Applying System Combination to Base Noun Phrase Identification

Erik Tjong Kim Sang, Walter Daelemans,
Hervé Déjean, Rob Koeling, Yuval Krymolowski,
Vasin Punyakanok and Dan Roth

Goal

Finding a better learning method for recognizing base noun phrases (baseNPs).

Inspiration

The project Learning Computational Grammars in which seven European sites apply machine learning methods for NP recognition: http://lcg-www.uia.ac.be

Tjong Kim Sang (2000) which shows that baseNP recognition can be improved by combining the results of different classifiers.

Base Noun Phrases

BaseNPs are non-overlapping, non-recursive base noun phrases. Here is an example sentence:

In [early trading] in [Hong Kong]
[Monday] , [gold] was quoted at
[$ 366.50] [an ounce] .

This sentence contains six baseNPs. They can also be represented with so-called chunk tags:

In$_{\mathrm{O}}$ early$_{\mathrm{I}}$ trading$_{\mathrm{I}}$ in$_{\mathrm{O}}$ Hong$_{\mathrm{I}}$ Kong$_{\mathrm{I}}$
Monday$_{\mathrm{B}}$,$_{\mathrm{O}}$ gold$_{\mathrm{I}}$ was$_{\mathrm{O}}$ quoted$_{\mathrm{O}}$ at$_{\mathrm{O}}$
\$$_{\mathrm{I}}$ 366.50$_{\mathrm{I}}$ an$_{\mathrm{B}}$ ounce$_{\mathrm{I}}$.$_{\mathrm{O}}$

Tag I is used for words inside a baseNP, tag O for words outside baseNPs and tag B for baseNP-initial words after another baseNP.

Classifier combination

Suppose we use five learning algorithms for predicting whether an NP starts at a certain position or not.

	c_1	c_2	c_3	c_4	c_5	correct
word$_1$	.	.	.	.	.	.
word$_2$	[	[	[	[	[	[
word$_3$	.	.	.	.	.	.
word$_4$	[	.	[	[	[	[
word$_5$	.	.	[	.	.	.
word$_6$	[	[	[	[	.	[
word$_7$	[	.	.	.	.	.
word$_8$	[	[	[	.	[	[

We can combine the results with majority voting: choose the result that has been predicted most often.

Obtaining different classifiers

How do we obtain different results for one task?

1. Use one learning algorithm with different parameters or with different versions of the data.

   Disadvantage: the errors made by these systems are more related and therefore there is fewer improvement to gain.

2. Use different learning algorithms.

   Disadvantage: most algorithms require a lot of tuning in order to get a reasonable performance.

We have used both methods: we combined the results of seven learning algorithms of which three used a combination of processing five different output representations.

Different NP representations

	IOB1	IOB2	IOE1	IOE2	O	C
In	O	O	O	O	.	.
early	I	B	I	I	[	.
trading	I	I	I	E	.	]
in	O	O	O	O	.	.
Hong	I	B	I	I	[	.
Kong	I	I	E	E	.	]
Monday	B	B	I	E	[	]
,	O	O	O	O	.	.
gold	I	B	I	E	[	]
was	O	O	O	O	.	.
quoted	O	O	O	O	.	.
at	O	O	O	O	.	.
$	I	B	I	I	[	.
366.50	I	I	E	E	.	]
an	B	B	I	I	[	.
ounce	I	I	I	E	.	]
.	O	O	O	O	.	.

Machine learning methods (1)

ALLiS

is a theory refinement system which starts from a chunking model based on local part-of-speech (POS) tags and refines it by including lexical information and information about context.

C5.0 & **IGTREE**

build decision trees from the training material and use information gain for determining feature weights. C5.0 uses POS information only while IGTREE uses both POS and lexical information.

IB1IG (MBL)

is a memory-based learner which classifies new items based on their similarity with training data items and uses information gain for determining feature weights.

Machine learning methods (2)

MaxEnt

makes a probabilistic model that matches the training material as well as possible according to the principle of maximum entropy.

MBSL

determines the most likely division in chunks by comparing test material to sequences of POS tags that make up chunks in the training data.

SNoW

is a network of linear units which learn with the Winnow update rule. It predicts open and close brackets and combines these to chunks with a constraint satisfaction algorithm.

Combination methods

We compare five different voting methods and four versions of stacked classifiers:

Majority voting
Each classifier gets one vote. The majority wins.

TotPrecision, TagPrecision, Precision-Recall
The weight of each classifier is determined by its performance on some held-out part of the training data.

TagPair
Uses weights for results which are associated with results of classifier pairs.

Stacked classifiers
A second classifier processes the results and determines the most probable classification.

Approach

1. Choose data sets for tuning the parameters of the learning methods and selecting the best combination method (WSJ sections 15-18 for training and 21 for testing).

2. Apply three learning methods to this data while using the five output representations.

3. Combine the results with majority voting.

4. Apply the four other learning methods to the data.

5. Combine the seven results with the nine combination methods.

6. Select the best combination method and apply the seven learners with this method to a standard data set (WSJ sections 15-18 for training and 20 for testing).

System-internal combination

section 21	MBL	MaxEnt	IGTree
IOB1	91.68	92.43	87.88
IOB2	91.79	92.14	90.03
IOE1	91.54	92.37	82.80
IOE2	92.06	92.13	89.98
O+C	92.03	92.26	89.37
Majority	92.82	92.60	91.92

The performances are measured with $F_{\beta=1}$, a combination of precision and recall for baseNP recognition:

$$F_{\beta=1} = \frac{(\beta^2+1) * \text{precision} * \text{recall}}{\beta^2 * \text{precision} + \text{recall}} \qquad (\beta = 1)$$

System-external combination

section 21	O	C	$F_{\beta=1}$
ALLiS	97.87%	98.08%	92.15
C5.0	97.05%	97.76%	89.97
IGTree	97.70%	97.99%	91.92
MaxEnt	97.94%	98.24%	92.60
MBL	98.04%	98.20%	92.82
MBSL	97.27%	97.66%	90.71
SNoW	97.78%	97.68%	91.87
Simple Voting			
Majority	98.08%	98.21%	92.95
TotPrecision	98.08%	98.21%	92.95
TagPrecision	98.08%	98.21%	92.95
Precision-Recall	98.08%	98.21%	92.95
Pairwise Voting			
TagPair	98.13%	98.23%	93.07
Memory-Based			
Tags	98.24%	98.35%	93.39
Tags + POS	98.14%	98.33%	93.24
Decision Trees			
Tags	98.24%	98.35%	93.39
Tags + POS	98.13%	98.32%	93.21

Best-n combination

section 21		O	C	$F_{\beta=1}$
3	Majority	98.17%	98.29%	93.30
4	Majority	98.23%	98.29%	93.37
5	Majority	98.22%	98.31%	93.44
6	Tag-Pair	98.22%	98.31%	93.45
7	Memory-Based	98.24%	98.35%	93.39

Results standard baseNP data sets

section 20	Precision	Recall	$F_{\beta=1}$
Best-five combination	94.18%	93.55%	93.86
Tjong Kim Sang (2000)	93.63%	92.89%	93.26
Muñoz et al. (1999)	92.4%	93.1%	92.8
Ramshaw and Marcus (1995)	91.80%	92.27%	92.03
Argamon et al. (1999)	91.6%	91.6%	91.6

Concluding remarks

1. Combining classifiers improves performance for NP recognizers.

2. In this experiment setup, the simple majority voting applied to the best-n classifiers performs as well as any other evaluated combination method.

3. The combination methods which were tested are sensitive to the inclusion of results of poor quality.

Future work

Include results of more classifiers?

Pointers

E-mail addresses of authors

Erik Tjong Kim Sang	erikt@uia.ua.ac.be
Walter Daelemans	daelem@uia.ua.ac.be
Hervé Déjean	dejean@sfs.nphil.uni-tuebingen.de
Rob Koeling	koeling@cam.sri.com
Yuval Krymolowski	yuvalk@macs.biu.ac.il
Vasin Punyakanok	punyakan@cs.uiuc.edu
Dan Roth	danr@cs.uiuc.edu

Related web page

http://lcg-www.uia.ac.be/~erikt/npcombi/