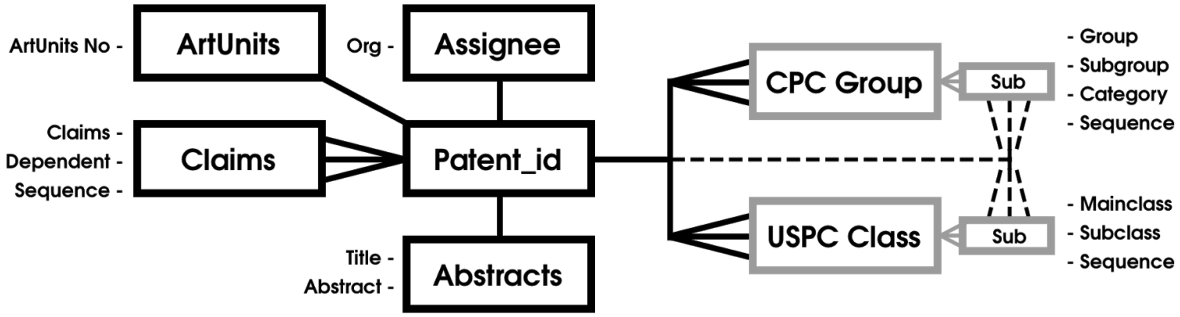
Research Log

0 Data



I Discrete Classifiers

Classifiers:

1. Group Art Unit
2. USPC
    a. Main Class
    b. Sub Class
3. CPC
    a. Group
    b. Subgroup

Main ideas behind the Classifiers:

a. Represent Training Examples for the model as close as possible to the real use cases that our clients will encounter.
b. Universal structure among Classifiers. Possibly employ Transfer Learning to improve performance across Classifiers.
c. Have modular design (Feature Extraction) to be easily expandable later on.

Basic Structure:

1. **Input**: N Claims of a Patent (N>=1) from Client. Alternatively work with Titles and Abstracts.

2. **Feature Extraction**: Create a Claim representation. Any bit of information we can extract from a Claim. Possibly a ClaimBiRNN.
    a. Character / Word / Document Embeddings.
    b. Part Of Speech Tagging
    c. Dependency Tagging
    d. Named Entity Recognition Tagging
    e. Orthographic, Custom, etc. Features

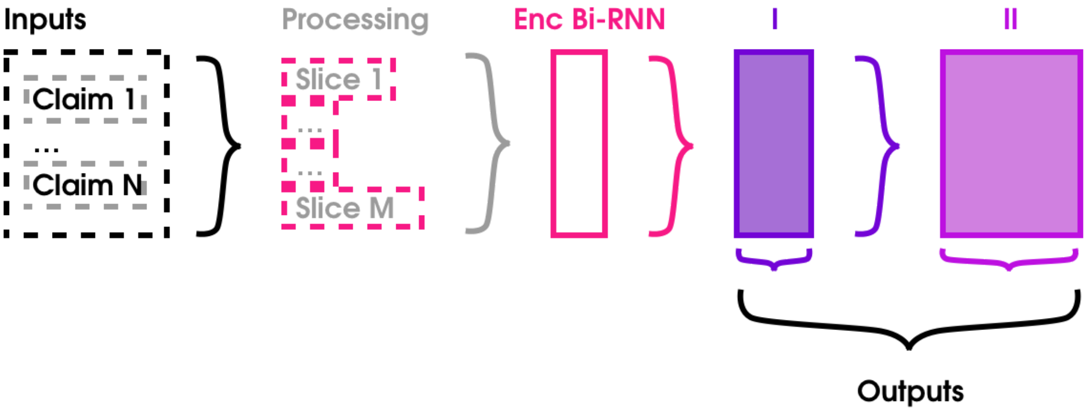
3. **Processing/Segmenting**:
    a. Patent Claims Graph: Reconstruct a dependency Graph using ‘dependent’ and ‘sequence’ fields.
    b. Slicing: Slice Graph Tree into 1D Claim Branches (Slices), where each Branch consists from N Claims (N>=1).

4. **Encoder Dynamic Bidirectional RNN**: Encode each Slice into a vector representation using an RNN where each timestep is a single Claim.

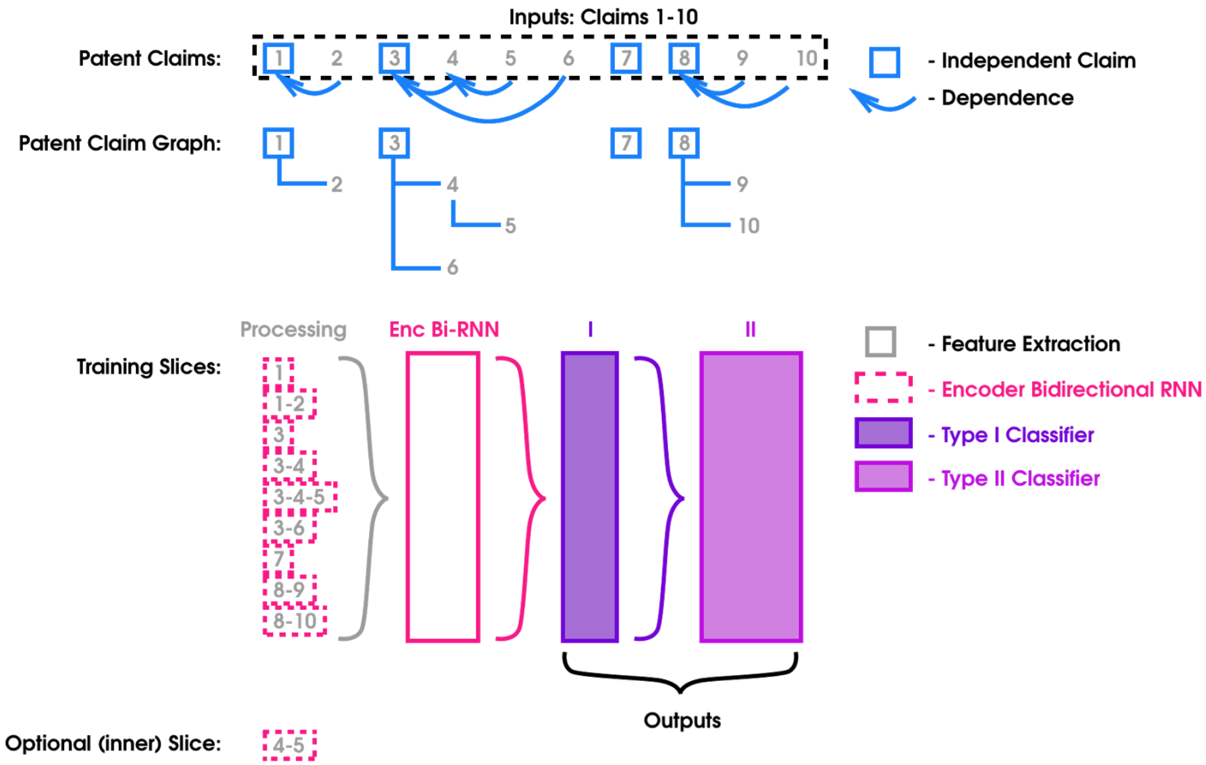
5. **Type I Classifier**: Using EncBiRNN output, Trains/Classifies each Training Example (Slice) into 1st order Class [USPC Mainclass, CPC Group, Art Unit] ***Probabilities***.

6. **Type II Classifier**: Using EncBiRNN output and Type I Classifier output, Trains/Classifies each Training Example into 2nd order Class [USPC Subclass, CPC Subgroup] ***Probabilities***. Use Candidate Sampling for multi-label classification.

7. **Output**: Groups Type I and II Probabilities.



Example:



Issues:

1. Question we will need to decide is how exactly we are slicing our reconstructed graph.
2. If Clients input during inference is sliced into more than 1 Slice we can:
   a. Return averaged OUTPUT for all slices.
   b. Return inference per each slice. (not recommended, might confuse Client)

II Research Papers

1. A Simple but Tough-to-beat Baseline for Sentence Embeddings
2. A Hierarchical Recurrent Encoder-Decoder For Generative Context-Aware Query Suggestion
3. Hierarchical Attention Networks for Document Classification
4. Probabilistic Siamese Network for Learning Representations #Similarity
5. Highway Networks