Selected Topics in Applied Machine Learning: An integrating view on data analysis and learning algorithms

ESSLLI '2015
Barcelona, Spain

http://ufal.mff.cuni.cz/esslli2015

Barbora Hladká
hladka@ufal.mff.cuni.cz

Martin Holub
holub@ufal.mff.cuni.cz

Charles University in Prague,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Block 2.1
Data analysis (cntnd)

Motivation No. 1

We, as students of English, want to understand the following sentences properly

- *He broke down and cried when we talked to him about it.*
- *Major cried, jabbing a finger in the direction of one heckler.*

If we are not sure, we check definitions for the verb *cry* in a dictionary

Verb Patterns Recognition

CRY -- dictionary definitions

cry ◆◆◆◆◇

1 cry; cries; crying; cried
When you cry, tears come from your eyes, usually because you are unhappy or hurt.
I hung up the phone and started to cry.
Please don't cry.
He cried with anger and frustration.
...a crying baby.
VB

2 cry; cries; crying; cried
If you cry something, you shout it or say it loudly.
`Nancy Drew,' she cried, `you're under arrest!'.
I cried: `It's wonderful news!'
VB

5 cry; cries
You can refer to a public protest about something or appeal for something as a cry of some kind. (JOURNALISM)
There have been cries of outrage about this expenditure.
Many other countries have turned a deaf ear to their cries for help.
N-COUNT: usu N of/for n

Verb Patterns Recognition

Based on the explanation and the examples of usage, we can recognize the two meanings of *cry* in the sentences

- *He broke down and cried when we talked to him about it.* **[1]**
- *Major cried, jabbing a finger in the direction of one heckler.* **[2]**

Verb Patterns Recognition

Motivation No. 2

We, as developers of natural language application, need to recognize verb meanings automatically.

Verb Patterns Recognition task (VPR) is the computational linguistic task of lexical disambiguation of verbs

- a lexicon consists of verb usage patterns that correspond to dictionary definitions
- disambiguation is recognition of the verb usage pattern in a given sentence

VPR – Verb patterns

CRY -- Pattern definitions

Pattern 1	**[Human] cry [no object]**
Explanation	[[Human]] weeps usually because [[Human]] is unhappy or in pain
Example	*His advice to stressful women was: ` If you **cry**, do n't cry alone.*

Pattern 4	**[Human] cry [THAT-CL|WH-CL|QUOTE] ({out})**
Explanation	[[Human]] shouts ([QUOTE]) loudly typically, in order to attract attention
Example	*You can hear them screaming and banging their heads, **crying** that they want to go home.*

Pattern 7	**[Entity | State] cry [{out}] [{for} Action] [no object]**
Explanation	[[Entity | State]] requires [[Action]] to be taken urgently
Example	*Identifying areas which **cry** out for improvement or even simply areas of muddle and misunderstanding, is by no means negative -- rather a spur to action.*

E.g., the pattern 1 of *cry* consists of a subject that is supposed to be a Human and of no object.

VPR – Getting examples

Examples for the VPR task are the output of **annotation**.

1. Choosing verbs you are interested in –> **cry**, **submit**
2. Defining their patterns
3. Collecting sentences with the chosen verbs

VPR – Getting examples

4. Annotating the sentences
   - assign a pattern that fits best the given sentence
   - if you think that no pattern matches the sentence, choose "u"
   - if you do not think that the given word is a verb, choose "x"

VPR – Data

Basic statistics

	CRY					SUBMIT				
instances	250					250				
classes	1	4	7	u	x	u	1	2	4	5
frequency	131	59	13	33	14	7	177	33	12	21

VPR – Data representation

instance id	**feature vector**				**target pattern**
	morphological feature family (MS)	morpho-syntactic feature family (STA)	morpho-syntactic feature family (MST)	semantic feature familiy (SEM)	
129825	0 ...	0 ...	0 ...	0 ...	1
...	...	...	...	...	...
...	0 ...	0 ...	0 ...	0 ...	7

For more details, see **vpr.handout** posted at the course webpage.

VPR – Feature extraction

*He broke down and **cried** when we talked to him about it.*

MF_tense_vbd	1	verb past tense – OK
MF_3p_verbs	1	third word preceding the verb is verb – *broke*, OK
MF_3n_verbs	1	third word following the verb is verb – *talked*, OK
...	...	...
STA.LEX_prt_none	1	there is no particle dependent on the verb – OK
STA.LEX_prep_none	1	there is no preposition dependent on the verb – OK
...	...	...
MST.GEN_n_subj	1	nominal subject of the verb – OK
...	...	...
SEM.s.ac	1	verb's subject is Abstract – *he*, KO
...	...	...
tp	1	true target pattern

VPR – Details on annotation

Annotation by 1 expert and 3 annotators

verb	target classes	number of instances	baseline (%)	avg human accuracy (%)	perplexity $2^{H(P)}$	kappa
CRY	1,4,7,u,x	250	52,4	92,2	3,5	0,84
SUBMIT	1,2,4,5,u	250	70,8	94,1	2,6	0,88

- **baseline** is accuracy of the most frequent classifier
- **avg human accuracy** is average accuracy of 3 annotators with respect to the expert's annotation
- **perplexity** of a target class
- **kappa** is Fleiss kappa of inter-annotator agreement

Questions?

Data analysis (cntnd)

Deeper understanding the task by statistical view on the data

We exploit the data in order to make prediction of the target value.

- Build intuition and understanding for both the task and the data
- Ask questions and search for answers in the data
    - **What values do we see**
    - **What associations do we see**
- Do plotting and summarizing

Analyzing distributions of values
Feature frequency

- **Feature frequency**

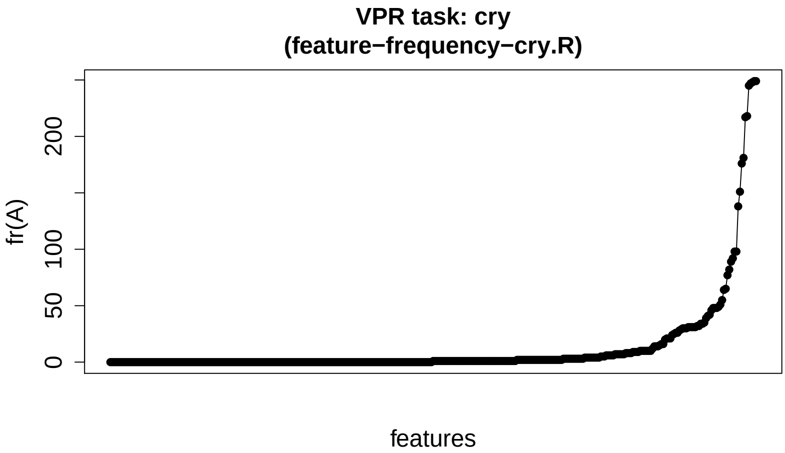
$$\mathrm{fr}(A_j) = \#\{\mathbf{x}_i \mid x_i^j > 0\}$$

where A_j is the j-th feature, $\mathbf{x}_i$ is the feature vector of the i-th instance, and x_i^j is the value of A_j in $\mathbf{x}_i$.

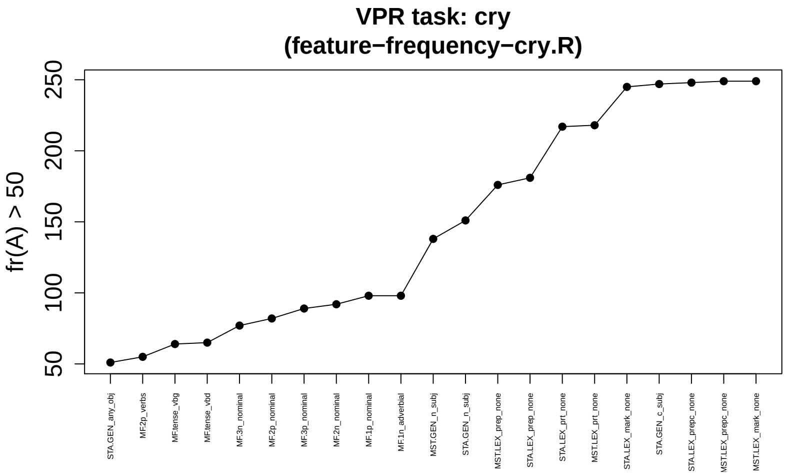
Analyzing distributions of values
Feature frequency

```
> examples <- read.csv("cry.development.csv", sep="\t")
> c <- examples[,-c(1,ncol(examples))]
> length(names(c)) # get the number of features
[1] 363
# compute feature frequencies using the fr function
> ff <- apply(c, 2, fr.feature)
> table(sort(ff))
  0   1   2   3   4   5   6   7   8   9  10  12  14  15  16  20
181  47  26  12   9   3   5   6   4   4   7   1   3   1   2   1

21 24 25 26 28  29  30  31  32  34  35  39  41  42  46  48  49
 3  1  1  2  1   1   3   5   2   2   1   1   1   1   1   3   1

51  55  64  65  77  82  89  92  98 138 151 176 181 217 218 245
 1   1   1   1   1   1   1   1   2   1   1   1   1   1   1   1

247 248 249
  1   1   2
```

Analyzing distributions of values
Feature frequency



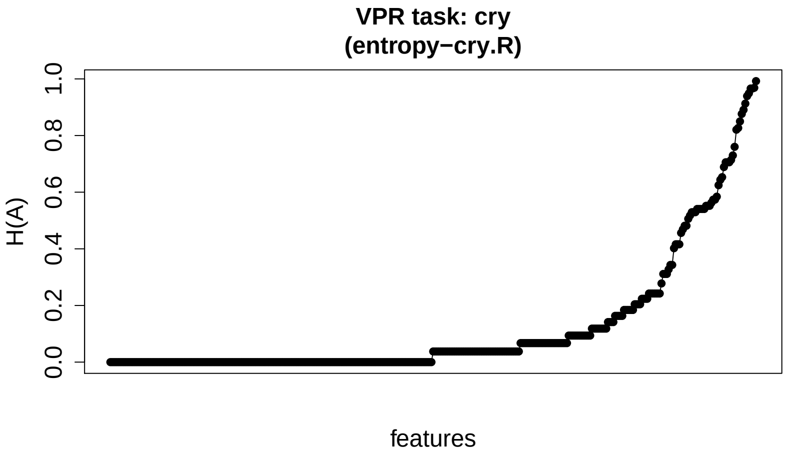
Analyzing distributions of values
Feature frequency



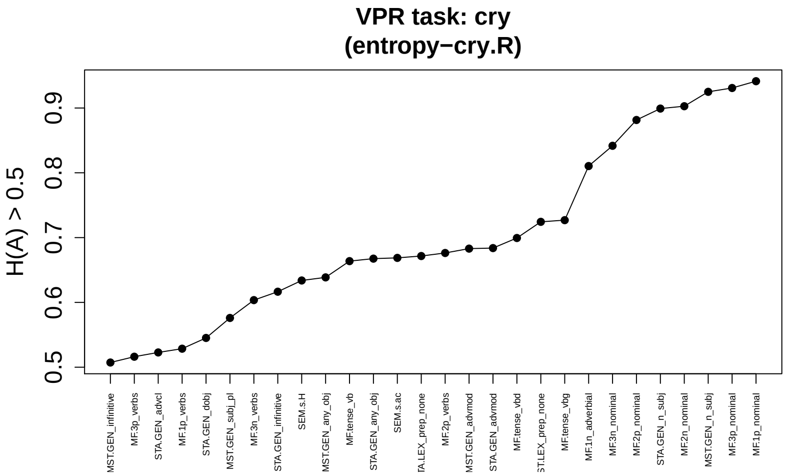
Analyzing distributions of values
Entropy

```
# compute entropy using the entropy function
> e <- apply(c, 2, entropy)
> table(sort(round(e,2)))
   0 0.04 0.07 0.09 0.12 0.14 0.16 0.18  0.2 0.22 0.24 0.28 0.31 0.33
 181   49   27   13    9    4    5    6    4    4    7    1    3    1

 0.34  0.4 0.42 0.46 0.47 0.48 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58
    2    1    3    1    1    2    1    1    3    5    3    1    2    1

 0.62 0.64 0.65 0.69 0.71 0.73 0.76 0.82 0.83 0.85 0.88 0.89 0.91 0.94
    1    1    1    1    4    1    1    1    1    1    1    1    1    1

 0.95 0.97 0.99
    1    3    1
```

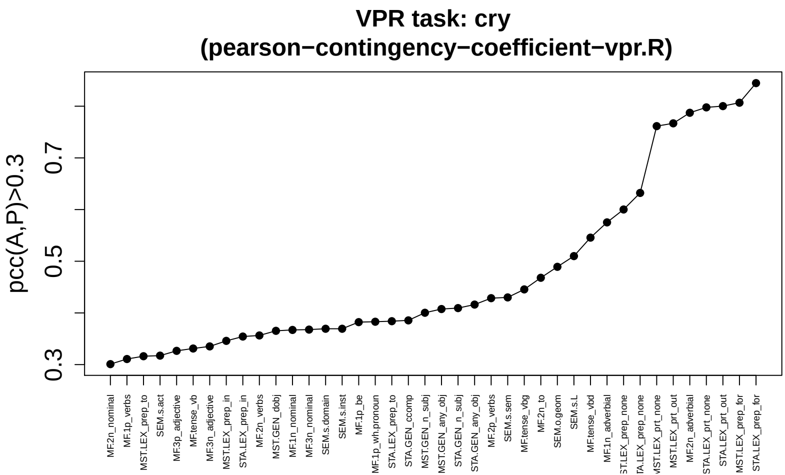
Analyzing distributions of values
Entropy



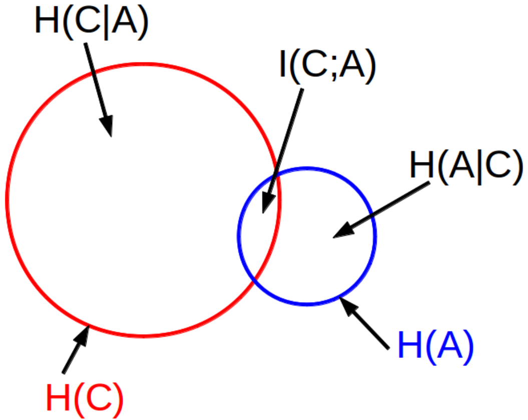
Analyzing distributions of values
Entropy



Association between feature and target value
Pearson contingency coefficient



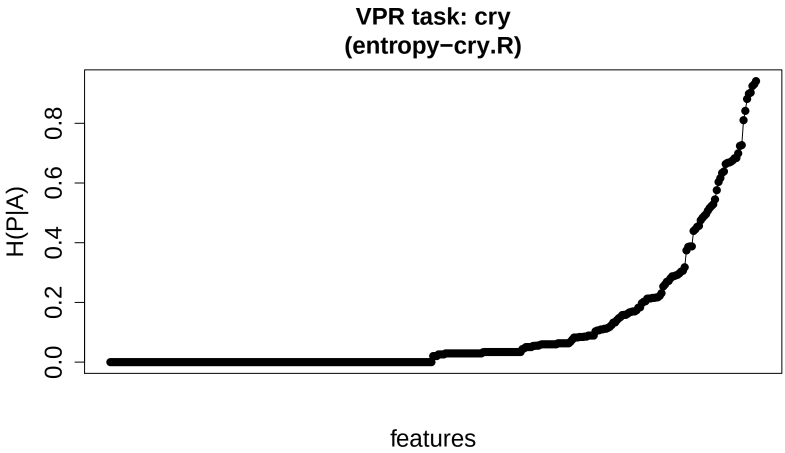
Association between feature and target value
Conditional entropy



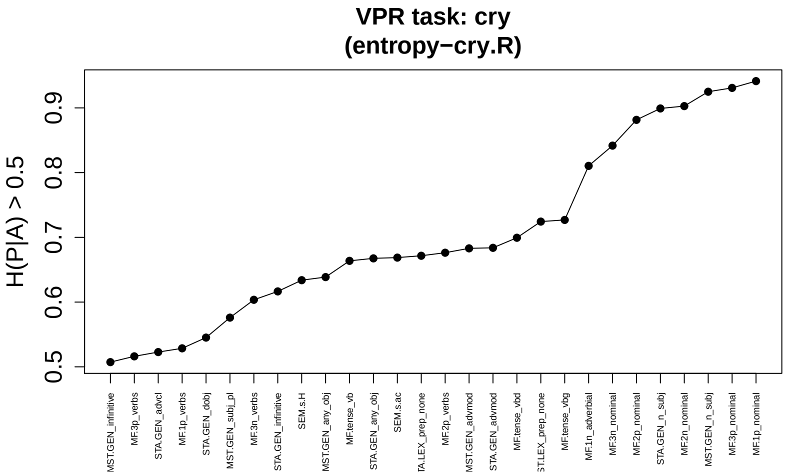
Association between feature and target value
Conditional entropy

```
# compute conditional entropy using the entropy.cond function
ce <- apply(c, 2, entropy.cond, y=examples$tp)
table(sort(round(ce,2)))
  0 0.04 0.07 0.09 0.12 0.14 0.16 0.18  0.2 0.22 0.24 0.28 0.31 0.33
 181   49   27   13    9    4    5    6    4    4    7    1    3    1

 0.34  0.4  0.42 0.46 0.47 0.48 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58
    2    1     3    1    1    2    1    1    3    5    3    1    2    1
 0.62 0.64 0.65 0.69  0.71 0.73 0.76 0.82 0.83 0.85 0.88 0.89 0.91 0.94
    1    1    1    1     4    1    1    1    1    1    1    1    1    1

 0.95 0.97 0.99
    1    3    1
```

Association between feature and target value
Conditional entropy



Association between feature and target value
Conditional entropy



What values do we see

Analyzing distributions of values

Filter out uneffective features from the CRY data

```
> examples <- read.csv("cry.development.csv", sep="\t")
> n <- nrow(examples)
> ## remove id and target class tp
> c.0 <- examples[,-c(1,ncol(examples))]

> ## remove features with 0s only
> c.1 <- c.0[,colSums(as.matrix(sapply(c.0, as.numeric))) != 0]
> ## remove features with 1s only
> c.2 <- c.1[,colSums(as.matrix(sapply(c.1, as.numeric))) != n]
> ## remove column duplicates
> c <- data.frame(t(unique(t(as.matrix(c.2)))))

> ncol(c.0) # get the number of input features
[1] 363
> ncol(c)   # get the number of effective features
[1] 168
```

Methods for basic data exploration
Confusion matrix

Confusion matrices are contingency tables that display results of classification algorithms/annotations. They enables to perform error/difference analysis.

Example Two annotators A_1 and A_2 annotated 50 sentences with *cry*.

		A_2				
		1	**4**	**7**	**u**	**x**
	1	24	3	1	3	0
	4	3	3	0	1	1
A_1	**7**	0	2	4	0	1
	u	1	0	0	0	0
	x	0	1	0	0	2

What agreement would be reached by chance?

Example 1

Assume two annotators (A_1, A_2), two classes (t_1, t_2), and the following distribution:

	t_1	t_2
A_1	50 %	50 %
A_2	50 %	50 %

Then

- the best possible agreement is

What agreement would be reached by chance?

Example 1

Assume two annotators (A_1, A_2), two classes (t_1, t_2), and the following distribution:

	t_1	t_2
A_1	50 %	50 %
A_2	50 %	50 %

Then

- the best possible agreement is 100 %
- the worst possible agreement is

What agreement would be reached by chance?

Example 1

Assume two annotators (A_1, A_2), two classes (t_1, t_2), and the following distribution:

	t_1	t_2
A_1	50 %	50 %
A_2	50 %	50 %

Then

- the best possible agreement is 100 %
- the worst possible agreement is 0 %
- the "agreement-by-chance" *would be*

What agreement would be reached by chance?

Example 1

Assume two annotators (A_1, A_2), two classes (t_1, t_2), and the following distribution:

	t_1	t_2
A_1	50 %	50 %
A_2	50 %	50 %

Then

- the best possible agreement is 100 %
- the worst possible agreement is 0 %
- the "agreement-by-chance" *would be* 50 %

What agreement would be reached by chance?

Example 2

Assume two annotators (A_1, A_2), two classes (t_1, t_2), and the following distribution:

	t_1	t_2
A_1	90 %	10 %
A_2	90 %	10 %

Then

- the best possible agreement is

What agreement would be reached by chance?

Example 2

Assume two annotators (A_1, A_2), two classes (t_1, t_2), and the following distribution:

	t_1	t_2
A_1	90 %	10 %
A_2	90 %	10 %

Then

- the best possible agreement is 100 %
- the worst possible agreement is

What agreement would be reached by chance?

Example 2

Assume two annotators (A_1, A_2), two classes (t_1, t_2), and the following distribution:

	t_1	t_2
A_1	90 %	10 %
A_2	90 %	10 %

Then

- the best possible agreement is 100 %
- the worst possible agreement is 80 %
- the "agreement-by-chance" *would be*

What agreement would be reached by chance?

Example 2

Assume two annotators (A_1, A_2), two classes (t_1, t_2), and the following distribution:

	t_1	t_2
A_1	90 %	10 %
A_2	90 %	10 %

Then

- the best possible agreement is 100 %
- the worst possible agreement is 80 %
- the "agreement-by-chance" *would be* 82 %

What agreement would be reached by chance?

Example 3

Assume two annotators (A_1, A_2), two classes (t_1, t_2), and the following distribution:

	t_1	t_2
A_1	90 %	10 %
A_2	80 %	20 %

Then

- the best possible agreement is

What agreement would be reached by chance?

Example 3

Assume two annotators (A_1, A_2), two classes (t_1, t_2), and the following distribution:

	t_1	t_2
A_1	90 %	10 %
A_2	80 %	20 %

Then

- the best possible agreement is 90 %
- the worst possible agreement is

What agreement would be reached by chance?

Example 3

Assume two annotators (A_1, A_2), two classes (t_1, t_2), and the following distribution:

	t_1	t_2
A_1	90 %	10 %
A_2	80 %	20 %

Then

- the best possible agreement is 90 %
- the worst possible agreement is 70 %
- the “agreement-by-chance” *would be*

What agreement would be reached by chance?

Example 3

Assume two annotators (A_1, A_2), two classes (t_1, t_2), and the following distribution:

	t_1	t_2
A_1	90 %	10 %
A_2	80 %	20 %

Then

- the best possible agreement is 90 %
- the worst possible agreement is 70 %
- the “agreement-by-chance” *would be* 74 %

Example in R

The situation from Example 3 can be simulated in R

```
# N will be the sample size
> N = 10^6

# two annotators will annotate randomly
> A1 = sample(c(rep(1, 0.9*N), rep(0, 0.1*N)))
> A2 = sample(c(rep(1, 0.8*N), rep(0, 0.2*N)))

# percentage of their observed agreement
> mean(A1 == A2)
[1] 0.740112

# exact calculation -- just for comparison
> 0.9*0.8 + 0.1*0.2
[1] 0.74
```

Cohen's kappa

Cohen's kappa was introduced by Jacob Cohen in 1960.

$$\boxed{\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}}$$

- $\Pr(a)$ is the relative observed agreement among annotators
  $=$ percentage of agreements in the sample
- $\Pr(e)$ is the hypothetical probability of chance agreement
  $=$ probability of their agreement if they annotated randomly
- $\kappa > 0$ if the proportion of agreement obtained exceeds the proportion of agreement expected by chance

Limitations

- Cohen's kappa measures agreement between two annotators only
- for more annotators you should use Fleiss' kappa
  - see `http://en.wikipedia.org/wiki/Fleiss'_kappa`

Cohen's kappa

		A_2				
		1	**4**	**7**	**u**	**x**
	1	24	3	1	3	0
	4	3	3	0	1	1
A_1	**7**	0	2	4	0	1
	u	1	0	0	0	0
	x	0	1	0	0	2

Cohen's kappa: ?

Homework 2.1

Work with the SUBMIT data

1. Filter out uneffective features from the data using the filtering rules that we applied to the CRY data.
2. Draw a plot of the conditional entropy $\mathrm{H}(P|A)$ for the effective features. Then focus on the features for which $\mathrm{H}(P|A) \geq 0.5$. Comment what you see on the plots.

VPR vs. MOV – comparison

	MOV	VPR
type of task	regression	classification
getting examples by	collecting	annotation
# of examples	100,000	250
# of features	32	363
categorical/binary	29/18	0/363
numerical	3	0
output values	1–5	5 discrete categories

Block 2.2
Introductory remarks on VPR classifiers

VPR task – accuracy estimated by 9-fold cross-validation

All numbers are in %

method/task	VPR.cry	VPR.submit
MFC baseline	52.4	70.8
AVG Human	92.2	94.1
Best model with the provided features	80.4	90.0
Best model with additional features	84.8	93.6
|		
SIMPLE MODELS		
Single Decision Tree	61.6	86.0
SVM	73.2	86.0
Simple Logistic Regression	67.2	82.4
RESAMPLING METHODS		
simple bagging	70.8	84.4
random forest	79.6	87.2

Example Decision Tree classifier – *cry*

Trained using a cross-validation fold