Градење модел на длабоко учење за Сентимент анализа со помош на BERT

1. **Вовед**

*Зошто сентимент анализа?*

Една од главните причини поради која сентимент анализата е значајна е тоа што најчесто делата што ги правиме се темелат на нашите емоции. Нашите емоции претставуваат мотив за дејствата кои ги извршуваме секојдневно. Емоцијата која ја има во различни видови како состојба се јавува кај сите луѓе. Со други зборови емоциите кои ги покажуваме говорат за нашиот карактер.

Тогаш зошто да не го одредиме карактерот на човекот наместо да одредуваме каква емоција ќе предизвика одредена ставка кај него?

Човекот како систем е сложен за да може да се класифицира.

И тестовите кои што постојат денес, кои го одредуваат карактерот на човекот служат со цел да кажат каков е карактерот на човекот во поголем дел од времето, не порекнуваат дека карактерот на човекот не може да се смени.

Делот на кој ќе се задржиме во овој извештај е да се одреди сентимент на твитови според објективен поглед врз твитовите, без разлика на тоа каква би била емоцијата кај поединци.

1. **Објектива**

Во овој извештај ќе бидат разгледувани две податочни множества и тоа:

* 140 Сентимент
* CrowdFlower

140 Сентимент податочното множество има два типа лабели на твитовите (позитивен и негативен). Како множество со над милион твитови, 140 Сентимент податочното множество ќе биде искористено за градење на модел за валентоност на твитови кој ќе биде искористен над податочното множество CrowdFlower. Дополнително со помош на лексиконот NRC-VAD и податочното множество 140 Сентимент ќе биде изграден и модел за побудување (arousal) и модел за доминантност (dominance) кои исто така ќе бидат применети врз твитовите од множеството CrowdFlower.

1. **Претпроцесирање на твитови**

Претпроцесирањето на твитови претставува обемен процес и во суштина тоа е сегментот кој ќе има големо влијание врз крајниот резултат што ќе го добиеме од моделот што го градиме.

Затоа потребно е да се земат во предвид сите аспекти на човековото изразување и да се покријат сите не толку очигледни тест случаи.

Со оглед на тоа врз какво податочно множество бил трениран моделот на длабоко учење BERT кој што ќе биде користен за сентимент анализата, потребно е на сличен начин да ги имаме претпроцесирано твитовите со кои ќе го продолжиме тренирањето или би направиле фино изменување со мала рата на учење.

Во рамките на ова претпроцесирање беа употребени следните правила:

1. Отстранети беа сите интерпукциски знаци

на два начина:

* заменети со празно место (за полесно отстранување на стоп зборовите)

пример: don t

* заменети со ништо

(реченица што ќе биде влез во тренирачкото множество)

пример: dont

1. Сите големи букви беа заменети со мали
2. Отстранети беа сите означени корисници на твитер што започнуваат со симболот @ и заменети со usertag
3. Отстранети беа сите html ентитетни референци како &amp;
4. Отстранети беа сите референци и линкови и заменети со linktag
5. Отстранети беа сите не ASCII карактери
6. Отстранети беа сите броеви (при користење на лексиконите)
7. Остранети беа сите повеќе празни места меѓу зборовите.
8. Извршена беше лематизација со цел полесно наоѓање зборови во лексиконот

Пред претпроцесирањето беше извршена и проверка дали твитот содржи извичник (пример: cant!), зборови напишани со сите големи букви (пример: SOON) и букви што се повторуваат во зборови од твитот, (пример: hereeee)

Единствено нешто што не беше извршено во претпроцесирањето е отстранувањето на Named Entity Recognition, поради тоа што имаше слаби перформанси.

Пример: Во твит каде што Emily е име кое треба да биде препознаено како ентитет, е класифицирано како прилог (adverb) поради тоа што завршува на ly.

1. **Лексикони**

Зборовите во лексиконот NRC-VAD се претставени на следниов начин:

|  |  |  |  |
| --- | --- | --- | --- |
| **word** | *Valence* | *Arousal* | *Dominance* |
| example | 0.0 – 1.0 | 0.0 -1.0 | 0.0 – 1.0 |

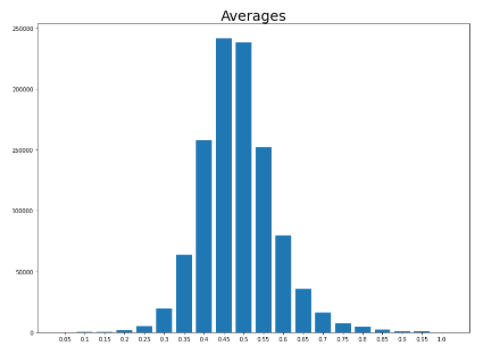
Лексиконот VAD содржи повеќе од 20000 зборови. Откако беше извршено препроцесирање зборовите беа мапирани во броеви соодветно за побудувањето и за доминантноста.

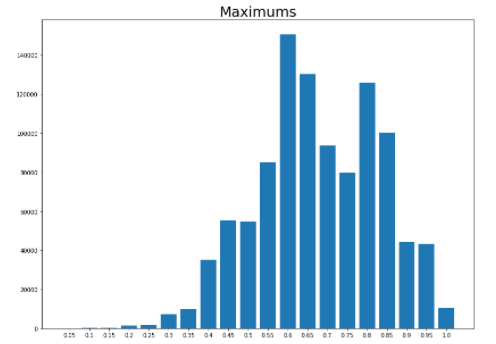
|  |  |  |
| --- | --- | --- |
| оригинален | претпроцесиран | мапиран |
| my whole body feels itchy and like its on fire | whole body feel itchy like fire | [0.357, 0.461, 0.602, 0.726, 0.353, 0.804] |
| usertag linktag awww thats a bummer you should... | awww bummer shoulda get david carr third day | [-1, 0.5, -1, 0.594, -1, 0.54, 0.187, 0.269] |

Доколку одреден збор не е пронајден во лексиконот неговата вредност во мапата за побудување/доминантност е -1.

Начинот на кој се лабелирани твитовите за побудување / доминантност претставуваше просекот / максимумот од мапираните зборови.

Со цел да се намали загубата на информација во крајното податочно множество за моделите на побудување и доминантност беа исфилтрирани твитови кои имаат повеќе од половина пронајдени зборови од лексиконот.





1. **Стратификација на твитови**

За да се изберат записи со кои ќе биде трениран моделот BERT беше направена стратификација на твитовите. Со цел да се заштеди на време за тренирање на предтренираниот модел BERT како и процесорска моќ беше избран мал BERT модел со 2 слоја од 256 филтри и 4 слоја со механизам на внимание. Поради тоа што сакаме да предвидуваме сентимент на твитови од податочното множество на „CrowdFlower“ беше извршено филтрирање над твитовите од податочното множество на „140 Сентимент“ така што во тренирачкото множество ќе влезат твитови кои содржат триграми кои ги има во твитовите од податочното множество „CrowdFlower“.

1. **Градење на моделот за валентност**

За дефинирање на влезот на моделот се користат tokenizer кој е дефиниран од дадени параметри на моделот (дали моделот е трениран на lowercased текст и вокабулар кој се симнува при симнување на предтренираниот BERT модел). Токенизаторот ги дели твитовите на токени што во суштина се дел од зборови (не цели зборови). По токенизацијата за тренирање се земаат ID вредностите од токените. За дефинирање на излез се користи one-hot encoding.

Предтренираниот мал модел BERT на влез прима речник (dictionary) со три клуча и тоа:

* input\_word\_ids – ragged tensors или вгнездена листа од тензори во која се ID – вредностите на токените.
* input\_mask – ragged tensor за маскирање на токените
* input\_type\_ids – ragged tensor за типот на токенот, дали се работи за регуларен токен или токен од типот [CLS] или [SEP].

\*\* [CLS] токенот е токен за класификација кој ја овозможува шемата на pooling кај BERT и се додава на почеток на тензорите, додека [SEP] токенот се додава на крајот од тензорите за одделување на речениците.

Пред да се енкодираат токените во ID вредностите потребно е да се избере максимална големина на тензорите што исто така е важен сегмент од моделот кој го градиме. Имено големината на тензорот треба да биде малку поголема од максималната големина од тензорите на токенизирани твитови (во овој случај 1.5 \* максималната големина), со цел доколку во тестирачкото множество имаме тензор на твит поголем од максимумот пронајден во тренирачкото множество да можеме успешно да го претставиме. Сепак треба да земеме предвид дека доколку оваа вредност ја зголемиме многу пати, освен што ќе добиеме многу непополнети ID на токени ќе изгубиме ID на токени што носат корисна информација при pooling на вистинските ID вредности на токените.

***Репрезентација на моделот за валентност***

|  |  |  |
| --- | --- | --- |
| input\_word\_ids | input\_mask | input\_type\_ids |
| pretrained\_BERT | | |
| Conv1D 128 Kernel = 3 | | |
| Conv1D 128 Kernel = 3 | | |
| MaxPool1D Size = 2 | | |
| Conv1D 64 Kernel = 3 | | |
| Conv1D 16 Kernel = 3 | | |
| Flatten | | |
| Dense 2 | | |