Studiu privind starea artei: Alinierea transcrierilor aproximative cu semnalul de vorbire

Cristian Manolache Dan Oneață
Horia Cucu Corneliu Burileanu Dragoș Burileanu

Cuprins

Acest document prezintă starea artei pentru utilizarea transcrierilor aproximative în procesul de adnotare automată a bazelor de date de vorbire. Adnotarea automată este un pas standard în sarcina de auto-învățare (en., *self-training*; Triguero et al. 2015), în care urmărim să îmbunătățim un sistem de recunoaștere automată a vorbirii (RAV) utilizând adnotările pe care acesta le produce pentru baze de date de vorbire neadnotate. Ideea metodelor bazate pe texte aproximative este de a filtra transcrierea obținută automat folosind textul aproximativ care însoțește semnalul audio: partea din transcriere care se potrivește cu textul adiacent se presupune corectă și este folosită pentru re-antrenarea sistemului de recunoaștere automată a vorbirii.

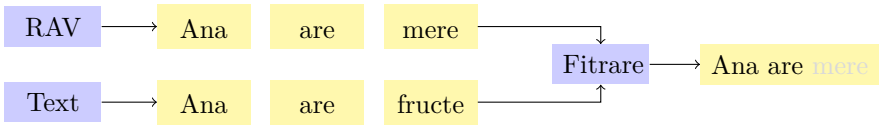

Figura 1: Exemplu de folosire a transcrierilor aproximative ("text") pentru a selecta o parte din transcrierea sistemului de recunoaștere automată a vorbirii (RAV). Porțiunea selectată împreună cu semnalul vocal aferent pot fi folosite pentru reantrenarea sistemului de RAV.

Documentul este structurat astfel. Secțiunea 1 motivează și prezintă sarcina de aliniere a transcrierilor aproximative pentru aplicațiile de recunoaștere automată a vorbirii. Secțiunea 2 descrie principalele metode din literatură care tratează această sarcină. Secțiunea 3 prezintă metode generale de aliniere, metode care se folosesc și în mulți din algoritmii de menționați în secțiunea 2. Secțiunea 4 introduce sarcina normalizării textului, problemă cu care ne confruntăm când dorim să aliniem un text cu vorbire.

1 Introducere

Vorbirea este adesea însoțită de informație textuală; spre exemplu: *(i)* multe discursuri sunt rostite pe baza unui script; *(ii)* filmele sau emisiunile de televiziune au subtitrări asociate; *(iii)* reportajele de pe site-urile de știri vin cu o scurtă descriere jurnalistică. Aceste exemple le numim *transcrieri aproximative* și au avantajul că sursele de date (înregistrări audio și text) fie există, fie sunt mai ușor de procurat decât transcrierile standard, care sunt utilizate pentru antrenarea sistemelor de vorbire. Acest proiect investighează tocmai utilizarea acestor surse text existente în procesul de obținere a bazelor de date pentru antrenare, prin generarea de noi adnotări prin alinierea semnalului audio cu bucățile de text existente (Figura 1).

Transcrierile aproximative, deși mai ușor de obținut, au potențialul dezavantaj de a fi mai zgomotoase decât transcrierile standard. Cele mai frecvente dificultăți sunt lipsa alinierii temporale (nu știm cărei secvențe audio îi corespunde o anumită secvența text) și cuvinte lipsă sau în plus (textul existent nu corespunde fidel mesajului rostit). Acuitatea problemelor variază în funcție de domeniu, la un capăt sunt filmele și subtitrările, destul de bine aliniate și transcrise, la celălalt transcrierile jurnalistice pe baza unor știri, adesea sunt incomplete, cu text extra care nu este rostit și fără nici un fel de informații

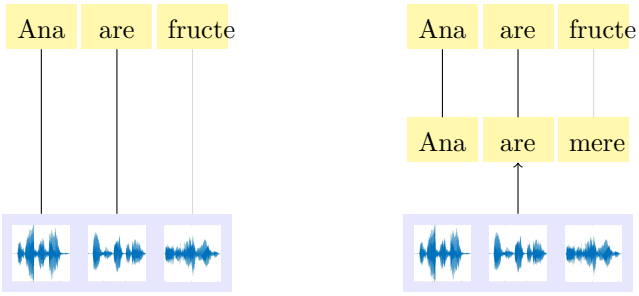

Figura 2: Exemplificarea tipului de *semnal de aliniat*: textul aproximativ este aliniat direct cu semnalul audio (stânga); textul aproximativ este aliniat cu transcrierea sistemului de RAV și prin aceasta se face alinierea cu semnalul audio (dreapta).

temporale.

Cantitativ, Braunschweiler et al. (2010) raportează o eroare la nivel de cuvânt (en., *word error rate*; WER) de 5.4% pentru audio-book-uri, comparând textul cărții cu transcrierea manuală a materialului audio. Iar Hazen (2006) observă că și transcrierile obținute cu servicii comerciale au erori la nivel de cuvânt destul de ridicate, de aproximativ 10% WER; multe din aceste erori sunt datorate faptului că transcrierile sunt mai curate decât vorbirea rostită în mod spontan, care conține greșeli precum cuvinte repetate sau inserții precum “ok” sau “ah”.

2 Alinierea transcrierilor aproximative

Această secțiune descrie principalele metode care folosesc transcrieri aproximative. Am delimitat trei direcții principale după care grupăm articolele:

- *Calitatea transcrierilor* care poate varia de la imprecisă (știri și site-uri web) până la bună (subtitrări și filme) sau foarte bună (cărți citite).
- *Alinierea* între transcrierea aproximativă și semnalul vocal; aceasta se poate face la nivel de cuvânt sau propoziție.
- *Semnalul de aliniat* cu transcrierea aproximativă; acesta poate fi direct audio (Figura 2, stânga) sau o variantă intermediară, textul transcris (Figura 2, dreapta).

Articol	Calitate transcrieri	Nivel de aliniere	Durata semnal	Semnal de aliniat
Stan et al. (2016)	★★★★★	propoziție	ore	audio
Moreno and Alberti (2009)	★★★★★	propoziție	minute-ore	audio
Tao et al. (2010)	★★★★★	propoziție	ore	audio
Lecouteux et al. (2012)	★★★★★	cuvânt	minute	audio
Hazen (2006)	★★★★★	cuvânt	ore	audio + text
Braunschweiler et al. (2010)	★★★★★	propoziție	secunde	text
Cardinal et al. (2005)	★★★★★	propoziție	minute	text
Liao et al. (2013)	★★★★★	cuvânt	minute	text
Buzo et al. (2013)	★★★★★	cuvânt	minute	text

Tabelul 1: Sumar al principalelor metode de aliniere din literatură din punct de vedere al: *(i)* calității transcrierilor, *(ii)* nivelului la care se efectuează alinierea, *(iii)* duratei semnalului de vorbire aliniat, *(iv)* procesării semnalului de vorbire.

Tabelul 1 folosește direcțiile enunțate pentru a sumariza metodele discutate în lucrare. Ca o privire de ansamblu, am putea spune că la un capăt al spectrului avem metode bazate pe transcrieri precise și pentru care alinierea se face la nivel audio iar semnalul audio este lung, iar la celălalt capăt al spectrului avem metode bazate pe transcrieri imprecise și pentru care alinierea se face pe baza transcrierii. În prima categorie intră și metode care tratează sarcina de aliniere forțată (en. *forced alignment*) – alinierea semnalului vocal cu o transcriere care se presupune perfectă. Metodele de aliniere forțată nu le-am detaliat în acest raport, pentru că dorim să ne concentrăm pe cazurile în care transcrierile nu sunt perfecte. Metodele actuale sunt robuste la mici erori ale textului aproximativ. Un exemplu este metoda lui Liao et al. (2013) care folosește transcrierile utilizatorilor pentru video-urile de pe YouTube pentru îmbunătățirea serviciului de recunoaștere vocală. Procedura lor folosește un pas de aliniere între transcrierile oferite de utilizatori și transcrierile generate automat, iar bucățile care sunt în acord sunt utilizate pentru re-antrenarea sistemului RAV.

2.1 Metode de aliniere text-text

Această secțiune descrie metodele de aliniere ce utilizează în prima fază un sistem de recunoaștere automată a vorbirii (RAV) pentru a obține o ipote-

ză de transcriere, urmând ca în faza a doua să alinieze această ipoteză cu transcrierea aproximativă. În final, alinierea cu semnalul audio se realizează pe baza ștampilelor de timp generate tot de sistemul de RAV.

Braunschweiler et al. (2010) realizează un studiu în care alinierea (audio-transcrieri aproximative) se aplică unor cărți audio, materiale pentru transcrierea aproximativă o reprezintă cartea și este, în consecință, de foarte bună calitate. Transcrierea materialului audio se realizează într-un scenariu semi-supervizat:textul aferent cărților audio este utilizat în vederea antrenării unui model de limbă ce va ajuta procesul de RAV pentru partea audio. Modelul de limbă pentru RAV este adaptat la textul aferent cărții audio pentru a obține o ipoteză RAV cât mai corectă. Transcrierea aproximativă este normalizată atât pentru a putea fi utilizată pentru crearea modelului de limbă, cât și pentru aliniere ulterioară cu ipoteza RAV (secțiunea 4 detaliază metodele de normalizare).

Textul generat de RAV și transcrierea aproximativă (normalizată) sunt aliniate prin găsirea celei mai lungi subsecvențe comune. Alinierea aceasta este apoi folosită pentru a împarți secvențele de cuvinte recunoscute în propoziții conform textului din carte. Ulterior, se calculează ratele de eroare la nivel de cuvânt (en., *word error rate*; WER) pentru fiecare propoziție aliniată în vederea luării unei decizii privind selectarea propoziției (propoziția este selectată dacă obține un WER sub un anumit prag). În cazul în care o propoziție este selectată, secvența de cuvinte obținută în urma RAV este înlocuită cu textul din carte.

Metoda propusă de Braunschweiler et al. (2010) este evaluată folosind o secvență audio transcrisă și verificată manual. Autorii raportează, în general, o acuratețe ridicată în identificarea limitelor temporale ale propozițiilor selectate și un WER extrem de mic. În funcție de pragul ales pentru selecție, se obțin transcrieri mai corecte pentru o parte mai mică din textul original sau transcrieri mai puțin corecte pentru o parte mai mare din material. De exemplu, pentru un prag de 100% (cel mai exigent) s-au extras 65.4% din propoziții, cu WER de 0.2%, în timp ce cu un prag de 50% s-au extras 92.1% din propoziții, cu WER de 0.5%.

Cardinal et al. (2005) are ca scop crearea unei baze de date de antrenare dintr-un set de programe de știri înregistrate pentru care transcrierile sunt imprecise. Ca și punct de plecare s-a folosit un fișier audio lung (o ora și jumătate sau mai mult) și unul sau mai multe seturi de texte, care ar putea sau nu sa fie asociate cu un segment audio. Abordarea din acest articol cuprinde patru module:

1. Modulul de RAV – produce o transcriere însoțită de ștampile de timp pentru fișierul audio;

2. Modulul de aliniere – aliniază toate transcrierile aproximative candidate cu transcrierea RAV;

3. Modulul de analiză – examinează fiecare aliniere rezultată la pasul anterior pentru a stabili acceptarea sau rejecția la nivelul fiecărui segment de text;

4. Modulul de adaptare acustică – reantrenează modelul acustic folosind segmentările selectate pentru a îmbunătăți sistemul de RAV.

Alinierea propriu-zisă a transcrierii aproximative cu ipoteza RAV se realizează prin calculul unei distante care reprezintă costul minim necesar transformării unui șir de cuvinte în celălalt șir prin operații de inserție, ștergere, înlocuire și potrivire de cuvinte. Procedura alinierii fiecărui text cu o transcriere a fișierului audio generată automat prezintă rezultate promițătoare. Autorii nu abordează problema normalizării textului.

Selecția segmentelor aliniate se realizează impunând mai multe constrângeri. Un segment este rejectat dacă: *(i)* numărul de cuvinte componente este mai mic decât un prag t_c; *(ii)* numărul de cuvinte care se potrivesc este mai mic decât un prag t_m; *(iii)* numărul de ștergeri este mai mare decât un prag t_d. De asemenea, sunt rejectate segmentele ale căror limite temporale sunt în contradicție cu cele ale segmentelor vecine.

Analiza performanțelor metodei se realizează calculând rata de falsă rejecție a segmentelor de text (un segment nu se află în segmentarea finală deși ar trebui să fie) și acuratețea segmentării. Acuratețea segmentării se referă la faptul că un text este considerat aliniat corect dacă timpul de start se află după finalul segmentului anterior și înaintea începutului segmentului următor. Experimentele realizate pe un set de programe de știri înregistrate cu texte asociate indică o acuratețe de segmentare de 80% cu o rată de falsă rejecție de 30%. Experimentele arată, de asemenea, că procesul de filtrare identifică corect și rejectează segmentele de text ce nu se află în înregistrare.

Liao et al. (2013) antrenează în mod semi-supervizat o rețea neurală adâncă (en., *deep neural networks*; DNN) pentru a crea un model acustic folosind clip-uri de pe YouTube cu transcrieri încărcate de utilizatori. Primul pas al metodei este unul de pre-procesare – se filtrează transcrierile care conțin nepotriviri de limbă și text:

- *Nepotriviri de limbă.* Folosind un detector de limbă bazat pe text, porțiunile din transcrieri care conțin text într-o altă limbă decât limba țința nu sunt luate în considerare. Acest lucru se întâmpla deoarece se încearcă o fonetizare a cuvintelor din text, chiar dacă cuvântul transcris există în limba țintă sau nu. Astfel textul trebuie să conțină cuvinte doar în limba țintă.

- *Nepotriviri probabile de text.* Multe dintre transcrierile încărcate de utilizatori conțin reclame sau URL-uri care nu au legătura cu conținutul audio al clipului și deci trebuie îndepărtate. De asemenea s-au folosit și câteva filtre adiționale precum un filtru pentru detecția caracterelor non-ASCII.

Ca urmare a acestor filtre, dintr-un total de 26.000 de ore de video-uri cu transcrieri sunt selectate doar 14.000 de ore.

Următorul pas constă în filtrarea transcrierilor încărcate de utilizatori. Mai întâi porțiunea audio este transcrisă folosind un model de limba de tip trigram, construit pe baza transcrierii aproximative. Cele două texte sunt apoi aliniate minimizând distanța de editare (sau Levenshtein; vezi secțiunea 3), iar cuvintele care au o distanță nulă primesc un *scor de încredere* de unu, iar restul primesc un scor de zero. Filtrarea se realizează pe baza acestei măsuri binare: dacă cel puțin N cuvinte consecutive din transcriere au un scor de încredere de unu, atunci s-a detectat o *insula de încredere* și aceasta este selectată mai departe. Datorită cantității mari de date disponibile, s-a stabilit că cea mai practică abordare este o filtrare agresivă: folosind o insulă de încredere de lungime minima 50 de cuvinte pe setul de 14.000 de ore de segmente video trecute prin filtrele descrise mai sus, s-au obținut 1.450 de ore de segmente video de "înaltă încredere". Analiza curbei ROC corespunzătoare acestui filtru, folosind transcrieri corupte artificial, arată că pentru $N = 50$, rata de falsă acceptare este sub 10%.

Această procedură a fost aplicată pe setul de testare `YtiDev11`, obținându-se un WER de 38.7%; dintre aceste erori, 76% sunt ștergeri. Setul de testare `YtiDev11` conține video-uri de pe YouTube cu număr mare de vizualizări. Aceste video-uri tind să fie foarte variate datorită vorbirii spontane, zgomotului și muzicii.

În (Buzo et al., 2013) este prezentată o metodă de aliniere a transcrierilor aproximative cu semnalul de vorbire pentru materiale ce conțin știri, interviuri, talk show-uri, etc. Datele folosite în această lucrare sunt obținute automat de pe pagini web ce conțin materiale video și transcrierile text

aproximative. Gradul de fidelitate al transcrierilor este relativ redus: există zone lungi de audio care nu au corespondent în transcriere, precum și unele zone de text care nu se regăsesc în materialul vorbit. De asemenea, în unele cazuri textul este reformulat (relativ la materialul vorbit) pentru a fi corect din punct de vedere gramatical, a nu conține ezitări, bâlbe sau alte efecte ale vorbirii spontane. În cele mai multe cazuri transcrierile aproximative sunt scrise fără diacritice. Autorii estimează că diferența dintre transcrierile aproximative și o eventuală transcriere exactă a materialului vorbit, exprimată ca WER, este de peste 30%.

Datele descărcate automat de pe Internet sunt pre-procesate astfel: materialele audio sunt filtrate pentru a elimina zonele ce conțin muzică, reclame sau alte elemente nedorite și a păstra strict zonele ce conțin vorbire; în materialele text sunt restaurate diacriticele, numerele scrise cu cifre sunt transformate în text, abrevierile sunt expandate, caracterele speciale sunt fie eliminate, fie înlocuite cu secvențe de cuvinte în funcție de cum aceste caractere sunt pronunțabile sau nu.

Procedura generală de lucru este cea standard:

- un sistem de RAV cu model de limbă adaptat la domeniul respectiv (dar fără a folosi transcrierile aproximative) transcrie materialul audio;
- ipotezele de RAV sunt aliniate cu transcrierile aproximative;
- zonele de text aliniate sunt asociate cu zone de material audio pe baza ștampilelor de timp generate de sistemul de RAV;
- materialele audio și textul asociat sunt utilizate la reantrenarea sistemului de RAV.

Lucrarea vine cu inovații la nivelul metodelor de aliniere. Autorii propun reprezentarea alinierii dintre transcrierea aproximativă și ipoteza de RAV sub forma unei matrice $N \times M$, unde N este numărul de cuvinte din ipoteza de RAV și M este numărul de cuvinte din transcriere. În matrice se reprezintă cu valoarea 1 cuvintele potrivite corecte și cu 0 nepotrivirile de cuvinte. În continuare matricea este procesată ca o imagine alb-negru. Astfel, potrivirile de secvențe de cuvinte vor fi reprezentate în imagine de linii diagonale albe. Metoda de procesare de imagini propusă își propune identificarea de diagonale cât mai lungi.

Autorii evaluează metoda propusă din mai multe perspective. În primul rând se pune problema cantității de date aliniate, autorii afirmând că au

reușit alinierea a 19.6%, 30.2%, respectiv 32.0% din materialele audio după aplicarea iterativă (3 iterații) a metodei propuse. În al doilea rând autorii evaluează gradul de scădere a erorilor RAV pe măsură ce datele aliniate sunt utilizate la antrenare. Din acest punct de vedere, sunt raportate erori (WER) de 55.5% pentru sistemul RAV inițial, respectiv 50.3%, 47.2% și 46.3% pentru sistemele reantrenate după cele trei iterații de aplicare a metodei.

2.2 Metode de aliniere text-audio

Lecouteux et al. (2012) prezintă un sistem de recunoaștere ghidat de transcrieri imperfecte. Scopul acestei abordări este de a îmbunătății aceste transcrieri imperfecte. Metoda detectează insule de transcriere (en., *transcript islands*) în semnalul de vorbire, care sunt apoi utilizate pentru a ghida sistemul de recunoaștere a vorbirii. Insulele de transcriere sunt segmente scurte din transcrierea aproximativă care se potrivesc exact cu vorbirea pronunțată.

Scopul metodei este de a folosi transcrierile imperfecte, fără a avea informația de timp disponibilă pentru localizarea insulelor de transcriere. Sistemul profită de transcrierile aproximative atâta timp cât acestea se potrivesc cu conținutul vorbirii și schimbă în modul de recunoaștere liberă când observațiile acustice nu se potrivesc cu transcrierea sugerată. Aceasta metodă permite folosirea de știri întregi fără pre-segmentare. Astfel, problema principală a fost de a integra în sistemul de recunoaștere de vorbire un modul de identificare care sa decidă în fiecare nod din graful de căutare momentul în care sistemul de recunoaștere trece peste o insulă de transcriere. Figura 3 prezintă grafic acești pași.

Metoda propusă constă în mare parte într-un algoritm de decodare ghidat care combină recunoașterea asincronă cu alinierea transcriere-semnal. Rezultatele au demonstrat eficacitatea acestei tehnici, deși calitatea transcrierilor obținute este mică: îmbunătățirea ratei de eroare la nivel de cuvânt este între 28% și 40% comparativ cu transcrierea inițială anterioară.

În Moreno and Alberti (2009) se pune problema alinierii înregistrărilor de vorbire lungi cu transcrierile lor respective. Înregistrările folosite în experimentele din acest articol constau în emisiuni și cursuri în engleză, extrase de pe YouTube, împreună cu transcrieri profesionale. Metoda propusă constă în folosirea unui automat cu stări finite, acesta fiind o metodă de reprezentare a subșirurilor de caractere dintr-un șir.

Pentru realizarea experimentelor s-a folosit infrastructura Google speech indexing. Clipurile audio sunt segmentate în unități mai mici, fiind împărțite

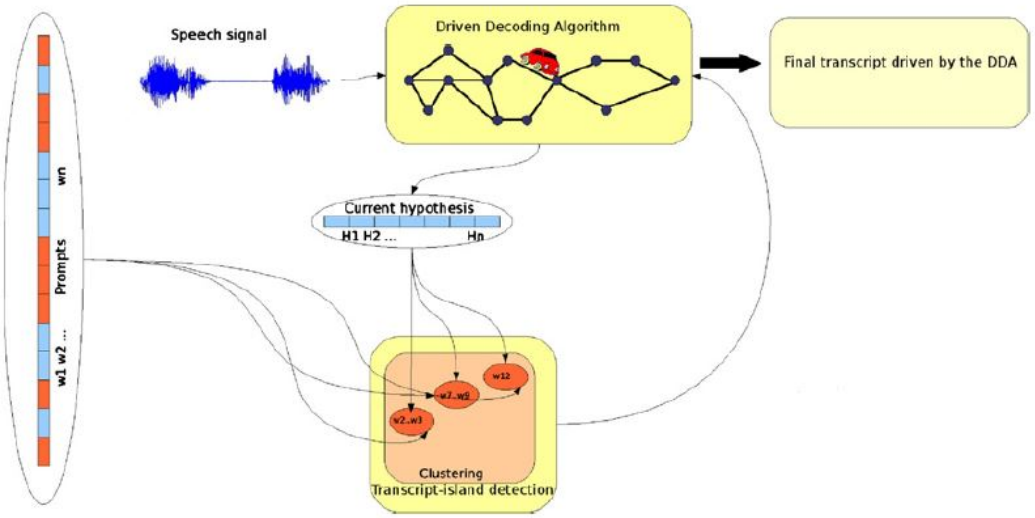

Figura 3: Schema procesului de transcriere utilizat în (Lecouteux et al., 2012).

după momentele de liniște sau alte puncte de întrerupere. De asemenea, segmentele care conțin prea mult zgomot sau muzică sunt respinse. Segmentele audio care nu sunt respinse de către componenta de segmentare sunt trimise către sistemul de recunoaștere de vorbire Google cu vocabular mare.

Pentru alinierea audio, componenta de recunoaștere folosește 2 module. Primul modul, de pronunție, generează entități în dicționar pentru cuvinte din afara vocabularului găsite în transcriere. Cel de-al doilea modul construiește un model de limbă bazat pe transcriere.

Componenta de segmentare și cea de recunoaștere sunt încapsulate într-o arhitectură de server replicată. Un client trimite către server clipul audio și transcrierea corespunzătoare, după care serverul creează ipoteza RAV. Folosind ipoteza și transcrierea, se realizează alinierea, care este trimisă înapoi la client. Rata de eroare este măsurată comparând adnotările manuale cu ipoteza RAV returnată. În aceste experimente, deoarece transcrierile primite sunt de încredere, informația de aliniere în timp produsă de sistem este de obicei precisă atunci când există o potrivire între transcriere și ipoteză. Astfel, WER în acest articol reprezintă o aproximare a numărului de cuvinte aliniate corect.

Ca rezultate, pe un video de 10 minute de știri s-a obținut un WER de

5.4% folosind n-gram de ordin 4, 5 și 6 iar pentru automatul cu stări finite s-a obținut 2.5%. În cazul unor clipuri video cu WER mai mic de 10%, între 10% și 20% și peste 20%, orice model n-gram de ordin 4 sau mai mare au produs un rezultat identic. Performanțele dintre n-gram și factor automaton sunt similare, deși factor automaton este mult mai rapid. În cazul segmentelor audio zgomotoase, ambele metode întâlnesc probleme.

Stan et al. (2016) descrie tool-ul ALISA, care implementează o metodă semi-supervizată pentru alinierea vorbirii la nivel de propoziție cu transcrieri aproximative. Scopul acestui tool este de a crea corpus de vorbire dintr-o multitudine de resurse. Aceste resurse constau în materiale audio găsite pe internet, majoritatea fiind înregistrate într-un mediu profesional sau semi-profesional (de exemplu cărți audio, cursuri video, bloguri video, buletine de știri). Testele/experimentele sunt realizate pe cărți audio în engleză și franceză. Metoda poate fi aplicată numai în cazul în care transcrierile aproximative au un grad foarte înalt de fidelitate cu materialul audio.

Procesul de aliniere (vorbire-transcriere aproximativă) are ca scop generarea unor ștampile de timp pentru propozițiile din textul aproximativ. ALISA folosește o abordare în doi pași pentru alinierea vorbirii cu transcrieri imperfecte: *(i)* segmentarea vorbirii la nivel de propoziție (folosind un detector a activității vocale bazat pe GMM); *(ii)* alinierea vorbirii și a textului la nivel de propoziție (folosind un aliniator de vorbire foarte constrâns bazat pe grafeme). Pasul de segmentare folosește două GMM-uri antrenate pe segmente de vorbire și de liniște. GMM reprezintă modele statistice simple, care au puterea de a face diferența între vorbire și liniște. După ce momentele de liniște sunt etichetate, se face diferența între segmentele de liniște din și dintre propoziții, cele din propoziții nefiind luate în considerare. În etapa de aliniere se folosesc iterativ modele acustice auto-antrenate și rețele de cuvinte foarte constrânse construite din resursele de text disponibile pentru a folosi un decodor Viterbi la nivel de grafeme pe datele de vorbire. Datorită corelației mari dintre text și vorbire, modelul de limbă este "ghidat" de transcriere. Astfel, se obține o rețea de cuvinte foarte constrânsă numită și *skip network*. Skip networks sunt un mod de a specifica modelul de limbă și sunt construite pe baza textului aferent (textul cărții audio) și au rolul de a constrânge procesul de decodare al HMM-ului (decodarea Viterbi) să producă secvențe care apar în text. Figura 4 prezintă grafic metoda ALISA.

Folosind 10 minute de vorbire transcrisă manual, a fost antrenat un set inițial de modele acustice la nivel de grafeme, G0-ML. Aceste modele sunt folosite împreuna cu skip networks pentru a efectua o primă trecere prin date,

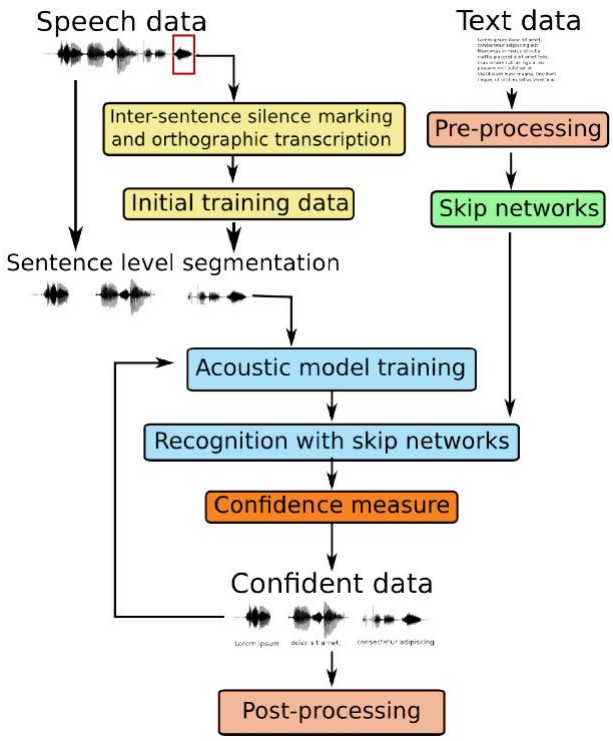

Figura 4: Diagrama metodei ALISA (Stan et al., 2016).

obținând un set de alinieri. Aceste alinieri sunt folosite ca date de antrenare pentru o nouă iterație de creere de model acustic și aliniere de date, G1-ML. Rezultatele metodei sunt prezentate în tabelul 2.

Hazen (2006) propune o metodă de aliniere a transcrierilor aproximative în trei pași. Primul pas constă în aplicarea unui sistem de RAV pentru găsirea unor cuvinte ancoră, cuvinte care se regăsesc atât în transcrierea aproximativă, cât și în textul transcris. Sistemul de RAV folosește un model de limbă de tip tri-gram construit preponderent din transcrierea aproximativă (se utilizează o pondere de 0.99 pentru transcrierea aproximativă, respectiv 0.01 pentru un model de limbă generic). Pasul al doilea constă în alinierea pseudo-forțată a transcrierii aproximative cu semnalul vocal între două cuvinte ancoră selectate în pasul anterior: alinierea se realizează cu un model de limbă de tip FSG construit pe baza transcrierii aproximative, dar care permi-

Language model	G0-ML		G1-ML	
	SER [%]	WER [%]	SER [%]	WER [%]
Book LM	95.34	46.53	90.12	38.67
1SKIP	22.51	2.98	21.13	3.33
3SKIP	79.51	11.65	62.12	10.45
3SKIP with LM	50.56	5.50	47.20	5.12

Tabelul 2: Rezultate metodei ALISA (Stan et al., 2016).

te inserții, substituții, ștergeri (acest tip de tranziții sunt penalizate). Pasul al treilea constă în re-utilizarea sistemului de RAV pentru a completa zonele marcate ca inserții, substituții sau ștergeri din pasul precedent. Acest ultim pas permite corectarea transcrierilor aproximative, și rezultatele confirmă o îmbunătățire de la 10% la 8% WER. Din punctul de vedere al semnalului aliniat, metoda propusă de Hazen (2006) aliniază transcrierea aproximativă atât la nivel de text (primul pas), cât și la nivel de audio (al doilea pas). Metoda este similară cu cea din Moreno et al. (1998), dar cu două diferențe: în primul rând, această abordare nu folosește recursivitatea pentru a reduce procesul de aliniere pentru a funcționa pe segmente din ce în ce mai mici, ci are un număr fix de pași (trei), fiecare executând un tip specific de "rafinare" asupra alinierii transcrierii propuse; în al doilea rând, sistemul caută explicit discrepanțe între transcrierea manuală și acustică observată în fișierul audio și încearcă să corecteze aceste discrepanțe.

Metoda prezentată în (Tao et al., 2010) se bazează pe detecția erorilor de inserție și ștergere la nivel de propoziție, respectiv paragraf. Problema alinierii este transformată într-o serie de sub-probleme ce sunt rezolvate recursiv de un algoritm de programare dinamică. Datele de intrare (audio și text) sunt segmentate în unități mici (propoziții). Transcrierea este segmentată folosind metoda *maximum entropy* (Berger et al., 1996). Pentru a determina limitele propozițiilor în semnalul audio se realizează o pre-procesare în trei etape: *(i)* părțile ce nu conțin vorbire sunt separate și îndepărtate; *(ii)* se detectează pauzele în vorbire folosind un algoritm de detecție de voce (en., *voice activity detection*; VAD); *(iii)* delimitările detectate de algoritmul VAD sunt filtrate pentru a obține delimitări de prozodie. Toți acești pași sunt redați grafic în figura 5.

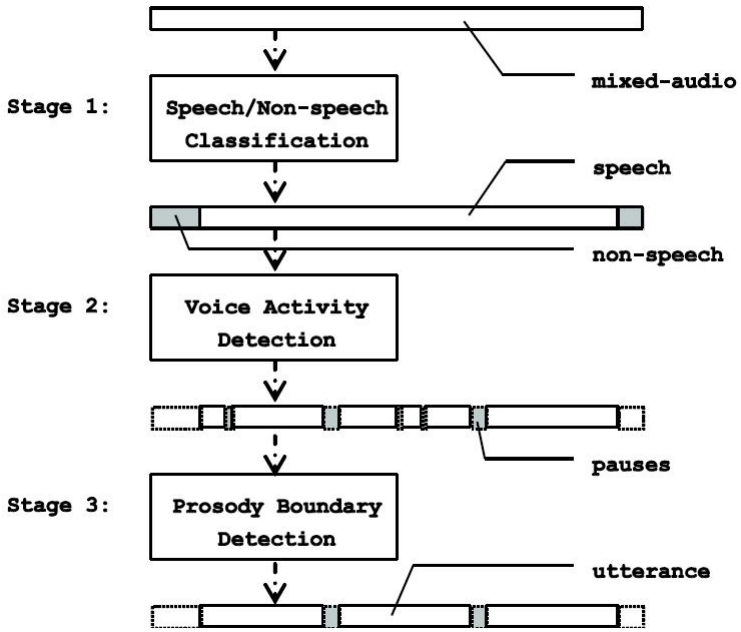

Figura 5: Etapele de pre-procesare pentru detecția propozițiilor (Tao et al., 2010).

Alinierea audio-text se realizează folosind semnalul audio segmentat și se bazează pe scorurile acustice. Valoarea scorurilor acustice indică probabilitatea ca un segment de vorbire să reprezinte un simbol conform modelelor statistice. Deoarece această valoare depinde de lungimea segmentului respectiv, aceasta trebuie normalizată; acest lucru presupune împărțirea scorului la numărul de ferestre din segmentul respectiv.

Rezultatele indică îmbunătățiri în aliniere față de metodele tradiționale de *forced-alignment*. Majoritatea erorilor sunt cauzate de alinieri greșite consecutive, pe care algoritmul nu le mai poate corecta ulterior.

3 Metode generale de aliniere

În această secțiune considerăm situația în care avem date două secvențe și vrem să le aliniem, adică să găsim corespondențele între elementele celor două secvențe. Pe baza alinierii putem compara cele două secvențe calculând distanța (sau alternativ, similaritatea); vom vedea în continuare că distanța alinierii este strâns legată de distanța de editare (en., *edit distance*)—numărul de modificări între cele două secvențe. Sarcina de aliniere și comparare a două secvențe este comună atât în problemele de bio-informatică (alinierea secvențelor de ADN), cât și în problemele de procesare a limbajului natural (corectarea greșelilor de scriere, calcularea erorii la nivel de cuvânt, traducerea automată):

- *Alinierea secvențelor de ADN* (Mount, 2013) are ca scop detectarea genelor comune sau a mutațiilor. Elementele secvențelor de ADN sunt nucleotide și sunt în număr de patru: A, C, T, G. Majoritatea algoritmilor de aliniere au fost dezvoltați în contextul sarcinilor din bio-informatică (Needleman and Wunsch, 1970; Smith and Waterman, 1981; Gotoh, 1982). Un exemplu de aplicație este arătat de Roberts (1997) care a folosit tehnicile de aliniere pentru a descoperi enzimele de restricție.

- *Corectarea greșelilor de scriere* (Jurafsky and Martin, 2008) este o facilitate întâlnită în procesoarele de text. În acest caz, secvențele sunt cuvinte, iar elementele sunt caractere. Cu ajutorul metodelor de aliniere se poate stabili similaritatea dintre cuvintele de intrare (introduse de utilizator) și cuvintele din dicționar (care se presupun corecte); astfel, se pot propune în mod automat variante corectate ale cuvintelor de intrare.

- *Calculul erorii la nivel de cuvânt* (en., *word error rate*) este metrica cea mai des întâlnită în evaluarea sistemelor de recunoaștere automată a vorbirii (RAV). Cele două secvențe sunt textul de referință și transcrierea produsă de către sistemul de RAV; elementele secvențelor sunt cuvinte. Cele două texte sunt aliniate urmărind minimizarea numărului de erori (eventual ponderate): cuvinte în plus, lipsă sau transcrise greșit. Distanța pe baza alinierii corespunde erorii la nivel de cuvânt.

- *Traducerea automată* (Koehn, 2010) este procesul prin care un text dintr-o limbă este transformat în textul corespunzător dintr-o altă limbă dată. În aplicațiile de traducere automată secvențele sunt propoziții, iar elementele sunt cuvinte. Alinierea constă în găsirea de corespondențe între cuvintele celor două limbi. Metodele de aliniere pentru traducere cele mai întâlnite sunt modele statistice, spre exemplu, modele de tip IBM (Brown et al., 1993) sau modele de tip HMM (Vogel et al., 1996). Spre deosebire de metodele folosite în aplicațiile menționate anterior, metodele pentru traducere permit alinieri mai puțin constrânse – elementele pot fi rotite între cele două secvențe pentru a modela ordinea variabilă a cuvintelor dintr-o limbă:

$$\begin{array}{ccccc} \cdots & x_i & & x_{i+1} & \cdots \\ & & \times & & \\ \cdots & y_j & & y_{j+1} & \cdots \end{array}$$

În continuare, pentru a evidenția diferențele între diferitele tipuri de algoritmi de aliniere, prezentăm problema într-un mod formal; prezentarea e inspirată din (Clote and Backofen, 2000). Considerăm că avem un alfabet finit de simboluri Σ și construim *cuvinte* concatenând simboluri din acest alfabet, $\mathbf{a} = a_1 a_2 \cdots a_N \in \Sigma^*$. Astfel secvențele pe care dorim să le aliniem sunt cuvinte formate din acest alfabet $\mathbf{a}, \mathbf{b} \in \Sigma^*$. *Alinierea a două cuvinte*, $\mathbf{a}$ și $\mathbf{b}$, este procesul de inserare a unor simboluri vide $- \notin \Sigma$ în cuvintele inițiale astfel încât cuvintele aliniate, $\mathbf{a}^\diamond$ și $\mathbf{b}^\diamond$: *(i)* să aibă aceeași lungime, adică $|\mathbf{a}^\diamond| = |\mathbf{b}^\diamond|$; *(ii)* să nu aibă elemente vide simultan, adică $a_i \neq -$ sau $b_i \neq -$ pentru orice i.

Considerăm un exemplu inspirat din alinierea secvențelor de ADN. În acest caz, alfabetul are patru simboluri: $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{T}, \mathtt{G}\}$. Dacă considerăm

cele două secvențe de intrare:

$$\mathbf{a} = \texttt{ACGGAT}$$
$$\mathbf{b} = \texttt{CCGCTT}$$

atunci câteva alinieri posibile ar putea fi:

$\mathbf{a}^\diamond$	AC-GG-AT	ACGG---AT	ACGGAT
$\mathbf{b}^\diamond$	-CCGCT-T	--CCGCT-T	CCGCTT

Definim *costul de aliniere* ca fiind numărul de simboluri diferite (marcate cu `x` mai jos) între cele două cuvinte aliniate, $\mathbf{a}^\diamond$ și $\mathbf{b}^\diamond$. Pentru cele trei exemple anterioare de alinieri obținem următoarele costuri de aliniere:

$\mathbf{a}^\diamond$	AC-GG-AT	ACGG---AT	ACGGAT
$\mathbf{b}^\diamond$	-CCGCT-T	--CCGCT-T	CCGCTT
diferențe	x.x.xxx.	xxxxxxxx.	x..xx.
cost	5	8	3

Distanța de aliniere între două cuvinte este minimul costului de aliniere peste toate alinierile posibile între cele doua cuvinte, **a** și **b**. Pentru cuvintele **a** și **b**, menționate anterior, distanța de aliniere este de 3 și este obținută pentru alinierea din cel de-al treilea exemplu.

Un concept strâns legat de ideea de aliniere este procesul de editare; acesta se referă la o succesiune de transformări (inserare, ștergere, modificare a simbolurilor) aplicate primului cuvânt pentru a-l obține pe cel de-al doilea. Ca în cazul alinierii, putem defini *costul unei editări* ca fiind numărul de transformări din procesul de editare, iar *distanța de editare* (Wagner and Fischer, 1974) între două cuvinte ca fiind costul asociat celei mai scurte secvențe de editare. Aceasta este succesiunea de transformări care corespunde primei alinieri din tabelul precedent (am notat prin $(-, \sigma)$ inserarea, $(\sigma, -)$ ștergerea, (σ, τ) modificarea unui simbol):

$$\texttt{ACGGAT} \xrightarrow{\texttt{A,-}} \texttt{CGGAT} \xrightarrow{\texttt{-,C}} \texttt{CCGGAT} \xrightarrow{\texttt{G,C}} \texttt{CCGCAT} \xrightarrow{\texttt{-,T}} \texttt{CCGCTAT} \xrightarrow{\texttt{A,-}} \texttt{CCGCTT}$$

Se poate arăta că între cele două concepte, cel de aliniere și cel de editare, există o legătură precisă: fiecărei editări îi corespunde o aliniere și cele două tipuri de distanțe, distanța de aliniere și cea de editare, sunt egale. Atât pentru distanța de aliniere, cât și cea de editare, putem avea diferite ponderi pentru tipurile de erori (inserare, ștergere, modificare). Când toate erorile au

o pondere egală, obținem distanța Levenshtein (Levenshtein, 1966); această distanță este des întâlnită în algoritmii de corectare a greșelilor de scriere (en., *spell checking*).

Scopul metodelor descrise este găsirea alinierii ce produce o distanță minimă între două cuvinte. Găsirea eficientă a alinierii optime se face cu ajutorul tehnicilor de programare dinamică. Programarea dinamică (Bellman, 1957) găsește soluții optime pe baza soluțiilor parțiale optime. Needleman and Wunsch (1970) au arătat că problema calculării costului de aliniere are proprietatea de a se putea descompune în sub-probleme și au propus o ecuație recursivă pentru calculul costului și care depinde de costul obținut pentru secvențele cu un caracter mai scurte. Complexitatea algoritmului este de $\mathcal{O}(mn)$ ca timp și spațiu, unde m și n reprezintă lungimea celor două secvențe.

Dynamic time warping (DTW) este un alt algoritm care rezolvă o problemă de aliniere, dar optimizează o funcție în care ponderile sunt modificate astfel încât să permită repetări ale simbolului precedent (acestea au un cost nul; alternativ, putem permite ștergeri ale simbolului precedent la cost nul) și nu permite simboluri vide. Informal, DTW-ul "deformează" o secvență în cealaltă; această caracteristică îl face potrivit pentru aplicații de clasificare a seriilor temporale, spre exemplu, recunoașterea vorbirii, unde o valoare se poate repeta pentru că viteza de vorbire diferă în funcție de utilizator. De remarcat că DTW nu reprezintă o distanță validă pentru că nu respectă inegalitatea triunghiului; aceasta este în contrast cu distanța de aliniere care este o distanța validă dacă metrica din care e compusă reprezintă o distanța validă. O altă diferență între cele două metode este că DTW se aplică de obicei pe valori continue, în timp ce metodele de aliniere inspirate din bio-informatică folosesc valori discrete (simbolurile alfabetului).

Metodele descrise anterior sunt metode de aliniere globală. O alternativă o reprezintă metodele de aliniere locală în care căutăm alinieri optime pentru fiecare sub-secvență din cele două secvențe date. Algoritmii de aliniere globală și locală fac presupuneri diferite: primii presupun că cele două secvențe de text sunt similare și au lungimi similare; cei din urmă presupun că o secvență este conținută în cealaltă. Alinierea locală este o sarcină comună în problemele de căutare a unei sub-secvențe într-o secvență și se pretează pentru situația când avem două secvențe foarte diferite. Ideea este de a modifica funcția de cost pentru eliminarea costului pentru ștergeri la începutul sau finalul uneia dintre secvențe. Soluția pentru acest algoritm se bazează tot pe programare dinamică și a fost propusă de (Smith and Waterman, 1981); algoritmul propus de Smith-Waterman are o complexitate de $\mathcal{O}(m^2n)$ (un-

de n este lungimea secvenței mai scurte), dar a fost îmbunătățită la $\mathcal{O}(mn)$ de către Gotoh (1982). Un exemplu de aliniere locală este următorul (unde culoarea neagră indică sub-secvențele aliniate):

```
AWGVIACAI--LAGRS
   VIVTAIAV-AGYY
```

Aplicații ale metodelor de aliniere. O aplicație a metodelor de aliniere pentru sarcina discutată în acest document, de aliniere a transcrierilor aproximative, este metoda lui Székely et al. (2012). Aceștia compară două șiruri lungi de caractere folosind alinieri obținute cu programare dinamica, similar cu metoda utilizată pentru evaluările de eroare în ASR (Automatic Speech Recognition). Toate rostirile ce au avut o distanta Levenshtein mai mare de un prag (2 caractere în cazul acesta) au fost excluse din procesările următoare. Rezultate arată că aproximativ 74% din datele de vorbire au fost păstrate și aproximativ 52% din citirile greșite au fost filtrate. Alte metode generale de aliniere aplicate în articolele menționate în secțiunea 2 sunt alinierea globală (Cardinal et al., 2005), distanța de editare (Liao et al., 2013), cea mai lungă secvență comună (Braunschweiler et al., 2010). Un alt exemplu în care algoritmul Smith-Waterman e aplicat pe text este implementarea propusă de Mullen (2016).

Recent, algoritmii de aliniere au fost reformulați ca funcții diferențiabile și poate fi astfel integrați în procesul de învățare (Cuturi, 2011; Cuturi and Blondel, 2017). Această idee este interesantă în special în contextul actual în care s-au observat rezultate remarcabile când procesul de învățare integrează cât mai mulți din pași – două astfel de tehnici sunt *end-to-end learning* și *deep learning*.

Alte metode de aliniere inspirate din bio-informatică. Metoda bazată pe cuvinte (en., *word methods*) verifică dacă secvențe de k gene consecutive se găsesc în ambele secvențe. Aceasta este o euristică care poate pierde din regiunile comune dacă acestea diferă, dar are avantajul că e mult mai rapidă. O altă categorie o constituie metodele de tip *dot-matrix*, în contextul nostru, (Buzo et al., 2013) au utilizat o metodă similară pentru sarcina transcrierilor aproximative.

4 Tehnici de normalizare a textului

Rolul tehnicilor de normalizare a textului este de a transforma reprezentarea scrisă a unui text într-o reprezentare a modului în care acesta este citit. Exemple de unități de text care se scriu diferit față de cum se citesc includ numerele, abrevierile, unitățile de măsură, datele calendaristice, URL-urile. Astfel, propoziția "o girafă cântărește 1200 kg și măsoară 5 m", se citește "o girafă cântărește o mie două sute de kilograme și măsoare cinci metri".

Această sarcină, deși poate părea minoră la prima vedere, joacă un rol important în problema de aliniere a transcrierilor aproximative. Pentru a facilita alinierea textului transcris de RAV cu transcrierea aproximativă trebuie ca cele două surse să folosească un vocabular comun, iar etapa de normalizare asigură acest lucru. De asemenea, normalizarea textului este utilă și pentru antrenarea modelelor de limbă, care sunt indispensabile oricărui sistem de RAV.

Dintre articolele prezentate în secțiunea 2, Braunschweiler et al. (2010) folosesc o normalizare destul rudimentară a transcrierilor aproximative; aceasta are trei pași: *(i)* eliminarea textelor din antetul și subsolul paginilor; *(ii)* tratarea caracterelor speciale și a stilurilor de formatare (*e.g.*, secvențe cu asterisc sau linii); *(iii)* împărțirea textului în propoziții.

Un articol recent legat de normalizarea textului este (Sproat and Jaitly, 2016), în care autorii vin cu o provocare pentru comunitate, și anume, un nou corpus mare de text scris, aliniat cu forma normalizată vorbită. Autorii învață funcția de normalizare folosind rețele neurale recurente (en., *recurrent neural network*; RNN) de tip *sequence-to-sequence*. Arhitecturile utilizate sunt inspirate din alte domenii care mapează secvențe la secvențe: conversia grafemelor la foneme (Rao et al., 2015), conversiei vorbirii în text (Chan et al., 2016). Unele arhitecturi produc într-adevăr rezultate bune pentru problema de normalizare, dacă considerăm acuratețea totală, dar erorile produse sunt problematice, ar putea transmite un mesaj complet greșit dacă un astfel de sistem ar fi folosit într-o aplicație. Exemple de astfel de erori ar fi, "82.55 mm" care este normalizat la "eighty two one five five meters" sau "2 mA" care este normalizat la "two units".

Concluziile articolului legate de utilitatea RNN-urilor pentru problema de normalizare sunt în mare parte negative, dar autorii nu exclud RNN-urile ca o soluție posibilă pentru problema normalizării. În schimb, ei lasă această sarcină ca o provocare pentru întreaga comunitate. Articolul constantă că o metodă mai tradițională, bazată pe un filtru de tip *finite state transducer*

(FST) poate diminua numărul de erori și poate obține o acuratețe mai bună decât un RNN. Un filtru FST face corespondența dintre expresii de forma `<numar> <unitate>` la un număr cardinal sau decimal și verbalizările posibile ale abrevierilor unităților de măsură. Astfel, "24.2 kg" poate fi normalizat la "twenty four point two kilograms". Utilizarea FST-ului în algoritmul de RNN duce la îmbunătățirea rezultatelor.

5 Concluzii

Acest document a trecut în revistă articole care tratează sarcina alinierii transcrierilor aproximative cu semnalul vocal. Există două clase principale de metode: cele care aliniază transcrierea aproximativă direct la semnalul vocal (aliniere text-audio) și cele care aliniază transcrierea aproximativă la reprezentarea textuală a semnalului vocal, obținută cu un sistem de RAV (aliniere text-text). Aceste clase nu sunt exclusive și, astfel, există metode care combină ambele idei, spre exemplu, (Hazen, 2006). Multe metode au mai mulți pași de aliniere succesivă (Moreno et al., 1998; Hazen, 2006) și etape de segmentare a semnalului vocal prin împărțirea în unități mai mici (Tao et al., 2010). Alinierea este adesea bazată pe algoritmi de aliniere generici care sunt optimizați cu programarea dinamică. Din păcate, este dificil de concluziat care dintre metode funcționează cel mai bine pentru că seturile de date și metodologia de lucru variază – aceste motive fac imposibilă o comparație echitabilă. Din punctul de vedere al proiectului, cele mai promițătoare lucrări sunt cele care folosesc transcrierile foarte imprecise, în principal pentru că genul acesta de transcrieri sunt cele mai uzuale în practică: (Liao et al., 2013) care lucrează pe date uploadate de utilizatori și (Buzo et al., 2013) care folosesc date descărcate de pe internet.

Bibliografie

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, first edition.

Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Braunschweiler, N., Gales, M. J., and Buchholz, S. (2010). Lightly supervised recognition for automatic alignment of large coherent speech recordings. In *Eleventh Annual Conference of the International Speech Communication Association*.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Buzo, A., Cucu, H., and Burileanu, C. (2013). Text spotting in large speech databases for under-resourced languages. In *Speech Technology and Human-Computer Dialogue (SpeD), 2013 7th Conference on*, pages 1–6. IEEE.

Cardinal, P., Boulianne, G., and Comeau, M. (2005). Segmentation of recordings based on partial transcriptions. In *Interspeech*, pages 3345–3348.

Chan, W., Jaitly, N., Le, Q. V., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 4960–4964.

Clote, P. and Backofen, R. (2000). *Computational Molecular Biology: An Introduction*. John Wiley & Sons.

Cuturi, M. (2011). Fast global alignment kernels. In *Proceedings of the 28th International Conference on Machine Learning*, pages 929–936.

Cuturi, M. and Blondel, M. (2017). Soft-DTW: A differentiable loss function for time-series. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of

Proceedings of Machine Learning Research, pages 894–903, International Convention Centre, Sydney, Australia. PMLR.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, 162(3):705–708.

Hazen, T. J. (2006). Automatic alignment and error correction of human generated transcripts for long speech recordings. In *Interspeech*, pages 1606–1609.

Jurafsky, D. and Martin, J. H. (2008). *Speech and language processing*. Prentice Hall.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.

Lecouteux, B., Linarès, G., and Oger, S. (2012). Integrating imperfect transcripts into speech recognition systems for building high-quality corpora. *Computer Speech & Language*, 26(2):67–89.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Liao, H., McDermott, E., and Senior, A. (2013). Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription. In *Workshop on Automatic Speech Recognition and Understanding*, pages 368–373. IEEE.

Moreno, P. J. and Alberti, C. (2009). A factor automaton approach for the forced alignment of long speech recordings. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4869–4872. IEEE.

Moreno, P. J., Joerg, C., Thong, J.-M. V., and Glickman, O. (1998). A recursive algorithm for the forced alignment of very long audio segments. In *Fifth International Conference on Spoken Language Processing*.

Mount, D. (2013). *Bioinformatics: Sequence and Genome Analysis*. Cold Spring Harbor Laboratory Press.

Mullen, L. (2016). Text alignment. https://cran.r-project.org/web/packages/textreuse/vignettes/textreuse-alignment.html.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Rao, K., Peng, F., Sak, H., and Beaufays, F. (2015). Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4225–4229. IEEE.

Roberts, R. J. (1997). Hunting for new restriction enzymes in GenBank. In *Proceedings of the First Annual International Conference on Computational Molecular Biology*, RECOMB '97, page 251, New York, NY, USA. ACM.

Smith, T. and Waterman, M. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197.

Sproat, R. and Jaitly, N. (2016). RNN approaches to text normalization: A challenge. *CoRR*, abs/1611.00068.

Stan, A., Mamiya, Y., Yamagishi, J., Bell, P., Watts, O., Clark, R. A., and King, S. (2016). ALISA: An automatic lightly supervised speech segmentation and alignment tool. *Computer Speech & Language*, 35:116–133.

Székely, E., Csapó, T. G., Tóth, B., Mihajlik, P., and Carson-Berndsen, J. (2012). Synthesizing expressive speech from amateur audiobook recordings. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 297–302. IEEE.

Tao, Y., Xueqing, L., and Bian, W. (2010). A dynamic alignment algorithm for imperfect speech and transcript. *Computer Science and Information Systems*, 7(1):75–84.

Triguero, I., García, S., and Herrera, F. (2015). Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems*, 42(2):245–284.

Vogel, S., Ney, H., and Tillmann, C. (1996). Hmm-based word alignment in statistical translation. In *International Conference on Computational Linguistics*, pages 836–841. Association for Computational Linguistics.

Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173.