Improving machine learning results with filter rules

Erik Tjong Kim Sang
CNTS, University of Antwerp

Main research questions

Let us suppose that we have optimized the parameters and features of a certain machine learning for a particular task.

- Can we obtain an even better performance?
- What should we do to achieve this?

Motives

We are increasingly interested in performing more complex natural language processing tasks: summarization, question answering, document classification...

These tasks will benefit from basic NLP tools that perform very well: taggers, chunkers, (shallow) parsers...

There is also an interest in adapting these tools to specific domains, like medical text.

What should we do?

The answer is simple...

We need more training data!

There still will be the problem of unknown words

Evaluation of an English named entity recognition system:

development data	Precision	Recall	$F_{\beta=1}$
Known unambiguous (40%)	94.3%	96.3%	95.2
Known ambiguous (37%)	89.6%	90.3%	89.9
Unknown (23%)	69.8%	68.7%	69.3
All phrases (100%)	87.1%	87.8%	87.4

Phrases are unknown when they include a word that does not appear in the training data. They are ambiguous when they consists of known words of which one receives more than one class in the training data.

A careful error analysis might help

their stay on top , though , may be short-lived as title rivals Essex , Derbyshire and Surrey/OL all closed in on victory while Kent made up for lost time in their rain-affected match against Nottinghamshire .
by the close Yorkshire had turned that into a 37-run advantage but off-spinner Such/PM had scuttled their hopes , taking four for 24 in 48 balls and leaving them hanging on 119 for five and praying for rain .
Australian Tom/PP Moody/PO took six for 82 but Chris Adams , 123 , and Tim O'Gorman , 109 , took Derbyshire to 471 and a first innings lead of 233 .
they were held up by a gritty 84 from Paul Johnson but ex-England/M. fast bowler Martin McCague took four for 55 .

So...

A quick look at the errors in the output seems to suggest that the system makes many basic errors which can be corrected easily.

We can give the system access to more information but this will probably slow it down even when processing 'easy' cases.

We will attempt to correct the errors with post-processing filter rules.

The task: named entity recognition

The task involves identifying three types of named entities in English text. Example:

[ORG U.N.] official [PER Ekeus] heads for [LOC Baghdad] .

Apart for names for persons (PER), organizations (ORG) and locations (LOC), we are also looking for miscellaneous names (MISC).

Base performance

A base performance for this task has been obtained by optimizing the feature parameters of the Memory-Based Tagger (MBT) with a bidirectional search with beam size 1.

development	Precision	Recall	$F_{\beta=1}$
LOC	91.1%	91.8%	91.5
MISC	84.6%	79.2%	81.8
ORG	78.4%	84.0%	81.1
PER	91.0%	90.8%	90.9
overall	87.1%	87.8%	87.4

The first word of each sentence and all words of completely capitalized sentences have been changed to their default case.

Output filter rules

name	$F_{\beta=1}$	cases	level	extra data
length2	87.89	33	phrase	-
unify	87.86	52	document	-
length3	87.82	27	phrase	-
titles	87.77	25	phrase	-
hyphen	87.58	45	word	case info / exceptions
sportresults	87.55	12	sentence	-
firstnames	87.52	26	phrase	first names / exceptions
lastword	87.50	11	phrase	exceptions (2 lists)
unknown	87.48	5	phrase	-
numbers	87.47	4	phrase	-
conj	87.46	4	phrase	-
isocaps	87.45	3	phrase	exceptions
bigram	87.43	15	document	-

Rule examples

IF annotated phrase has length 2
THEN
    IF phrase contains PER tag
    THEN change all tags to PER
    ELSE change first tag to second tag

IF annotated word was tagged PER in document
AND word received PER tag more often than current tag
THEN change current tag to PER

Rule combination

Rules have been combined by starting with the best rule and adding rules until no further performance gain could be obtained.

development	Precision	Recall	$F_{\beta=1}$
base	87.1%	87.8%	87.4
after filtering	89.7%	90.0%	89.9

Both overall precision and overall recall improved and the error for the $F_{\beta=1}$ rate decreased with 20%.

Generalization

The rules have been written by examining the test data. Therefore we need to evaluate them on another test set in order to test their generalization capabilities.

test data	Precision	Recall	$F_{\beta=1}$
base	75.8%	78.0%	76.9
after filtering	79.0%	80.2%	79.6

Again both overall precision and overall recall improved. The $F_{\beta=1}$ error decreased with 12%.

Effects on ambiguous and unknown words

development data	Precision	Recall	$F_{\beta=1}$	
Known unambiguous (40%)	96.2%	96.8%	96.5	-27%
Known ambiguous (37%)	92.1%	91.5%	91.8	-19%
Unknown (23%)	74.6%	75.4%	75.0	-19%
All phrases (100%)	89.7%	90.0%	89.9	-20%

test data	Precision	Recall	$F_{\beta=1}$	
Known unambiguous (29%)	88.9%	89.7%	89.3	-15%
Known ambiguous (36%)	81.3%	80.3%	80.8	-7%
Unknown (35%)	68.9%	72.4%	70.6	-13%
All phrases (100%)	79.0%	80.2%	79.6	-12%

Comparison with other work

test data	Precision	Recall	$F_{\beta=1}$
System combination	88.99%	88.54%	88.76±0.7
Maximum Entropy	88.12%	88.51%	88.31±0.7
10 more systems...	...	...	...
Ortographic tries	81.60%	78.05%	79.78±1.0
Memory-Based Learning	78.97%	80.24%	79.60±0.7
Memory-Based Learning	76.33%	80.17%	78.20±1.0
Memory-Based Learning	75.84%	78.13%	76.97±1.2
Neural Networks	69.09%	53.26%	60.15±1.3
Baseline	71.91%	50.90%	59.61±1.2

Concluding remarks

- Applying output filter rules to machine learning output helps.
- An error analysis of the data reveals some of the weaknesses of the system.

Future work

- Integration of NLP tools in this system.
- Acquire more training data.

Where will we be in 10 years?

- Computers will be faster and have more memory.
- Learning systems will perform a little bit better.
- We will have more training data for most NLP tasks.
- We will be working on integrating systems that perform different tasks.
- Hopefully the NLP systems have been improved enough to do useful things with unannotated data.