

Uppgift 3

Litteratur: Läs kapitel 8.4 och (eventuellt A1-A4) i Jurafsky-Martin, samt titta på föreläsning 5.

Kod: Skelettkoden kan laddas ned från Canvas eller från

http://www.csc.kth.se/~jboye/teaching/language_engineering/a03/HMM.zip

Unzippa koden i lämplig mapp. Öppna ett kommandofönster, gå till foldern HMM och skriv:

```
pip install -r requirements.txt
```

Nu ska allting du behöver för att göra labben vara installerat.

Problem:

1. Här är en förenklad modell av feltryckningar på ett tangentbord: Vi antar att sannolikheten är 0.1 att felaktigt råka trycka ner någon tangent som ligger bredvid den avsedda tangenten. Om vi t.ex. menar att trycka ner tangenten A, är sannolikheten att vi istället kommer åt Q, W, S, eller Z 0.1 vardera. Sannolikheten att faktiskt trycka ner A som avsett är $1.0 - 0.4 = 0.6$. För att ytterligare förenkla problemet bortser vi från alla tangenter förutom A-Z och MELLANSLAG. Vi antar vidare att om vi avser att trycka MELLANSLAG så trycker vi rätt med sannolikhet 1, samt att MELLANSLAG aldrig trycks ned av misstag.

~	!	@	#	\$	%	^	&	*	()	-	+	Backspace
Tab	Q	W	E	R	T	Y	U	I	O	P	{	}	
Caps Lock	A	S	D	F	G	H	J	K	L	:	"	Enter	
Shift	Z	X	C	V	B	N	M	<	>	?	/	Shift	
Ctrl	Win Key	Alt								Alt	Win Key	Menu	Ctrl

Givet en text som innehåller feltryckningar enligt sannolikhetsfördelningen ovan, är **din första uppgift att återskapa den avsedda texten så väl som möjligt**. Till din hjälp finns en fil som innehåller sannolikheter för bokstavsbigram, en annan fil som innehåller sannolikheter för bokstavstrigram (för engelska).

Titta i filen `bigram_probs.txt`. Varje rad i filen har tre siffror. Det två första siffrorna motsvarar indexsiffror: 0 för a, 1 för b, ..., 25 för z, och slutligen 26 för SPACE/START/END-symbolen, som representerar början och slutet av ett ord, samt början och slutet av en mening. Den sista siffran på varje rad är (den naturliga logaritmen av) bigram-sannolikheten.

Till exempel,

```
0 1 -3.748896861435106
```

innebär att $P(\text{ab}) = P(\text{b}|\text{a}) = -3.748896861435106$, medan

```
0 26 -2.481764189851945
```

motsvarar sannolikheten för bigramet “a följt av mellanslag/slut på meningen”. På samma sätt betyder raden

```
0 1 2 -5.969906514008791
```

i filen `trigram_probs.txt` att $P(\text{abc}) = P(\text{c}|\text{ab}) = -5.969906514008791$.

- (a) Förklara hur denna modell av tangenttryckningar kan modelleras som en Hidden Markov Model. Vilka är **gömda tillstånd**, **observationerna**, **övergångssannolikheterna** (state transition probabilities), och **observationssannolikheterna**?
- (b) `ViterbiBigramDecoder`-filen innehåller ett kodskelett för att applicera Viterbialgoritmen på problemet, och på så vis göra den feltryckta texten mer läsbar. Utöka koden så att den fungerar som den ska (leta efter kommentarerna `YOUR CODE HERE` och `REPLACE THE STATEMENT BELOW WITH YOUR CODE` i koden). Testa din implementation på några testfall genom att köra skriptet `run_bigram_decoder.sh`. Du kan jämföra dina resultat med de förväntade resultaten i filen `test_bigram_decoding.txt` och/eller genom att använda flaggan `--check`.

Tips: Notera att programmet adderar en `START_END`-symbol i slutet av input-strängen. För att få ut resultatet bör din implementation helt enkelt följa backpointers från `START_END` i det sista tidssteget.

2. Kanske kan man få ett bättre resultat genom att ta hänsyn till mer kontext i modellen?

- (a) Implementera Viterbialgoritmen med trigram-sannolikheter genom att utöka klassen `ViterbiTrigramDecoder`, så att den fungerar som den ska. Testa din implementation på testfallen genom att köra skriptet `run_trigram_decoder.sh`, och jämför med de förväntade resultaten i filen `test_trigram_decoding.txt`, och/eller genom att använda flaggan `--check`.
- (b) Kör dina bigram- och trigram-avkodare på de 5 filerna `mistyped_1.txt` till och med `mistyped_5.txt`. Skriv ner svaren som returneras från båda programmen.
- (c) Vad handlar texterna om? Kan du identifiera (eller gissa) källorna?