

```
from fastai import *
import matplotlib.pyplot as plt
from fastai.text import *
import pandas as pd
from sklearn.model_selection import train_test_split
```

```
! git clone https://github.com/ankitbhadu/deep-learning.git
```

```
Cloning into 'deep-learning'...
remote: Enumerating objects: 27581, done.
remote: Counting objects: 100% (27581/27581), done.
remote: Compressing objects: 100% (27576/27576), done.
remote: Total 27581 (delta 2), reused 27578 (delta 2), pack-reused 0
Receiving objects: 100% (27581/27581), 357.27 MiB | 37.99 MiB/s, done.
Resolving deltas: 100% (2/2), done.
Checking out files: 100% (27566/27566), done.
```

```
!cd /content/deep-learning/NLP
!ls
```

```
deep-learning sample_data
```

```
df=pd.read_json(r'/content/deep-learning/NLP/news_train.json',lines=True)
```

```
df1 = df[['category','headline','short_description']]
```

```
df1["headline"] = df1["headline"].astype(str) + ". " + df1["short_description"]
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/10min.html>  
 """Entry point for launching an IPython kernel.

```
df2=df1[['category','headline']]
```

```
df2['headline'][0]
```

```
'There Were 2 Mass Shootings In Texas Last Week, But Only 1 On TV. She left her husband'
```

```
df2.head()
```

```
category      headline
0      CRIME  There Were 2 Mass Shootings In Texas Last Week...
1  ENTERTAINMENT  Will Smith Joins Diplo And Nicky Jam For The 2...
2  ENTERTAINMENT  Hugh Grant Marries For The First Time At Age 5...
3  ENTERTAINMENT  Jim Carrey Blasts 'Castrato' Adam Schiff And D...
4  ENTERTAINMENT  Julianna Margulies Uses Donald Trump Poop Bags...
```

```
df_trn, df_val = train_test_split(df2, stratify = df2['category'], test_size = 0.1, random_state
```

```
df_trn.shape, df_val.shape
```

```
↳ ((152999, 2), (17000, 2))
```

```
data_lm = TextLMDDataBunch.from_df(train_df = df_trn, valid_df = df_val, path = "",)
```

```
data_clas = TextClasDataBunch.from_df(path = "", train_df = df_trn, valid_df = df_val, vocab=data
```

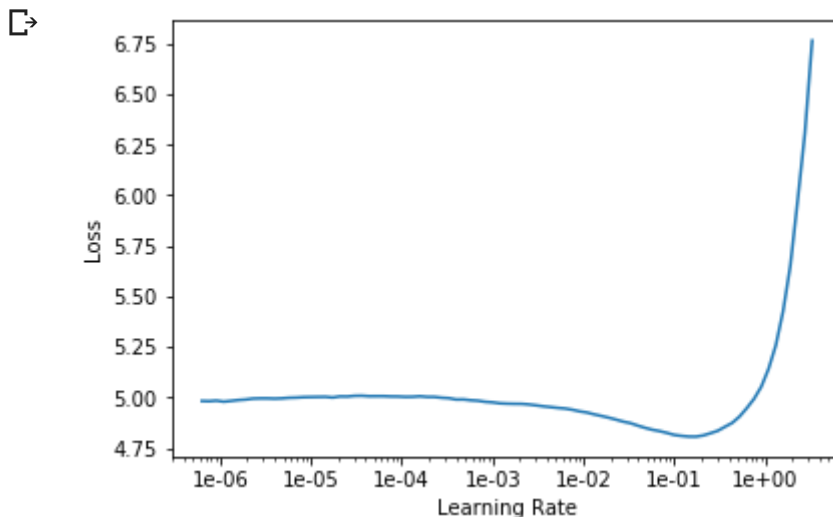
```
#using pre trained model
```

```
learn = language_model_learner(data_lm, pretrained=True, drop_mult=0.7, arch=AWD_LSTM,)
```

```
learn.lr_find()
```

```
↳ LR Finder is complete, type {learner_name}.recorder.plot() to see the graph.
```

```
learn.recorder.plot()
```



```
learn.fit_one_cycle(8, 2e-2, moms=(0.8,0.7)).
```

```
↳
```

epoch	train_loss	valid_loss	accuracy	time
0	3.922114	3.630635	0.396247	08:55
1	4.091992	3.794288	0.383884	09:09
2	4.112128	3.804137	0.385743	09:08
3	4.071332	3.766826	0.388740	09:09
4	3.933746	3.691568	0.393807	09:10
5	3.795717	3.608340	0.400197	09:08

```
learn.save('first')
```

```
learn.fit_one_cycle(8, 2e-2, moms=(0.8,0.7))
```

```
learn.fit_one_cycle(8, 2e-2, moms=(0.8,0.7))
```



epoch	train_loss	valid_loss	accuracy	time
0	3.691448	3.610318	0.400235	09:07
1	4.046556	3.790074	0.383743	09:09
2	4.065940	3.803834	0.386016	09:08
3	4.027109	3.765557	0.388424	09:08
4	3.905745	3.688263	0.393309	09:09
5	3.770785	3.612364	0.400165	09:10
6	3.656035	3.564404	0.403991	09:11
7	3.569078	3.554479	0.405225	09:08

```
learn.save('second')
```

```
learn.fit_one_cycle(8, max_lr=slice(1e-4,1e-2),)
```

```
learn.save('third')
```

```
learn.unfreeze()
learn.fit_one_cycle(5, max_lr=slice(1e-4,1e-2))
```



epoch	train_loss	valid_loss	accuracy	time
0	3.467162	3.433694	0.418205	10:34
1	3.469379	3.355090	0.428539	10:36
2	3.345504	3.278799	0.436857	10:36
3	3.175318	3.230121	0.442727	10:39
4	3.043306	3.227922	0.443971	10:40

```
learn.fit_one_cycle(1, max_lr=slice(1e-4,1e-2))
```

```
↳
```

epoch	train_loss	valid_loss	accuracy	time
0	3.205231	3.232610	0.443766	10:42

```
learn.fit_one_cycle(5, max_lr=slice(1e-4,1e-2))
```

```
↳
```

epoch	train_loss	valid_loss	accuracy	time
0	3.159442	3.267662	0.440093	10:42
1	3.256481	3.271077	0.439858	10:42
2	3.168242	3.230816	0.444812	10:42
3	3.016427	3.202396	0.448468	10:42
4	2.868422	3.208369	0.449251	10:42

```
learn.save_encoder('ft_enc').
```

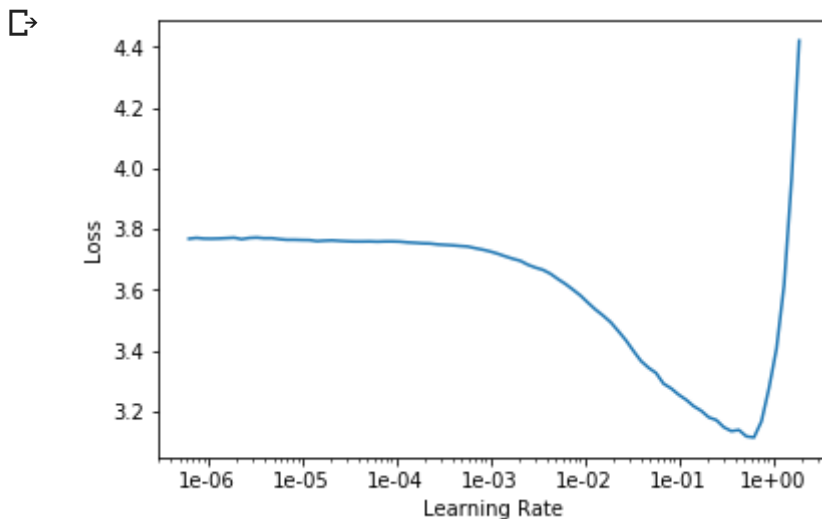
```
learn = text_classifier_learner(data_clas, drop_mult=0.7, arch= AWD_LSTM).
```

```
learn.load_encoder('ft_enc')
```

```
learn.lr_find().
```

```
↳ LR Finder is complete, type {learner_name}.recorder.plot() to see the graph.
```

```
learn.recorder.plot().
```



```
learn.fit_one_cycle(8,max_lr=slice(1e-4,3e-2) , moms=(0.8,0.7)).
```

↗

epoch	train_loss	valid_loss	accuracy	time
0	1.743195	1.398623	0.589176	02:16
1	1.695991	1.372731	0.595176	02:11
2	1.666972	1.366648	0.593176	02:17
3	1.658538	1.329636	0.606294	02:25
4	1.663385	1.295295	0.613824	02:16
5	1.635196	1.272218	0.624471	02:21
6	1.546003	1.249758	0.630000	02:16
7	1.548230	1.241340	0.630941	02:27

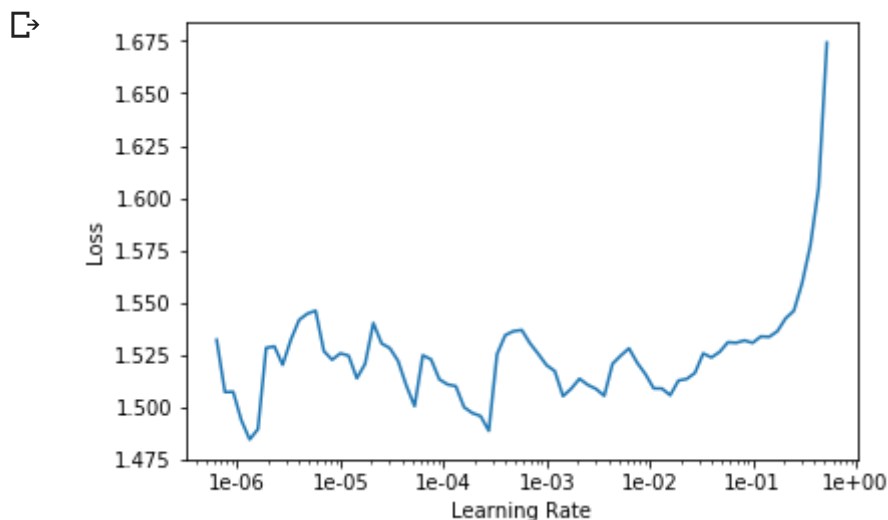
```
learn.save('fourth')
```

```
learn.load('fourth').
```

```
learn.lr_find()
```

↗ LR Finder is complete, type {learner\_name}.recorder.plot() to see the graph.

```
learn.recorder.plot().
```



```
learn.save('fifth').
```

```
learn.fit_one_cycle(8,max_lr=slice(1e-5,3e-4). )
```

↳

epoch	train_loss	valid_loss	accuracy	time
0	1.549968	1.237967	0.632235	02:19
1	1.535224	1.243587	0.631000	02:28
2	1.564070	1.240166	0.631706	02:12
3	1.536250	1.239880	0.631059	02:19
4	1.601330	1.239378	0.631471	02:26
5	1.585539	1.242845	0.630824	02:29
6	1.605733	1.237384	0.631706	02:22
7	1.539147	1.238826	0.631118	02:19

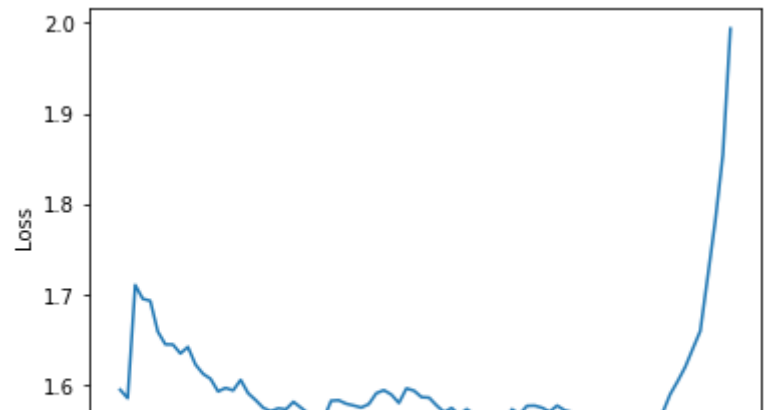
```
learn.save('sixth')
```

```
learn.lr_find().
```

↳ LR Finder is complete, type {learner\_name}.recorder.plot() to see the graph.

```
learn.recorder.plot().
```

↳



```
learn.unfreeze()
learn.fit_one_cycle(8, max_lr=slice(1e-6,5e-3))
```

↗

epoch	train_loss	valid_loss	accuracy	time
0	1.530138	1.211437	0.637471	06:07
1	1.403279	1.158794	0.651353	06:01
2	1.339522	1.121151	0.661000	05:44
3	1.254799	1.085845	0.673294	05:47
4	1.264540	1.064717	0.679353	06:11
5	1.243960	1.050771	0.684765	06:13
6	1.155666	1.048056	0.684647	05:34
7	1.202180	1.052596	0.684824	05:47

```
learn.save('seventh')
```

```
learn.load('seventh').
```

```
dft=pd.read_json(r'/content/deep-learning/NLP/news_test.json',lines=True)
```

```
dft.head()
```

↗

	authors	date	headline	
0	Emily Bennington, Contributor\nAuthor, 'Who Sa...	2012-12-26	Making Sense of a Senseless World	https://www.huffingtonpost.com/er
1	Andrea Wachter, Contributor\nPsychotherapist a...	2012-12-26	Dog Talk	https://www.huffingtonpost.com,
2	CafeMom,	2012-	7 Soothing Ways to	https://www.huffinatonpost.com/er

```
dft1 = dft[['headline', 'short_description']]
```

```
dft1.head()
```

	headline	short_description
0	Making Sense of a Senseless World	I know I speak for many when I say that ev
1	Dog Talk	Helping my clients learn to accept who the
2	7 Soothing Ways to Beat Stress During Pregnancy	If you're pregnant and already feeling stre
3	The Hard Truth About Sharing Your Birthday Wit...	As a kid, my birthday parties weren't alway
4	Optical Illusion And Photobomb Finally Come To...	We assume the man in the background looks s

```
dft1["headline"] = dft1["headline"].astype(str) + ". " + dft1["short_description"];
```

```
[>] /usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/10min.html>  
 """Entry point for launching an IPython kernel.

```
dft2=dft1[['headline']]
```

```
dft2.head()
```

	headline
0	Making Sense of a Senseless World. I know I sp...
1	Dog Talk. Helping my clients learn to accept w...
2	7 Soothing Ways to Beat Stress During Pregnanc...
3	The Hard Truth About Sharing Your Birthday Wit...
4	Optical Illusion And Photobomb Finally Come To...

```
learn.export()
```

```
test = TextList.from_df(dft2, path="").
```

```
data_clas.add_test(test).
```



```
learn = load_learner('/content/', test=test,)
```

```
preds,y = learn.get_preds(ds_type=DatasetType.Test,ordered=True,.)
```

```
preds[:2]
```

```
↳ tensor([[7.9760e-03, 1.1141e-03, 4.9176e-03, 2.6160e-03, 3.3261e-03, 2.0246e-02,
            1.0710e-02, 2.3880e-03, 3.8122e-03, 8.7094e-03, 4.8552e-03, 1.9411e-04,
            8.2586e-03, 9.0683e-04, 7.9221e-03, 1.0391e-03, 7.7044e-03, 1.6521e-03,
            1.7690e-01, 1.0896e-03, 4.0199e-02, 4.6206e-04, 4.7040e-01, 5.9914e-02,
            4.2765e-02, 5.0672e-03, 2.2732e-02, 5.3126e-04, 9.6783e-04, 6.8785e-04,
            2.1071e-03, 3.1334e-04, 2.7189e-03, 1.7827e-03, 9.2445e-03, 6.0844e-04,
            2.8249e-03, 3.2871e-02, 1.3806e-02, 1.3948e-03, 1.2260e-02],
          [1.1523e-03, 1.9402e-04, 1.2032e-03, 1.2216e-02, 3.1955e-04, 4.2079e-02,
            2.7932e-03, 3.7650e-04, 4.0225e-03, 9.4209e-04, 2.4714e-03, 1.0671e-02,
            2.8136e-03, 2.9836e-03, 1.0009e-01, 6.0194e-02, 8.8269e-02, 6.9580e-03,
            7.9310e-02, 4.2922e-04, 1.0736e-03, 1.0250e-03, 1.7955e-01, 5.8358e-02,
            5.6279e-03, 9.5996e-04, 2.2967e-03, 2.9585e-03, 1.7565e-03, 1.0573e-03,
            1.2028e-03, 1.7026e-03, 1.0351e-03, 6.8225e-04, 1.2554e-03, 1.6038e-03,
            1.1604e-02, 2.8741e-01, 1.8081e-02, 3.1631e-04, 9.5016e-04]])
```

```
↳ -----
NameError                                Traceback (most recent call last)
<ipython-input-92-a038622b3b89> in <module>()
----> 1 labelled_preds = [' '.join([learn.data_clas.classes[i] for i,p in enumerate(r

----- 1 frames -----
<ipython-input-92-a038622b3b89> in <listcomp>(.0)
----> 1 labelled_preds = [' '.join([learn.data_clas.classes[i] for i,p in enumerate(r

NameError: name 'thresh' is not defined
```

SEARCH STACK OVERFLOW

```
↳ -----
RuntimeError                                Traceback (most recent call last)
<ipython-input-91-70e099aaf06d> in <module>()
----> 1 labelled_preds.values()

RuntimeError: values is not implemented for type CPUPType
```

SEARCH STACK OVERFLOW

```
labelled_preds
```

```
↳ tensor([22, 37, 22, ..., 28, 28, 28])
```

```
labelled_preds_name = [data_clas.classes[pred] for pred in labelled_preds]
```

```
labelled_preds_name
```



```
[ 'PARENTING',  
  'WELLNESS',  
  'PARENTING',  
  'PARENTING',  
  'CULTURE & ARTS',  
  'TRAVEL',  
  'DIVORCE',  
  'PARENTING',  
  'STYLE & BEAUTY',  
  'STYLE & BEAUTY',  
  'WELLNESS',  
  'PARENTING',  
  'HOME & LIVING',  
  'HOME & LIVING',  
  'HOME & LIVING',  
  'WELLNESS',  
  'FOOD & DRINK',  
  'TRAVEL',  
  'FOOD & DRINK',  
  'TRAVEL',  
  'FOOD & DRINK',  
  'STYLE & BEAUTY',  
  'WELLNESS',  
  'TRAVEL',  
  'TRAVEL',  
  'TRAVEL',  
  'STYLE & BEAUTY',  
  'WELLNESS',  
  'STYLE & BEAUTY',  
  'DIVORCE',  
  'TRAVEL',  
  'TRAVEL',  
  'COMEDY',  
  'COMEDY',  
  'PARENTING',  
  'CRIME',  
  'CRIME',  
  'STYLE & BEAUTY',  
  'SCIENCE',  
  'SCIENCE',  
  'WELLNESS',  
  'QUEER VOICES',  
  'QUEER VOICES',  
  'QUEER VOICES',  
  'PARENTING',  
  'PARENTING',  
  'PARENTING',  
  'WELLNESS',  
  'POLITICS',  
  'POLITICS',  
  'TECH',  
  'BUSINESS',  
  'BUSINESS',  
  'TECH',  
  'TECH',  
  'TECH',  
  'TECH',  
  'POLITICS',  
  'ENTERTAINMENT',  
  'RELIGION',  
  'PARENTING',
```

'ENTERTAINMENT',  
 'WELLNESS',  
 'ENVIRONMENT',  
 'ENVIRONMENT',  
 'GREEN',  
 'ENVIRONMENT',  
 'PARENTING',  
 'ENVIRONMENT',  
 'CULTURE & ARTS',  
 'CULTURE & ARTS',  
 'QUEER VOICES',  
 'CULTURE & ARTS',  
 'BLACK VOICES',  
 'FIFTY',  
 'SPORTS',  
 'SPORTS',  
 'STYLE & BEAUTY',  
 'PARENTING',  
 'FOOD & DRINK',  
 'HOME & LIVING',  
 'WELLNESS',  
 'TRAVEL',  
 'WELLNESS',  
 'WELLNESS',  
 'TRAVEL',  
 'ARTS & CULTURE',  
 'HOME & LIVING',  
 'PARENTING',  
 'TRAVEL',  
 'TRAVEL',  
 'IMPACT',  
 'WELLNESS',  
 'STYLE',  
 'STYLE & BEAUTY',  
 'STYLE & BEAUTY',  
 'STYLE & BEAUTY',  
 'FOOD & DRINK',  
 'HOME & LIVING',  
 'FOOD & DRINK',  
 'DIVORCE',  
 'RELIGION',  
 'WEDDINGS',  
 'POLITICS',  
 'STYLE & BEAUTY',  
 'FOOD & DRINK',  
 'PARENTING',  
 'FOOD & DRINK',  
 'WELLNESS',  
 'FOOD & DRINK',  
 'WELLNESS',  
 'TRAVEL',  
 'DIVORCE',  
 'HOME & LIVING',  
 'TRAVEL',  
 'WELLNESS',  
 'WELLNESS',  
 'HOME & LIVING',  
 'PARENTING',  
 'TRAVEL',  
 'TRAVEL',  
 'FOOD & DRINK',  
 'POLITICS'

'HEALTHY LIVING',  
'COMEDY',  
'POLITICS',  
'CRIME',  
'CRIME',  
'CRIME',  
'FIFTY',  
'DIVORCE',  
'POLITICS',  
'STYLE & BEAUTY',  
'SCIENCE',  
'SCIENCE',  
'SCIENCE',  
'SCIENCE',  
'ENVIRONMENT',  
'COMEDY',  
'SCIENCE',  
'QUEER VOICES',  
'QUEER VOICES',  
'QUEER VOICES',  
'PARENTING',  
'WORLDPOST',  
'IMPACT',  
'HOME & LIVING',  
'HOME & LIVING',  
'BUSINESS',  
'BUSINESS',  
'BUSINESS',  
'BUSINESS',  
'BUSINESS',  
'BUSINESS',  
'BUSINESS',  
'BUSINESS',  
'POLITICS',  
'POLITICS',  
'IMPACT',  
'BUSINESS',  
'BUSINESS',  
'HOME & LIVING',  
'BUSINESS',  
'BUSINESS',  
'BUSINESS',  
'WELLNESS',  
'TECH',  
'TECH',  
'WELLNESS',  
'TECH',  
'WELLNESS',  
'TECH',  
'PARENTING',  
'HOME & LIVING',  
'WELLNESS',  
'PARENTING',  
'STYLE & BEAUTY',  
'FOOD & DRINK',  
'PARENTING',  
'HOME & LIVING',  
'STYLE & BEAUTY',  
'STYLE & BEAUTY',  
'PARENTING',  
'PARENTING',  
'HOME & LIVING',

'FOOD & DRINK',  
'PARENTING',  
'WELLNESS',  
'WELLNESS',  
'TRAVEL',  
'TRAVEL',  
'HOME & LIVING',  
'STYLE & BEAUTY',  
'TRAVEL',  
'TRAVEL',  
'STYLE & BEAUTY',  
'WEIRD NEWS',  
'HOME & LIVING',  
'WELLNESS',  
'WELLNESS',  
'TRAVEL',  
'TRAVEL',  
'WELLNESS',  
'WELLNESS',  
'WELLNESS',  
'PARENTING',  
'WELLNESS',  
'FOOD & DRINK',  
'WELLNESS',  
'WELLNESS',  
'TRAVEL',  
'STYLE & BEAUTY',  
'TRAVEL',  
'WELLNESS',  
'WELLNESS',  
'WELLNESS',  
'STYLE & BEAUTY',  
'ENVIRONMENT',  
'STYLE & BEAUTY',  
'COMEDY',  
'RELIGION',  
'COMEDY',  
'COMEDY',  
'COMEDY',  
'FIFTY',  
'HEALTHY LIVING',  
'POLITICS',  
'COMEDY',  
'POLITICS',  
'PARENTING',  
'POLITICS',  
'POLITICS',  
'TECH',  
'WEDDINGS',  
'STYLE & BEAUTY',  
'STYLE & BEAUTY',  
'WELLNESS',  
'SCIENCE',  
'BUSINESS',  
'QUEER VOICES',  
'COMEDY',  
'QUEER VOICES',  
'POLITICS',  
'QUEER VOICES',  
'QUEER VOICES',  
'CRIME',  
'IMPACT',  
'IMPACT'.