

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4426

**Povećanje brzine unosa teksta na
zaslonu osjetljivom na dodir
prediktivnim vizualnim isticanjem**

Mihael Marović

Zagreb, srpanj 2016.

Zahvaljujem se mentoru prof. dr. sc. Vladi Glaviniću na pruženoj pomoći i savjetima oko završnog rada. Zahvaljujem se kolegi Juraju Šušnjari na genralnoj dostupnosti i suradnji u vezi provedbe eksperimenta. Zahvaljujem se svim kolegama koje su sudjelovale u provedbi eksperimenta, a i onima koji su odbili sudjelovanje u eksperimentu iz opravdanog razloga. Zahvaljujem se svim autorima literature i drugih resursa koje sam koristio pri izradi završnog rada.

SADRŽAJ

1. Uvod	1
2. Unos teksta na zaslonima osjetljivima na dodir	3
3. Algoritam za prediktivno vizualno isticanje	5
4. Predikcija za zaslone osjetljive na dodir	7
4.1. Radijalni izbornici	7
4.2. Predikcija radijalnim izbornicima	8
5. Vrednovanje mehanizama prediktivnog vizualnog isticanja	13
5.1. Oblikovanje eksperimenta	13
5.1.1. Oblikovanje eksperimenta unutar sudionika i između sudionika	13
5.1.2. Učinak poretka, suprotstavljanje, latinski kvadrati	14
5.1.3. Upitnik	14
5.2. Provođenje eksperimenta	14
5.2.1. Varijable eksperimenta	14
5.2.2. Grupiranje sudionika u eksperimentu	15
5.3. Prikaz i analiza rezultata	17
5.3.1. Rezultati grupe koja tipka na 11" tabletu	17
5.3.2. Rezultati grupe koja tipka na 7" tabletu	19
5.3.3. Rezultati grupe koja tipka na 5" mobitelu	20
5.3.4. Usporedba rezultata triju grupa	21
5.3.5. Mane i prednosti testiranih mehanizma predikcije	22
5.3.6. Analiza varijance	23
5.3.7. Analiza rezultata upitnika	24
6. Zaključak	27

Literatura	28
A. Neki tehnički detalji	31
B. Autorska implementacija standardnog mehanizma predikcije	32
C. Implementacija alternativnog mehanizma predikcije - bitni isječci koda	34
D. Upitnik korišten za vrednovanje mehanizama predikcije	37

1. Uvod

Tema ovog završnog rada je ubrzanje mehanizma povećanja brzine unosa teksta standardne QWERTY tastature za mobilne naprave sa zaslonom osjetljivim na dodir. U ovom radu neće nas zanimati drugi potencijalni razmještaji tipaka osim QWERTY, nego ćemo taj parametar zanemariti i fokusirati se na mehanizam predikcije. Temeljna ideja je, za razliku od standardnog mehanizma vizualnog isticanja koji korisniku nudi riječi iznad razmještaja (layouta) tipkovnice, a ponuđene riječi imaju (ili ne moraju) imati zajednički prefiks kao riječ koja se upravo tipka, ponuditi samo sufiks i to na poseban način koji je predložio Poika Isokoski. Njegova ideja je koristiti mehanizam analogan radijalnom izborniku. Dakle, pritiskom na tipku otvorit će se radijalni izbornik koji će nuditi potencijalne sufikse koji odgovaraju prefiksu trenutno utipkane riječi.

Kako bismo došli do nekih saznanja o razvijenom mehanizmu predikcije, upustit ćemo se u vrednovanje istoga. Razvijeni mehanizam predikcije treba usporediti sa standardnim i vrednovati u odnosu na poznate attribute upotrebljivosti. Razvijeni mehanizam predikcije bi također trebao koristiti standardni T9 algoritam za predikciju. Vrednovanje će se provesti na mobilnim napravama različitih dimenzija - pločnim računalima (tabletima) od 10,1" i 7" te pametnom mobilnom telefonu od cca 5"-5,5". To činimo da bismo usporedili i kvantitativno opisali različite načine unosa teksta. Naša implementacija se treba moći izvršavati na Android operacijskom sustavu, a po želji i na drugim operacijskim sustavima koji podržavaju zaslone osjetljive na dodir. Kako bismo dobili što bolje rezultate, vrednovanje ćemo provesti uz ispitne korisnike koji su "početnici" i nisu imali prethodnog kontakta s razvijenom tastaturom. Zbog jednostavnosti (s obzirom na to da algoritam i njegova prilagodljivost za različite jezike nije glavna tema ovog rada) predložak teksta za utipkavanje uzet ćemo is standardnog korpusa engleskoga jezika. Naime, za engleski jezik jednostavnije je ponuditi mehanizam (algoritam) predikcije zbog ne postojanje padežnih nastavaka kao u hrvatskom jeziku. Dakle, ciljani jezik će biti engleski jezik. U sljedećim poglavljima opisat će se korišteni algoritam te razraditi mehanizmi predikcije na zaslonima osjetljivim na dodir. U konačnici biti će obavljeno vrednovanje razrađenih mehanizama. U sklopu vredno-

vanja biti će ponuđena anketa u kojoj će ispitni korisnici moći dati svoje mišljenje o vrednovanim mehanizmima predikcije.

2. Unos teksta na zaslonima osjetljivima na dodir

Ljudi su u konačnici nakon pojave jezika počeli bilježiti glasove (ili nizove glasova) odgovarajućim znakovima (simbolima). Primjerice, stari Egipćani koristili su papirus za zapis teksta i drugih simbola. Izumom pisaćeg stroja dolazi nova mogućnost zapisa teksta. Pritiskom na gumb pokreće se fizički mehanizam koji za rezultat ostavlja trag tinte na papiru. Samim tim izumom prirodno dolazi do toga da ta same tipke moraju biti nekako razmještene u prostoru. Christopher Latham Sholes primijetio je da kod klasičnog abecednog rasporeda slova latinskog pisma dolazi do zaglavljivanja za vrijeme tipkanja. Naime, parovi slova koji su česti u riječima (bigrami) istovremeno su bili blizu jedno drugome na tipkovnici abecednog rasporeda, kao i u čestim riječima (primjerice "s" i "t" su uzastopni u abecedi te se pojavljuju kao bigram u riječi "stop" i mnogim drugim riječima). Dakle, da bi riješio taj problem (i ubrzao tipkanje na pisaćim mašinama) izumio je QWERTY raspored tipaka. Tu je zanimljivo napomenuti i to da se Sholesov raspored tipaka ipak nešto razlikuje od današnjeg. Najprimjetnije, u nedostatku tipaka za brojeve 0 i 1. Razvitkom tehnologije, pojavila su se osobna računala i pisači (printeri) te su oni zamijenili nekadašnje pisaće strojeve. Svejedno, raspored tipaka QWERTY ostao je isti. Jasno je da su nastale varijante QWERTY rasporeda ovisno o jeziku (a kojeg je moguće zapisivati varijantom latiničnog pisma) za koji je taj raspored tipaka prilagođen. Mi ćemo u našem radu vrednovanja vršiti na varijanti QWERTY tipkovnice za hrvatski jezik (koju tada možemo nazvati QWERTZ tipkovnicom). Zatim, nastankom mobilnih uređaja, pojavile su se tipkovnice koje su imale gumbe s brojevima (telephone keypad), a višestruki pritisak na pojedini broj generirao bi pojedino slovo. Takva tipkovnica prikaza je na slici 2.1 (preuzeto s <http://www.yorku.ca/mack/chapter5-f2.jpg>).

Sada dolazimo do naše tipkovnice, odnosno do tipkovnice za zaslone osjetljive na dodir. Tu ponovno u igru ulaze QWERTY tipkovnice. Za tipkovnice za zaslone osjetljive na dodir razvijena je posebna verzija QWERTY razmještaja. Raspored slova je



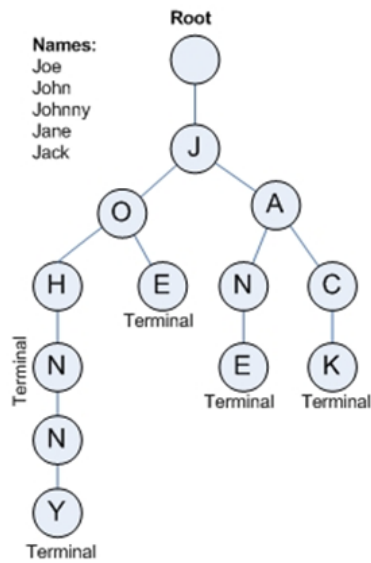
Slika 2.1: Tipkovnica starih mobitela

kao i na fizičkoj tipkovnici, dok su na istom prikazu nadodani najčešći znakovi poput zareza i točke. Pritiskom na poseban gumb moguće je promijeniti prikazanu sliku tipkovnice i tada se počne nuditi standardna paleta dodatnih znakova. S obzirom na to da se sada tipka na zaslonu osjetljivom na dodir, nameće se mogućnost da se korisniku ponudi prijedlog, odnosno nastavak riječi koje pokušava utipkati s ciljem da mu se ubrza tipkanje. Takvo poboljšanje je potrebno, ponajviše u kontekstu toga da je čak i objektivno nešto nezgrapnije tipkati na takvim zaslonima nego na fizičkim tipkovnicama. U nastavku rada razradit ćemo jedan alternativni mehanizam prediktivnog vizualnog isticanja pored standardnog.

3. Algoritam za prediktivno vizualno isticanje

Ovdje ćemo dati osvrt na korišteni algoritam koji koristimo da bismo generirali riječi koje prikazujemo prediktivnim vizualnim isticanjem. Algoritam je T9 i nisam nigdje našao službenu implementaciju sa službenim rječnikom, ali proći ću kroz dva načina na koji bi se taj algoritam za dani rječnik (koristimo engleski jezik) mogao implementirati. Pseudokod korištenog algoritma je onaj kojeg sam "smislio" (odnosno, pojedini izvori su također predložili takvu ili vrlo sličnu implementaciju, primjerice: <https://www.quora.com/How-does-predictive-text-work>, ali nigdje nisam našao službenu verziju) i vrlo je intuitivan: od svih riječi koje počinju na prefiks koji je korisnik utipkao (radi jednostavnosti pretpostavljamo da je prefiks ispravan, dakle nećemo nuditi ispravke korisniku ako nam se čini da nije ponudio valjani prefiks) odaberemo n (primjerice 5 ili manje od 5 ako nema niti 5) najfrekventnijih u rječniku. Za ovo nam je uz riječi potrebna i frekvencija pripadnih riječi. Za rječnik odabrao sam Googleovu listu 10000 najfrekventnijih riječi engleskog jezika (postoji i nova lista od 20000 riječi) dostupnu na <https://github.com/first20hours/google-10000-english>. S obzirom na to da su riječi sortirane po padajućoj frekvenciji za svaku riječ poznat nam je relativan odnos (za svake dvije riječi iz rječnika znamo koja je frekventnija). Dakle, možemo sortirati riječi abecedno i sačuvati jedan broj uz njih koji nam govori koji su po redu bili u originalnoj listi riječi (odnosno njegovu frekvenciju, manji broj u originalnoj listi, znači veća frekvencija pojavljivanja u prirodnom engleskom jeziku). Prvi način implementacije opisanog algoritma je tako da nad rječnikom izgradimo strukturu podataka trie koja je prikazana na slici 3.1 (preuzeto s <http://www.codeproject.com/KB/recipes/PhoneDirectory/Trie.jpg>).

Nakon što se struktura izgradi traženje svih riječi koji počinju kao i zadana riječ je jednostavno, primjera radi vidimo da za stablo (rječnik) sa slike sa prefiksom "Jo" počinju Johnny i Joe. Implementaciju pomoću Triea sam isprobao dok sam radio na standardnoj metodi prediktivnog vizualnog isticanja (za detalje pogledati Dodatak B)



Slika 3.1: Struktura podataka Trie

i za rječnik koji sam koristio tada (više od 10000 riječi, negdje oko 100000 riječi) Trie je zauzimaao previše memorije, tako da sam u konačnici (i za standardnu i za alternativnu) tipkovnicu implementirao drugi način traženja riječi u rječniku sa zadanim prefiksom. Drugi način implementacije temelji se na već spomenutoj činjenici da je moguće sortirati sve riječi abecedno i sačuvati uz njih u paru još jedan broj koji pamti njezinu relativnu frekvenciju u odnosu na druge riječi. Prvo to napravimo, a zatim binarnom pretragom nađemo (neku, onu koju pretraga izgenerira) riječ u rječniku koja ima zajednički prefiks kao i utipkani niz znakova, sada kada znamo "gdje smo" u sortiranoj listi riječi lagano nađemo sve riječi koje imaju zajednički prefiks (tj. prefiks im je trenutno utipkani niz znakova) i od njih odaberemo n ili manje (primjerice 5 ili manje) najfrekventnijih.

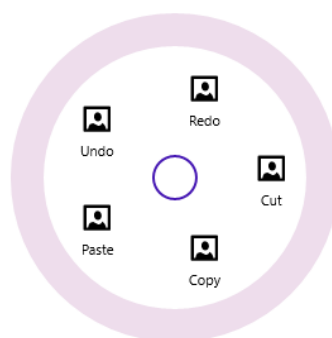
Tu još treba napomenuti da su algoritmi korišteni u komercijalnim tipkovnicama (iako temeljeni na postupku opisanom gore) kompliciraniji i još u obzir uzimaju riječi koje korisnik češće utipkava (čak i parove riječi) i time mijenjaju relativnu frekvenciju između riječi.

4. Predikcija za zaslone osjetljive na dodir

Predikcija na zaslonima osjetljivim na dodir može se vršiti na razne načine. Standardni od njih je da se iznad tipkovnice na kojoj korisnik tipka prikazuju moguće riječi za koje mehanizam predviđa da ih korisnik želi utipkati. Također je moguće, a to ćemo i u okviru ovog rada istražiti, predikciju vršiti tako da se mogući sufixi prikazuju radijalno u odnosu na pritisnutu tipku.

4.1. Radijalni izbornici

Radijalni izbornik je izbornik koji nakon klika izborničke stavke nudi radijalno. Jedan radijalni izbornik prikazan je na slici 4.1 (preuzetoj s https://www.syncfusion.com/blogs/image.axd?picture=image_35.png).



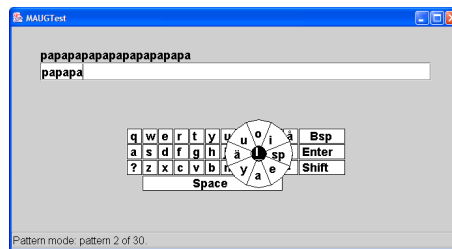
Slika 4.1: Radijalni izbornik

Radijalni izbornici (na engleskom još i pie menu) imaju određene prednosti u odnosu na linearne menije. Recimo, radijalni izbornici su brži i pouzdaniji u odnosu na linearne zato što odabir ovisi o smjeru, a ne o udaljenosti. Zanimljivo je napomenuti da je ustanovljeno da je najbolje da se broj elementa radijalnog izbornika kreće od naj-

manje 3 do najviše 12. U nastavku rada, razmotrit ćemo mehanizam predikcije koji se zasniva na radijalnom izborniku.

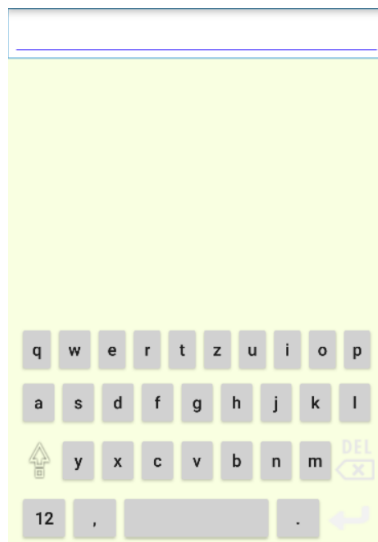
4.2. Predikcija radijalnim izbornicima

U ovom poglavlju opisat će se kako smo došli do testiranog mehanizma predikcije. Želimo postići ponašanje slično onome prikazanome na slici 4.2.



Slika 4.2: Verzija tipkovnice Isokoskog

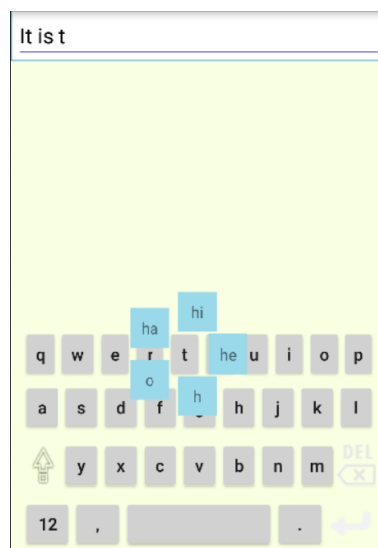
Dakle, prediktivno vizualno isticanje pri čemu se mogući sufixi prikazuju radikalno u odnosu na tipku. Korišteni algoritam opisan je u poglavlju "Algoritam za prediktivno vizualno isticanje". Započet ćemo tako da ćemo izgraditi aplikaciju u koju ćemo smjestiti pogled na tekst i gumbе raspoređeni u takav raspored (layout) da tvore tipkovnicu. Dobiveni izgled je prikazan na slici 4.3.



Slika 4.3: Dobiveni izgled

Trebamo nekako osigurati da se pojave skočni gumbi pritiskom na tipku. Treba biti oprezan iz više razloga, odnosno na više faktora treba obratiti pažnju. Prvi od

njih posljedica je činjenica da se tipkovnica proteže preko cijelog ekrana (gledano pejsažno). Naime, to je prirodno (a i nužno, na manjim ekranima - primjerice ekranima mobitela). Stoga, treba paziti na to da nam se klikom na primjerice "q" otvore samo prijedlozi na kutevima s domenom od (-90, 90), tako da prijedlozi ne izlaze iz ekrana. Također treba osigurati da je veličina tipaka (kako tipkovnice), tako i skočnih gumbova takva da nikad ne izlazi iz ekrana neovisno o dimenziji uređaja. Implementacijski gledano, u Androidu je moguće definirati skočni (Pop up) prozor, međutim odlučio sam potražiti alternativni mehanizam kojim bi najjednostavnije (i najefikasnije) mogli dobiti željeno ponašanje. Na poveznici <https://github.com/oguzbilgener/CircularFloatingActionMenu> dostupan je kod (otvorenog formata) pod MIT-ovom licencom i dostupan je za uporabu pod uvjetom da se fajl o licenci uključi u projekt rada i stoji upozorenje da je software dostupan takav kakav je i da oni (MIT) nisu odgovorni ni za kakve eventualne štetne posljedice korištenja softvera. Odlučio sam pokušati uključiti ovaj kod u svoj projekt i vidjeti kakve će on rezultate dati i kakve mogućnosti nudi. Isprva nisam mogao dodati (attachirati) radijalne gumbe na tipke (gumbe) svoje tipkovnice i trebalo je uključiti neke nove linije koda u izvorni kod (što me iznenadilo). Na slici 4.4 je prikazan rezultat konačne implementacije prediktivnog vizualnog isticanja.

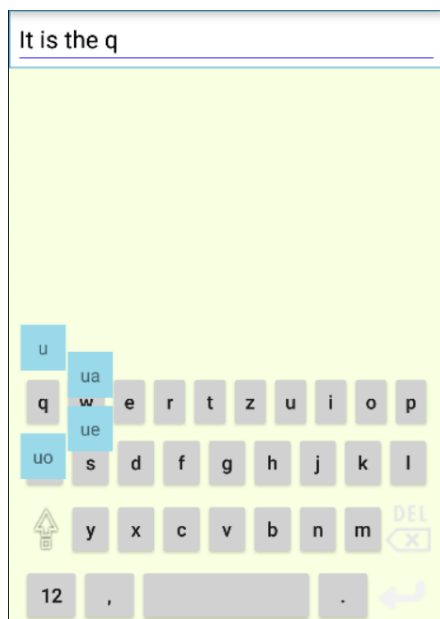


Slika 4.4: Konačan rezultat

Unutar koda možemo specificirati veličinu radijusa skočnih tipaka. Uzimajući u obzir da se veličina tipaka tastature "automatski" generira kroz razmještaj (Androidov layout) tako da se prilagođava veličini ekrana napravio sam da je veličina skočnog gumba takvih dimenzija da ovisi o dimenzijama gumba tipkovnice tako da je njegova

širina i visina odgovaraju minimalnoj od vrijednosti - širina, visina gumba tipkovnice. Zvat ćemo od sada tu duljinu d . Također, treba biti svjestan da širina tipke nije širina sive površine gumba nego je u nju uključen i dio žute površine (zato je veličina plavih skočni gumbiju takva kakva jest). Moguće je podesiti dimenziju radijusa, i njegovu dimenziju sam postavio na $1.1d$ za sve tipke osim onih na rubovima tipkovnice ("q,a,p,l"), njih sam postavio na $1.2d$. Dakle, broj skočnih tipka je 5 za nerubne tike, a za rubne tipke je 4. U gornjem tekstu navedene su korištene dimenzije, a sada ih treba opravdati. Isokoski je originalno zamislio skočni izbornik s 8 ponuđenih slova (ruža vjetrova) kao na slici 4.2. Pri tome skočni izbornik s tipkama zauzima znatan dio površine ekrana. Moja ideja je da bi skočne tipke trebale zauzimati manju ukupnu površinu ekrana, zbog toga što se ipak ne možemo s tolikom sigurnošću pouzdati da će mehanizam predikcije ponuditi ispravan sufiks koji korisnik namjerava utipkati. Također, skočni prozor bi trebao biti što bliže utipkanoj tipki. Oblik skočne tipke će bit kvadratičast - analogno dimenzijama same tipke. Nadalje, skočna tipka ne bi trebala izlaziti iz ekrana. Kada se postave ovakvi uvjeti na izgled tipkovnice, vrlo brzo se prirodno dođe do dimenzija tipaka navedenih gore. Duljina stranice kvadrata na kojem je prediktivni sufiks je minimalna od duljine širine i visine tipke tipkovnice (prvenstveno da su sumjerljive i da tipka ne izlazi iz ekrana) . Kao što smo već napomenuli, ovu veličinu označavamo sa d . S obzirom na to da se radijus računa kao udaljenost središta stisnute tipke tipkovnice od središta skočne tipke njegovu veličinu ćemo uzeti za faktor veći od d da se gumbi ne bi međusobno preklapali, a da bi i dalje bili dovoljno blizu jedan drugoga. Kada postavimo i ove uvjete vidimo (eksperimentalno) da je broj skočnih najviše 5 ili 6 (za veće brojeve se potpuno preklapaju). Temeljni razlog zašto sam se odlučio za 5 tipki je da bi se osigurala bolja preglednost i veći razmak među tipki na manjim (primjerice mobilnim) uređajima. Također, u kontekstu činjenice da korisnik koji tipka mora pročitati sve ponuđene sufikse da bi ustanovio odgovara li mu ponuđeni sufiks, moglo bi se reći da je bolje za korisnikovu koncentraciju, dakle i brzinu tipkanja ponuditi manji broj sufiksa. Dodatno, zbog navedenih razloga koji se tiču dimenzija tipki, a i preglednosti, najbolji broj skočnih tipaka za rubne tipke je 4. Tada, da se broj skočnih tipki kod rubnih tipaka ne bi previše razlikovao od onih nerubnih - kao najbolja opcija je se čini prikazati 5 skočnih tipki za nerubne tipke. Na slici 4.5a moguće je vidjeti kako su riješene rubne tipke. Treba razriješiti što ćemo raditi u slučaju da je mogućih prijedloga manje od 5 (ili 4 za rubne tipke). Odlučio sam jednostavno pokazati neki od sufiksa više puta kao što se može uočiti na slici 4.5b.

Moja implementacija tipkovnica ima brojeve i dodatne znakove kao i standardna tipkovnica. Pri čemu, nema ukupno 3 tastatura tj. rasporeda tipki (jedan raspored sa

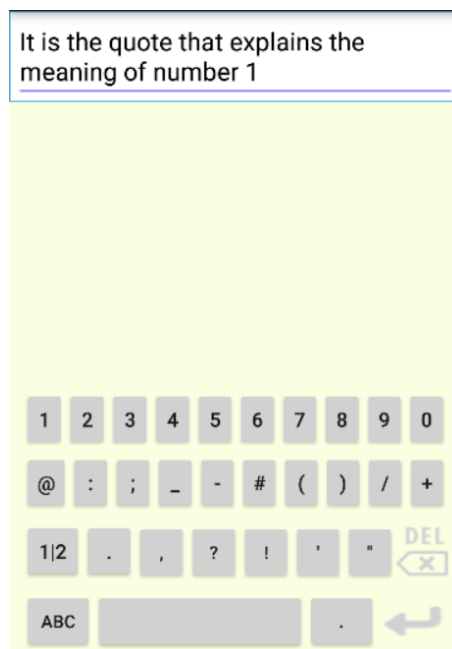


(a) Rubne tipke



(b) Manjak sufiksa

Slika 4.5: Rubni slučajevi



Slika 4.6: Cjelokupnost tipkovnice

standardnim tipkama, jedan s brojevima i dodatnim znakovima, jedan s vrlo rijetkim znakovima), već je procjena da je za naše potrebe (primarno vrednovanje tipkovnice) dovoljno implementirati 2 rasporeda tipki. To vidimo na slici 4.6.

Za nešto konkretnije implementacijske detalje mogu se pogledati Dodatci A i C.

5. Vrednovanje mehanizama prediktivnog vizualnog isticanja

5.1. Oblikovanje eksperimenta

U nastavku ćemo objasniti pojmove, metode i tehnologije koje su korištene pri oblikovanju i provedbi eksperimenta.

5.1.1. Oblikovanje eksperimenta unutar sudionika i između sudionika

Ako se svako mjerenje ponavlja za svakog ispitanika onda je oblikovanje eksperimenta *within-subjects* tj. unutar sudionika (drugačije rečeno, ispitanici imaju potpuno jednake uvjete pri provedbi eksperimenta te istim redoslijedom obavljaju postupke uključene u eksperiment), dok se kod *between-subjects* tj. između sudionika oblikovanja eksperimenta ispitanici podijele na više grupa i svaka grupa dobiva drugačije uvjete ispitivanja (dakle, ispitanici unutar fiksne grupe imaju iste uvjete ispitivanja, ali između grupa su uvjeti različiti). Ponekad je bolje da je eksperiment između sudionika, a ponekad je bolje da je unutar sudionika. Prednost oblikovanja eksperimenta unutar sudionika je njegova jednostavnost - nije potrebno balansirati grupe jer postoji samo jedna, dok se kod oblikovanja eksperimenta između sudionika potrebno pobrinuti oko balansiranja grupa. Prednost oblikovanja eksperimenta između sudionika je to što smanjuje interferenciju među uvjetima ispitivanja. Interferencija se pojavljuje kada ispitanik prelazi s jednog testa na drugi, a u tom procesu na drugi test prenosi iskustva i znanje koje je stekao u prethodnom.

5.1.2. Učinak poretka, suprotstavljanje, latinski kvadrati

Kao što je već navedeno, kad se koristi oblikovanje eksperimenta unutar sudionika, svakog ispitanika se testira na svakom testu te zbog toga se učinak ispitanika svakim testom može poboljšati ili čak pogoršati u nekim slučajevima. Zbog toga se ne dobivaju mjerodavni rezultati u testovima nakon prvog. Da bi se to spriječilo koristi se metoda suprotstavljanja i latinski kvadrati, odnosno svaki ispitanik dobiva testove u drugačijem poretku (što je onda već suštinski eksperiment između sudionika).

		Subject		
		1	2	3
Order	3	A	B	C
	2	B	C	A
	1	C	A	B

Slika 5.1: Latinski kvadrat reda 3

Slika 5.1 predstavlja latinski kvadrat reda 3. Svako slovo predstavlja test koji ispitanik izvršava. U ovom slučaju svi bi ispitanici bili podijeljeni u 3 grupe i svaki redak bi odgovarao redoslijedu kojim bi pojedina grupa radila zadane testove.

5.1.3. Upitnik

U sklopu eksperimenta dat ćemo i odgovarajući upitnik u kojem će korisnici osim ocjenjivanja mehanizma predikcije moći i dati neko svoje mišljenje o istom.

5.2. Provođenje eksperimenta

5.2.1. Varijable eksperimenta

U nastavku ćemo opisati varijable provedenog eksperimenta i njihovu osnovnu podjelu. U oznaci 7" uočavamo simbol " koji je oznaka za inch (palac).

Nezavisne varijable su:

- Veličina tableta: 7", 10" i mobitela: 5"
- Položaj tableta pri unošenju teksta: portretno (horizontalno), pejzažno (vertikalno)

Zavisne varijable su:

- Brzina unošenja teksta: varijabla pokazuje broj riječi koje ispitanik prosječno unese u jednoj minuti, mjerna jedinica je stoga *wpm* (words per minute).
- Broj pogrešaka (error rate, *ER*): varijabla pokazuje broj pogrešaka koje je ispitanik učinio za vrijeme unosa jednog bloka teksta.

Na kraju je u tablice unijeta samo prva zavisna varijabla (brzina unošenja teksta), koja se izračunala uzevši u obzir *ER* prema sljedećoj formuli:

$$wpm = \frac{60(n - ER)}{t}$$

Gdje *n* predstavlja broj unesenih riječi, *ER* predstavlja broj pogrešaka, a *t* predstavlja vrijeme koje je korisniku bilo potrebno da unese *n* riječi (mjereno u sekundama).

Ostale varijable su:

- Slučajne varijable: tu spadaju karakteristike ispitanika (npr. visina, težina, broj godina, razina obrazovanja, iskustvo u korištenju raznih tipkovnica na tabletima itd.). Ove varijable nisu uzete u obzir pri statističkom računu niti pri odabiru ispitanika.
- Zbunjujuće varijable: u ovom slučaju zbunjujuća varijabla može biti iskustvo i učenje jer ispitanik svakim unosom teksta sistematično postaje sve vještiji pa se predviđa poboljšanje rezultata svakim idućim unosom teksta na istoj tipkovnici.

5.2.2. Grupiranje sudionika u eksperimentu

Imat ćemo tri grupe - jedna će tipkati na 10" tabletu, a druga na 7", treća na mobitelu 5". Svaki redak predstavlja redoslijed za jednog ispitanika unutar pojedina grupe. Značenja simbola su: A - alternativno, S-standardno, h-pejsažno (horizontalno), v-portretno (vertikalno). Standardna tipkovnica koja vrši predikzivno vizualno isticanje je SwiftKey prikazana na slici 5.2, a našu alternativnu tipkovnicu nazvat ćemo Radi-alKey.

Prve dvije grupe (one koja tipkaju na tabletu redom dimenzija 10" i 7" imat će 4 ispitanika kao to je prikazano na tabeli 5.1). Treća grupa (koja tipka na mobitelu) imat će samo 2 ispitanika (kao što je prikazano na tabeli 5.2) i tipkat će se samo vertikalno jer je taj način najčešći za tipkanje na mobilnom uređaju (i zbog nedostatka ljudi na kojima se obavlja testiranje). Da smo pribavili dovoljno ljudi i treća grupa bi tipkala prema tabeli 5.1. Naime, relativno velik broj ljudi se nažalost nije odazvao na poziv na sudjelovanje u eksperimentu. Da bi se mogla provesti još iscrpnija analiza varijance

(koju je moguće provesti primjerice za prvu grupu) potrebno je unutar retka latinskog kvadrata koji čini pojedinu grupu imati više ispitanika koji tipkaju u tome poretku (tj. jedan redak našeg latinskog kvadrata odgovara grupi - većem broju ispitanika). Tome zahtjevu je bilo nemoguće udovoljiti zbog nedostatka ispitanika.

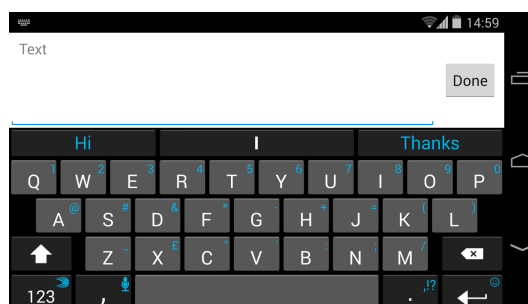
Tablica 5.1: Raspored testiranja 1. i 2. grupe

A+h	A+v	S+h	S+v
S+h	S+v	A+h	A+v
A+v	A+h	S+v	S+ h
S+v	S+h	A+v	A+h

Tablica 5.2: Raspored testiranja 3. grupe

A+v	S+v
S+v	A+v

Podsjetimo se da je ciljni jezik (jezik tipkanja) engleski. Standardna (SwiftKey) tipkovnica ima vrlo dobro razvijen mehanizam kontekstne predikcije, konkretno - kada korisnik tipkač utipka "Where" već se nude kontekstno smislene riječi (čak i sintagme) poput "are, are you, I". Ovaj mehanizam radi čim je tipkovnica instalirana, ali pomoću strojnog učenja mehanizam predikcije poboljšava sam sebe kako upoznaje korisnika i njegov "način tipkanja" (odnosno skupove uzastopnih riječi koje on često koristi). S obzirom na to da procjenjujem da je ovaj mehanizam premoćan; za testiranje koristit ćemo semantički nepovezane riječi (ne čine rečenicu). Vrijedi procijeniti koji broj riječi je optimalan za rezultat eksperimenta. Nakon što sam sebi mjerio potrebno vrijeme pri tipkanju jednog bloka teksta na tipkovnici RadialKey (s alternativnim meh-



Slika 5.2: SwiftKey tipkovnica

anizmom predikcije), došao sam do zaključka da je najbolje da se u pojedinom paru (vrsta tipkovnice, položaj) tipka 25 riječi za što će početnicima vjerojatno trebati oko 3-4 minute. U prve dvije grupe to znači da će pojedini ispitanik utipkati ukupno $4 \cdot 25 = 100$ riječi. U trećoj grupi pojedini ispitanik utipkat će $2 \cdot 25 = 50$ riječi. Izvor riječi (korpus) bit će mrežno sjedište koje testira brzinu tipkanja milijuna korisnika <http://10fastfingers.com/typing-test/english>. Dakle, ponudit ćemo korisnicima riječi koje se izgeneriraju na ovoj stranici pri pokretanju jednog testa brzine tipkanja. Riječi se izgeneriraju upravo u obliku koji je predviđen za ovaj eksperiment (nisu semantički povezane kroz jezičnu strukturu rečenice). Mogli bi se iskonstruirati eksperimenti koji uzimaju u obzir način tipkanja s obzirom na položaj prsta. Primjerice, moguće je testirati tipkanje s dva palca kao i tipkanje kažiprstom dok se tablet drži u drugoj ruci - ovo su doista poprilično jednakovrijedne mogućnosti tipkanja u vertikalnom položaju. Međutim, zbog neizbježnog nastanka kombinatorne eksplozije (latinski kvadrat odnosno broj ispitanika, nam se povećava za faktor dva - eksponencijalno) testiranjem položaja prsta, odlučio sam ispitanicima dopustiti da tipkaju onako kako oni smatraju da će najbrže utipkati zadani tekst.

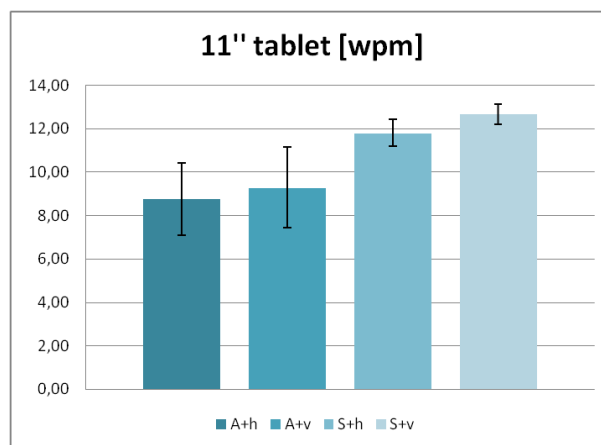
5.3. Prikaz i analiza rezultata

5.3.1. Rezultati grupe koja tipka na 11" tabletu

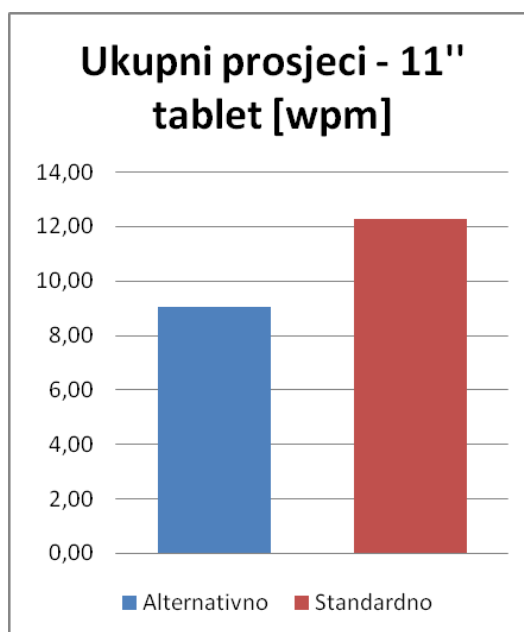
Tabela 5.3 prikazuje prosječnu brzinu unošenja teksta za svakog ispitanika te konačno prosjek za svaki način tipkanja i odgovarajuće standardne devijacije, a slika 5.3 histogram nastao na temelju tablice. Ukupni prosjeci po tipkovnici prikazani su na slici 5.4.

Tablica 5.3: Brzina tipkanja izražena u WPM

Tablet 11"	Alternativno [wpm]		Standardno [wpm]	
Ispitanik	Pejsažno	Portretno	Pejsažno	Portretno
1.	7,98	10,27	12,31	12,50
2.	11,63	11,72	11,36	13,09
3.	7,50	6,82	11,03	13,09
4.	7,98	8,33	12,50	12,00
Prosjek	8,77	9,29	11,80	12,67
St. devijacija	1,66	1,86	0,62	0,46



Slika 5.3: Histogram na temelju tablice



Slika 5.4: Ukupni prosjeci

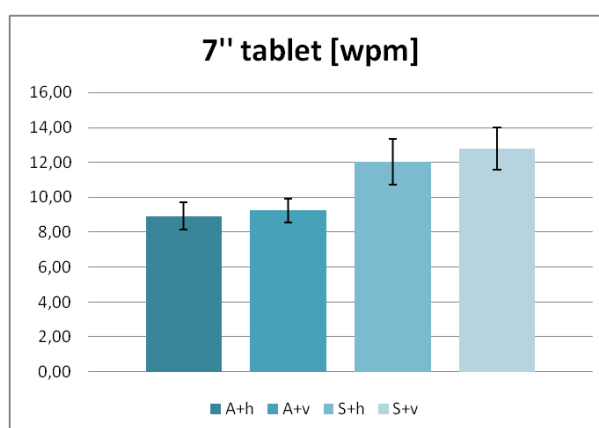
Iz svih priloženih grafova zaključujemo da je na velikom tabletu za korisnike početnike brže tipkati standardnim mehanizmom prediktivnog vizualnog isticanja (za otprilike 3 i pol riječi po minuti). Iz slike 5.3 se jasno vidi koji su način tipkanja najbrži (standardno portretno tj. vertikalno) i koji je najsporiji (alternativno pejisažno tj. horizontalno).

5.3.2. Rezultati grupe koja tipka na 7" tabletu

Tablica 5.4: Brzina tipkanja izražena u wpm

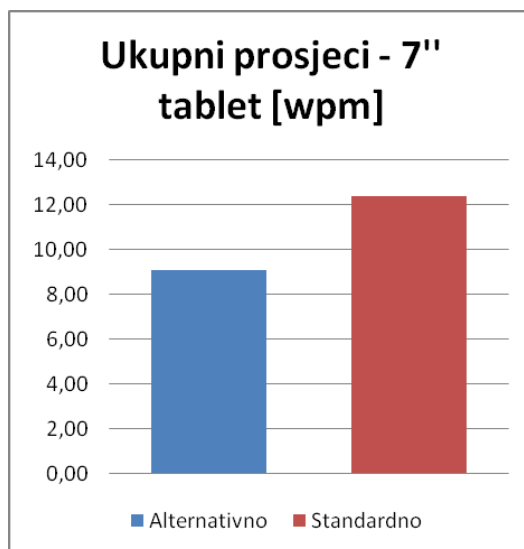
Tablet 7"	Alternativno [wpm]		Standardno [wpm]	
Ispitanik	Pejsažno	Portretno	Pejsažno	Portretno
1.	7,81	8,52	10,99	10,91
2.	9,09	10,07	10,83	13,71
3.	8,82	8,67	14,12	13,98
4.	10,00	9,68	12,10	12,50
Prosjek	8,93	9,24	12,01	12,78
St. devijacija	0,78	0,66	1,31	1,21

Tabela 5.4 prikazuje prosječnu brzinu unošenja teksta za svakog ispitanika te konačno prosjek za svaki način tipkanja i odgovarajuće standardne devijacije, a slika 5.5 histogram nastao na temelju tablice. Ukupni prosjeci po tipkovnici prikazani su na slici 5.6.



Slika 5.5: Histogram na temelju tablice

Vidimo da je odnos brzine unosa riječi jednak kao i kod 11" tableta, ali u većini pozicija (u svakoj osim pejisažno-alternativno) je tipkanje na 7" tabletu nešto brže. To



Slika 5.6: Ukupni prosjeci

je vrlo vjerojatno zato što je dimenzija manjeg tableta puno bliža ljudskim rukama i samim time je tipkanje na takvom tabletu prirodnije i brže.

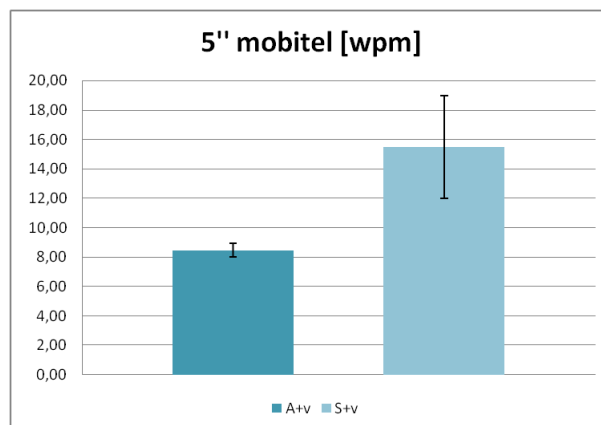
5.3.3. Rezultati grupe koja tipka na 5'' mobitelu

Tablica 5.5: Brzina tipkanja izražena u WPM

Tablet 7''	Alternativno [wpm]	Standardno [wpm]
Ispitanik	Portretno	
1.	7,98	12,00
2.	8,93	18,95
Prosjek	8,46	15,48
St. devijacija	0,47	3,48

Tabela 5.5 prikazuje prosječnu brzinu unošenja teksta za svakog ispitanika te konačno prosjek za svaki način tipkanja i odgovarajuće standardne devijacije. Kod tipkanja na mobitelu smo ispitivali samo portretno (vertikalno) tipkanje.

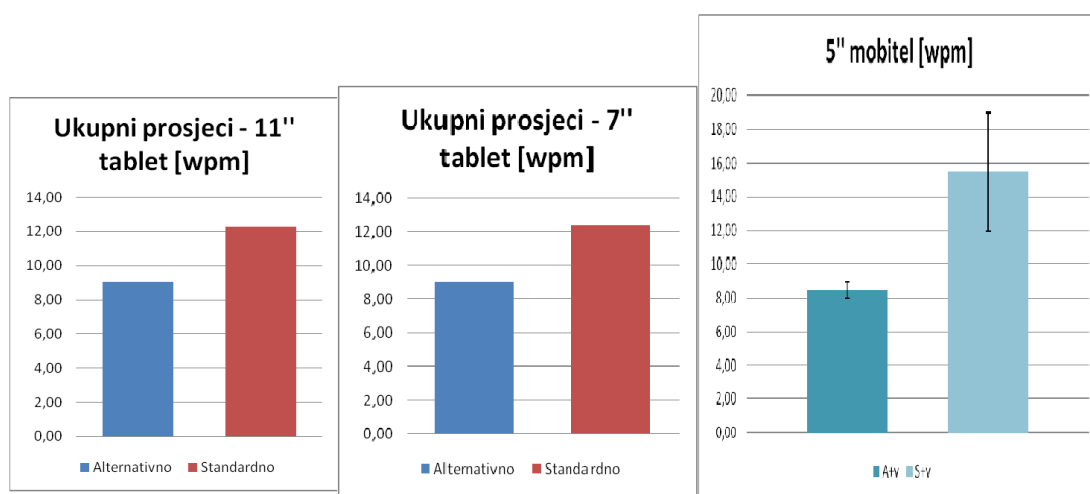
Kao što vidimo na slici 5.7 ovdje je standardni mehanizam predikcije polučio najbolje rezultate (u usporedbi s grupama 1. i 2.), a alternativni najlošije. Standardni mehanizam je ovdje bio najbolji, vjerojatno, zato što su korisnici već najuvježbaniji u tipkanju na standardnoj tipkovnici mobitela, te ih standardni mehanizam predikcije nije previše ometao. Valja napomenuti da ovi rezultati nisu sasvim usporedivi s rezultatima 1. i 2. grupe zato što je u ovoj grupi ispitivano samo portretno (vertikalno)



Slika 5.7: Histogram na temelju tablice

tipkanje.

5.3.4. Usporedba rezultata triju grupa



Slika 5.8: Komparativna analiza

Pogledajmo konačno, naše rezultate u usporednoj perspektivi kako su prikazani na slici 5.8. Dakle, vidimo da je sumarno tipkanje brže 7" tabletu nego na 11" tabletu i na alternativnoj i standardnoj tipkovnici (za poprilično mali faktor). Također neovisno o uređaju na kojemu se tipka, standardno tipkanje brže je od alternativnog. Slika za 5" mobitel je nešto izdužena da bi se dobile sumjerljive skale. Iako je na mobitelu testirano samo vertikalno tipkanje (koje je po našim dosadašnjim rezultatima nešto brže od horizontalnog) alternativni mehanizam predikcije pokazao se nešto sporiji nego na

tabletima (iako ne značajno), dok je standardni mehanizam predikcije pokazao značajno bolje rezultate. Standardni mehanizma predikcije je značajno bolji jer je on vrlo sličan i načinu na koji ispitanici i inače tipkaju (na svojim mobitelima). To što se alternativni mehanizam predikcije pokazao nešto lošijim može se objasniti i efektom učenja, naime s obzirom na to da alternativni mehanizam predikcije ipak zahtjeva neko kratko vrijeme privikavanja, a ispitanici su na 5" mobitelu utipkali samo 25 riječi na alternativnoj tipkovnici, to je dvostruko manje nego ispitanici koji su koristili tablete, pa se i tu može tražiti razlog za nešto manji rezultat *wpm*-a.

5.3.5. Mane i prednosti testiranih mehanizma predikcije

Pokušajmo objasniti koje su prednosti i mane testiranih mehanizma predikcije. Primjerice, želimo utipkati riječ "prijestolonasljednik". Standardni mehanizam predikcije bi vjerojatno tek nakon ručno utipkane riječi "prijestolo" počeo nuditi traženu riječ "prijestolonasljednik", dok bi alternativan mehanizma predikcije nakon utipkanog slova "p" predvidio sufiks "ri", nakon toga bi nakon utipkanog slova "j" i dobivene riječi "prij" ponudio "es" i tako dalje. Dakle, vidimo da ovim mehanizmom predikcije možemo pokrivati veći broj riječi (konkretno, možda je alternativan mehanizam predikcije imao na umu riječ "prije" dok je nudio sufiks "ri" nakon utipkanog slova "p"). Međutim, opet se pokazuje jedna mana ovog mehanizma kada dođemo do sufiksa "prijestolo". Standardni mehanizam ovdje nudi veliku pomoć i nudi prijedlog kojim dokrajči riječ dok je za alternativni mehanizam potrebno još nekoliko puta pritisnuti tipke. U ovom kontekstu, bilo bi zanimljivo promotriti kombinaciju standardnog i alternativnog mehanizma predikcije, ali to bi vjerojatno bila prevelika distrakcija korisniku.

Iako to nije tema rada, odlučili smo kod jednog ispitanika koje je tipkao na vlastitom mobitelu (nešto većem od 5") pogledati kakvi su rezultati pri tipkanju bez ikakvog mehanizma predikcije i *wpm* je ispao otprilike duplo veći nego pri tipkanju s mehanizmima predikcije (što je značajno brže). Također, isti korisnik je na vlastitom mobilnom uređaju testirao unos znakova gestom (path input), te je taj način ispao ne sporiji od normalnog unošenja znakova, ali nešto brži nego koristeći mehanizme predikcije. Iako rezultati koje spominjem u ovom odlomku nisu podvrgnuti formalnom statističkom tretmanu te su kao takvi nelegitimni, ipak se iz njih može zaključiti da je za dobro izvježbanog korisnika vjerojatno najbrži način tipkanja bez ikakvih mehanizama predikcije, premda ostaje otvoreno pitanje bi li standardni mehanizam predikcije ipak bio učinkovitiji da se korisnik izvježba na njemu (kvalitetno napravljenom standardnom mehanizmu predikcije).

5.3.6. Analiza varijance

ANOVA test ili analiza varijance parametrijski je test, što znači da se ravna prema određenoj razdiobi, za razliku od neparametarskih testova koji ne pretpostavljaju o kakvoj statističkoj razdiobi vjerojatnosti se radi. Rezultati ANOVA testa daju se u obliku F-statistike. Tu su nam važni pojmovi nul hipoteze i p vrijednosti. Što je vrijednost p viša, to je vjerojatnije da je naša hipoteza oborena. Drugim riječima, tada dvije 'stvari' koje pokušavamo dovesti u korelaciju nisu korelirane u mjeri u kojoj bi trebale biti da bismo rezultate eksperimenta proglasili validnima (da bi se nul hipoteza potvrdila). Obično se uzima da bi vrijednost p trebala biti 0.05 ili manja, što nam daje veću pouzdanost u ispravnost nul hipoteze.

Ovdje će biti dana analiza varijance za 1. i 2. iz provedenog eksperimenta. Idejno, testirat ćemo:

- postoji li statistički značajna razlika između dva različita položaja držanja tableta (portretno, pejzažno)?
- postoji li statistički značajna razlika u dva načina predikcije (alternativno, standardno)?

Sada ću u vrlo kratkim crtama opisati, ne kako alat iz MacKenzijske knjige suštinski funkcionira, nego što možemo pomoću njega odrediti. Pomoću njega je moguće kroz unos broja redaka (sudionika u eksperimentu), broja različitih instanci prve vrste nezavisnih varijabli (u našem slučaju to znači da možemo tipkati horizontalno i vertikalno), te broja različitih instanci druge vrste nezavisnih varijabli (u našem slučaju to znači da ćemo unijeti da postoje dvije vrste načina predikcije - alternativno i standardno) doznati statističke podatke važne za analizu varijance. Dodatno, alat omogućuje definiranje još jednog broja. Radi se o broju različitih grupa. Svaka grupa obavlja testiranje u eksperimentu svojim fiksnim redoslijedom (ovaj broj se upisuje ako se želi doznati kakav je utjecaj efekta poretka). Mi smo pri samom oblikovanju eksperimenta pretpostavili da postoji određeni efekt poretka i nije nam bio cilj ispitati taj efekt, nego smanjiti njegov potencijalni utjecaj. Iz toga razloga, odlučili smo unijeti podatke u alat, ne uzimajući u obzir grupe. U potencijalnom, unaprijednom eksperimentu, trebalo bi uzeti u obzir grupe i eksperiment oblikovati tako da unutar pojedine grupe ima statistički dostatan broj sudionika.

Testirali smo rezultate 1. grupe, alatom Anova2 (pisanim u programskom jeziku JAVA), uz sljedeće podatke:

- broj sudionika = 4
- broj načina položaja tipkanja = 2 (oznaka $f1$)

- broj različitih tipkovnica = 2 (oznaka $f2$)

U sljedećem bloku zapisani su rezultati za pojedine (standardne) hipoteze koje se postavljaju pri provedbi ANOVA testa. Hipoteza se označava velikim slovom H uz pridruženi indeks hipoteze (npr. nul hipoteza se označava s $H0$).

- $f1 - H0$ = nema razlike u srednjoj vrijednosti wpm -a između s obzirom na položaj tipkanja
Rezultat je : $p = 0.0427$.
- $f2 - H1$ = nema razlike u srednjoj vrijednosti wpm -a s obzirom na različite tipkovnice
Rezultat je : $p = 0.0897$.
- $f1 \times f2 - H2$ = nema razlike u srednjoj vrijednosti wpm -a između s obzirom na uređeni par (tipkovnica, tipkanje)
Rezultat je : $p = 0.7688$.

Samo je vjerojatnost p hipoteze $f1$ manja od 0.05 pa se ona prihvaća (ostale hipoteze se odbacuju). Dakle, zaključuje se da na 10" tabletu položaj tableta nije od statistički značajnog utjecaja na wpm .

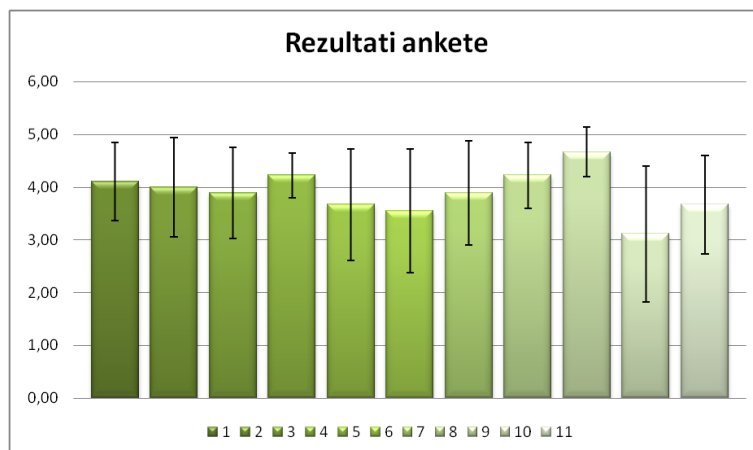
Pogledajmo sada rezultate 2. grupe:

- $f1 - H0$ = nema razlike u srednjoj vrijednosti wpm -a između s obzirom na položaj tipkanja
Rezultat je : $p = 0.0158$.
- $f2 - H1$ = nema razlike u srednjoj vrijednosti wpm -a s obzirom na različite tipkovnice
Rezultat je : $p = 0.3416$.
- $f1 \times f2 - H2$ = nema razlike u srednjoj vrijednosti wpm -a između s obzirom na uređeni par (tipkovnica, tipkanje)
Rezultat je : $p = 0.4791$.

Slično kao i kod 1. grupe, samo je vjerojatnost p hipoteze $f1$ manja od 0.05 pa se ona prihvaća (ostale hipoteze se odbacuju). Dakle, zaključuje se da na 7" tabletu položaj tableta nije od statistički značajnog utjecaja na wpm .

5.3.7. Analiza rezultata upitnika

Rezultati upitnika pokazuju da su ispitanici generalno zadovoljni alternativnom tipkovnicom (tj. alternativnim mehanizmom predikcije) kao što vidimo na slici 5.9, ali je



Slika 5.9: Histogram sa srednjim vrijednostima i standardnim devijacijama ocjena ankete

ANKETA ZA EVALUACIJU KORISNOSTI KORISNIČKOG SUČELJA

		1	2	3	4	5	
1. Korištenje ove tipkovnice bilo je jednostavno.	uopće se ne slažem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	u potpunosti se slažem
2. Osjećao/la sam se ugodno koristeći ovu tipkovnicu.	uopće se ne slažem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	u potpunosti se slažem
3. Navigacija na ovoj tipkovnici je intuitivna i od pomoći.	uopće se ne slažem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	u potpunosti se slažem
4. Organizacija informacija na tipkovnici je jasna.	uopće se ne slažem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	u potpunosti se slažem
5. Čitanje znakova na zaslonu je jednostavno.	uopće se ne slažem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	u potpunosti se slažem
6. Ova tipkovnica ima sve očekivane mogućnosti.	uopće se ne slažem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	u potpunosti se slažem
7. Lako je pronaći traženu informaciju.	uopće se ne slažem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	u potpunosti se slažem
8. Brzo sam naučio/la koristiti ovu tipkovnicu.	uopće se ne slažem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	u potpunosti se slažem
9. Lako se prisjećam kako ju koristiti.	uopće se ne slažem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	u potpunosti se slažem
10. Preporučio/la bih ovu tipkovnicu prijatelju.	uopće se ne slažem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	u potpunosti se slažem
11. Sveukupno, zadovoljan/na sam ovom tipkovnicom.	uopće se ne slažem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	u potpunosti se slažem

Imate li ideju za poboljšanje ovog mehanizma predikcije (dimenzije tipki, broj iskočnih tipaka, duljina sufiksa)? Objasnite.

Slika 5.10: Slika uputnika

svejedno (u skladu s rezultatima eksperimenta) smatraju inferiornom u odnosu na standardnu tipkovnicu. Histogram s prosjekom i standardnim devijacijama nam prikazuje da su korisnici mahom uvjereni da se lako prisjećaju kako se dana tipkovnica koristi.

Najslabije ocjene dobila je kućica u kojoj se pita bi li korisnici danu tipkovnicu preporučili prijatelju. Više od 50% ispitanika nije imalo nikakvu ideju za poboljšanje mehanizma predikcije. Neki jesu, i tu ćemo iznijeti najkonstruktivnije komentare. Jedna od ideja je se približava klasičnom modelu Isokoskog, a tvrdi da bi trebali gestom moći odabrati slovo na skočnom gumbu, ne dižući prst sa zaslona. Jedna vrlo zanimljiva ideja, koja je došla od čak dvojce ispitanika nezavisno, je iterativna ponuda sufiksa skočnim prozorima. Konkretno, nakon što se stisne tipka i pojavi skočni prozor te korisnik odabere neki od ponuđenih sufiksa definiranih skočnim prozorom, njihova ideja je ponovo nad tim odabirom otvoriti skočni prozor i ponuditi sufikse, i tako dalje sve dok korisnik odabire sufikse na skočnim gumbima. Ovo komplicira implementaciju, ali je u svakom slučaju je zanimljiva ideja.

6. Zaključak

U ovome radu bilo je potrebno implementirati alternativan mehanizam vizualnog prediktivnog isticanja i usporediti ga sa standardnim mehanizmom predikcije i vrednovati s obzirom na poznate attribute uporabljivosti. Testiranje smo vršili na korisnicima početnicima. Kroz izradu alternativnog mehanizma predikcije bilo je potrebno savladati određena tehnička znanja o izradi aplikacija za operacijski sustav Android. Prvotno je trebalo doskočiti tome koji algoritam ćemo koristiti za predikciju riječi. S obzirom na to da nije pronađena službena verzija algoritma korištena je autorska implementacija koja se pokazala dovoljno dobrom za potrebe ovog završnog rada i vrednovanja. Uočili smo da postoje verzije standardnog mehanizma predikcije koje vrše vrlo kvalitetnu predikciju tako da se nakon upisane riječi (primjerice, ako govorimo o predikciji za engleski jezik, kada korisnik upiše *I*, odmah mu se ponudi riječ *am*). Također, ovaj mehanizam predikcije se prilagođava korisniku (mehanizmi strojnog učenja), tj. uči na temelju fraza/sintagmi koje korisnik često koristi i onda njih počinje češće nuditi. Vrednovanje je pokazalo da je standardni mehanizam predikcije nešto bolji od alternativnog. To se može objasniti i time da je za alternativni mehanizam ipak potrebno više vrijeme učenja (privikavanja ispitanika na taj mehanizam). Osim toga, iako ograničavanje duljine sufiksa ima određene prednosti (veći ukupan broj riječi koje je moguće predvidjeti), čini se da je uz kvalitetan mehanizam predikcije puno efikasnije nuditi cijele riječi (manji je ukupan broj klikova). Iako korisnici nisu zatvoreni s obzirom na nove mogućnosti interakcije i tipkovnica je dosta dobro ocijenjena u anketi, rezultati vrednovanja zasad ipak pokazuju da je standardni način predikcije bolji od alternativnog.

LITERATURA

Dragutin Ivanec Boris Petz, Vladimir Kolesarić. *Petzova statistika*. Naklada Slap, Zagreb, 2012.

Poika Isokoski. Performance of menu-augmented soft keyboards. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2004. URL http://www.sis.uta.fi/~pi52316/chi2004/paper/maug_manual.html.

I. Scott MacKenzie. *Human Computer Interaction*. Moragan Kaufmann, 225 Wyman Street, Waltham, 2013.

Wikipedia. Pie menu. https://en.wikipedia.org/wiki/Pie_menu, 2016.a. Pristupljeno: 15.06.2016.

Wikipedia. Qwerty. <https://en.wikipedia.org/wiki/QWERTY>, 2016.b. Pristupljeno: 15.06.2016.

Povećanje brzine unosa teksta na zaslonu osjetljivom na dodir prediktivnim vizualnim isticanjem

Sažetak

Tema ovog rada je povećanje brzine unosa teksta na zaslonu osjetljivom na dodir prediktivnim vizualnim isticanjem. Da bismo pristupili tome problemu odlučili smo na korisnicima početnicima usporediti dva mehanizma predikcije. Jedan je standardan mehanizam predikcije (kakvoga koriste neke komercijalne tipkovnice ili su već instalirane pri kupnji uređaja određene marke), a drugi je alternativan temeljen na radu Isokoskog. Iako smo u konačnici za standardnu tipkovnicu uzeli komercijalnu SwiftKey tipkovnicu, u sklopu ovog rada proučeno je i kako samo samostalno implementirati standardni mehanizam predikcije. Za alternativnu tipkovnicu napravljena je autorska implementacija naziva RadialKey. Zatim, kroz razrađenu metodologiju, izvršeno je eksperimentalno vrednovanje ovih dviju tipkovnica na zaslonima različitih veličina (tabletima i mobitelu). Napravljena je statistička obrada rezultata i analizirani su rezultati ankete. Anketirani korisnici dali su kvalitetno mišljenje o vrednovanim mehanizmima predikcije i neke nove ideje za poboljšanje alternativnog mehanizma predikcije.

Ključne riječi: tipkovnica, predikcija, isticanje, vrednovanje, grupiranje, implementacija, eksperiment, histogram, T9 algoritam, sufiks, latinski kvadrat, operacijski sustav Android, riječ, radijalno, ANOVA

Increasing Text Input Rate on Touchscreens by Predictive Highlighting

Abstract

The subject of this work is increasing text input rate on touchscreens by predictive highlighting. To tackle that problem, we decided to test two prediction mechanism on beginner users. One of them is standard prediction mechanism (such mechanism is used on some commercial keyboards or it comes along with the electronic device itself), and the other one is alternative based Isokoski's work. Even though, we eventually took commercial SwiftKey keyboard as our standard keyboard, in the scope of this work, we tried to make the custom implementation of standard prediction mechanism. The alternative keyboard (bases on Isokoski's work) was made and named RadialKey. Then, through an elaborate methodology, we made an experimental evaluation of these two keyboards on different sizes of touch screen. The statistical analysis of results was made. Results of survey were analysed. Through survey, we got quality opinion about evaluated prediction mechanisms, and some new ideas to improve alternative prediction mechanism.

Keywords: keyboard, prediction, highlighting, validation, grouping, implementation, experiment, histogram, T9 algorithm, suffix, Latin square, the operating system Android, word, radial, ANOVA

Dodatak A

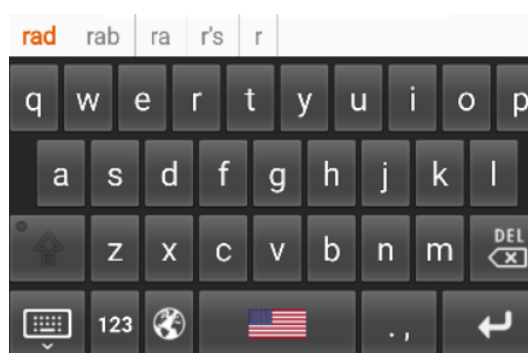
Neki tehnički detalji

Implementacija je napravljena u razvojnom okruženju Android Studio u programskom jeziku JAVA i xml-u (oblikovanje tj. dizajn). Razvoju tastature za operacijski sustav Android (koja nam treba za potrebe vrednovanja) možemo pristupiti na (barem) dva načina. Prvi je da tipkovnicu implementiramo kroz standardni predložak (Keyboard service) za koji je predviđen da se koristi pri implementaciji tipkovnica. Takvu tipkovnicu doista možemo instalirati na uređaj kao tipkovnicu i koristiti ju pri pisanju npr. poruka ili mailova. Drugi pristup je napraviti aplikaciju koja će u sebi sadržavati pogled na tekst (TextView), te vlastitu tipkovnicu te po potrebi druge funkcionalnosti. Za razvoj bilo kakve standardne tipkovnice (pa i one koja može nuditi prijedloge) prikladniji je prvi način, dok je drugi način prikladniji za razvoj alternativne tipkovnice (temeljene na radu Isokoskog).

Dodatak B

Autorska implementacija standardnog mehanizma predikcije

Želimo postići ponašanje slično onome prikazano na slici B.1.



Slika B.1: Autorska implementacija standardnog mehanizma predikcije

Naime, standardni (službeni) mehanizam predikcije prikazuje cijelu riječ koja se nudi, a ja sam pokušao implementirati da se nudi samo sufiks, i to ograničen na duljinu 3. Prva ideja mogla bi biti našu tastaturu (Keyboard layout) staviti u novi raspored (layout), a unutar njega, iznad tastature dodati neki pogled na tekst koji bi prikazivao riječi (koje predviđamo da korisnik želi napisati), međutim, nije zamišljeno da se to tako radi, nego već postoji metoda *onCandidatesView()* koja vraća pogled (View) koju treba implementirati (a to nije tako teško s obzirom na to da već postoje neki službeni primjeri kako bi se otprilike ta metoda mogla implementirati). Nakon što detaljno proučimo danu implementaciju metode *onCandidatesView()* moguće ju je prilagoditi tako da prikazuje prijedloge. To se radi tako da se napravi nova klasa *CandidateView* koja će prihvaćati referencu na klasu u kojoj jest metoda *onCandidatesView()*, odnosno klasu *SoftKeyboard* koja predstavlja našu tipkovnicu te nasljeđuje Androidovu klasu *InputMethodService* i implementira *KeyboardView*. *OnKeyboardActionListener*. U klasi

CandidateView tada možemo generirati željeni izgled postave na kojoj se prikazuju prijedlozi kao i pozvati algoritam koji će nam izgenerirati prijedloge temeljene na trenutno unesenom tekstu. Treba još obratiti pažnju na to da bi se sam tekst koji su izgenerirali prijedlozi "submita" odnosno pojavi na ekranu pritiskom na gumb na kojem se pojavljuje generirani prijedlog sufiksa riječi, za to pak treba malo modificirati metodu *pickSuggestionManually(int index)* koja se nalazi u klasi *SoftKeyboard*. Detaljnije u implementacijske detalje (kao niti u vrednovanje) ovog mehanizma predikcije nismo ulazili jer to nije središnja tema ovog rada.

Dodatak C

Implementacija alternativnog mehanizma predikcije - bitni isječki koda

Građenje cirkularnog menija, odnosno dodavanje (attachiranje) pogleda na tekst na gumba redom

```
circleMenus[i] = new FloatingActionMenu.Builder(getActivity())
    .setStartAngle(0) // A whole circle!
    .setEndAngle(360)
    .setRadius(((int)(px_h * 1.1f)) //radius
    .setAnimationHandler(new SlideInAnimationHandler())
    .addSubActionView(textViews[i][0])
    .addSubActionView(textViews[i][1])
    .addSubActionView(textViews[i][2])
    .addSubActionView(textViews[i][3])
    .addSubActionView(textViews[i][4])
    .attachTo(buttonArray[i])
    .build();
```

Definiranje što će se dogoditi pritiskom na gumb

```
circleMenus[i].setChangeListener(new FloatingActionMenu.MenuChangeListener() {
    @Override
    public void onMenuOpened(FloatingActionMenu menu) {
        for (int k=0; k<26; k++){
            circleMenus[k].close(true);
        }

        Button curr = (Button) menu.getActionView();
        String buttonText = curr.getText().toString();
        tv.append(buttonText);
        String tvS = tv.getText().toString().toLowerCase();
        String arr []= tvS.split("\\s+");
        String lastWord = arr[arr.length - 1];
        int index = buttonText.toLowerCase().toCharArray()[0] - 'a';

        List<String> ponuda = MyT9.getSuggestions(lastWord);
        //much more code follows
    }
});
```

Uočimo da se u zadnjoj liniji poziva autorska implementacija T9 algoritma.

Struktura podataka koja nam je bila od pomoći u autorskoj implementaciji samog algoritma predikcije (T9), opisano u poglavlju Algoritam

```
public static class ParSI implements Comparable<ParSI>{
    String s1;
    Integer s2;

    public ParSI(String _s1, Integer _s2) {
        s1 = _s1;
        s2 = _s2;
    }

    @Override
    public boolean equals(Object obj) {
        if (!(obj instanceof ParSI))
            return false;
        ParSI other = (ParSI) obj;
        return this.s1.equals(other.s1) && this.s2.equals(other.s2);
    }

    @Override
    public int compareTo(ParSI drugi) {
        return this.s2.compareTo(drugi.s2);
        //samo po broju usporedimo
    }
}
```

Ova struktura podataka omogućava da uz samo riječ (String) pamtimo i njezinu frekvenciju (koliko je česta) u komunikaciji.

Dio u binarnoj pretrazi za vrijeme autorske implementacije samog algoritma predikcije (T9), opisano u poglavlju Algoritam

```
while (down != up){
    mid = (down + up)/2;

    String currWord = svi.get(mid);

    if (currWord.startsWith(searchWord)){
        //vrijeme je da startamo, tbc
        //prvo ovog gledamo
        if (currWord.length() > searchWord.length()){

            //much more code follows

        }

        else if (currWord.compareTo(searchWord) > 0){
            up = mid;
        }
        else {
            down = mid + 1;
        }
    }
}
```

Dio u konstruktoru kružnog menija (inicijalizacija objekta)

```
public FloatingActionMenu(final View mainActionView,
                          int startAngle,
                          int endAngle,
                          int radius,
                          List<Item> subActionItems,
                          MenuAnimationHandler animationHandler,
                          boolean animated,
                          MenuStateChangeListener stateChangeListener,
                          final boolean systemOverlay) {
    this.mainActionView = mainActionView;
    this.startAngle = startAngle;
    this.endAngle = endAngle;
}
```

```
this.radius = radius;  
this.subActionItems = subActionItems;  
this.animationHandler = animationHandler;  
this.animated = animated;  
this.systemOverlay = systemOverlay;  
// The menu is initially closed.  
this.open = false;
```

Ovo je kod je preuzet iz projekta otvorenog koda na <https://github.com/oguzbilgener/CircularFloatingActionMenu> koji je pod MIT-ovom licencom.

Dodatak D

Upitnik korišten za vrednovanje mehanizama predikcije

ANKETA ZA EVALUACIJU KORISNOSTI KORISNIČKOG SUČELJA

		1	2	3	4	5	
1. Korištenje ove tipkovnice bilo je jednostavno.	uopće se ne slažem						u potpunosti se slažem
2. Osjećao/la sam se ugodno koristeći ovu tipkovnicu.	uopće se ne slažem						u potpunosti se slažem
3. Navigacija na ovoj tipkovnici je intuitivna i od pomoći.	uopće se ne slažem						u potpunosti se slažem
4. Organizacija informacija na tipkovnici je jasna.	uopće se ne slažem						u potpunosti se slažem
5. Čitanje znakova na zaslonu je jednostavno.	uopće se ne slažem						u potpunosti se slažem
6. Ova tipkovnica ima sve očekivane mogućnosti.	uopće se ne slažem						u potpunosti se slažem
7. Lako je pronaći traženu informaciju.	uopće se ne slažem						u potpunosti se slažem
8. Brzo sam naučio/la koristiti ovu tipkovnicu.	uopće se ne slažem						u potpunosti se slažem
9. Lako se prisjećam kako ju koristiti.	uopće se ne slažem						u potpunosti se slažem
10. Preporučio/la bih ovu tipkovnicu prijatelju.	uopće se ne slažem						u potpunosti se slažem
11. Sveukupno, zadovoljan/na sam ovom tipkovnicom.	uopće se ne slažem						u potpunosti se slažem

Imate li ideju za poboljšanje ovog mehanizma predikcije (dimenzije tipki, broj iskočnih tipaka, duljina sufiksa)? Objasnite.

Slika D.1: Slika uputnika