CSCI 4152/6509 — Natural Language Processing

5-Oct-2009

Lecture 11: CNG Classification

Room: FASS 2176
Time: 11:35 – 12:25

Previous Lecture

- Aside: SpamAssassin `http://spamassassin.apache.org/`
- IR Evaluation: F-measure,
- Text mining: text classification and text clustering,
- Text classification: problem definition, types of text classification,
- Evaluation measures in text classification, micro- and macro-averaging,
- Evaluation methods for classification:
  - general issues — overfitting and underfitting,
  - methods:
    1. training error,
    2. train and test,
    3. n-fold cross-validation

Evaluation Methods for Classification

- General issues in classification: overfitting and underfitting
- Example with polynomial-based function learning
- Evaluation methods in classification:
  1. training error
  2. train and test
  3. n-fold cross-validation

N-fold cross-validation. In this method, the data is randomly partitioned into n equal parts. n experiments are performed, where in each another part is taken as the testing data, while remaining parts are used for training. At the end, the results over experiments are averaged. This is unbiased testing that gives more statistical significance than train-and-test, but it is not applicable if we need to examine the training data during classifier construction.

6.3 Parser Evaluation

PARSEVAL Measures for Parser Evaluation

Described in section 14.7, page 479, of the textbook.
Parsers are usually evaluated using the **PARSEVAL measures.** To compute the PARSEVAL measures, the parse trees are first decomposed into **labelled constituents (LC),** which are triples consisting of the starting and ending point of a constituent's span in a sentence, and the constituent's label. For each sentence, the sets of constituents obtained from the parse tree obtained from a parser (PT), and from the given, "gold standard" parse tree (GT) are compared. The following measures are usually calculated:

$$\begin{aligned}
\text{labelled recall} &= \frac{\text{number of correct LC in PT}}{\text{number of LC in GT}} \\
\text{labelled precision} &= \frac{\text{number of correct LC in PT}}{\text{number of LC in PT}} \\
\text{F-measure} &= \frac{2 \cdot (\text{labelled precision}) \cdot (\text{labelled recall})}{(\text{labelled precision}) + (\text{labelled recall})}
\end{aligned}$$

The fourth measure, number of cross-brackets, is also used.
The state-of-the-art parsers obtain up to 90% precision and recall on the Penn Treebank data.

Example

Let us consider the following two sentences:

> Time flies like an arrow.
> *and*
> He ate the cake with a spoon.

These could easily be ambiguous sentences for a parser, and let us assume that our "gold standard" parse trees; i.e., preferred or correct parse trees are as follows:

Gold standard

```
(S (NP (NN time) (NN flies))
   (VP (VB like)
       (NP (DT an) (NN arrow))))

(S (NP (PRP he))
   (VP (VBD ate) (NP (DT the)
                     (NN cake))
       (PP (IN with)
           (NP (DT a) (NN spoon)))))
```

while the parse trees returned by a parser are:

Parser result

```
(S (NP (NN time))
   (VP (VB flies)
       (PP (IN like)
           (NP (DT an) (NN arrow)))))

(S (NP (PRP he))
   (VP (VBD ate)
       (NP (DT the) (NN cake)
           (PP (IN with)
               (NP (DT a) (NN spoon))))))
```

First, we list the labeled edges of the parse trees:

```
  time flies like an arrow
0     1     2    3  4     5

Gold standard:
S  0,5 time flies like an arrow
NP 0,2 time flies
NN 0,1 time
NN 1,2 flies
VP 2,5 like an arrow
VB 2,3 like
NP 3,5 an arrow
DT 3,4 an
NN 4,5 arrow

Parser result:
S  0,5 time flies like an arrow
NP 0,1 time
NN 0,1 time
VP 1,5 flies like an arrow
VB 1,2 flies
PP 2,5 like an arrow
IN 2,3 like
NP 3,5 an arrow
DT 3,4 an
NN 4,5 arrow
```

```
 he ate the cake with a spoon
0  1   2   3    4    5 6     7

Gold standard:
S   0,7 he ate ...ke with a spoon
NP  0,1 he
PRP 0,1 he
VP  1,7 ate the cake with a spoon
VBD 1,2 ate
NP  2,4 the cake
DT  2,3 the
NN  3,4 cake
PP  4,7 with a spoon
IN  4,5 with
NP  5,7 a spoon
DT  5,6 a
NN  6,7 spoon

Parser result:
S   0,7 he ate the cake with a spoon
NP  0,1 he
PRP 0,1 he
VP  1,7 ate the cake with a spoon
VBD 1,2 ate
NP  2,7 the cake with a spoon
DT  2,3 the
NN  3,4 cake
PP  4,7 with a spoon
IN  4,5 with
NP  5,7 a spoon
DT  5,6 a
NN  6,7 spoon
```

After counting the number of correctly identified edges (true positives), non-identified edges (false negatives), incorrectly identified edges (false positives), and the total number of edges, we can calculate precision and recall: Precision$= \frac{17}{23} \approx 0.739130434782609$ and Recall $= \frac{17}{22} \approx 0.772727272727273$.

6.4 Text Clustering

- task definition
- the simple k-means approach
- hierarchical clustering
  - agglomerative, and
  - divisive
- evaluation
  - inter-cluster similarity
  - cluster purity (classes known)
  - entropy or information gain (classes known)

6.5 CNG—Common N-Gram analysis for text classification

- Method based on character n-grams
- Language independent
- Based on creating n-gram based author profiles
- kNN method (k Nearest Neighbours)
- similarity measure:

$$\sum_{g \in D_1 \cup D_2} \left(\frac{f_1(g) - f_2(g)}{\frac{f_1(g)+f_2(g)}{2}} \right)^2 = \sum_{g \in D_1 \cup D_2} \left(\frac{2 \cdot (f_1(g) - f_2(g))}{f_1(g) + f_2(g)} \right)^2 \quad (1)$$

where $f_i(g) = 0$ if $g \notin D_i$.