Towards fast and accurate neural Chinese word segmentation with multi-criteria learning

Short paper presentation with emphasis on Big Data aspects

Antoni Maciąg, January 2022

The task - Chinese Word Segmentation

- Given a sentence in Mandarin Chinese, divide it into words.
- There are no spaces between characters or words!
- Words may be composed of one or many characters (most commonly one or two).
- The same characters may appear in different words (or constitute their own words).
- Example: 我--愛--好吃--的--餃子。I love tasty dumplings.
- 吃--餃子--是--我--的--愛好。Eating dumplings is my hobby.
- Crucial in many higher-level NLP tasks for Chinese and in the industry (e. g. search engines).
- Standard F1 as evaluation metric.

A subtle problem - granularity

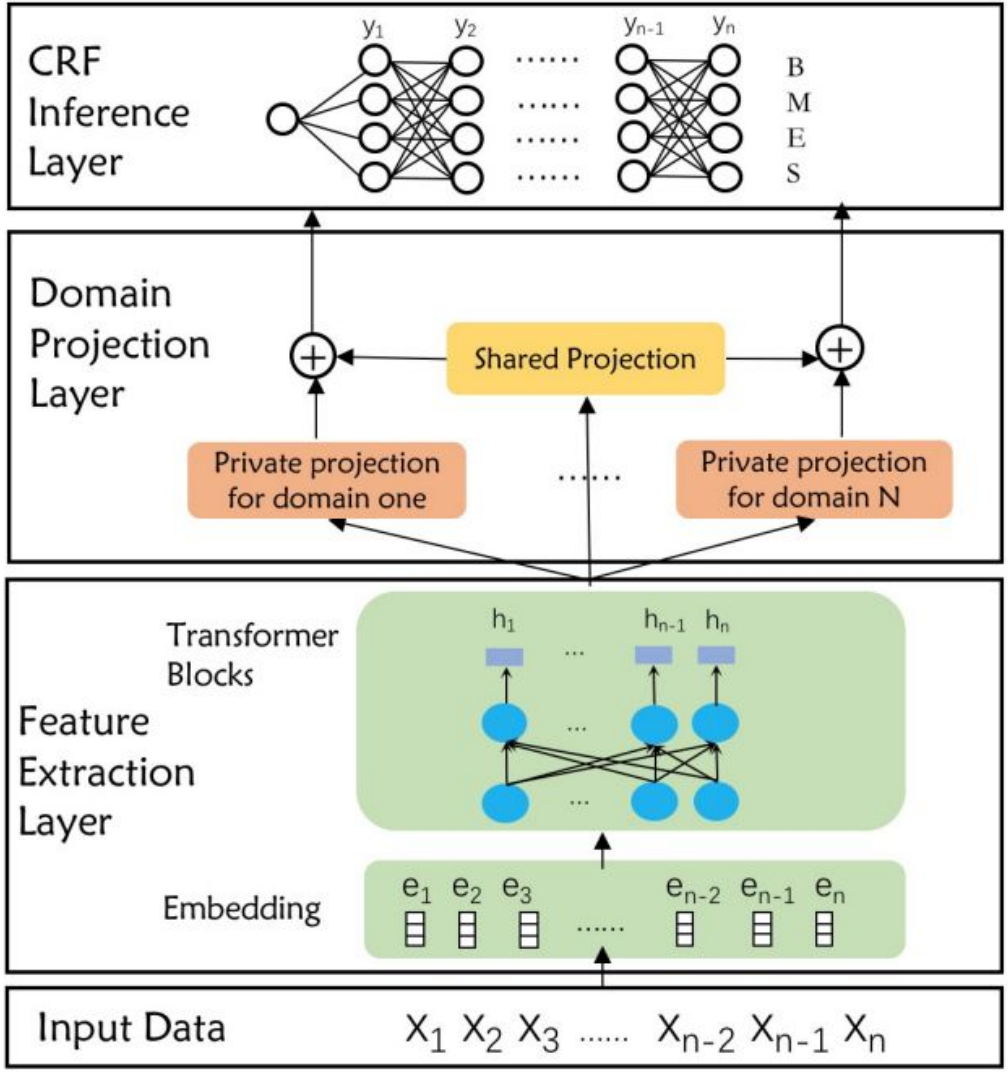
- It might be ambiguous what division we want.
- 刘国梁--赢 得--世界冠军 liu guoliang – win – world champion
- 刘--国梁--赢 得--世界--冠军 liu - guoliang – win – world - champion
- Vocabulary size vs. specific meanings tradeoff.
- Maybe this first one is better while learning to segment sports news articles?
- This also means there are different datasets (with overlapping knowledge),
- We want to learn from them and not get confused.

The paper

- 1903.04190.pdf (arxiv.org)
- Huang et al. (2019)
- Proposes a model for the CWS task.
- What interests us - optimization methods for quick inference and training on large datasets.
- Search engines have to be quick.
- Also a solution to the granularity problem.

BERT

- Good starting point.
- Usually used as whole seq2seq encoder-decoder model, here only encoding.
- Better than RNNs due to parallelization.
- However, still computationally expensive.


CRF
Inference
Layer
y1
y2
yn-1
yn
B
M
E
S
Domain
Projection
Layer
Shared Projection
Private projection
for domain one
Private projection
for domain N
Transformer
Blocks
Feature
Extraction
Layer
h1
hn-1
hn
Embedding
e1
e2
e3
en-2
en-1
en
Input Data
X1 X2 X3 ..... Xn-2 Xn-1 Xn

Domain Projection Layer

- Different granularities ~ different domains.
- While learning, for each domain *d*, learn a matrix M_d that projects vectors into domain space.
- Also learn a matrix M^* that projects vectors into common space.
- While inferring, concatenate both representations and feed the result into inference layer.

Distillation

- It was shown that lower and middle layers of BERT are mostly responsible for phrase-level dependencies and syntax. Authors showed that the 3rd layer is the most important for CWS.
- We can reasonably expect to cut down from 12 to 3 layers with hardly any accuracy loss.
- Distillation - a truncated BERT learns to simulate the predictions of the full 12-layer BERT.
- More time for training, but then inference about 2 times faster.

Quantization

- Using 16-bit floats instead of 32-bit ones.
- Obviously reduces training/inference time, model size, carbon footprint…
- Also goes well with GPU architecture.
- In this case precision loss was shown to be negligible.
- Performance gains not dramatic but noticeable (up to about 20%).

Compiler optimization - Accelerated Linear Algebra

- Speeds up TensorFlow operations.
- Finds operations that are often performed sequentially and performs them at once, without writing intermediate results to memory.
- In this model, performance gains comparable to these obtained by quantization.

Sources

Towards Fast and Accurate Neural Chinese Word Segmentation with Multi-Criteria Learning (aclanthology.org)

Multi-task Domain Adaptation for Sequence Tagging (aclanthology.org)

XLA: Optimizing Compiler for Machine Learning | TensorFlow - btw paper's authors copypasted from here :)

[1706.03762] Attention Is All You Need (arxiv.org)