Eidos, INDRA, & Delphi: From Free Text to Executable Causal Models

Rebecca Sharp, Adarsh Pyarelal, Benjamin M. Gyori[†], Keith Alcock,
Egoitz Laparra, Marco A. Valenzuela-Escárcega, Ajay Nagesh, Vikas Yadav,
John A. Bachman[†], Zheng Tang, Heather Lent, Fan Luo, Mithun Paul,
Steven Bethard, Kobus Barnard, Clayton T. Morrison, Mihai Surdeanu
University of Arizona, Tucson, Arizona, USA
[†]Harvard Medical School, Boston, Massachusetts, USA
`{bsharp,adarsh}@email.arizona.edu`

Abstract

Building causal models of complicated phenomena such as food insecurity is currently a slow and labor-intensive manual process. In this paper, we introduce an approach that builds executable probabilistic models from raw, free text. The proposed approach is implemented through three systems: Eidos[1], INDRA[2] and Delphi[3]. Eidos is an open-domain machine reading system designed to extract causal relations from natural language. It is rule-based, allowing for rapid domain transfer, customizability, and interpretability. INDRA aggregates multiple sources of causal information and performs assembly to create a coherent knowledge base and assess its reliability. This assembled knowledge serves as the starting point for modeling. Delphi is a modeling framework that assembles quantified causal fragments and their contexts into executable probabilistic models that respect the semantics of the original text and can be used to support decision making.

1 Introduction

Food insecurity is an extremely complex phenomenon that affects wide swathes of the global population, and is governed by factors ranging from biophysical variables that affect crop yields, to social, economic, and political factors such as migration, trade patterns, and conflict.

For any attempt to combat food insecurity to be effective, it must be informed by a model that comprehensively considers the myriad of factors influencing it. Furthermore, for analysts and decision makers to truly trust such a model, it must be *causal* and *interpretable*, as in, it must provide a mechanistic explanation of the phenomenon, rather than just being a black-box statistical construction.

Currently, however, these models are hand-built for each new situation and require many months to construct, resulting in long delays for much-needed interventions.

Here we propose an end-to-end system that combines open-domain information extraction (IE) with a quantitative model-building framework, transforming free text into executable probabilistic models that capture complex real-world systems. All code and data described here is open-source and publicly available, and we provide a short video demonstration[4].

Contributions:

(1) We introduce Eidos, a rule-based open-domain IE system that extracts causal statements from raw text. To maximize domain independence, Eidos is largely unlexicalized (with the exception of causal cues such as *promotes*), and implements a top-down approach where causal interactions are extracted first, followed by the participating concepts, which are grounded with specific geospatial and temporal contexts for model contextualization. Eidos also extracts quantifiable adjectives (e.g. *significant*) that can be used to form a bridge between qualitative statements and quantitative modeling.

(2) We describe an extension of the Integrated Network and Dynamical Reasoning Assembler (INDRA, Gyori et al., 2017), an automated knowledge and model assembly system which implements interfaces to Eidos and multiple other machine reading systems. Originally developed to assemble models of biochemical mechanisms, we generalized INDRA to represent general causal influences as INDRA Statements, and load a taxonomy of concepts to align related Statements from multiple readers and documents.

(3) We introduce Delphi, a Bayesian modeling

---

[1] https://github.com/clulab/eidos
[2] https://github.com/sorgerlab/indra
[3] https://github.com/ml4ai/delphi

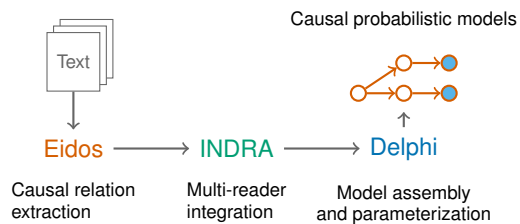[4] https://youtu.be/FcLEJej1uAg

Figure 1: Overall architecture showing the flow of information between the systems.

framework that converts the above statements into executable probabilistic models that respect the semantics of the source text. These models can help decision-makers rapidly build intuition about complicated systems and their dynamics. The proposed framework is interpretable due to its foundation in rule-based IE and Bayesian generative modeling.

Architecture: In Fig. 1, we show a high-level depiction of the information flow pipeline. First, natural language texts serve as inputs to Eidos, which performs causal relation extraction, grounding, and spatiotemporal contextualization. The extracted relations are subsequently aggregated by INDRA into data structures called INDRA Statements for downstream modeling. These serve as an input to Delphi, which assembles a causal probabilistic model from them.

2 Causal Information Extraction

Eidos was designed as an open-domain IE system (Banko et al., 2007) with a top-down approach that allows us to not be limited to a fixed set of concepts, as determining this set across multiple distinct domains (e.g., agronomy and socioeconomics) is close to impossible. First, we find trigger words signaling a relation of interest and then extract and expand the participating concepts (2.1), link these concepts to a taxonomy (2.2), and annotate them with temporal and spatial context (2.3).[5,6]

In addition to an API that can be used for machine reading at scale, Eidos has a webapp that provides users a way to see what rules were responsible for the extracted content, as well as brat visualizations (Stenetorp et al., 2012) of the output,

facilitating rapid development of the interpretable rule-grammars.

2.1 Reading Approach

To understand our top-down approach, let us consider the individual steps involved in processing the following sentence: *The significantly increased conflict seen in South Sudan forced many families to flee in 2017.*

(1) We begin by preprocessing the text with dependency syntax using Stanford CoreNLP (Manning et al., 2014) and the processors library[7].

(2) Then, Eidos finds any occurrences of *quantifiers* (gradable adjectives and adverbs). These are common in the high-level texts relevant to food insecurity, such as reports from UN agencies and nonprofits, but they are difficult to use in quantitative models without additional information. In the example above, the word *significantly* is found as a quantifier of *increased*. Delphi uses these quantifiers to construct probability density functions using the crowdsourced data of Sharp et al. (2018), as detailed in 4.

(3) Next, Eidos uses a set of trigger words to find causal and correlational relations with an Odin grammar (Valenzuela-Escárcega et al., 2016). Odin is an information extraction framework which includes a declarative language supporting both surface and syntactic patterns and a runtime system. Eidos's grammar was based in part on the biomedical grammar developed by Valenzuela-Escárcega et al. (2018) but adapted to the open domain and our representation of concepts. This rule grammar is fully interpretable and easily editable, allowing users to make modifications without needing to retrain a complex model. In the example sentence from earlier, the extraction of a causal relation would be triggered by the word *forced*, with *conflict* and *families* identified as the initial cause and effect, respectively.

(4) The initial cause and effect are then expanded using dependency syntax following the approach of Hahn-Powell et al. (2017). Namely, from each of the initial arguments, we traverse outgoing dependency links to expand the arguments into their dependency subgraph. Here, the resulting arguments are *significantly increased conflict seen in South Sudan* and *many families to flee in 2017.*

---

[5]This has some similarities to FrameNet (Baker et al., 1998), whose Causation frame has targets (triggers) and frame elements (participating concepts) that are associated with a taxonomy (the FrameNet hierarchy). In our case, the concepts come from a domain-specific taxonomy.

[6]We assume here that causal relations are specified within sentences rather than across sentences at the document level, and that the concepts involved in the causal relations can be linked to an appropriate taxonomy.
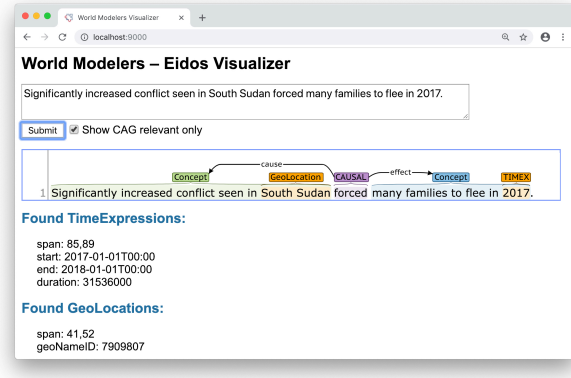
[7]https://github.com/clulab/processors

Figure 2: Screenshot of the Eidos output for the running example sentence, visualized in Eidos's webapp.

(5) Relevant state information is then added to the expanded concepts. Representing the polarity of an influence on the causal relation edge (i.e., in terms of *promotes* or *inhibits*) can be lossy, so Eidos instead uses concept states (i.e., concepts can be increased, decreased, and/or quantified). In the example above, Eidos marks the concept pertaining to conflict as being increased and quantified. If desired, the promotion/inhibition representation with edge polarity can be straightforwardly recovered. The final output of the Eidos system for the running example sentence, as displayed in the Eidos webapp, is shown in Fig. 2.

2.2 Concept linking

The Eidos reading system, with its top-down approach, was designed to keep extracted concepts as close to the text as possible, intentionally allowing downstream users to make decisions about event semantics depending on their use cases. As a result, linking concepts to a taxonomy becomes critical for preventing sparsity.

Eidos's concept linking is based on word-embedding similarities. A given concept (with stop words removed), is represented by the average of the word embeddings for each of its words. A vector for each node in the taxonomy is similarly calculated (using the provided "examples" for the node), and the taxonomy node whose vector is closest to the concept vector is considered to be the grounding. In practice, Eidos returns the top k groundings, allowing for downstream disambiguation. The concept linking strategy is modular and allows for grounding to any taxonomy provided in the human-readable YAML format. With this method, Eidos is able to link to an arbitrary number of taxonomies, at both high and low levels of abstraction.

2.3 Temporal and geospatial normalization

Time normalization   The context surrounding the extractions is often critical for downstream reasoning. Eidos integrates the temporal parser of Laparra et al. (2018) that uses a character recurrent neural network to identify time expressions in the text which are then linked together with a set of rules into semantic graphs which follow the SCATE schema (Bethard and Parker, 2016) and can be interpreted using temporal logic to obtain the intervals referred to by the time expressions.

After the time expressions are identified and normalized, an Odin grammar attaches them to the causal relations extracted by Eidos. If the document creation time is provided, it is also parsed by our model and used as the default temporal attachment for those causal relations without a temporal expression in their close context.

Geospatial normalization   Eidos's geospatial normalization module (Yadav et al., 2019) has two components: a detection component consisting of the word-level LSTM named entity recognition (NER) model of Yadav and Bethard (2018), and a normalization component which implements population heuristics (i.e., selecting the most populous location (Magge et al., 2018)) and filters using a distance-based heuristic (Magge et al., 2018).

3 Assembly of causal relations

The output of Eidos is processed by INDRA into a collection of INDRA Statements, each of which represents a causal influence relation. INDRA is also able to process the output of multiple other reading systems that extract causal relations from text (these systems are not described in detail here). INDRA implements input processor modules to extract standardized Statements from each reading system. A Statement represents a causal influence between two Concepts (a subject and an object), each of which is linked to one or more taxonomies (see Section 2.2). The Statement also captures the polarity and magnitude of change in both subject and object, if available. Finally, one or more Evidences are attached to each Statement capturing provenance (reader, document, sentence) and context (time, location) information. This common representation establishes a link between diverse knowledge sources and several model formalism endpoints.

Given the attributes of each Statement and a tax-

onomy to which Concepts are linked, INDRA creates a Statement graph whose edges capture (i) redundancy between two Statements (ii) hierarchical refinement between two Statements, and (iii) contradiction between two Statements. Statements that are redundant, or in other words, capture the same causal relation, are merged and their evidences are aggregated. A probability model is then used which captures the empirical precision of each reader to calculate the overall support (a "belief" score) for a Statement given the joint probability of correctness implied by the evidence. As a seed to this probability model, INDRA loads empirical precision values collected via human curation for each Eidos rule. INDRA exposes a collection of methods to filter Statements that can be composed to form a problem-specific assembly pipeline, including (i) filtering by Statement belief and Concept linking accuracy (ii) filtering to more general or specific Statements (with respect to a taxonomy), and (iii) filtering contradictions by belief. INDRA also exposes a REST API and JSON-based serialization of Statements.

INDRA contains multiple modules that can assemble Statements into causal graphs (for visualization or inference) and executable ODE models. In the architecture presented here, Delphi (our Bayesian modeling framework) takes INDRA Statements directly as input, and serves as a probabilistic model assembly system.

4   Causal Probabilistic Models from Text

Statements produced by INDRA are assembled by Delphi into a structure called a *causal analysis graph*, or CAG. In Fig. 3, we show the CAG resulting from our running example sentence (cell [1]). The node labels (*conflict* and *human migration*) in the CAG correspond to entries in the high-level taxonomy that the concepts have been grounded to.

Representation   We represent abstract concepts such as *conflict* and *human migration* as real-valued *latent* variables in a dynamic Bayes network (DBN) (Dagum et al., 1992), and the indicators corresponding to these concepts as *observed* variables. By an indicator, we mean a tangible quantity that serves as a proxy for the abstract concept[8]. For example, the variable *Net migration* (as defined in World Bank (2018)) is one of several indicators for the

---

[8]Note that these are not the same as the indicator random variables encountered in probability theory.
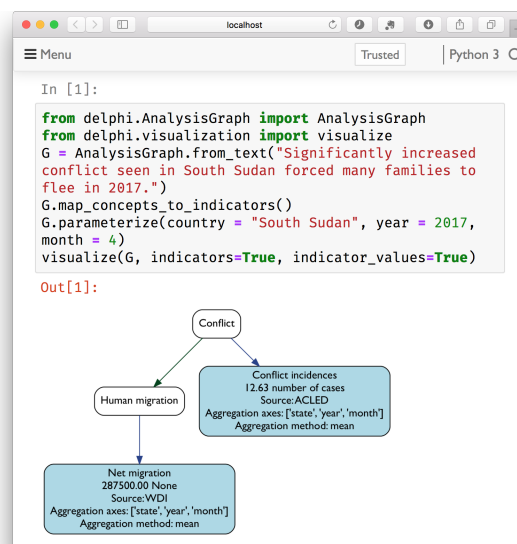

Figure 3: Construction of a causal analysis graph from the running example sentence.

concept of *human migration*. To capture the uncertainty inherent in interpreting natural language, we take the transition model of the DBN itself to be a random variable with an associated probability distribution. We interpret sentences about causal relations as saying something about the functional relationship between the concepts involved. For example, we interpret the running example sentence as giving us a clue about the shape of $\partial(\text{human migration})/\partial(\text{conflict})$.

Assembly   To assemble our model, we do the following[9]:

(1) We construct the aforementioned distribution over the transition model of the DBN using the extracted polarities of the causal relations as well as the gradable adjectives associated with the concepts involved in the relations. The transition model is a matrix whose elements are random variables representing the coefficients of a system of linear differential equations (Guan et al., 2015), with distributions obtained by constructing a Gaussian kernel density estimator over Cartesian products of the crowdsourced responses collected by Sharp et al. (2018) for the adjectives in each relation.

(2) To provide more tangible results, we map the abstract concepts to indicator variables for which we have time series data. This data is gathered from a number of databases, including but not limited to

---

[9]The complete mathematical details of the model assembly process are out of the scope of this paper, but can be found at: http://vision.cs.arizona.edu/adarsh/Arizona_Text_to_Model_Procedure.pdf

```
In [2]:
%matplotlib inline
%config InlineBackend.figure_format = 'retina'
from delphi.jupyter_tools import run_experiment
run_experiment(G, "UN/events/human/conflict", 0.1, 4)
```

Figure 4: Results of conditional forecasting experiment with CAG built from example sentence.

the FAOSTAT (Food and Agriculture Organization of the United Nations, 2018) and the World Development Indicators (World Bank, 2018) databases. The mapping is done using the OntologyMapper tool in Eidos that uses word embedding similarities to map entries in the high-level taxonomy to the lower-level variables in the time series data.

(3) Then, we associate values with indicators using a parameterization algorithm that takes as input some spatiotemporal context, and retrieves values for the indicators from the time series data, falling back to aggregation over a (configurable) set of aggregation axes in order to prevent null results. In Fig. 3, we show the indicators automatically mapped to the conflict and human migration nodes (*conflict incidences* and *net migration*, respectively) and their values for the spatiotemporal context of South Sudan in April 2017.

Conditional forecasting Once the model is assembled, we can run experiments to obtain quantitative predictions for indicators, which can build intuitions about the complex system in question and support decision making. The outputs take the form of time series data, with associated uncertainty estimates. An example is shown in Fig. 4, in which we investigate the impact of increasing conflict on human migration using our model, with $\partial(\text{conflict})/\partial t = 0.1e^{-t}$. The predictions of the model reflect (i) the semantics of the source text (increased conflict leads to increased migration)

and (ii) the uncertainty in interpreting the source sentence. The confidence bands in the lower plot reflect the distribution of the crowdsourced gradable adjective data.

5 Assessment

We are currently in the process of developing a framework to quantitatively evaluate the models assembled using this pipeline, primarily via backcasting. However, the systems have been qualitatively evaluated by MITRE, an independent performer group in the World Modelers program charged with designing and conducting evaluations of the technologies developed. For the evaluation, a causal analysis graph larger than the toy running example in this paper (≈ 20 nodes) was created and executed. Noted strengths of the system include the ability to *drill down* into the provenance of the causal relations, the integration of multiple machine readers, and the plausible directionality of the produced forecast (given the sentences used to construct the models). Some limitations were also noted, i.e., that the initialization and parameterization of the models were somewhat opaque (which hindered explainability) and some aspects of uncertainty are captured by the readers but not fully propagated to the model. We are actively working on addressing both of these limitations.

6 Conclusion

Complex causal models are required in order to address key issues such as food insecurity that span multiple domains. As an alternative to expensive, hand-built models which can take months to years to construct, we propose an end-to-end framework for creating executable probabilistic causal models from free text. Our entire pipeline is interpretable and intervenable, such that domain experts can use our tools to greatly reduce the time required to develop new causal models for urgent situations.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 17th international conference on Computational Linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Steven Bethard and Jonathan Parker. 2016. A semantically compositional annotation scheme for time normalization. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA). [Acceptance rate 60%].

Paul Dagum, Adam Galper, and Eric Horvitz. 1992. Dynamic network models for forecasting. In Didier Dubois, Michael P. Wellman, Bruce D'Ambrosio, and Phillipe Smets, editors, *Uncertainty in Artificial Intelligence*, pages 41 – 48. Morgan Kaufmann.

Food and Agriculture Organization of the United Nations. 2018. FAOSTAT Database.

Jinyan Guan, Kyle Simek, Ernesto Brau, Clayton T. Morrison, Emily Butler, and Kobus Barnard. 2015. Moderated and drifting linear dynamical systems. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2473–2482, Lille, France. PMLR.

Benjamin M. Gyori, John A. Bachman, Kartik Subramanian, Jeremy L. Muhlich, Lucian Galescu, and Peter K. Sorger. 2017. From word models to executable models of signaling networks using automated assembly. *Molecular Systems Biology*, 13(11).

Gus Hahn-Powell, Marco A. Valenzuela-Escárcega, and Mihai Surdeanu. 2017. Swanson linking revisited: Accelerating literature-based discovery across domains using a conceptual influence graph. *Proceedings of ACL 2017, System Demonstrations*, pages 103–108.

Egoitz Laparra, Dongfang Xu, and Steven Bethard. 2018. From characters to time intervals: New paradigms for evaluation and neural parsing of time normalizations. *Transactions of the Association for Computational Linguistics*, 6:343–356.

Arjun Magge, Davy Weissenbacher, Abeed Sarker, Matthew Scotch, and Graciela Gonzalez-Hernandez. 2018. Deep neural networks and distant supervision for geographic location mention extraction. *Bioinformatics*, 34(13):i565–i573.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Rebecca Sharp, Mithun Paul, Ajay Nagesh, Dane Bell, and Mihai Surdeanu. 2018. Grounding gradable adjectives through crowdsourcing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics.

Marco A. Valenzuela-Escárcega, Özgün Babur, Gus Hahn-Powell, Dane Bell, Thomas Hicks, Enrique Noriega-Atala, Xia Wang, Mihai Surdeanu, Emek Demir, and Clayton T. Morrison. 2018. Large-scale automated machine reading discovers new cancer driving mechanisms. *Database: The Journal of Biological Databases and Curation*.

Marco A. Valenzuela-Escárcega, Gustave Hahn-Powell, and Mihai Surdeanu. 2016. Odin's runes: A rule language for information extraction. In *Proceedings of the 10th edition of the Language Resources and Evaluation Conference (LREC)*.

World Bank. 2018. World Development Indicators Database.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158.

Vikas Yadav, Egoitz Laparra, Ti-Tai Wang, Mihai Surdeanu, and Steven Bethard. 2019. University of arizona at semeval-2019 task 12: Deep-affix named entity recognition of geolocation entities. In *Proceedings of The 13th International Workshop on Semantic Evaluation*, Minneapolis, USA. Association for Computational Linguistics.