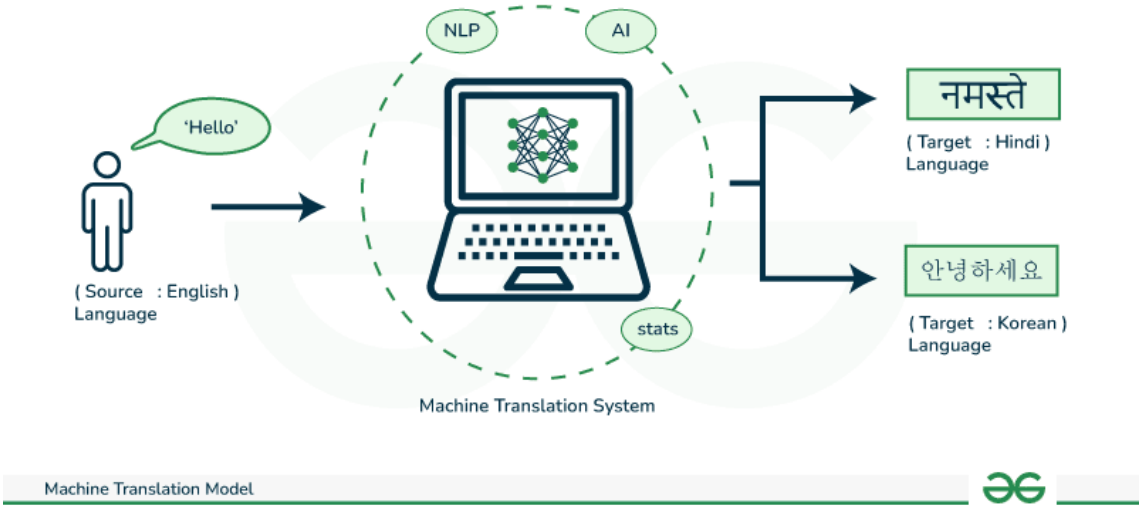
MACHINE TRANSLATION – XNN SEQ2SEQ



Contributors

Enric Canudas (1631674)

Bruno León (1631333)

Ramón Álvaro (163583)

1. Introducció i Objectius

Aquest projecte consisteix en un model RNN que utilitza una arquitectura d'aprenentatge Seq2Seq per tal d'aprendre a realitzar traduccions.

En aquest document hi consta el procés realitzat a partir del següent starting point: https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html amb diferents modificacions aplicades per tal de millorar el rendiment del model inicial. El codi ja està preparat per generar gràfiques de l'execució tant de la los d'entrenament com de validació i el bleu score de les traduccions realitzades per el model. Per veure les gràfiques generades s'ha de tindre un usuari a **Weights & Biases** i posar la key personal de l'usuari abans d'executar la resta de codi.

El principal objectiu d'aquest projecte és entendre com funciona el model i la seva estructura interna per tal d'aconseguir una millor versió del model que realitzi unes millors traduccions. Això consta en buscar reduir el overfitting i fer augmentar els bleu scores amb l'ajustament dels hiperparàmetres i la implementació de millores al model.

Code Structure

Al repositori del GitHub conté tot el necessari per poder executar el codi. A la carpeta idiomes tenim tots els datasets que hem utilitzat. Els arxius .pth són el codificador i el decodificador ja entrenats amb també la configuració dels hiperparàmetres a l'execució (model_config.pth).

Després tenim diferents notebooks (.ipynb) amb els codis a executar per posar el model en funcionament, hi ha diferents notebooks que hem anat actualitzant per arribar al notebook definitiu que conté el codi de l'execució final (model_word2vec.ipynb). Aquest arxiu conté tot el codi on primer consten totes les funcions que realitzen el preprocessament de les dades. Després hi tenim les classes que defineixen el encoder, el decoder i el decoder amb atenció. Posteriorment, hi consta el codi que crea els dataloaders i finalment el que realitza l'entrenament i validació del model on al final es veuen exemples de diferents traduccions i la visualització del heatmap de l'atenció.

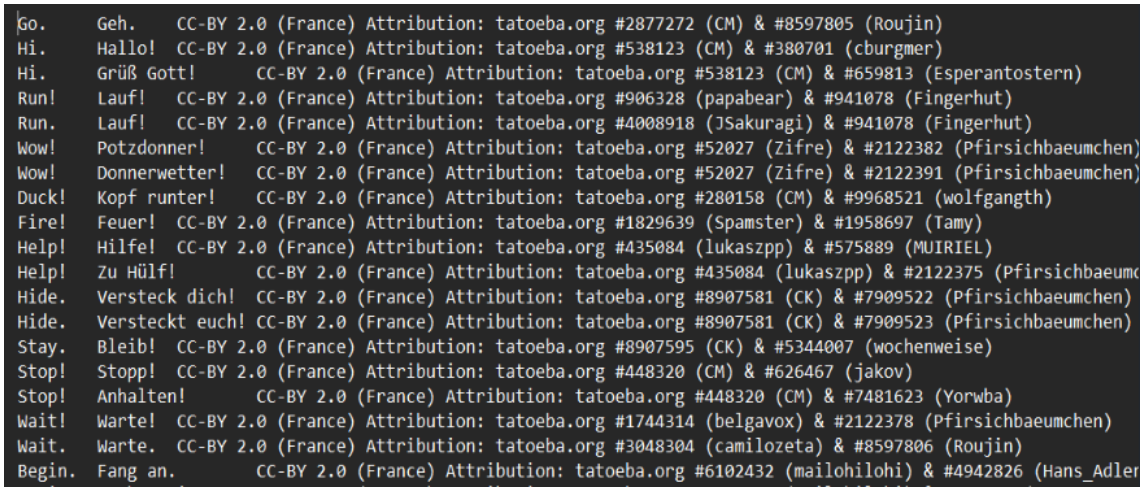
How to use the code?

1) Descarregar els diferentes datasets.
2) A l'anar al notebook posar la key d'usuari a la cel·la de wandb
3) Executar el codi cel·la per cel·la

Dataset

Per aquest projecte hem utilitzat el dataset Anki que conté una recopilació de traduccions d'oracions de diverses longituds per diferents idiomes. És un dataset ja especialitzat per l'entrenament i validació de models de traducció automàtica. Per al projecte hem agafat el dataset amb traduccions de l'anglès a l'alemany (270.000 traduccions) i de l'anglès a l'holandès (80.000 traduccions).

L'estructura dels datasets és la mateixa per qualsevol idioma, s'organitza per parelles d'oracions. Per cada fila de l'arxiu .txt conté una parella d'oracions amb el text en l'idioma d'origen (majoritàriament anglès) i la seva traducció en l'idioma que hem escollit. Les oracions estan ordenades per la longitud, comença amb oracions d'una paraula i va augmentant la longitud de les oracions.



```
Go.     Geh.    CC-BY 2.0 (France) Attribution: tatoeba.org #2877272 (CM) & #8597805 (Roujin)
Hi.     Hallo!  CC-BY 2.0 (France) Attribution: tatoeba.org #538123 (CM) & #380701 (cburgmer)
Hi.     Grüß Gott!      CC-BY 2.0 (France) Attribution: tatoeba.org #538123 (CM) & #659813 (Esperantostern)
Run!    Lauf!   CC-BY 2.0 (France) Attribution: tatoeba.org #906328 (papabear) & #941078 (Fingerhut)
Run.    Lauf!   CC-BY 2.0 (France) Attribution: tatoeba.org #4008918 (JSakuragi) & #941078 (Fingerhut)
Wow!    Potzdonner!     CC-BY 2.0 (France) Attribution: tatoeba.org #52027 (Zifre) & #2122382 (Pfirsichbaeumchen)
Wow!    Donnerwetter!   CC-BY 2.0 (France) Attribution: tatoeba.org #52027 (Zifre) & #2122391 (Pfirsichbaeumchen)
Duck!   Kopf runter!    CC-BY 2.0 (France) Attribution: tatoeba.org #280158 (CM) & #9968521 (wolfgangth)
Fire!   Feuer!  CC-BY 2.0 (France) Attribution: tatoeba.org #1829639 (Spamster) & #1958697 (Tamy)
Help!   Hilfe!  CC-BY 2.0 (France) Attribution: tatoeba.org #435084 (lukaszpp) & #575889 (MUIRIEL)
Help!   Zu Hülf!       CC-BY 2.0 (France) Attribution: tatoeba.org #435084 (lukaszpp) & #2122375 (Pfirsichbaeumc
Hide.   Versteck dich!  CC-BY 2.0 (France) Attribution: tatoeba.org #8907581 (CK) & #7909522 (Pfirsichbaeumchen)
Hide.   Versteckt euch! CC-BY 2.0 (France) Attribution: tatoeba.org #8907581 (CK) & #7909523 (Pfirsichbaeumchen)
Stay.   Bleib!  CC-BY 2.0 (France) Attribution: tatoeba.org #8907595 (CK) & #5344007 (wochenweise)
Stop!   Stopp!  CC-BY 2.0 (France) Attribution: tatoeba.org #448320 (CM) & #626467 (jakov)
Stop!   Anhalten!      CC-BY 2.0 (France) Attribution: tatoeba.org #448320 (CM) & #7481623 (Yorwba)
Wait!   Warte!  CC-BY 2.0 (France) Attribution: tatoeba.org #1744314 (belgavox) & #2122378 (Pfirsichbaeumchen)
Wait.   Warte.  CC-BY 2.0 (France) Attribution: tatoeba.org #3048304 (camilozeta) & #8597806 (Roujin)
Begin.  Fang an.        CC-BY 2.0 (France) Attribution: tatoeba.org #6102432 (mailohilohi) & #4942826 (Hans_Adler
```

Avaluació i mètriques de rendiment.

Per avaluar el rendiment del model utilitzem una **Cross-Entropy Loss** que compara la probabilitat per cada token de la seqüència de sortida amb la distribució real (etiquetada) i penalitza amb major contundència les prediccions incorrectes.

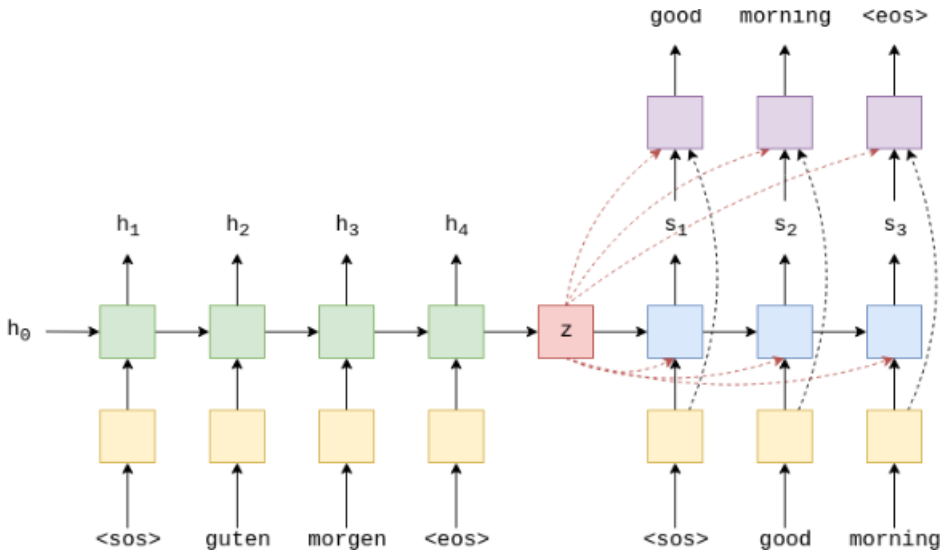
Per avaluar la qualitat de les prediccions del model hem utilitzat **Bleu Score** que compara les traduccions generades amb les traduccions de referència. Hi ha diferents Bleu Score depenent del nombre de n-grams que vulguem utilitzar des de 1 fins a 4. Si agafem n-grams de 1 el Bleu avaluarà les seqüències paraula per paraula, en canvi si agafem n-grams de 4 el Bleu avaluarà la seqüència en agrupacions de 4 paraules dins la seqüència. A mesura que augmentem els n-grams les puntuacions del Bleu aniran disminuint.

2. Arquitectura

Utilitzem un model Seq2Seq basat en RNN. Aquest model consta de dos components principals: l'encodificador (encoder) i el decodificador (decoder). L'encodificador processa la seqüència d'entrada (la frase en l'idioma d'origen) i la converteix en un vector de context, que encapsula la informació essencial de tota la seqüència. Aquest vector es

passa al decodificador, que genera la seqüència de sortida (la frase en l'idioma de destí) de manera seqüencial. A cada pas de temps, el decodificador utilitza l'estat ocult anterior i la paraula generada anteriorment per predir la següent paraula. Aquest procés continua fins que es produeix un símbol de final de seqüència.

El nostre decodificador utilitza el mecanisme d'atenció (Attention) que permet al decodificador accedir a tots els estats ocults de l'encodificador. A cada pas del decodificador, es calcula un pes d'atenció per a cada estat ocult de l'encodificador, que determina quanta importància s'ha de donar a cada part de la seqüència d'entrada en predir la següent paraula. La combinació ponderada dels estats ocults d'entrada es converteix en el vector de context dinàmic per al pas de temps actual del decodificador. Els dos tipus de RNN que es solen utilitzar per aquests projectes son LSTM i GRU.



L'atenció utilitzada per models Seq2Seq es Bahdanau Attention que permet que el descodificador d'un model seq2seq enfoqui la seva atenció en diferents parts de la seqüència d'entrada en cada pas de la generació de la seqüència de sortida. En lloc d'utilitzar només l'últim estat ocult del codificador, el model calcula una ponderació per a cada estat ocult del codificador en funció de la seva rellevància per a l'estat actual del descodificador. Aquestes ponderacions s'utilitzen per crear un vector de context ponderat, que proporciona al descodificador informació més específica i rellevant en cada pas.

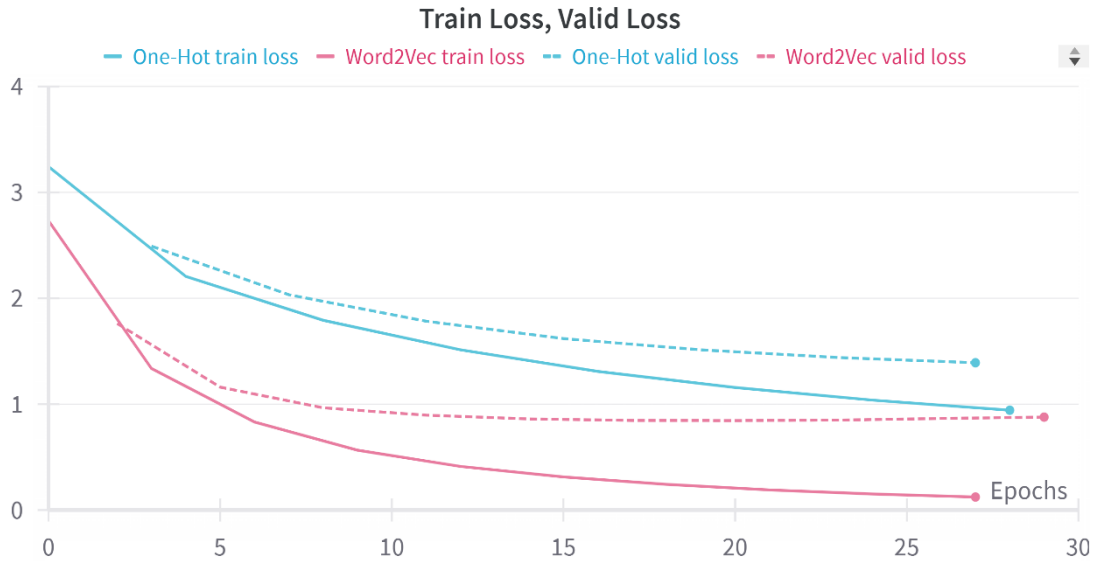
3. Experiments Realitzats

3.1. One-Hot Encoding vs Embedding Word2Vec

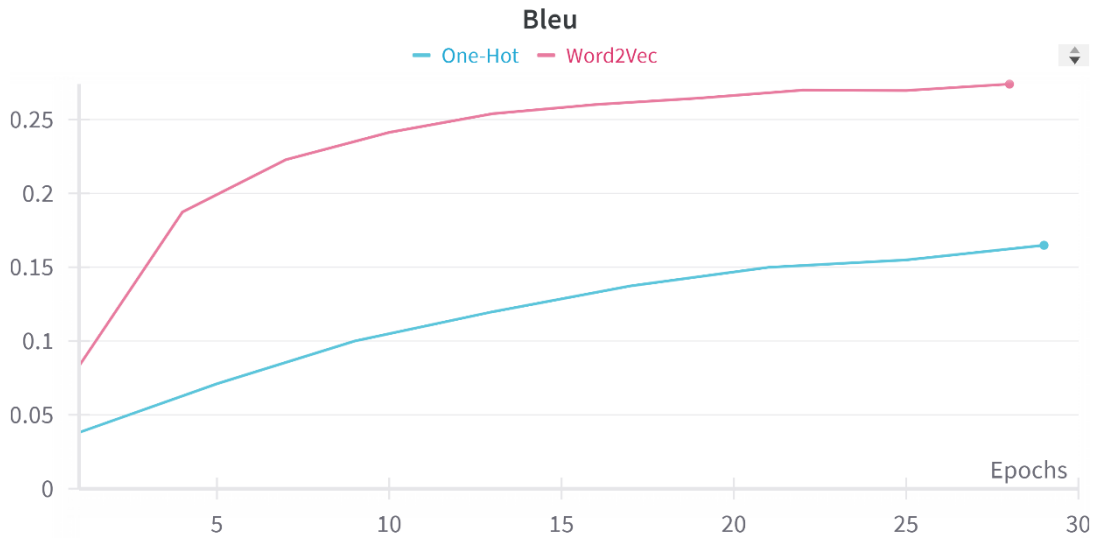
El model del Starting Point portava implementat una codificació dels tokens de la seqüència One-Hot que agafa el token a codificar i retorna un vector One-Hot que representa la paraula, és un vector binari amb tot zeros i un 1 a la posició que representa

la paraula dins del vocabulari. El principal problema d'una codificació One-Hot es que no captura informació semàntica dels tokens dins la seqüència i no té en compte el context de la paraula.

Per intentar millorar el model en aquest aspecte hem implementat un word embedding Word2Vec amb arquitectura CBOW (Continuous Bag of Words) que agafa el context de la paraula dins de la seqüència i retorna un vector Embedding amb la paraula target codificada. Aquest tipus de vectorització si que captura informació semàntica de la paraula i té en compte el context de cada paraula dins la seqüència.



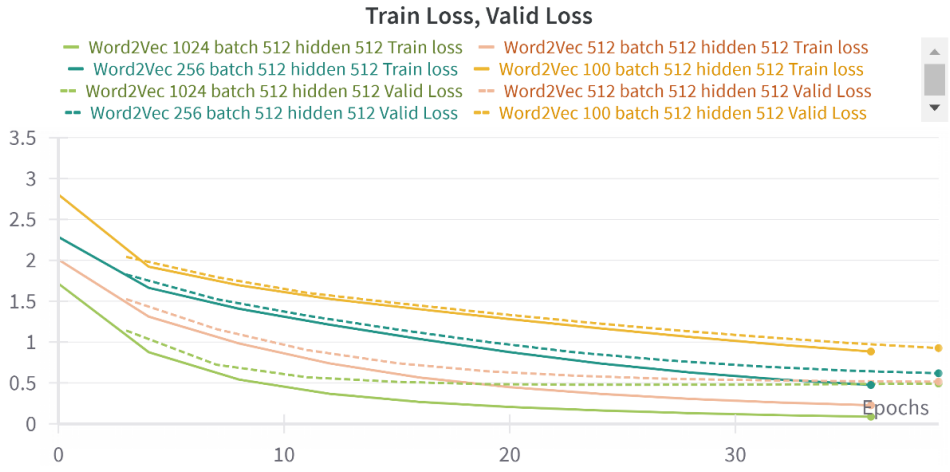
En aquesta gràfica podem veure que el model que porta implementat un word embedding Word2Vec obté un millor rendiment que el model que té implementat una codificació One-Hot.



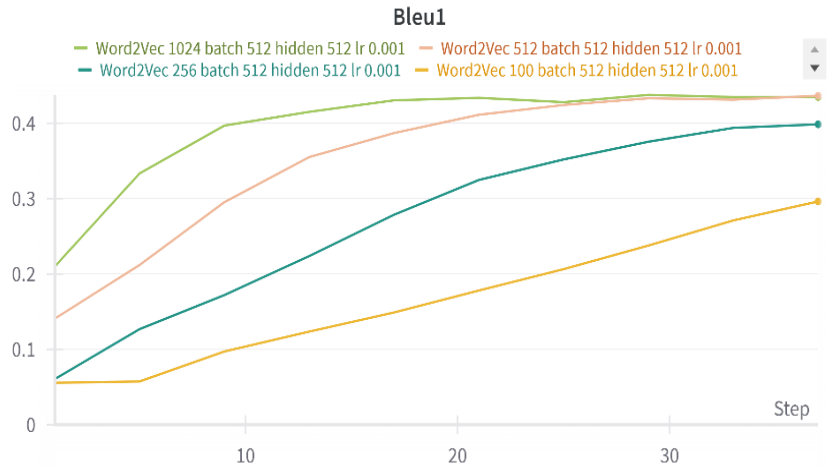
Si mirem un Bleu Standard, que calcula el Bleu score amb tots els n-grams possibles i fa la mitjana, també observem que el model Word2Vec obté millors resultats per aquesta mètrica, el que vol dir que el model amb Word2Vec genera unes millors traduccions que el que utilitza una codificació One-Hot. Amb aquests resultats obtinguts podem concloure que implementar un embedding de paraules que capturi informació semàntica i sintàctica dels tokens dins de la seqüència aporta una millora de rendiment i de qualitat de prediccions generades per el model.

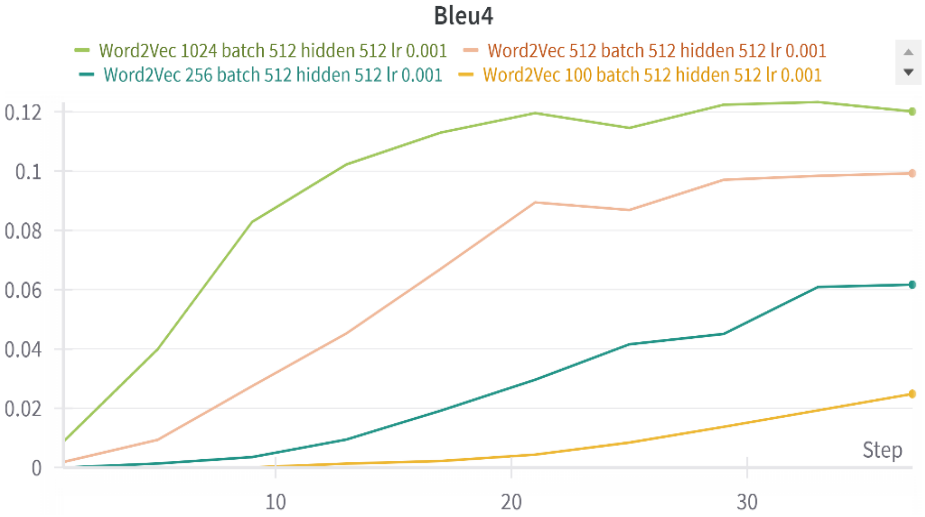
Mida del Vector Embedding

Una vegada hem vist que implementar un word embedding ens aporta una millora general al model volem veure que passa quan augmentem la mida del nostre vector embedding, mirant la teoria sabem que un vector embedding més gran aporta una major capacitat d'aprendre característiques més complexes de les dades.



Per realitzar aquest experiment hem executat el codi amb una mida del vector d'Embedding diferent, 4 mides 100, 256, 512 i 1024. A la gràfica podem veure que a mesura que augmentem aquesta paràmetre millora el rendiment del model, hem de tindre en compte que una mida excessivament gran del vector pot provocar overfitting al nostre model. Finalment a la gràfica veiem que les mides 512 i 1024 donen uns valors molt semblants per el que fa la loss del model.





Podem observar en aquestes dues gràfiques del Bleu score, tant amb n-grams de 1 com de 4, obtenim millors resultats a mesura que augmentem la mida del vector, vol dir que genera millors traduccions gràcies a tindre capacitat per aprendre característiques més complexes de les dades. Per al Bleu amb n-grams de 1 una mida de vector de 1024 i de 512 obté uns resultats molt semblants. Per al Bleu de n-grams de 4 una mida de vector de 1024 si que millora respecte les altres, el qual ens diu que una mida més gran del vector genera unes traduccions més coherents.

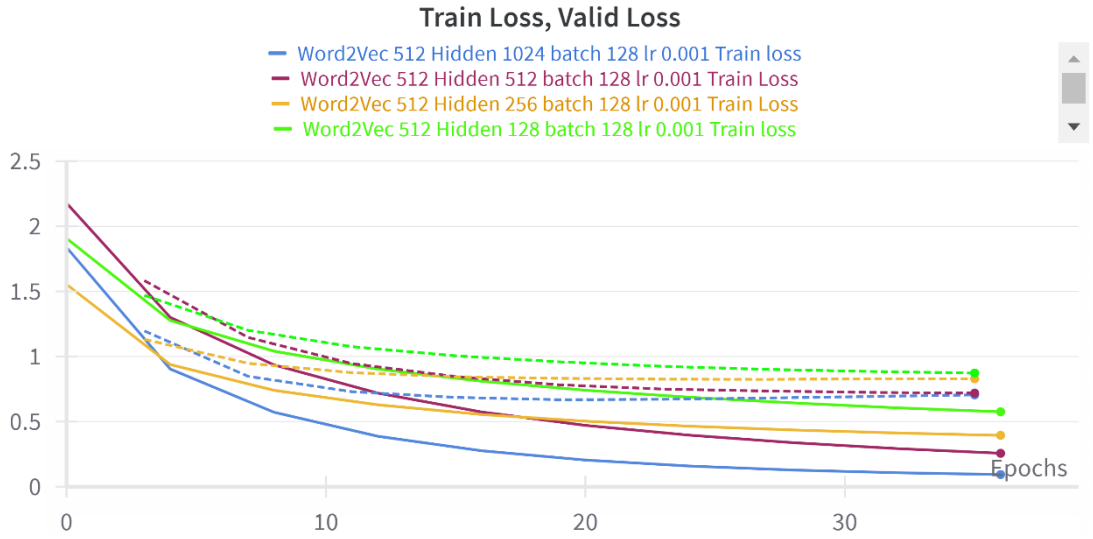
Després de realitzar aquest experiment podem concloure que un word embedding Word2Vec millora el model que utilitza One-Hot encoding. A més una mida més gran del vector que genera el word embedding ens aporta una major capacitat d'apendre característiques de les dades i més complexes, però també pot provocar overfitting al model si augmentem aquest paràmetre en excés.

3.2. Mida del Hidden State

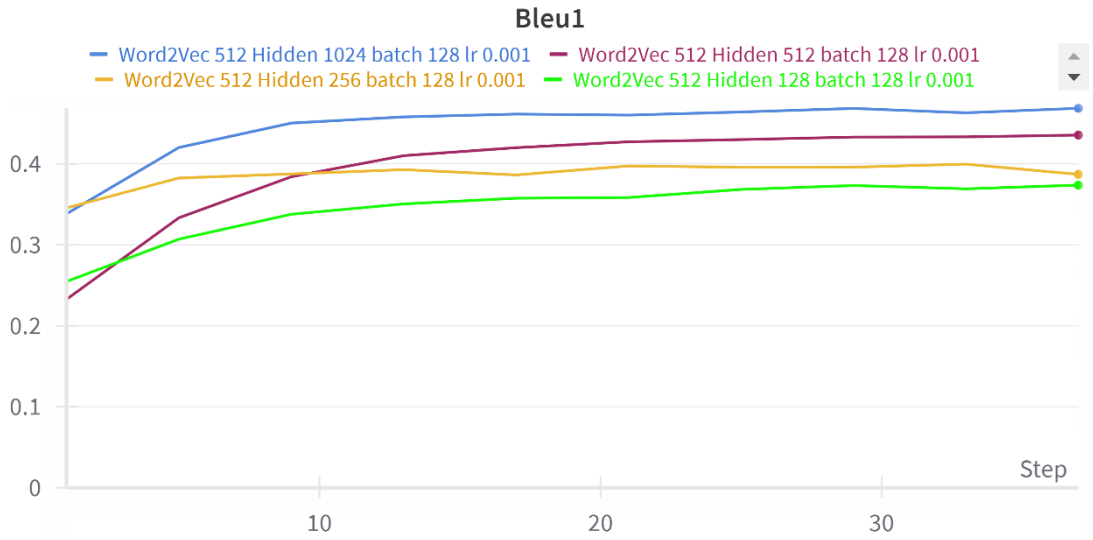
El paràmetre hidden_size especifica el nombre d'unitats en les capes ocultes d'una xarxa neuronal. En el codificador del model, determina la dimensió del vector d'estat ocult, que acumula la informació a mesura que es processa la seqüència d'entrada. En el descodificador, defineix la mida del vector d'estat ocult utilitzat per generar la seqüència de sortida paraula per paraula, basant-se en el vector de context i l'estat ocult anterior.

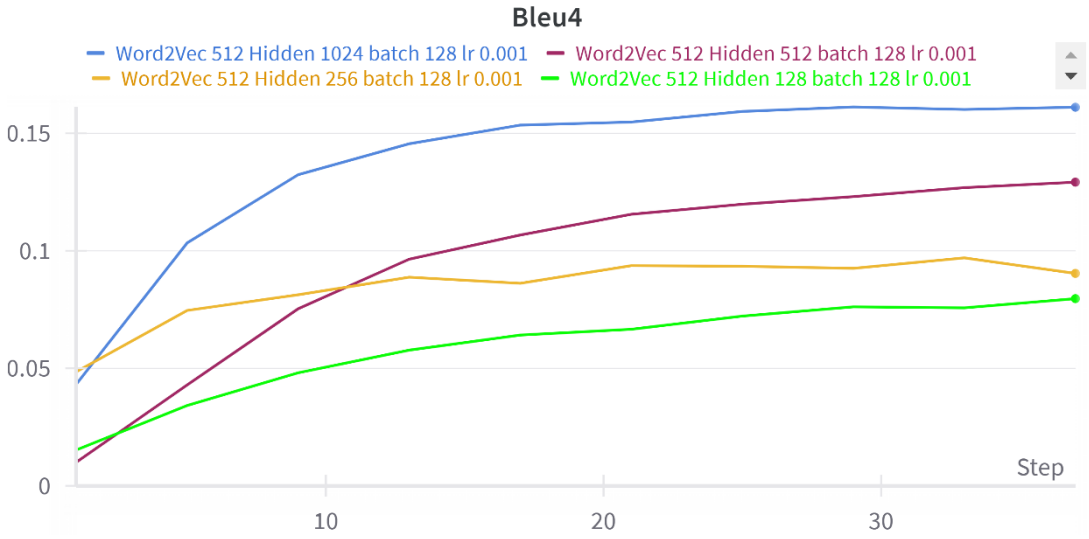
Un hidden_size més gran permet a la xarxa capturar més informació i patrons complexos en les seqüències d'entrada, però també augmenta el nombre de paràmetres del model, incrementant així el temps d'entrenament i la demanda de memòria. En canvi, un hidden_size massa petit pot no ser suficient per modelar les dependències en les dades adequadament, mentre que una mida massa gran pot provocar sobreajustament.

A continuació, vam realitzar amb 4 tamanys de hidden diferents (1024,512,256 i 128) i com es pot veure a la gràfica quan obtenim menys valid loss és amb un tamany de 1024, encara que un valor de 512 dona un resultat molt similar. També ens vam fixar en l'overfitting que amb 1024 tenim més que amb 512. Per tant, com a conclusions d'aquest experiment podem dir que contra més hidden_size tenim millors resultats obtenim, però que no ens podem excedir en aquest valor ja que sino patim overfitting. Per això, el valor que seleccionem com a millor es el 512.



També vam voler comprobar si el tamany del hidden afecta a la qualitat de les traduccions que obtenim. Per això vam ens vam centrar en els resultats del Bleu1 i el Bleu4. I com es pot veure en les dues gràfiques amb valors més grans de hidden size obtenim millors traduccions. A més, ens vam adonar que amb aquests valors més elevats tracten millors seqüències llargues de paraules.



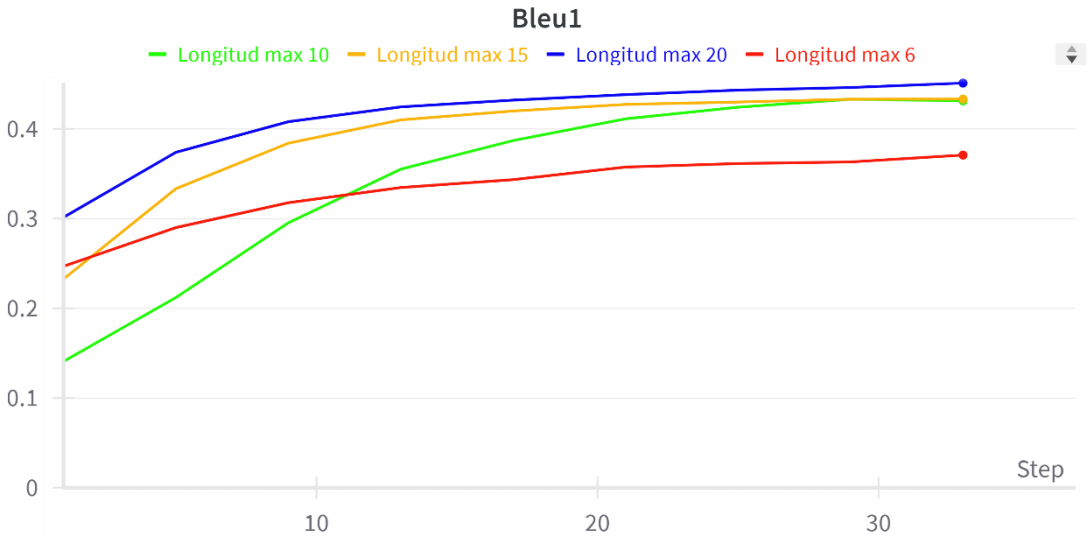


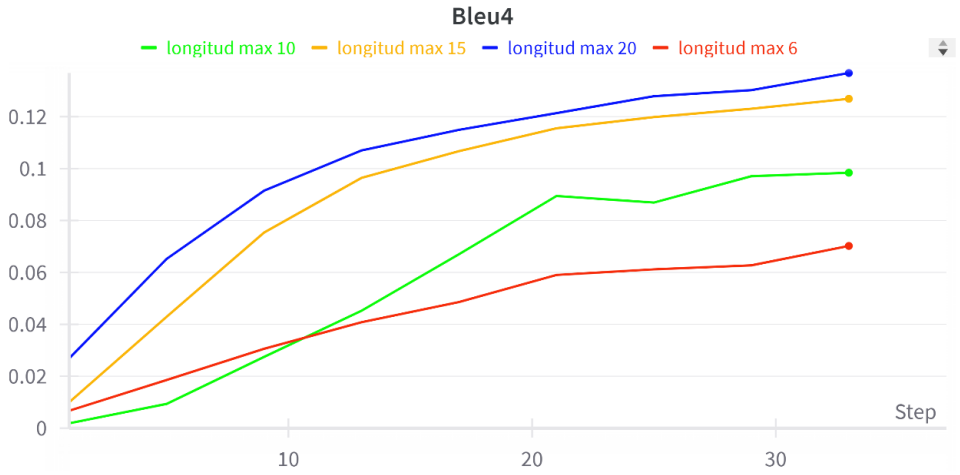
3.3. Longitud de les Frases

Seguidament, hem fet un anàlisi sobre el tamany de les dades, que en el nostre codi es modifica amb el paràmetre max_length. Hem dut a terme diferents proves on agafem des de 40.000 parells d'oracions (max_length de 6) fins a proves amb 150.000 parells d'oracions (max_length de 20). Com es pot veure a la gràfica a mesura que augmentem la longitud màxima de les frases obtenim millors resultats, ja que tant el valid com el traint loss van disminuint. Per tant, podem concloure que contra més longitud de frase tenim, més volum de dades tenim i conseqüentment millora el rendiment del nostre model.

Aquí tenim una comparació entre la longitud de frase més petita que hem provat que obtenim un valid loss d' 1,5 i es pot apreciar que té un overfitting més elevat que quan ho fem amb el tamany més gran que arribem a obtenir un valor de valid loss de 0,5, reduint així en 1 punt sencer.

També hem volgut analitzar si el tamany de les frases afectava la qualitat de les traduccions. I com es pot veure en les dues gràfiques a mesura que augmentem el tamany de les frases obtenim millors traduccions. A més, al tenir una longitud de frase més llarga el nostre model pot capturar millor el context de la frase i, per tant, traduir millor.





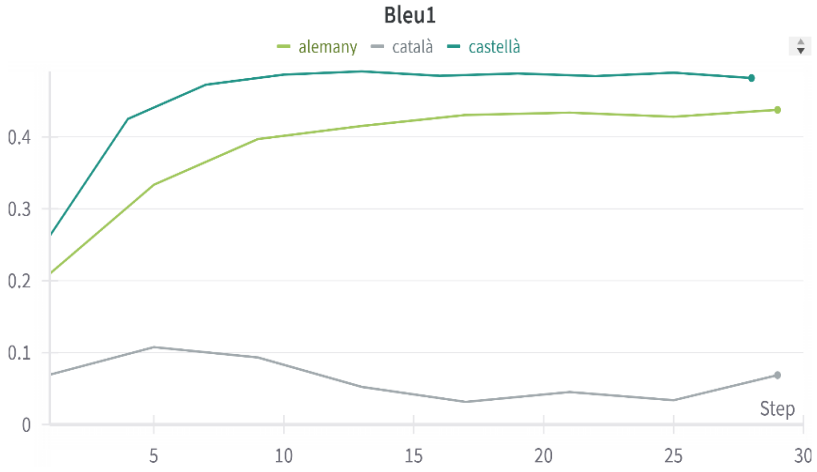
3.4. Comparativa amb diferents idiomes.

Idiomes amb diferents mides del vocabulari i de dades.

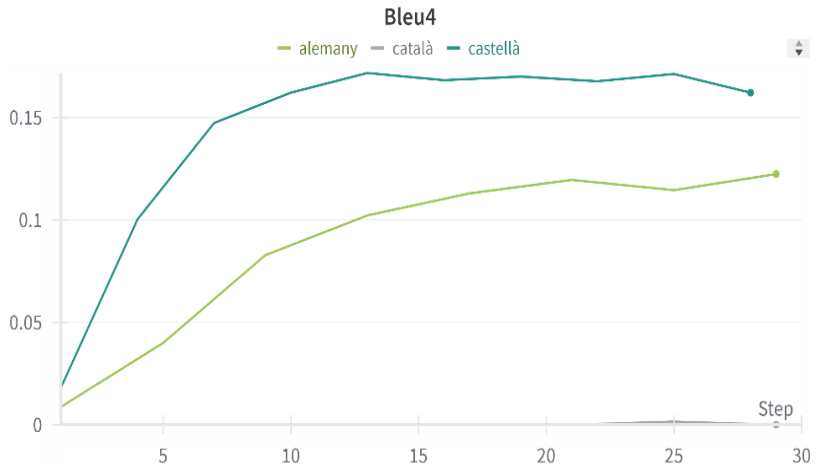
Primer volem realitzar un experiment per veure si idiomes amb vocabularis més grans i amb un major volum de dades obtenen millors resultats al executar el model amb ells. Per aquest experiment hem utilitzat datastes en diferents idiomes. El català amb 1375 frases de longitud màxima de 18 caràcters i un vocabulari de 1819 paraules úniques, també un data set en castellà amb 141543 frases de longitud màxima de 30 caràcters i un vocabulari de 26889 paraules úniques, finalment un data set en alemany que n'hem utilitzat només el 60% d'aquest per temes computacionals, el qual conté 158000 frases amb una longitud màxima de 15 caràcters i un vocabulari de 25483 paraules úniques.

En aquesta primera gràfica veiem clarament que el model que ha utilitzat el data set en català obté uns resultats de rendiment molt pitjors que les altres dues execucions degut a les poques dades disponibles que tenim en català.

Per altre banda, l'execució em castellà ha obtingut uns millors resultats ja que s'ha utilitzat la totalitat del data set amb frases de fins a 30 caràcters, per tant ha utilitzat més dades que l'execució en alemany i per tant, ha obtingut millors resultats de rendiment.



En aquesta segona gràfica veiem els resultats de les diferents execucions amb la mètrica de Bleu amb n-grams de 1 on podem tornar a veure que les traduccions generades en català son molt pitjors que les realitzades amb altres idiomes, al utilitzar aquest Bleu paraula a paraula les traduccions en català encara coincideixen en algunes paraules respecte a les de referència. Per altre banda, en castellà tornem a obtindré millors resultats que en alemany.



Per a un Bleu amb n-grams de 4 podem observar que les traduccions en català pràcticament no aconsegueixen obtindré resultats majors que 0. Tornem a veure que el castellà obté millors resultats que l'alemany.

Podem concloure que augmentar la mida de les dades utilitzades, ja sigui una major quantitat de frases o una major longitud fa que el model obtingui millors resultats. Aquest experiment es podria ampliar mirant l'efecte de la mida del vocabulari per veure si la

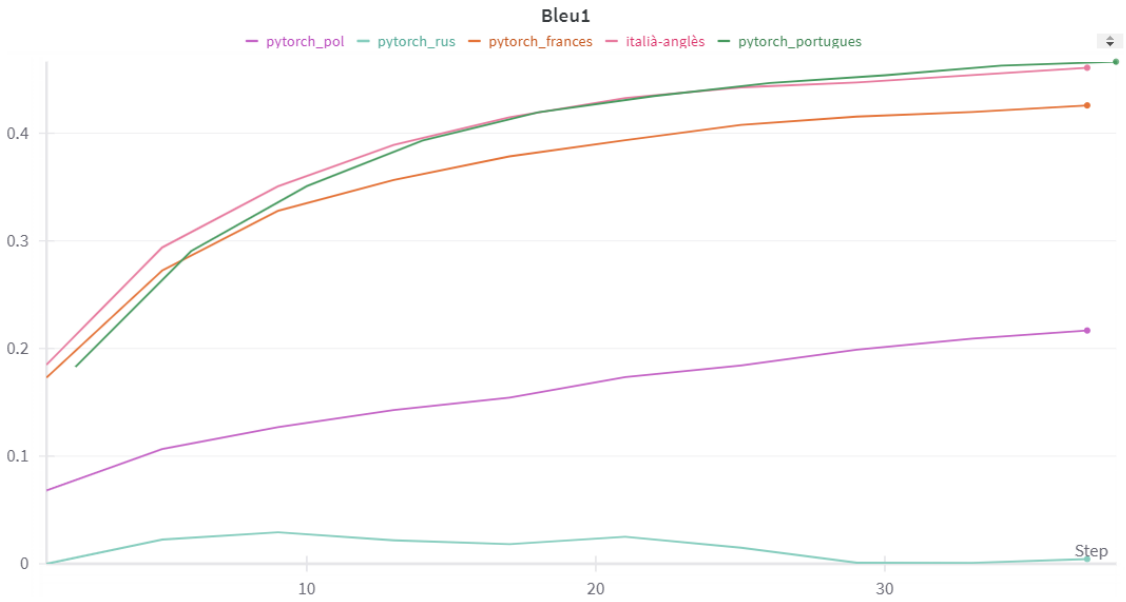
millora del rendiment és degut a augmentar la quantitat de dades o té alguna relació amb el vocabulari obtingut al processar les diferents seqüències de l'idioma.

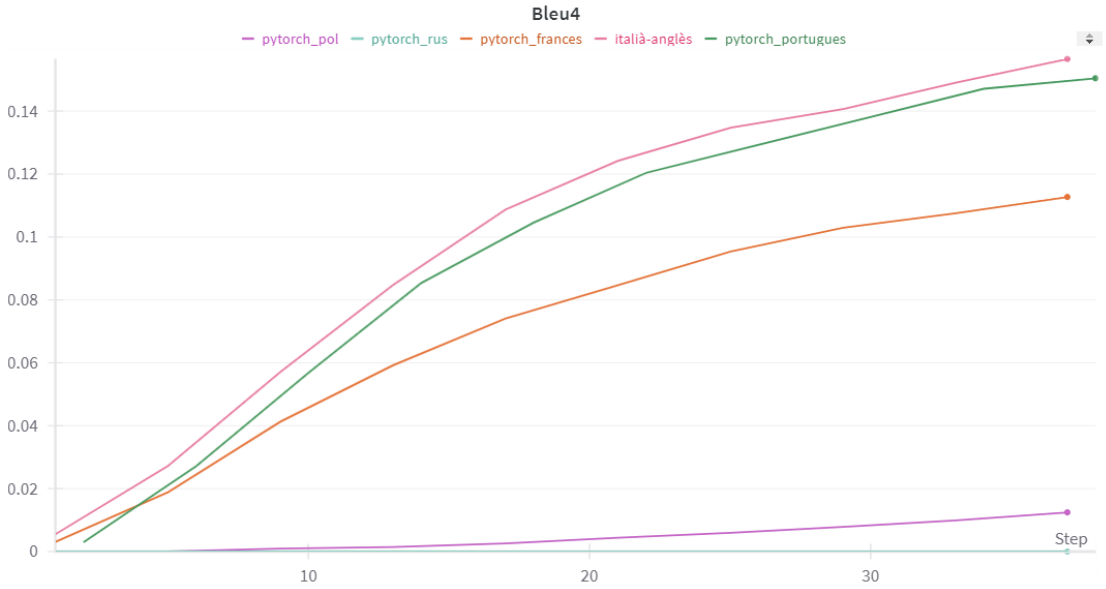
Famílies d'idiomes.

També hem volgut veure el rendiment del model amb diferents idiomes de diverses famílies lingüístiques. Per realitzar-ho hem utilitzat dos idiomes eslaus com el Rus o el Polonès i idiomes llatins com el Francès, l'italià o el Portuguès.

A la gràfica podem observar com els idiomes eslaus com el Rus no millora el rendiment ja que no capta bé el significat de la frase degut al canvi de l'alfabet, a una escala més reduïda tenim el cas del Polonès que si que millora més el rendiment del model ja que l'alfabet es més semblant al de l'Anglès. Podem concloure que alfabets més diferents complicaran la realització de traduccions, es possible que per aconseguir aquestes traduccions amb idiomes com el Rus es necessitin certs preprocessaments i embeddings especials per capturar informació significativa de les dades de text amb aquests alfabets.

Per altre banda, els idiomes llatins tenen una evolució molt millor que els eslaus i ho podem veure a la gràfica.





Per el que fa el Bleu, tant amb n-grams de 1 com de 4 es veu una diferència abismal a les puntuacions de Bleu, on el Rus i el Polonès estan molt propers al 0 i els llatins arriben a valors més normals amb el vist anteriorment.

Hem detectat problemes perquè l'alfabet diferent impedeix que el model capti les paraules correctament, resultant en una pèrdua que no disminueix i un BLEU de 0. Així doncs, no hem pogut experimentar adequadament amb la família eslava, com el rus o l'ucraïnès. El BLEU1 del rus ofereix un petit rendiment perquè capta només frases curtes, però amb BLEU4 no hi ha cap rendiment.

Podem observar que el BLEU1 dona millors resultats que el BLEU4, ja que és més fàcil traduir una paraula sola que no pas un conjunt de paraules que formen una frase i que requereixen traduir la gramàtica i la sintaxi correctament. En les llengües llatines hem seleccionat espanyol, italià, portuguès i francès degut a la gran quantitat de dades disponibles per realitzar un bon experiment: 170k frases de màxima longitud de 10. Com que hi ha datasets més grans que altres, hem seleccionat una longitud màxima de 10 i hem mostrejat per a que la distribució de llargada fos uniforme entre tots, garantint un bon experiment.

En conclusió es més fàcil traduir amb les llengües llatines o les que tenen un alfabet similar entre elles que amb les llengües eslaves o orientals que tenen un alfabet diferents i que es necessitarien processos especials de preprocessament de dades.

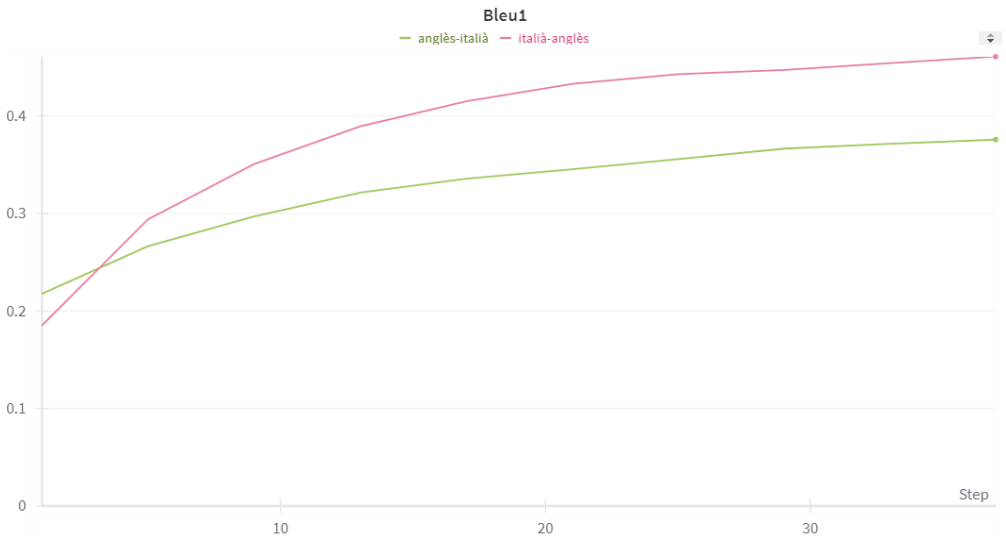
3.5. Traducció inversa (Bidireccional).

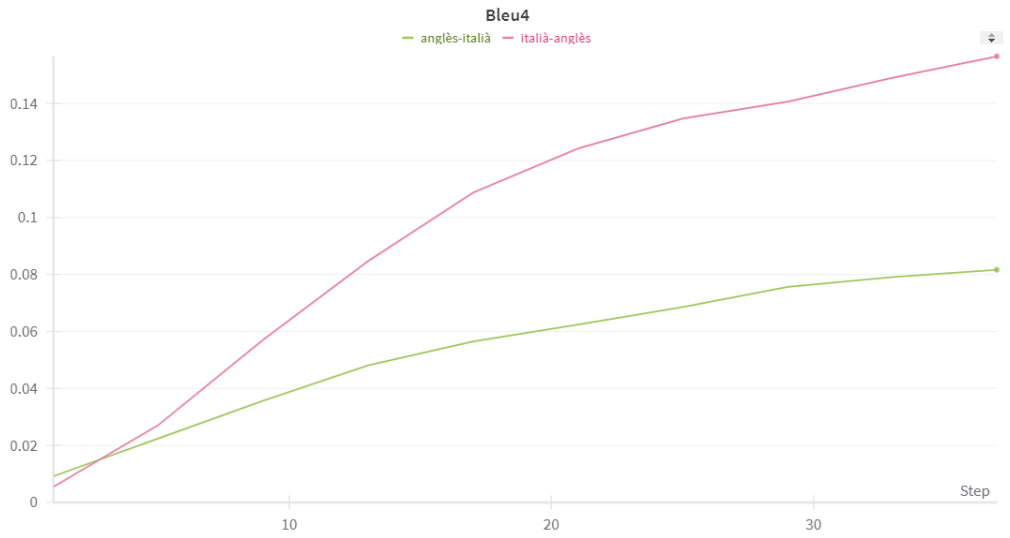
Hem volgut veure quin seria l'efecte de traduir en l'altre direcció de la que estàvem realitzant, es a dir des de un idioma a l'anglès. Per això hem utilitzat el datastet Italià i hem entrenat el model per tal de que aprengui a traduir tant de l'Anglès a l'Italià com de l'Italià a L'anglès i veure si hi ha algun efecte al rendiment del model.

Podem veure que traduir des de l'italià dona millors resultats que traduir des de l'anglès.

L'anglès té 14k paraules en el vocabulari, mentre que les llengües llatines tenen aproximadament les mateixes paraules (20,000): portuguès 19k, italià 20k, francès 21k i espanyol 23k.

Amb les dades del vocabulari dels diferents idiomes i la gràfica anterior podem dir que al tindre un major vocabulari de l'idioma d'origen obtenim millors resultats que quan traduïm des de un idioma amb menys vocabulari a un que més. Per tant, podem dir que es més important tindre un vocabulari gran d'origen que de destí per obtindré millors resultats i generar millors traduccions.





Per els diferents Bleu obtenim el mateix resultat, el qual és millor traduir des de l'Italià (idioma amb major vocabulari) que des de l'Anglès (idioma amb menor vocabulari).

També hem de ressaltar, com a dinàmica general als diferents experiments, que amb un BLEU de n-grams de 1 es mostra un rendiment més semblant entre les puntuacions de les diferents execucions ja que és més simple avaluar les traduccions paraula a paraula que en agrupacions de diferents paraules.

4. Execució Final.

Paràmetres i configuració final establerta.

Finalment, vam fer la nostra execució final amb els hiperparàmetres que hem anat comentant al llarg de l'informe que son els següents:

Mida embedding: 512

Mida hidden size: 512

Mida batch: 128

Lr: 0.001

Max length: 20

150k frases

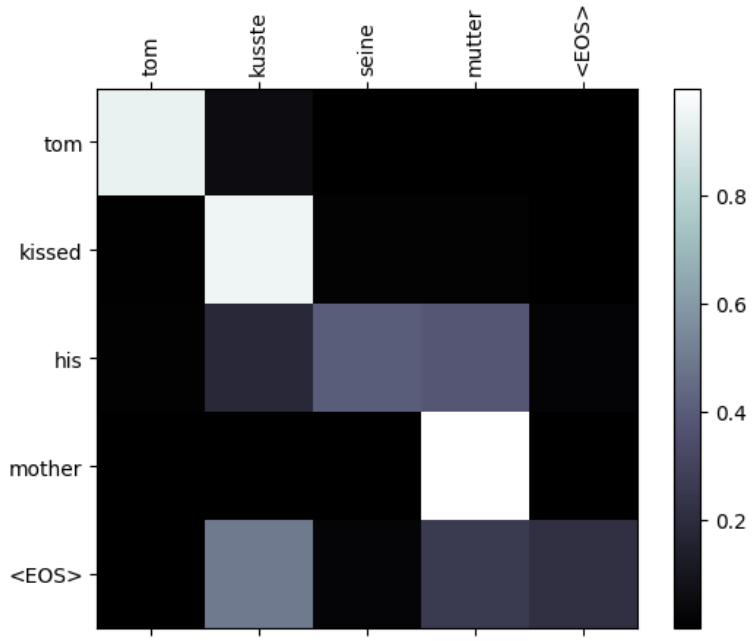
Com podem observar en la gràfica final aconseguim reduir l'overfitting molt respecte el model inicial on el valor de valid loss estava per sobre de 1,5 i aquí el reduïm fins a 0,5.

Matriu d'atenció i exemples de traduccions.

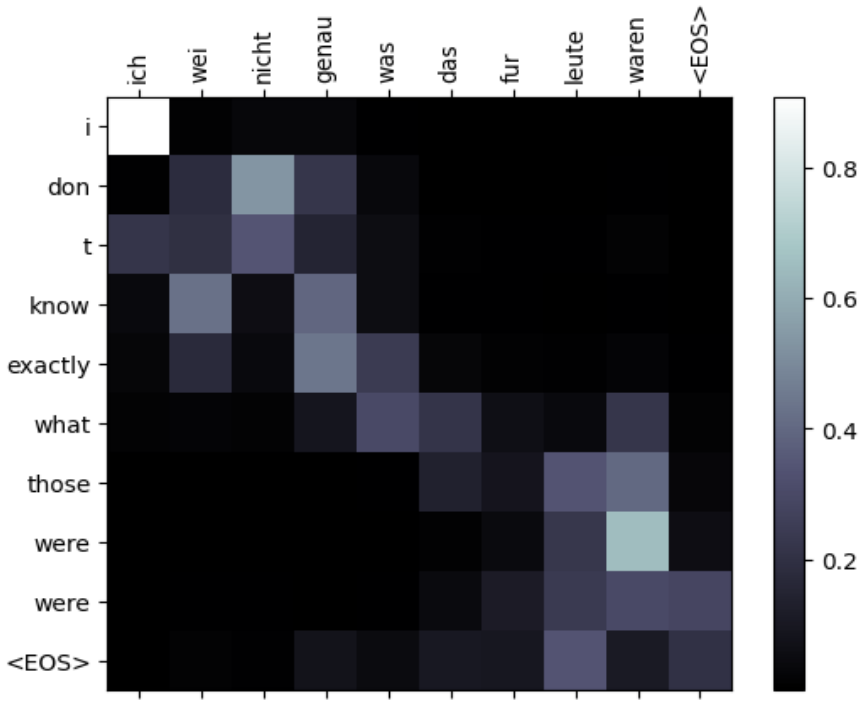
Com podem veure les traduccions que fa actualment per a frases llargues no es del tot perfecte ja que distribuir l'atenció en una frase llarga no es tan fàcil com en una curta. En canvi, les frases curtes sí que les tradueix de forma exacte.

Per visualitzar clarament això hem generat unes matrius d'atenció que ens mostren com es relacionen les paraules d'un idioma amb les de l'altre durant una traducció. En el nostre cas en concret a l'eix vertical trobem les paraules de sortida en anglès i a l'eix horitzontal les paraules d'entrada en alemany. Cada cel·la en la matriu conté un valor que indica el "pes" o l'atenció que es presta a la paraula d'entrada (columna) mentre es genera la paraula de sortida corresponent (fila). Gràcies a les matrius d'atenció podem veure com el model distribueix la seva atenció sobre les paraules d'entrada per generar cada paraula de l'oració de sortida, facilitant una millor traducció mitjançant la consideració de contextos complets.

```
input = tom kusste seine mutter
reference = tom kissed his mother
output = tom kissed his mother <EOS>
```




```
input = ich wei nicht genau was das fur leute waren
reference = i m not sure who they were
output = i don t know exactly what those were were <EOS>
```



5. Conclusions

Hem realitzat diversos estudis per millorar el rendiment del nostre model de traducció automàtica, obtenint resultats prometedors en diversos aspectes tècnics. Primerament, utilitzar un embedding de paraules Word2Vec en comptes d'una codificació One-Hot ha demostrat ser molt beneficiós, ja que l'embedding captura informació semàntica de les paraules, la qual cosa enriqueix la representació dels textos i millora la qualitat de les traduccions.

Augmentar la mida del vector embedding també ha portat millores significatives. Tanmateix, cal tenir cura amb una mida excessivament alta, ja que pot provocar overfitting, on el model aprèn massa bé les dades d'entrenament però falla en generalitzar a noves dades.

L'ampliació del vocabulari utilitzat pel model ha resultat en millors traduccions, ja que el model té accés a una gamma més àmplia de paraules i expressions. Això permet una traducció més precisa i natural, reflectint millor les diverses formes d'expressió del llenguatge.

Pel que fa als estats ocults, un augment de la seva mida permet capturar característiques més complexes de les dades, millorant el rendiment del model en tasques de traducció. Tot i això, igual que amb els embeddings, una mida excessiva pot portar a problemes d'overfitting.

Finalment, utilitzar una longitud màxima de frases més elevada ha augmentat el vocabulari i la precisió del model. Això permet que el model processi i entengui millor frases llargues i complexes, millorant la seva capacitat per gestionar contextos amplis i mantenir la coherència en les traduccions.

En conjunt, aquestes millores tècniques han contribuït a un model de traducció automàtica més robust i eficient, capaç de produir traduccions més precises i naturals.