In [27]:

```
import os
import numpy as np
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Embedding, Flatten, Dense
from keras.utils.np_utils import to_categorical
```

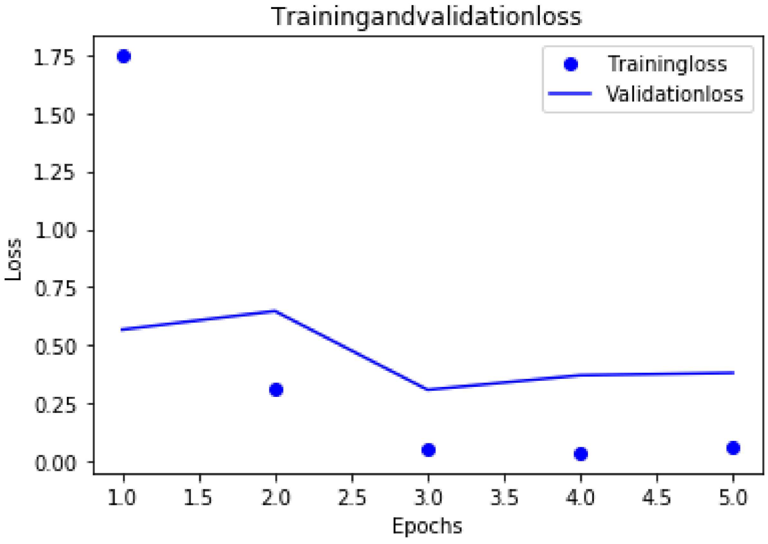
In [59]:

```
data_dir = 'D:/Deeplearning/datasets/bbc'
labels = []
texts = []
label_count = 0
for label_type in ['business', 'entertainment', 'politics', 'sport', 'tech']:
    dir_name = os.path.join(data_dir, label_type)
    for fname in os.listdir(dir_name):
        f = open(os.path.join(dir_name, fname), encoding="utf8", errors='ignor
e')
        texts.append(f.read())
        f.close()
        labels.append(label_count)
    label_count = label_count + 1

maxlen = 375 # Cut off after 375 words in tokenizer
training_samples = 1725
validation_samples = 500
max_words = 10000 # Size of dictionary for our problem
tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(texts)
sequences = tokenizer.texts_to_sequences(texts)

word_index = tokenizer.word_index
data = pad_sequences(sequences, maxlen=maxlen)
labels = np.asarray(labels)

# Randomly get training and validation samples
indices = np.arange(data.shape[0])
np.random.shuffle(indices)
data = data[indices]
labels = labels[indices]
x_train = data[:training_samples]
y_train = labels[:training_samples]
x_val = data[training_samples: training_samples + validation_samples]
y_val = labels[training_samples: training_samples + validation_samples]

y_train = to_categorical(y_train)
y_val = to_categorical(y_val)

# Get pre-trained embedding vectors
# Each vector has a size of 300
glove_dir = 'D:/Deeplearning/datasets/bbc/glove/'
embeddings_index = {}
f = open(os.path.join(glove_dir, 'glove.6B.300d.txt'),encoding='utf8')
for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[word] = coefs
f.close()

# Embedding dimension is the same as our embedding vector size
embedding_dim = 300
embedding_matrix = np.zeros((max_words, embedding_dim))
for word, i in word_index.items():
    if i < max_words:
        embedding_vector = embeddings_index.get(word)
        if embedding_vector is not None:
            embedding_matrix[i] = embedding_vector

# Start creating the model
model = Sequential()
model.add(Embedding(max_words, embedding_dim, input_length=maxlen))
model.add(Flatten())
model.add(Dense(32, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(5, activation='softmax'))
model.summary()

# Set weights of the embedding layer from our pretrained embedding matrix
model.layers[0].set_weights([embedding_matrix])
model.layers[0].trainable = False

# Compile and start training for 20 epochs
model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=[
'acc'])
history = model.fit(x_train, y_train, epochs=5, batch_size=32, validation_data
=(x_val, y_val), shuffle=True)
model.save_weights('bbc_news_classfication_model.h5')
```

```
Model: "sequential_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_4 (Embedding)      (None, 375, 300)          3000000   
_________________________________________________________________
flatten_4 (Flatten)          (None, 112500)            0         
_________________________________________________________________
dense_21 (Dense)             (None, 32)                3600032   
_________________________________________________________________
dense_22 (Dense)             (None, 16)                528       
_________________________________________________________________
dense_23 (Dense)             (None, 5)                 85        
=================================================================
Total params: 6,600,645
Trainable params: 6,600,645
Non-trainable params: 0
_________________________________________________________________
Train on 1725 samples, validate on 500 samples
Epoch 1/5
1725/1725 [==============================] - 6s 3ms/step - loss: 1.7467 - ac
c: 0.6319 - val_loss: 0.5676 - val_acc: 0.8160
Epoch 2/5
1725/1725 [==============================] - 6s 3ms/step - loss: 0.3121 - ac
c: 0.9119 - val_loss: 0.6472 - val_acc: 0.7640
Epoch 3/5
1725/1725 [==============================] - 6s 4ms/step - loss: 0.0518 - ac
c: 0.9843 - val_loss: 0.3082 - val_acc: 0.8840
Epoch 4/5
1725/1725 [==============================] - 6s 3ms/step - loss: 0.0363 - ac
c: 0.9901 - val_loss: 0.3705 - val_acc: 0.9000
Epoch 5/5
1725/1725 [==============================] - 6s 3ms/step - loss: 0.0634 - ac
c: 0.9872 - val_loss: 0.3812 - val_acc: 0.9080
```

In [60]:

```python
import matplotlib.pyplot as plt
history_dict=history.history
loss_values=history_dict['loss']
acc = history_dict['acc']
val_loss_values=history_dict['val_loss']
epochs=range(1,len(acc)+1)
plt.plot(epochs,loss_values,'bo',label='Trainingloss')
plt.plot(epochs,val_loss_values,'b',label='Validationloss')
plt.title('Trainingandvalidationloss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

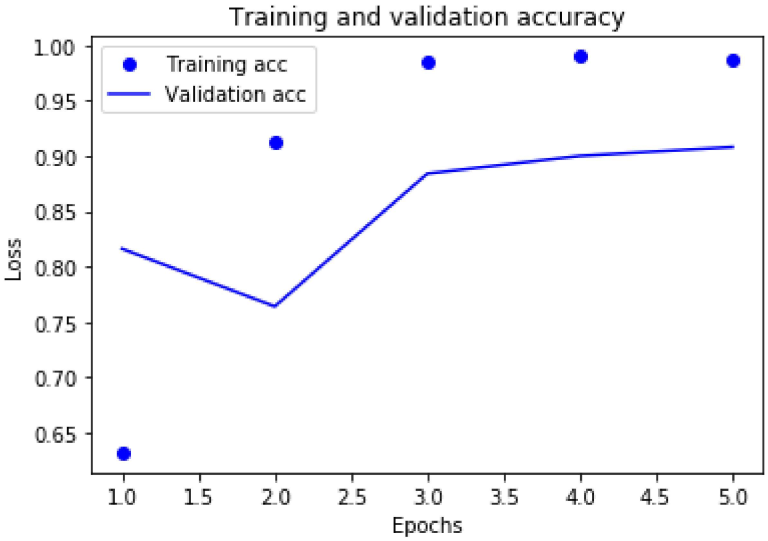


```
In [61]: texts[0]
```

```
Out[61]: 'Ad sales boost Time Warner profit\n\nQuarterly profits at US media giant Tim
         eWarner jumped 76% to $1.13bn (£600m) for the three months to December, from
         $639m year-earlier.\n\nThe firm, which is now one of the biggest investors in
         Google, benefited from sales of high-speed internet connections and higher ad
         vert sales. TimeWarner said fourth quarter sales rose 2% to $11.1bn from $10.
         9bn. Its profits were buoyed by one-off gains which offset a profit dip at Wa
         rner Bros, and less users for AOL.\n\nTime Warner said on Friday that it now
         owns 8% of search-engine Google. But its own internet business, AOL, had has
         mixed fortunes. It lost 464,000 subscribers in the fourth quarter profits wer
         e lower than in the preceding three quarters. However, the company said AOL
         \'s underlying profit before exceptional items rose 8% on the back of stronge
         r internet advertising revenues. It hopes to increase subscribers by offering
         the online service free to TimeWarner internet customers and will try to sign
         up AOL\'s existing customers for high-speed broadband. TimeWarner also has to
         restate 2000 and 2003 results following a probe by the US Securities Exchange
         Commission (SEC), which is close to concluding.\n\nTime Warner\'s fourth quar
         ter profits were slightly better than analysts\' expectations. But its film d
         ivision saw profits slump 27% to $284m, helped by box-office flops Alexander
         and Catwoman, a sharp contrast to year-earlier, when the third and final film
         in the Lord of the Rings trilogy boosted results. For the full-year, TimeWarn
         er posted a profit of $3.36bn, up 27% from its 2003 performance, while revenu
         es grew 6.4% to $42.09bn. "Our financial performance was strong, meeting or e
         xceeding all of our full-year objectives and greatly enhancing our flexibilit
         y," chairman and chief executive Richard Parsons said. For 2005, TimeWarner i
         s projecting operating earnings growth of around 5%, and also expects higher
         revenue and wider profit margins.\n\nTimeWarner is to restate its accounts as
         part of efforts to resolve an inquiry into AOL by US market regulators. It ha
         s already offered to pay $300m to settle charges, in a deal that is under rev
         iew by the SEC. The company said it was unable to estimate the amount it need
         ed to set aside for legal reserves, which it previously set at $500m. It inte
         nds to adjust the way it accounts for a deal with German music publisher Bert
         elsmann\'s purchase of a stake in AOL Europe, which it had reported as advert
         ising revenue. It will now book the sale of its stake in AOL Europe as a loss
         on the value of that stake.\n'
```

In [99]:

```
plt.clf()
acc_values=history_dict['acc']
val_acc=history_dict['val_acc']
plt.plot(epochs,acc,'bo',label='Training acc')
plt.plot(epochs,val_acc,'b',label='Validation acc')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```



In [ ]:

```
```