In [1]:

```
from keras.datasets import imdb
import numpy as np

(train_data, train_labels), (test_data, test_labels) = imdb.load_data(num_words=10000)

def vectorize_sequences(sequences, dimension=10000):
    # 크기가 (len(sequences), dimension))이고 모든 원소가 0인 행렬을 만듭니다
    results = np.zeros((len(sequences), dimension))
    for i, sequence in enumerate(sequences):
        results[i, sequence] = 1.  # results[i]에서 특정 인덱스의 위치를 1로 만듭니다
    return results

x_train = vectorize_sequences(train_data)
x_test = vectorize_sequences(test_data)
y_train = np.asarray(train_labels).astype('float32')
y_test = np.asarray(test_labels).astype('float32')
```

```
Using TensorFlow backend.
```

In [2]:

```
from keras import models, layers

dpt_model = models.Sequential()
dpt_model.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
dpt_model.add(layers.Dropout(0.5))
dpt_model.add(layers.Dense(16, activation='relu'))
dpt_model.add(layers.Dropout(0.5))
dpt_model.add(layers.Dense(1, activation='sigmoid'))
```

```
WARNING:tensorflow:From C:WUsersWJWAnaconda3WlibWsite-packagesWtensorflowWpythonWf
rameworkWop_def_library.py:263: colocate_with (from tensorflow.python.framework.op
s) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From C:WUsersWJWAnaconda3WlibWsite-packagesWkerasWbackendWtenso
rflow_backend.py:3445: calling dropout (from tensorflow.python.ops.nn_ops) with ke
ep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_p
rob`.
```

In [3]:

```
dpt_model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
```

In [4]:

```
dpt_model_hist = dpt_model.fit(x_train, y_train, epochs=20, batch_size=512,
                               validation_data = (x_test, y_test))
```

```
WARNING:tensorflow:From C:WUsersWJWAnaconda3WlibWsite-packagesWtensorflowWpythonWo
psWmath_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecated
and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Train on 25000 samples, validate on 25000 samples
Epoch 1/20
25000/25000 [==============================] - 6s 235us/step - loss: 0.5973 - acc:
0.6846 - val_loss: 0.4663 - val_acc: 0.8636
Epoch 2/20
25000/25000 [==============================] - 4s 148us/step - loss: 0.4667 - acc:
0.8029 - val_loss: 0.3810 - val_acc: 0.8801
Epoch 3/20
25000/25000 [==============================] - 4s 148us/step - loss: 0.3891 - acc:
0.8605 - val_loss: 0.3253 - val_acc: 0.8868
Epoch 4/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.3364 - acc:
0.8924 - val_loss: 0.3020 - val_acc: 0.8879
Epoch 5/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2963 - acc:
0.9133 - val_loss: 0.2912 - val_acc: 0.8856
Epoch 6/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.2607 - acc:
0.9208 - val_loss: 0.3004 - val_acc: 0.8876
Epoch 7/20
25000/25000 [==============================] - 4s 149us/step - loss: 0.2352 - acc:
0.9311 - val_loss: 0.3052 - val_acc: 0.8869
Epoch 8/20
25000/25000 [==============================] - 4s 147us/step - loss: 0.2103 - acc:
0.9381 - val_loss: 0.3182 - val_acc: 0.8823
Epoch 9/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1933 - acc:
0.9432 - val_loss: 0.3355 - val_acc: 0.8772
Epoch 10/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1785 - acc:
0.9480 - val_loss: 0.3556 - val_acc: 0.8749
Epoch 11/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1621 - acc:
0.9512 - val_loss: 0.3849 - val_acc: 0.8793
Epoch 12/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1533 - acc:
0.9547 - val_loss: 0.4121 - val_acc: 0.8787
Epoch 13/20
25000/25000 [==============================] - 4s 147us/step - loss: 0.1441 - acc:
0.9585 - val_loss: 0.4231 - val_acc: 0.8765
Epoch 14/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1364 - acc:
0.9603 - val_loss: 0.4445 - val_acc: 0.8726
Epoch 15/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1277 - acc:
0.9604 - val_loss: 0.4616 - val_acc: 0.8738
Epoch 16/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1194 - acc:
0.9655 - val_loss: 0.5095 - val_acc: 0.8754
Epoch 17/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.1185 - acc:
0.9671 - val_loss: 0.5165 - val_acc: 0.8745
Epoch 18/20
25000/25000 [==============================] - 4s 145us/step - loss: 0.1109 - acc:
0.9684 - val_loss: 0.5467 - val_acc: 0.8745
Epoch 19/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1144 - acc:
0.9669 - val_loss: 0.5422 - val_acc: 0.8719
Epoch 20/20
25000/25000 [==============================] - 4s 146us/step - loss: 0.1119 - acc:
0.9698 - val_loss: 0.5519 - val_acc: 0.8682
```

In [8]:

```
original_model = models.Sequential()
original_model.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
original_model.add(layers.Dense(16, activation='relu'))
original_model.add(layers.Dense(1, activation='sigmoid'))

original_model.compile(optimizer='rmsprop',
                       loss='binary_crossentropy',
                       metrics=['acc'])
```

In [10]:

```
original_hist = original_model.fit(x_train, y_train,
                                   epochs=20,
                                   batch_size=512,
                                   validation_data=(x_test, y_test))
```

```
Train on 25000 samples, validate on 25000 samples
Epoch 1/20
25000/25000 [==============================] - 4s 155us/step - loss: 0.4453 - acc:
0.8222 - val_loss: 0.3350 - val_acc: 0.8782
Epoch 2/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.2567 - acc:
0.9119 - val_loss: 0.3015 - val_acc: 0.8783
Epoch 3/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.1991 - acc:
0.9306 - val_loss: 0.2894 - val_acc: 0.8835
Epoch 4/20
25000/25000 [==============================] - 4s 144us/step - loss: 0.1676 - acc:
0.9416 - val_loss: 0.3063 - val_acc: 0.8791
Epoch 5/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.1443 - acc:
0.9498 - val_loss: 0.3329 - val_acc: 0.8741
Epoch 6/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.1288 - acc:
0.9544 - val_loss: 0.3357 - val_acc: 0.8732
Epoch 7/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.1141 - acc:
0.9613 - val_loss: 0.3662 - val_acc: 0.8694
Epoch 8/20
25000/25000 [==============================] - 4s 142us/step - loss: 0.1034 - acc:
0.9649 - val_loss: 0.3949 - val_acc: 0.8675
Epoch 9/20
25000/25000 [==============================] - 4s 142us/step - loss: 0.0921 - acc:
0.9686 - val_loss: 0.4115 - val_acc: 0.8667
Epoch 10/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.0835 - acc:
0.9719 - val_loss: 0.5040 - val_acc: 0.8504
Epoch 11/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.0754 - acc:
0.9751 - val_loss: 0.4614 - val_acc: 0.8622
Epoch 12/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.0689 - acc:
0.9774 - val_loss: 0.5184 - val_acc: 0.8548
Epoch 13/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.0612 - acc:
0.9806 - val_loss: 0.5218 - val_acc: 0.8584
Epoch 14/20
25000/25000 [==============================] - 4s 144us/step - loss: 0.0554 - acc:
0.9830 - val_loss: 0.5489 - val_acc: 0.8570
Epoch 15/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.0481 - acc:
0.9858 - val_loss: 0.6190 - val_acc: 0.8502
Epoch 16/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.0442 - acc:
0.9868 - val_loss: 0.6074 - val_acc: 0.8538
Epoch 17/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.0400 - acc:
0.9878 - val_loss: 0.6518 - val_acc: 0.8505
Epoch 18/20
25000/25000 [==============================] - 4s 144us/step - loss: 0.0322 - acc:
0.9912 - val_loss: 0.6761 - val_acc: 0.8509
Epoch 19/20
25000/25000 [==============================] - 4s 143us/step - loss: 0.0295 - acc:
0.9921 - val_loss: 0.7162 - val_acc: 0.8482
Epoch 20/20
25000/25000 [==============================] - 4s 144us/step - loss: 0.0250 - acc:
0.9944 - val_loss: 0.7418 - val_acc: 0.8492
```

In [12]:

```
epochs = range(1, 21)
original_val_loss = original_hist.history['val_loss']
```

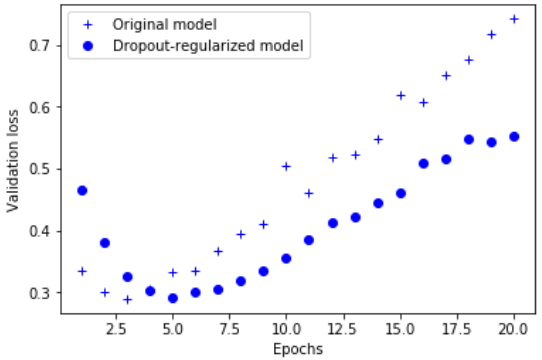
In [13]:

```
import matplotlib.pyplot as plt

dpt_model_val_loss = dpt_model_hist.history['val_loss']

plt.plot(epochs, original_val_loss, 'b+', label='Original model')
plt.plot(epochs, dpt_model_val_loss, 'bo', label='Dropout-regularized model')
plt.xlabel('Epochs')
plt.ylabel('Validation loss')
plt.legend()

plt.show()
```



In [ ]:

```

```