently trained.

Machine Learning (ML) Basics

- Supervised machine learning begins with training data, which are examples that have been labelled (“annotated”) by a person with the correct answers.
  - To train a part-of-speech tagger, a human would label each word with its correct part-of-speech in sample texts.
  - To train a parser, a human would generate the correct parse tree for each sentence in sample texts.
- The ML algorithm learns how to solve the problem using the labelled examples. The more training data, the better!
- The training texts should be the same kind of texts that the NLP system will be expected to process when it is deployed!

Data Sets Used for Experimentation

To determine how well an NLP system performs, you also need labelled data. So the annotated texts are usually divided into 3 subsets:

Training Set: Data used to train the ML algorithm. The developer may also look at this data to help design the system. This is usually the largest subset.

Tuning Set: Data set aside to assess how well the program performs on unseen data and/or to set parameters. Helps to minimize overfitting.

Blind Test Set: Data set aside to perform a final evaluation of how well the program performs on new data. The developer should never look at these texts!

Cross-Validation

An alternative experimental set-up is called cross-validation, or informally, jack-knifing. The data is partitioned into $n$ subsets (folds) and $n$ experiments are done.

For each subset $T_i$, the data is trained using the other $n - 1$ partitions and then evaluated on $T_i$. Then the results over the folds are averaged.

Two advantages of cross-validation are:
(1) can use all of the data for testing
(2) simulates $n$ experiments
Machine Learning Techniques

- ML algorithms gather statistical properties over a collection of texts to learn how to generalize from the examples (instances) and make future decisions.

- Many different learning algorithms and models exist!

- Each example is typically represented as a feature vector. Each feature can take a range of values.

- The classifier uses combinations of the features, often weighted, to assign class labels.

Feature Engineering

- The performance of any ML-based system depends crucially on the features used to represent the examples.

- The choice of features is more important than the choice of learning algorithm!
  - given inadequate features, no ML algorithm will do well.
  - given strong features, most ML algorithms will do well.

- There are many ML algorithms that have different strengths, weaknesses, and biases. But designing good features is always critical to performance.

Feature Engineering: Bird or Mammal?

<table>
<thead>
<tr>
<th>Feature</th>
<th>Example #1</th>
<th>Example #2</th>
<th>Example #3</th>
<th>Example #4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Utah</td>
<td>Idaho</td>
<td>Oregon</td>
<td>Arizona</td>
</tr>
<tr>
<td>Behavior</td>
<td>eating</td>
<td>drinking</td>
<td>sleeping</td>
<td>walking</td>
</tr>
</tbody>
</table>
### Feature Engineering: Bird or Mammal?

<table>
<thead>
<tr>
<th>Feature</th>
<th>Example #1</th>
<th>Example #2</th>
<th>Example #3</th>
<th>Example #4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Utah</td>
<td>Idaho</td>
<td>Oregon</td>
<td>Arizona</td>
</tr>
<tr>
<td>Behavior</td>
<td>eating</td>
<td>drinking</td>
<td>sleeping</td>
<td>walking</td>
</tr>
<tr>
<td>Weight</td>
<td>2 lbs</td>
<td>3 lbs</td>
<td>5 lbs</td>
<td>20 lbs</td>
</tr>
<tr>
<td>Color</td>
<td>tan</td>
<td>black</td>
<td>black</td>
<td>brown</td>
</tr>
<tr>
<td>Wings</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

The more discriminating the features, the better the system will perform!
The Importance of Baselines

- It is essential to understand how hard a problem is! Otherwise, you can be fooled by results that aren’t as good as they seem.
  Ex: a POS tagger with < 90% accuracy is worthless!

- Simple techniques often work surprisingly well and should be used to establish baseline performance levels. Examples:
  - **POS tagging**: most frequent tag
  - **Sentence splitting**: split after . ! ?
  - **Word sense disambiguation**: most frequent sense

- It is also essential to know the underlying class distributions.
  Ex: Suppose 90% of documents are news and 10% are editorials. A classifier that identifies 91% as news isn’t very impressive!

Performance Measures

Some common performance measures are:

- **Accuracy**: the percentage of instances assigned a correct label

- **Recall**: for a specific category C, the percent of true instances of C that are correctly labeled ($\frac{\# \text{ correctly labeled as } C}{\# \text{ true instances of } C}$)

- **Precision**: for a specific category C, the percent of instances assigned the label C that are correctly labeled ($\frac{\# \text{ correctly labeled as } C}{\# \text{ labeled as } C}$)

- **F Score**: the harmonic mean of Recall and Precision
  \[ \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \]

Manual Annotation

We manually annotate texts for several reasons:

- to understand the nature of text (e.g., what % of sentences in news articles are opinions?)

- to establish the level of human performance (e.g., how well can people assign POS tags?)

- to evaluate a computer model for some phenomenon (e.g., how often does my tagger or parser find the correct answer?)

The Perils of Manual Annotation

*Manual annotation can be deceptively tricky!* Suppose you are asked to annotate the targets of terrorist attacks in news reports:

- What is a terrorist attack? A pipe bomb in the library? A kidnapping of a child by their father? A kidnapping of a CEO for ransom?

- What is a target? Someone's house during an assassination attempt? The cars in the parking lot during a stadium bombing? The lampposts?

- Which references do you label? Every reference to the target or only the ones in a context describing the attack?


- Do you label a conjunction as a one or two targets?
Annotation Consistency is a Must

• It is essential to have good inter-annotator agreement (i.e., different annotators produce similar annotations) to ensure reproducibility and a well-defined task.

• To achieve high inter-annotator agreement, you must:
  – Precisely define the annotation task
  – Write detailed annotation guidelines, with many examples and anticipated boundary cases
  – Train the annotators
  – Establish good levels of inter-annotator agreement.