Summary
The Natural Language Decathlon: Multitask Learning as Question Answering
Author: Shobhit Raj Gautam (2019201056)

1. Introduction:
Most of NLP models focuses on single task and can not be related to another. To overcome this problem, decaNLP took ten tasks : question answering, machine translation, summarization, natural language inference, sentiment analysis, semantic role labeling, relation extraction, goal-oriented dialogue, semantic parsing, and commonsense pronoun resolution. All these tasks are taken as question answers with provided context. This is done by specifiying a task as question q, i.e such that all inputs have question, context and answers, The task constraints are also fed to model as input thus making a single a multitask performing model that can also transfer learning of its pretrained models by allowing a model to generalize to completely new tasks through different but related task descriptions.
After that, a new multitask question answering network (MQAN) that jointly learns all tasks by novel dual coattention and multi-pointer-generator decoder to multitask across all task. Thus making MQAN a good model for not just one task but many tasks.

2. Tasks and Metrics:
- *Question Answering*: It receives a question and a context that contains information necessary to output the desired answer.
  English Wiki forms context and answers are sequences of words copied from the context.
- *Machine Translation*: Machine translation models receive an input document in a source language that must be translated into a target language using IWSLT 2016 English to German training data
- *Summarization*: It take in a document and output a summary of that document using CNN/daily-mail corpus.
- *Natural Language Inference*: It receive two input sentences: a premise and a hypothesis. Models must then classify the inference relationship between the two as one of entailment, neutrality, or contradiction using MNLI corpus.
- *Sentiment Analysis*: Sentiment analysis models are trained to classify the sentiment expressed by input text.
- *Semantic Role Labeling:* This model are given a sentence and predicate (typically a verb) and must determine ‘who did what to whom,’ ‘when,’ and ‘where’.
- *Relation Extraction:* It take in a piece of unstructured text and the kind of relation that is to be extracted from that text. It uses a corpus-level F1 metric (cF1) in order to accurately account for unanswerable questions.
- *Goal-Oriented Dialogue*: The key component is Dialogue state tracking i.e user requests, convo history, dialogue tracking,user interaction.
- *Semantic Parsing*: It translates natural language questions into structured SQL queries so that users can interact with a database.
- *Pronoun Resolution:* The model modified them to ensure that answers were a single word from the context taken from MWSC schema.

3. Multitask Question Answering Network (MQAN):
The model contains context, question and answer. Moreover, Question has key info restricting answer space so coattention is applied to its representation. Also, the pointer-mechanism is generalized into a hierarchical, multi-pointer-generator that enables the capacity to copy directly from the question and the context.
The model input: 3 vectors with each row is dimensional embedding. The matrices are: a context c with l tokens, a question q with m tokens, and an answer a with n token. These are given to encoder which uses a deep stack of recurrent, coattentive, and selfattentive layers to produce final representations of both context and question sequences designed to capture local and global interdependencies.
The Answers are obtained from decoder by projecting the answer embeddings onto a d-dimensional space with self attention layers. Also we add positional encoding to answer representation for recurrence and convolution. The self attention helps decoder in knowing previous outputs and attention over the context to prepare for the next output, thus making it multi-headed attention.
The decoder state uses LSTM with attention that generates recurrent context state that is used along with previous word to generate intermediate state h(t). This further generates attention weights for

context and questions to get encoded info.
These attention weights are combined with context and fed through a feedforward network with tanh activation to form the recurrent context state and question state.
To generate out of context/question words access to v additional vocabulary tokens is given. The three distribution cover the union of the tokens in the context, question, and external vocabulary so that each distribution is in R^{l+m+v}. The train using a token-level negative log-likelihood loss over all time-steps.

4.Experiments and Analysis:

- The pointer-generator sequence-to-sequence (S2S) model take in only a single input sequence, so we concatenate the context and question for this model, This was better than previous baselines but worse than MQAN.
- Adding self attention encoder-decoder layer to S2S model increases performance and extracts better information from context and questions.
- Now model is fed context and questions separately with a coattention mechanism. The performance dropped on many tasks as pointer generator mechanism was able to copy directly from the question, so with different inputs, model working was not good.
- To overcome this, question pointer was introduced in model making it the highest performing question answering model trained on SQuAD dataset.

The questions uses external vocabulary efficiently as context and question pointer uses coattention which allows information from the question to flow directly into the decoder which is better than copying as it generates output directly.
The various tasks requires different iterations to converge thus categorising tasks in 2 : easy and hard tasks.
Multi-Pointer-Generator and task identification: MQAN toggles between 3 choices: generating from the vocabulary, pointing to the question, and pointing to the context. This was totally task dependent. Ex: SQuAD, QA-SRL, and WikiSQL, the model mostly copies from the context. SST, MNLI, and MWSC, the model prefers the question pointer. IWSLT and WOZ, the model prefers generating from the vocabulary because German words and dialogue state fields are rarely in the context.
MQAN was also adjustable to newly adapted tasks and performed well. For text classification, SNLI was fine-tuned in MQAN which gave 87% score better than state of art score. Also, MQAN can perform well on binary sentiment classification.

5. Related Work:

- Transfer Learning in NLP: The pretrained embedding using Glove and Word2Vec, intermediate representations from machine translation model and model weights can be transferred that improves performance. MQAN and decaNLP makes it possible to transfer an entire end-to-end model that can be adapted for any NLP task cast as question answering.
- Multitask Learning in NLP: Sequence-to-sequence architectures can be used to multitask across translation, parsing, and image captioning using varying numbers of encoders and decoders. This can helps in donig one task through other.
- Meta-Learning: Meta-learning attempts to train models on a variety of tasks so that they can easily learn new tasks, train meta-agents that control parameter updates, augment models with special memory mechanisms, and maximize the degree to which models can learn new tasks.

6. Conclusion:
The decaNLP allows us to multitask, 10 tasks in this case, although all were casted to question answering task. MQAN uses a multi-pointer-generator decoder to capitalize on questions as natural language descriptions of tasks. MQAN was trained jointly on all tasks without specific task modules and implemented some improvements in it. Furthermore, MQAN uses transfer learning that improves results with pretrained weights and demonstrated zero-shot domain adaptation capabilities. Thus setting MQAN as benchmark for general NLP model.