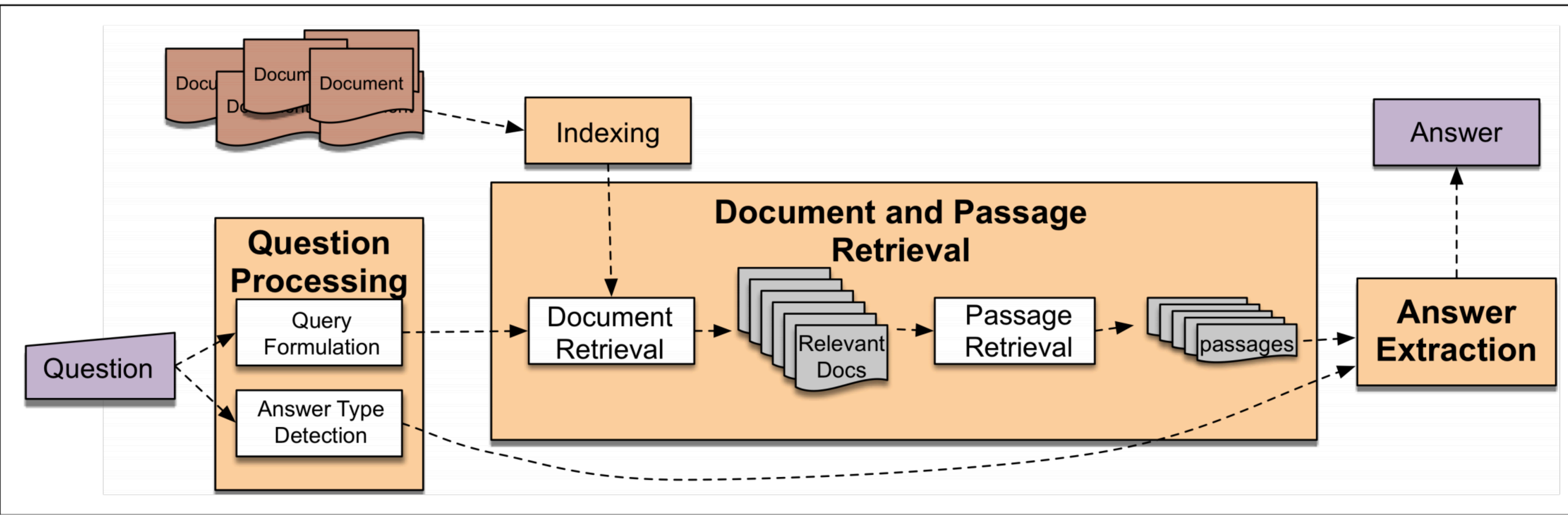
COMP90042
Web search and text analysis

Workshop Week 9

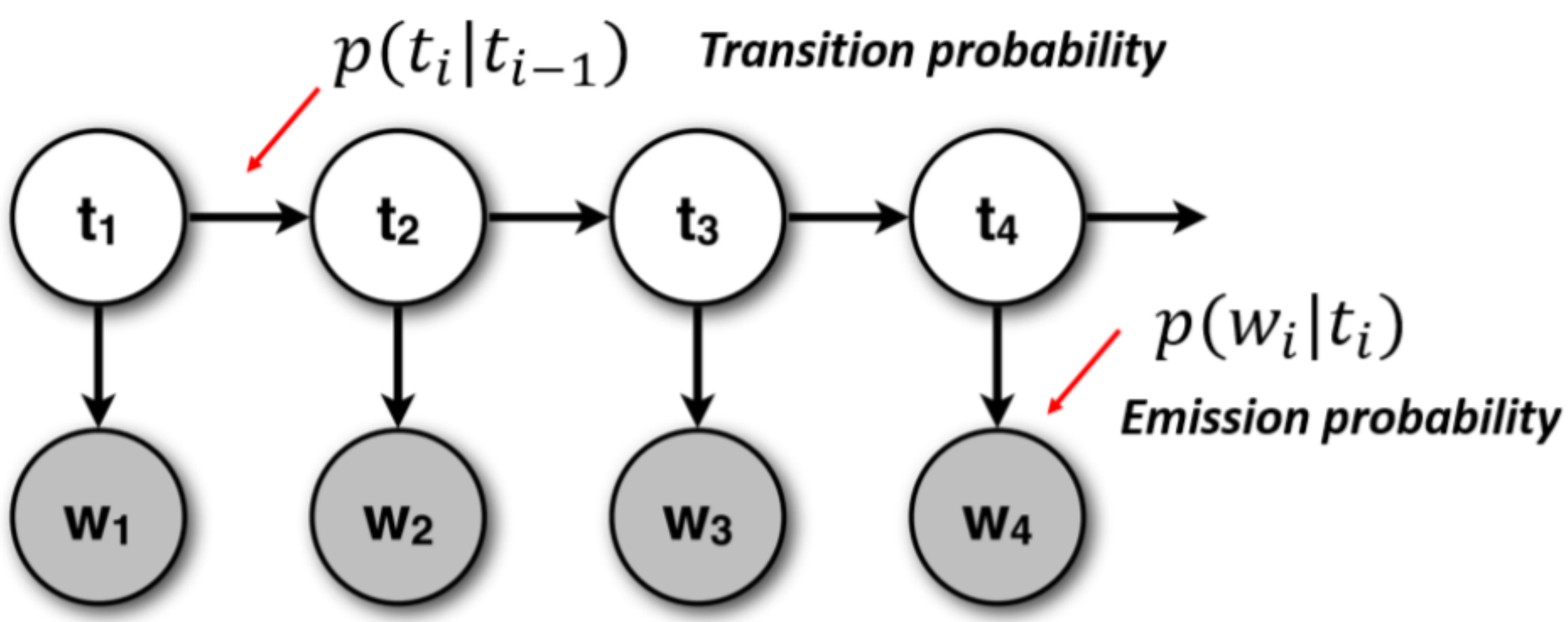
xudong.han@unimelb.edu.au
https://github.com/HanXudong/COMP90042_Workshops

What is Question Answering, and how is it related to Information Retrieval and Information Extraction?



Q1 What are the assumption that go into a Hidden Markov Model? What is the time complexity of the Viterbi algorithm? Is this practical?

- Markov assumption
- Output independence assumption



Q1b

- silver-JJ wheels-NNS turn-VBP
- wheels-NNS turn-VBP right-JJ
- right-JJ wheels-NNS turn-VBP

Estimate initial state probabilities π

Q1b

- silver-JJ wheels-NNS turn-VBP
- wheels-NNS turn-VBP right-JJ
- right-JJ wheels-NNS turn-VBP

Estimate initial state probabilities π

$$\pi[JJ, NNS, VBP] = [\frac{2}{3}, \frac{1}{3}, 0]$$

Q1b

- silver-JJ   wheels-NNS   turn-VBP
- wheels-NNS   turn-VBP   right-JJ
- right-JJ   wheels-NNS   turn-VBP

Estimate transition probability A

	JJ	**NNS**	**VBP**
JJ			
NNS			
VBP			

Q1b

- silver-JJ wheels-NNS turn-VBP
- wheels-NNS turn-VBP right-JJ
- right-JJ wheels-NNS turn-VBP

Estimate transition probability A

	JJ	**NNS**	**VBP**
JJ	0	2	0
NNS	0	0	3
VBP	1	0	0

Q1b

- silver-JJ wheels-NNS turn-VBP
- wheels-NNS turn-VBP right-JJ
- right-JJ wheels-NNS turn-VBP

Estimate transition probability A

	JJ	**NNS**	**VBP**
JJ	0	1	0
NNS	0	0	1
VBP	1	0	0

Q1b

- silver-JJ wheels-NNS turn-VBP
- wheels-NNS turn-VBP right-JJ
- right-JJ wheels-NNS turn-VBP

Estimate emission probability

	right	**Silver**	**turn**	**wheels**
JJ				
NNS				
VBP				

Q1b

- silver-JJ wheels-NNS turn-VBP
- wheels-NNS turn-VBP right-JJ
- right-JJ wheels-NNS turn-VBP

Estimate emission probability

	right	**Silver**	**turn**	**wheels**
JJ	2/3	1/3	0	0
NNS	0	0	0	1
VBP	0	0	1	0

Q2

$$\pi[\text{JJ}, \text{NNS}, \text{VBP}] = [0.3, 0.4, 0.3]$$

A	JJ	NNS	VBP	B	silver	wheels	turn
JJ	0.4	0.5	0.1	JJ	0.8	0.1	0.1
NNS	0.1	0.4	0.5	NNS	0.3	0.4	0.3
VBP	0.4	0.5	0.1	VBP	0.1	0.3	0.6

silver wheels turn

Q2

$\pi[\text{JJ}, \text{NNS}, \text{VBP}] = [0.3, 0.4, 0.3]$

A	JJ	NNS	VBP	B	silver	wheels	turn
JJ	0.4	0.5	0.1	JJ	0.8	0.1	0.1
NNS	0.1	0.4	0.5	NNS	0.3	0.4	0.3
VBP	0.4	0.5	0.1	VBP	0.1	0.3	0.6

	silver	**wheels**	**turn**
JJ	**π**[JJ] * B[JJ, silver] = P1	P1 * A[JJ, JJ] * B[JJ, wheels] = P1_1 P2 * A[NNS, JJ] * B[JJ, wheels] = P1_2 P3 * A[VBP, JJ] * B[JJ, wheels] = P1_3 (Pick the largest)	
NNS	P2		
VBP	P3		

α	*1:silver*	*2:wheels*	*3:turn*	
JJ:	0.24	0.0096	JJ $\rightarrow$ JJ	A[JJ,JJ]B[JJ, turn]
		JJ $\rightarrow$ JJ	0.0096	$\times 0.4 \times 0.1 = 0.000384$
			NNS $\rightarrow$ JJ	A[NNS,JJ]B[JJ, turn]
			0.048	$\times 0.1 \times 0.1 = 0.00048$
			VBP $\rightarrow$ JJ	A[VBP,JJ]B[JJ, turn]
			0.018	$\times 0.4 \times 0.1 = \mathbf{0.00072}$
NNS:	0.12	0.048	JJ $\rightarrow$ NNS	A[JJ,NNS]B[NNS, turn]
		JJ $\rightarrow$ NNS	0.0096	$\times 0.5 \times 0.3 = 0.00144$
			NNS $\rightarrow$ NNS	A[NNS,NNS]B[NNS, turn]
			0.048	$\times 0.4 \times 0.3 = \mathbf{0.00576}$
			VBP $\rightarrow$ NNS	A[VBP,NNS]B[NNS, turn]
			0.018	$\times 0.5 \times 0.3 = 0.0027$
VBP:	0.03	0.018	JJ $\rightarrow$ VBP	A[JJ,VBP]B[VBP, turn]
		NNS $\rightarrow$ VBP	0.0096	$\times 0.1 \times 0.6 = 0.000576$
			NNS $\rightarrow$ VBP	A[NNS,VBP]B[VBP, turn]
			0.048	$\times 0.5 \times 0.6 = \mathbf{0.0144}$
			VBP $\rightarrow$ VBP	A[VBP,VBP]B[VBP, turn]
			0.018	$\times 0.1 \times 0.6 = 0.00108$

Q3 What are regular grammar and regular language? How are they different?

- A language is a set of acceptable strings and a grammar is a generative description of a language.

- Regular language is a formal language that can be expressed using a regular expression.

- Regular grammar is a formal grammar defined by a set of production rules in the form of A -> xB, A - x and A->ϵ, where A and B are non-terminals, X is a terminal and ϵ is the empty string.

- A language is regular if and only if it can be generated by a regular grammar.

Q3

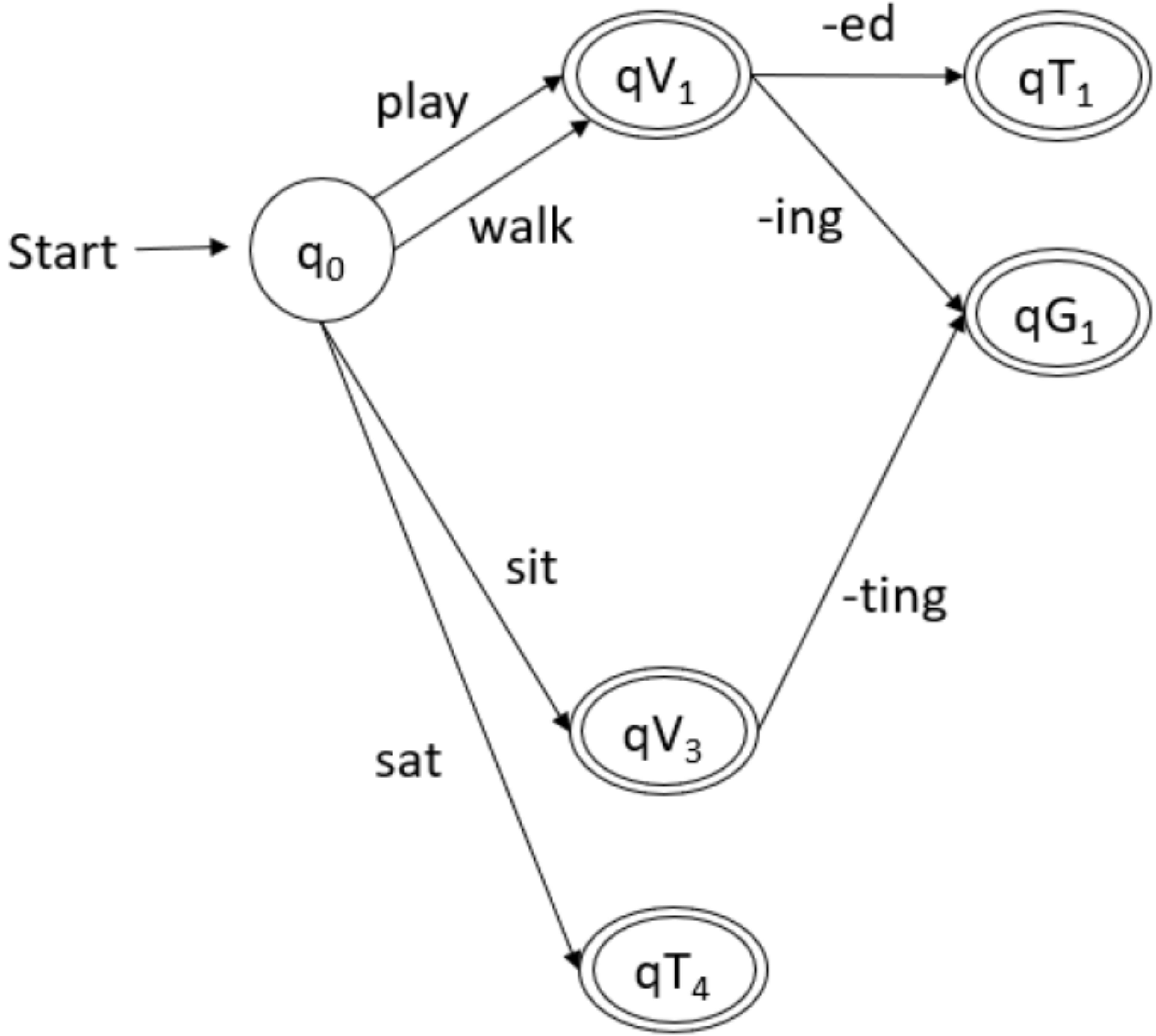
- For example: a simple regular grammar
- Rules: S -> A, A -> aA, A->ϵ
- S is the start symbol
- It will generate words such as a, aa, aaa, aaa
- The set of words generated by this regular grammar is a regular language
- This regular language can also be expressed in regular expression a*

Q3a

Regular languages are closed under union, intersection and concatenation.

What does it mean? Why is it important?

- This means that if L1 and L2 are two regular languages, then L1 ∪ L2, L1 ∩ L2, and the language strings that are the concatenation of L1 and L2 are also regular languages.
- This closure property allows us to apply operations on regular languages to produce a regular language.
- This allows for NLP problems to be factored into small simple parts, such that we can develop regular languages for each part, and combine them into a complex system to handle the NLP problems. This is particularly relevant for transducers and the composition operation, which are used in many NLP pipelines. (Note that FSTs implement "regular relations" rather that regular languages, but the distinction is a bit subtle and is not something we will dive into.)


Start
q0
play
walk
qV1
-ed
qT1
-ing
qG1
sit
sat
qV3
-ting
qT4

Q3-c
What are Weighted Finite State Acceptors(WFSAs)? When and Why are they useful?

- WFSAs are generalisation of FSAs, with each path assigned a score, computed from the transitions, the initial state, and the final state. The total score for any path is equal to the sum of the scores of the initial state, the transitions, and the final state.

- WFSAs can produce a score for each valid word for sub-word decomposition problems or sentence for words-in-sentence problems, while FSAs have no way to express preference among technically valid words or sentences

Q3-c
What are Weighted Finite State Acceptors(WFSAs)? When and Why are they useful?

- For example, WFSAs can assign scores to all strings of characters forming words, so that spelling mistakes, new words, or strange-but-acceptable words can still be handled.

- The same argument holds for sequences of words forming sentences. Clearly some word sequences are gibberish, but being able to provide a numerical score can help in many application, like how Los can be used in sentence generation, speech recognition, OCR, translation.