UNIVERSITAS INDONESIA

MENGKLASIFIKASIKAN *TWEET* LAPORAN BENCANA BANJIR MENGGUNAKAN *SEMI-SUPERVISED MULTI-MODAL DEEP LEARNING* GAMBAR DAN TEKS

MAKALAH

THREE EPOCHS
Eko Julianto Salim
Jonathan Nicholas
Jovi Handono Hutama

PEMBIMBING
Evi Yulianti, S.Kom., M.Kom.t, M.Comp.Sc., Ph.D.

FAKULTAS ILMU KOMPUTER
2020

DAFTAR ISI

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Selama dekade terakhir, Twitter sudah menjadi sebuah alat yang berguna untuk mendeteksi dan melacak bahaya yang ada di lingkungan seperti banjir, kebakaran hutan, dan gempa bumi. Data yang ada di Twitter telah diaplikasikan dalam berbagai fase dari siklus penanggulangan bencana, contohnya untuk membantu pemulihan korban bencana dan mendapat bantuan dengan cepat.

Dalam konteks bencana banjir di Indonesia, data *tweet* banjir dari Twitter sudah pernah dan juga dapat digunakan untuk meng-*crowdsource* informasi tentang banjir. Salah satu skema dari *crowdsourcing* ini adalah dengan merujuk pengguna twitter yang menge-*tweet* laporan banjir untuk mengisi laporan di *website/platform* terpisah (melalui reply ke *tweet* mereka). Mengumpulkan dan memantau *tweet-tweet* ini dapat menunjukkan dimana bencana terjadi dan memberikan data yang berguna untuk *aid organizations*.

Namun, setiap harinya ada banyak sekali *tweet* yang berisikan kata ”banjir”, sebagai contoh, prakiraan banjir dan penampakan banjir. Tetapi kata ”banjir” ini juga sering digunakan dalam menggambarkan sesuatu hal yang dapat dikatakan berlebihan, seperti misalnya banjir air mata, banjir pesanan, dan banjir pujian. Selain itu juga banyak *tweet* yang isinya tidak relevan di saat bencana banjir, seperti candaan, promosi, ataupun iklan. Oleh karena itu, pemfilteran yang efektif dari *tweet-tweet* ini diperlukan untuk menambah performa *downstream algorithms*, memperbagus *user experience*, dan mengurangi tenaga kerja manual.

Tantangan lain adalah *nature social media* sekarang ini dimana konten yang ada semakin beragam. Sebuah *tweet* dapat mengandung video, url, gambar, teks dan lain-lain. Sebuah modalitas (baik teks atau gambar) tidak dapat memberikan hasil yang terbaik. Oleh karena itu, penting untuk menggabungkan modalitas dan mengevaluasi hasilnya. Mengeksploitasi informasi dari berbagai pandangan juga dapat meningkatkan akurasi sistem secara keseluruhan.

Selain itu pada era digital ini, Big Data bukanlah sesuatu yang jarang ditemukan. Setiap harinya ada banyak sekali *tweet* yang dibuat oleh para pengguna, jumlahnya yang sangat banyak dapat dikategorikan Big Data. Dampaknya, twitter harus menyediakan tempat penyimpanan yang besar dan sistem yang kuat dan juga ”mahal” untuk melabel data untuk *supervised learning*. Oleh karena itu digunakanlah *semi-supervised learning*.

1.2 Tujuan dan Manfaat

Kami bertujuan untuk mengembangkan sistem klasifikasi yang kuat yang memanfaatkan tekstual maupun visual untuk sampai pada prediksi apakah *tweet* yang diberikan dan gambar yang terkait dengannya mengandung *actionable intelligence* atau informasi tentang situasi banjir sekarang (yang pantas untuk di *follow-up* serta diinput ke sistem sentral) atau situasi yang tidak relevan dengan banjir.
Manfaat (dalam konteks banjir):

1. Analisis kegunaan pretrained *image* dan *text model* (EfficientNet dan BERT) dalam klasifikasi ini.
2. Analisis akurasi menggunakan *text*, *image*, atau *text* dan *image* dalam menentukan apakah suatu *tweet* bersifat informatif + *error analysis*
3. Mengevaluasi berbagai macam metode fusi / penggabungan multi-modality
4. Mengevaluasi dampak *semi-supervised learning* dibanding *supervised learning* biasa
5. Pembuatan model yang dapat membantu *aid organizations* menolong korban banjir

Selain itu, model yang akan dibuat juga dapat di-*deploy* untuk mengklasifikasi *tweet* yang di sebarkan oleh orang-orang secara real-time melalui *API* dengan kata kunci "banjir" agar pihak yang berwajib dapat segera memberikan bantuan dengan cepat. Tidak hanya pihak berwajib, informasi tentang banjir pun juga dapat disebarkan melalui *platform* lain sehingga masyarakat luas dapat dengan sigap mengatasi banjir sebelum terlambat.

1.3 Batasan yang digunakan

Dalam dataset pada makalah kali ini, batasan kami adalah sebagai berikut:

1. *tweet* yang digunakan hanyalah *tweet* yang mengandung teks dan gambar jika ada. Jika *tweet* hanya mengandung teks dan video, maka diambil thumbnail dari video yang bersangkutan.

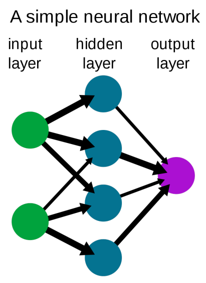
2. Hanya 1 gambar per *tweet* yang dipertimbangkan
3. Hanya *tweet* yang berbahasa Indonesia
4. Label *tweet* hanya terdiri jadi 2 jenis, yaitu *tweet* yang layak atau tidak layak untuk di-follow-up.
5. *tweet* yang digunakan hanya bersumber dari *tweet* yang mengandung *keyword* 'banjir' dan bukan *tweet* lainnya

BAB 2
METODE

2.1 Metode Penambangan Data

2.1.1 Neural Network

Sebuah sistem komputasi yang terdiri dari sejumlah elemen pemrosesan sederhana yang saling berhubungan dengan kuat, yang memproses informasi dengan respons keadaan dinamisnya ke input eksternal. Neural Network termasuk hal terpenting dalam Deep Learning.



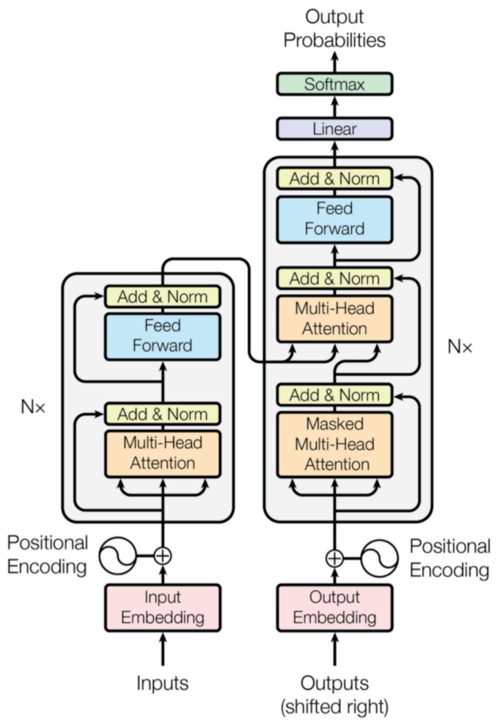
Gambar 2.1: Contoh neural network

2.1.2 EfficientNet

Convolutional Neural Network (CNN) adalah salah satu jenis neural network yang biasa digunakan pada data image dan dikembangkan dengan anggaran *resource* tetap, kemudian ditingkatkan untuk mencapai akurasi yang lebih baik ketika lebih banyak *resource* tersedia. Implementasi dari CNN, **EfficientNet**, model yang ditemukan karena dilakukannya neural architecture search (NAS) dimana awalnya ditujukan untuk merancang baseline network baru dan meningkatkan performanya. EfficientNet memiliki akurasi dan efisiensi yang jauh lebih baik dibanding CNN.

2.1.3 BERT

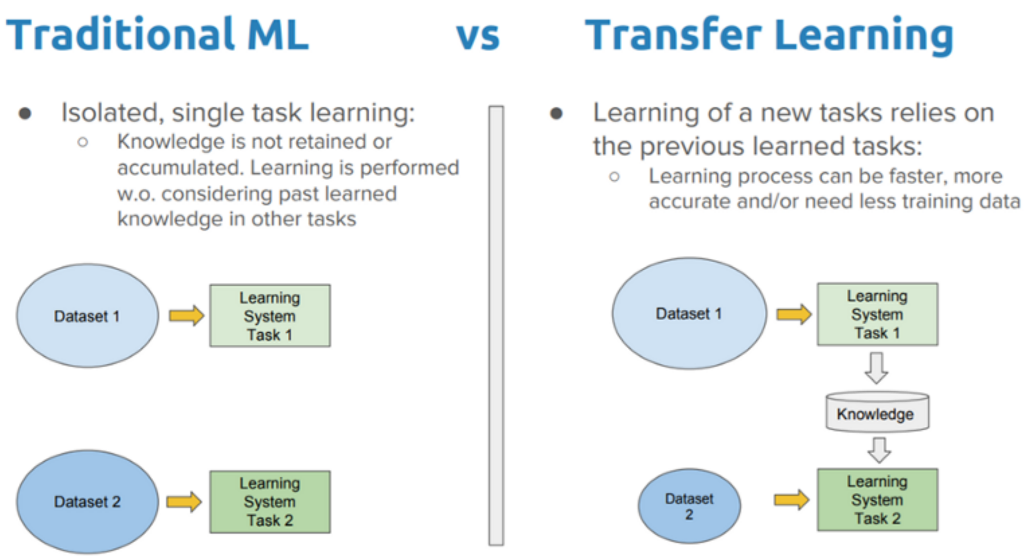
BERT (Bidirectional Encoder Representations from Transformers) adalah model yang memanfaatkan Transformer, yang merupakan sebuah *attention-mechanism* yang mempelajari hubungan kontekstual antara kata (atau sub-kata) dalam sebuah teks. Transformer di NLP adalah arsitektur yang bertujuan untuk menyelesaikan pekerjaan *sequence-to-sequence* sambil menangani dependensi jarak jauh dengan mudah. Dalam bentuk vanilla, Transformer menyertakan dua mekanisme terpisah, yaitu encoder yang membaca input teks dan decoder yang menghasilkan prediksi untuk pekerjaan tersebut. Karena tujuan BERT adalah menghasilkan model bahasa, hanya mekanisme encoder yang diperlukan.



Gambar 2.2: Transformer

2.1.4 Transfer Learning & Pre-trained Model

Transfer Learning (TL), yaitu masalah penelitian dalam machine learning yang berfokus pada menyimpan pengetahuan yang diperoleh sambil memecahkan satu masalah dan mengaplikasikannya ke masalah yang berbeda tetapi tetap berkaitan. Sedangkan Pre-trained Model adalah model yang dibuat oleh orang lain untuk memecahkan suatu masalah. Dibandingkan membuat model dari awal, pre-trained model dapat digunakan sebagai titik awal untuk menyelesaikan masalah serupa. Karena merupakan implementasi dari CNN, EfficientNet termasuk pre-trained model. BERT juga termasuk ke dalam pre-trained model. Untuk itu, kami menggunakan Efficientnet dan BERT untuk Transfer Learning dalam makalah kali ini.


Traditional ML
vs
Transfer Learning
Isolated, single task learning:
Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks
Learning of a new tasks relies on the previous learned tasks:
Learning process can be faster, more accurate and/or need less training data
Dataset 1
Learning System Task 1
Dataset 2
Learning System Task 2
Dataset 1
Learning System Task 1
Knowledge
Dataset 2
Learning System Task 2

Gambar 2.3: Perbedaan *Traditional ML* dengan *Transfer Learning*

2.1.5 Multimodal Deep Learning

Multimodal deep learning memberi gagasan bahwa ketika sejumlah indera kita (penglihatan, pendengaran, kinestetik) semua berikut serta dalam pemrosesan informasi, maka kita dapat mengerti dan mengingat lebih banyak. Hal ini juga dapat diaplikasikan ke dalam *machine learning* dan *deep learning*, model gabungan *machine learning* dapat memproses lebih dari satu macam tipe *input* dengan cara menggabungkan model tunggal yang hanya memproses satu tipe input. Dengan menggabungkan model - model ini, model gabungan dapat menggabungkan informasi dari berbagai macam sumber dan memiliki performa yang lebih baik.

2.1.6 Pseudo Labeling

Pseudo labeling melatih jaringan dengan data berlabel dan data tidak berlabel secara bersamaan untuk setiap kelompok. *Pseudo labeling* merupakan metode yang digunakan ketika dataset yang tersedia memiliki data tidak berlabel dalam jumlah banyak, sedangkan data yang berlabel hanya berjumlah sedikit.

Pseudo labeling termasuk metode *semi-supervised learning*. Dengan *semi-supervised learning* menggunakan *pseudo labeling*, model *machine learning* dapat memiliki performa lebih baik karena model tersebut akan mendapatkan lebih banyak data untuk diproses. Selain itu, *semi-supervised learning* juga memudahkan *labeler* dan *data scientist* untuk tidak melakukan prose *labeling* manual yang bisa sangat memakan waktu dan tenaga mengingat format data sekarang yang cendurung berupa *Big Data*.

BAB 3
DATASET

3.1 Pengumpulan Data

Untuk pengumpulan data, kami melakukan *scraping* di *platform* Twitter dengan kriteria rentang waktu 6 bulan dari tanggal 6 Oktober 2019, maksimal 500 *tweet* per harinya, hanya yang mengandung kata banjir, dan hanya yang mengandung kedua teks dan gambar.

3.2 Label Data

Ada 2 label untuk *dataset* ini:

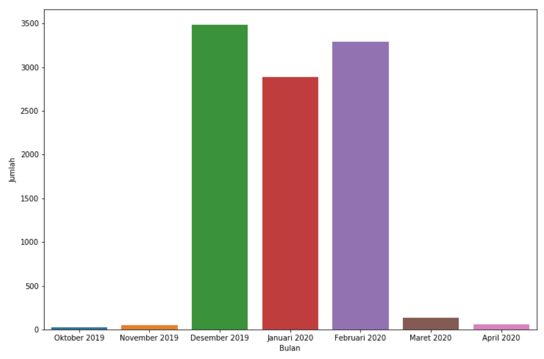
1. **NO_INFO** - *tweet* yang tidak mengandung *actionable intelligence* atau informasi yang berguna bagi *aid organizations* atau kepada upaya pengdataan dampak bnajir.
2. **INFO** - *tweet* informatif yang biasanya mengandung gambar banjir, keadaan banjir dan informasi lokasi banjir.

3.3 Analisis Dataset

Pada bagian ini kami melakukan analisis dasar terhadap 3 dataset. Pertama dataset yang belum dilakukan *labeling* yang selanjutnya akan disebut *unlabeled*, kedua data *train*, dan ketiga data *test*. Dataset *unlabeled* akan digunakan untuk melakukan *semi-supervised learning* menggunakan *pseudo-labeling*. Berikut ini adalah tabel statistik untung masing-masing dataset:

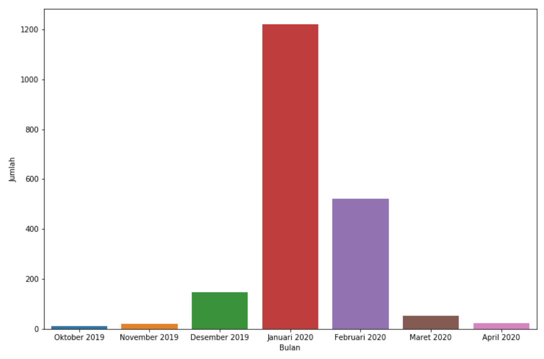
Dataset	Panjang *tweet* min	Panjang *tweet* maks	Jumlah *tweet*
Unlabeled	30	811	9936
Train	30	328	2000
Test	32	323	1000

Untuk distribusi jumlah *tweet* terhadap waktu (dalam bulan) pada dataset *Unlabeled* digambarkan melalui grafik di bawah ini:



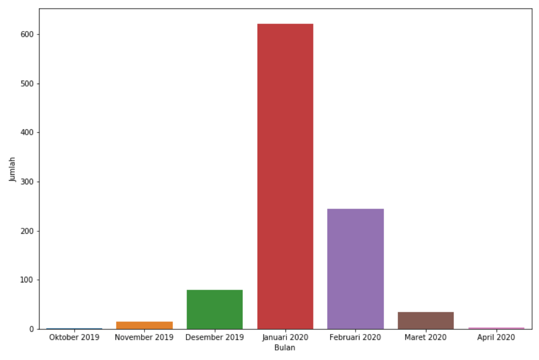
Gambar 3.1: Grafik distribusi pada dataset *Unlabeled*

Untuk distribusi yang sama pada dataset *Train* digambarkan melalui grafik di bawah ini:



Gambar 3.2: Grafik distribusi pada dataset *Train*

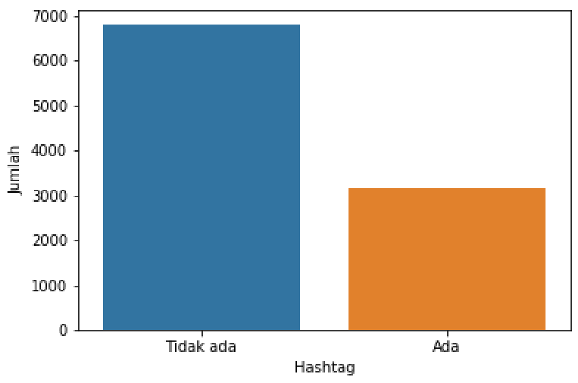
Sedangkan pada data *Test* digambarkan melalui grafik di bawah ini:



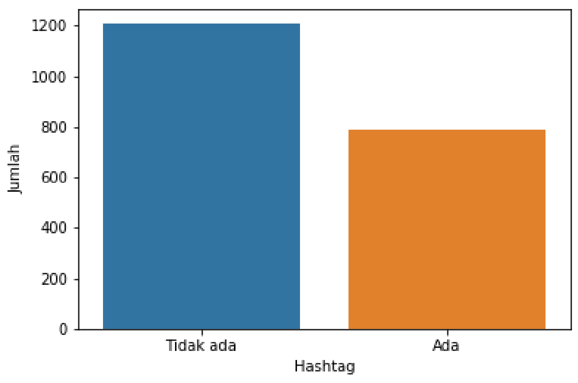
Gambar 3.3: Grafik distribusi pada dataset *Test*

Dapat dilihat pada dataset *Train* dan *Test* memiliki *tweet* paling banyak di Januari 2020, sedangkan pada data *Unlabeled*, *tweet* terbanyak terdapat di Desember 2019, disusul Februari 2020, dan juga Januari 2020. Hal ini sesuai dengan keadaan banjir Jakarta yang terjadi pada bulan Januari dan Februari 2020.

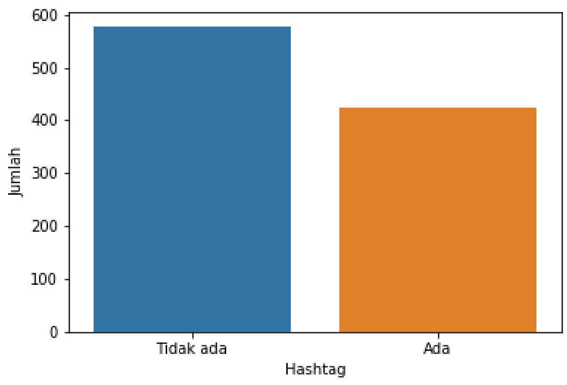
Kemudian kami melihat bahwa *tweet* yang tidak menggunakan hashtag lebih banyak dibandingkan yang tidak menggunakan hashtag, berikut ketiga grafiknya:



Gambar 3.4: Grafik perbandingan pada dataset *Unlabeled*



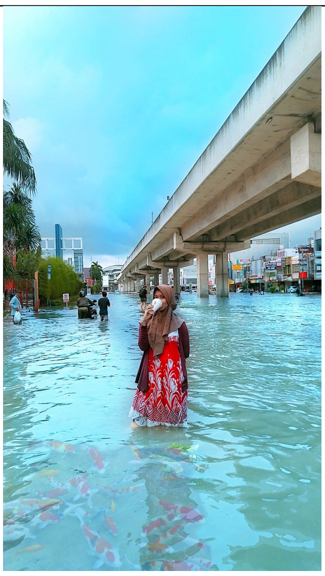
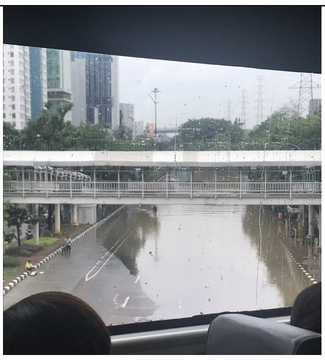
Gambar 3.5: Grafik perbandingan pada dataset *Train*

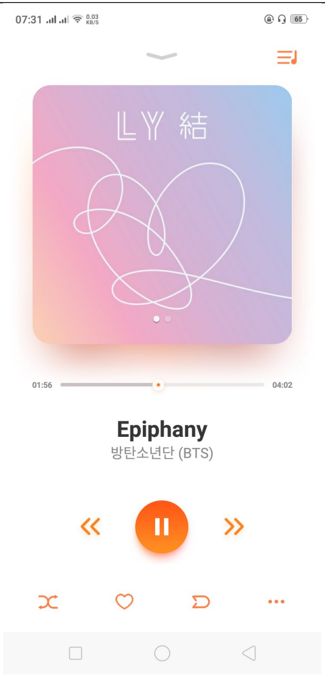
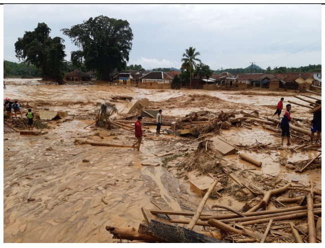
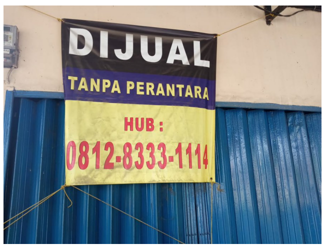
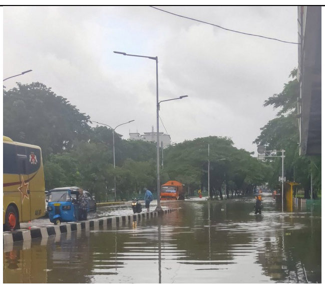


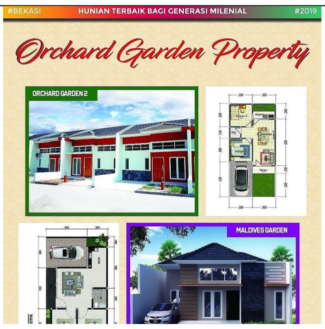
Gambar 3.6: Grafik perbandingan pada dataset *Test*

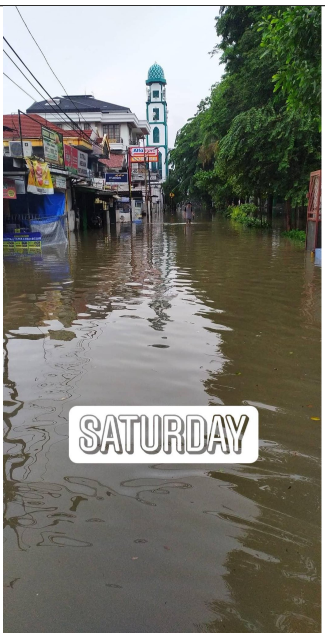
3.4 Sampel Dataset

Berikut ini merupakan beberapa contoh dari dataset kami:

Text	Image	Label
Coba aja air banjir itu bisa di filter dulu pake lightroom, pasti jernih kayak gini		NO_INFO
kebon nanas - cawang dpn apt tamansari sebelum underpass #Banjir		INFO

Banjir gua denger lagu ini pas hati gua lagi gini	 07:31 01:56 04:02 Epiphany 방탄소년단 (BTS)	NO_INFO
Sedih bgt kampung deket rumah kena musibah banjir, rumah sekolah pada hayut kebawa:(		INFO
Postingan dijual seger bangunan kios ada 3 unit beserta bangunan rumah dibelakang kios surat SHM,air jetpam,ada tempat parkir,tidak kena banjir,posisi sertategis dipinggir jalan raya Pertamina,Babelan Bekasi Utara,hubungi hp ,081283331114		NO_INFO
08:01 #Banjir 50-60 cm di Jl. Letjen Suprapto #Jakarta Pusat (arah ke Senen), sementara lalin dialihkan. @yanuariant		INFO

Cluster Orchard Garden 2 tambun Lokasi strategis Rumah ready stock Harga 500 jtan Bebas banjir	 #BEKASI HUNIAN TERBAIK BAGI GENERASI MILENIAL #2019 Orchard Garden Property ORCHARD GARDEN 2 MALDIVES GARDEN	NO_INFO
Banjir sudah hampir selutut, sudah masuk ke bbrp rmh.. bbrp motor sudah mogok.. anak2 byk bermain air bahkan berenang di jalan.. berbahaya jika pas mobil lewat tidak kelihatan krn anaknya berenang.. @e100s		INFO
Pasca Banjir Bandang #Lebak hari ini, (19/1/20) PMI Kab. Lebak dan PMI Cilegon di sela-sela kegiatan Psychosocial Support Program kepada anak-anak di posko pengungsian kp. Seupang ds. Pajagan kec sajira memberikan makanan buah semangka dan buah melon. @Pak_JK @jokowi @DPR_RI		NO_INFO

Akhirnya pertama kali bisa merasakan banjir di Kota Jakarta tercinta #banjir		INFO

BAB 4
PEMBAHASAN

4.1 Desain dan Implementasi

4.1.1 *Software* dan *Environment*

Penambangan dan analisis data dilakukan menggunakan bahasa pemrograman *Python* beserta *library* penambangan data yang tersedia oleh komunitas *Python*. Berikut adalah *library* yang kami gunakan:

- fast.ai 1.0.61 - *high-level library* untuk *deep learning* yang dibuat berdasarkan *library* Pytorch
- numpy
- pandas
- sklearn
- transformers 3.0.2 - *library* penyedia *pretrained transformers model* sepert BERT dan RoBERTa
- efficientnet_pytorch
- ohmeow-blurr
- tweet-preprocessor 0.6.0
- catboost
- nasty

Selain itu, proses *scraping*, penambangan dan analisis data dilakukan pada platform *Kaggle Kernels* dengan GPU Nvidia Tesla P100.

4.1.2 *High-level Overview*

Eksperimen yang kami lakukan adalah untuk membuat model *binary classification* menggunakan *Deep Learning* yang memanfaatkan gambar dan juga teks *tweet*.

Untuk *learning rate*, kami menggunakan *learning rate* https://sgugger.github.io/how-do-you-find-a-good-learning-rate.html yang disediakan oleh fast.ai dan smeua model di-*train* selama 4-6 epochs sampai *convergence*. Untuk *optimizer*, kami menggunakan *optimizer* Adam dengan parameter `Adam(params, lr, mom=0.9, sqr_mom=0.99, eps=1e-05, wd=0.01, decouple_wd=True)`.

4.1.3 Metrik Evaluasi

Karena ada *imbalance* cukup tinggi di *dataset* yang kami gunakan, kami menggunakan 2 metrik: akurasi dan F1. F1 memiliki formula:

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

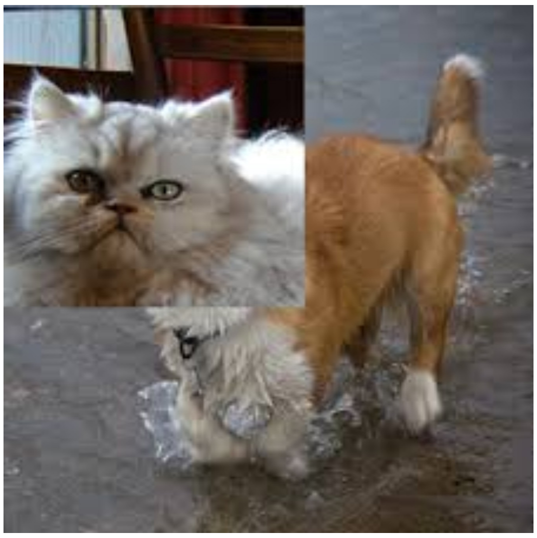
Kami memilih metrik F1 karena dengan metrik tersebut kami bisa lebih jelas melihat model yang tidak bisa mengklasfikasikan *class* **INFO** (yang lebih sedikit) dengan baik.

4.1.4 *Image Modality*

Untuk mengklasifikasikan *tweet* dengan menggunakan gambar, kami mencoba mengevaluasi 2 model *pretrained* yang sering digunakan. Berikut adalah 2 model yang kami gunakan dan *preprocessing* khususnya:

1. ResNet50
    - Semua gambar di-*resize* menjadi 224x224 pixel
2. EfficientNet-B4
    - Semua gambar di-*resize* menjadi 380x380 pixel

Selain itu, semua gambar di-*normalize* ke rata-rata dari gambar ImageNet karena model yang kami gunakan adalah model *pretrained* yang sudah *trained* dengan dataset ImageNet. Augmentasi gambar yang kami lakukan adalah augmentasi standar dari fast.ai yang melakukan augmentasi *flip, rotate, zoom, warp, lighting* dan juga CutMix yang kami tambahkan untuk mengurangi *overfitting*. Kami menggunakan `batch_size` sebesar 32.



Gambar 4.1: Contoh Augmentasi Cutmix

4.1.5 *Text Modality*

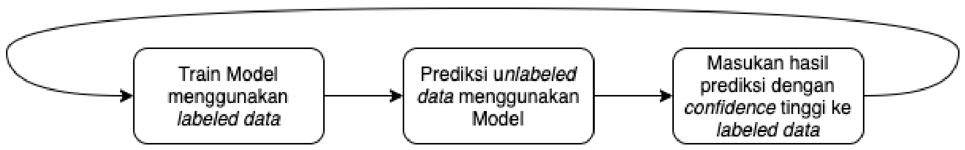
Model yang kami gunakan untuk melakukan klasifikasi teks adalah model BERT(*Bidirectional Encoder Representations from Transformers*). Kami menggunakan model dari *library* `transformers` huggingface dan *pretrained weights* oleh https://huggingface.co/cahya/bert-base-indonesian-522M yang dihasilkan dari 522 juta kata Wikipedia Indonesia.

Untuk preprocessing *tweet*, kami menggunakan *library* `tweet-preprocessor` untuk menghilangkan emoji, mention, dan hashtag(selain #banjir). Selain itu, kami tidak melakukan *preprocessing* tambahan karena sifat model berbasis *transformer* sekarang yang bisa mengakomodasi tulisan tanpa *preprocessing* terlalu banyak.

Kami menggunakan `batch_size` sebesar 4.

4.1.6 *Pseudo-Labeling*

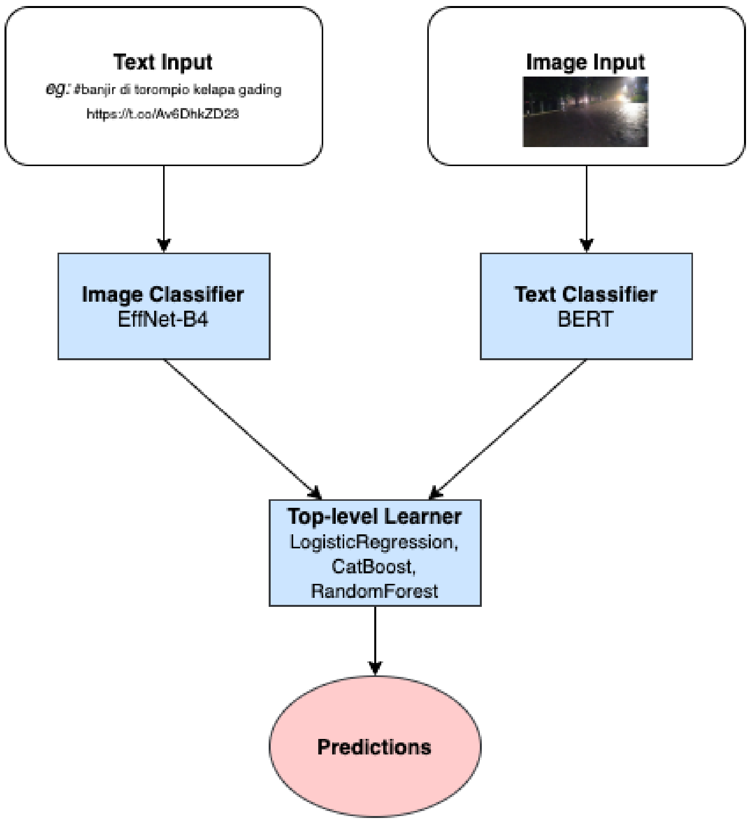
Bentuk *semi-supervised learning* yang kami gunakan untuk eksperimen ini adalah *pseudo labeling*. Implementasi *pseudo-labeling* yang kami gunakan adalah implementasi sederhana dengan cara mengambil prediksi dengan tingkat *confidence* tinggi dari model awal di *dataset* yang tidak di label dan menggunakan prediksi ini untuk kembali meng-*train* model baru(bersama juga dengan data yang sudah di label sebelumnya). Proses ini hanya kami lakukan sekali untuk setiap model.


Train Model menggunakan *labeled data*
Prediksi u*nlabeled data* menggunakan Model
Masukan hasil prediksi dengan *confidence* tinggi ke *labeled data*

Gambar 4.2: Pseudo-labeling

4.1.7 *Modality Fusion*

Untuk menggabungkan input dari *classifier* dan *classifier* teks, kami menggunakan *classifier* tambahan yang menggunakan keluaran *class probabilities* dari model awal untuk mengklasifikasikan *tweet*. Kami mencoba 3 *classifier* tambahan yaitu CatBoost(*Gradient Boosted Decision Trees*), LogisticRegression(sklearn), dan RandomForest(sklearn).


Text Input
eg: #banjir di torompio kelapa gading https://t.co/Av6DhkZD23
Image Input
Image Classifier
EffNet-B4
Text Classifier
BERT
Top-level Learner
LogisticRegression, CatBoost, RandomForest
Predictions

Gambar 4.3: Decision-level Multimodal Fusion

4.2 Analisis

4.2.1 *Baseline* untuk *Image Classification*

Model berbasis EfficientNet memiliki performa yang lebih baik dibandingkan model ResNet. Hal ini mungkin dipengaruhi juga oleh resolusi model EfficientNet yang lebih tinggi.

4.2.1.1 ResNet50

	Precision	Recall	F1	Support
INFO	0.6420	0.5136	0.5707	220
NO_INFO	0.8701	0.9192	0.8940	780
Accuracy			0.8300	1000
Macro Avg.	0.7561	0.7164	0.7324	1000
Weighted Avg.	0.8200	0.8300	0.8229	1000

4.2.1.2 EfficientNet-B4

Model EfficientNet memiliki akurasi yang 0.8% lebih tinggi dibandingkan dengan model ResNet. Nilai F1 juga 5.1% lebih tinggi di model EfficientNet dibandingkan dengan model ResNet.

	Precision	Recall	F1	Support
INFO	0.6183	0.6773	0.6464	220
NO_INFO	0.9065	0.8821	0.8941	780
Accuracy			0.8370	1000
Macro Avg.	0.7624	0.7797	0.7703	1000
Weighted Avg.	0.8431	0.8370	0.8396	1000

4.2.2 *Baseline* untuk *Text Classification*

Model BERT ini memiliki akurasi yang hanya 1% lebih rendah daripada model EffNet-B4 namun memiliki nilai F1 yang sekitar 7% lebih rendah.

	Precision	Recall	F1	Support
INFO	0.6561	0.4682	0.5464	220
NO_INFO	0.8612	0.9308	0.8946	780
Accuracy			0.8290	1000
Macro Avg.	0.7586	0.6995	0.7205	1000
Weighted Avg.	0.8161	0.8290	0.8180	1000

4.2.3 Dampak *Pseudo-Labeling* dan *Semi-Supervised Learning*

Berikut adalah tabel perbandingan hasil sebelum dan sesudah *pseudo-labeling*: Ha-

Model	**Metrik**	
	Accuracy	F1
ResNet50 - Baseline	0.8300	0.7324
EffNetB4 - Baseline	0.8370	0.7703
BERT - Baseline	0.8290	0.7205
ResNet50 - Pseudo Labeling	0.8140	0.7325
EffNetB4 - Pseudo Labeling	**0.8510**	**0.7789**
BERT - Pseudo Labeling	0.8300	0.7227

sil terbaik dari *semi-supervised learning* menggunakan teknik *pseduo-labeling* dicapai oleh model EfficientNet-B4 (modalitas gambar). Model BERT tidak menunjukkan pengingkatan performa yang signifikan dan model ResNET menujukkan penurunan skor akurasi tetapi juga peningkatan skor F1.

4.2.4 *Error Analysis*

Untuk mengetahui seberapa berguna penggabungan 2 modalitas ini, kami melakukan *error analysis* sebagai berikut:

4.2.4.1 Tidak *Pseudo Labeled*

Kondisi	**Jumlah**	**Persentase**
Teks Benar & Gambar Benar	726	72.6%
Teks Benar & Gambar Salah	103	10.3%
Teks Salah & Gambar Benar	111	11.1%
Teks Salah & Gambar Salah	60	6%

Kondisi dimana kedua model menghasilkan prediksi yang salah hanya merupakan 6% dari total *dataset test*. Ada ruang *improvement* sebesar 21.4% yang bisa didapatkan dengan cara menggabungkan kedua modalitas ini.

4.2.4.2 *Pseudo Labeled*

Ada 19.7% ruang untuk *improvement* jika menggunakan model yang sudah melalui proses *pseudo-labeling*, namun persentase akurasi masih lebih tinggi di model ini.

Kondisi	**Jumlah**	**Persentase**
Teks Benar & Gambar Benar	742	74.2%
Teks Benar & Gambar Salah	88	8.8%
Teks Salah & Gambar Benar	109	10.9%
Teks Salah & Gambar Salah	61	6.1%

4.2.5 *Modality Fusion*

Pada kasus terbaik, penggabungan kedua modalitas ini menghasilkan kenaikan akurasi sebesar 0.016(1.8%) dan kenaikan F1 sebesar 0.024(3.3%) dibandingkan dengan model *pseduo-labeled* terbaik.

Penggabungan model *pseudo-labeled* tidak menghasilkan hasil sebaik dengan model bisa meskipun memiliki perform awal yang lebih baik. Hal ini mungkin disebabkan oleh ruang *improvement* yang lebih sedikit (19.7% vs 21.4%).

Top-level Classifier	**Tidak Pseudo-Labeled**		**Pseudo-Labeled**	
	Accuracy	F1	Accuracy	F1
CatBoost (GBDT)	0.8600	0.7947	0.8530	0.7896
Logistic Regression	**0.8670**	**0.8053**	**0.8620**	**0.8022**
Random Forest	0.8590	0.7929	0.8460	0.7813

BAB 5
PENUTUP

5.1 Kesimpulan

Classifier Deep Learning yang menggunakan kedua modalitas gambar dan teks memiliki performa lebih baik jika dibandingkan dengan model *single modality*, sekitar 1-3% lebih baik dibandingkan dengan model *pseudo-labeled* dan sekitar 3.5-4.5% lebih baik dibandingkan dengan model non *pseudo-labeled*.

Hal ini menunjukkan bahwa klasifkasi *tweet* memang merupakan *task* yang sesuai untuk dijadikan *task* klasifikasi *multi-modal* mengingat bahwa banyak *tweet* yang memiliki konten teks dan gambar yang saling terhubung dan sulit dimengerti jika dipisahkan.

Model berbeda modalitas yang sudah melalui proses *pseudo labeling* jika digabungkan memiliki performa yang lebih buruk dengan model ekivalen tanpa *pseudo labeling*. Ada juga penurunan *room for improvement* (19.7% vs 21.4%) dibandingkan dengan model tnapa *pseudo labeling*. Namun, *pseudo-labeling* berhasil meningkatkan performa model tunggal yang tidak digabungkan.

Interaksi antar *semi-supervised learning* dan *multi-modal learning* sepertinya tidak memiliki efek *stacking*. Interaksi antar kedua metode ini menghasilkan model yang memiliki performa lebih buruk dibandingkan dengan hanya *multi-modal learning* saja namun masih lebih baik dibandingkan dengan model yang hanya *semi-supervised learning* saja.

5.2 Saran

- Melakukan eksperimen lebih lanjut untuk mendalami efek penggabungan *semi-supervised learning* dan *multi-modal learning*
- Melakukan eksperimen dengan teknik *semi-supervised learning* lain
- Mencoba menggunakan *feature-level fusion* menggantikan *decision-level fusion* pada *multi-modal learning*

BAB 6
DOKUMENTASI

Dataset dan source code:

https://drive.google.com/drive/folders/15DyhpctJum3DK9ShCebYtqpbs3Xb-Oe4?usp=sharing

Setup

In [1]:

```
!pip install torch==1.6.0+cu101 torchvision==0.7.0+cu101 -f https://download.pytorch.org/w
hl/torch_stable.html
!pip install efficientnet_pytorch
!pip install --upgrade kornia
!pip install allennlp==1.1.0.rc4
!pip install --upgrade fastai
```

```
Looking in links: https://download.pytorch.org/whl/torch_stable.html
Collecting torch==1.6.0+cu101
  Downloading https://download.pytorch.org/whl/cu101/torch-1.6.0%2Bcu101-cp37-cp37m-linux_x
86_64.whl (708.0 MB)
     |████████████████████████████████| 708.0 MB 7.2 kB/s  eta 0:00:01     |████
| 94.3 MB 69.3 MB/s eta 0:00:09     |████                             | 101.1 MB 69.3 MB/s
eta 0:00:09     |████                             | 104.0 MB 28.5 MB/s eta 0:00:22��█
| 106.9 MB 28.5 MB/s eta 0:00:22MB/s eta 0:00:21     |█████                            | 11
3.7 MB 28.5 MB/s eta 0:00:21     |█████                            | 121.2 MB 28.5 MB/s eta
0:00:21��███                           | 211.0 MB 53.8 MB/s eta 0:00:10��███████████████
| 534.1 MB 46.5 MB/s eta 0:00:04     |████████████████████████        | 537.7 MB 6.1 MB/s e
ta 0:00:29     |████████████████████████        | 539.5 MB 6.1 MB/s eta 0:00:28     |████
██████████████████████     | 618.1 MB 64.1 MB/s eta 0:00:02     |███████████████████████████
███  | 661.1 MB 51.2 MB/s eta 0:00:01     |██████████████████████████████  | 662.3 MB 51.2
MB/s eta 0:00:01�█  | 675.6 MB 52.0 MB/s eta 0:00:01�█  | 678.1 MB 52.0 MB/s eta 0:00:01
|██████████████████████████████  | 681.5 MB 52.0 MB/s eta 0:00:01��███████████| 698.3 MB 5
2.0 MB/s eta 0:00:01
Collecting torchvision==0.7.0+cu101
  Downloading https://download.pytorch.org/whl/cu101/torchvision-0.7.0%2Bcu101-cp37-cp37m-l
inux_x86_64.whl (5.9 MB)
     |████████████████████████████████| 5.9 MB 50.1 MB/s eta 0:00:01
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages (from torch=
=1.6.0+cu101) (1.18.5)
Requirement already satisfied: future in /opt/conda/lib/python3.7/site-packages (from torch
==1.6.0+cu101) (0.18.2)
Requirement already satisfied: pillow>=4.1.1 in /opt/conda/lib/python3.7/site-packages (fro
m torchvision==0.7.0+cu101) (7.2.0)
Installing collected packages: torch, torchvision
  Attempting uninstall: torch
    Found existing installation: torch 1.5.1
    Uninstalling torch-1.5.1:
      Successfully uninstalled torch-1.5.1
  Attempting uninstall: torchvision
    Found existing installation: torchvision 0.6.0a0+35d732a
    Uninstalling torchvision-0.6.0a0+35d732a:
      Successfully uninstalled torchvision-0.6.0a0+35d732a
ERROR: After October 2020 you may experience errors when installing or updating packages. T
his is because pip will change the way that it resolves dependency conflicts.

We recommend you use --use-feature=2020-resolver to test your packages with the new resolve
r before it becomes the default.

kornia 0.3.2 requires torch<1.6.0,>=1.5.0, but you'll have torch 1.6.0+cu101 which is incom
patible.
allennlp 1.0.0 requires torch<1.6.0,>=1.5.0, but you'll have torch 1.6.0+cu101 which is inc
ompatible.
Successfully installed torch-1.6.0+cu101 torchvision-0.7.0+cu101
WARNING: You are using pip version 20.2.2; however, version 20.2.3 is available.
You should consider upgrading via the '/opt/conda/bin/python3.7 -m pip install --upgrade pi
p' command.
Collecting efficientnet_pytorch
  Downloading efficientnet_pytorch-0.7.0.tar.gz (20 kB)
Requirement already satisfied: torch in /opt/conda/lib/python3.7/site-packages (from effici
entnet_pytorch) (1.6.0+cu101)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages (from torch-
>efficientnet_pytorch) (1.18.5)
Requirement already satisfied: future in /opt/conda/lib/python3.7/site-packages (from torch
->efficientnet_pytorch) (0.18.2)
```

```
Building wheels for collected packages: efficientnet-pytorch
  Building wheel for efficientnet-pytorch (setup.py) ... done
  Created wheel for efficientnet-pytorch: filename=efficientnet_pytorch-0.7.0-py3-none-any.
whl size=16035 sha256=cd1ab612a67a7fd617d8eac175183449431cd0c42c423cc11824d654ceaeb826
  Stored in directory: /root/.cache/pip/wheels/b7/cc/0d/41d384b0071c6f46e542aded5f8571700ac
e4f1eb3f1591c29
Successfully built efficientnet-pytorch
Installing collected packages: efficientnet-pytorch
Successfully installed efficientnet-pytorch-0.7.0
WARNING: You are using pip version 20.2.2; however, version 20.2.3 is available.
You should consider upgrading via the '/opt/conda/bin/python3.7 -m pip install --upgrade pi
p' command.
Collecting kornia
  Downloading kornia-0.4.0-py2.py3-none-any.whl (195 kB)
     |████████████████████████████████| 195 kB 573 kB/s eta 0:00:01
Requirement already satisfied, skipping upgrade: numpy in /opt/conda/lib/python3.7/site-pac
kages (from kornia) (1.18.5)
Requirement already satisfied, skipping upgrade: torch<1.7.0,>=1.6.0 in /opt/conda/lib/pyth
on3.7/site-packages (from kornia) (1.6.0+cu101)
Requirement already satisfied, skipping upgrade: future in /opt/conda/lib/python3.7/site-pa
ckages (from torch<1.7.0,>=1.6.0->kornia) (0.18.2)
Installing collected packages: kornia
  Attempting uninstall: kornia
    Found existing installation: kornia 0.3.2
    Uninstalling kornia-0.3.2:
      Successfully uninstalled kornia-0.3.2
Successfully installed kornia-0.4.0
WARNING: You are using pip version 20.2.2; however, version 20.2.3 is available.
You should consider upgrading via the '/opt/conda/bin/python3.7 -m pip install --upgrade pi
p' command.
Collecting allennlp==1.1.0.rc4
  Downloading allennlp-1.1.0rc4-py3-none-any.whl (484 kB)
     |████████████████████████████████| 484 kB 574 kB/s eta 0:00:01
Requirement already satisfied: scipy in /opt/conda/lib/python3.7/site-packages (from allenn
lp==1.1.0.rc4) (1.4.1)
Requirement already satisfied: nltk in /opt/conda/lib/python3.7/site-packages (from allennl
p==1.1.0.rc4) (3.2.4)
Requirement already satisfied: tensorboardX>=1.2 in /opt/conda/lib/python3.7/site-packages
 (from allennlp==1.1.0.rc4) (2.1)
Requirement already satisfied: pytest in /opt/conda/lib/python3.7/site-packages (from allen
nlp==1.1.0.rc4) (5.4.1)
Requirement already satisfied: torch<1.7.0,>=1.6.0 in /opt/conda/lib/python3.7/site-package
s (from allennlp==1.1.0.rc4) (1.6.0+cu101)
Requirement already satisfied: requests>=2.18 in /opt/conda/lib/python3.7/site-packages (fr
om allennlp==1.1.0.rc4) (2.23.0)
Requirement already satisfied: scikit-learn in /opt/conda/lib/python3.7/site-packages (from
allennlp==1.1.0.rc4) (0.23.2)
Requirement already satisfied: h5py in /opt/conda/lib/python3.7/site-packages (from allennl
p==1.1.0.rc4) (2.10.0)
Requirement already satisfied: filelock<3.1,>=3.0 in /opt/conda/lib/python3.7/site-packages
 (from allennlp==1.1.0.rc4) (3.0.10)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages (from allenn
lp==1.1.0.rc4) (1.18.5)
Collecting transformers<3.1,>=3.0
  Downloading transformers-3.0.2-py3-none-any.whl (769 kB)
     |████████████████████████████████| 769 kB 3.8 MB/s eta 0:00:01
Requirement already satisfied: jsonpickle in /opt/conda/lib/python3.7/site-packages (from a
llennlp==1.1.0.rc4) (1.4.1)
Requirement already satisfied: jsonnet>=0.10.0; sys_platform != "win32" in /opt/conda/lib/p
ython3.7/site-packages (from allennlp==1.1.0.rc4) (0.16.0)
Requirement already satisfied: spacy<2.4,>=2.1.0 in /opt/conda/lib/python3.7/site-packages
 (from allennlp==1.1.0.rc4) (2.2.4)
Requirement already satisfied: tqdm>=4.19 in /opt/conda/lib/python3.7/site-packages (from a
llennlp==1.1.0.rc4) (4.45.0)
Collecting overrides==3.1.0
  Downloading overrides-3.1.0.tar.gz (11 kB)
Requirement already satisfied: boto3<2.0,>=1.14 in /opt/conda/lib/python3.7/site-packages (
from allennlp==1.1.0.rc4) (1.14.48)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from nltk->al
lennlp==1.1.0.rc4) (1.14.0)
Requirement already satisfied: protobuf>=3.8.0 in /opt/conda/lib/python3.7/site-packages (f
rom tensorboardX>=1.2->allennlp==1.1.0.rc4) (3.13.0)
Requirement already satisfied: py>=1.5.0 in /opt/conda/lib/python3.7/site-packages (from py
test->allennlp==1.1.0.rc4) (1.8.1)
Requirement already satisfied: packaging in /opt/conda/lib/python3.7/site-packages (from py
test->allennlp==1.1.0.rc4) (20.1)
Requirement already satisfied: attrs>=17.4.0 in /opt/conda/lib/python3.7/site-packages (fro
m pytest->allennlp==1.1.0.rc4) (19.3.0)
Requirement already satisfied: more-itertools>=4.0.0 in /opt/conda/lib/python3.7/site-packa
ges (from pytest->allennlp==1.1.0.rc4) (8.2.0)
Requirement already satisfied: pluggy<1.0,>=0.12 in /opt/conda/lib/python3.7/site-packages
(from pytest->allennlp==1.1.0.rc4) (0.13.0)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.7/site-packages (from pyte
st->allennlp==1.1.0.rc4) (0.1.9)
Requirement already satisfied: importlib-metadata>=0.12 in /opt/conda/lib/python3.7/site-pa
ckages (from pytest->allennlp==1.1.0.rc4) (1.6.0)
Requirement already satisfied: future in /opt/conda/lib/python3.7/site-packages (from torch
<1.7.0,>=1.6.0->allennlp==1.1.0.rc4) (0.18.2)
Requirement already satisfied: chardet<4,>=3.0.2 in /opt/conda/lib/python3.7/site-packages
(from requests>=2.18->allennlp==1.1.0.rc4) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/lib/python3.7/site-packages (from
requests>=2.18->allennlp==1.1.0.rc4) (2.9)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.7/site-packages
(from requests>=2.18->allennlp==1.1.0.rc4) (2020.6.20)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /opt/conda/lib/py
thon3.7/site-packages (from requests>=2.18->allennlp==1.1.0.rc4) (1.24.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in /opt/conda/lib/python3.7/site-packag
es (from scikit-learn->allennlp==1.1.0.rc4) (2.1.0)
Requirement already satisfied: joblib>=0.11 in /opt/conda/lib/python3.7/site-packages (from
scikit-learn->allennlp==1.1.0.rc4) (0.14.1)
Requirement already satisfied: regex!=2019.12.17 in /opt/conda/lib/python3.7/site-packages
(from transformers<3.1,>=3.0->allennlp==1.1.0.rc4) (2020.4.4)
Requirement already satisfied: sacremoses in /opt/conda/lib/python3.7/site-packages (from t
ransformers<3.1,>=3.0->allennlp==1.1.0.rc4) (0.0.43)
Requirement already satisfied: sentencepiece!=0.1.92 in /opt/conda/lib/python3.7/site-packa
ges (from transformers<3.1,>=3.0->allennlp==1.1.0.rc4) (0.1.91)
Collecting tokenizers==0.8.1.rc1
  Downloading tokenizers-0.8.1rc1-cp37-cp37m-manylinux1_x86_64.whl (3.0 MB)
     |████████████████████████████████| 3.0 MB 5.1 MB/s eta 0:00:01
Requirement already satisfied: blis<0.5.0,>=0.4.0 in /opt/conda/lib/python3.7/site-packages
(from spacy<2.4,>=2.1.0->allennlp==1.1.0.rc4) (0.4.1)
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in /opt/conda/lib/python3.7/site-package
s (from spacy<2.4,>=2.1.0->allennlp==1.1.0.rc4) (1.0.2)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /opt/conda/lib/python3.7/site-packa
ges (from spacy<2.4,>=2.1.0->allennlp==1.1.0.rc4) (3.0.2)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.7/site-packages (from s
pacy<2.4,>=2.1.0->allennlp==1.1.0.rc4) (46.1.3.post20200325)
Requirement already satisfied: plac<1.2.0,>=0.9.6 in /opt/conda/lib/python3.7/site-packages
(from spacy<2.4,>=2.1.0->allennlp==1.1.0.rc4) (1.1.3)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /opt/conda/lib/python3.7/site-package
s (from spacy<2.4,>=2.1.0->allennlp==1.1.0.rc4) (2.0.3)
Requirement already satisfied: thinc==7.4.0 in /opt/conda/lib/python3.7/site-packages (from
spacy<2.4,>=2.1.0->allennlp==1.1.0.rc4) (7.4.0)
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in /opt/conda/lib/python3.7/site-pac
kages (from spacy<2.4,>=2.1.0->allennlp==1.1.0.rc4) (1.0.0)
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in /opt/conda/lib/python3.7/site-packag
es (from spacy<2.4,>=2.1.0->allennlp==1.1.0.rc4) (0.7.1)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /opt/conda/lib/python3.7/site-p
ackages (from spacy<2.4,>=2.1.0->allennlp==1.1.0.rc4) (1.0.2)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/lib/python3.7/site-pack
ages (from boto3<2.0,>=1.14->allennlp==1.1.0.rc4) (0.10.0)
Requirement already satisfied: s3transfer<0.4.0,>=0.3.0 in /opt/conda/lib/python3.7/site-pa
ckages (from boto3<2.0,>=1.14->allennlp==1.1.0.rc4) (0.3.3)
Requirement already satisfied: botocore<1.18.0,>=1.17.48 in /opt/conda/lib/python3.7/site-p
ackages (from boto3<2.0,>=1.14->allennlp==1.1.0.rc4) (1.17.48)
Requirement already satisfied: pyparsing>=2.0.2 in /opt/conda/lib/python3.7/site-packages (
from packaging->pytest->allennlp==1.1.0.rc4) (2.4.7)
Requirement already satisfied: zipp>=0.5 in /opt/conda/lib/python3.7/site-packages (from im
portlib-metadata>=0.12->pytest->allennlp==1.1.0.rc4) (3.1.0)
Requirement already satisfied: click in /opt/conda/lib/python3.7/site-packages (from sacrem
oses->transformers<3.1,>=3.0->allennlp==1.1.0.rc4) (7.1.1)
Requirement already satisfied: docutils<0.16,>=0.10 in /opt/conda/lib/python3.7/site-packag
es (from botocore<1.18.0,>=1.17.48->boto3<2.0,>=1.14->allennlp==1.1.0.rc4) (0.15.2)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/lib/python3.7/site
-packages (from botocore<1.18.0,>=1.17.48->boto3<2.0,>=1.14->allennlp==1.1.0.rc4) (2.8.1)
Building wheels for collected packages: overrides
```

```
  Building wheel for overrides (setup.py) ... done
  Created wheel for overrides: filename=overrides-3.1.0-py3-none-any.whl size=10173 sha256=
e13e34ccc22cf9ea2fc61f340559ad555e5f338ae3233379635b7c33ed80819f
  Stored in directory: /root/.cache/pip/wheels/3a/0d/38/01a9bc6e20dcfaf0a6a7b552d03137558ba
1c38aea47644682
Successfully built overrides
Installing collected packages: tokenizers, transformers, overrides, allennlp
  Attempting uninstall: tokenizers
    Found existing installation: tokenizers 0.7.0
    Uninstalling tokenizers-0.7.0:
      Successfully uninstalled tokenizers-0.7.0
  Attempting uninstall: transformers
    Found existing installation: transformers 2.11.0
    Uninstalling transformers-2.11.0:
      Successfully uninstalled transformers-2.11.0
  Attempting uninstall: overrides
    Found existing installation: overrides 3.0.0
    Uninstalling overrides-3.0.0:
      Successfully uninstalled overrides-3.0.0
  Attempting uninstall: allennlp
    Found existing installation: allennlp 1.0.0
    Uninstalling allennlp-1.0.0:
      Successfully uninstalled allennlp-1.0.0
Successfully installed allennlp-1.1.0rc4 overrides-3.1.0 tokenizers-0.8.1rc1 transformers-3
.0.2
WARNING: You are using pip version 20.2.2; however, version 20.2.3 is available.
You should consider upgrading via the '/opt/conda/bin/python3.7 -m pip install --upgrade pi
p' command.
Collecting fastai
  Downloading fastai-2.0.10-py3-none-any.whl (354 kB)
     |████████████████████████████████| 354 kB 575 kB/s eta 0:00:01
Requirement already satisfied, skipping upgrade: packaging in /opt/conda/lib/python3.7/site
-packages (from fastai) (20.1)
Requirement already satisfied, skipping upgrade: fastprogress>=0.2.4 in /opt/conda/lib/pyth
on3.7/site-packages (from fastai) (1.0.0)
Requirement already satisfied, skipping upgrade: spacy in /opt/conda/lib/python3.7/site-pac
kages (from fastai) (2.2.4)
Requirement already satisfied, skipping upgrade: pillow in /opt/conda/lib/python3.7/site-pa
ckages (from fastai) (7.2.0)
Requirement already satisfied, skipping upgrade: torchvision>=0.7 in /opt/conda/lib/python3
.7/site-packages (from fastai) (0.7.0+cu101)
Requirement already satisfied, skipping upgrade: torch>=1.6.0 in /opt/conda/lib/python3.7/s
ite-packages (from fastai) (1.6.0+cu101)
Requirement already satisfied, skipping upgrade: matplotlib in /opt/conda/lib/python3.7/sit
e-packages (from fastai) (3.2.1)
Requirement already satisfied, skipping upgrade: pyyaml in /opt/conda/lib/python3.7/site-pa
ckages (from fastai) (5.3.1)
Requirement already satisfied, skipping upgrade: pandas in /opt/conda/lib/python3.7/site-pa
ckages (from fastai) (1.1.1)
Requirement already satisfied, skipping upgrade: scikit-learn in /opt/conda/lib/python3.7/s
ite-packages (from fastai) (0.23.2)
Collecting fastcore>=1.0.5
  Downloading fastcore-1.0.9-py3-none-any.whl (37 kB)
Requirement already satisfied, skipping upgrade: requests in /opt/conda/lib/python3.7/site-
packages (from fastai) (2.23.0)
Requirement already satisfied, skipping upgrade: pip in /opt/conda/lib/python3.7/site-packa
ges (from fastai) (20.2.2)
Requirement already satisfied, skipping upgrade: scipy in /opt/conda/lib/python3.7/site-pac
kages (from fastai) (1.4.1)
Requirement already satisfied, skipping upgrade: pyparsing>=2.0.2 in /opt/conda/lib/python3
.7/site-packages (from packaging->fastai) (2.4.7)
Requirement already satisfied, skipping upgrade: six in /opt/conda/lib/python3.7/site-packa
ges (from packaging->fastai) (1.14.0)
Requirement already satisfied, skipping upgrade: numpy in /opt/conda/lib/python3.7/site-pac
kages (from fastprogress>=0.2.4->fastai) (1.18.5)
Requirement already satisfied, skipping upgrade: setuptools in /opt/conda/lib/python3.7/sit
e-packages (from spacy->fastai) (46.1.3.post20200325)
Requirement already satisfied, skipping upgrade: tqdm<5.0.0,>=4.38.0 in /opt/conda/lib/pyth
on3.7/site-packages (from spacy->fastai) (4.45.0)
Requirement already satisfied, skipping upgrade: plac<1.2.0,>=0.9.6 in /opt/conda/lib/pytho
n3.7/site-packages (from spacy->fastai) (1.1.3)
Requirement already satisfied, skipping upgrade: preshed<3.1.0,>=3.0.2 in /opt/conda/lib/py
thon3.7/site-packages (from spacy->fastai) (3.0.2)
```

```
Requirement already satisfied, skipping upgrade: cymem<2.1.0,>=2.0.2 in /opt/conda/lib/pyth
on3.7/site-packages (from spacy->fastai) (2.0.3)
Requirement already satisfied, skipping upgrade: catalogue<1.1.0,>=0.0.7 in /opt/conda/lib/
python3.7/site-packages (from spacy->fastai) (1.0.0)
Requirement already satisfied, skipping upgrade: wasabi<1.1.0,>=0.4.0 in /opt/conda/lib/pyt
hon3.7/site-packages (from spacy->fastai) (0.7.1)
Requirement already satisfied, skipping upgrade: blis<0.5.0,>=0.4.0 in /opt/conda/lib/pytho
n3.7/site-packages (from spacy->fastai) (0.4.1)
Requirement already satisfied, skipping upgrade: thinc==7.4.0 in /opt/conda/lib/python3.7/s
ite-packages (from spacy->fastai) (7.4.0)
Requirement already satisfied, skipping upgrade: murmurhash<1.1.0,>=0.28.0 in /opt/conda/li
b/python3.7/site-packages (from spacy->fastai) (1.0.2)
Requirement already satisfied, skipping upgrade: srsly<1.1.0,>=1.0.2 in /opt/conda/lib/pyth
on3.7/site-packages (from spacy->fastai) (1.0.2)
Requirement already satisfied, skipping upgrade: future in /opt/conda/lib/python3.7/site-pa
ckages (from torch>=1.6.0->fastai) (0.18.2)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.1 in /opt/conda/lib/pyt
hon3.7/site-packages (from matplotlib->fastai) (2.8.1)
Requirement already satisfied, skipping upgrade: kiwisolver>=1.0.1 in /opt/conda/lib/python
3.7/site-packages (from matplotlib->fastai) (1.2.0)
Requirement already satisfied, skipping upgrade: cycler>=0.10 in /opt/conda/lib/python3.7/s
ite-packages (from matplotlib->fastai) (0.10.0)
Requirement already satisfied, skipping upgrade: pytz>=2017.2 in /opt/conda/lib/python3.7/s
ite-packages (from pandas->fastai) (2019.3)
Requirement already satisfied, skipping upgrade: joblib>=0.11 in /opt/conda/lib/python3.7/s
ite-packages (from scikit-learn->fastai) (0.14.1)
Requirement already satisfied, skipping upgrade: threadpoolctl>=2.0.0 in /opt/conda/lib/pyt
hon3.7/site-packages (from scikit-learn->fastai) (2.1.0)
Requirement already satisfied, skipping upgrade: idna<3,>=2.5 in /opt/conda/lib/python3.7/s
ite-packages (from requests->fastai) (2.9)
Requirement already satisfied, skipping upgrade: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/opt/conda/lib/python3.7/site-packages (from requests->fastai) (1.24.3)
Requirement already satisfied, skipping upgrade: certifi>=2017.4.17 in /opt/conda/lib/pytho
n3.7/site-packages (from requests->fastai) (2020.6.20)
Requirement already satisfied, skipping upgrade: chardet<4,>=3.0.2 in /opt/conda/lib/python
3.7/site-packages (from requests->fastai) (3.0.4)
Requirement already satisfied, skipping upgrade: importlib-metadata>=0.20; python_version <
"3.8" in /opt/conda/lib/python3.7/site-packages (from catalogue<1.1.0,>=0.0.7->spacy->fasta
i) (1.6.0)
Requirement already satisfied, skipping upgrade: zipp>=0.5 in /opt/conda/lib/python3.7/site
-packages (from importlib-metadata>=0.20; python_version < "3.8"->catalogue<1.1.0,>=0.0.7->
spacy->fastai) (3.1.0)
Installing collected packages: fastcore, fastai
  Attempting uninstall: fastai
    Found existing installation: fastai 1.0.61
    Uninstalling fastai-1.0.61:
      Successfully uninstalled fastai-1.0.61
Successfully installed fastai-2.0.10 fastcore-1.0.9
WARNING: You are using pip version 20.2.2; however, version 20.2.3 is available.
You should consider upgrading via the '/opt/conda/bin/python3.7 -m pip install --upgrade pi
p' command.
```

In [2]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
```

In [3]:

```
import torch
print(torch.__version__)
print(torch.cuda.is_available())

import fastai
print(fastai.__version__)

from fastai.vision.all import *
```

```
1.6.0+cu101
True
2.0.10
```

In [4]:

```
def random_seed(seed_value):
    import random
    random.seed(seed_value) # Python
    import numpy as np
    np.random.seed(seed_value) # cpu vars
    import torch
    torch.manual_seed(seed_value) # cpu  vars

    if torch.cuda.is_available():
        torch.cuda.manual_seed(seed_value)
        torch.cuda.manual_seed_all(seed_value) # gpu vars
        torch.backends.cudnn.deterministic = True  #needed
        torch.backends.cudnn.benchmark = False

random_seed(42)
```

Data

In [5]:

```
test_df      = pd.read_csv("../input/3pochs-tweet-banjir/test-g13.csv")
train_df     = pd.read_csv("../input/3pochs-tweet-banjir/train-g13.csv")
unlabeled_df = pd.read_csv("../input/3pochs-tweet-banjir/unlabeled-g13.csv")
```

In [6]:

```
y_test = test_df.label.apply(lambda l: 0 if l == "INFO" else 1)
```

Image Baseline

In [7]:

```
fpath = Path("/kaggle/input")
```

In [8]:

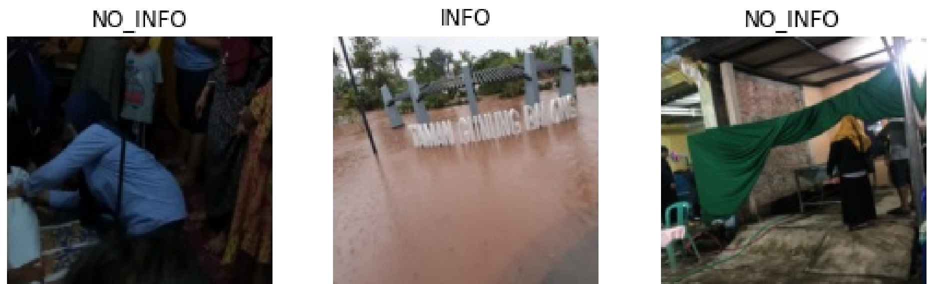
```
item_tfms =  [RandomResizedCrop(380, min_scale=0.75)]
batch_tfms = [*aug_transforms(max_warp=0.1), Normalize.from_stats(*imagenet_stats)]
```

In [9]:

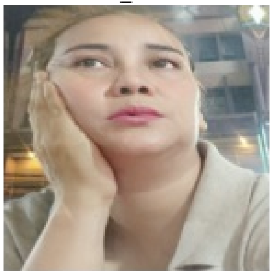
```
def get_dls_from_df(df):
    df = df.copy()
    options = {
        "item_tfms": item_tfms,
        "batch_tfms": batch_tfms,
        "bs": 32,
        "valid_pct": 0.2
    }
    dls = ImageDataLoaders.from_df(df, fpath, **options)
    return dls
```

In [10]:

```
dls = get_dls_from_df(train_df[["media", "label"]])
dls.show_batch()
```

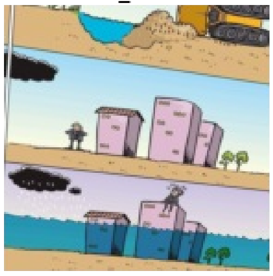


NO_INFO

NO_INFO

NO_INFO

NO_INFO

NO_INFO

NO_INFO


In [11]:

```
from fastai.callback.cutmix import *
from efficientnet_pytorch import EfficientNet
cutmix = CutMix()
model = EfficientNet.from_pretrained('efficientnet-b4', num_classes=2)
```

```
Downloading: "https://github.com/lukemelas/EfficientNet-PyTorch/releases/download/1.0/effic
ientnet-b4-6ed6700e.pth" to /root/.cache/torch/hub/checkpoints/efficientnet-b4-6ed6700e.pth
```

```
Loaded pretrained weights for efficientnet-b4
```

In [12]:

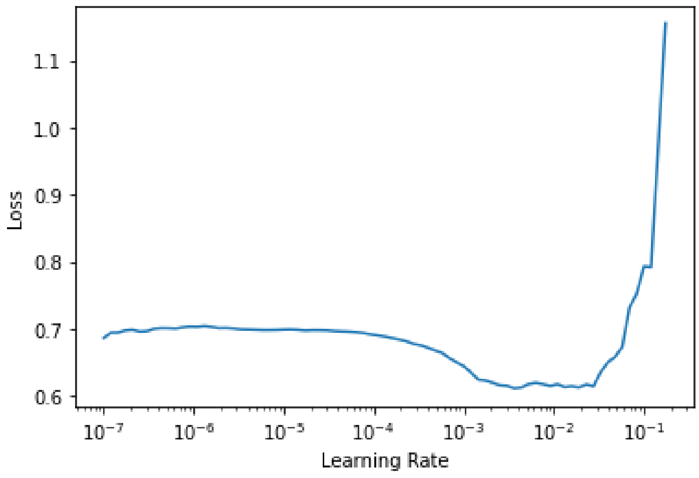
```
learn = Learner(dls, model, metrics=[accuracy, F1Score()], cbs=[cutmix], path="/kaggle/wor
king/").to_fp16()
```

In [13]:

```
slr = learn.lr_find()
slr
```

Out[13]:

```
SuggestedLRs(lr_min=0.00036307806149125097, lr_steep=0.0012022644514217973)
```



In [14]:

```
learn.fine_tune(5, 4e-4)
```

epoch	train_loss	valid_loss	accuracy	f1_score	time
0	0.541036	0.426585	0.807500	0.871022	01:07

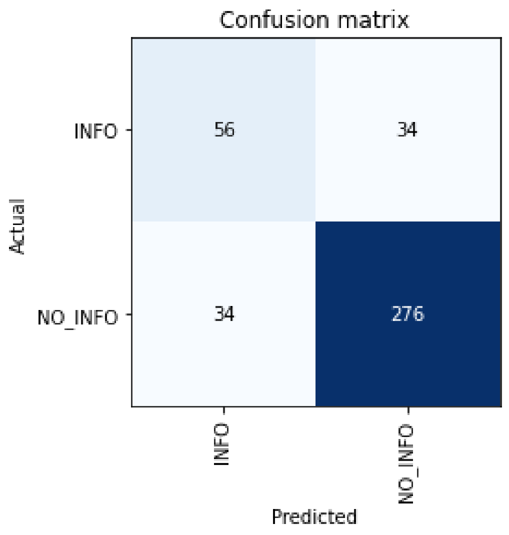
epoch	train_loss	valid_loss	accuracy	f1_score	time
0	0.395365	0.420972	0.800000	0.866221	01:03
1	0.384562	0.491703	0.780000	0.846690	01:03
2	0.362773	0.448873	0.802500	0.865417	01:03
3	0.323742	0.410481	0.827500	0.888889	01:03
4	0.285487	0.414047	0.830000	0.890323	01:04

In [15]:

```
interp = ClassificationInterpretation.from_learner(learn)
# fix for bug not applying argmax and softmax to decoded
interp.decoded = np.argmax(interp.decoded, 1)
```

In [16]:

```
interp.plot_confusion_matrix()
```



In [17]:

```
interp.print_classification_report()
```

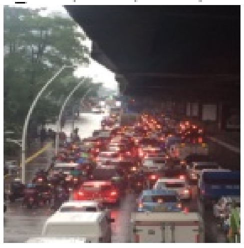
```
              precision    recall  f1-score   support

        INFO       0.62      0.62      0.62        90
     NO_INFO       0.89      0.89      0.89       310

    accuracy                           0.83       400
   macro avg       0.76      0.76      0.76       400
weighted avg       0.83      0.83      0.83       400
```

In [18]:

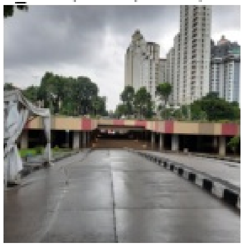
```
interp.plot_top_losses(k=25)
```

Prediction/Actual/Loss/Probability

NO_INFO/INFO / 4.46 / 2.20



NO_INFO/INFO / 3.62 / 1.95



NO_INFO/INFO / 3.59 / 1.96

MAL
TAMAN ANGGREK

PEMBERITAHUAN

[illegible]

NO_INFO/INFO / 3.58 / 1.71

NO_INFO/INFO / 3.58 / 1.71

INFO/NO_INFO / 3.20 / 1.50
NO_INFO/INFO / 2.96 / 1.56
INFO/NO_INFO / 2.89 / 1.37
INFO/NO_INFO / 2.85 / 1.36
NO_INFO/INFO / 2.77 / 1.32
INFO/NO_INFO / 2.58 / 1.15
NO_INFO/INFO / 2.57 / 1.35
NO_INFO/INFO / 2.54 / 1.36
NO_INFO/INFO / 2.39 / 1.06
NO_INFO/INFO / 2.37 / 1.07
INFO/NO_INFO / 2.27 / 1.06
INFO/NO_INFO / 2.18 / 0.98
INFO/NO_INFO / 2.16 / 0.99
NO_INFO/INFO / 2.15 / 0.94
INFO/NO_INFO / 2.14 / 1.02
INFO/NO_INFO / 2.07 / 0.90
NO_INFO/INFO / 1.93 / 0.99
INFO/NO_INFO / 1.89 / 0.86
INFO/NO_INFO / 1.88 / 0.78
NO_INFO/INFO / 1.74 / 0.70

In [19]:

```
learn.save('image-baseline')
```

Out[19]:

```
Path('/kaggle/working/models/image-baseline.pth')
```

Test Result

In [20]:

```
from sklearn.metrics import classification_report
def get_test_metrics(y, ypred):
    print(classification_report(y, ypred, target_names=['INFO', 'NO_INFO'], digits=4))
```

In [21]:

```
image_test_dl = dls.test_dl(test_df.media.apply(lambda fn: fpath/fn))
```

In [22]:

```
image_test_preds = learn.get_preds(dl=image_test_dl, with_decoded=True)
```

In [23]:

```
image_y_pred = np.argmax(image_test_preds[0], 1)
get_test_metrics(y_test, image_y_pred)
```

```
              precision    recall  f1-score   support

        INFO     0.6183    0.6773    0.6464       220
     NO_INFO     0.9065    0.8821    0.8941       780
```

```
    accuracy                         0.8370      1000
   macro avg     0.7624    0.7797    0.7703      1000
weighted avg     0.8431    0.8370    0.8396      1000
```

Image + Pseudo Labeling

Simple Approach (Taking Confident Predictions)

In [66]:

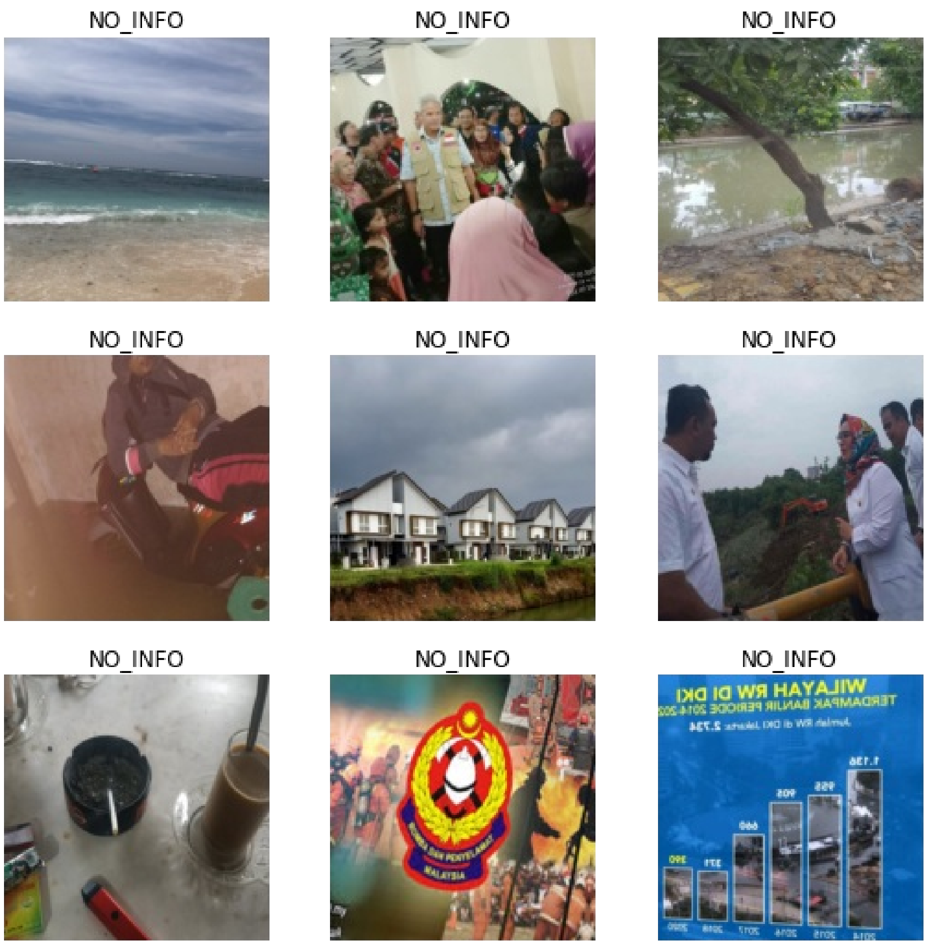
```
unlabeled_dls = dls.test_dl(str(fpath)+"/"+unlabeled_df.media)
pred_unlabeled = learn.get_preds(dl=unlabeled_dls, with_decoded=True)
```

In [67]:

```
pred_proba = pd.DataFrame(torch.sigmoid(pred_unlabeled[0]))

pred_proba.loc[pred_proba[0] < 0.1, 'label'] = 'NO_INFO'
pred_proba.loc[pred_proba[0] > 0.93, 'label'] = 'INFO'

to_add = pred_proba[~pd.isnull(pred_proba.label)]['label']

c = pd.concat([unlabeled_df.reset_index(drop=True), to_add], 1)
to_add = c[~pd.isnull(c.label)]

combined_image_df = pd.concat([train_df, to_add], ignore_index=True)
```

In [68]:

```
pi_dls = get_dls_from_df(combined_image_df[['media', 'label']])
pi_dls.show_batch()
```



In [69]:

```
pi_model = EfficientNet.from_pretrained('efficientnet-b4', num_classes=2)
```

```
pi_learn = Learner(pi_dls, pi_model, metrics=[accuracy, F1Score()], cbs=[CutMix()], path="
/kaggle/working/").to_fp16()
```

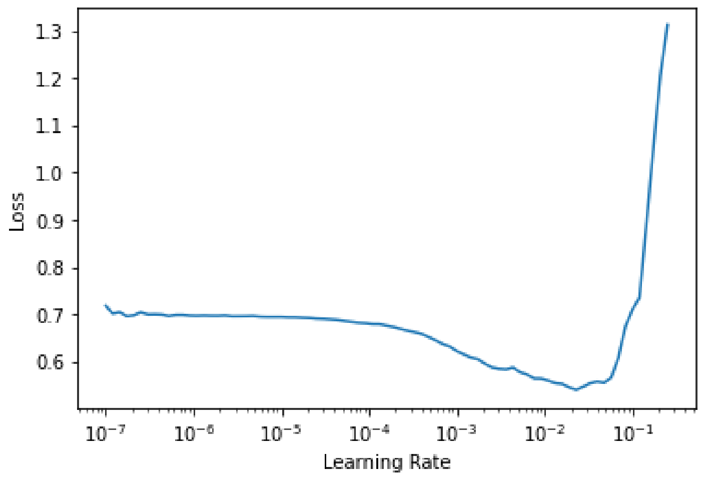
```
Loaded pretrained weights for efficientnet-b4
```

In [70]:

```
slr = pi_learn.lr_find()
slr
```

Out[70]:

```
SuggestedLRs(lr_min=0.002290867641568184, lr_steep=0.004365158267319202)
```



In [71]:

```
pi_learn.fine_tune(4, 3e-4)
```

epoch	train_loss	valid_loss	accuracy	f1_score	time
0	0.456657	0.289795	0.874101	0.923913	01:25

epoch	train_loss	valid_loss	accuracy	f1_score	time
0	0.329626	0.295380	0.881295	0.927313	01:27
1	0.297884	0.264945	0.883094	0.932221	01:25
2	0.269046	0.258433	0.875899	0.926829	01:25
3	0.249324	0.258654	0.875899	0.926674	01:25

In [72]:

```
pi_preds = pi_learn.get_preds(dl=image_test_dl, with_decoded=True)
get_test_metrics(y_test, np.argmax(pi_preds[0], 1))
```

```
              precision    recall  f1-score   support

        INFO     0.6699    0.6364    0.6527       220
     NO_INFO     0.8989    0.9115    0.9052       780

    accuracy                         0.8510      1000
   macro avg     0.7844    0.7740    0.7789      1000
weighted avg     0.8485    0.8510    0.8496      1000
```

Text Baseline

In [31]:

```
!pip install transformers
!pip install ohmeow-blurr
!pip install tweet-preprocessor
```

```
Requirement already satisfied: transformers in /opt/conda/lib/python3.7/site-packages (3.0.
2)
Requirement already satisfied: sentencepiece!=0.1.92 in /opt/conda/lib/python3.7/site-packa
ges (from transformers) (0.1.91)
Requirement already satisfied: tqdm>=4.27 in /opt/conda/lib/python3.7/site-packages (from t
ransformers) (4.45.0)
Requirement already satisfied: requests in /opt/conda/lib/python3.7/site-packages (from tra
nsformers) (2.23.0)
Requirement already satisfied: sacremoses in /opt/conda/lib/python3.7/site-packages (from t
ransformers) (0.0.43)
Requirement already satisfied: packaging in /opt/conda/lib/python3.7/site-packages (from tr
ansformers) (20.1)
Requirement already satisfied: filelock in /opt/conda/lib/python3.7/site-packages (from tra
nsformers) (3.0.10)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages (from transf
ormers) (1.18.5)
Requirement already satisfied: regex!=2019.12.17 in /opt/conda/lib/python3.7/site-packages
(from transformers) (2020.4.4)
Requirement already satisfied: tokenizers==0.8.1.rc1 in /opt/conda/lib/python3.7/site-packa
ges (from transformers) (0.8.1rc1)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/lib/python3.7/site-packages (from
requests->transformers) (2.9)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.7/site-packages
(from requests->transformers) (2020.6.20)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /opt/conda/lib/py
thon3.7/site-packages (from requests->transformers) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in /opt/conda/lib/python3.7/site-packages
(from requests->transformers) (3.0.4)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from sacremos
es->transformers) (1.14.0)
Requirement already satisfied: click in /opt/conda/lib/python3.7/site-packages (from sacrem
oses->transformers) (7.1.1)
Requirement already satisfied: joblib in /opt/conda/lib/python3.7/site-packages (from sacre
moses->transformers) (0.14.1)
Requirement already satisfied: pyparsing>=2.0.2 in /opt/conda/lib/python3.7/site-packages (
from packaging->transformers) (2.4.7)
WARNING: You are using pip version 20.2.2; however, version 20.2.3 is available.
You should consider upgrading via the '/opt/conda/bin/python3.7 -m pip install --upgrade pi
p' command.
Collecting ohmeow-blurr
  Downloading ohmeow_blurr-0.0.12-py3-none-any.whl (39 kB)
Collecting nlp
  Downloading nlp-0.4.0-py3-none-any.whl (1.7 MB)
     |████████████████████████████████| 1.7 MB 1.3 MB/s eta 0:00:01
Requirement already satisfied: fastai>=2.0.0 in /opt/conda/lib/python3.7/site-packages (fro
m ohmeow-blurr) (2.0.10)
Collecting seqeval
  Downloading seqeval-0.0.12.tar.gz (21 kB)
Collecting transformers>=3.1.0
  Downloading transformers-3.1.0-py3-none-any.whl (884 kB)
     |████████████████████████████████| 884 kB 5.5 MB/s eta 0:00:01
Requirement already satisfied: ipykernel in /opt/conda/lib/python3.7/site-packages (from oh
meow-blurr) (5.1.1)
Collecting rouge-score
  Downloading rouge_score-0.0.4-py2.py3-none-any.whl (22 kB)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages (from nlp->o
hmeow-blurr) (1.18.5)
Requirement already satisfied: pyarrow>=0.16.0 in /opt/conda/lib/python3.7/site-packages (f
rom nlp->ohmeow-blurr) (0.16.0)
Requirement already satisfied: filelock in /opt/conda/lib/python3.7/site-packages (from nlp
->ohmeow-blurr) (3.0.10)
Collecting xxhash
  Downloading xxhash-2.0.0-cp37-cp37m-manylinux2010_x86_64.whl (243 kB)
     |████████████████████████████████| 243 kB 6.4 MB/s eta 0:00:01
Requirement already satisfied: requests>=2.19.0 in /opt/conda/lib/python3.7/site-packages (
from nlp->ohmeow-blurr) (2.23.0)
Requirement already satisfied: dill in /opt/conda/lib/python3.7/site-packages (from nlp->oh
meow-blurr) (0.3.2)
Requirement already satisfied: tqdm>=4.27 in /opt/conda/lib/python3.7/site-packages (from n
lp->ohmeow-blurr) (4.45.0)
Requirement already satisfied: pandas in /opt/conda/lib/python3.7/site-packages (from nlp->
ohmeow-blurr) (1.1.1)
Requirement already satisfied: packaging in /opt/conda/lib/python3.7/site-packages (from fa
stai>=2.0.0->ohmeow-blurr) (20.1)
Requirement already satisfied: scikit-learn in /opt/conda/lib/python3.7/site-packages (from
fastai>=2.0.0->ohmeow-blurr) (0.23.2)
Requirement already satisfied: fastprogress>=0.2.4 in /opt/conda/lib/python3.7/site-package
s (from fastai>=2.0.0->ohmeow-blurr) (1.0.0)
Requirement already satisfied: pyyaml in /opt/conda/lib/python3.7/site-packages (from fasta
i>=2.0.0->ohmeow-blurr) (5.3.1)
Requirement already satisfied: spacy in /opt/conda/lib/python3.7/site-packages (from fastai
>=2.0.0->ohmeow-blurr) (2.2.4)
Requirement already satisfied: pillow in /opt/conda/lib/python3.7/site-packages (from fasta
i>=2.0.0->ohmeow-blurr) (7.2.0)
Requirement already satisfied: scipy in /opt/conda/lib/python3.7/site-packages (from fastai
>=2.0.0->ohmeow-blurr) (1.4.1)
Requirement already satisfied: torch>=1.6.0 in /opt/conda/lib/python3.7/site-packages (from
fastai>=2.0.0->ohmeow-blurr) (1.6.0+cu101)
Requirement already satisfied: fastcore>=1.0.5 in /opt/conda/lib/python3.7/site-packages (f
rom fastai>=2.0.0->ohmeow-blurr) (1.0.9)
Requirement already satisfied: torchvision>=0.7 in /opt/conda/lib/python3.7/site-packages (
from fastai>=2.0.0->ohmeow-blurr) (0.7.0+cu101)
Requirement already satisfied: pip in /opt/conda/lib/python3.7/site-packages (from fastai>=
2.0.0->ohmeow-blurr) (20.2.2)
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.7/site-packages (from f
astai>=2.0.0->ohmeow-blurr) (3.2.1)
Requirement already satisfied: Keras>=2.2.4 in /opt/conda/lib/python3.7/site-packages (from
seqeval->ohmeow-blurr) (2.4.3)
Requirement already satisfied: sacremoses in /opt/conda/lib/python3.7/site-packages (from t
ransformers>=3.1.0->ohmeow-blurr) (0.0.43)
Requirement already satisfied: sentencepiece!=0.1.92 in /opt/conda/lib/python3.7/site-packa
ges (from transformers>=3.1.0->ohmeow-blurr) (0.1.91)
Requirement already satisfied: regex!=2019.12.17 in /opt/conda/lib/python3.7/site-packages
(from transformers>=3.1.0->ohmeow-blurr) (2020.4.4)
Collecting tokenizers==0.8.1.rc2
  Downloading tokenizers-0.8.1rc2-cp37-cp37m-manylinux1_x86_64.whl (3.0 MB)
     |████████████████████████████████| 3.0 MB 6.5 MB/s eta 0:00:01
Requirement already satisfied: ipython>=5.0.0 in /opt/conda/lib/python3.7/site-packages (fr
om ipykernel->ohmeow-blurr) (7.13.0)
Requirement already satisfied: tornado>=4.2 in /opt/conda/lib/python3.7/site-packages (from
ipykernel->ohmeow-blurr) (5.0.2)
Requirement already satisfied: jupyter-client in /opt/conda/lib/python3.7/site-packages (fr
om ipykernel->ohmeow-blurr) (6.1.3)
Requirement already satisfied: traitlets>=4.1.0 in /opt/conda/lib/python3.7/site-packages (
from ipykernel->ohmeow-blurr) (4.3.3)
Requirement already satisfied: absl-py in /opt/conda/lib/python3.7/site-packages (from roug
e-score->ohmeow-blurr) (0.10.0)
Requirement already satisfied: six>=1.14.0 in /opt/conda/lib/python3.7/site-packages (from
rouge-score->ohmeow-blurr) (1.14.0)
Requirement already satisfied: nltk in /opt/conda/lib/python3.7/site-packages (from rouge-s
core->ohmeow-blurr) (3.2.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /opt/conda/lib/py
thon3.7/site-packages (from requests>=2.19.0->nlp->ohmeow-blurr) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.7/site-packages
(from requests>=2.19.0->nlp->ohmeow-blurr) (2020.6.20)
Requirement already satisfied: chardet<4,>=3.0.2 in /opt/conda/lib/python3.7/site-packages
(from requests>=2.19.0->nlp->ohmeow-blurr) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/lib/python3.7/site-packages (from
requests>=2.19.0->nlp->ohmeow-blurr) (2.9)
Requirement already satisfied: python-dateutil>=2.7.3 in /opt/conda/lib/python3.7/site-pack
ages (from pandas->nlp->ohmeow-blurr) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /opt/conda/lib/python3.7/site-packages (from
pandas->nlp->ohmeow-blurr) (2019.3)
Requirement already satisfied: pyparsing>=2.0.2 in /opt/conda/lib/python3.7/site-packages (
from packaging->fastai>=2.0.0->ohmeow-blurr) (2.4.7)
Requirement already satisfied: joblib>=0.11 in /opt/conda/lib/python3.7/site-packages (from
scikit-learn->fastai>=2.0.0->ohmeow-blurr) (0.14.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /opt/conda/lib/python3.7/site-packag
es (from scikit-learn->fastai>=2.0.0->ohmeow-blurr) (2.1.0)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /opt/conda/lib/python3.7/site-packa
ges (from spacy->fastai>=2.0.0->ohmeow-blurr) (3.0.2)
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in /opt/conda/lib/python3.7/site-pac
kages (from spacy->fastai>=2.0.0->ohmeow-blurr) (1.0.0)
Requirement already satisfied: blis<0.5.0,>=0.4.0 in /opt/conda/lib/python3.7/site-packages
(from spacy->fastai>=2.0.0->ohmeow-blurr) (0.4.1)
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in /opt/conda/lib/python3.7/site-packag
```

```
es (from spacy->fastai>=2.0.0->ohmeow-blurr) (0.7.1)
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in /opt/conda/lib/python3.7/site-package
s (from spacy->fastai>=2.0.0->ohmeow-blurr) (1.0.2)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /opt/conda/lib/python3.7/site-package
s (from spacy->fastai>=2.0.0->ohmeow-blurr) (2.0.3)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.7/site-packages (from s
pacy->fastai>=2.0.0->ohmeow-blurr) (46.1.3.post20200325)
Requirement already satisfied: plac<1.2.0,>=0.9.6 in /opt/conda/lib/python3.7/site-packages
(from spacy->fastai>=2.0.0->ohmeow-blurr) (1.1.3)
Requirement already satisfied: thinc==7.4.0 in /opt/conda/lib/python3.7/site-packages (from
spacy->fastai>=2.0.0->ohmeow-blurr) (7.4.0)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /opt/conda/lib/python3.7/site-p
ackages (from spacy->fastai>=2.0.0->ohmeow-blurr) (1.0.2)
Requirement already satisfied: future in /opt/conda/lib/python3.7/site-packages (from torch
>=1.6.0->fastai>=2.0.0->ohmeow-blurr) (0.18.2)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-packages (from
matplotlib->fastai>=2.0.0->ohmeow-blurr) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.7/site-packages
(from matplotlib->fastai>=2.0.0->ohmeow-blurr) (1.2.0)
Requirement already satisfied: h5py in /opt/conda/lib/python3.7/site-packages (from Keras>=
2.2.4->seqeval->ohmeow-blurr) (2.10.0)
Requirement already satisfied: click in /opt/conda/lib/python3.7/site-packages (from sacrem
oses->transformers>=3.1.0->ohmeow-blurr) (7.1.1)
Requirement already satisfied: pickleshare in /opt/conda/lib/python3.7/site-packages (from
ipython>=5.0.0->ipykernel->ohmeow-blurr) (0.7.5)
Requirement already satisfied: pexpect; sys_platform != "win32" in /opt/conda/lib/python3.7
/site-packages (from ipython>=5.0.0->ipykernel->ohmeow-blurr) (4.8.0)
Requirement already satisfied: backcall in /opt/conda/lib/python3.7/site-packages (from ipy
thon>=5.0.0->ipykernel->ohmeow-blurr) (0.1.0)
Requirement already satisfied: pygments in /opt/conda/lib/python3.7/site-packages (from ipy
thon>=5.0.0->ipykernel->ohmeow-blurr) (2.6.1)
Requirement already satisfied: jedi>=0.10 in /opt/conda/lib/python3.7/site-packages (from i
python>=5.0.0->ipykernel->ohmeow-blurr) (0.15.2)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /opt/conda/l
ib/python3.7/site-packages (from ipython>=5.0.0->ipykernel->ohmeow-blurr) (3.0.5)
Requirement already satisfied: decorator in /opt/conda/lib/python3.7/site-packages (from ip
ython>=5.0.0->ipykernel->ohmeow-blurr) (4.4.2)
Requirement already satisfied: jupyter-core>=4.6.0 in /opt/conda/lib/python3.7/site-package
s (from jupyter-client->ipykernel->ohmeow-blurr) (4.6.3)
Requirement already satisfied: pyzmq>=13 in /opt/conda/lib/python3.7/site-packages (from ju
pyter-client->ipykernel->ohmeow-blurr) (19.0.0)
Requirement already satisfied: ipython-genutils in /opt/conda/lib/python3.7/site-packages (
from traitlets>=4.1.0->ipykernel->ohmeow-blurr) (0.2.0)
Requirement already satisfied: importlib-metadata>=0.20; python_version < "3.8" in /opt/con
da/lib/python3.7/site-packages (from catalogue<1.1.0,>=0.0.7->spacy->fastai>=2.0.0->ohmeow-
blurr) (1.6.0)
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/lib/python3.7/site-packages (f
rom pexpect; sys_platform != "win32"->ipython>=5.0.0->ipykernel->ohmeow-blurr) (0.6.0)
Requirement already satisfied: parso>=0.5.2 in /opt/conda/lib/python3.7/site-packages (from
jedi>=0.10->ipython>=5.0.0->ipykernel->ohmeow-blurr) (0.5.2)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.7/site-packages (from prom
pt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=5.0.0->ipykernel->ohmeow-blurr) (0.1.9)
Requirement already satisfied: zipp>=0.5 in /opt/conda/lib/python3.7/site-packages (from im
portlib-metadata>=0.20; python_version < "3.8"->catalogue<1.1.0,>=0.0.7->spacy->fastai>=2.0
.0->ohmeow-blurr) (3.1.0)
Building wheels for collected packages: seqeval
  Building wheel for seqeval (setup.py) ... done
  Created wheel for seqeval: filename=seqeval-0.0.12-py3-none-any.whl size=7423 sha256=fa41
4751e4414e33ea921f40fff4db1611073e0cf1f41fc7d90fd4a069998f47
  Stored in directory: /root/.cache/pip/wheels/dc/cc/62/a3b81f92d35a80e39eb9b2a9d8b31abac54
c02b21b2d466edc
Successfully built seqeval
Installing collected packages: xxhash, nlp, seqeval, tokenizers, transformers, rouge-score,
ohmeow-blurr
  Attempting uninstall: tokenizers
    Found existing installation: tokenizers 0.8.1rc1
    Uninstalling tokenizers-0.8.1rc1:
      Successfully uninstalled tokenizers-0.8.1rc1
  Attempting uninstall: transformers
    Found existing installation: transformers 3.0.2
    Uninstalling transformers-3.0.2:
      Successfully uninstalled transformers-3.0.2
ERROR: After October 2020 you may experience errors when installing or updating packages. T
```

```
his is because pip will change the way that it resolves dependency conflicts.

We recommend you use --use-feature=2020-resolver to test your packages with the new resolve
r before it becomes the default.

allennlp 1.1.0rc4 requires transformers<3.1,>=3.0, but you'll have transformers 3.1.0 which
is incompatible.
Successfully installed nlp-0.4.0 ohmeow-blurr-0.0.12 rouge-score-0.0.4 seqeval-0.0.12 token
izers-0.8.1rc2 transformers-3.1.0 xxhash-2.0.0
WARNING: You are using pip version 20.2.2; however, version 20.2.3 is available.
You should consider upgrading via the '/opt/conda/bin/python3.7 -m pip install --upgrade pi
p' command.
Collecting tweet-preprocessor
  Downloading tweet_preprocessor-0.6.0-py3-none-any.whl (27 kB)
Installing collected packages: tweet-preprocessor
Successfully installed tweet-preprocessor-0.6.0
WARNING: You are using pip version 20.2.2; however, version 20.2.3 is available.
You should consider upgrading via the '/opt/conda/bin/python3.7 -m pip install --upgrade pi
p' command.
```

In [32]:

```
from transformers import *
from fastai.text.all import *

from blurr.data.all import *
from blurr.modeling.all import *
```

```
wandb: WARNING W&B installed but not logged in.  Run `wandb login` or set the WANDB_API_KEY
env variable.
```

In [33]:

```
task = HF_TASKS_AUTO.SequenceClassification

pretrained_model_name = "cahya/bert-base-indonesian-522M"
hf_arch, hf_config, hf_tokenizer, hf_model = BLURR_MODEL_HELPER.get_hf_objects(pretrained_
model_name, task=task)
```

```
Some weights of the model checkpoint at cahya/bert-base-indonesian-522M were not used when
initializing BertForSequenceClassification: ['cls.predictions.bias', 'cls.predictions.trans
form.dense.weight', 'cls.predictions.transform.dense.bias', 'cls.predictions.transform.Laye
rNorm.weight', 'cls.predictions.transform.LayerNorm.bias', 'cls.predictions.decoder.weight'
, 'cls.predictions.decoder.bias']
- This IS expected if you are initializing BertForSequenceClassification from the checkpoin
t of a model trained on another task or with another architecture (e.g. initializing a Bert
ForSequenceClassification model from a BertForPretraining model).
- This IS NOT expected if you are initializing BertForSequenceClassification from the check
point of a model that you expect to be exactly identical (initializing a BertForSequenceCla
ssification model from a BertForSequenceClassification model).
Some weights of BertForSequenceClassification were not initialized from the model checkpoin
t at cahya/bert-base-indonesian-522M and are newly initialized: ['classifier.weight', 'clas
sifier.bias']
You should probably TRAIN this model on a down-stream task to be able to use it for predict
ions and inference.
```

In [34]:

```
blocks = (HF_TextBlock(hf_arch=hf_arch, hf_tokenizer=hf_tokenizer), CategoryBlock)

dblock = DataBlock(blocks=blocks,
                   get_x=ColReader('text'), get_y=ColReader('label'),
                   splitter=ColSplitter(col='is_valid'))
```

In [35]:

```
import preprocessor as p
def preprocess_text_df(df, sample=False):
    d = df.copy()
    d['text'] = d.text.apply(lambda s: re.sub('#[Bb]anjir\w*', 'banjir', s))
    d['text'] = d['text'].apply(str)
    d['text'] = d.text.apply(p.clean)
    if sample:
        d['is_valid'] = False
        d.loc[d.sample(int(0.2*len(d))).index, 'is_valid'] = True
    return d
```

In [36]:

```
d = preprocess_text_df(train_df, sample=True)
tdls = dblock.dataloaders(d, bs=4)
tdls.show_batch(hf_tokenizer=hf_tokenizer, max_n=4)
```

	text	category
0	**brebes ada banjir juga. tapi untungnya jauh dari pabrik. aman.**	**NO_INFO**
1	**waspada leptospirosis saat banjir terjadi.**	**NO_INFO**
2	**badan nasional penanggulangan bencana melaporkan, banyak taksi terendam banjir di pool bluebird kramat jati belakang pasar hek di jalan pondok gede. foto ini diambil tadi pagi pukul wib.**	**INFO**
3	**setelah banjir menerjang.. banjir**	**NO_INFO**

In [37]:

```
tdls.vocab
```

Out[37]:

```
(#2) ['INFO','NO_INFO']
```

In [38]:

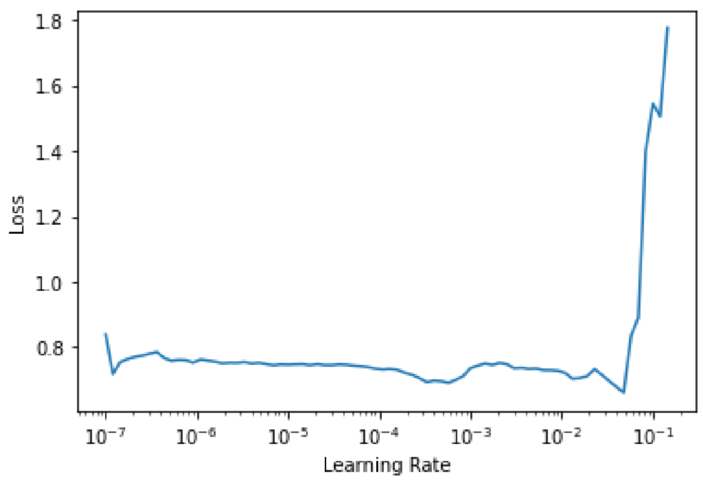
```
tmodel = HF_BaseModelWrapper(hf_model)

tlearn = Learner(tdls,
                 tmodel,
                 loss_func=CrossEntropyLossFlat(),
                 metrics=[accuracy],
                 cbs=[HF_BaseModelCallback],
                 splitter=hf_splitter)
```

In [39]:

```
tlearn.freeze()
tslr = tlearn.lr_find()
tslr
```

Out[39]:

```
SuggestedLRs(lr_min=0.004786301031708717, lr_steep=0.10000000149011612)
```



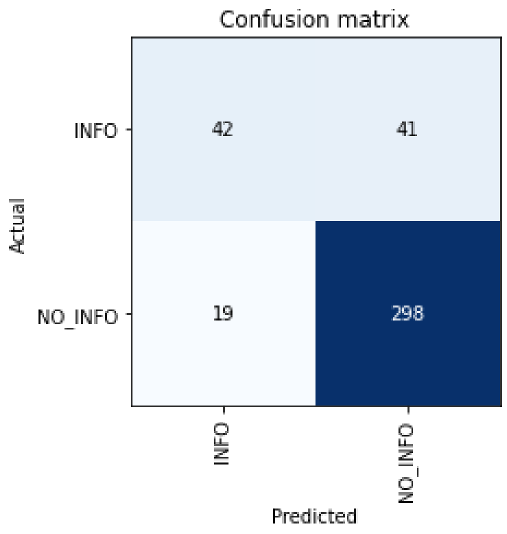
In [40]:

```
tlearn.fit_one_cycle(5, lr_max=1e-3)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.512943	0.591693	0.642500	00:22
1	0.453488	0.397313	0.840000	00:21
2	0.386424	0.366559	0.845000	00:22
3	0.383829	0.350131	0.845000	00:21
4	0.329276	0.350658	0.850000	00:21

In [41]:

```
interp = ClassificationInterpretation(tdls.valid, tdls.valid.items.text.values, *tlearn.ge
t_preds(dl=tdls.valid, with_input=False, with_loss=True, with_decoded=True))
interp.plot_confusion_matrix()
```



In [42]:

```
interp.print_classification_report()
```

```
              precision    recall  f1-score   support

        INFO       0.69      0.51      0.58        83
     NO_INFO       0.88      0.94      0.91       317

    accuracy                           0.85       400
   macro avg       0.78      0.72      0.75       400
weighted avg       0.84      0.85      0.84       400
```

In [43]:

```
tlearn.save('text-baseline')
```

Out[43]:

```
Path('models/text-baseline.pth')
```

Test Result

In [44]:

```
t_test_loader = tdls.test_dl(preprocess_text_df(test_df))
```

In [45]:

```
t_preds = tlearn.get_preds(dl=t_test_loader, with_decoded=True)
```

In [47]:

```
get_test_metrics(y_test, t_preds[2])
```

```
              precision    recall  f1-score   support

        INFO     0.6561    0.4682    0.5464       220
     NO_INFO     0.8612    0.9308    0.8946       780

    accuracy                         0.8290      1000
   macro avg     0.7586    0.6995    0.7205      1000
weighted avg     0.8161    0.8290    0.8180      1000
```

Text + Pseduo Labeling

High Confidence

In [50]:

```
tpred_unlabeled = tlearn.get_preds(dl=tdls.test_dl(preprocess_text_df(unlabeled_df)), with
_decoded=True)
```

In [51]:

```
pred_proba = pd.DataFrame(tpred_unlabeled[0])
pred_proba.loc[pred_proba[0] < 0.02, 'label'] = 'NO_INFO'
pred_proba.loc[pred_proba[0] > 0.98, 'label'] = 'INFO'
to_add = pred_proba[~pd.isnull(pred_proba.label)]['label']
```

In [52]:

```
c = pd.concat([unlabeled_df.reset_index(drop=True), to_add], 1)
to_add = c[~pd.isnull(c.label)]
```

In [53]:

```
pi_combined_tdf = pd.concat([train_df, to_add], ignore_index=True)
```

In [55]:

```
pi_tdls = dblock.dataloaders(preprocess_text_df(pi_combined_tdf, sample=True), bs=4)
pi_tdls.show_batch(hf_tokenizer=hf_tokenizer, max_n=4)
```

	text	category
0	brebes ada banjir juga. tapi untungnya jauh dari pabrik. aman.	NO_INFO
1	sore nanti di : / / promo diskon mantul!!!	NO_INFO
2	selasa, januari membawa truk tangki air bersih dibantu oleh damkar dan mendistribusikannya untuk korban banjir di kel. pengadegan, kec. pancoran, jakarta selatandiv. infokomksr pmi umk jakarta selatan2018 - 2020	NO_INFO
3	waspada leptospirosis saat terjadi banjir dan di usahakan jangan terlalu lama terendam dalam genangan air saat banjir.	NO_INFO

In [56]:

```
pi_hf_arch, pi_hf_config, pi_hf_tokenizer, pi_hf_model = BLURR_MODEL_HELPER.get_hf_objects
(pretrained_model_name, task=task)
```

```
Some weights of the model checkpoint at cahya/bert-base-indonesian-522M were not used when
initializing BertForSequenceClassification: ['cls.predictions.bias', 'cls.predictions.trans
form.dense.weight', 'cls.predictions.transform.dense.bias', 'cls.predictions.transform.Laye
rNorm.weight', 'cls.predictions.transform.LayerNorm.bias', 'cls.predictions.decoder.weight'
, 'cls.predictions.decoder.bias']
- This IS expected if you are initializing BertForSequenceClassification from the checkpoin
t of a model trained on another task or with another architecture (e.g. initializing a Bert
ForSequenceClassification model from a BertForPretraining model).
- This IS NOT expected if you are initializing BertForSequenceClassification from the check
point of a model that you expect to be exactly identical (initializing a BertForSequenceCla
ssification model from a BertForSequenceClassification model).
Some weights of BertForSequenceClassification were not initialized from the model checkpoin
t at cahya/bert-base-indonesian-522M and are newly initialized: ['classifier.weight', 'clas
sifier.bias']
You should probably TRAIN this model on a down-stream task to be able to use it for predict
ions and inference.
```

In [57]:

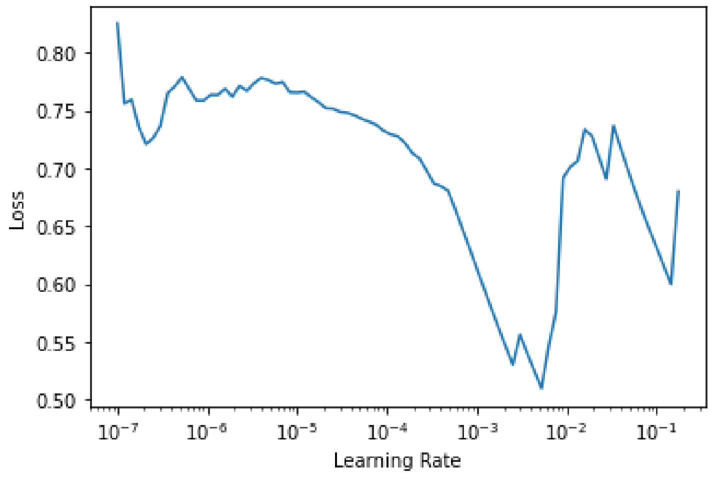
```
pi_tmodel = HF_BaseModelWrapper(pi_hf_model)

pi_tlearn = Learner(pi_tdls,
                pi_tmodel,
                loss_func=CrossEntropyLossFlat(),
                metrics=[accuracy, F1Score()],
                cbs=[HF_BaseModelCallback],
                splitter=hf_splitter)
```

In [59]:

```
pi_tlearn.freeze()
slr = pi_tlearn.lr_find()
slr
```

Out[59]:

```
SuggestedLRs(lr_min=0.0005248074419796466, lr_steep=0.033113110810518265)
```



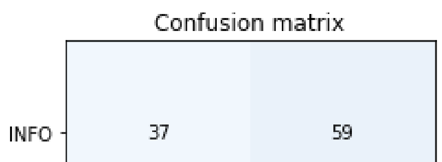
In [60]:

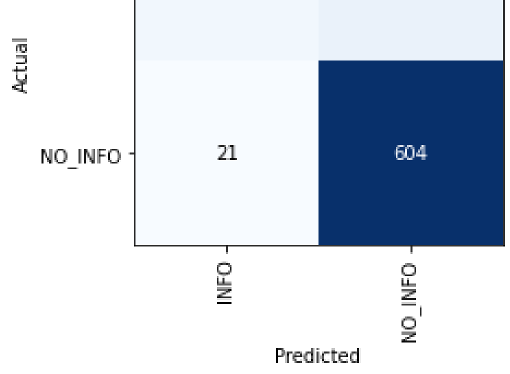
```
pi_tlearn.fit_one_cycle(5, lr_max=4e-4)
```

epoch	train_loss	valid_loss	accuracy	f1_score	time
0	0.271276	0.303114	0.864078	0.926426	00:39
1	0.218176	0.291510	0.872399	0.930827	00:38
2	0.167314	0.243185	0.886269	0.936631	00:38
3	0.229362	0.229158	0.886269	0.936434	00:40
4	0.229943	0.227303	0.889043	0.937888	00:39

In [61]:

```
pit_interp = ClassificationInterpretation(pi_tdls.valid, pi_tdls.valid.items.text.values,
*pi_tlearn.get_preds(dl=pi_tdls.valid, with_input=False, with_loss=True, with_decoded=True
))
pit_interp.plot_confusion_matrix()
```





In [63]:

```
tp2 = pi_tlearn.get_preds(dl=t_test_loader, with_decoded=True)
get_test_metrics(y_test, tp2[2])
```

```
              precision    recall  f1-score   support

        INFO     0.6582    0.4727    0.5503       220
     NO_INFO     0.8622    0.9308    0.8952       780

    accuracy                         0.8300      1000
   macro avg     0.7602    0.7017    0.7227      1000
weighted avg     0.8174    0.8300    0.8193      1000
```

Error Analysis

In [76]:

```
edf = pd.DataFrame([y_test.values,t_preds[2].numpy(), image_y_pred.numpy()]).transpose()
edf
```

Out[76]:

	0	1	2
0	1	1	1
1	1	1	1
2	1	1	1
3	1	1	1
4	0	1	1
...	...	...	...
995	1	1	1
996	0	1	1
997	0	1	0
998	1	0	0
999	1	1	1

1000 rows × 3 columns

In [77]:

```
# 2-2nya salah
print(sum((edf[0] != edf[2])&(edf[1] != edf[0])))
# image bener, text salah
print(sum((edf[0] == edf[2])&(edf[0] != edf[1])))
# text bener, image salah
print(sum((edf[0] == edf[1])&(edf[0] != edf[2])))
# 2"nya bener
print(sum((edf[0] == edf[1])&(edf[0] == edf[2])))
```

```
60
111
103
726
```

In [80]:

```
# pseudo labeled
edf2 = pd.DataFrame([y_test.values,tp2[2].numpy(), np.argmax(pi_preds[0].numpy(), 1)]).tra
nspose()
edf2
```

Out[80]:

	0	1	2
0	1	1	1
1	1	1	1
2	1	1	1
3	1	1	1
4	0	1	1
...	...	...	...
995	1	1	1
996	0	0	1
997	0	1	0
998	1	0	1
999	1	1	1

1000 rows × 3 columns

In [81]:

```
# 2-2nya salah
print(sum((edf2[0] != edf2[2])&(edf2[1] != edf2[0])))
# image bener, text salah
print(sum((edf2[0] == edf2[2])&(edf2[0] != edf2[1])))
# text bener, image salah
print(sum((edf2[0] == edf2[1])&(edf2[0] != edf2[2])))
# 2"nya bener
print(sum((edf2[0] == edf2[1])&(edf2[0] == edf2[2])))
```

```
61
109
88
742
```

Multi-Modality

Decision-level Fusion

In [82]:

```
pred_image_train = learn.get_preds(dl=dls.test_dl('/kaggle/input/'+train_df.media), with_d
ecoded=True)
```

In [83]:

```
t_train = tdls.test_dl(d)
```

In [84]:

```
pred_text_train = tlearn.get_preds(dl=t_train, with_input=False, with_decoded=True)
```

In [85]:

```
ddf = pd.concat([pd.DataFrame(torch.sigmoid(pred_image_train[0])),pd.DataFrame(pred_text_t
rain[0])], 1)
ddf.columns = ["image_info", "image_no_info", "text_info", "text_no_info"]
# ddf['label'] = y.values
```

In [102]:

```
# untuk pseudo-labeled
pi_pred_image_train = pi_learn.get_preds(dl=dls.test_dl('/kaggle/input/'+train_df.media),
with_decoded=True)
pi_pred_text_train = pi_tlearn.get_preds(dl=t_train, with_input=False, with_decoded=True)
piddf = pd.concat([pd.DataFrame(torch.sigmoid(pi_pred_image_train[0])),pd.DataFrame(pi_pre
d_text_train[0])], 1)
piddf.columns = ["image_info", "image_no_info", "text_info", "text_no_info"]
```

In [91]:

```
eddf = pd.concat([pd.DataFrame(torch.sigmoid(image_test_preds[0])),pd.DataFrame(t_preds[0]
)], 1)
eddf.columns = ["image_info", "image_no_info", "text_info", "text_no_info"]
```

Catboost

In [93]:

```
from catboost import CatBoostClassifier
ctb = CatBoostClassifier(verbose=100)
```

In [94]:

```
ctb.fit(ddf, train_df.label.apply(lambda l: 0 if l == "INFO" else 1))
```

```
Learning rate set to 0.013851
0: learn: 0.6665503 total: 63.6ms remaining: 1m 3s
100: learn: 0.1285915 total: 348ms remaining: 3.1s
200: learn: 0.1099088 total: 589ms remaining: 2.34s
300: learn: 0.1024305 total: 816ms remaining: 1.9s
400: learn: 0.0968221 total: 1.04s remaining: 1.55s
500: learn: 0.0918180 total: 1.27s remaining: 1.27s
600: learn: 0.0864928 total: 1.5s remaining: 998ms
700: learn: 0.0821580 total: 1.72s remaining: 735ms
800: learn: 0.0791010 total: 1.94s remaining: 483ms
900: learn: 0.0760982 total: 2.17s remaining: 238ms
999: learn: 0.0725266 total: 2.39s remaining: 0us
```

Out[94]:

```
<catboost.core.CatBoostClassifier at 0x7f5a25ff6f90>
```

In [116]:

```
ctb.fit(ddf, train_df.label.apply(lambda l: 0 if l == "INFO" else 1))
```

```
Learning rate set to 0.013851
0: learn: 0.6665503 total: 3.68ms remaining: 3.68s
100: learn: 0.1285915 total: 217ms remaining: 1.93s
200: learn: 0.1099088 total: 431ms remaining: 1.71s
300: learn: 0.1024305 total: 645ms remaining: 1.5s
400: learn: 0.0968221 total: 859ms remaining: 1.28s
500: learn: 0.0918180 total: 1.07s remaining: 1.07s
600: learn: 0.0864928 total: 1.29s remaining: 856ms
700: learn: 0.0821580 total: 1.5s remaining: 642ms
800: learn: 0.0791010 total: 1.72s remaining: 426ms
900: learn: 0.0760982 total: 1.93s remaining: 212ms
999: learn: 0.0725266 total: 2.14s remaining: 0us
```

Out[116]:

```
<catboost.core.CatBoostClassifier at 0x7f5a25ff6f90>
```

In [117]:

```
get_test_metrics(y_test,ctb.predict(eddf))
```

```
              precision    recall  f1-score   support

        INFO     0.6852    0.6727    0.6789       220
     NO_INFO     0.9082    0.9128    0.9105       780

    accuracy                         0.8600      1000
   macro avg     0.7967    0.7928    0.7947      1000
weighted avg     0.8591    0.8600    0.8595      1000
```

In [119]:

```
#pseudo
ctb.fit(piddf, train_df.label.apply(lambda l: 0 if l == "INFO" else 1))
get_test_metrics(y_test,ctb.predict(eddf))
```

```
Learning rate set to 0.013851
0: learn: 0.6665503 total: 3.04ms remaining: 3.04s
100: learn: 0.1285915 total: 236ms remaining: 2.1s
200: learn: 0.1099088 total: 466ms remaining: 1.85s
300: learn: 0.1024305 total: 696ms remaining: 1.62s
400: learn: 0.0968221 total: 925ms remaining: 1.38s
500: learn: 0.0918180 total: 1.16s remaining: 1.16s
600: learn: 0.0864928 total: 1.39s remaining: 924ms
700: learn: 0.0821580 total: 1.61s remaining: 685ms
800: learn: 0.0791010 total: 1.83s remaining: 455ms
900: learn: 0.0760982 total: 2.06s remaining: 226ms
999: learn: 0.0725266 total: 2.29s remaining: 0us
              precision    recall  f1-score   support

        INFO     0.6852    0.6727    0.6789       220
     NO_INFO     0.9082    0.9128    0.9105       780

    accuracy                         0.8600      1000
   macro avg     0.7967    0.7928    0.7947      1000
weighted avg     0.8591    0.8600    0.8595      1000
```

Linear Regression

In [97]:

```
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression()
```

In [114]:

```
clf.fit(ddf, train_df.label.apply(lambda l: 0 if l == "INFO" else 1))
```

Out[114]:

```
LogisticRegression()
```

In [115]:

```
get_test_metrics(y_test, clf.predict(eddf))
```

```
              precision    recall  f1-score   support

        INFO     0.7005    0.6909    0.6957       220
     NO_INFO     0.9132    0.9167    0.9149       780

    accuracy                         0.8670      1000
   macro avg     0.8068    0.8038    0.8053      1000
weighted avg     0.8664    0.8670    0.8667      1000
```

In [120]:

```
#pseudo
clf.fit(piddf, train_df.label.apply(lambda l: 0 if l == "INFO" else 1))
get_test_metrics(y_test, clf.predict(eddf))
```

```
              precision    recall  f1-score   support

        INFO     0.6783    0.7091    0.6933       220
     NO_INFO     0.9169    0.9051    0.9110       780

    accuracy                         0.8620      1000
   macro avg     0.7976    0.8071    0.8022      1000
weighted avg     0.8644    0.8620    0.8631      1000
```

Random Forest

In [112]:

```
from sklearn.ensemble import RandomForestClassifier
clf2 = RandomForestClassifier().fit(ddf, train_df.label.apply(lambda l: 0 if l == "INFO" e
lse 1))
```

In [113]:

```
get_test_metrics(y_test, clf2.predict(eddf))
```

```
              precision    recall  f1-score   support

        INFO     0.6837    0.6682    0.6759       220
     NO_INFO     0.9070    0.9128    0.9099       780

    accuracy                         0.8590      1000
   macro avg     0.7954    0.7905    0.7929      1000
weighted avg     0.8579    0.8590    0.8584      1000
```

In [118]:

```
# pseudo
clf2 = RandomForestClassifier().fit(piddf, train_df.label.apply(lambda l: 0 if l == "INFO"
else 1))
get_test_metrics(y_test, clf2.predict(eddf))
```

```
              precision    recall  f1-score   support

        INFO     0.6398    0.6864    0.6623       220
     NO_INFO     0.9097    0.8910    0.9003       780

    accuracy                         0.8460      1000
   macro avg     0.7748    0.7887    0.7813      1000
weighted avg     0.8503    0.8460    0.8479      1000
```

BIBLIOGRAFI

E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning, 2019.

A. Kumar and J. Vepa. Gated mechanism for attention based multimodal sentiment analysis, 2020.

C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era, 2017.

M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2019.

T. Zahavy, A. Magnani, A. Krishnan, and S. Mannor. Is a picture worth a thousand words? a deep multi-modal fusion architecture for product classification in e-commerce, 2016.