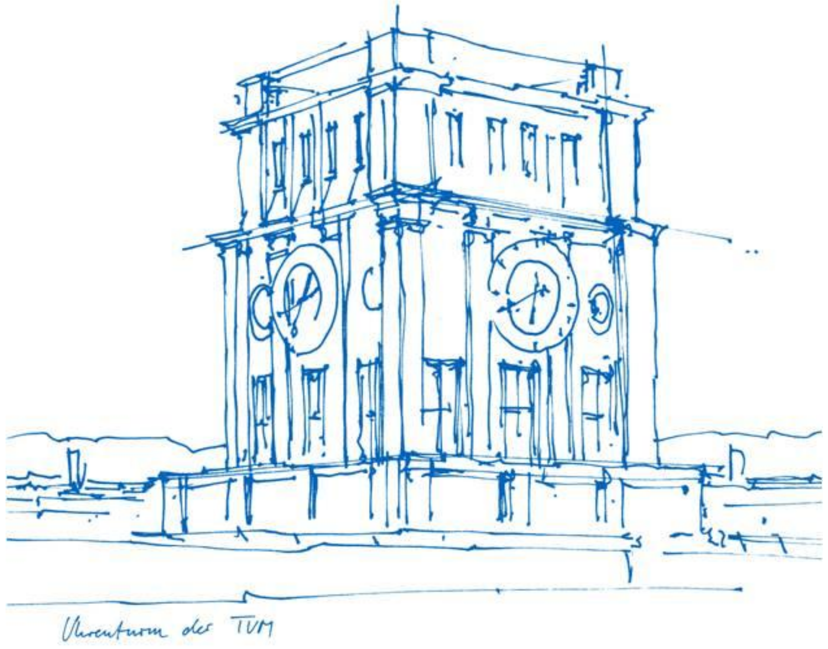
Comparing **Sentiment Analysis** of Discourse Units and Sentences

Group 9: Li Canchen, Hendrik Pauthner, Tim Pfeifle
Technische Universität München
Informatics
Research Group Social Computing
Munich, 03. July, 2019



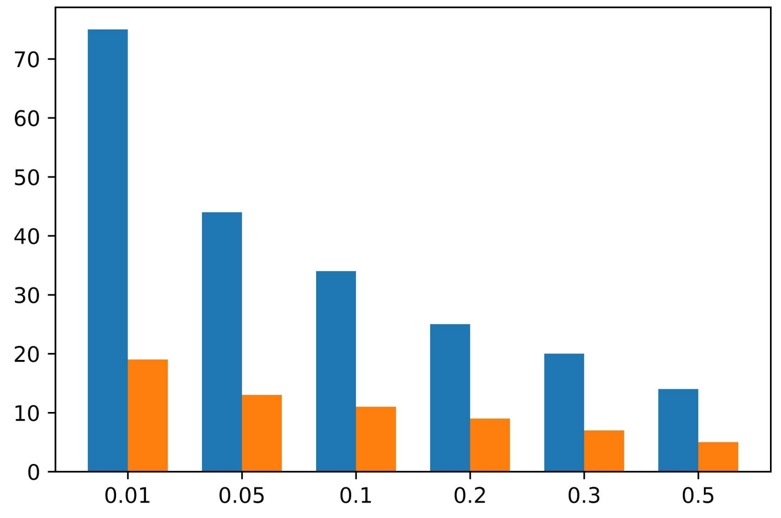
Progress Report

- Set up tf.data pipeline with on-the-fly embeddings computation
- Debugging the MILNET model
- Fix the memory usage error, made it possible to train the model on Colab
- Customized metrics for model
- Run the model on Amazon Electronics dataset with sentence-segmented level and EDU-segmented level
- Run and test the model with multiple configurations
- Comparison with baseline model

- small EDU suitability evaluation

Deciding a ‘Reasonable’ segment length and document length

To make the input to the model have identical shape, we require identical document length as well as segment length. The input will be padded if the length is not enough, and will be cutted if the length exceed.
So, how to decide a ‘reasonable’ document length and segment length?



Training Result on word2vec Embeddings

Configurations

number of classes chosen:	3; 5
segment level:	EDU; sentence; document
use attention:	yes; no

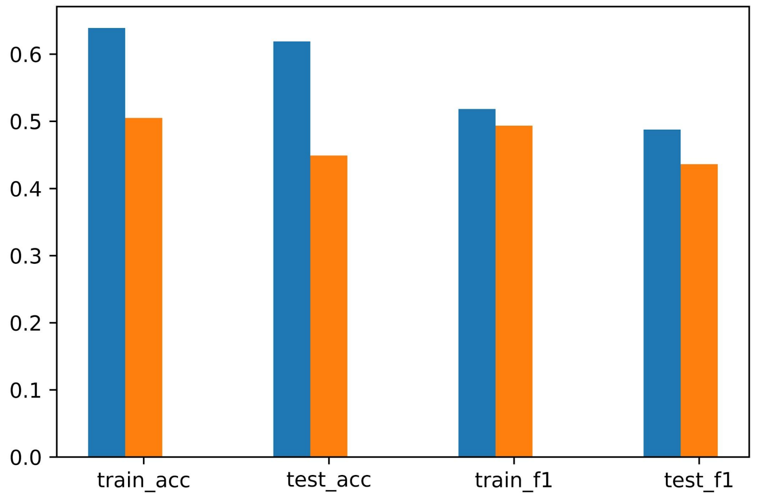
Metrics

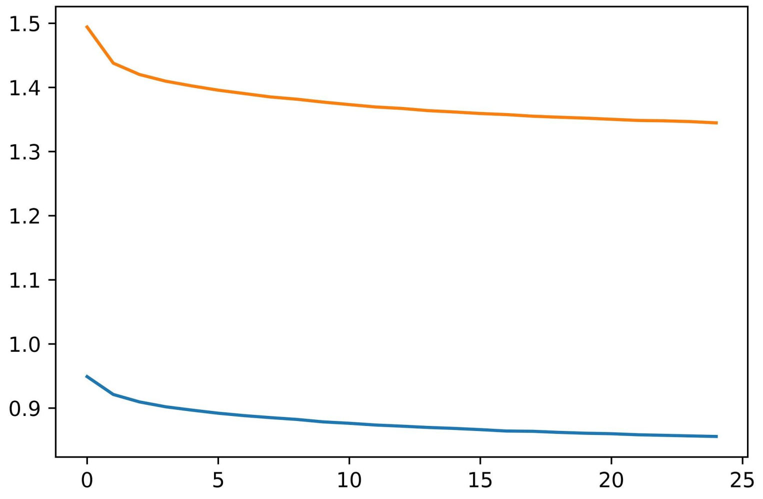
- Loss & accuracy during training process
- Training / Test accuracy on whole training / test set
- Training / Test precision, recall, and F1 score of each classes on whole training / test set

Training Result on word2vec Embeddings

Comparison between number of sentiment classes

💙: 3-classification 🧡: 5-classification

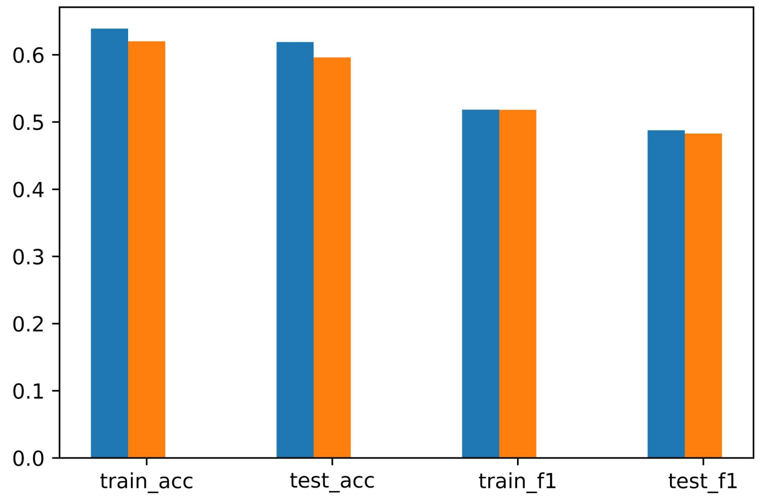


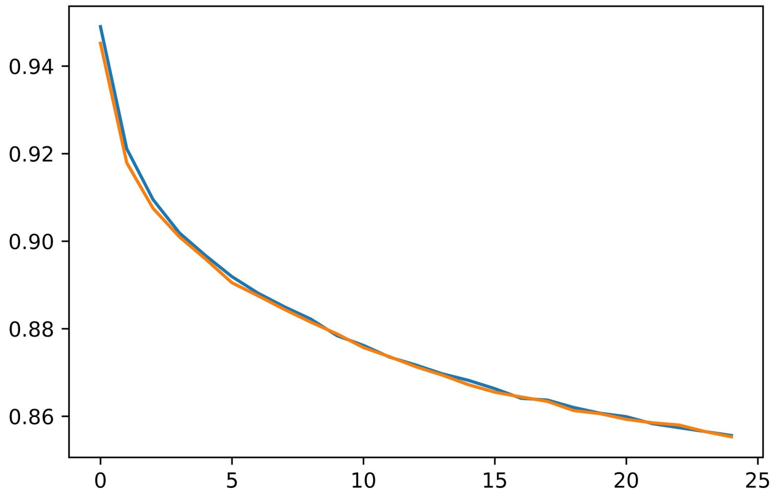


Training Result on word2vec Embeddings

Comparison between using attention model or not

💙: with attention 🧡: without attention

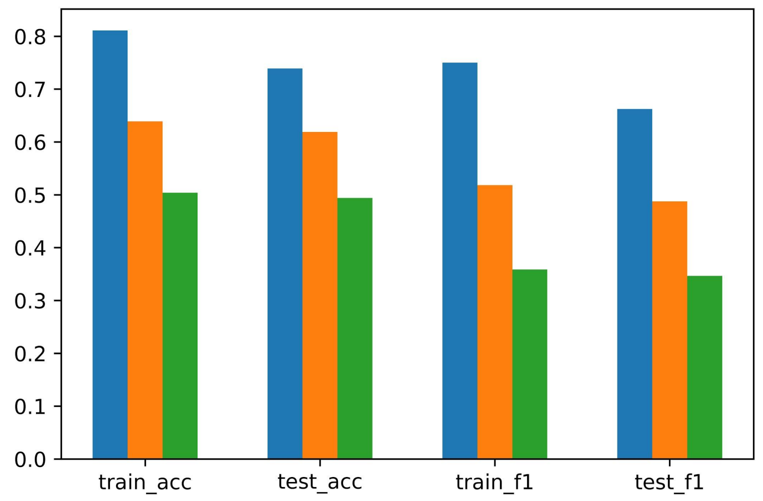


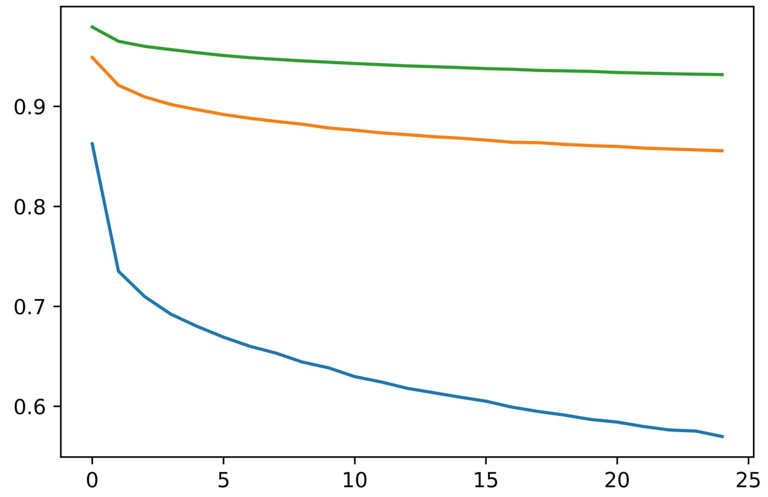


Training Result on word2vec Embeddings

Comparison between segment level used

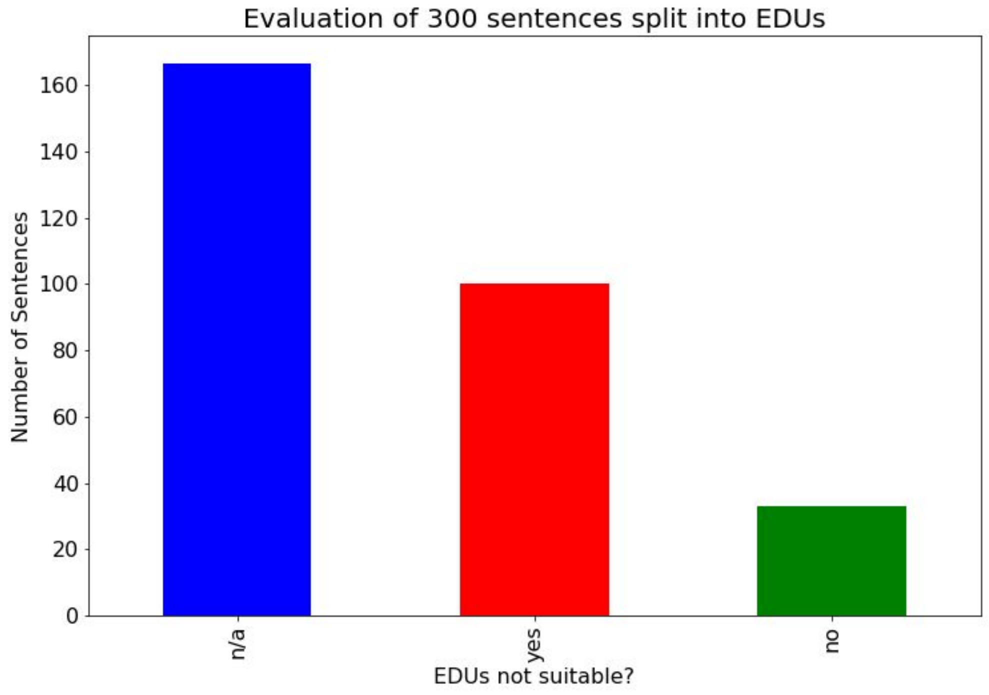
💙: document 🧡: sentence 💚: EDU





Are EDUs worth the deal?

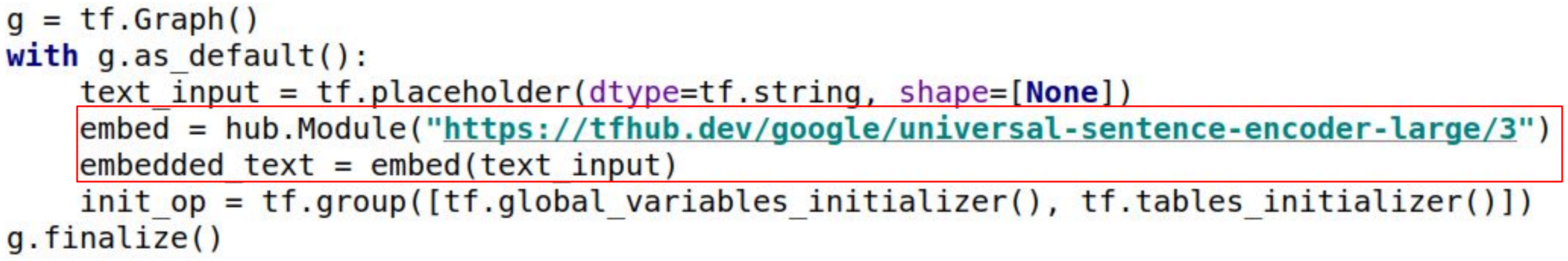
Question:
Is least one triplet (Attribute/Entity/Sentiment) distributed over **more than one** EDU in a sentence?



⇒ Splitting the sentences into EDUs seems not to be beneficial!

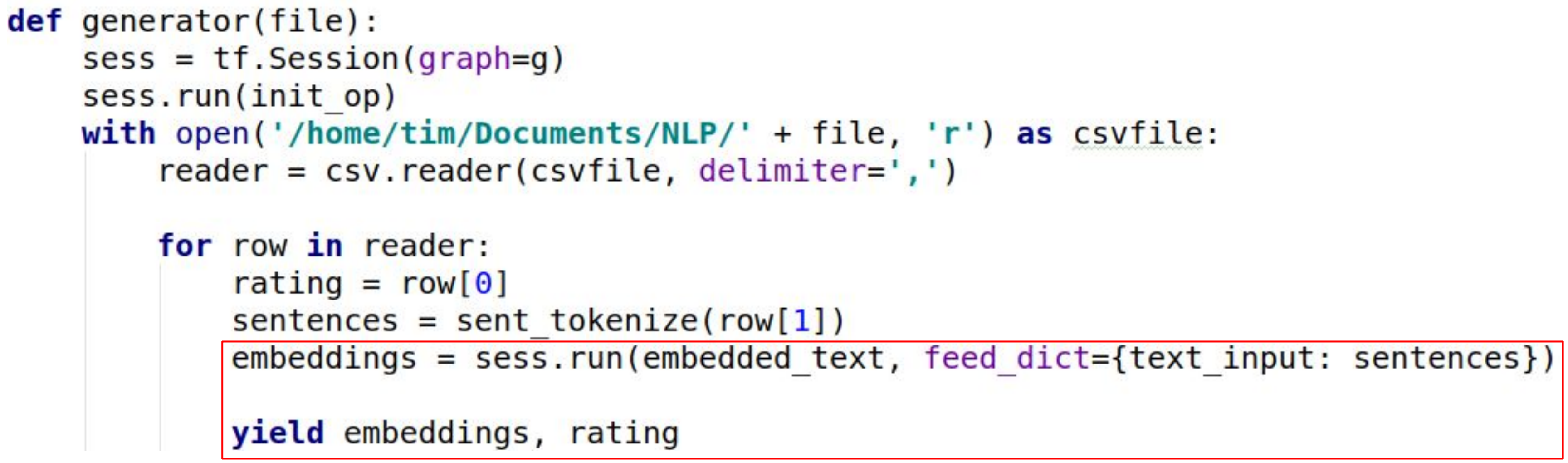
tf-Hub Module in tf.data Generator Pipeline

Initialize tf-Hub Module:



```
g = tf.Graph()
with g.as_default():
    text_input = tf.placeholder(dtype=tf.string, shape=[None])
    embed = hub.Module("https://tfhub.dev/google/universal-sentence-encoder-large/3")
    embedded_text = embed(text_input)
    init_op = tf.group([tf.global_variables_initializer(), tf.tables_initializer()])
g.finalize()
```

Generator:



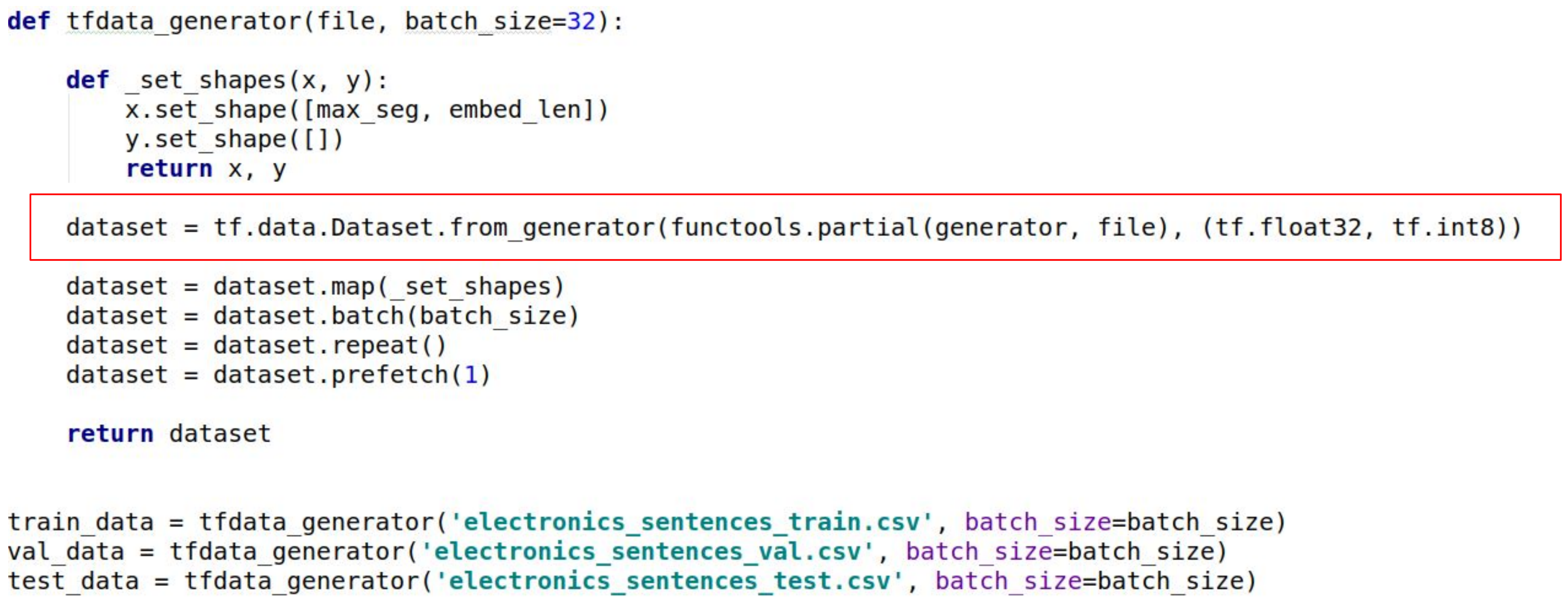
```
def generator(file):
    sess = tf.Session(graph=g)
    sess.run(init_op)
    with open('/home/tim/Documents/NLP/' + file, 'r') as csvfile:
        reader = csv.reader(csvfile, delimiter=',')

        for row in reader:
            rating = row[0]
            sentences = sent_tokenize(row[1])
            embeddings = sess.run(embedded_text, feed_dict={text_input: sentences})

            yield embeddings, rating
```

tf-Hub Module in tf.data Generator Pipeline

Create Datasets:



```python
def tfdata_generator(file, batch_size=32):

    def _set_shapes(x, y):
        x.set_shape([max_seg, embed_len])
        y.set_shape([])
        return x, y

    dataset = tf.data.Dataset.from_generator(functools.partial(generator, file), (tf.float32, tf.int8))

    dataset = dataset.map(_set_shapes)
    dataset = dataset.batch(batch_size)
    dataset = dataset.repeat()
    dataset = dataset.prefetch(1)

    return dataset


train_data = tfdata_generator('electronics_sentences_train.csv', batch_size=batch_size)
val_data = tfdata_generator('electronics_sentences_val.csv', batch_size=batch_size)
test_data = tfdata_generator('electronics_sentences_test.csv', batch_size=batch_size)
```

Problem: Computing embeddings is quite slow → long training time

Next Steps

- Try different embeddings including XLING and fastText
- Train the model on Amazon food dataset and evaluate it on the labeled organic dataset
- Visualize the predicted segment probability distribution and attention weights

Muchas Gracias!

Li Canchen, Hendrik Pauthner, Tim Pfeifle

Munich, 03. July 2019