„Feeding the BEAST” – A BEA Speech Transcriber továbbfejlesztése és integrálása neurális nyelvmodellel

Kádár Máté Soma[1, 3], Dobsinszki Gergely[1], Mády Katalin[2], Mihajlik Péter[1, 2]

[1] Budapesti Műszaki és Gazdaságtudományi Egyetem,
Villamosmérnöki és Informatikai Kar,
Távközlési és Médiainformatikai Tanszék,
H-1117, Budapest, Magyar tudósok körútja 2.
[2] Nyelvtudományi Kutatóközpont,
H-1394, Budapest, VI. Ker. Benczúr utca 33.
[3] SpeechTex Kft.,
1181, Budapest, Madách Imre utca 47.

Kivonat: Cikkünkben a korábban BEAST néven publikált, a BEA-Base adatbázison tanított mély-neuronháló alapú beszédleiratozó modellt meghaladó struktúrát mutatunk be. A használt architektúra magába foglal egy unigram szótöredékeken tanított wav2vec2 alapú akusztikus és egy Transformer alapú nyelvi modellt. Az akusztikus modell az uráli nyelvcsaládba tartozó nyelveken (magyar, finn, észt) önfelügyelten előtanított wav2vec-large struktúrára épül, mely a BEA-Base-en történő finomhangolása és egyes hiperparaméterek optimalizálása után önmagában is felülmúlta a BEAST eredményeit: a korábbi 16.62%-os szóhibarátát 12.08%-ra csökkentette. Az akusztikus modellhez integráltuk a Magyar Nemzeti Szövegtár beszélt nyelvi alkorpuszán tanított mély neurális nyelvi modellt, mely a nyalábkeresés segítségével 10.98%-ra javította a leiratozó szóhibaarányát. Tudomásunk szerint eddig ez a legjobb beszédfelismerési eredmény ezen az adathalmazon.

Kulcsszavak: mélytanulás, automatikus beszédfelismerés, nyelvmodell, önfelügyelt tanulás, Transformer, nyalábkeresés

1 Bevezetés

Az elmúlt években a „neurális forradalom” az automatikus beszédfelismerés (ASR) területét is elérte: sorra jelennek meg újabb, konkrétan a beszédfelismerésre optimalizált neurális architektúrák. A természetes nyelvfeldolgozás (NLP) esetén már évek óta elérhető a word2vec architektúra (Mikolov és mtsai, 2013), mely segítségével az egyes tokenek (szavak és szótöredékek, akár karakterek is) beágyazhatók, azaz szemantikájuk alapján osztályozhatók, vektorizálhatók. Mióta ez a lehetőség létezik szövegekre, azóta próbálnak hasonló elven működő architektúrát létrehozni, mely képes hullámformák vektorizációjára. Erre kínál megoldást a wav2vec2 struktúra, mely önfelügyelt, kontrasztív módon tanítható, valamint számos state-of-the-art eredményt tudhat magáénak (Baevski és mtsai, 2020).

Hála a BEÁ-nak (BEszélt nyelvi Adatbázis) (Gósy, 2008, Gósy és mtsai, 2012), most már magyar nyelven is létezik egy ingyenesen elérhető, kutatási célokra használható beszélt nyelvi adatbázis. Ennek felhasználásával lehetőség adódik kurrens ASR modellek készítésére, valamint az egyes modelleket össze lehet hasonlítani leiratozási hibájuk alapján a BEA teszthalmazán (Mihajlik és mtsai, 2022).

A klasszikus - maximum likelihood - megközelítéssel szemben a tisztán mély tanuláson alapuló neurális rendszerek jóval alkalmasabbak mind a mintaillesztés végrehajtására, mind a nyelvmodellezés („Language Model(ing)") megvalósítására. A továbbiakban egy wav2vec2-t felhasználó enkóder-dekóder akusztikus és egy Transformer (Vaswani és mtsai, 2017) alapú nyelvi modellből álló, szótöredékekre épülő ASR rendszer kerül ismertetésre. A rendszer a BEA-Base 114 beszélő spontán beszédét tartalmazó hanganyagain tanult és a cikk írásának pillanatáig a legjobb eredménnyel rendelkezik a BEA-Base spontán teszthalmazán. A fejlesztéshez a SpeechBrain (Ravanelli és mtsai, 2021) névre hallgató nyílt forráskódú keretrendszert használtuk. A tanított neurális modelleket akadémiai kutatás céljából ingyenesen elérhetővé teszszük, mint a BEAST (BEA Speech Transcriber) továbbfejlesztése, BEAST2 néven.

2 A tanító és kiértékelő halmazok áttekintése

A BEA-Base adatbázis 114 beszélő hanganyagait és azok szöveges átiratait tartalmazza. Az adatbázis két felé osztható: a spontán, valamint a kötött beszédes részhalmazokra. Mivel az emberi kommunikáció természetéből adódóan spontán, így a tanított akusztikus és nyelvi modell az adatbázis spontán beszédből álló részhalmazán lett finomhangolva, illetve a hálózatok hiperparaméterei is a spontán teszthalmazra lettek optimalizálva.

A nyelvmodellek tanításához elengedhetetlen a sok adat, különösen, ha Transformer hálózatról beszélünk. Ahogy az 1. táblázatban látható, a BEA-Base 3.38M karakternyi leiratot tartalmaz, ami a nyelvmodellezési feladatokhoz meglehetősen kevés. Emiatt szükséges volt egy nagyobb tanítókorpusz használata, melyen a Transformer nyelvi modell előtanítása történt. A választás a Magyar Nemzeti Szövegtár (későbbiekben SPOK-ként hivatkozva, mint "SPOKen language" alkorpusz) beszéltnyelvi részadatbázisára esett (Oravecz és mtsai, 2014). A SPOK ezen része a MR1 Kossuth Rádió 2004–2012 évek során rögzített hír- és riportműsorainak leiratát tartalmazza.

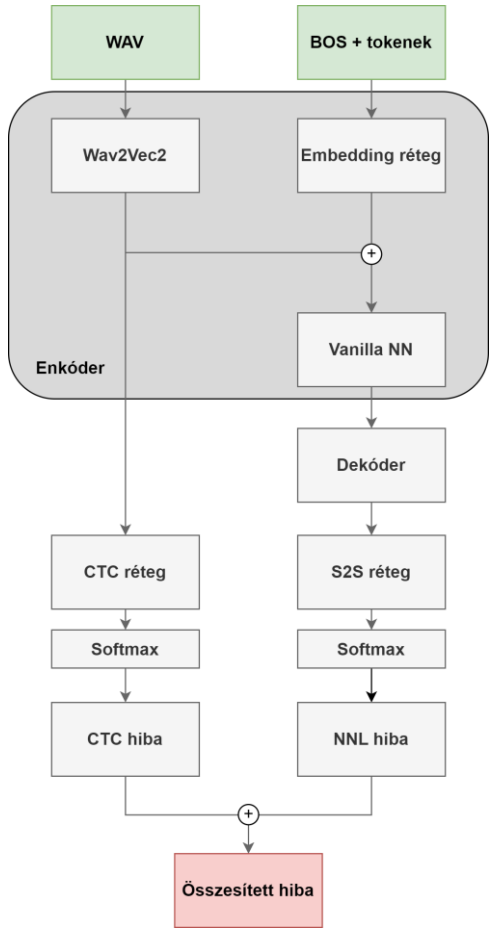
1. táblázat: A tanításhoz felhasznált adatbázisok és méreteik

Adatbázis	Karakterek (db)	Szavak (db)	Tokenek (db)	Audió (h)
BEA-Base	3.38M	0.56M	1.18M	71.2
SPOK	516.84M	56.13M	172.97M	-

Mind az akusztikus, mind a nyelvi modell szótöredék, pontosabban unigram szótöredék (Kudo és Richardson, 2018) alapon tanult. Ez esetünkben a BEA-Base spontán tanítókorpuszának önfelügyelt tokenizációját jelentette, 600-as szótárművel (mivel a magyar nyelv 44 darab karaktert tartalmaz, így <unk> („unknown") tokenek nem keletkeznek a tokenizációkor).

3 Az akusztikus modell

A klasszikus rejtett Markov-modell alapú megközelítés helyett a korábban említett wav2vec2 alapú architektúrát alkalmaztuk, mint az akusztikus „end-to-end” modell enkóder része. A választott hálózat a VoxPopuli projekt (Wang és mtsai, 2021) keretein belül előtanított, ~317M paraméterrel rendelkező modell volt. A wav2vec egy adott hullámforma tanult reprezentációját állítja elő, melyet utána egy dekóder használ fel.



1. ábra: Az akusztikus modell architektúrája az előtanított wav2vec2-vel felszerelt enkóderrel, dekóderrel és a tanításkor használt költségfüggvényekkel

Az önfelügyelt előtanítás során felhasznált tanítóadatok különlegessége, hogy az csak uráli nyelvcsaládba tartozó nyelveket tartalmazott, ideértve a magyart is. Az előtanítás során a hálózat 17.7K órányi, leirattal nem rendelkező magyar nyelvű beszéden tanult a többi uráli nyelven (finn 14.2K, észt 10.6K óra) felül. Az előtanított

modellt finomhangoltuk a BEA-Base spontán hanganyagain. A finomhangolás klaszszikus tanítási metodikával – Connectionist Temporal Classification (CTC) és attention költség minimalizálással (Graves és mtsai, 2006, Watanabe és mtsai, 2017) – történt.

3.1 Az adatok augmentációja

Tanításkor a bementi adatokon sebességperturbációt hajtottunk végre. Ez esetünkben azt jelentette, hogy a 16kHz-en mintavételezett beszédet 0.95, 1.0 vagy 1.05-szoros sebességű jelként adjuk a hálózat bementére. További augmentációként véletlenszerű időintervallum(ok)ban kimaszkoltuk a megfigyelhető jelet (Park és mtsai, 2019). A maszkolás minden köteg esetén – tehát 1.0 valószínűséggel – bekövetkezett 0–5 alkalommal mintánként.

Az augmentáció hozzájárul ahhoz, hogy a tanított hálózat generalizáltabb módon tanulhasson. Ezt kísérleteink során is igazoltnak láttuk: csökkentve a maszkolás valószínűségét 0.5-re, a tanított hálózat rosszabb WER eredményt ért el (13.19%), mint a növelt értékkel (12.08%). Az ismertetett augmentációs eljárások a túltanulás ellen is hatnak, így egyfajta regularizációként is tekinthetünk rájuk.

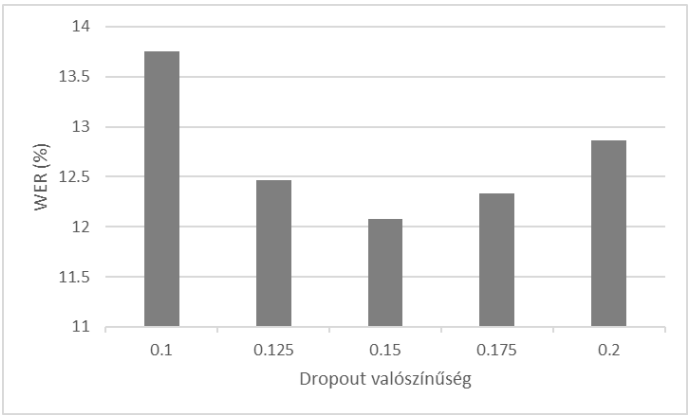
3.2 Hiperparaméterek és kísérletek

A tervezett architektúrában az enkóder magába foglalja a wav2vec2 modult és egy vanilla előrecsatolt alhálózatot, ami két rétegből, rétegenként 1024 neuronból áll. Dekóderként egy attention-ön alapuló rekurrens GRU (Cho és mtsai, 2014) hálózatot specifikáltunk (enkódolási méret: 1024, csatornák száma: 10, bemeneti méret: 128, attention dimenzió: 1024).

Az architektúra enkóder-dekóder része az Adadelta (Zeiler, 2012), míg a wav2vec2 része az Adam (Kingma és Ba, 2017) optimalizálót használta, rendre 1.0 és 1e-4 kezdeti tanulási rátával. Az Adam β_1 értéke 0.9, β_2 értéke 0.999 volt. Az Adadelta esetén a ρ értéke 0.95. A tanítási ráta ütemezése a validációs halmazon mért költségváltozáson alapult (NewBob ütemezés): amennyiben egy epoch során a költség egy bizonyos határértéknél kisebb mértékben csökkent – vagy esetleg nőtt –, úgy csökkentettük a tanulási rátát. A konkrét kötegméretet 3-ra állítottuk, ám az effektív kötegméret 12 volt. A kettőt a gradiens-akkumulációs tényező köti össze: e tényező állításával a hibavisszaterjesztés egy konstans számú lépés után hajtódik csak végre (ebben az esetben 4), nem pedig minden köteg után, így lehetőség adódik nagyobb kötegmérettel történő tanítás szimulálására.

További regularizációként a dropout technikát használtuk (Srivastava és mtsai, 2014). Észrevételeink alapján az ismertetett architektúra erre a hiperparaméterre érzékeny volt, így ezzel mélyebben kísérleteztünk. A hálózatot összesen 100 epoch-on keresztül tanítottuk. Az első 20 epoch során 0.4 súllyal vettük figyelembe a CTC költséget, 0.6-tal a „Negative Log-Likelihood” (NLL) költséget. A fennmaradó 80 epoch során tisztán a NLL alapján történt a hiba számítása. A NLL hiba kalkulálása során 0.1-es értékkel címkesimítást használtunk a túltanulás ellen (Müller és mtsai, 2020).

A tanított rendszer eredményét a szóhibaráta („Word Error Rate”, WER) határozta meg. Értelemszerűen, az alacsonyabb WER érték pontosabb átiratok létrehozását jelenti. A legjobb eredményt (12.08%) 0.15-ös globális dropout értékkel értük el (2. ábra), így a BEAST2 rendszer is ezt az akusztikus modellt foglalja magába.



2. ábra: A modell elért WER eredményei a BEA-Base spontán teszthalmazán a dropout regularizáció függvényében

4 A nyelvi modell

A számláláson alapuló N-gram nyelvmodellek N darab token figyelembevételével tudnak becslést adni a következő token valószínűségére. Elterjedt 4, illetve 5-gram modellek használata, ám belátható, hogy ennyi előtörténet még szavak esetén sem elegendő. A rekurrens neurális hálózatok ezt hivatottak kiküszöbölni: segítségükkel lehetőség adódik távolibb kapcsolatok modellezésére, ami a magyar nyelv esetén is rendkívül hasznos. Jó pár évig megkerülhetetlenek voltak a rekurrens hálók, ám szekvenciális felépítésük miatt lassú és nehézkes volt a tanításuk (Pascanu és mtsai, 2013).

A Transformer egy új megközelítést alkalmaz: a szekvenciális tanítást párhuzamosra cserélve nagyságrendekkel gyorsítja a nyelvi modellek konvergenciáinak sebességét. Ezen felül a grafikus kártyákra (GPU) is jobban illeszkedik, kihasználva azok masszív párhuzamosítási képességeit.

A tervezett hálózatot az akusztikus modellel megegyező unigram szótöredékek segítségével tanítottuk. A hálózatot a GPT (Radford és mtsai, 2018) architektúra alapján terveztük, a SPOK halmazon tanítottuk és validáltuk, majd a legjobbnak ítélt – legalacsonyabb perplexitással (PPL) rendelkező – modellt tovább hangoltuk a BEA-Base spontán átiratain. Kontrollként tanítottunk egy klasszikus 5-gram módosított Kneser-Ney simított (Chen és Goodman, 1999) nyelvmodellt is a KenLM (Heafield, 2011) eszköz felhasználásával.

4.1 Hiperparaméterek és szöveg alapú kísérletek

Annak érdekében, hogy a hálózat optimalizációja során pontosabb becslést kaphassunk a költségfüggvény gradiensére, fontos a megfelelő kötegméret megválasztása. Technikai megfontolásból, a 256 karakternél hosszabb mondatokat kivettük a SPOK adathalmazból, így a kötegméretet nem kellett irracionálisan alacsonyan tartani, ami a tanítási procedúra degradálását idézné elő.

Az előtanítás során használt adatbázis (SPOK) 97%-a adta a tanító-, 2%-a a validációs és 1%-a a teszthalmazt. Az előtanítás során 20 epoch-on át tanult az összes ismertetett hálózat. Az effektív kötegméret minden háló esetén 512 volt, a konkrét kötegméret viszont a tanított modell méretétől függően változott az elérhető memóriakapacitás miatt. Közös vonás továbbá, hogy a tanulási ráta dinamikusan változott a Noam ütemezésnek megfelelően: kezdetben lineárisan nőtt a megadott tanulási rátának függvényében, majd fordítottan négyzetesen csökkent. Az optimalizáló algoritmusnak az Adam-et választottuk, melynek β_1 paraméterét 0.9-re és β_2 értékét 0.98-re vettük. A túltanulás ellen dropout regularizációt alkalmaztunk 0.1 valószínűséggel. A további hiperparaméter-értékek a 2. táblázatban láthatók.

2. táblázat: Az egyes Transformer nyelvmodellek eltérő hiperparaméterei és az előtanítás utáni perplexitás a SPOK-ból képzett validációs halmazon

Modell / param. (db)	Tanulási ráta	Enkóderek (db)	Beágyazás (db)	Belső (db)	Fejek (db)	PPL
baseline / ~67.7M	0.1	12	768	2048	12	12.61
I. / ~100.7M	0.5	14	768	3072	12	6.29
II. / ~149.3M	0.75	14	1024	3072	16	**6.15**
III. / ~171.6M	0.5	24	768	3072	16	6.22

Az előtanítást követően a legalacsonyabb perplexitást elérő nyelvmodellt (II.) a BEA-Base spontán átiratain finomhangolva tanítottuk tovább. A finomhangolásnál elterjedt a mélyebben lévő – tehát a bementhez közelebbi – rétegek befagyasztása. A megoldásunkban az első 4 réteg került befagyasztásra, így a tanítható paraméterek száma ~149.3M-ról ~107.3M-ra csökkent. A továbbtanítás 32 epochon keresztül zajlott, kezdetben 1.75e-6 tanulási rátával. Mivel ez esetben egy előtanított modellt használtunk fel, így a bemelegítési lépésekre már nem volt szükség, helyette a korábban ismertetett NewBob ütemezést használtuk. A további hiperparaméter-értékek a korábban ismertetettekkel megegyeztek. A modell finomhangolás előtti PPL értéke a BEA-Base teszthalmazán 32.2, míg utána 18.5 lett.

A kontroll 5-gram nyelvmodell esetében a finomhangolás nem értelmezhető, így azt külön a SPOK-on és külön a BEA-Base megegyező részén tanítottuk. Előbbi esetben 13.79, utóbbi esetben 42.13 PPL értéket ért el, ami mindkét esetben rosszabb, mint a neurális hálózatok által elért eredmény.

5 Kísérleti eredmények

Az akusztikus modell kimeneteit különböző stratégiák mentén lehet dekódolni: „greedy” módon, mindig a legvalószínűbb tokeneket választva, valamint a nyelvmodell segítségével átsúlyozva. Utóbbit a „beam-search” (nyalábkeresés) segítségével érdemes implementálni.

A finomhangolt modelleket a „shallow fusion” (Toshniwal és mtsai, 2018) megközelítéssel kapcsoltuk össze: az akusztikus háló hipotézisei a nyelvi modell segítségével kerültek újrasúlyozásra.

A nyalábkeresési algoritmus hiperparaméter-optimalizációját követően a következő hiperparaméter-értékekkel történt végleges kiértékelés: a nyaláb szélességét 84-re állítottuk, külön számításba vettük a CTC költséget 0.014 súllyal. A nyelvi modell súlyát 0.285-re vettük. A számontartott hipotéziseket a hosszukkal normalizáltuk (Wu és mtsai, 2016). Ez különösen fontos, mert enélkül az eljárás a rövidebb szekvenciákat preferálná, hiszen azok log-valószínűségeinek összege nagyobb, mint a hosszabb szekvenciáknak. Mind az akusztikus, mind a nyelvi modell predikcióit 1.05 „temperature”-rel mintavételeztük. Az „end-of-sentence” határ (Hannun és mtsai, 2019) értékét 2.5-re, valamint a „coverage penalty” (Chorowski és Jaitly, 2016) értékét 3.0-ra állítottuk.

A tanítás mindkét modell esetén két darab NVIDIA RTX 3090-es, 24 GB VRAM-mal rendelkező GPU-n történt. Az egyes tanítások több napot vettek igénybe az összes tanított hálózat esetén. A könnyebb összehasonlíthatóság érdekében a 3. táblázat tartalmaz információt a korábbi és a mostani modellek teljesítményéről.

3. táblázat: Az alap BEAST, egy 440k órányi adaton előtanult és finomhangolt wav2vec alapú rendszer, az uráli nyelveken előtanított, majd finomhangolt wav2vec2 struktúra, valamint a BEAST2 WER eredményei a BEA-Base spontán teszthalmazán mérve

Modell	LM	WER (%)
BEAST[1]	-	16.62
440k-wav2vec-finetuned[2]	-	15.61
43k-wav2vec2-uralic-finetuned	-	12.08
BEAST2	neurális (II.)	**10.98**

Az eredmények tekintetében kijelenthető, hogy a wav2vec2 előtanítása uráli nyelveken, majd finomhangolása a BEA megfelelő részadatbázisán szignifikáns javulást ért el a BEAST-hez képest. Ehhez egy explicit nyelvmodellt kapcsolva (BEAST2) képesek voltunk a nyalábkeresés során összességében ~34%-kal csökkenteni a spontán teszthalmazon mért szóhibaarányt a BEAST rendszerhez képest. A továbbfejlesztett modelleket ingyenesen elérhetővé tesszük[3] kutatási célokra.

A kiértékelése használt spontán teszthalmaz 4.91 órányi hanganyagát a BEAST2 rendszer ~40 perc alatt dekódolta egy NVIDIA RTX 3090 GPU-t használva.

[1] Mihajlik és mtsai, MSZNY 2022
[2] Mihajlik és mtsai, LREC 2022
[3] https://phon.nytud.hu/bea/bea-base

6. Összegzés

A Transformer alapú beszédleiratozás ma már az első számú beszédfelismerési megközelítéssé vált világszerte. Ugyanakkor magyar nyelvre csak minimális mennyiségű felügyelt adaton tanult modellek érhetők el a BEAST rendszeren kívül. Megmutattuk, hogy az előtanított modell megfelelő megválasztásával, hiperparaméter-optimalizációjával, valamint korszerű, mély neurális nyelvmodell integrálásával jelentős javulás érhető el még a kiforrott BEAST-hez képest is – a cikk írásának pillanatában a legjobb eredményt értük el (WER=10.98%). Ez a várakozásaink szerint még tovább javítható további hiperparaméter-optimalizációval, illetve a BEA-Base adathalmaz várható bővülésével.

Köszönetnyilvánítás

Köszönettel tartozunk a munkánkat támogató, NKFIH-135038 projektazonosítójú, „Prozódiai szerkezet és mondattípusok vizsgálata nagy beszédadatbázisokon mély tanulási támogatással” című, valamint az NKFIH-143075-ös és NKFIH-828-2/2021 (MILab) projekteknek. Ezen felül köszönjük mindenkinek, aki munkájával hozzájárult a BEA adatbázis létrejöttéhez.

Bibliográfia

Baevski, A., Zhou, H., Mohamed, A., Auli, M.: wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, arXiv: 2006.11477v3 (2020)

Chen, S F., Goodman, J.: "An empirical study of smoothing techniques for language modeling,” Computer Speech & Language, vol. 13, no. 4, pp. 359–393 (1999)

Cho, K., Merrienboer, B., Bahdanau, D.: On the Properties of Neural Machine Translation: Encoder–Decoder Approaches, arXiv: 1409.1259v2 (2014)

Chorowski, J., Jaitly, N.: Towards Better Decoding and Language Model Integration in Sequence-to-Sequence models, arXiv: 1612.02695v1 (2016)

Gósy, M.: Magyar spontánbeszéd-adatbázis – BEA, Beszédkutatás, pp. 194–207 (2008)

Gósy, M., Gyarmathy, D., Horváth, V., Gráczi, T.E., Beke, A., Neuberger, T., Nikléczy, P.: BEA: Beszélt nyelvi adatbázis, In: Gósy Mária (szerk.): Beszéd, adatbázis, kutatások, Budapest, Akadémiai Kiadó, pp. 9–24 (2012)

Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks, ICML 2006 – Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376 (2006)

Hannun, A., Lee, A., Xu, Q., Collobert, R.: Sequence-to-sequence Speech Recognition with Time-Depth Separable Convolutions, arXiv: 1904.02619v1 (2019)

Heafield, K.: KenLM: Faster and Smaller Language Model Queries, Proceedings of the Sixth Workshop on Statistical Machine Translation, pp.: 187–197 (2011)

Kingma, D.P., Ba, J.L.: Adam: A Method for Stochastic Optimization, arXiv: 1412.6980v9 (2017)

Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, arXiv: 1808.06226v1 (2018)

Mihajlik, P., Balog, A., Gráczi, T. E., Kohári, A., Fegyó, T., Mády, K.: „Releasing the Beast" – A BEA gépi beszédleiratozási feladat, megközelítések, eredmények, XVIII. Magyar Számítógépes Nyelvészeti Konferencia (2022)

Mihajlik, P., Balog, A., Gráczi, T.E., Kohári, A., Tarján, B., Mády, K.: BEA-Base: A Benchmark for ASR of Spontaneous Hungarian, In: Proceedings of the 13th Conference of LREC 2022, pp. 1970–1977 (2022)

Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space, arXiv: 1301.3781v3 (2013)

Müller, R., Kornblith, S., Hinton, G.: When Does Label Smoothing Help?, arXiv: 1906.02629v3 (2020)

Oravecz Cs., Váradi T., Sass B.: The Hungarian Gigaword Corpus, In: Proceedings of LREC 2014 (2014)

Park, D.S., Chan, W., Zhang, Y., Chiu, C., Zoph, B., Cubuk, E.D., Le, Q.V.: SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition, arXiv: 1904.08779v3 (2019)

Pascanu, R., Mikolov, T., Bengio, Y.: On the Difficulty of Training Recurrent Neural Networks, arXiv: 1211.5063v2 (2013)

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language Models are Unsupervised Multitask Learners, OpenAI blog 1(8), 9 (2019)

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J., Yeh, S., Fu, S., Liao, C., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R., Bengio, Y.: SpeechBrain: A General-Purpose Speech Toolkit, arXiv: 2106.04624v1 (2021)

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research 15, pp. 1929–1958 (2014)

Toshniwal, S., Kannan, A., Chiu, C., Wu, Y., Sainath, T.N., Livescu, K.: A Comparison of Techniques for Language Model Integration in Encoder-Decoder Speech Recognition, arXiv: 1807.10857v2 (2018)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I.: Attention Is All You Need, arXiv: 1706.03762v5 (2017)

Wang, C., Riviere, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., Williamson, M., Pino, J., Dupoux, E.: VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation, arXiv: 2101.00390v2 (2021)

Watanabe, S., Hori, T., Kim, S., Hershey, J.R., and Hayashi T.: Hybrid CTC/Attention Architecture for End-to-End Speech Recognition, in IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 8, pp. 1240–1253 (2017)

Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M.: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, arXiv: 1609.08144v2 (2016)

Zeiler, M.D.: Adadelta: An Adaptive Learning Rate Method, arXiv 1212.01v1 (2012)