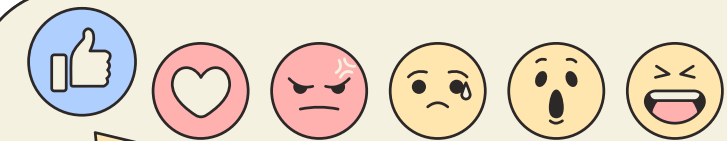


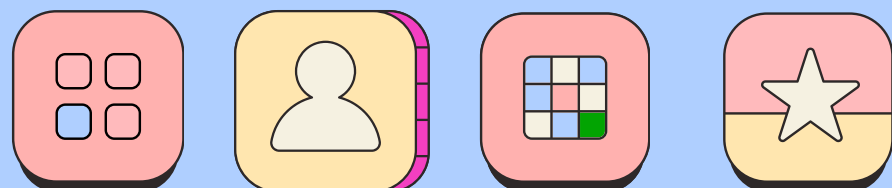
LOG IN 

ATTENTION!

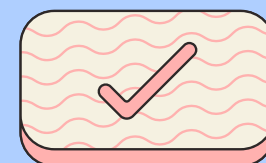


सस्ताGPT

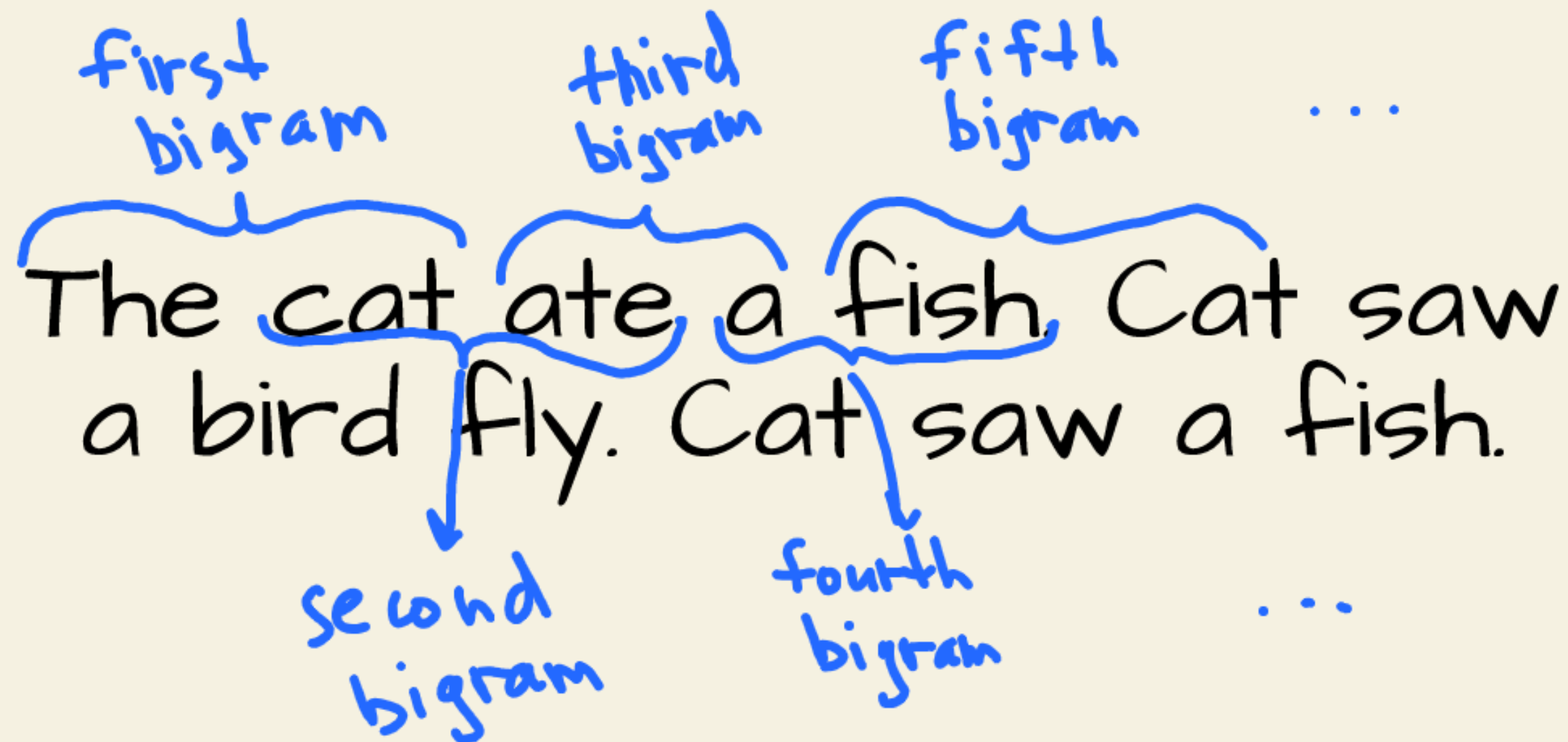
AMAN K. FOUJDAR



IML: PROF SANDEEP JUNEJA



BIGRAM MODELS



BIGRAM MODELS

0	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
4510	1306	1542	1690	1531	417	669	874	591	2422	2963	1572	2538	1146	394	515	92	1639	2055	1308	78	376	307	134	535	929	
aa	ab	ac	ad	ae	af	ag	ah	ai	aj	ak	al	am	an	ao	ap	aq	ar	as	at	au	av	aw	ax	ay	az	
556	541	470	1042	692	134	168	2332	1650	175	568	2528	1634	3438	63	82	60	3264	1118	687	381	834	161	182	2050	435	
ba	bb	bc	bd	be	bf	bg	bh	bi	bj	bk	bl	bm	bn	bo	bp	bq	br	bs	bt	bu	bv	bw	bx	by	bz	
321	38	1	65	655	0	0	41	217	1	0	103	0	4	105	0	0	842	8	2	45	0	0	0	83	0	
ca	cb	cc	cd	ce	cf	cg	ch	ci	cj	ck	cl	cm	cn	co	cp	cq	cr	cs	ct	cu	cv	cw	cx	cy	cz	
815	0	42	1	551	0	2	664	271	3	316	116	0	0	380	1	11	76	5	35	35	0	0	3	104	4	
da	db	dc	dd	de	df	dg	dh	di	dj	dk	dl	dm	dn	do	dp	dq	dr	ds	dt	du	dv	dw	dx	dy	dz	
1303	1	3	149	1283	5	25	118	674	9	3	60	30	31	378	0	1	424	29	4	92	17	23	0	317	1	
ea	eb	ec	ed	ee	ef	eg	eh	ei	ej	ek	el	em	en	eo	ep	eq	er	es	et	eu	ev	ew	ex	ey	ez	
679	121	153	384	1271	82	125	152	818	55	178	3248	769	2675	269	83	14	1958	861	580	69	463	50	132	1070	181	
fa	fb	fc	fd	fe	ff	fg	fh	fi	fl	fk	20	fm	fn	fo	fp	fq	fr	fs	ft	fu	fv	fw	fx	fy	fz	
242	0	0	0	123	44	1	1	160	0	2	20	0	4	60	0	0	114	6	18	10	0	4	0	14	2	
ga	gb	gc	gd	ge	gf	gg	gh	gi	gj	gk	gl	gm	gn	go	gp	gq	gr	gs	gt	gu	gv	gw	gx	gy	gz	
350	0	0	19	334	0	2	360	190	3	0	32	6	27	83	0	0	201	50	31	85	0	26	0	31	1	
ha	hb	hc	hd	he	hf	hg	hh	hi	hj	hk	hl	hm	hn	ho	hp	hq	hr	hs	ht	hu	hv	hw	hx	hy	hz	
2244	8	2	24	674	2	2	1	729	9	29	185	117	138	287	1	1	204	31	71	166	39	10	0	213	20	
ia	ib	ic	id	ie	if	ig	ih	ii	ij	ik	il	im	in	io	ip	iq	ir	is	it	iu	iv	iw	ix	iy	iz	
2445	110	509	440	1653	101	428	95	82	76	445	1345	427	2126	588	53	52	849	1316	541	109	269	8	89	779	277	
ja	jb	jc	jd	je	jf	ig	ih	ii	ij	ik	il	im	in	io	ip	iq	ir	is	it	iu	iv	iw	ix	iy	iz	
1473	1	4	4	440	0	0	45	119	2	2	9	5	2	479	1	0	11	7	2	202	5	6	0	10	0	
ka	kb	kc	kd	ke	kf	kg	kh	ki	kj	kk	kl	km	kn	ko	kp	kq	kr	ks	kt	ku	kv	kw	kx	ky	kz	
1731	2	2	2	895	1	0	307	509	2	20	139	9	26	344	0	0	109	95	17	50	2	34	0	379	2	
la	lb	lc	ld	le	lf	lg	lh	li	lj	lk	ll	lm	ln	lo	lp	lq	lr	ls	lt	lu	lv	lw	lx	ly	lz	
2623	52	25	138	2921	22	6	19	2480	6	24	1345	60	14	692	15	3	18	94	77	324	72	16	0	1588	10	
ma	mb	mc	md	me	mf	mg	mh	mi	mj	mk	ml	mn	mo	mp	mq	mr	ms	mt	mu	mv	mw	mx	my	mz		
2590	112	51	24	818	1	0	5	1256	7	1	5	168	20	452	38	0	97	35	4	139	3	2	0	287	11	
na	nb	nc	nd	ne	nf	ng	nh	ni	nj	nk	nl	nm	no	np	nq	nr	ns	nt	nu	nv	nw	nx	ny	nz		
2977	8	213	704	1359	11	273	26	1725	44	58	195	19	1906	496	5	2	278	443	96	55	11	6	465	145		
oa	ob	oc	od	oe	of	og	oh	oi	oj	ok	ol	om	on	oo	op	oq	or	os	ot	ou	ov	ow	ox	oy	oz	
149	140	114	190	132	34	44	171	69	16	68	619	261	2411	115	95	3	1059	504	118	275	176	114	45	103	54	
pa	pb	pc	pd	pe	pf	pg	ph	pi	pj	pk	pl	pm	pn	po	pp	pq	pr	ps	pt	pu	pv	pw	px	py	pz	
209	2	1	0	197	1	0	204	61	1	1	16	1	1	59	39	0	151	16	17	4	0	0	0	12	0	
qa	qb	qc	qd	qe	qf	qg	qh	qi	qj	qk	ql	qm	qn	qo	qp	qq	qr	qs	qt	qu	qv	qw	qx	qy	qz	
13	0	0	0	1	0	0	0	13	0	0	1	2	0	2	0	0	1	2	0	206	0	3	0	0	0	
ra	rb	rc	rd	re	rf	rg	rh	ri	rj	rk	rl	rm	rn	ro	rp	rq	rr	rs	rt	ru	rv	rw	rx	ry	rz	
2356	41	99	187	1697	9	76	121	3033	25	90	413	162	140	869	14	16	425	190	208	252	80	21	3	773	23	
sa	sb	sc	sd	se	sf	sg	sh	si	sj	sk	sl	sm	sn	so	sp	sq	sr	ss	st	su	sv	sw	sx	sy	sz	
1201	21	60	9	884	2	2	1285	684	2	82	279	90	24	531	51	1	55	461	765	185	14	24	0	215	10	
ta	tb	tc	td	te	tf	tg	th	ti	tj	tk	tl	tm	tn	to	tp	tq	tr	ts	tt	tu	tv	tw	tx	ty	tz	
1027	1	17	0	716	2	2	647	532	3	0	134	0	22	667	0	0	352	35	374	78	15	11	2	341	105	
ua	ub	uc	ud	ue	uf	ug	uh	ui	uj	uk	ul	um	un	uo	up	uq	ur	us	ut	uu	uv	uw	ux	uy	uz	
163	103	103	136	169	19	47	58	121	14	93	301	154	275	10	16	10	414	474	82	3	37	86	34	13	45	
va	vb	vc	vd	ve	vf	vg	vh	vi	vj	vk	vl	vm	vn	vo	vp	vq	vr	vs	vt	vu	vv	vw	vx	vy	vz	
642	1	0	1	568	0	0	1	911	0	3	14	0	8	153	0	0	48	0	0	7	7	0	0	121	0	
wa	wb	wc	wd	we	wf	wg	wh	wi	wj	wk	wl	wm	wn	wo	wp	wq	wr	ws	wt	wu	wv	ww	wx	wy	wz	
280	1	0	8	149	2	1	23	148	0	6	13	6	58	36	0	0	22	20	8	25	0	2	0	73	1	
xa	xb	xc	xd	xe	xf	xg	xh	xi	xj	xk	xl	xm	xn	xo	xp	xq	xr	xs	xt	xu	xv	xw	xx	xy	xz	
103	1	4	5	36	3	0	1	102	0	0	39	1	1	41	0	0	0	31	70	5	0	3	38	30	19	
ya	yb	yc	yd	ye	yf	yg	yh	yi	yj	yk	yl	ym	yn	yo	yp	yq	yr	ys	yt	yu	yv	yw	yx	yy	yz	
2007	2143	27	115	272	301	12	30	22	192	23	86	1104	148	1826	271	15	6	291	401	104	141	106	4	28	23	78
za	zb	zc	zd	ze	zf	zg	zh	zi	zj	zk	zl	zm	zn	zo	zp	zq	zr	zs	zt	zu	zv	zw	zx	zy	zz	
160	4	2	2	373	0	1	43	364	2	2	123	35	4	110	2	0	32	4	4	73	2	3	1	147	45	

```
for w in words:
    chs = ['.'] + list(w) + ['.']
    for ch1, ch2 in zip(chs, chs[1:]):
        ix1 = stoi[ch1]
        ix2 = stoi[ch2]
        N[ix1, ix2] += 1
```

BIGRAM MODELS

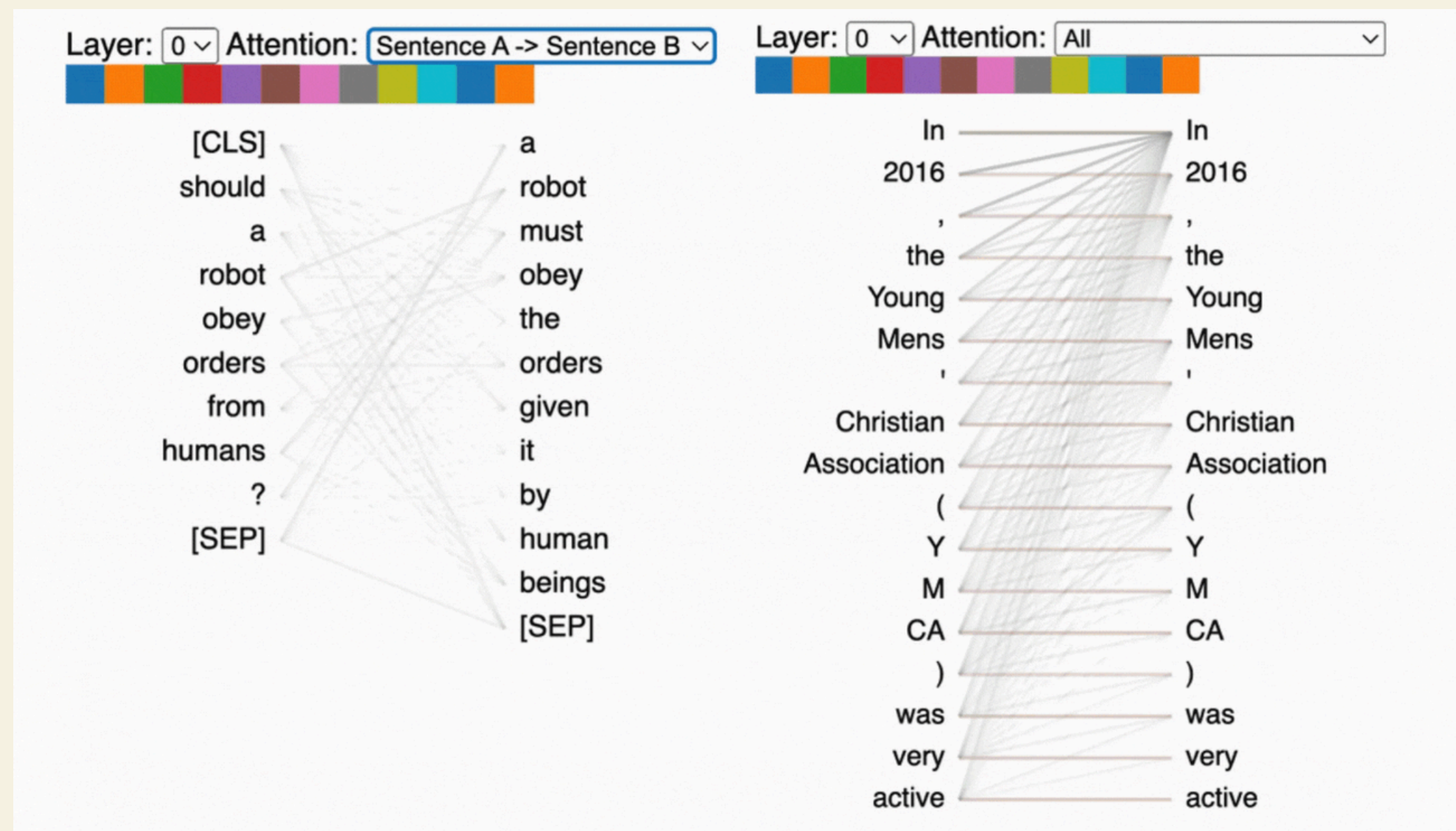
mor.
axx.
minaymoryles.
kondlaisah.
anchthizarie.



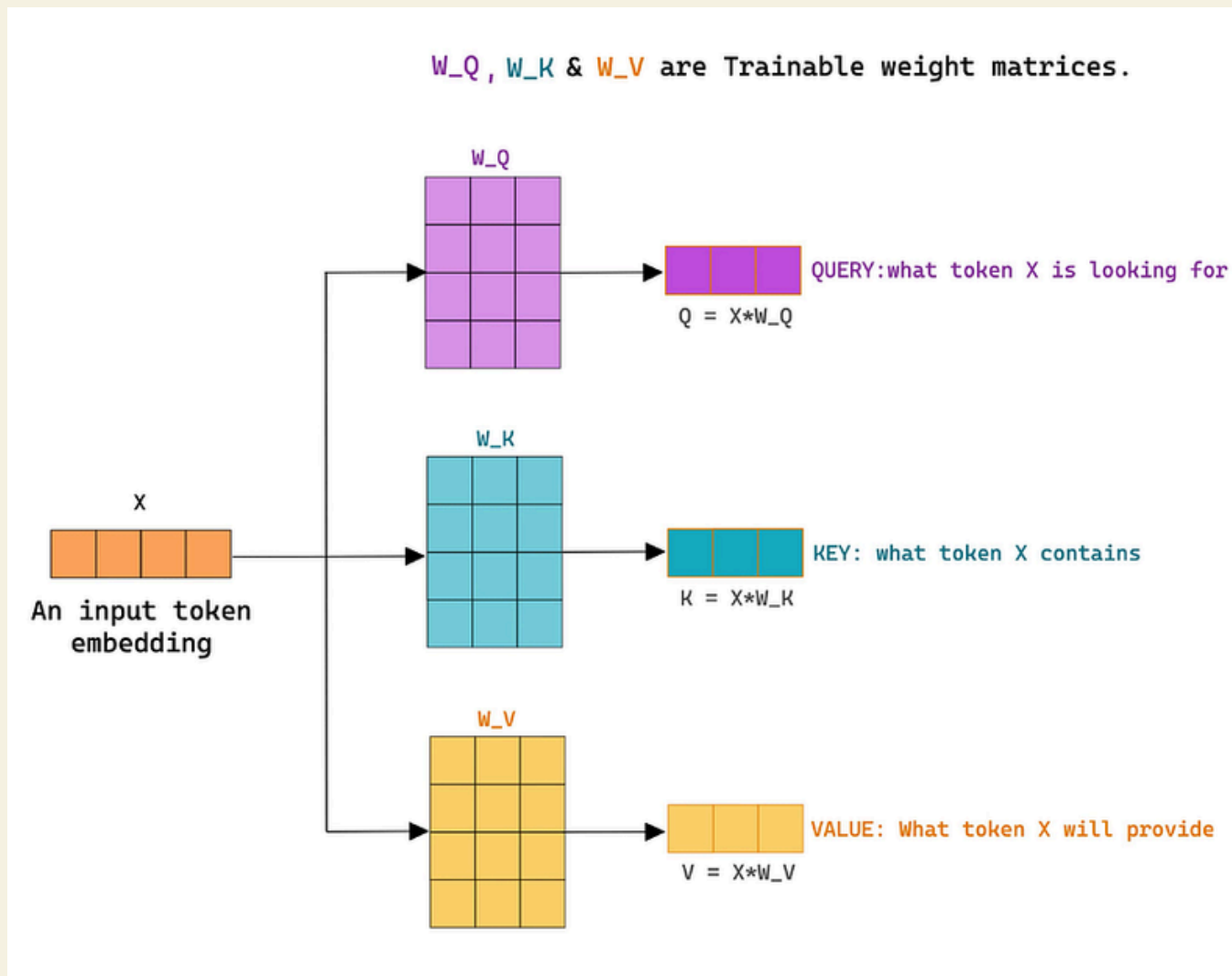
DRUM ROLLS



ATTENTION



SELF ATTENTION

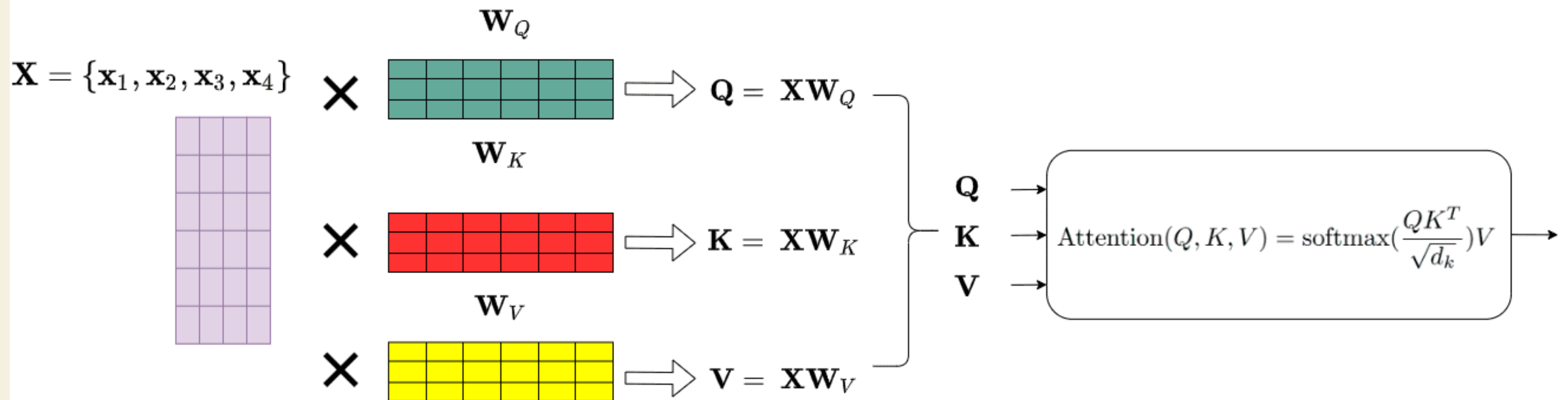


```
#single head of attention
class Head(nn.Module):
    def __init__(self, head_dim):
        super().__init__()
        self.key = nn.Linear(EMBEDDING_DIM, head_dim, bias=False)
        self.query = nn.Linear(EMBEDDING_DIM, head_dim, bias=False)
        self.value = nn.Linear(EMBEDDING_DIM, head_dim, bias=False)
        self.register_buffer('tril', torch.tril(torch.ones(CONTEXT_WINDOW, CONTEXT_WINDOW)))
        self.dropout = nn.Dropout(DROPOUT_RATE)

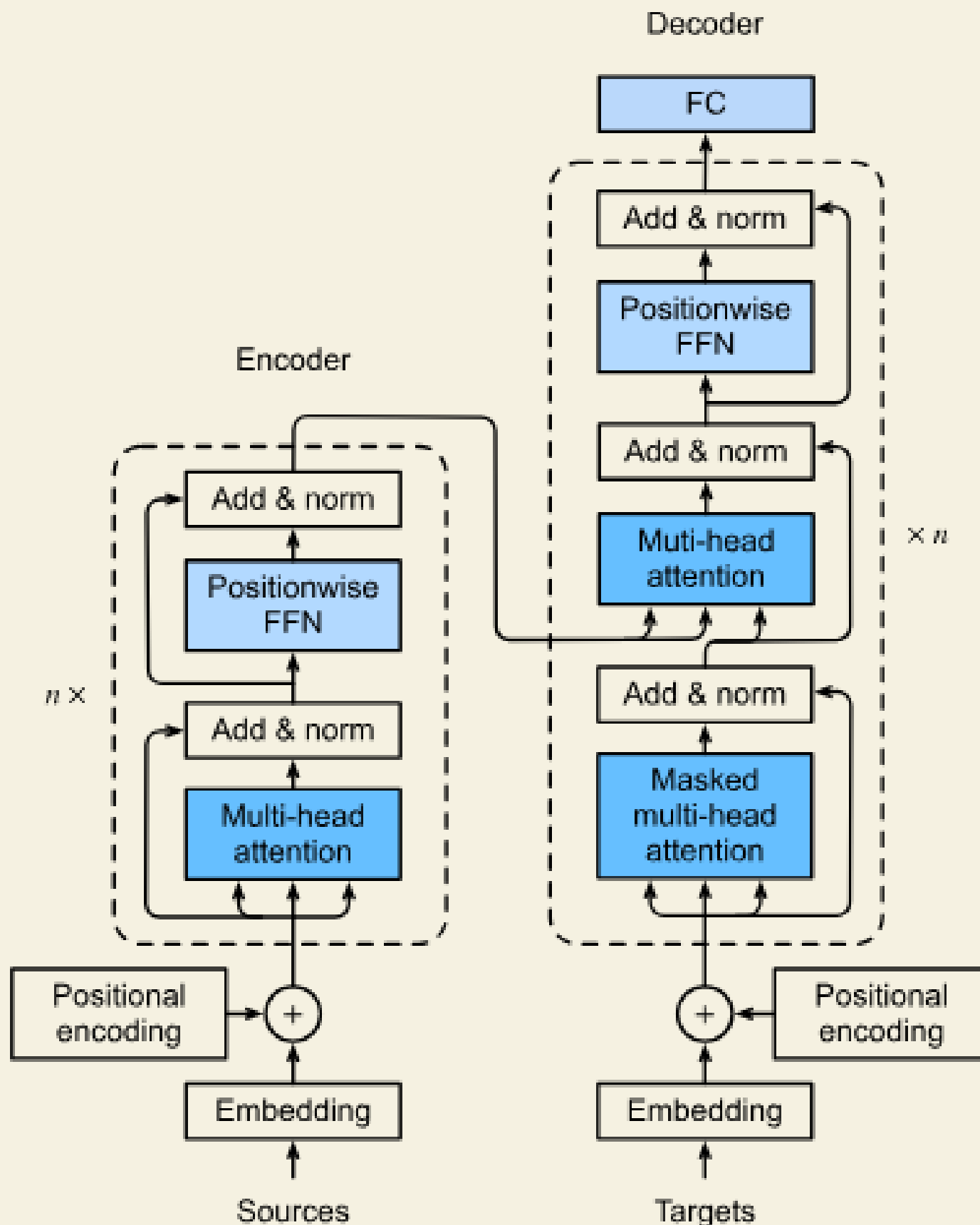
    def forward(self, x):
        #here B-> batch, T -> time-step, C -> channels
        B, T, C = x.shape
        k, q, v = self.key(x), self.query(x), self.value(x) # dimesnsion of this is (B,T,HeadSize)

        # compute attention scores ("affinities")
        attention_scores = (q @ k.transpose(-2, -1)) * k.shape[-1]**-0.5
        attention_scores = attention_scores.masked_fill(self.tril[:T, :T] == 0, float('-inf'))
        attention_probs = F.softmax(attention_scores, dim=-1)
        attention_probs = self.dropout(attention_probs)
        return attention_probs @ v
```

SELF ATTENTION



TRANSFORMER



```
class Block(nn.Module):
    def __init__(self, EMBEDDING_DIM, HEADS):
        File display iper().__init__()
        head_size = EMBEDDING_DIM // HEADS
        self.mha = MultiHeadAttention(HEADS, head_size)
        self.ffwd = FeedForward(EMBEDDING_DIM)
        self.ln1 = nn.LayerNorm(EMBEDDING_DIM)
        self.ln2 = nn.LayerNorm(EMBEDDING_DIM)

    def forward(self, x):
        x = x + self.mha(self.ln1(x))
        x = x + self.ffwd(self.ln2(x))
        return x
```

सस्ताGPT 10M

A sell to of on the regage to tke
they ophind, anyou be them pranive anter, the prest be
in mad, and
in the he women the to see noth weo you hare forthinl i
is I afferly
be purice,
you to hy eximist it also eever posssuat how
reithes of all
it the matisger plearate out to thaning imple the

सस्ताSTOIC

I whorld forther mere have bu lemm, as thy I chand revous
With s?

DUKE NINE:

Nogne say,

I rest reads speec mone,-rato his law'Y, cacesmid, say,

I is wou's domid

BaGnother,

But your our cover'd at of this fold is lass? the as than

सस्ताSHAKESPEARE

→ DEEPSEEK?

10.713691 M parameters

File display
step 0: train loss 4.4460, val loss 4.4495
step 500: train loss 2.1563, val loss 2.2279
step 1000: train loss 2.0441, val loss 2.1356
step 1500: train loss 1.9811, val loss 2.0880
step 2000: train loss 1.9382, val loss 2.0526
step 2500: train loss 1.8820, val loss 2.0114
step 3000: train loss 1.8584, val loss 1.9786
step 3500: train loss 1.8271, val loss 1.9861
step 4000: train loss 1.8289, val loss 1.9594
step 4500: train loss 1.7941, val loss 1.9273
step 4999: train loss 1.7831, val loss 1.9317

THANK YOU

LOG OUT

