Exam INF5830, 2013, some solutions

1 Accuracy and estimation (15%)

Kim is testing a classifier for entailment vs. non-entailment on a test set of 400 items. The results may be summarized in the following table.

		Test results	
		entailment	non-entailment
Reference	entailment	90	30
	non-entailment	10	270

(a) What is the accuracy of the classifier on this test set?

- The true positives are $tp = 90$
- The true negatives are $tn = 270$
- The sample size is $n = 400$
- From its definiton the accuracy equals $\frac{tp+tn}{n} = \frac{90+270}{400} = 0.9$

(b) Assume the test set is a random sample from a large population. Estimate an interval with a 95% confidence level for the accuracy of the classifier on the population.

This is a proportion with probability p. We estimate it from a sample of size $n = 400$. We have an estimate of p from the sample: $\hat{p} = 0.9$ This is a binomial distribution, but we may approximate it with a normal distibution since $np = 360 > 10$ and $n(1-p) = 36 > 10$. The formula for the interval is then

$$[\hat{p} - \frac{z^* s}{\sqrt{n}}, \hat{p} + \frac{z^* s}{\sqrt{n}}]$$

where $s^2 = \hat{p}(1-\hat{p})$. The z^*-value is the well-known 1.96. Putting this into the formula yields

$$[0.9 - \frac{1.96\sqrt{0.9 * 0.1}}{\sqrt{400}}, 0.9 + \frac{1.96\sqrt{0.9 * 0.1}}{\sqrt{400}}]$$

Rounding off 1.96 to 2 yields the more readable

$$[0.87, 0.93]$$

If anybody has used the t-distribution, that is accepted as equally good. If anybody has managed to calculate the correct interval from the binomial distribution, that is of course also fine.

2 Dependency syntax and parsing (20%)

(a) Draw the dependency graph for the sentence *A hearing is scheduled on the issue today.*, here provided in the so-called CoNLL-format:

1	A	a	DT	DT	2	NMOD
2	hearing	hearing	NN	NN	3	SBJ
3	is	be	VBZ	VBZ	0	ROOT
4	scheduled	schedule	VBD	VBD	3	VC
5	on	on	IN	IN	2	PP
6	this	this	DT	DT	7	NMOD
7	issue	issue	NN	NN	5	PCOMP
8	today	today	NN	NN	4	TMP
9	.	.	.	.	3	PUNC

(b) Choose three different dependencies in the graph above and use these to present three different criteria for syntactic head status, i.e. you should provide at least one criterion per dependency.

(i) issue - this
government: the head governs the form of the dependent, (would be 'these' if head was plural).

(ii) is - scheduled
morphosyntactic locus: the head is inflected (is scheduled, are scheduled etc.)

(iii) hearing - on
subcategorization: the head selects the dependent and determines whether it is obligatory. In the example, the head 'hearing' selects the argument headed by 'on'

(c) Nivre's arc eager algorithm operates with four parse **transitions**, two of which are parameterized by the dependency relation r: Shift, Reduce, Left-Arc$_r$, Right-Arc$_r$

(i) Show the transition sequence that results from applying the algorithm to the sentence in (a) above, at each step providing the transition employed (Shift, Reduce, Left-Arc$_r$, Right-Arc$_r$), as well as the contents of the stack and queue.

```
Initial state: [root] [A hearing is scheduled on this issue today]

Shift
[root A] [hearing is scheduled on this issue today]

LeftArc(nmod)
Initial state: [root] [hearing is scheduled on this issue today]

Shift
[root hearing] [is scheduled on this issue today]
```

```
Left-Arc(sbj)
[root] [is scheduled on this issue today]

Right-Arc(root)
[root is] [scheduled on this issue today]

Right-Arc(vc)
[root is scheduled] [on this issue today]
```

Since the Left-Arc transition pops the dependent off the stack, there is no way of connecting 'on' to 'hearing'. OK, if you stop here. Also ok if you continue, as long as you end up with a non-empty stack.

(ii) Does the algorithm terminate successfully? Why/why not?
No, the algorithm does not terminate successfully. The reason for this is that the graph contains non-projective arcs, i.e. crossing arcs or arcs that violate the constraint of projectivity: if $i \to j$, then $i \to *k$, for every node k such that $i < k < j$ or $j < k < i$. The algorithm ensures projectivity by the formulation of the Left-Arc(r) transition. Since this transition pops the stack after application, this ensures that the dependent may not have further dependents to the right (which would incur a violation of the projectivity constraint). The arc between 'hearing' and 'on' violates this constraint.

3 Semantic Role Labeling (20%)

(a) "In most languages, it is often the case that subjects correspond to agents". Comment on this statement in the light of the sentence in (1a) above. What does this tell you more generally about the relationship between syntax and semantics?
Whereas generalizations like these certainly exist, the relation between syntax and semantics is not a one-to-one mapping, due to so-called mismatches. Syntactic alternations like the passive alternation and the dative alternation offer alternative ways of expressing the same semantic content syntactically. The sentence in (1) is a passive sentence, where the object has been promoted to subject position and is therefore not a prototypical subject in this respect.

(b) Briefly describe Dowty's theory of semantic roles. What is the Argument Selection Principle and how does it account for the analysis of our example sentence in (1a)?
Proto-roles are clusters of semantic entailments determined for each predicate of which there are only two: proto-agent and proto-patient. Arguments have different degrees of membership (prototype) in a proto-role. Linking to syntactic structure is formulated in the Argument Selection Principle (ASP): the argument with the most p-a properties becomes the subject, the argument with the most p-p properties becomes the object. Proto-agent properties:

- volition
- sentience
- causes event
- movement

Proto-patient properties:

- change of state
- incremental theme
- causally affected
- stationary

For the example sentence (a hearing):

- p-a: –
- p-p: causally affected

This means that in an active version of this sentence (They scheduled a hearing on . . .) *a hearing* would be the object of the sentence, since it has the most p-p properties.

(c) We want to improve our semantic role system by including generalizations like the one expressed in (2a) above. We therefore wish to implement the Parse Tree Path feature first described in Gildea & Jurafsky (2002). Consider the simple sentence in (i) below:

(i) *The man ate cake*

Provide a phrase structure tree for this sentence and explain how to extract the path for the two arguments *man* and *cake*. Could we extract the same type of information from a dependency analysis of the same sentence? The path feature should show the path from the predicate to the argument in question:

- 'man': v $\uparrow$ vp $\uparrow$ s $\downarrow$ np $\downarrow$ n
- 'cake': v $\uparrow$ vp $\downarrow$ np $\downarrow$ n

A dependency representation does not provide phrasal categories like vp, np, so we could not extract the exact same information. Rather the syntactic relation between the predicate and the arguments would be encoded directly as dependency relations 'subj' and 'obj'. A similar path feature using a dependency structure would be:

- 'man': $\uparrow$ subj
- 'cake': $\uparrow$ obj

6 Decision trees (10%)

(The following example is of course simplified.) Kim is training an entailment classifier on 25 training items. Each item consists of a premise, P, and a hypothesis, H. The test items belong to one of two classes: Entailment or Non-entailment. Kim has decided to use two features only, whether the premise contains the word "not" and whether the hypothesis contains "not". The 25 observations are summed up in the following table.

P contains "not"	H contains "not"	class	Number of obs.
yes	yes	entailment	4
yes	no	non-entail	6
no	yes	non-entail	3
no	no	entailment	12
all other combinations			0

(a) Construct a decision tree classifier from these training data. You do not have to consider information gain or other measures to select the first feature for splitting.

(b) Evaluate the classifier on the training data. What is its accuracy, precision and recall?

We see from the tree that all training items are classified correctly, hence accuracy, recall and precision all equal 1.0.

7 Classifiers (35%)

(a) Give a short description of the main principles underlying a Naive Bayes classifier. You do not have to discuss the differences between the binomial and the multinomial approach to text classification.

Setup:

- A well-defined set of classes $C = \{c_1, c_2, \dots, c_n\}$.
- A set of features $\{f_1, f_2, \dots, f_m\}$.
- For each feature f_i there is a corresponding set of values $V_i = \{v^i_1, v^i_2, \dots v^i_{k_i}\}$.
- An observation is represented by a feature vector $\mathbf{f} = \langle f_1 = v_1, f_2 = v_2, \dots f_n = v_n \rangle$ where each v_i is one of the values in the corresponding value set V_i.
- The goal is a classifier which to each feature vector $\mathbf{f}$ assigns the correct class c from C.

Naive Bayes is based on probabilities. It assigns a probability $P(c_j \mid \mathbf{f})$ to each class c_j given $\mathbf{f}$ and selects the class for which this probability is largest, in notation:

$$\arg\max_{c \in C} P(c \mid \mathbf{f})$$

To estimate this, Bayes' formula is used:

$$P(c \mid \mathbf{f}) = \frac{P(\mathbf{f} \mid c)P(c)}{P(\mathbf{f})}$$

Since there are very many different vectors and we cannot assume to have seen them before. hence it is assumed that

$$\begin{aligned}
&(1) \quad P(\mathbf{f} \mid c) &&= P(\langle f_1 = v_1, f_2 = v_2, \dots f_n = v_n \rangle \mid c) \\
&(2) &&= P(f_1 = v_1 \mid c) \times P(f_2 = v_2 \mid c) \times \cdots \times P(f_n = v_n \mid c) \\
&(3) &&= \prod_{i=1}^{n} P(f_i = v_i \mid c)
\end{aligned}$$

This it the naive assumption which is formally only correct if the features are independent, which they in general are not.

The final expression for the Bayes'classifier can be written:

$$\arg\max_{c \in C} P(c \mid \mathbf{f}) = \arg\max_{c \in C} P(c) \prod_{i=1}^{n} P(f_i = v_i \mid c)$$

(We can remove $\mathbf{f}$ from the denominator since it is the same for all the classes and does not influence which class is argmax.)

To estimate the probabilities we use maximum likelihood estimations on the training set

$$\hat{P}(c_i) = \frac{C(c_i)}{C(o)}$$

where $C(o)$ is a count of all observations and $C(c_i)$ is the count of the observations in class c_i, and similalry

$$\hat{P}(f_j = v_j \mid c_i) = \frac{C(f_j = v_j, c_i)}{C(c_i)}$$

where $C(f_j = v_j, c_i)$ is a count of all observations from class c_i where f_j takes the value v_j.

(b) Kim is training a Naive Bayes classifier on the same training data as in exercise (6). How will this classifier classify an observation where H contains "not", while P does not contain "not"? State reasons for your answer.

- There are two classes: *ent,non*
- There are two features f_1, f_2 which each may take one of two values: y, n
- $f_1 = y$ if P contains "not" and $f_1 = n$ if P does not contain "not"
- $f_2 = y$ if H contains "not" and $f_2 = n$ if H does not contain "not"

We shall determine the class for an observation $\langle f_1 = n, f_2 = y \rangle$. From point (a), we must compare

$P(\text{ent}) * P(f_1 = n \mid \text{ent}) * P(f_2 = y \mid \text{ent})$ to

$P(\text{non}) * P(f_1 = n \mid \text{non}) * P(f_2 = y \mid \text{non})$

We use maximum likelihood to estimate the probabilities:

$$P(\text{ent}) = \frac{C(\text{ent})}{n} = \frac{4+12}{25} = 16/25$$

$$P(\text{non}) = \frac{C(\text{non})}{n} = \frac{6+3}{25} = 9/25$$

$$P(f_1 = n \mid \text{ent}) = \frac{C(f_1 = n, \text{ent})}{C(\text{ent})} = \frac{12}{4+12} = 3/4$$

$$P(f_2 = y \mid \text{ent}) = \frac{C(f_2 = y, \text{ent})}{C(\text{ent})} = \frac{4}{4+12} = 1/4$$

$$P(f_1 = n \mid \text{non}) = \frac{C(f_1 = n, \text{non})}{C(\text{non})} = \frac{3}{6+3} = 1/3$$

$$P(f_2 = y \mid \text{non}) = \frac{C(f_2 = y, \text{non})}{C(\text{non})} = \frac{3}{6+3} = 1/3$$

We may fill in the formulas and see that

$$P(\text{ent}) * P(f_1 = n \mid \text{ent}) * P(f_2 = y \mid \text{ent}) = 16/25 * 3/4 * 1/4 = 3/25$$

$$P(\text{non}) * P(f_1 = n \mid \text{non}) * P(f_2 = y \mid \text{non}) = 9/25 * 1/3 * 1/3 = 1/25$$

Hence, the classifier will wrongly classify this item as entailment.

(c) Given the training data, will you say that this task is linearly separable? State reasons for your answer.

The features are categorical, but since they take two values, they may be considered numerical where we set 0 for n and 1 for y. The observations are then in a two-dimensional vector space, a plane, which we may "draw" like:

H contains "no"	1	3 non	4 ent
	0	12 ent	6 non
		0	1
		P contains "no"	

We see that it is impossible to draw a line which put the entailment points (0,0) and (1,1) on one side and the non-entailment points (0,1) and (1,0) on the other side.

(Observation, not part of the exercise. The Naive Bayes classificator will misclassify two of the four groups, it will misclassify 7 observations. It is possible to make a linear classifier which only misclassifies one group of 3 items, but not a correct linear classifier.)

END