```
def lstm_step(inp, prev, state):
    with tf.name_scope('dimensionality'):
        dstate = state.get_shape()[1].__int__()
        din = inp.get_shape()[1].__int__()
        dout = prev.get_shape()[1].__int__()
        gates = {}
    for g in ['forget', 'input', 'output', 'state']:
        with tf.name_scope(g):
            W = tf.Variable(tf.truncated_normal([din, dstate], stddev=1 / tf.sqrt(tf.to_float(din))))
            U = tf.Variable(tf.truncated_normal([dout, dstate], stddev=1 / tf.sqrt(tf.to_float(dout))))
            b = tf.Variable(tf.zeros([1, dstate]))
            combo = tf.matmul(inp, W) + tf.matmul(prev, U) + b
            if g in ['forget', 'input', 'output']:
                gates[g] = tf.sigmoid(combo)
            else:
                state = tf.multiply(gates['forget'], state) + tf.multiply(gates['input'], tf.tanh(combo))
    with tf.name_scope('output'):
        W = tf.Variable(tf.truncated_normal([dstate, dout], stddev=1 / tf.sqrt(tf.to_float(dstate))))
        b = tf.Variable(tf.zeros([1, dout]))
        outp = tf.matmul(tf.multiply(gates['output'], tf.tanh(state)), W) + b
    return outp, state
```