



دانشکده علوم ریاضی
گروه علوم کامپیوتر

گزارش تمرین ۴

نگارش

ریحانه داورزنی

استاد

دکتر سعیدرضا خردپیشه

تیر ۱۴۰۰

فهرست مطالب

۳	مقدمه	۱
۴	Preprocessing	۲
۴	Lower case	۱.۲
۴	contractions	۲.۲
۴	Remove URLs	۳.۲
۵	Remove HTML tags	۴.۲
۵	Remove Non-ASCII	۵.۲
۵	Remove special characters	۶.۲
۶	Remove punctuations	۷.۲
۶	Tokenization	۸.۲
۷	Remove Stop Words	۹.۲
۷	Stemming	۱۰.۲
۸	Part of Speech Tagging	۱۱.۲
۹	Lemmatization	۱۲.۲
۱۰	Text to vector	۳
۱۲	Classify	۴
۱۴	Transformer	۵

۱ مقدمه

شبکه عصبی بازگشتی (RNN) که به آن شبکه عصبی مکرر نیز گفته می شود، نوعی از شبکه عصبی مصنوعی است که در تشخیص گفتار، پردازش زبان طبیعی (NLP) و همچنین در پردازش داده های ترتیبی^۱ استفاده می شود. بسیاری از شبکه های عمیق مانند CNN شبکه های Feed Forward هستند یعنی سیگنال در این شبکه ها فقط در یک جهت از لایه ورودی، به لایه های مخفی و سپس به لایه خروجی حرکت می کند و داده های قبلی به حافظه سپرده نمی شوند. اما شبکه های عصبی بازگشتی (RNN) یک لایه بازخورد دارند که در آن خروجی شبکه به همراه ورودی بعدی، به شبکه بازگردانده می شود. RNN می تواند به علت داشتن حافظه داخلی، ورودی قبلی خود را به خاطر بسپارد و از این حافظه برای پردازش دنباله ایی از ورودی ها استفاده کند. به بیان ساده، شبکه های عصبی بازگشتی شامل یک حلقه بازگشتی هستند که موجب می شود اطلاعاتی را که از لحظات قبلی بدست آورده ایم از بین نروند و در شبکه باقی بمانند. کاربردهای شبکه عصبی بازگشتی:

شرح نویسی عکس^۲، پیش بینی سری های زمانی^۳، پردازش زبان طبیعی^۴، ترجمه^۵.

شبکه های LSTM نوع خاصی از شبکه های عصبی بازگشتی هستند که توانایی یادگیری وابستگی های بلندمدت را دارند. این شبکه ها برای اولین بار توسط Hochreiter و Schmidhuber در سال ۱۹۹۷ در معرفی شدند. در حقیقت هدف از طراحی شبکه های LSTM، حل کردن مشکل وابستگی بلندمدت بود.

^۱Sequential data

^۲Image Captioning

^۳Time Series Prediction

^۴Natural Language Processing

^۵Machine Translation

۲ Preprocessing

در ابتدا پیش پردازش های لازم را روی دیتاست انجام می دهیم که به ترتیب زیر می باشد:

۱.۲ Lower case

تمام کلمات موجود در دیتاست را به حروف کوچک تبدیل می کنیم به صورت زیر :

```
1 train_data["text_clean"] = train_data["text"].apply(lambda x: x.lower())
2 display(train_data.head())
```

	id	keyword	location	text	target	text_clean
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	our deeds are the reason of this #earthquake m...
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask. canada

۲.۲ contractions

ما از پکیج contractions برای گسترش انقباض به زبان انگلیسی استفاده می کنیم
مانند این که

we shouldn't've -> we should not have.

و به صورت زیر انجام می دهیم:

```
1 train_data["text_clean"] = train_data["text_clean"].apply(lambda x: contractions.fix(x))
2
3 #check
4 print(train_data["text"][67])
5 print(train_data["text_clean"][67])
```

```
'I can't have kids cuz I got in a bicycle accident & split my testicles. it's impossible for me to have kids' MICHAEL YOU ARE THE FATHER
'i cannot have kids cuz i got in a bicycle accident & split my testicles. it is impossible for me to have kids' michael you are the father
```

۳.۲ Remove URLs

داده های متنی می توانند شامل کاراکترها یا علائم نگارشی غیر ضروری مختلف مانند URL ها ، تگ های HTML ، کاراکترهای غیر ASCII باشند. به همین دلیل می توان این هارا نیز حذف کرد.

URL هارا به صورت زیر از متن حذف می کنیم:

```

1 def remove_URL(text):
2     return re.sub(r"https?://\S+|www\.\S+", "", text)

1 # remove urls from the text
2 train_data["text_clean"] = train_data["text"].apply(lambda x: remove_URL(x))
3
4 #check
5 print(train_data["text"][31])
6 print(train_data["text_clean"][31])

@bbcmd Wholesale Markets ablaze http://t.co/1HYXEOHY6C
@bbcmd wholesale markets ablaze

```

Remove HTML tags ۴.۲

تگ‌های HTML را به صورت زیر حذف می‌کنیم:

```

1 def remove_html(text):
2     html = re.compile(r"<.*?>|&[a-z0-9]+|#[0-9]{1,6}|#x[0-9a-f]{1,6}");
3     return re.sub(html, "", text)

1 # remove html from the text
2 train_data["text_clean"] = train_data["text"].apply(lambda x: remove_html(x))
3
4 #check
5 print(train_data["text"][62])
6 print(train_data["text_clean"][62])

Rene Ablaze & Jacinta - Secret 2k13 (Fallen Skies Edit) - Mar 30 2013 https://t.co/7MLMsUzV1Z
rene ablaze jacinta - secret 2k13 (fallen skies edit) - mar 30 2013

```

Remove Non-ASCII ۵.۲

کاراکترهای غیر ASCII را به صورت زیر حذف می‌کنیم:

```

1 train_data["text_clean"] = train_data["text"].apply(lambda x: remove_non_ascii(x))
2
3 # double
4 print(train_data["text"][38])
5 print(train_data["text_clean"][38])

Barbados #Bridgetown JAMAICA 🚗🚗 Two cars set ablaze: SANTA CRUZ 🚒🚒 Head of the St Elizabeth Police Superintende...
barbados #Bridgetown jamaica two cars set ablaze: santa cruz head of the st elizabeth police superintende...

```

Remove special characters ۶.۲

کاراکترهای خاص می‌توانند نمادها ، ایموجی‌ها و کاراکترهای گرافیکی باشند. برای حذف این کاراکترها به صورت زیر عمل می‌کنیم:

```

1 def remove_special_characters(text):
2     emoji_pattern = re.compile(
3         '['
4         u'\U0001F600-\U0001F64F' # emoticons
5         u'\U0001F300-\U0001F5FF' # symbols & pictographs
6         u'\U0001F680-\U0001F6FF' # transport & map symbols
7         u'\U0001F1E0-\U0001F1FF' # flags (iOS)
8         u'\U00002702-\U000027B0'
9         u'\U000024C2-\U0001F251'
10        ']+',
11        flags=re.UNICODE)
12    return emoji_pattern.sub(r'', text)

1 train_data["text_clean"] = train_data["text_clean"].apply(lambda x: remove_special_characters(x))

```

Remove punctuations ۷.۲

علائم نگارشی را به صورت زیر حذف می‌کنیم:

```

1 def remove_punct(text):
2     # return re.sub(r'[! "%&\'()*+,-./:;=#@?[\^_`{|}~]+', "", text)
3     return text.translate(str.maketrans('', '', string.punctuation))

1 # remove punctuations from the text
2 train_data["text_clean"] = train_data["text_clean"].apply(lambda x: remove_punct(x))
3
4 #check
5 print(train_data["text"][5])
6 print(train_data["text_clean"][5])

#RockyFire Update => California Hwy. 20 closed in both directions due to Lake County fire - #CAfire #wildfires
rockyfire update  california hwy 20 closed in both directions due to lake county fire  cafire wildfires

```

Tokenization ۸.۲

توکن سازی یک تکنیک رایج است که یک جمله را به توکن تقسیم می‌کند ، جایی که توکن می‌تواند کاراکترها ، کلمات ، عبارات ، نمادها یا سایر عناصر معنی دار باشد. با تقسیم جملات به قطعات کوچکتر ، این امر به بررسی راحت‌تر کلمات در یک جمله و همچنین مراحل بعدی در خط لوله NLP ، کمک می‌کند. به صورت زیر به توکن تبدیل می‌کنیم:

```
from nltk.tokenize import word_tokenize

train_data['tokenized'] = train_data['text_clean'].apply(word_tokenize)
train_data.head()
```

id	keyword	location	text	target	text_clean	tokenized
1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	our deeds are the reason of this earthquake ma...	[our, deeds, are, the, reason, of, this, earth...
4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask canada	[forest, fire, near, la, ronge, sask, canada]
5	NaN	NaN	All residents asked to 'shelter in place' are ...	1	all residents asked to shelter in place are be...	[all, residents, asked, to, shelter, in, place...
6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1	13000 people receive wildfires evacuation orde...	[13000, people, receive, wildfires, evacuation...
7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1	just got sent this photo from ruby alaska as s...	[just, got, sent, this, photo, from, ruby, ala...

Remove Stop Words ۹.۲

کلمات Stop کلمات متداول در هر زبانی هستند که با فرکانس بالا وجود دارند اما اطلاعات معنی داری را برای کل جمله ارائه نمی دهند. به عنوان مثال ”a ”، ” about ”، ” above ”، ” after ”، ” afterward ”، ” again ”، ... را می توان کلمات Stop دانست. به طور سنتی ، ما می توانیم همه آنها را در مرحله پیش پردازش متن حذف کنیم.

به صورت زیر آن ها را حذف می کنیم:

```
1 # Download stopwords
2 nltk.download("stopwords")
3 from nltk.corpus import stopwords
4 stop = set(stopwords.words('english'))

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
```

```
1 train_data['stopwords_removed'] = train_data['tokenized'].apply(lambda x: [word for word in x if word not in stop])
2 train_data.head()
```

id	keyword	location	text	target	text_clean	tokenized	stopwords_removed
0	1	NaN	Our Deeds are the Reason of this #earthquake M...	1	our deeds are the reason of this earthquake ma...	[our, deeds, are, the, reason, of, this, earth...	[deeds, reason, earthquake, may, allah, forgiv...
1	4	NaN	Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask canada	[forest, fire, near, la, ronge, sask, canada]	[forest, fire, near, la, ronge, sask, canada]
2	5	NaN	All residents asked to 'shelter in place' are ...	1	all residents asked to shelter in place are be...	[all, residents, asked, to, shelter, in, place...	[residents, asked, shelter, place, notified, o...
3	6	NaN	13,000 people receive #wildfires evacuation or...	1	13000 people receive wildfires evacuation orde...	[13000, people, receive, wildfires, evacuation...	[13000, people, receive, wildfires, evacuation...
4	7	NaN	Just got sent this photo from Ruby #Alaska as ...	1	just got sent this photo from ruby alaska as s...	[just, got, sent, this, photo, from, ruby, ala...	[got, sent, photo, ruby, alaska, smoke, wildf...

Stemming ۱۰.۲

Stemming یک فرآیند استخراج یک کلمه ریشه ای است - شناسایی یک ساقه مشترک در میان اشکال مختلف (به عنوان مثال اسم مفرد و جمع) از یک کلمه ، به

عنوان مثال ، کلمات ”gardening” ، ”gardener” یا ”gardens” از یک ساقه مشترک هستند .

Stemming پسوندهای کلمات را حذف می‌کند تا کلمات با معنی مشابه را باهم ادغام کند.

```
from nltk.stem import LancasterStemmer

def lancaster_stemmer(text):
    stemmer = nltk.LancasterStemmer()
    stems = [stemmer.stem(i) for i in text]
    return stems

train_data['lancaster_stemmer'] = train_data['stopwords_removed'].apply(lambda x: lancaster_stemmer(x))
train_data.head()
```

id	keyword	location	text	target	text_clean	tokenized	stopwords_removed	lancaster_stemmer
1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	our deeds are the reason of this earthquake ma...	[our, deeds, are, the, reason, of, this, earth...	[deeds, reason, earthquake, may, allah, forgiv...	[dee, reason, earthquak, may, allah, forg, us]
4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask canada	[forest, fire, near, la, ronge, sask, canada]	[forest, fire, near, la, ronge, sask, canada]	[forest, fir, near, la, rong, sask, canad]
5	NaN	NaN	All residents asked to 'shelter in place' are ...	1	all residents asked to shelter in place are be...	[all, residents, asked, to, shelter, in, place...	[residents, asked, shelter, place, notified, o...	[resid, ask, shelt, plac, not, off, evacu, she...

۱۱.۲ Part of Speech Tagging

یرچسب گذاری POS بخشی از گفتار (اسم ، فعل ، صفت و غیره) هر کلمه را در متن متمایز می‌کند. این مرحله برای بسیاری از برنامه های NLP مهم است ، زیرا با شناسایی POS یک کلمه ، می توانیم معنای متنی آن را استنباط کنیم.

```
wordnet_map = [{"N":wordnet.NOUN,
                "V":wordnet.VERB,
                "J":wordnet.ADJ,
                "R":wordnet.ADV
                }]

train_sents = brown.tagged_sents(categories='news')
t0 = nltk.DefaultTagger('NN')
t1 = nltk.UnigramTagger(train_sents, backoff=t0)
t2 = nltk.BigramTagger(train_sents, backoff=t1)
def pos_tag_wordnet(text, pos_tag_type="pos_tag"):
    pos_tagged_text = t2.tag(text)
    # map the pos tagging output with wordnet output
    pos_tagged_text = [(word, wordnet_map.get(pos_tag[0])) if pos_tag[0] in wordnet_map.keys() else (word, wordnet.NOUN) for (word, pos_tag) in pos
    return pos_tagged_text

train_data['combined_postag_wnet'] = train_data['stopwords_removed'].apply(lambda x: pos_tag_wordnet(x))
train_data.head()
```

id	keyword	location	text	target	text_clean	tokenized	stopwords_removed	lancaster_stemmer	lemmatize_word_wo_pos	combined_postag_wnet
1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	our deeds are the reason of this earthquake ma...	[our, deeds, are, the, reason, of, this, earth...	[deeds, reason, earthquake, may, allah, forgiv...	[dee, reason, earthquak, may, allah, forg, us]	[deed, reason, earthquake, may, allah, forgive...	[(deeds, n), (reason, n), (earthquake, n), (ma...

Lemmatization ۱۲.۲

Lemmatization وظیفه تعیین این که دو کلمه ، علیرغم اختلاف سطح ، دارای ریشه یکسان هستند را دارد.

به عنوان مثال ، ”good” ، ”better” یا ”best” به good تبدیل می شود و فعل ”gardening” باید به ”garden” تبدیل شود ، در حالی که ”garden” و ”gardener” هر دو lemma های مختلف هستند. در این بخش در مورد lemmatize کردن بدون مثالهای برچسب گذاری POS و برچسب گذاری POS می پردازیم.

```
from nltk.stem import WordNetLemmatizer

def lemmatize_word(text):
    lemmatizer = WordNetLemmatizer()
    lemma = [lemmatizer.lemmatize(word, tag) for word, tag in text]
    return lemma
```

Lemmatization without POS Tagging •

که به صورت زیر می باشد:

```
lemmatizer = WordNetLemmatizer()

train_data['lemmatize_word_wo_pos'] = train_data['stopwords_removed'].apply(lambda x: [lemmatizer.lemmatize(word) for word in x])
train_data['lemmatize_word_wo_pos'] = train_data['lemmatize_word_wo_pos'].apply(lambda x: [word for word in x if word not in stop])
train_data.head()
```

id	keyword	location	text	target	text_clean	tokenized	stopwords_removed	lancaster_stemmer	combined_postag_wnet	lemmatize_word_wo_pos
1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	our deeds are the reason of this earthquake ma...	[our, deeds, are, the, reason, of, this, earth...	[deeds, reason, earthquake, may, allah, forgiv...	[dee, reason, earthquak, may, allah, forg, us]	[(deeds, n), (reason, n), (earthquake, n), (ma...	[deed, reason, earthquake, may, allah, forgive...
4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask canada	[forest, fire, near, la, ronge, sask, canada]	[forest, fire, near, la, ronge, sask, canada]	[forest, fir, near, la, rong, sask, canad]	[(forest, n), (fire, n), (near, n), (la, n), (...]	[forest, fire, near, la, ronge, sask, canada]

Lemmatization with POS Tagging •

که به صورت زیر می باشد:

```

1 lemmatizer = WordNetLemmatizer()
2
3 train_data['lemmatize_word_w_pos'] = train_data['combined_postag_wnet'].apply(lambda x: lemmatize_word(x))
4 train_data['lemmatize_word_w_pos'] = train_data['lemmatize_word_w_pos'].apply(lambda x: [word for word in x if word not in stop]) # double check to
5 train_data['lemmatize_text'] = [' '.join(map(str, l)) for l in train_data['lemmatize_word_w_pos']] # join back to text
6
7 train_data.head()

```

text	target	text_clean	tokenized	stopwords_removed	lancaster_stemmer	combined_postag_wnet	lemmatize_word_wo_pos	lemmatize_word_w_pos	lemmatize_text
Our Deeds are the Reason of this earthquake M...	1	our deeds are the reason of this earthquake ma...	[our, deeds, are, the, reason, of, this, earth...	(deeds, reason, earthquake, may, allah, forgiv...	[dee, reason, earthquak, may, allah, forg, us]	[(deeds, n), (reason, n), (earthquake, n), (ma...	[deed, reason, earthquake, may, allah, forgive...	[deed, reason, earthquake, may, allah, forgive...	deed reason earthquake may allah forgive u
Forest fire near La Ronge Sask. Canada	1	forest fire near la ronge sask canada	[forest, fire, near, la, ronge, sask, canada]	[forest, fire, near, la, ronge, sask, canada]	[forest, fir, near, la, rong, sask, canad]	[(forest, n), (fire, n), (near, n), (la, n), (...]	[forest, fire, near, la, ronge, sask, canada]	[forest, fire, near, la, ronge, sask, canada]	forest fire near la ronge sask canada

یک نمونه که خروجی هر دو حالت را نشان دهد به صورت زیر می‌باشد:

```

1 print(train_data["text"][8])
2 print(train_data["combined_postag_wnet"][8])
3 print(train_data["lemmatize_word_wo_pos"][8])
4 print(train_data["lemmatize_word_w_pos"][8])

```

There's an emergency evacuation happening now in the building across the street
 [(['emergency', 'n'), ('evacuation', 'n'), ('happening', 'v'), ('building', 'n'), ('across', 'n'), ('street', 'n')]
 ['emergency', 'evacuation', 'happening', 'building', 'across', 'street']
 ['emergency', 'evacuation', 'happen', 'building', 'across', 'street']

Text to vector ۳

در این بخش، هر پیام را که به بردار تبدیل می‌کنیم که مدل‌های یادگیری ماشین می‌توانند آن را درک کنند. به این معنی که ما به هر کلمه منحصر به فرد یک شاخص (عدد) می‌دهیم و آن را در دیکشنری *word2idx* ذخیره می‌کنیم و همچنین یک دیکشنری جدید تعبیه می‌کنیم که این اعداد را به ضریب تعبیه شده از embedding های glove ترسیم می‌کند. اگر این کلمه در تعبیه glove وجود نداشته باشد، ضرایب تصادفی از همان بعد را به آنها می‌دهیم.

: glove embeddings

```

1 embeddings_index = {}
2 with open('/content/files2/glove.6B.100d.txt', 'r', encoding = 'utf8') as f:
3     for line in f:
4         values = line.split()
5         word = values[0]
6         coefs = np.array(values[1:]).astype(np.float)
7         embeddings_index[word] = coefs
8
9 print('Found %s word vectors.' % len(embeddings_index))

```

Found 400000 word vectors.

که به صورت زیر اعمال می‌شود:

```
train_X_list = []
index = 1

embed_keys = embeddings_index.keys()
for x in X_train['lemmatize_text']:
    list1 = x.split(' ')
    new_list = []
    for i in list1:
        if((i in embed_keys) and (i not in word2idx.keys())):
            new_embedding_index[index] = embeddings_index[i]
            word2idx[i] = index
            new_list.append(index)
            index=index+1

        elif(i not in word2idx.keys()):
            new_embedding_index[index] = np.random.normal(scale=0.4, size=(50,)).astype(np.float)
            word2idx[i] = index
            new_list.append(index)
            index=index+1

    else:
        new_list.append(word2idx[i])

    train_X_list.append(new_list)
```

با توجه به طول جملات که متفاوت از یکدیگر است ، یک padding بر روی آنها به صورت زیر اعمال می‌کنیم:

```
def pad_features(reviews_int, seq_length):
    features = np.zeros((len(reviews_int), seq_length), dtype = int)
    for i, review in enumerate(reviews_int):
        review_len = len(review)

        if review_len <= seq_length:
            zeroes = list(np.zeros(seq_length-review_len))
            new = zeroes+review

        elif review_len > seq_length:
            new = review[0:seq_length]

        features[i,:] = np.array(new)

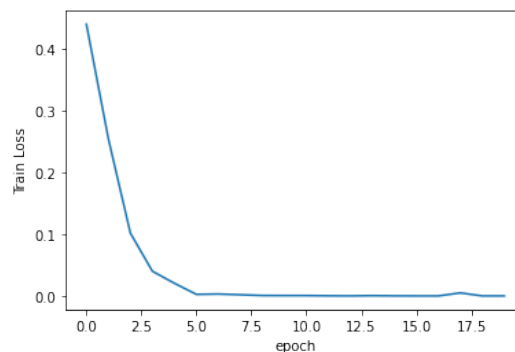
    return features
```

۴ Classify

مدل LSTM را به صورت زیر پیاده‌سازی می‌کنیم:

```
BiLSTM(  
    (embedding): Embedding(17703, 100)  
    (lstm): LSTM(100, 200, num_layers=2, batch_first=True, dropout=0.5, bidirectional=True)  
    (dropout): Dropout(p=0.3, inplace=False)  
    (fullyconnect1): Linear(in_features=200, out_features=50, bias=True)  
    (fullyconnect2): Linear(in_features=50, out_features=1, bias=True)  
    (sig): Sigmoid()  
)
```

مدل را با learningRate برابر 0.001 ، criterion برابر BCELoss ، -opti- mizer برابر Adam و تعداد epoch برابر 20 آموزش می‌دهیم. مقدار loss در حالت train به صورت زیر می‌باشد:



در ابتدا و قبل از آموزش مدل، ما مجموعه train را به دو زیرمجموعه train و validation با نسبت 80 به 20 تقسیم بندی کرده‌ایم. حال اگر روی مدل را روی مجموعه Validation اجرا کنیم نتایج زیر به دست می‌آید:

```
[[680 203]  
 [189 447]]  
Accuracy: 0.7419354838709677
```

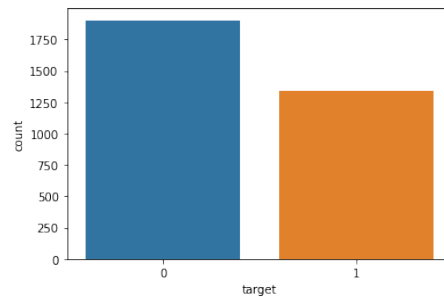
و خروجی آن به صورت زیر خواهد شد :

	actual	predict
5682	0	0
5138	0	0
7387	0	0
5894	1	1
2527	0	0
...
5437	1	1
2452	1	1
5329	0	0
1680	0	1
586	1	1

حال اگر روی مجموعه test، مدل را اجرا کنیم، خروجی به صورت زیر خواهد شد:

	id	target
0	0	1
1	2	0
2	3	1
3	9	1
4	11	1
...
3242	10792	0
3243	10796	1
3244	10797	0
3245	10801	0
3246	10804	0

و نسبت طبقه‌بندی جواب‌ها به صورت زیر خواهد بود:



Transformer ۵

در این بخش از transformer برای آموزش مدل استفاده می‌کنیم که به صورت زیر تعریف می‌شود:

```
from simpletransformers.classification import ClassificationModel

# Create a TransformerModel
model = ClassificationModel('bert', 'bert-base-cased', num_labels=2,
                           args={'reprocess_input_data': True, 'overwrite_output_dir': True},
                           use_cuda=False)
```

مدل ذکر شده را با مجموعه داده train آموزش می‌دهیم که میزان خطا در آن برابر 0.4368 است. سپس مدل را روی مجموعه داده validation اجرا کرده که نتایج زیر به دست می‌آید:

```
{'mcc': 0.6533225478972492, 'tp': 27, 'tn': 36, 'fp': 6, 'fn': 7,
 'auroc': 0.9292717086834734, 'auprc': 0.9226137022189955,
 'eval_loss': 0.3497443050146103}
```

	precision	recall	f1-score	support
fake	0.84	0.86	0.85	42
real	0.82	0.79	0.81	34
accuracy			0.83	76
macro avg	0.83	0.83	0.83	76
weighted avg	0.83	0.83	0.83	76