Natural Language Processing for Extracting Crop Model Parameters from Literature

Maria Alexeeva[1] Vijaya Raj Joshi[2] Hubert Kanyamahanga[3] Isaac Kobby Anni[3]
Keith Alcock[1] Gerrit Hoogenboom[2] Mihai Surdeanu[1]

[1]University of Arizona, [2]University of Florida, [3]International Crops Research Institute for the Semi-Arid Tropics

Introduction

Decision Support System for Agrotechnology Transfer (DSSAT) is a software application program to simulate crop growth, development, and yield:

+ consists of models for more than 40 crops
- requires a lot of input data on crop cultivar, weather, soil, etc
- parameters are di□cult and expensive to obtain

Approach: automatically extract parameters from scientific publications and reports for the regions of interest

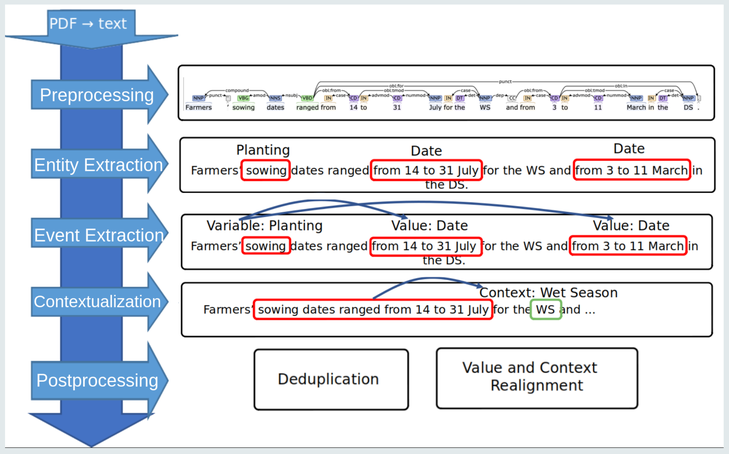
Types of information extracted: yield and fertilizer amounts, planting dates, area sizes, and more

System Overview

Our machine reading system allows us to automatically extract information from scientific papers and reports.

The system is built using the rule-based information extraction framework named Odin (Valenzuela-Escárcega et al. 2015) in combination with preprocessing (pdf to text conversion and text preprocessing) and post-processing (redundancy filtering, binarization, etc) components.

The code is available at https://github.com/clulab/habitus



File Preparation

PDF to text conversion: pdf2txt package (https://github.com/clulab/pdf2txt)
a scala wrapper for several PDF to text converters
methods to refine text (resolving unicode characters and ligatures, standardizing number formats, fixing line-breaks within words, etc)

NLP preprocessing: Processors package (https://github.com/clulab/processors):
sentence and work tokenization, chunking, syntactic parsing, etc

Information Extraction

We extract entities using domain-specific lexicons (e.g., crops and fertilizers) and rules based on the use of consistent language patterns (e.g., dates and generic agriculture-related terms, such as *sowing*). We then use the entities to construct events with rules of two types:

Surface rules use proximity and token-level patterns: here, the variable is a planting-related term, e.g., *planting* or *sowing*, separated from the date value by an open parenthesis; the rule can find multiple dates connected with commas or conjunctions; the date cannot immediately follow the variable.

```
- name: planting-date-parenthesis-1
  priority: ${rulepriority}
  label: PlantingDate
  action: splitIntoBinary
  type: token
  example: "The first two sowing periods (from 22 December to 1 January and
from 31 January to 22 February) are too short to sow all the fields."
  pattern: |
    @variable:Planting [! (mention = "Date")]{, 4} (?<trigger> [lemma="("])
@value:Date ((","  | "and")* @value:Date)*
```

Dependency rules use the sentence syntactic structure represented as a graph: here, we find the variable (a planting-related term) as the subject (nsubj) or a direct object (dobj) of trigger words, e.g., *range*, and the value (the date) can be found as a modifier (nmod).

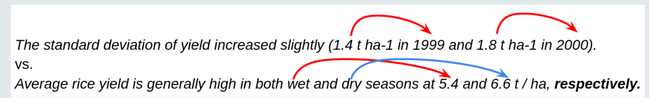
```
- name: planting-date-range-1
  priority: ${rulepriority}
  label: PlantingDate
  graph: "hybrid"
  example: "Farmers' sowing dates ranged from 14 to 31 July for the WS and
from 3 to 11 March in the DS ."
  pattern: |
    trigger = [lemma=/range|occur|be/ & tag=/^V|^J/]
    variable:Planting = /nsubj|dobj/
    value:Date = /nmod_from|nmod_between|nmod_like/
```

Contextualization

For every event we extract, we provide available context, e.g., location, season, date, crop, fertilizer, etc. This allows for better filtering of extractions in downstream tasks. For instance, for planting date extractions, knowing the associated crop can help the user select planting events for specific types of crops that they need to model.

Post-processing

Value and Context Realignment: in contexts where there are multiple variables and values, we make sure that variable-value pairs are properly aligned:



Deduplication: we pick longer or more complete events out of overlapping ones.

Evaluation

We evaluate our system on two sets of papers related to agriculture in Senegal.

Domain	Papers	Extractions	Accuracy
Rice	6	334	0.84
Peanut	12	224	0.95

Limitations and Future Work

The approach can be extended to new crops and regions:
rice ! peanut took 14 hours (two for analysis and 12 for debugging)
the process should become less time consuming with every iteration

The approach can be extended to extract other types of information if they are expressed using consistent language patterns (see sample rules).

The quality of extractions may depend on the quality of the papers used as input:
papers from recent years have a better chance of successful PDF to text conversion
papers on relevant topics require less filtering of irrelevant extractions

Sample Output

sentenceText	label	mentionText	publicationYear	location	country	date	process	season	value_norm	variable_text
Potential yield (limited by radiation and temperature only) is about 9 t ha-1 for WS sowing in July and about 9-10 t ha-1 for dry season (DS) sowing in February in the Senegal River delta (Dingkuhn and Sow , 1997) .	YieldAmount	yield (limited by radiation and temperature only) is about 9 t ha-1	2002	Senegal River	SN	July	harvesting	WS	9.0 t/ha	yield

Acknowledgements

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under the Habitus program. Maria Alexeeva and Mihai Surdeanu declare a financial interest in lum.ai. This interest has been properly disclosed to the University of Arizona Institutional Review Committee and is managed in accordance with its conflict of interest policies.