Vortrag 2 von Joseph Birkner (15.05.2017):
Ranking with neural network derived document vectors

Joseph first explained the Vision “Ubiquitous Vertical Search” and the motivation “Encoding of Documents” of his work.
He also explained what neural networks are.
Neural network cells are defined by inputs and outputs on specific time steps. A neural network predicts labels and shows accumulated differences and errors between results.
He talked about Project IROM e.g. “Intelligent Recommendation of Massive Open Online Courses” and the use of ubiquitous vertical search in such projects. Every information need can be satisfied instantly with a vertical search engine, where vertical search means the search within a specific domain.
Also the solution for a specific domain may be applicable to other domains. He explained the term “Recommendation” and told that it implies Information Need and Information Retrieval.
Here Information Need is expressed through course query and user metadata. The “Information Need” will be satisfied by finding the relevant courses.
The motivation is the “Encoding of documents” and the axiom is that we need efficient document representations to instantaneously rank recommended courses based on student need.
That means we need courses defined by textual descriptions in a high dimensional manner.
The goal is to develop an evolving system that can generate good document representations for efficient and effective rankings. A good document representation means, that it enables a fast (constant-time) ranking function needed for efficiency.
Also the ranking has to seem intelligent to search engine user e.g. it has to be effective.
He uses document representation in traditional IR: TF-IDF
Problems of TF-IDF are ignored word order and a flawed word independence assumption.
His objective is to get a semantic space for documents like “Latent Semantic Spaces (LSS)”.
That means there is a k-dimensional space, where each dimension encodes an orthogonal semantic concept. In such an LSS, a ranking may be achieved by embedding all documents and also the user query.
At the end of this presentation he talked about the tasks and his schedules. His tasks were generating document vectors (embeddings). For his task and prototyping he uses the RWTH LM (language model) LSTM (long short-term memory)implementation.
He also created 30-dimensional document vectors and trained them with approximately 1200 course descriptions.
His schedule is to train LSTM Seq2Seq Models in Tensor Flow and to create an API to generate document/query vectors from trained LSTM’s and to retrieve ranked sets with queries among document sets. Then he wants to evaluate ranking performance on TREC datasets and the select features from the document vectors with heat maps.