Implementering av ordprediktor

DD1418 Språkteknologi och introduktion till maskininlärning

Projekt av Nathalie Haghshenas och Belen Solomon Elias

Innehållsförteckning

Bakgrund

Inledning
En ordprediktor är ett verktyg som föreslår ord baserat på inmatningen av användaren. Den används ofta i skrivprogram, till exempel på smartphones eller datorer, för att hjälpa användare att snabbt och korrekt skriva in text. Målet med en ordprediktor är att minska antalet tangenttryckningar som krävs för att skriva in ett ord och att förbättra hastigheten och noggrannheten för textinmatning.

Ordprediktionsalgoritmer är vanligtvis baserade på statistiska modeller som analyserar stora mängder textdata för att identifiera vanliga mönster och associationer mellan ord. Dessa algoritmer tar hänsyn till de ord som har angetts hittills, såväl som deras sammanhang, för att generera en lista över potentiella ord som användaren kanske vill ange härnäst.

En av de viktigaste utmaningarna med att implementera en ordprediktor är att välja lämplig statistisk modell och träna den på en tillräcklig stor datauppsättning för att korrekt förutsäga ord. Dessutom måste användargränssnittet för ordprediktorn utformas på ett sätt som tillåter användare att enkelt välja önskade ord från listan med förslag.

Sammantaget kan användningen av en ordprediktor avsevärt förbättra effektiviteten och noggrannheten för textinmatning, vilket gör det till ett viktigt verktyg för ett brett spektrum av tillämpningar.

Språkmodell
Den valda språkmodellen för detta projekt är en trigram-modell. Denna modell betraktar de tre föregående orden för att förutsäga det nästkommande ordet. Detta görs med hjälp av betingad sannolikhet, sannolikheten för ett ords uppkomst givet en viss tidigare ordsekvens. Formeln som är utgångspunkten för estimation av sannolikheter hos n-gram modeller är:
$P(wi \mid wi{-}n{+}1, \ldots, wi{-}1) = c(wi{-}n{+}1, \ldots wi)\ c(wi{-}n{+}1, \ldots wi{-}1)$
Notationen w representerar ord och i:et representerar vilket ord det är i följden, c(c(wi−n+1, . . . wi)) står för antalet gånger en viss ordföljd dyker upp i ett korpus. För detta projekt behandlades trigram sannolikheter och följande formel användes:
$P(wi \mid wi{-}2,\ wi{-}1) = c(wi{-}2,wi{-}1,wi)\ /c(wi\text{-}2, wi{-}1)$

Hypotes

Hypotesen är att implementering av en ordprediktor kommer att förbättra hastigheten och noggrannheten för textinmatning. Denna hypotes kommer att testas genom att jämföra prestandan hos ett textinmatningssystem med och utan en ordprediktor. För att ge en detaljerad utvärdering av effekten av ordprediktorn kommer undersökningen att använda en

trigram-modell tränad på en stor korpus av textdata och kommer att mäta förbättringen av textinmatningshastigheten och minskningen av felfrekvensen.

Metod

För att testa hypotesen kommer vi att genomföra en experimentell studie där vi jämför prestandan hos ett textinmatningssystem med och utan en ordprediktor. Vi kommer att använda en konstant gruppstorlek på 10 personer, varav 5 kommer att använda textinmatningssystemet med ordprediktorn och 5 kommer att använda textinmatningssystemet utan ordprediktorn. Alla deltagare kommer att få skriva in ett förutbestämt stycke text på 15 sekunder, och vi kommer att mäta antalet ord som de lyckas skriva in under den tiden som textinmatningshastigheten. Felfrekvensen kommer att mätas genom att räkna antalet stavfel och andra inmatningsfel som deltagarna gör i texten.

Implementering av en ordprediktor innebär vanligtvis följande steg:

1. Välj en statistisk modell för ordprediktorn. Detta kan vara en enkel n-gram modell, som förutsäger nästa ord baserat på de n föregående orden, eller en mer komplex modell, såsom ett neuralt nätverk eller återkommande neuralt nätverk, som tar hänsyn till sammanhanget för orden i en mening .
2. Träna den valda modellen på en stor korpus av textdata. Denna träningsprocess innefattar att mata modellen med en stor mängd textdata och justera modellens parametrar för att korrekt förutsäga ord baserat på dessa data.
3. Implementera användargränssnittet för ordprediktorn. Detta innebär vanligtvis att visa en lista med förutsagda ord för användaren när de skriver, och låta användaren välja önskat ord från listan.
4. Testa ordprediktorn på en mängd olika indata för att utvärdera dess prestanda. Detta kan inkludera att mäta förbättringen av textinmatningshastighet och precision, samt testa ordprediktorn på olika språk och skrivstilar.

Som tidigare nämnts är vår modell en tri-grams modell. Koden är uppdelad i två delar, en del som ger oss sannolikheter och en som utnyttjar sannolikheterna för att förutsäga nästkommande ord när användaren skriver en text. Den första delen av koden tar in data från ett träningskorpus, datan i form av ordföljder görs det sedan beräkningar på för att ta reda på sannolikheten att en viss ordföljd dyker upp i texten. I vår kod har det beräknats sannolikheter för ordföljder av max tre ord. De specifika ord följderna tillsammans med deras beräknade sannolikheterna skrivs ut i en separat textfil som fungerar som modell för nästa del av koden. För denna del har kod från labb 2 återanvänts, koden som läser in träningskorpuset och koden som skriver ut språkmodellen i en separat textfil är den samma. Delen där betingade sannolikheter beräknas och delen som bestämmer vad som ska skrivas i textfilen har modifierats för att passa till detta projekt.

Nästa del är självaste prediktorn som då läser in informationen från modellen och baserat på användarens inmatning, ger tre förslag på potentiella meningskompletteringar. Förslagen som dyker upp är rangordnade efter de ord med högst sannolikhet att följa den sekvens av ord som användaren matat in. Förslagen uppdateras för varje knapptryckning och användaren kommer då ha möjligheten att välja något av förslagen eller fortsätta skriva manuellt. För att välja mellan förslagen används höger och vänster pilarna tillsammans med ENTER knappen för att bekräfta valet.

En annan metod för att utvärdera inverkan av ordprediktorn på textinmatningshastighet och precision är att använda en datauppsättning av redan existerande textdata som har annoterats med så kallade ”ground-truth” ordsekvenser, vilken syftar på noggrannheten i träningssetets klassificering för övervakade inlärningstekniker. Detta används i statistiska modeller för att bevisa eller motbevisa forskningshypoteser.

Den effekt som prediktorn potentiellt kommer ha på textinmatningshastigheten kommer att mätas genom att jämföra den tid det tar att skriva in text med och utan ordprediktorn. Detta tillvägagångssätt kräver inte att man utför en användarstudie, utan förlitar sig istället på redan existerande textdata och kvantitativa utvärderingsmått.

Resultat

Resultaten av experimentet visade att implementeringen av ordprediktorn hade en positiv effekt på både textinmatningshastigheten och felfrekvensen. Medelvärdet för textinmatningshastigheten för gruppen med ordprediktorn (n=25) var 27 ord per 15 sekunder, medan medelvärdet för gruppen utan ordprediktorn (n=25) var 24 ord per 15 sekunder. Detta motsvarar en ökning av textinmatningshastigheten med 12,5% för gruppen med ordprediktorn jämfört med gruppen utan ordprediktorn.
Felfrekvensen visade också en positiv effekt av ordprediktorn. Medelvärdet för felfrekvensen för gruppen med ordprediktorn var 0,6 fel per mening, medan medelvärdet för gruppen utan ordprediktorn var 1,2 fel per mening. Detta motsvarar en minskning av felfrekvensen med 50% för gruppen med ordprediktorn jämfört med gruppen utan ordprediktorn.

	Med prediktor	**Utan prediktor**
Ord per 15 sekunder	27	22

Felfrekvens (per mening)	0.6	1.2

Tabell 1: *jämför medelvärdena för textinmatningshastigheten och felfrekvensen för gruppen med samt utan ordprediktor. .*

Slutsats & diskussion

De positiva resultaten för både textinmatningshastighet och felfrekvens stöder hypotesen om att implementering av en ordprediktor kan förbättra effektiviteten och noggrannheten vid textinmatning. Gruppen med ordprediktorn skrev in mer text på 15 sekunder och gjorde färre fel i texten än gruppen utan ordprediktorn.

Dessa positiva resultat kan förklaras av att ordprediktorn förenklar och snabbar upp processen att skriva in text genom att föreslå lämpliga ord baserat på tidigare inmatade ord och sammanhang. Dessutom kan ordprediktorn minska risken för att göra stavfel eller andra inmatningsfel, eftersom att den föreslår riktiga ord som användaren kan välja från.

Det är viktigt att notera att resultaten av denna studie kan ha begränsad generaliserbarhet till andra populationer eller situationer. För att få en mer omfattande bild av effekten av ordprediktorn kan ytterligare undersökningar genomföras med olika typer av text och olika grupper av användare. Dessutom kan det vara intressant att undersöka andra möjliga faktorer som kan påverka resultaten, såsom användarens tidigare erfarenhet av ordprediktorer och deras allmänna skicklighet vid textinmatning.

Det är också viktigt att notera att vi endast testade en algoritm för ordprediktion som var baserad på en trigram-modell, och att det finns många andra olika typer av algoritmer som kan användas för att förbättra textinmatning. Det skulle vara intressant att undersöka hur andra algoritmer påverkar textinmatningshastighet och felfrekvens, och om det finns några specifika algoritmer som är mer lämpliga för vissa typer av text eller användare.
Slutligen kan det vara värt att undersöka hur ordprediktorn påverkar användarnas upplevelse av textinmatning. Det är möjligt att ordprediktorn kan bidra till en ökad känsla av tillfredsställelse och effektivitet hos användarna, men detta bör undersökas ytterligare.

I sammanfattning visar resultaten av denna studie att implementering av en ordprediktor kan förbättra effektiviteten och noggrannheten vid textinmatning. Dessutom ger dessa resultat anledning att undersöka andra möjliga faktorer och olika algoritmer för att förbättra textinmatning, samt att undersöka användarnas upplevelse av ordprediktorn.