Tutorial 4 – Hemanth Kumar Battula

Part I: Lexical semantics

Q1: WordNet

Consider this sentence:

Swedes like to fish for sea bass.

Q1a - Using WordNet, determine how many senses there are for each of the open-class words in this sentence. How many distinct combinations of senses are there for this sentence?

Sol

Word	Number of senses
Swedes	2
Like	5
To	1
Fish	4
For	1
Sea	3
Bass	8

Total distinct combination of senses= 960 (Product of all senses)

Q1b - Now tag each open-class word in the sentence with its correct tag (WordNet sense number). Was choosing the correct sense always a straightforward task? Report on any difficulties you encountered. **Note:** You will have to select *show sense numbers* in the display dropdown.

SOL: swedes- #1 (Noun)

Like- #1 (verb)

To-

Fish- #2 (verb)

For -

Sea- #1 (Noun)

Bass- #4 (Noun)

Q2: FrameNet

Consider the following sentence:

Eva baked some saffron buns for Tom using her new oven on Santa Lucia (13th December)

Q2a - Search the Berkeley FrameNet for a suitable frame which you think fits this situation. Report the name of the frame you chose, listing all of the frame elements which you think apply to this sentence and the parts of the sentence they correspond to.

Cooking_creation

Definition:

This frame describes food and meal preparation. A Cook creates a Produced_food from (raw) Ingredients. The Heating_instrument and/or the Container may also be specified.

Eva baked some saffron buns for Tom using her new oven on Santa Lucia (13th December)

FEs:

Core:

Cook :Eva

Produced_food :Saffron buns

Non-Core:

Heating_instrument [Heat_instr] :Oven

Ingredients [Ingr] :Saffron

Purpose [Purp] : *Santa Lucia*

Recipient [Rec]: Tom

Time [Time] : *13th December*

Q2a - Rewrite the sentence with the same frame and core elements, but three new non-core elements.

Sol: *Eva baked some saffron buns using **ingredients** like egg, cinnamon in her kitchen **(place)** for Tom using her new oven on Santa Lucia (13th December) and gave him a surprise by wrapping the buns in a gift paper using a tray. **(Container)***

Part II: Text classification

Q3. Naïve Bayes WSD:

[See: lecture 9, p.23]

Note: You should be able to do this exercise by hand. If you use code, make sure it's clear how you got each answer.

Let's build a word sense disambiguation model for the word cricket.

We have the following two senses (the prior probability is given):

	sense	**prior probability**
v1	cricket#1	P(v1) = 0.4
v2	cricket#2	p(v2) = 0.6

We will use the following words as features in a bag-of-words representation of context.

	word
a1	score
a2	game
a3	Bat
a4	insect

And we are given the following training data:

a1	a2	a3	a4	Label
1	1	0	0	v2
0	0	1	0	v2
1	1	0	0	v2

1	0	1	1	v2
0	0	1	0	v1
0	0	0	1	v1
1	0	0	1	v1
0	0	0	1	v1

Q3a - For each ai and vj, compute the conditional probabilities P(ai=0 | vj) and P(ai=1 | vj):

Attribute	value	v1	v2
a1=score	0	¾=0.75	¼=0.25
	1	0.25	0.75
a2=game	0	1	0.50
	1	0	0.50
a3=bat	0	0.75	0.50
	1	0.25	0.50
a4=insect	0	0.25	0.75
	1	0.75	0.25

Here is our testing data:

1. the main object of **cricket#2** is to score the most runs

   (cricket#2 * (score|cricket#2)* (game|cricket#2)* (bat|cricket#2)* (insect|cricket#2)) = 0.6 * 0.75*0.50*0.50*0.75 = 0.0843

   (cricket#1 * (score|cricket#1)* (game|cricket#1)* (bat|cricket#1)* (insect|cricket#1)) = 0.4 * 0.25 *1*0.75*0.25= 0.018

   Cricket#2 is correct guess

2. the early form of **cricket#2** differed from the modern game in key aspects

   (cricket#2 * (game|cricket#2)* (score|cricket#2)* (bat|cricket#2)* (insect|cricket#2)) = 0.6 * 0.50*0.75*0.50*0.75 = 0.3

   (cricket#1 * (game|cricket#1)*)* (score|cricket#1)* (bat|cricket#1)* (insect|cricket#1)) = 0.4 * 0*0.75*0.75*0.25 = 0

   Cricket#2 is correct guess

3. **cricket#2** is a bat and ball game played between two teams of eleven players

   cricket#2 * (score|cricket#2)* (game|cricket#2)* (bat|cricket#2)* (insect|cricket#2)= 0.6 *0.25* 0.50 * 0.50*0.75 = 0.0281

   cricket#1 * (score|cricket#1)* (game|cricket#1)* (bat|cricket#1)* (insect|cricket#1)= 0.4 * 0.75*0.25 * 0*0.25 = 0

   cricket#2 is correct guess

4. in **cricket#2** you score runs by striking the ball bowled at the wicket with the bat

(cricket#2 * (score|cricket#2)* (game|cricket#2)* (bat|cricket#2)* (insect|cricket#2)) = 0.6 * 0.75 * 0.50 *0.50*0.75= 0.0843

(cricket#1 * (score|cricket#1)* (game|cricket#1)* (bat|cricket#1)* (insect|cricket#1)) = 0.4 * 0.25 *1* 0.25*0.25 = 0.006

Cricket#2 is correct guess

5. a bat is a flying mammal that eat insects such the **cricket#1**

   (cricket#2 *(score|cricket#2)* (game|cricket#2)* (bat|cricket#2)* (insect|cricket#2)**) = 0.6 *0.25*0.50* 0.50 * 0.25 = 0.0093**

   (cricket#1 * (score|cricket#1)* (game|cricket#1)* (bat|cricket#1)* (insect|cricket#1)**) = 0.4 * 0.75*1*0.25 * 0.75 = 0.056**

   Cricket#1 wrong guess

6. in international **cricket**#2 the game is adjudicated by three umpires

   ((cricket#2 ***(score|cricket#2)* (game|cricket#2)* (bat|cricket#2)* (insect|cricket#2)) = 0.6 *0.25* 0.50*0.50*0.75 = 0.0281

   (cricket#1 * (score|cricket#1)* (game|cricket#1)* (bat|cricket#1)* (insect|cricket#1)) = 0.4 * 0 = 0

   Cricket#2 is correct guess

7. **cricket#1**

cricket#2 has more priority than the cricket#1. Cricket#1 is a wrong guess

Q3b - For each sentence in the testing data, use the Baysian model defined by the *prior probabilities* for vj (given) and the *conditional probabilites* for aj (part a).

Q3c - Complete the confusion matrix for **cricket#2** [**note**: In this case a *negative* means guessing cricket#1]**.** Compute precision, recall, and F-score.

True + Sentence1 Sentence2' Sentence3 Sentence4 Sentence6	False + Sentence5 Sentence7
True -	False –

Recall= True positives/(True positives+False Negative)= 5/5= 1

Precision= True positive/(True Positive+False positive)=5/7=0.714

F-score=(2*((precision*recall)/(Precision+recall)))=2*((0.714*1)/(0.714+1))=0.833

Q4. If you have a hammer...

Choose two of the following tasks, and explain how you would model them as a classification task.

- Diacritic restoration (e.g. *cote* to *côté, alska* to *älska*)
- **Language identification**
- Lexical selection in machine translation
- **Sentiment analysis**
- Spam filtering

No need to write code, just explain on a high level: **(a)** what features are relevant, **(b)** if you think a certain classifier would be particularly suitable (or unsuitable) for the task, and **(c)** what kind of training data you would need.

Language Identification:

(a)Feature Selection:

The solution of this problem is to use the features either words, sub words, stop words, multi lingual document or ngrams. Using sub word features would enhance the classifier by taking the structure of words into account. The effective way of incorporating such information is to divide each word

represented by the set of all character ngrams of a given length appearing in that word. ({Hemanth,hem,ema,man,ant,nth})

(b) if you think a certain classifier would be particularly suitable (or unsuitable) for the task

An approach with n-grams is suitable for language detection. N-grams should only be constructed based on underlying bytes that makes up the text. This simply means that we do not have to do any character detection, like determining which Unicode maps to which character or any special dealing with punctuation marks. This is a major advantage as it can be applied to any language. It will also accept misspelled words.

(c) what kind of training data you would need.

The data can be any kind of text when using a n-gram model, because we eliminate all kind of stop words, noisy text or punctuations from the text and we deal only with the data bytes. Training the n-gram model Using multiple documents of different languages would help identify new texts easily.

Sentiment analysis

(a) what features are relevant

A good classifier for sentiment analysis should be able to analyze polarity of the opinion like positive, negative, neutral, Emotion detection, Aspect-based analysis , intent analysis etc. Sometimes not only a single sentence or few words determine the sentiment of a text or document but the discourse is also very important. For polarity of opinion its important to consider bag of words which classify the sentiments.

A

(b) if you think a certain classifier would be particularly suitable (or unsuitable) for the task, and'

If we use N-gram approach, we can take into account all the preceding and following words which would provide better information. Sometimes a positive word can be preceded with a negated meaning or irony which would be difficult to identify easily. A5-gram model would be a good working model.

Also probabilistic algorithms like Naïve Bayes approaches which classifies text based on bag of words and their probabilities of occurrences in a sentence to determine the polarity would be helpful.

(c) what kind of training data you would need.

For considering like Naïve Bayes approach a list of bag of words which use a large data set trained with previous opinions, reviews , contexts, are used to classify new texts or documents.