HILBERT, magyar nyelvű BERT-large modell tanítása felhő környezetben

Feldmann Ádám [1], Hajdu Róbert [1], Indig Balázs [2], Sass Bálint [2], Makrai Márton [2], Mittelholcz Iván [2], Halász Dávid [2], Yang Zijian Győző [2], Váradi Tamás [2]

[1] Pécsi Tudományegyetem, Általános Orvostudományi Kar, Magatartástudományi Intézet, Alkalmazott Adattudomány és Mesterséges Intelligencia Csoport,
7624 Pécs, Szigeti u 12.
{feldmann.adam,hajdu.robert}@pte.hu
[2] Nyelvtudományi Intézet,
1394 Budapest, Pf. 360
{indig.balazs,sass.balint,makrai.marton,mittelholz.ivan,
halasz.david,yang.zijian.gyozo,varadi.tamas@nytud.hu}

Kivonat A dolgozatban bemutatjuk a magyar nyelvű BERT-large modell készítését, amely 3.667 milliárd szavas szövegkorpusz felhasználásával jött létre olyan megoldásokat alkalmazva, amelyek eddig egyedül angol nyelvi modellek létrehozásnál jelentek meg. A célunk olyan felhő alapú komplex számítási környezet létrehozása volt, amelyben mind szoftveres, mind pedig hardveres eszközök állnak rendelkezésre azért, hogy az új, mélytanulás alapú nyelvi modellek magyar nyelvi korpuszokkal tanítva is elérhetővé váljanak, hasonlóan a nagyobb nyelveken már elérhető state-of-the-art modellekhez. A környezet az ONNX keresztplatform megoldásait felhasználva sokkal erőforrás-optimalizáltabban hajtja végre a modellek tanítását. HILBERT, a magyar nyelvű BERT-large nyelvi keretrendszer ONNX, PyTorch, Tensorflow formátumokban rendelkezésre áll.

Kulcsszavak: BERT-large, ONNX, HILBERT, NER, Transformers

1 Bevezetés

Ebben a cikkben bemutatjuk a BERT-large nyelvi keretrendszer magyar adaptációját, az ahhoz szükséges számítási hátteret és magát a folyamatot. A BERT-modellt (Bidirectional Encoder Representations from Transformers), amely általános célú nyelvmegértő modell, a Google AI Language kutatócsoportja 2018 októberében publikálta (Devlin és mtsai, 2018). Céljuk egy általános, komplex és kontextus érzékeny beágyazott nyelvi eszköz létrehozása volt. A modell egyedinek számított a 340 millió paraméterével, mivel ezt megelőzően a mélytanuló modellek, területtől függetlenül sokkal kisebb paraméterszámmal jelentek meg. A BERT eszköz a nyelvi megértést célzó modellek rendkívüli mennyiségű tanítóadat igényét igyekszik mederbe terelni transzfer tanulás segítségével.

A BERT-modell alapkoncepciója szerint a felhasználónak elég egy előre megtanított modellt előkészítenie, majd ezt jóval kisebb adathalmazon transzfer tanulás

segítségével adott célfeladatokhoz kell finomhangolnia. A BERT-large modell előtanítása rendkívül számításigényes feladat, különleges technológiai hátteret igényel ennek megvalósítása, amely hatvannégy darab V100-as NVIDIA GPU felhasználásával közel 100 óra futásidőt vesz igénybe.

A BERT modelleknek két fajtája érhető el méretük szerint; az első a BERT-base, amely 110 millió paraméterrel rendelkezik, illetve a BERT-large, amely 340 millió paramétert tartalmaz. Devlin eredeti célja a BERT-base megalkotásával az volt, hogy a BERT modellt összevethesse a korábban megjelent, szintén 110 millió paraméteres GPT (Generative Pretrained Transformer) névre hallgató eszközzel. Mindkét BERT modell azonos architektúrával rendelkezik, de paramétereikben különböznek. A BERT-base 12 darab kódoló réteggel, míg a BERT-large modell 24 darab kódoló réteggel bír. További különbség, hogy a kódoló rétegen belül nagyobb a figyelmi fej; a BERT-base 12, míg a BERT-large 16 figyelmi fejjel rendelkezik. A feedforward rétegen belül, mely a kódoló réteg egyik része, 768 rejtett feldolgozó elem található a kisebb, míg 1024 a nagyobb modellnél.

A BERT-base változatot magyar nyelvre Nemeskey Dávid készítette el (Nemeskey, 2020), demonstrálva a modell kiemelkedő képességeit különböző nyelvi feladatokon. A BERT modell részletes, kellő mélységgel történő tárgyalása szintén Nemeskey Dávid előbb hivatkozott publikációjában olvasható. Jelen tanulmány a szükséges számítási környezet jellemzésére és bemutatására helyezi a hangsúlyt, valamint a BERT-large modell előtanítását és finomhangolását mutatja be.

2 A HILBERT modell létrehozása

2.1 Számítási környezet kialakítása

Az extrém nagy méretű mélytanuló modellek tanításhoz speciális hardver és szoftverkörnyezet szükséges. Mivel a GPU alapú számítási eszközök közül is csak a kifejezetten gépi tanulás támogatására létrehozott célprocesszorok alkalmasak, valamint ezekből több darabra is szükség van a tanításhoz, a felhő alapú számítási megoldások felé fordultunk. A Microsoft Azure felhőszolgáltatáson belül találtunk megfelelő méretű, bérelhető számítási kapacitást és szoftveres környezetet. Az AzureML környezetet kifejezetten gépi tanulási folyamatok megvalósítására és szolgáltatására fejlesztették. Modulokra bonthatóan kezelhetőek benne az egyes részfeladatok, melyhez tárolókat és egyéb erőforrásokat rendeltünk. Az AzureML SDK 1.6-os változatát használtuk Python 3.6 nyelven. A mélytanulási feladathoz pedig a PyTorch framework-öt választottuk az ONNX Runtime keresztplatform felhasználásával. A PyTorch szabványosan elérhető AzureML környezetben, az ONNX platform pedig integrálja a legújabb számításoptimalizáló és gyorsító megoldásokat, köztük a DeepSpeed technológiát (Rajbhandari és társai, 2019), amely akár ötszörösére gyorsítja a modellek tanítását a GPU memória használatának optimalizálásán keresztül. A szükséges számítási klasztert is itt hozzuk létre, ahol az AzureVM eszközök közül választhatjuk ki a feladathoz leginkább megfelelő tulajdonságokkal bíró csomópontokat. Kezdeti lépésként létrehoztunk egy eszközcsoportot az Azure-ben, melyben számítási csomók és tárolók egyaránt helyet kaptak. Fontos, hogy nagyobb adatmozgás esetén a virtualizált környezet ellenére az egyes

eszközök fizikailag is közel legyenek egymáshoz, mert a tárhelyműveletek sokmilliószor lassabbak, mint a számítási műveletek. Mivel a modell tanításához GPU alapú erőforrás szükséges, de a kód szerkesztése, módosítása ezt nem kívánta meg, így létrehozunk egy alapértelmezett számítási eszközt egy virtuális gép segítségével. Az allokált eszköz elegendő a környezet felparaméterezéséhez és a tárolókkal történő műveletek végrehajtásához.

2.2 Az adatok jellemzése

Az előtanító korpusz

A nyelvi modellek készítésének döntő fontosságú kérdése a korpusz minősége, amelyen a modell előtanítása készül. Az előtanításhoz szükséges korpuszt a nyelvmodell célja szabja meg. A mai gyakorlatban az honosodott meg, hogy rendszerint egy *általános célú* nyelvi modellt készítenek, melyet aztán adott feladat számára *finomhangolnak*. Az általános célú nyelvmodellt olyan korpuszon célszerű betanítani, amely a nyelvhasználat széles körét reprezentálja. A nyelvhasználat egészét átfogóan és arányaiban is modellálni nem jól definiált feladat, mert szigorú értelemben vett reprezentatív mintát nem lehetséges összeállítani. Ugyanis a teljes populációról (azaz a nyelvhasználat egészéről) nincsenek megbízható adataink. A legtöbb, amit tehetünk az, hogy egy úgynevezett kiegyensúlyozott korpusz (balanced corpus) összeállítására törekszünk, illetve figyelembe vesszük a korpusz felhasználásának a célját.

A BERT modellhez szükséges legalább 3,5 milliárd szónyi folyó szövegből álló korpuszt az alábbi forrásokból állítottuk össze.

MNSZ. Fontos forrás a Nyelvtudományi Intézetben készült Magyar Nemzeti Szövegtár. Egyrészt hat stílusrétegből (sajtó, szépirodalom, tudományos, hivatalos, személyes, beszéltnyelvi) tartalmaz szövegeket, másrészt ezen belül öt regionális nyelvváltozatra oszlik. A regionális nyelvváltozatok az egyes határon túli magyar területeket képviselik. Kiemelendő az önmagában is jelentős, 76 millió szavas beszéltnyelvi (rádiós) alkorpusz, ez az MR1 Kossuth rádió bizonyos anyagait öleli fel az 2004-2012 évekből, felolvasott szöveget (hírek) és spontán beszélgetést (riportok) vegyesen. Mérete 975 millió szó.

JSI. A szlovén Jožef Stefan Institute az eventregistry.org címen futó webszolgáltatás céljaira 2013 óta számos nyelven gyűjti a híreket internetes forrásokból (RSS-ből). Ennek a magyar anyagát használtuk fel. Ebben egészen friss hírek is szerepelnek, megjelennek az aktuális témák (koronavírus stb.). Mérete 1,06 milliárd szó.

NOL. A MNSZ sajtókorpuszát kiegészítettük a Mediaworkstől kapott *Népszabadság online* anyaggal. Ennek terjedelme 48 millió szó.

OS. A következő forrás a szabadon hozzáférhető filmfelirat-adatbázis, az opensubtitles.org magyar része. Amint említettük, erre jellemző a beszéltnyelvi stílus, rövid mondatok, párbeszédes forma. Mérete 471 millió szó.

KM. Az utolsó forrás egy jelentős, nyilvános közösségi média posztokból és kommentekből származó szöveganyag, melyet a Neticle Kft-től kaptunk meg korábban. Mérete 1,11 milliárd szó.

A szótár
Több milliárd szavas korpusz esetén a rendszer által használt szótár kritikus jelentőséget kap. A kihívást az jelenti, hogy a szótárnak lehetőleg le kell fednie a korpuszban előforduló szóalakok egészét, ugyanakkor kis méretűnek kell lennie a hatékonyság jegyében. A szavak belső reprezentációjára egy olyan szótárt használ, amelyekben a szavak statisztikai alapon *szóelemekre* vannak bontva, extrém esetben az egyes karakterekig. A BERT modell a Google által kifejlesztett WordPiece eljárást alkalmazza. A szótárak mérete általában 30 és 50 ezer elem között váltakozik. A magyar nyelv morfológiai sajátosságaira tekintettel a HILBERT modellhez 64000 elemes WordPiece szótárat fejlesztettünk ki. A szótár hatékonyságát Nemeskey Dávid kódjával mértük. Minél kevesebb szóelemre bontja a szótár a felszíni szavakat, annál jobbnak mondható. A HILBERT tanításánál használt WordPiece esetében ez a mutató 1, 15, azaz átlag egy szövegszót 1,15 szóelemre bont a tokenizáló.

2.3 Az adatok előfeldolgozása

A modell tanításához elsőként a szövegeket bináris formába kell hozni ahhoz, hogy a BERT modell tanításához felhasználhatóak legyenek. Az eredeti BERT modellek a Wikipédia angol nyelvű szövegkorpuszán és könyvkorpuszokon készültek. A magyar szövegek előkészítése során meghagytuk az eredeti, Wikipédiára utaló könyvtárszerkezetet. A nyers szövegfájlok összmérete 25 GB. A szöveg darabolására az előfeldolgozási lépések memóriaigénye miatt volt szükség. Az előfeldolgozás egy külön folyamat, melynek bemenete a 100 darab szövegfájl és a kimenete olyan bináris állomány, amelyben a tenzor bemenetek vannak elrendezve modelltanításhoz és validációhoz. Az adatfeldolgozáshoz külön programot készítettünk. A szöveg rendezése során a beolvasott szöveget úgy tisztítjuk, hogy csak az alfanumerikus és központozó karakterek maradjanak benne, illetve minden sorba egy mondat kerüljön. Ezután speciális tokeneket kell hozzáadni a tokenizált szöveghez <cls> és <sep> elválasztó karaktereket. A <cls> a szövegek különböző osztályozásakor játszik szerepet, míg a <sep> szeparátorként választ el mondatokat egymástól. A program iteratívan végighaladva az aktuális szövegrészen illeszti hozzá a szótárban található szóelemeket, ahol nem ismert szóelem tokennel találkozik, ott azt <unk> taggal helyettesíti. Ez a folyamat többféle szótárral, illetve tokenizáló eszközzel is történhet. A tokenizálás 25 GB szövegen 4 nap alatt futott le. A folyamat memória intenzív feldolgozás, ahol a számítások végrehajtásához egy STANDARD_D14_V2 virtuális gépet vettünk igénybe. Ennek eredményeként egy blob tárolóban létrejött 100 db bináris állomány 600GB körüli tárhely igénnyel, mely már készenállt a BERT-large modell tanításához.

2.4 GPU klaszter létrehozása

A modell számítási paramétereinek a megállapítása szorosan összefügg a felhasználható, rendelkezésre álló GPU kapacitás méretétől. Mivel GPU segítségével nagyon gyorsan lehet mátrixokat összeszorozni és feldolgozni, ezért kiválóan alkalmasak tenzor alapú számítások futtatásához, sokszoros teljesítménynövekedést nyújtva a CPU alapú feldolgozással szemben. A leginkább elterjedt eszközök az NVIDIA által gyártott

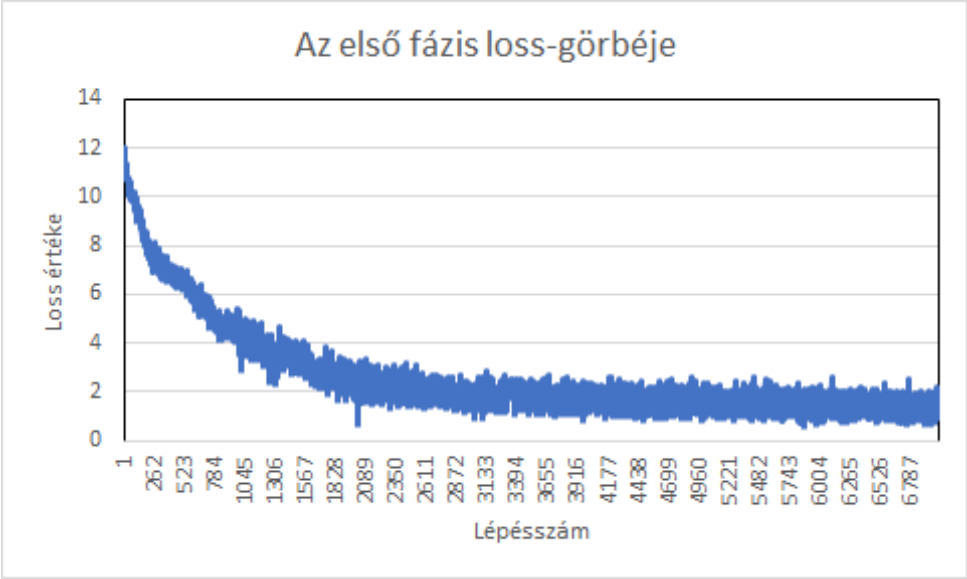
V100-as GPU-k, melyekhez különböző méretű VRAM tartozik. Az Azure környezetben elérhető, GPU alapú számítási csomópontok közül az NCv3-as széria NC24rs v3 kódjelű node-jára esett a választásunk. Ez az eszköz 4 db V100-as GPU-t tartalmaz, melyekhez egyenként 16GB VRAM tartozik a 448GB RAM mellett. Azért választottuk ezt a számítási csomót, mert RDMA-kompatibilisek és Infiniband alapú kapcsolat segítségével rövid látencia mellett biztosítják a számítási fürtön belül a node-ok közötti, alacsony szintű kommunikációt. Ez azért különösen fontos, mert MNI (Message Passing Interface) segítségével jobban párhuzamosíthatóak a több GPU-s feldolgozást igénylő feladatok, ha több csomópontot szeretnénk összekötni.

2.5 A tanítási paraméterek megadása

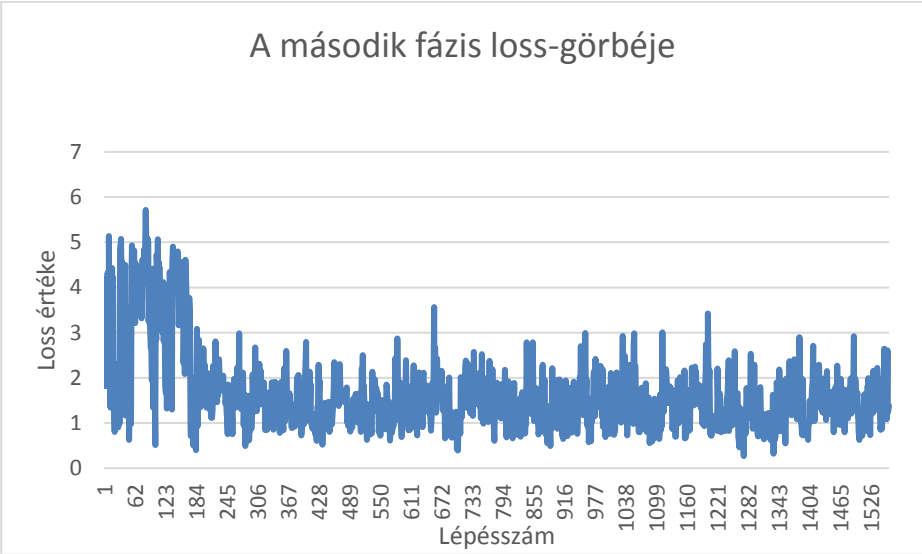
A BERT-large modell tanításához az AzureML Kísérlet modulján belül kell konfigurálni az MPI-t és meg kell adni, hogy egy számítási csomóban hány darab GPU található. Meg kell adni továbbá, hogy a GPU-ban található CuDA magok kezeléséhez szükséges csomagokat és az openmpi drivereit melyik docker image tartalmazza. A batch size paraméter függ a rendelkezésre álló GPU-k számától, illetve azok VRAM méretétől. A párhuzamos GPU használat esetén minden GPU külön számol grádiens losst különböző adatokon. Minél nagyobb a grádiens mérete, annál inkább csökken a zaj hatása a tanításra. Ennek ellenére a tanítás későbbi szakaszában a nagy grádiens méret kevésbé vezet optimális eredményhez.

A modell tanítását az NVIDIA scriptjével végeztük, amely két fázisra osztja a tanítást. Az első fázisban 128 token hosszúságú modellt készítünk, majd ezt követően 512 token hosszúsággal folytatjuk tovább a modell tanítását. Erre bontásra azért van szükség, mert a figyelmi fejek méretének növekedésével a számítási kapacitás négyzetesen növekszik. A második fázis gyakorlatilag egy finomhangolási lépés. A modell előtanításának 90%-a 128-as hosszúsággal, míg az utolsó 10% 512-es tokenhosszra történik (Devlin és mtsai, 2018). Az első fázis 7038 lépést, míg a második 1563 lépést tartalmazott.

A szkript paramétereit a kötegméret, a gárdiens akkumuláció és a GPU memória limitet kivételéve az alapértelmezett értékeken hagytunk. A modell 128-as szekvencia hosszon 32-es batch mérettel, 6e-3 tanulási rátával (0.2843 előmelegítési ráta), 512-es szekvencia hosszon pedig 8-as batch mérettel, 4e-3 tanulási rátával (0.128 előmelegítési ráta) paraméterekkel tanult. Ezeket mindkét fázis esetében külön meghatároztuk a használt számítási csomókhoz. A modell tanítási folyamatának állapotáról a tanulás veszteség-függvénye nyújt információt (Ábra1, Ábra2). A magyar BERT-large (HILBERT) tanítása során támpontként szolgált az NVIDIA által közzétett veszteség görbe az angol nyelvi modellhez, illetve ugyanezeket megkaptuk a Microsoft fejlesztői csapatától is.



1. ábra A BERT-large modell veszteségfüggvénye tanítás során 128 token hosszúságú szekvenciákkal.



2. ábra A BERT-large modell veszteségfüggvénye tanítás során 512 token hosszúságú szekvenciákkal.

2.6 A kész modell finomhangolása

Az elkészített modell finomhangolása egy gyakran használt transzfer tanulási módszer. Ezzel a felügyelt tanítási módszerrel specifikus feladatokra lehet tovább tanítani a modellt, mint a névelemek felismerése, vagy kontextusalapú kérdés-válasz generálásra, illetve különféle célú szövegosztályozó feladatok végrehajtása. Működését tekintve az előtanított modell utolsó rétege fölé egy klasszifikációs réteg kerül (Devlin és mtsai, 2018), ami a tovább tanítás során a bemenetet és annotációit tanulja meg.

A névelem-felismerés egy gyakran használt módszer a nyelvi modellek teljesítménymérésére. A szegedi Corpus of Business Newswire Texts (szegedNER) corpust alkalmaztuk a névelem-felismerés tanításához (Szarvas és mtsai, 2006). A korpuszt 80-10-10 arányban bontottuk fel tanító, validációs és teszt adathalmazokra. A transzfer tanítási megoldás a transformers könyvtár példái közül lett kiválasztva. Az F1-értékek számítása a seqeval könyvtárral történt. A finomhangolás feladathoz NVIDIA Tesla V100 16GB videókártyát használtunk felhőkörnyezetben. A finomhangolási paraméterek közül, a modell 3e-05-ös (lineárisan csökkenő) tanulási rátán és 3 epoch-on keresztül tanult, 8-as kötegmérettel.

A modell validációs F1-értéke a corpusban annotált 16 névelemosztályra összesen 95.39%-ot adott. A modell valódi képességeit leginkább jelző, a teszt adathalmaz F1-értékére, szintén 16 névelem osztályra 93.91%-ot kaptunk. Ezek az értékek azt mutatják, hogy a magyar nyelvű HILBERT teljesítménye a névelem keresés terén rendelkezik a BERT-large modellektől elvárt képességekkel (Virtanen és mtsai 2019; Martin és mtsai, 2019).

A többi modellel való összevetés lehetőségét árnyalja, hogy a nemzetközi szakirodalomban az egyes, különböző nyelveken elérhető, annotált névelem adatbázisok sokszor eltérő névelem kategóriákat(is) tartalmaznak, illetve az elérhető adatbázisok mérete nyelvenként nagyon eltérő lehet. További nehézség, hogy az egyes modellek tanítása gyakran eltérő epoch-számmal történik. Ezeken túl a finomhangolás random inicializálása is hatással van a finomhangolt modellek teljesítményére (Dodge és mtsai, 2020).

3. Összegzés

A HILBERT, magyar nyelvű BERT-large modellt sikerült létrehoznunk egy kereskedelmi számítási felhőben, ahol olyan horizontálisan és vertikálisan is skálázható infrastruktúrát alakítottunk ki, amelyben több, akár magasabb paraméterszámú modellek előállítása is lehetségessé vált. Elkészítettük a szegedNER corpus (Szarvas és mtsai, 2006) segítségével a modellünk finomhangolását névelemkereséshez, amelyben ~ 94%-os teszt eredményt sikerült elérnünk. Jelenleg is rendelkezésre áll több, a modellhez köthető alkalmazásunk, amelyben a HILBERT, mint extraktív szövegösszegző, illetve mint keresőmotor jelenik meg. A BERT-large igazi előnye azonban a kérdés-válasz típusú feladatokban mutatkozik meg a többi, kisebb paraméterszámú modellhez képest, de ilyen adathalmaz magyar nyelven egyelőre nem elérhető.

Köszönetnyilvánítás

A szerzők köszönetüket fejezik ki Varga Gábornak és a Microsoft Magyarország Kft. többi munkatársának a segítségükért, akik lehetővé tették, hogy a pandémiás időszak alatti korlátozások ellenére hozzáférjünk a szükséges számítási kapacitásokhoz. Külön szeretnénk megköszönni a lehetőséget, hogy a Microsoft Corporation ONNX Runtime fejlesztőcsapatával együtt dolgozva a legújabb fejlesztéseiket tesztelve tudtuk létrehozni a magyar nyelvű BERT-large modellt.

Hivatkozások

Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deepbidirectional transformers for language understanding. In: Proc. of NAACL (2019)

Dodge, J., Ilharco, G., Scwartz, R.,Farhadi,A.,Hajishirzi,H.,Smith, N.: Fine-tuning Pretrained Language Models:Weight Initalizations, Data Orders, and Early Stopping.(2020)

Martin, L., Muller, B., Ortiz, S., Dupont, P.J. Romary, L.,Villemonte de la Clergerie, E., Seddah, D., Sagot, B., CamemBERT: a Tasty French Language Model(2019)

Nemeskey, D.M.: Egy embert próbáló feladat. In: XVI. Magyar Számítógépes Nyelvészeti Konferencia. pp. 409–418. Szegedi Tudományegyetem, Szeged(2020a)

Rajbhandari, S., Rasley, J., Ruwase, O., He, Y.: Zero: Memory optimization towards training a trillion parameter models. (2019)

Szarvas, G., Farkas, R., Kocsor, A.: A multilingual named entity recognition system using boosting and C4.5 decision tree learning algorithms. In: Discovery Science, 9th International Conference, DS 2006, Barcelona, Spain, October 8-10, 2006, Proceedings. pp. 268–278 (2006)

Virtanen, A., Kanerva, J., Ilo,R., Louma, J., Luotolahti, J., Salakoski, T., Ginter, F., Pyysalo, S.: Multilingual is not enough: BERT for Finnish (2019)

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: A multi-task benchmark and analysis platform for natural language understanding (2018)