05.03 Prepositional Phrase Attachment 2/3

NLP

Parsing

Prepositional Phrase Attachment (2)

Supervised Learning: Evaluation

- Manually label a set of instances.
- Split the labeled data into training and testing sets.
- Use the training data to find patterns.
- Apply these patterns on the testing data set.
- For evaluation: use **accuracy** (the percentage of correct labels that a given algorithm has assigned on the testing data).
- Compare with a simple baseline method.
- What is the simplest baseline method?

Answer

- The simplest supervised baseline method is to find the more common class (label) in the training data and assign it to **all** instances of the testing data set.

Algorithm 1

label the tuple as “low” (default)

Random baseline

- A **random** unsupervised baseline would have been to label each instance in the testing data set with a random label, 0 or 1.
- Practically, random performance is the lower bound against which any non-random methods should be compared.

Measuring The Accuracy Of The Supervised Baseline Method (Algorithm 1)

- In the official training data set (RRR94), the rate of occurrence of the 1 label (low attachment) is 52.2% (10,865/20,801).
- Is the accuracy of this baseline method then equal to 52.2%?

Solution

- No, this is not how accuracy is computed. It has to be computed on the testing set, not the training set.
- Using the official split, the accuracy of this method on the testing set is 59.0% (1,826/3,097). One shouldn’t think of this number as a good result. The difference (+6.8% going from training to testing) could have been in the opposite direction, resulting in a performance below random.

Observations

- If the baseline method is simple and if the testing set is randomly drawn from the full data set and the data set is large enough, one could expect that the accuracy on the testing set is comparable to the one on the training set. Note that the PTB data set is drawn from business news stories. If one were to train a method on this data and test it on a different set of sentences, e.g., from a novel, it is possible that the two sets will have very different characteristics.
- The more complicated the method is, however, the more likely it will be that it will "overfit" the training data, learning patterns that are too specific to the training data itself and which may not appear in the testing data or which may be associated with the opposite class in the testing data.

Upper Bounds On Accuracy

- The 52% accuracy we've seen so far is our current lower bound. Now, what is the upper bound?
- Usually, human performance is used for the upper bound. For PP attachment, using the four features mentioned earlier, human accuracy is around 88%.
- So, a hypothetical algorithm that achieves an accuracy of 87% is in fact very close to the upper bound (on a scale from 52% to 88%, it is 97% of the way to the upper bound).

Using Linguistic Knowledge

- One way to beat the two baselines is to use linguistic information. For example, the preposition "of" is much more likely to be associated with low attachment than high attachment.
- In the training data set, this number is an astounding 98.7% (5,534/5,607)
- Therefore the feature *prep_of* is very valuable. What are the two main reasons?

Answer

- Reason 1 – it is very informative (98.7% of the time it is linked with the low attachment class)
- Reason 2 – it is very frequent (27.0% of the entire training set – 5607/20801).
- Reason 1 alone would not be sufficient!

Sidebar 1/3

- The PTB (Penn Treebank) data set has been used for competitive evaluation of pp attachment algorithms since 1994.
- Each new algorithm is allowed to look at the training and development sets and use any knowledge extracted from them.
- The test set data can never be looked at and can be used only once per algorithm for its evaluation.
- Doing the contrary (repeatedly tuning a new algorithm based on its performance on the designated test set) results in a performance level that is irreproducible on new data and such approaches are not allowed in NLP research.
- Note that the development set can be used in a well-specified way as part of the training process.

Sidebar 2/3

- Let’s look at the training data then and see if we can learn some patterns that would help us improve over the silly “label everything as noun attachment” baseline and its 52% expected accuracy.
- For example, some prepositions tend to favor particular attachments.
- Let’s start with “against”. It appears 172 times in the training set, of which 82 (48%) are noun attached, the rest (52%) being high attached. This ratio (48:52) is very similar to the baseline (52:48), so clearly, knowing that the preposition is “against” gives us very little new information.

Sidebar 3/3

- Furthermore, the total occurrence of “against” in the training corpus is rather small (less than 1% of the prepositional phrases).
- In two special cases, however, the identity of the preposition gives a lot of information.
- At one extreme, “of” is associated with low attachment in 99% of the cases, whereas at the other end of the scale, “to” is associated with high attachment in 82% of its appearances.
- It is also important to note that both of these prepositions are fairly common in the training set (“to” representing 27% of all prepositions and “of” accounting for another 11% of them).
- With this knowledge, we can build a very simple decision list algorithm (Brill and Resnik) that looks like this (the rules are sorted by their expected accuracy on their majority class).

Are There Any Other Such Features?

- Yes, *prep_to*: 2,182/2,699 = 80.8% in favor of high attachment.
- Which leads us to our next algorithm:

Algorithm 2

```
If the preposition is "of", label the tuple as "low".
Else
    If the preposition is "to", label the tuple as "high".
    Else
        label the tuple as "low" (default)
```

Sidebar 1/2

- Let's compare the performance of this simple decision list algorithm with that of the baseline.
- On the training set, the first rule would fire in 5,577 cases, of which 5,527 will be correctly labeled as low attachment and another 50 will be incorrectly labeled as high attachment (the accuracy of this rule is expected to be 99% on the training set).
- The second rule (with an expected accuracy of 82% on the training set) would result in 2,672 decisions, of which 2,172 will be correctly processed as high attachment and 500 will be mislabeled as low attachment.
- Finally, the default rule will kick in, whenever the first two rules are not applicable. In this case, it will apply on all remaining phrases (20,801 – 5,577 – 2,672 = 12,552 cases).
- Specifically, among these, it will result in 4,837 phrases correctly labeled as low and 7,714 incorrectly labeled as high.

Sidebar 2/2

- Overall, we have 5,527 + 2,172 + 4,837 = 12,536 correct decisions, or an accuracy of 12,536/20,801 = 60%.
- Clearly Algorithm 2 outperforms Algorithm 1 on the training data.
- This is not surprising, since its expected accuracy should be no less than the worst expected accuracy of its rules (that of rule 3) and it is likely to be higher than that lowest number because of the priority given to the easier to classify cases (rules 1 and 2).
- More complicated algorithms can look at additional features, e.g., the nouns or the verb, or some combination thereof. For example, the verb "dropped" appears 75/81 times (93%) along with high attachment and only 7% with low attachment.

NLP