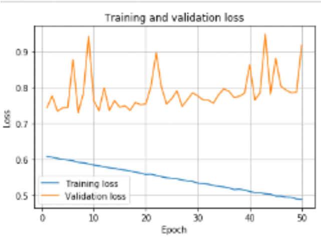
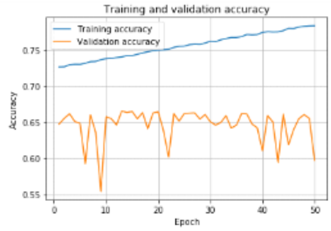
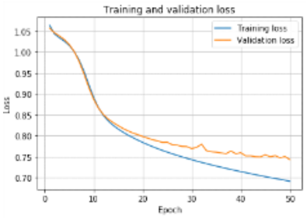
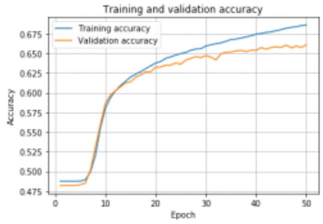
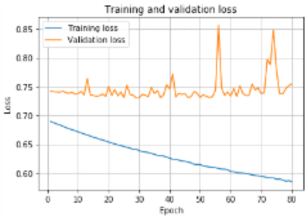
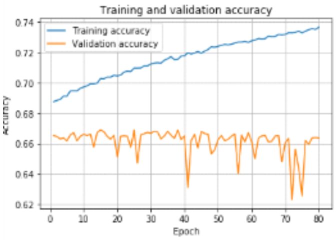
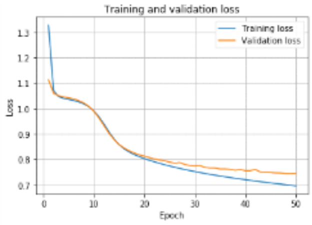
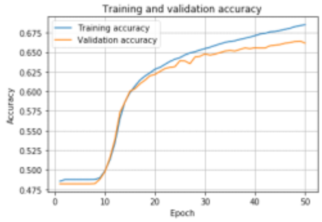
Donnerstag, 24. September 2020 20:23

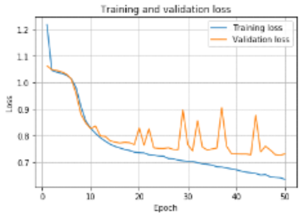
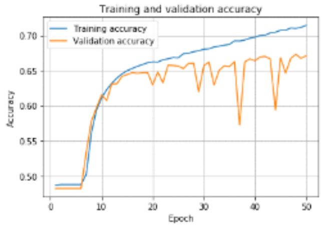
Versuch	Parameter	Training und Validation-Loss	Traning und Validation Accuracy	Accuracy der Testdaten

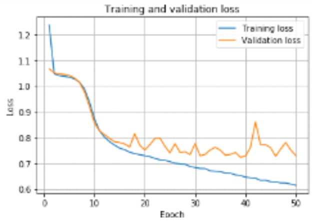
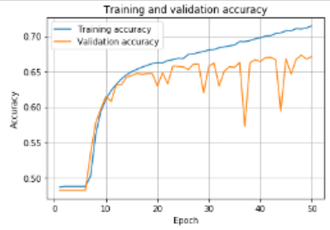
1	model_tfidf.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	
	Batchsize 50
				24631/24631 [==============================] - 8s 315us/step
				Test score: 0.9220649572030707
				Test accuracy: 0.5944947600364685

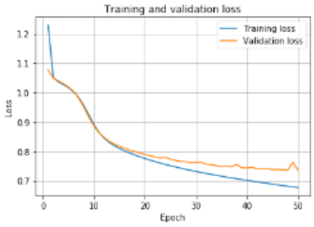
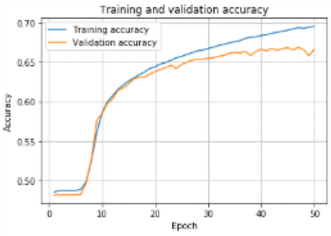
2	batch_size=150
	model_tfidf.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	
	Batch_size=150
				24631/24631 [==============================] - 8s 310us/step
				Test score: 0.7316935755105843
				Test accuracy: 0.6642036437988281

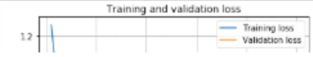
3	batch_size=150
	model_tfidf.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	
	epochs=80, batch_size=150
				Test score: 0.744206554377337
				Test accuracy: 0.6644878387451172

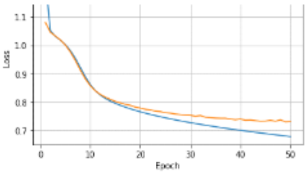
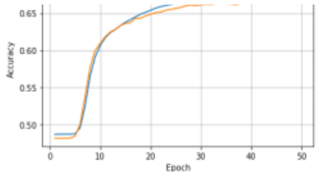
4	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(32, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	epochs=50, batch_size=150
				24631/24631 [==============================] - 9s 364us/step
				Test score: 0.7349869734544682
				Test accuracy: 0.6637570261955261

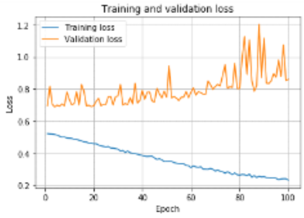
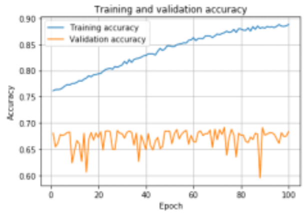
5	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(128, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(64, activation='relu'))
	model_tfidf.add(layers.Dense(32, activation='relu'))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Ohne Stopwörter
	
	epochs=50, batch_size=150
				24631/24631 [==============================] - 4s 170us/step
				Test score: 0.720617211860569
				Test accuracy: 0.6722016930580139

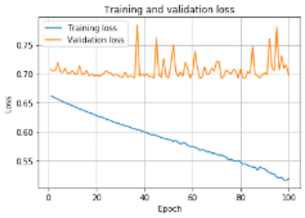
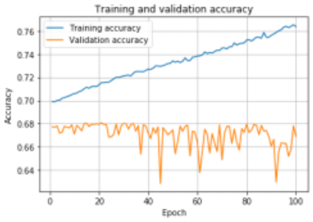
6	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(256, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(128, activation='relu'))
	model_tfidf.add(layers.Dense(64, activation='relu'))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
				24631/24631 [==============================] - 5s 198us/step
				Test score: 0.7204741581649496
				Test accuracy: 0.6687507629394531

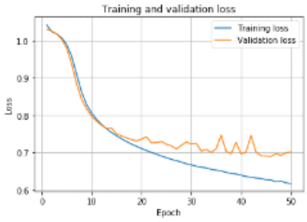
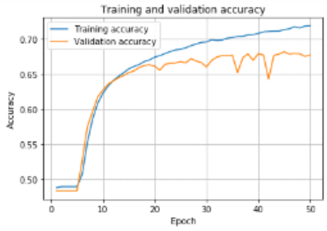
7	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(512, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(32, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Ohne Stopwörter
	
	epochs=50, batch_size=150,
				24631/24631 [==============================] - 4s 170us/step
				Test score: 0.7267410097597075
				Test accuracy: 0.6688725352287292

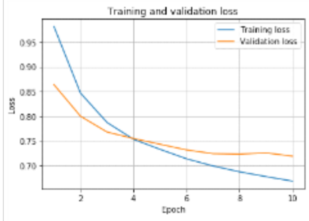
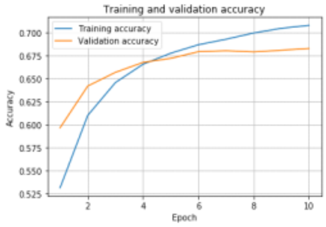
8	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(512, activation='relu', input_shape=(10000,)))
			Training and validation accuracy
			Training accuracy
			Validation accuracy
			0.70
				24631/24631 [==============================] - 5s 212us/step
				Test score: 0.7195253215576224
				Test accuracy: 0.6731760501861572

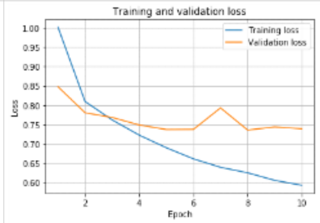
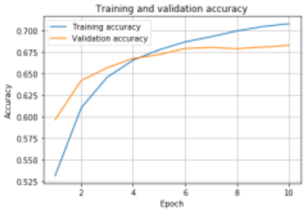
	model_tfidf.add(layers.Dense(32, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Mit Stopwörter
	
	epochs=50, batch_size=150,
				

9	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(512, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(64, activation='relu'))
	model_tfidf.add(layers.Dense(3, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	epochs=100, batch_size=100
		Training and validation loss	Training and validation accuracy	24582/24582 [==============================] - 7s 285us/step
				Test score: 0.8624409484607231
				Test accuracy: 0.680497944355011

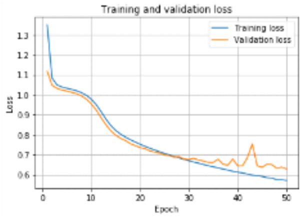
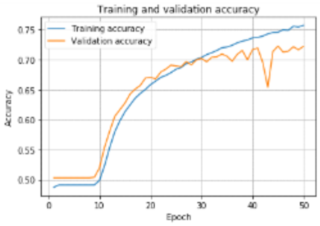
10	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(512, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(64, activation='relu'))
	model_tfidf.add(layers.Dense(3, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	epochs=100, batch_size=150,
		Training and validation loss	Training and validation accuracy	24582/24582 [==============================] - 4s 169us/step
				Test score: 0.6979884075610246
				Test accuracy: 0.6696770191192627

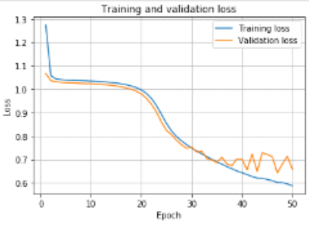
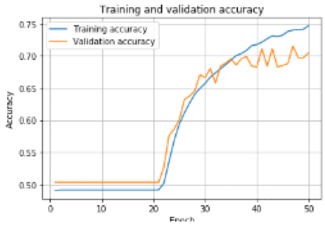
11	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(512, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(64, activation='relu'))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(3, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	epochs=50, batch_size=150
		Training and validation loss	Training and validation accuracy	24582/24582 [==============================] - 4s 172us/step
				Test score: 0.7023071024408373
				Test accuracy: 0.6768773794174194

12	model.add(Embedding(num_words, emb_dim, input_length=max_len))
	model.add(LSTM(32, recurrent_dropout=0.2, dropout=0.2))
	model.add(Dense(6, activation='softmax'))
	model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['acc'])
	history = model.fit(X_train,y_train, epochs=10, batch_size=128, validation_split=0.2)
	
	num_words = 10000
	
	# maximum length of reviews (only the first so many words are considered)
	max_len = 500
	
	# embedding dimensionality
	emb_dim = 32
		Training and validation loss	Training and validation accuracy	

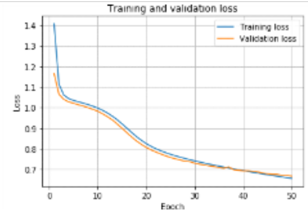
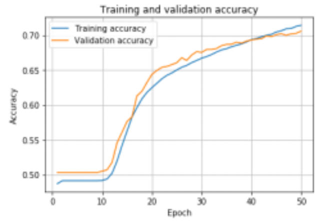
13	model = Sequential()
	model.add(Embedding(num_words, emb_dim, input_length=max_len))
	model.add(LSTM(32, recurrent_dropout=0.2, dropout=0.2))
	model.add(Dense(6, activation='softmax'))
	model.compile(optimizer='ADAM', loss='categorical_crossentropy', metrics=['acc'])
	history = model.fit(X_train,y_train, epochs=10, batch_size=128, validation_split=0.2)
	
	# number of most frequent words to consider
	num_words = 10000
	
	# maximum length of reviews (only the first so many words are considered)
	max_len = 500
	
	# embedding dimensionality
	emb_dim = 32
		Training and validation loss	Training and validation accuracy	24631/24631 [==============================] - 128s 5ms/step
				Test score: 0.7256278110176747
				Test accuracy: 0.6747594475746155

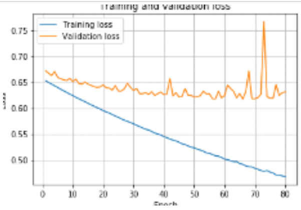
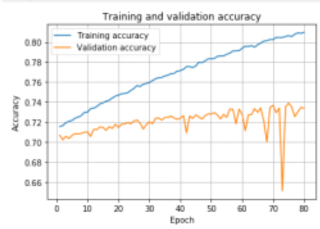
	Dataset geändert!			

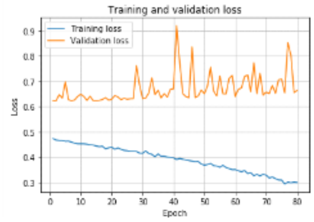
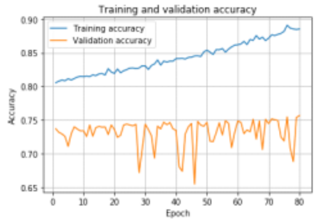
14	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(64, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(32, activation='relu'))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Ohne Stoppwörter
	
	epochs=50, batch_size=150
		Training and validation loss	Training and validation accuracy	12340/12340 [==============================] - 2s 172us/step
				Test score: 0.6478253104891537
				Test accuracy: 0.7169367671012878

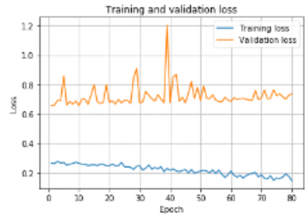
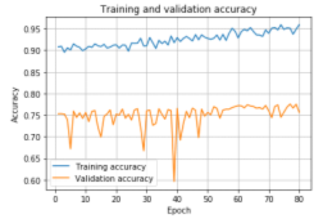
15	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(128, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(64, activation='relu'))
	model_tfidf.add(layers.Dense(32, activation='relu'))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Ohne Stoppwörter
		Training and validation loss	Training and validation accuracy	12340/12340 [==============================] - 2s 171us/step
				Test score: 0.6683487856020996
				Test accuracy: 0.6974068284034729

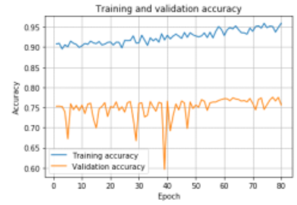
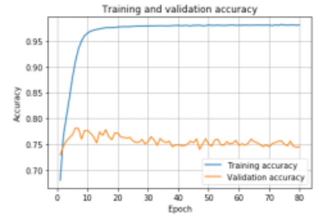
	epochs=50, batch_size=150			

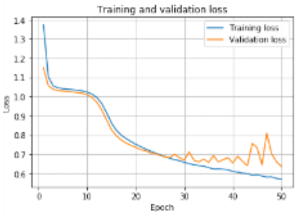
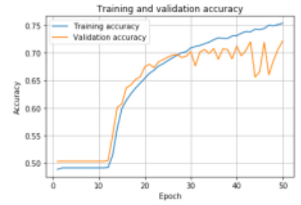
16	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(512, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(64, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Ohne Stoppwörter
	
	epochs=50, batch_size=150
				12340/12340 [==============================] - 2s 201us/step
				Test score: 0.6858836799618491
				Test accuracy: 0.6989465355873108

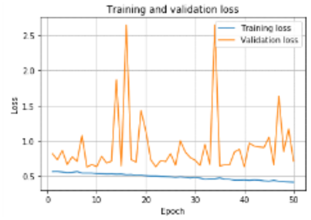
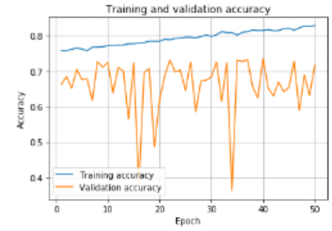
17	model_tfidf = models.Sequential()
	model_tfidf.add(laye rs.Dense(512, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(64, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Ohne Stoppwörter
	use_idf=false
	
	epochs=80, batch_size=150
				12340/12340 [==============================] - 2s 170us/step
				Test score: 0.6434801919920131
				Test accuracy: 0.7286871671676636

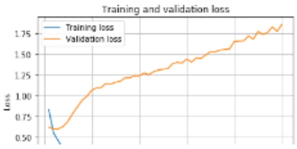
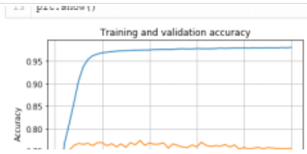
18	model_tfidf = models.Sequential()
	model_tfidf.add(laye rs.Dense(512, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(64, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Ohne Stoppwörter
	use_idf=false
	
	epochs=80, batch_size=100
				12340/12340 [==============================] - 4s 284us/step
				Test score: 0.6822377211464089
				Test accuracy: 0.7522690296173096

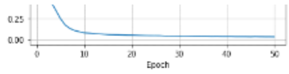
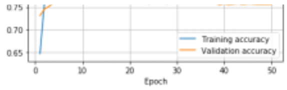
19	model_tfidf = models.Sequential()
	model_tfidf.add(laye rs.Dense(512, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(64, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()0
	
	Ohne Stoppwörter
	use_idf=false
	
	epochs=80, batch_size=200
				12340/12340 [==============================] - 2s 173us/step
				Test score: 0.7577678426931125
				Test accuracy: 0.7512965798377991

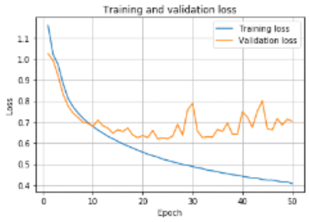
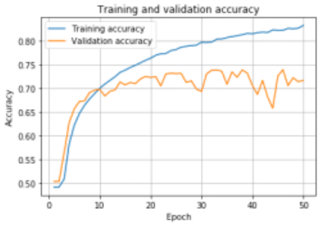
20	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(512, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(64, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Ohne Stoppwörter
	use_idf=false
	
	epochs=80, batch_size=200
				12340/12340 [==============================] - 2s 168us/step
				Test score: 3.6902647412576783
				Test accuracy: 0.7364667654037476

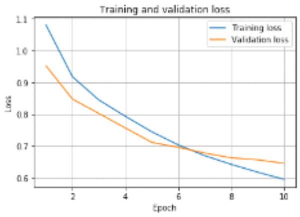
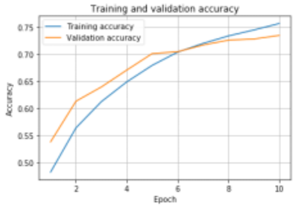
21	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(32, activation='relu'))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='SGD', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Ohne Stoppwörter
	use_idf=false
	
	epochs=50, batch_size=100
				12340/12340 [==============================] - 2s 172us/step
				Test score: 0.6544333292367586
				Test accuracy: 0.7135332226753235

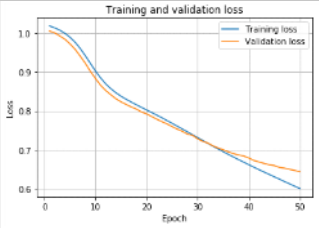
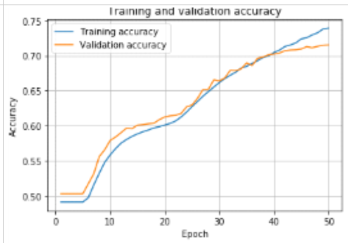
22	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(32, activation='relu'))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='SGD', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Ohne Stoppwörter
	use_idf=false
	
	epochs=50, batch_size=120
				12340/12340 [==============================] - 2s 171us/step
				Test score: 0.7175341204847278
				Test accuracy: 0.7175040245056152
				
				Aus <http://localhost:8888/notebooks/Documents/ClassifySongLyrics/ClassifySongLyrics.ipynb >

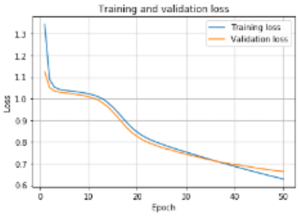
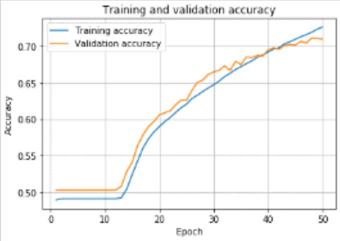
23	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(64, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(32, activation='relu'))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='ADAM', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd',
				12340/12340 [==============================] - 2s 178us/step
				Test score: 1.9657133096234338
				Test accuracy: 0.7523500919342041

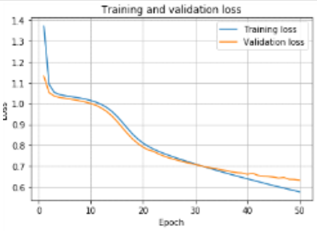
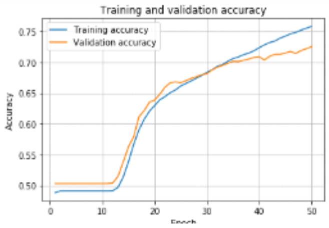
	loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Ohne Stoppwörter
	 use_idf=false
	
	epochs=50, batch_size=100
				

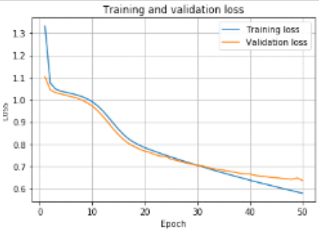
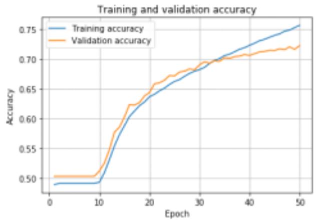
24	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(8, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(8, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='SGD', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	Ohne Stoppwörter
	 use_idf=false
	
	epochs=50, batch_size=25
				12340/12340 [==============================] - 3s 207us/step
				Test score: 0.7215690983752184
				Test accuracy: 0.7103727459907532

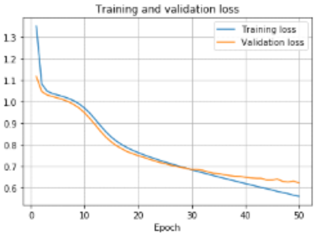
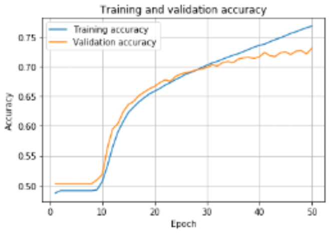
25	model = Sequential()
	model.add(Embedding(num_words, emb_dim, input_length=max_len))
	model.add(LSTM(16, recurrent_dropout=0.2, dropout=0.2))
	model.add(Dense(6, activation='softmax'))
	model.summary()
	
	model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['acc'])
	history = model.fit(X_train,y_train, epochs=10, batch_size=128, validation_split=0.2)
				12340/12340 [==============================] - 87s 7ms/step
				Test score: 0.6523191572394904
				Test accuracy: 0.7337114810943604
				
				Aus <http://localhost:8888/notebooks/Documents/ClassifySongLyrics/ClassifySongLyrics.ipynb >

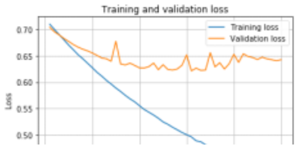
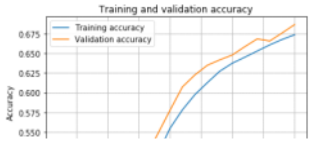
26	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(8, activation='relu', input_shape=(15000,)))
	model_tfidf.add(layers.Dense(8, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='SGD', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	pochs=50, batch_size=100
	stop_words="english", max_features=15000, ngram_range=(1, 2), min_df=20)
				12340/12340 [==============================] - 3s 262us/step
				Test score: 0.6581750529705029
				Test accuracy: 0.7123987078666687

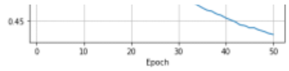
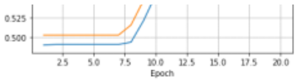
27	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(16, activation='relu', input_shape=(15000,)))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='SGD', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
				12340/12340 [==============================] - 3s 239us/step
				Test score: 0.6793370183901424
				Test accuracy: 0.6996758580207825

28	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(32, activation='relu', input_shape=(15000,)))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='SGD', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
				12340/12340 [==============================] - 3s 228us/step
				Test score: 0.649829062186725
				Test accuracy: 0.7188006639480591

29	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(64, activation='relu', input_shape=(15000,)))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='SGD', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	epochs=50, batch_size=100
	
	stop_words="english", max_features=15000, ngram_range=(1, 2), min_df=20
				12340/12340 [==============================] - 4s 289us/step
				Test score: 0.6538148880391299
				Test accuracy: 0.7162074446678162

30	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(64, activation='relu', input_shape=(15000,)))
	model_tfidf.add(layers.Dense(16, activation='relu'))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='SGD', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	tfidf_vectorizer=TfidfVectorizer(max_features=15000, ngram_range=(1, 2), min_df=20)
	
	epochs=50, batch_size=100
				12340/12340 [==============================] - 3s 277us/step
				Test score: 0.6425751422560582
				Test accuracy: 0.724230170249939

31	model_tfidf = models.Sequential()
	model_tfidf.add(layers.Dense(64, activation='relu', input_shape=(10000,)))
	model_tfidf.add(layers.Dense(256, activation='relu'))
	#model_tfidf.add(Dropout(0.01))
	model_tfidf.add(layers.Dense(6, activation='softmax'))
	model_tfidf.compile(optimizer='SGD', loss='categorical_crossentropy', metrics=['accuracy'])
	#model_tfidf.compile(optimizer='sgd',
				12340/12340 [==============================] - 3s 243us/step
				Test score: 0.6557778789699561
				Test accuracy: 0.7314424514770508

	loss='binary_crossentropy', metrics=['accuracy'])
	model_tfidf.summary()
	
	stop_words="english",max_features=10000,
	ngram_range=(1, 3)
	
	epochs=50, batch_size=64
		 0.45
		0
		10
		20
		30
		40
		50
		Epoch
			 0.525
			0.500
			2.5
			5.0
			7.5
			10.0
			12.5
			15.0
			17.5
			20.0
			Epoch	