Automatic Summarization of Scientific Papers

Robin Cosbey & Josh Loehr

Motivation

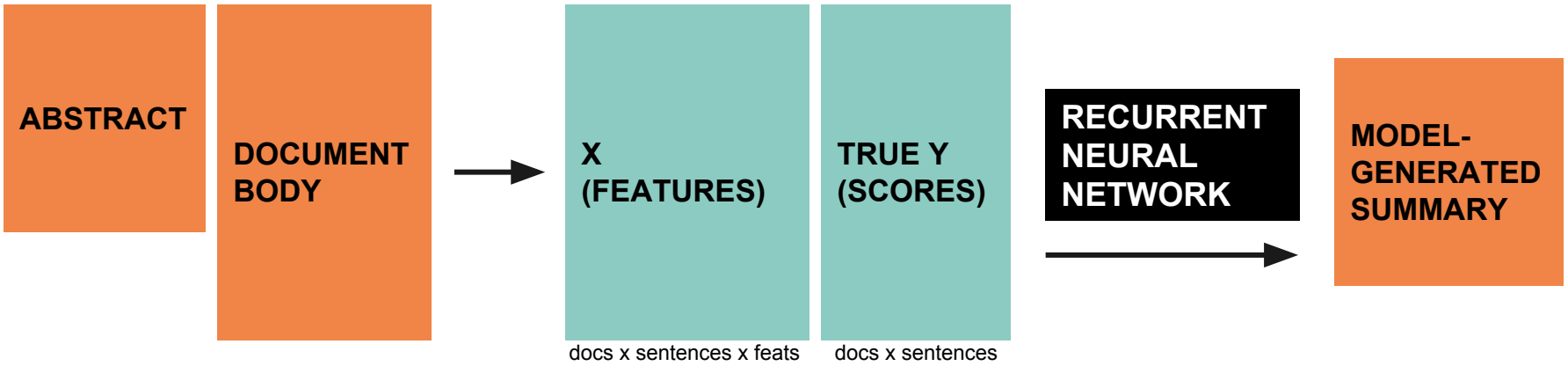
With a growing number of scientific papers, having access to concise summaries can help researchers.

Generate **extractive** summaries of scientific articles.

The goal is to build a model which can create summaries resembling the papers' abstracts, without having knowledge of the abstract itself.

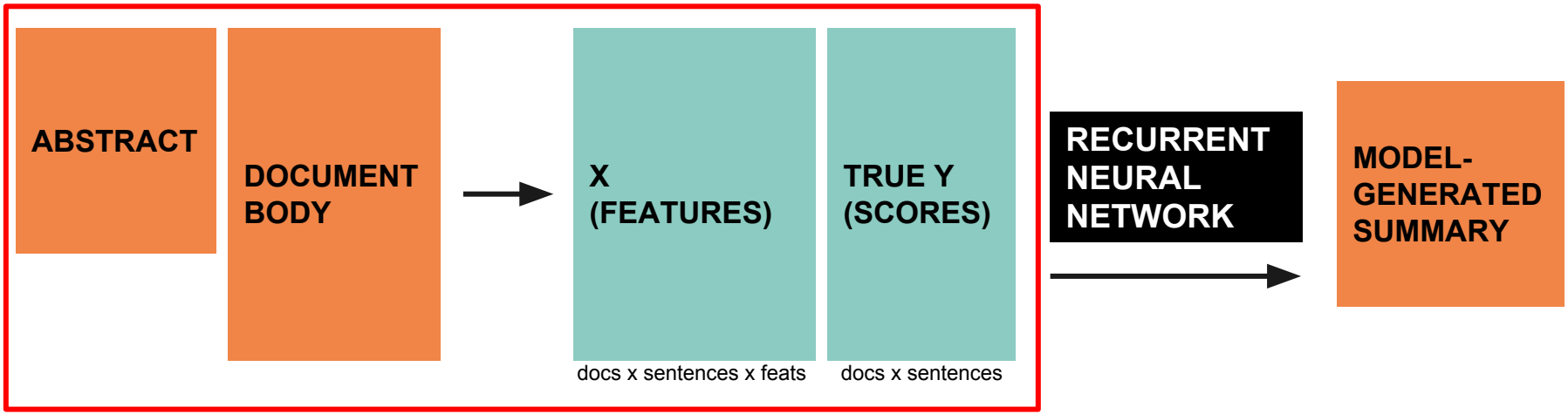
Approach

- Extraction-based
- Single document



Approach

- Extraction-based
- Single document



Datasets

- Computation and Language corpus
  - 183 scientific articles from Association for Computational Linguistics conferences
  - Documents automatically converted from LaTeX to XML
- MEDLINE/PubMed corpus
  - ~ 1.8 million documents available for free use
  - Documents converted to XML with specific MEDLINE/PubMed data elements

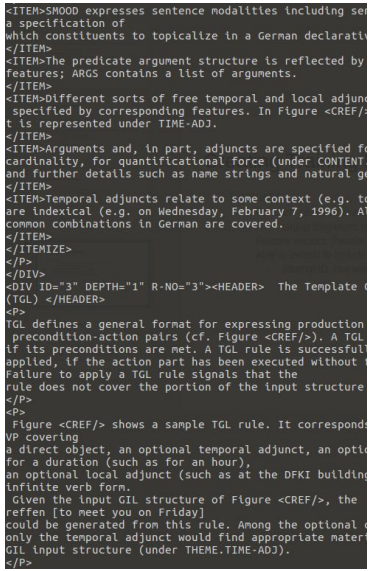
Preprocessing

- Download the documents
- Scrape XML files (-> abstract.txt, article.txt)
  - Extract <abstract> and <body> tag contents - leave out e.g. references, footnotes, acknowledgements
  - Convert XML tags to special tokens ({{HED}}, {{REF}}, {{EQN}})
- Parse sentences (-> *.sentences)
  - Custom boundary detection using regular expressions
  - E.g. `text = re.sub(r'(\s+(?!{{HED}}).*?)\s+({{HED}})', '\g<1>. {{HED}}', text)`
- Word Tokenization (-> *.tokens)
  - Replace numerics ({{NUM}}), symbols ({{SYM}})
  - Remove stopwords
  - Stemming

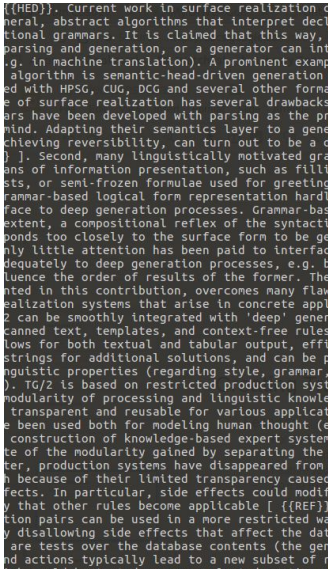
Training Data

- Features
  - Build a feature vector for each sentence in each document
  - Replace tokens with token IDs, grouping singletons together
  - Vector = [header_id] + [token_ids]
  - Can easily extend to include more features: journal_id, cue words, TF-IDF values, etc.
- Labels
  - TF-IDF vectorize each sentence in abstracts and articles
  - Compare article sentences with abstract sentences
  - Score each article sentence as maximum cosine similarity with abstract sentences
  - Rank and order scores by most similar to least, replace with sentence indices
- 80/10/10 split

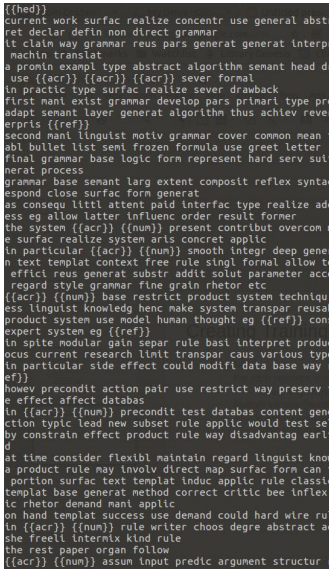
Preprocessing



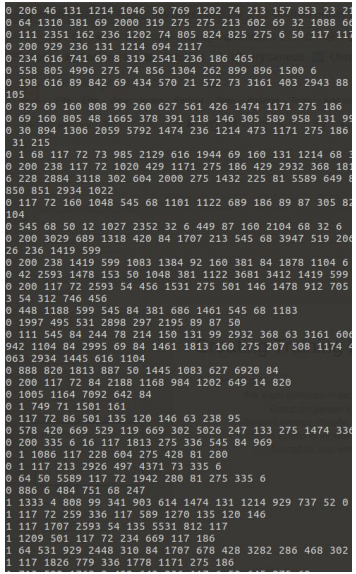
9605010.xml



article.txt



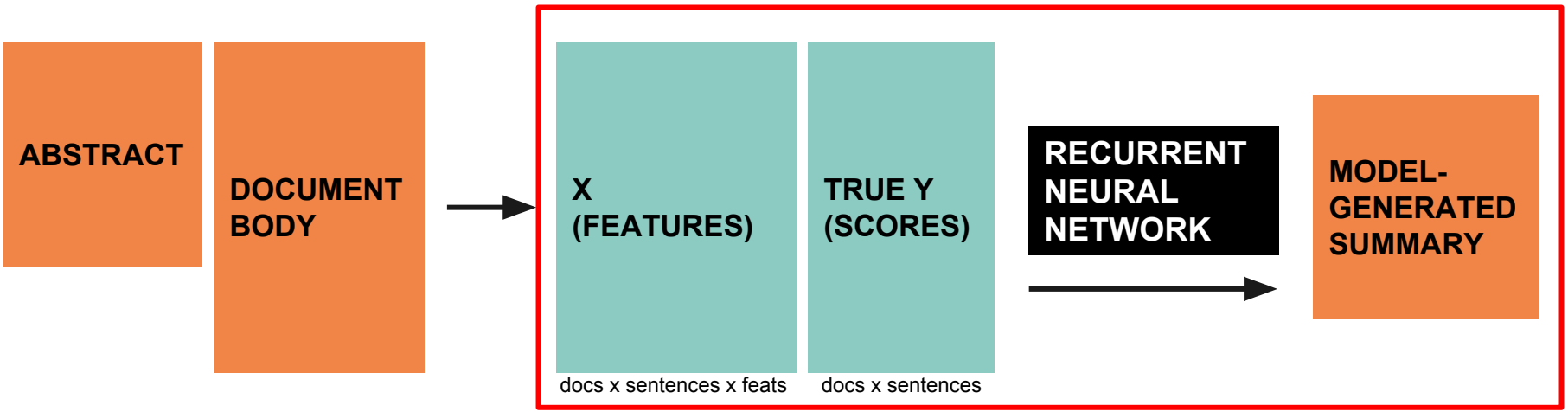
article.tokens



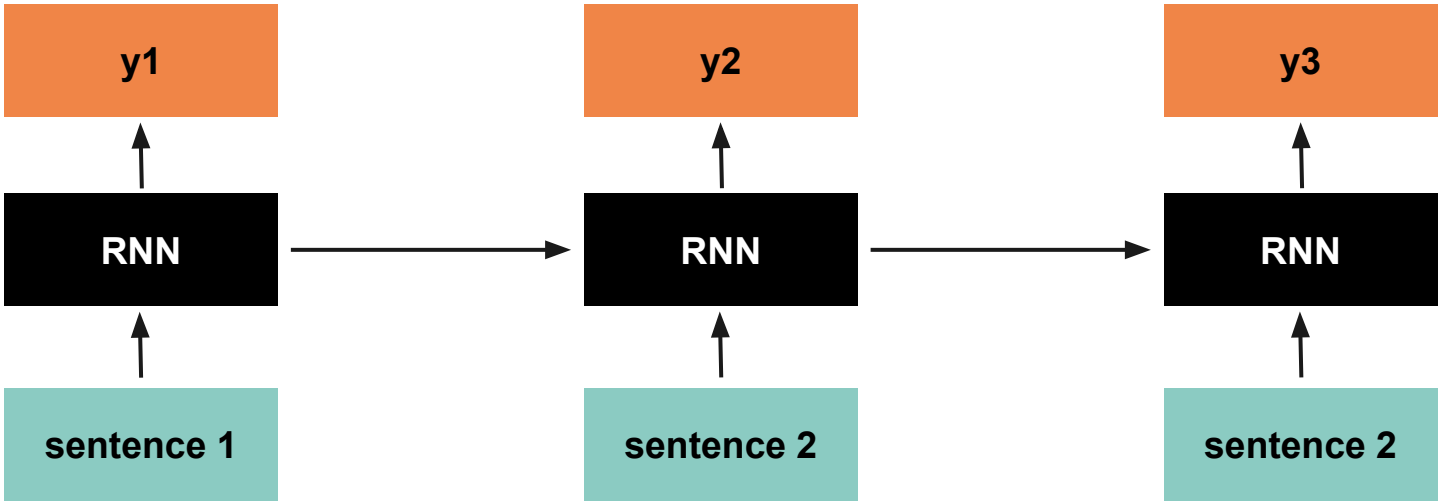
article.features

Approach

- Extraction-based
- Single document



Model Implementation



ABSTRACT

Current work in surface realization concentrates on the use of general abstract algorithms that interpret large reversible **grammars.** Only little attention has been paid so far to the many small and simple applications that require coverage of a small sublanguage at different degrees of sophistication. The system **TG 2** described in this paper **can be smoothly integrated with deep generation processes integrates canned text templates and context free rules into a single formalism allows for both textual and tabular output** and it can be parameterized according to **linguistic** preferences. These features are based on suitably restricted production system techniques and on a generic backtracking regime.

MODEL-GENERATED SUMMARY

Current work in surface realization concentrates on the use of general abstract algorithms that interpret declaratively defined non directional **grammars.** It is claimed that this way a grammar can be reused for parsing and generation or a generator can interpret different grammars eg in machine translation. Adapting their semantics layer to a generation algorithm and thus achieving reversibility can turn out to be a difficult enterprise. In particular **TG 2 can be smoothly integrated with deep generation processes integrates canned text templates and context free rules into a single formalism allows for both textual and tabular output** efficiently reuses generated substrings for additional solutions and can be parameterized according to **linguistic** properties regarding style grammar fine grained rhetorics etc.

Results

	Recall	Precision	F-Score
ROUGE-1	**0.04135**	**1**	**0.07941**
ROUGE-2	**0.04335**	**1**	**0.08309**
ROUGE-L	**0.08495**	**1**	**0.1566**

Future Work

- More feature engineering
  - Word embeddings? Metadata?
- Hyperparameter tuning
- Utilize the full MEDLINE/PubMed dataset
- Baselines

Questions?