Natural Language Processing

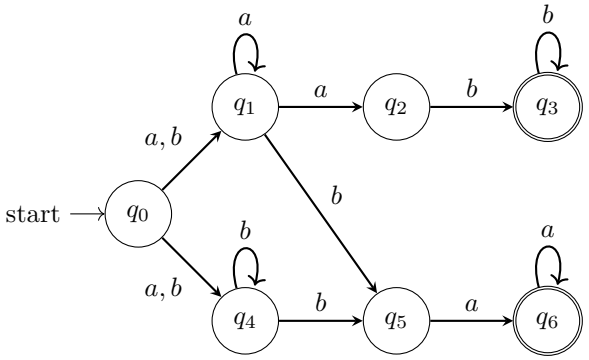
Assignment 1

September 23, 2024 (small update: September 26, 2024)

This is the first hand-in assignment for Natural Language Processing, and it counts towards 15% of your final grade. The pen& paper assignments constitute 40% of the assignment grade, the other 60% are for the Python assignments. You may work in pairs (i.e. at most two students); hand in the assignment as a single zip file on Brightspace.

Assignment 1: Finite State Automata (9 points)

A non-deterministic finite state automaton is a finite state automaton where different transitions between states can be allowed for a single symbol. Given the non-deterministic nature of such an automaton, there may be *ambiguity*, i.e. one input word may be accepted by multiple paths through the automaton. An example is given below:



The assignment

1. (*2 points*) Give two sequences of states (aka a path) for recognizing the string sequence *aba*.
2. (*3 points*) Give a symbol sequence that is recognized by the above automaton but in only one possible way.
3. (*4 points*) What is the regular expression that corresponds to this automaton?

Assignment 2: N-grams and smoothing (11 points)

Given is a very simple corpus with counts as below:

Unigram counts:

<s>	the	lecture	was	fun	</s>
20	9	5	8	7	14

Bigram counts:

	<s>	the	lecture	was	fun	</s>
<s>	0	8	3	6	2	1
the	0	0	6	0	3	0
lecture	0	0	0	4	0	1
was	0	4	0	0	4	0
fun	0	0	1	2	0	4
</s>	14	0	0	0	0	0

Additionally, we have the following counts:

$$C(\text{<s> the lecture}) = 5$$
$$C(\text{the lecture was}) = 2$$
$$C(\text{lecture was fun}) = 3$$
$$C(\text{was fun </s>}) = 1$$

The assignment

1. (*2 points*) Calculate the bigram probability $P(\text{was} \mid \text{lecture})$.
2. (*3 points*) Calculate the add-2 smoothed trigram probability $P(\text{fun} \mid \text{lecture was})$.
3. (*3 points*) Calculate the sequence probability of $P(\text{<s> the lecture was fun </s>})$, given a bigram model.
4. (*3 points*) Calculate the sequence probability of $P(\text{<s> the lecture was fun </s>})$, given a trigram model.

Assignment 3: Part-of-speech tagging (10 points)

Recall the part-of-speech tagset from the universal dependencies project, (different from the tagset for the Penn Treebank!)

	Tag	Description	Example
Open Class	**ADJ**	Adjective: noun modifiers describing properties	*red*, *young*, *awesome*
	ADV	Adverb: verb modifiers of time, place, manner	*very*, *slowly*, *home*, *yesterday*
	NOUN	words for persons, places, things, etc.	*algorithm*, *cat*, *mango*, *beauty*
	VERB	words for actions and processes	*draw*, *provide*, *go*
	PROPN	Proper noun: name of a person, organization, place, etc..	*Regina*, *IBM*, *Colorado*
	INTJ	Interjection: exclamation, greeting, yes/no response, etc.	*oh*, *um*, *yes*, *hello*
Closed Class Words	**ADP**	Adposition (Preposition/Postposition): marks a noun's spacial, temporal, or other relation	*in, on, by, under*
	AUX	Auxiliary: helping verb marking tense, aspect, mood, etc.,	*can, may, should, are*
	CCONJ	Coordinating Conjunction: joins two phrases/clauses	*and*, *or*, *but*
	DET	Determiner: marks noun phrase properties	*a, an, the, this*
	NUM	Numeral	*one, two, first, second*
	PART	Particle: a function word that must be associated with another word	*'s, not, (infinitive) to*
	PRON	Pronoun: a shorthand for referring to an entity or event	*she, who, I, others*
	SCONJ	Subordinating Conjunction: joins a main clause with a subordinate clause such as a sentential complement	*that*, *which*
Other	**PUNCT**	Punctuation	; , ()
	SYM	Symbols like $ or emoji	$, %
	X	Other	asdf, qwfg

The assignment Use the tags above to give the tag sequences for the sentences below:

1. (*2 points*) `I love Utrecht !`
2. (*4 points*) `Their successor may suffer a breakdown .`
3. (*4 points*) `Hello , is this the droid you are looking for ?`

Assignment 4: Constituency & dependencies (10 points)

The following grammar is given:

Non-terminal rules

S ⟶ NP VP
VP ⟶ ITV | TV NP | DTV NP NP | VP PP
NP ⟶ Det NN | NP PP | NNS | NN | Pron
PP ⟶ Prep NP

Terminal rules

ITV ⟶ sleep | eat
TV ⟶ eat | repeat | write
DTV ⟶ show | give | promise
Det ⟶ a | the
NN ⟶ rice | joy | grammar
NNS ⟶ llamas | lectures | exercises
Pron ⟶ I | you | me
Prep ⟶ to | on | without

The assignment

1. (*5 points*) Give the corresponding constituency tree for the following sentence:

   Exercises without grammar give me joy

2. (*5 points*) Now add the dependency annotation to your constituency tree above. You can ignore unary branches (i.e. a node that only refers to a single leaf does not require a dependency label). Just like in the exercises, use the *mod* label for prepositional phrases, consider the preposition the head of the phrase, and the noun (phrase) complement to be the prepositional object (*pobj*).

Assignment 5, 6, 7 (60 points)

See the Python notebook.