**Topic Models**

* Unsupervised learning
* Many variations of algorithms – comparable
* They are usually used to detect themes in documents, what could we do to change this to sections of documents?
  + Idea 1: Manually divide each documents to sections
  + Idea 2: Use n-grams to create multiple documents from one document, e.g. r
* Number of topics: Specified based on the ontology, how many different options are there?
* Will topic modelling algorithms return something useful in regards to the ontology, i.e. will the topics found be easily interpretable as some parts of the ontology?

**Training data**: Unlabeled SAD’s

**Testing data**: a few labelled SAD’s => Ground truth

* Comparing to a “gold standard”
* Threat to validity: Small amount of testing data (possibly training data as well) & is our labelling the true labelling?

**Benchmark / Metrics**:

* Precision/Recall for each topic, i.e. we would label some SAD’s and see whether the model identifies topics that match our labels
* Something about number of topics detected that are useful for ontology (how to define this?)
* Use word-frequency datasets to evaluate interpretability of the topics identified by the algorithms
* Paper mentions “probability” -> unsupervised learning

**Levels of difficulty**:

* Manually divide documents into sections
* Use n-grams to create sections (and then the algorithm to identify sections about one topic)
* Also identify images and tables and what topics they belong to

**Algorithms**

*LDA*: simple statistical model

*Probabilistic LDA*: Here we need the posterior distribution of the data - not computable, needs to be estimated.

* Sampling-based algorithms
* Variational algorithms

\* which approach is better depends on the particular topic model being used <- not sure what this means

*NMF*: Non-negative Matrix Factorization, Linear-algebraic model.

**Approach ideas**:

Compare different topic models

Compare topic model(s) and soft clustering (fuzzy)

Compare topic model(s), soft clustering and hard clustering.

NLP Summarization