Formalizing the Ontology

During the tutorial, we defined some classes and potential supra-sub classes as well as concrete individuals; we identified if some of the classes where to be borrowed from the ontologies discussed in the previous session. We proposed some properties we thought were essential with their domain and ranges. We came up with manchester syntax definitions in an informal way and we started “translating” those informal definitions into formal manchester syntax definitions.

A sample of the work done can be found here:

* 2 Classes, Subclasses, Helper Classes

LM - supra class of LLM

LLM - SLM is a subclass of LLM, LLM is a subclass of LM

ScietificPaper

Conference

Author (for papers)

Language (to borrow from other onto)

Date (to borrow)

supraclass NNs - Architecture\_NNs supraclass of Activation Functions and NN Layers?

Activation functions - individuals: relu, sigmoid, tahn are of type Activation function

NN Layers - subclasses or individuals?: input layer, hidden layer, outputlayer

Data - subclasses: training data, testing data, validation data, Big Data, textual\_data, spoken\_data

wordembeddings - we can put glove, word2vec, fastext as individuals (define after)

NaturalLanguageTask - individuals: NER, InformationExtraction, speech recognition, (add more)

EvaluationMetrics - subclass Extrinsic: individuals - F1 Score, Accuracy, Recall, Precision

subclass Intrinsic: Individuals - Entropy, perplexity

* 3 Object / Data properties with ranges-domains, properties, subproperties, chars of the properties, etc

Domain Property Range

author writes Scientific Paper - inverse - scientific paper is written by some author

Scientific Paper is\_presented\_in some conference

ScientificPaper was\_written\_when Date

LLMS speak language (the language that the llm speaks”)

LLMS was\_trained\_on textual/spoken data

LLMs solve NaturalLanguageTask

LLMs have\_accuracy float

LLMs have\_precision float

LLMs have\_recall float

LLMs have\_f1 float

LLMs have\_entropy float

LLMs have\_perplexity float

LLMs specializes\_in\_NLTask NaturalLanguageTask

LLMs are\_introduced\_in ScientificPaper

NN Layer has\_size integer x integer

* 4 Manchester syntax-ish definitions of the classes etc

LLM - is a LM that has\_been\_trained\_on big data, arch\_type (refine the name of the prop!!) is Neural

SLM - is a LLM that is\_small

LM and SLM are disjoint (they are not the same)

Big data - data that —---------------------------- (come up with a def)

training data - is data that has property used\_for with value training

testing data - is data that has property used\_for with value testing

validation data - is data that has property used\_for with validating training

training, testing and validation data are disjoint

textual\_Data is data produced writing

spoken\_data is data produced orally

textual and spoken data are disjoint

input layer - type NN Layer and comes\_first

hidden layer - type NN layer and is\_deep

output layer - type NN layer and comes\_last

neural isdisjoint with probabilistic? glove embeddings use both methods tho

4-Manchester Definitions:  
**Step 4: Class Definitions in Manchester Syntax:**

**LLM:**Class: LLM

SubClassOf: LM

EquivalentTo: LM and (llmWasTrainedOn some BigData) and (architectureType value Neural)

**SLM:**Class: SLM

SubClassOf: LLM

EquivalentTo: LLM and (hasSize value "small")

**BigData:**

Class: BigData

SubClassOf: Data

EquivalentTo: Data and (hasVolume some xsd:integer[>= 1000000])

**TrainingData:**

Class: TrainingData

SubClassOf: Data

EquivalentTo: Data and (dataUsedFor value Training)

DisjointWith: TestingData, ValidationData

**TestingData:**

Class: TestingData

SubClassOf: Data

EquivalentTo: Data and (dataUsedFor value Testing)

DisjointWith: TrainingData, ValidationData

**ValidationData:**

Class: ValidationData

SubClassOf: Data

EquivalentTo: Data and (dataUsedFor value Validation)

DisjointWith: TrainingData, TestingData

**TextualData:**

Class: TextualData

SubClassOf: Data

EquivalentTo: Data and (producedBy value Writing)

DisjointWith: SpokenData

**SpokenData:**Class: SpokenData

SubClassOf: Data

EquivalentTo: Data and (producedBy value Speaking)

DisjointWith: TextualData

**InputLayer:**Class: InputLayer

SubClassOf: NNLayer

EquivalentTo: NNLayer and (layerPosition value "first")

**HiddenLayer:**

Class: HiddenLayer

SubClassOf: NNLayer

EquivalentTo: NNLayer and (isDeep value true)

**OutputLayer:**Class: OutputLayer

SubClassOf: NNLayer

EquivalentTo: NNLayer and (layerPosition value "last")

note: and we can put pdf already i sent you in wp maybe is more organized