Claim Extraction via Subgraph Matching over Modal and Syntactic Dependencies

Benjamin Rozonoyer[1], Michael Selvaggio[2], David Zajic[2] and Ilana Heintz[3]

[1]University of Massachusetts Amherst
brozonoyer@cs.umass.edu
[2]Raytheon BBN Technologies
{michael.selvaggio,david.m.zajic}@rtx.com
[3]Synoptic Engineering
ilana@synopticengineering.com

Abstract

We propose the use of modal dependency parses (MDPs) aligned with syntactic dependency parse trees as an avenue for the novel task of claim extraction. MDPs provide a document-level structure that links linguistic expression of events to the conceivers responsible for those expressions. By defining the event-conceiver links as claims and using subgraph pattern matching to exploit the complementarity of these modal links and syntactic claim patterns, we outline a method for aggregating and classifying claims, with the potential for supplying a novel perspective on large natural language data sets.

Abstracting away from the task of claim extraction, we prototype an interpretable information extraction (IE) paradigm over sentence- and document-level parse structures, framing inference as *subgraph matching* and learning as *subgraph mining*. We make our code open-sourced at https://github.com/BBN-E/nlp-graph-pattern-matching-and-mining.

1 Introduction

A promise of natural language processing (NLP) tools is to bring fast understanding of large corpora of unstructured data. This has been achieved through tasks such as summarization (Prudhvi et al., 2020), knowledge base population (Glass and Gliozzo, 2018), and question-answering systems (Arbaaeen and Shah, 2020), among others. These outcomes provide different views into the chosen data source, highlighting aspects such as event timelines, known and novel relationships between entities, causality, and others. A less explored view of unstructured data may take the form of a Claim Bank, in which NLP tools provide access to a set of differentiated claims expressed by explicit claimants within a document corpus.

We will define a claim as an assertion that is explicitly linked to a source. The source may be a person or an organization. The source may be explicit or implicit; a common example of the latter case is when the author of a document is the source of a claim and is defined as part of the metadata, but is not explicit in the document content. Our goal is to automatically identify claims in, and learn claim structures from, natural language text. Such claims could then be the impetus for claim verification, clustering, provenance graph generation, etc., as described below.

2 Related Work

2.1 Claim Extraction

Recent work in claim verification is closely related to the effort at hand. In particular, Zhang et al. (2020) build a provenance graph for claims, linking each claim to its likely sources. The "query" claims are derived from the opinion corpus developed in Choi et al. (2005). In Zhang et al. (2020), sentences relevant to a query claim are first retrieved from a variety of documents, and the implicit (author) and explicit (named) sources identified via a Textual Entailment task, followed by a classification task to identify the relationship between source and statement. Thus, a set of related claims is derived from an original, provided claim. The provenance graph built from these efforts is used to identify supporting, contradictory, or neutral relationships between statements relating to a claim.

Similarly, FEVER (Thorne et al., 2018), HoVer (Jiang et al., 2020) and WiCE (Kamoi et al., 2023) are open source datasets of related facts for the NLP community to make shared progress on claim verification. The relationship between facts (the term is used interchangeably with claims in these studies) is of interest, rather than their relationship to sources. Facts or claims enter the corpus through a crowdsourced annotation effort or automatically from Wikipedia as in the case of WiCE.

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 122–135
June, 2023.

An earlier important work (Choi et al., 2005) identifies sources responsible for opinions, emotions, and sentiment through a dataset collection effort. The authors test an automated approach for detecting these relationships with conditional random fields, as an information extraction task.

These works do not give a definition of a claim, though the intuitive notion seems to be that a claim is a declarative statement that could be given a truth value. In the present work, we add the constraint that a claim must be associated with a claimer. The DARPA Active Interpretation of Disparate Alternatives (AIDA) program (Onyshkevych, 2017; Hovy, 2020) has helpfully defined the truth-valued statement as the "inner claim," and its epistemic or sentimental association with a claimer as the "outer claim" – together, the inner and outer claim constitute a claim that can be compared to others in terms of support, contradiction, relevance, etc.

The NEWSCLAIMS benchmark (Reddy et al., 2022) is the most recent and closely related parallel work on claim detection since it also stems from AIDA definitions of claims — the released dataset consists of claims annotated over the subset of articles from the LDC corpus LDC2021E11 related to COVID-19 (cf. §6.2). We recommend it as a reference for the task at hand despite slight terminological differences; unlike the methodology explored here, the authors experiment with zero-shot and prompt-based baselines.

The goal of this work is to introduce a novel way of automatically identifying claims according to this definition; in particular, we identify that inner and outer claims are often, but not always, identifiable through a sentence-level predicate-argument structure. In cases where the outer and inner claim are expressed over multiple sentences, a document-level structure must be accessed to reveal the relationship. We describe an algorithm for combining structural and semantic information from the sentence and document levels to automatically identify claims. This effort might be seen as a precursor to the studies described above; we aim to automate the initial task of finding the claims for analysis.

2.2 Subgraph Matching and Mining

DotMotif (Matelsky et al., 2021), a declarative library for identifying and extracting frequent motifs from large graphs of connectomes, begs the exploration of an analogous approach in NLP, where text can be viewed as the tip of the iceberg in a rich network of underlying syntactic, semantic and pragmatic parses or "deep structures". We describe our algorithmic approach to subgraph isomorphism in §4.7; "soft", neural implementations such as NeuroMatch of (Ying et al., 2020a) use graph neural network (GNN) encoders to achieve 100x speedup over traditional combinatorial approaches. While the latter GNN architectures were designed with molecular graphs in mind, Marcheggiani and Titov (2017); Bastings et al. (2017); Nguyen and Grishman (2018); Rozonoyer (2021) successfully tailored the GCN (Kipf and Welling, 2016) and GAT (Veličković et al., 2017) architectures to encode dependency syntax for semantic role labeling, neural machine translation, event detetion, and AMR parsing, respectively.

3 Linguistic Motivation

Our approach to claim extraction leverages Modal Dependency Parsing (MDP), a document-level annotation scheme for modality introduced by Vigus et al. (2019) and adopted by Yao et al. (2021) to crowdsource a dataset and train a neural parser. We use MDP in conjunction with sentence-level syntactic dependency parses. Document-level MDP restricts the extraction space to a pool of potential claims that include inter-sentence inner/outer claim relationships. Sentence-level dependency parses provide the grammatical structure that allows us to further constrain the pool of potential claims to those that match our analysis of the clausal structure of claims.

3.1 Modal Dependency Parsing

MDPs provide a document-level structure that links events to their conceivers, including the author conceiver, which is at the root of the document. Furthermore, the MDP edges provide epistemic values of the relationship between the conceiver and the event: whether the conceiver is certain that the event occurred, uncertain, or believes the event did not occur. This provides essential information to understanding how claimers and claims interact.

In Figure 1, we show the MDP for sentence (1) below, cited and manually annotated in Vigus et al. (2019):

(1) [About 200 people were believed **killed** and 1,500 others were **missing** in the Central Philippines on Friday when a **landslide buried** an entire village], **the Red Cross** said.

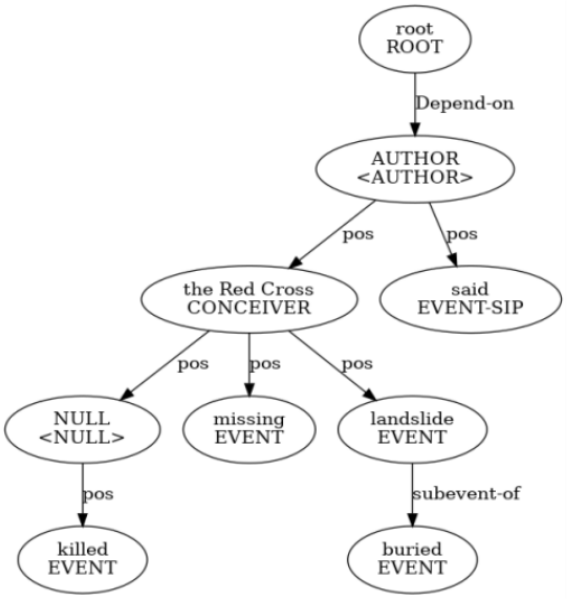

Figure 1: MDP for sentence (1)

In this sentence-sized document, the Red Cross is a conceiver making a claim about the "missing", "landslide", and "buried" events with positive (certain) epistemic value. The passive construction *believed killed* introduces a null conceiver ("believed by *whom*?") in the mental space of the governing "the Red Cross". The source introducing predicate "said" is a clue to attributing the other events in this sentence to "the Red Cross", while the saying event itself can be attributed to the author of the document with positive epistemic value.

3.2 Claim Structure in Syntax

We observe that there are typical syntactic structures associated with our intuition of claim. Consider the sentence:

(2) [**US officials** $_{\text{nsubj}}$] [said $_{\text{SIP}}$] [numerous social media sites **launched** an effort to spread misinformation $_{\text{ccomp}}$].

"Said" serves as the source introducing predicate (SIP) that syntactically governs the claimant US officials (its noun subject) and its clausal complement, the semantic "inner claim."

We can convey the exact same claim information in an altogether different lexical and syntactic structure:

(3) Numerous social media sites **launched** an effort to spread misinformation, [[according to $_{\text{case}}$] **US officials** $_{\text{obl}}$].

The conceiver *US officials*, which used to be the subject of the main clause, is now an oblique argument of the inner claim's predicate *launched*. Our claim-finding approach aims to account for this variation in expression, among other syntactic possibilities, and the presence of inter-sentence claim/claimant relationships.

3.3 Complementarity of Modal and Syntactic Structure

The modal and syntactic structures inform each other to the extent that the former is responsible for providing the evidential semantics of who-claims-what-with-what-certainty, while the latter comprises the clausal or grammatical form to express these semantic relations.

To see that the relationship between these structures is not trivial, consider the sentence:

(4) [[According to $_{\text{case}}$] the Pythagorean theorem $_{\text{obl}}$], the square of the hypotenuse equals the sum of the squares of the sides.

Although this sentence has the same syntactic skeleton as the previous sentence, the Pythagorean theorem is nevertheless not a viable claimant semantically. It is the job of the modal dependency parser to predict that *US officials* is a conceiver but the *Pythagorean theorem* is not; the basic syntactic structure does not provide us with the lexicosemantic information to tell apart potential claimants from non-claimants.

However, Modal Dependency Parsing is a budding technology that is not entirely robust, and due to the broad event annotation scheme in the crowdsourced dataset (Yao et al., 2021) the parsers in practice tend to be high-recall and low precision, such that automatically-generated MDPs are expected to contain false positive edges. This provides a motivation to use more reliable dependency parse (DP) clausal structures for pruning the original claim space consisting of every possible *Conceiver-Event* edge. In one preliminary study, a seedling set of DP patterns allowed us to focus our attention on 34% of the proposed MDP edges.

4 Claim Extraction via Subgraph Pattern Matching

In order to algorithmically exploit this synergy between evidential/modal and clausal/syntactic structure, we explore the task of claim extraction within the framework of subgraph isomorphism.

4.1 Problem Definition as Subgraph Matching

Our approach entails matching query graphs corresponding to basic *claim structures* against a composition of the document-level modal parse and sentence-level syntactic dependency parses.

The document-level MDP is a directed acyclic graph, and the sentence-level DP is a directed tree between token nodes. To compose the MDP and per-sentence DPs into a single graph, we construct directed edges from the MDP nodes to their corresponding token nodes, resulting in a composition that is also a directed acyclic graph for the entire document.

4.2 Node and Edge Types

Node Types. We define two node types in the composed modal-syntactic graph:

- ***modal nodes*** to represent abstract *Conceiver* and *Event* nodes
- ***token nodes*** to represent words

Edge Types. We define three edge types:

- ***modal edges*** connect conceivers with conceivers or conceivers with events, and are labeled with modal relations e.g. *pos*, *neg*, *pp*[1]
- ***syntax edges*** connect tokens, and are labeled with syntactic dependency relations e.g. *nsubj*, *ccomp*, *advcl*
- ***modal-token edges*** connect modal nodes with token nodes, e.g. the *Conceiver* node corresponding to the Washington Post entity in the MDP with every token in the multiword expression "The Washington Post".

4.3 Graph Structure Formalization

We formalize the definition of graph structures in our domain as $G = (V, E, \phi, \psi)$ where V is the set of nodes, E is the set of edges, and ϕ and ψ are the node and edge type assignment functions, respectively, for the edges described in §4.2:

$$\phi : V \rightarrow \{\textit{modal, token}\}$$
$$\psi : E \rightarrow \{\textit{modal, syntax, modal-token}\}$$

Additional categorical node and edge feature functions are contingent on the node's ϕ or edge's ψ type, respectively, as shown in Table 1.

[1]partially positive

Node/Edge Type	Feature Function(s)
$\phi(n) = \textit{modal}$	$\mu(n) \in \{\textit{Conceiver}, \textit{Event}\}$
$\phi(n) = \textit{token}$	$\tau(n) =$ text of token $\upsilon(n) =$ UPOS of token $\chi(n) =$ XPOS of token
$\psi(e) = \textit{modal}$	$\mu(e) \in \{\textit{pos}, \textit{neg}, \textit{neut}\}$
$\psi(e) = \textit{syntax}$	$\sigma(e) =$ syntactic relation
$\psi(e) = \textit{modal-token}$	no further typing

Table 1: Node and edge feature functions for composed modal-syntactic graphs

4.4 Document Digraph

We produce a document-level graph in the NetworkX[2] API by 1) storing the document-level modal dependency parse as a NetworkX DiGraph (directed graph), 2) storing each sentence's syntactic dependency parse as a NetworkX DiGraph, and 3) composing the graphs into a single NetworkX DiGraph. We connect the document-level modal nodes with the sentence-level token nodes via modal-token token edges described in §4.2.

4.5 Pattern Digraphs

We create pattern NetworkX DiGraphs as small graph structures that combine the core elements of the syntactic and modal structures constituting a claim. We use a subgraph isomorphism algorithm (§4.7) to match these claim pattern digraphs against the document digraph in order to discover claims. Each pattern digraph is accompanied by a node-match and edge-match function (§4.6) that allows a pattern node/edge to be underspecified with respect to irrelevant features but still match a fully annotated node/edge in the document digraph. In effect, we can view the pattern digraph as a "query" graph that is as generic a structure as possible for the intuition of the claim structure we are trying to match. The pattern digraphs expressing the two claim structures in sentences (2) and (3) are shown in Figures 2 and 3.

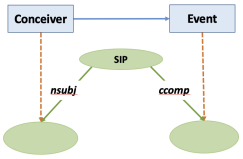

Figure 2: *ccomp* pattern graph

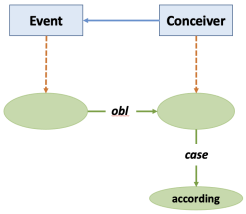

Figure 3: *according_to* pattern graph

[2]https://networkx.org/

4.6 Node-Match and Edge-Match Functions

In addition to simply defining modal-syntactic pattern structures to match claims in the document, as mentioned in §4.5, we define **node-match** and **edge-match** functions (with a Boolean return value) that allow us to specify custom criteria when checking for node or edge equivalence between a pattern structure and document structure. We show a base set of functions in Table 2. For example, we may require an exact match for the syntactic relation on a syntax edge, while allowing any value for the modal relation on a modal edge because it merely specifies the epistemic stance of the claimant toward the inner claim, and does not determine the inherent claim structure.

4.7 Algorithm

We employ the NetworkX GraphMatcher[3] API over the document digraph and a pattern digraph in order to return the nodes in the document graph isomorphic to the nodes in the pattern graph. The matcher uses the VF2 (sub)graph isomorphism algorithm (Cordella et al., 2004). While subgraph isomorphism is an NP-complete problem, the time complexity of the VF2 algorithm is $\Theta(N^2)$ in the best case and $\Theta(N!N)$ in the worst case, and maintains a $\Theta(N)$ space complexity. Given that N is the union of all tokens in a given document with abstract modal dependency nodes (which never exceed the number of tokens in sufficiently complex sentences), neither time nor space complexity poses an obstacle to the algorithm's practical application in prototyping this approach.

4.8 Relaxed Patterns with On-Match Filtering for Generalized Structures

Some claim structures may be accounted for with less deterministic pattern definitions. A salient instance is found in sentence (1) above and many other annotated examples in our analysis: multiple events are assigned the same modal pattern, and are grammatically subordinated to the clausal complement of the same SIP. However, each event trigger has a different location at a potentially different depth in the SIP subtree.

We generalize the 1-hop relation in the *ccomp* pattern graph into a k-hop relation in the *relaxed_ccomp* pattern, shown in Figure 4, by defining two distinct nodes, α and β, to match the immediate (1-hop) clausal complement of the SIP and any other (k-hop) event token node subordinated to it, respectively. The definition imposes no syntactic requirement on β (only that it be the token of a modal event node). To ensure that the match for β is actually subordinate to the SIP's clausal complement, we implement an on-match filter[4] that for each returned isomorphism checks $is\_ancestor(\alpha, \beta)$ and discards matches where β falls outside the subtree governed by α.

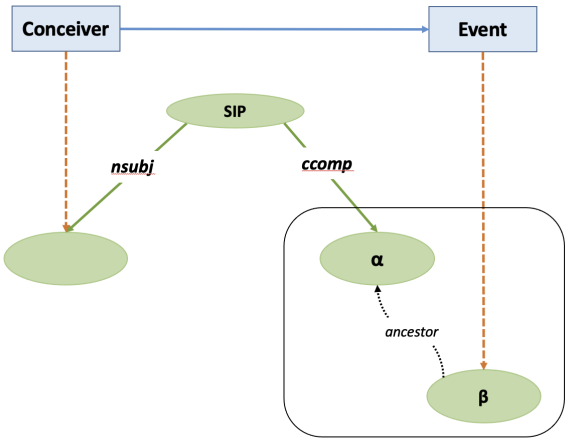

Figure 4: *relaxed_ccomp* pattern graph that requires ancestor-checking filter

On-match filtering allows us to generalize claim patterns insofar as we can exploit the topology of the graph to prune away uncompliant extractions.

5 Building a Claim Bank

We apply our claim extraction algorithm to the crowdsourced English modal dependency dataset (Yao et al., 2021), over which we additionally ran the default Stanza dependency parser[5] (Qi et al., 2020) (English *ewt* model) to have both the MDP and DP information available for every document.

As a minimal qualitative assessment for the prototyped approach, we build a Claim Bank by running the subgraph pattern matcher with the *ccomp* and *according_to* claim patterns over the train, dev and test portions of the crowdsourced dataset with 289, 32 and 32 parsed documents, respectively. We show in Table 3 raw MDP conceiver-event edge counts to illustrate the large cardinality of the claim

[3] https://networkx.org/documentation/stable/reference/algorithms/isomorphism.vf2.html#graph-matcher

[4] "on-match" terminology borrowed from spaCy's rule-based matching API (https://spacy.io/usage/rule-based-matching)

[5] Version 1.2

node type match	$\phi(n_1) = \phi(n_2)$
node match on feature function γ	$node\_type\_match(n_1, n_2) \wedge$ $(\gamma(n_1) \wedge \neg\gamma(n_2)) \vee (\neg\gamma(n_1) \wedge \gamma(n_2)) \vee (\gamma(n_1) = \gamma(n_2))$
edge type match	$\psi(e_1) = \psi(e_2)$
edge match on feature function δ	$edge\_type\_match(e_1, e_2) \wedge$ $(\delta(e_1) \wedge \neg\delta(e_2)) \vee (\neg\delta(e_1) \wedge \delta(e_2)) \vee (\delta(e_1) = \delta(e_2))$

Table 2: Customizable Boolean logic for determining equivalence of nodes and edges during subgraph matching of claim pattern graphs to document graph. γ and δ stand for generic node and edge feature functions, respectively

	train	**dev**	**test**
#C-E total	14460	1732	1641
#C-E, C linked to token(s)	8001	1022	1044
#C-E cross-sentence[6]	1736	240	163
#ccomp	558	76	86
#ccomp$_{1\text{-hop}}$	544	71	85
*#*ccomp$_{>1\text{-hop}}$*	1887	233	242
#ccomp$_{>1\text{-hop}}$	926	111	130
#according_to	61	5	6

Table 3: Conceiver-event (C-E) edges and claim extraction counts, grouped by pattern

pool provided by MDP that gets pared down by constraining MDP with DPs. We categorize the *ccomp* family of extractions in terms of how far away the event token is from the clausal complement of the source introducing predicate (*ccomp*$_{\geq 1\text{-hop}}$ correspond to the *relaxed_ccomp* pattern, and an asterisk indicates the number of extractions before applying the on-match *is_ancestor* filter).

We confirm that there is no intersection between the node isomorphism matches returned by each of the claim patterns of interest. We randomly sampled 10 examples from the train set for the extracted *according_to* claims and the *ccomp*-family claims. For all 20 examples, the extractions match our intuition of a claim as an assertion ("inner claim") evidentially related to an opinion-holding conceiver. Sentences (5) and (6) are examples of *according_to* and *ccomp* extractions, respectively, from our random samples:

(5) As of Wednesday, the state **had** more than 16,460 known cases and 539 known deaths, according to **the department**.

(6) The DPA is being **used** to obtain about 60,000 test kits, **Gaynor** told CNN's New Day.

A high-quality Claim Bank, containing overt claimants linkable to real-world entities, facilitates an exploration of claims by individuals of interest, and provides an avenue for sifting through conflicting perspectives on events.

6 Subgraph Matching as Inference, Subgraph Mining as Learning

Our approach outlines an entirely algorithmic inference procedure to extract knowledge elements (KEs), and may therefore be reminiscent of symbolic AI. We have so far discussed how to extract claims with predefined human-curated patterns, but curating such patterns manually is as unreliable and time-consuming as feature engineering, and we want to automatize extracting such patterns from parsed, annotated corpora. We propose subgraph mining as a general-purpose methodology for "learning" patterns, as capable of extracting schematically-defined KEs as the parses are expressive of them. The generality and interpretability of the method we formulate below make it as an appealing alternative to task-specific and at times brittle neural extractors, or to powerful but even less interpretable prompt-based approaches.

An annotated corpus of claims such as LDC2021E11 or NEWSCLAIMS consists of KEs containing the token indices a of the claimer and b of the inner claim, and labels for the claim topic and claimer stance. We can therefore view the claims corpus as $\mathcal{C} = \{c_1, ..., c_n\}$ where each claim c_i has the structure[7]:

$$c_i = \left\{ \begin{array}{l} \textsc{Text} = \text{sentence or document} \\ \textsc{Spans} = \left\{ \begin{array}{l} \{a_1, ..., a_k\} \\ \{b_1, ..., b_l\} \end{array} \right\} \\ \textsc{Labels} = \left\{ \begin{array}{l} \textsc{claim topic} \\ \textsc{claimer stance} \end{array} \right\} \end{array} \right\}$$

Instead of learning span extractors and classifiers from token-level "surface" annotations, we first

[7]This structure can be generalized to various KEs such as entity-relations, event-relations and event-argument frames

apply sentence- and document-level parsers and compose the resulting parses into a digraph as outlined in §4.4. [8]

We hypothesize that in the graph composed of available parse types $\mathcal{P}$, features predictive of the KEs of interest will be contained in the graph neighborhoods of the annotated tokens. We thus define a neighborhood function that returns the k-hop portion of the digraph $\mathcal{D}$ surrounding given tokens t, where directionality d can be specified to only outgoing ($\uparrow$) or only incoming ($\downarrow$) edges from the token node(s), or both ($\updownarrow$). Subgraphs created via those neighborhoods are then passed into a subgraph mining procedure (§6.1) for subgraph-matching-based inference. The learning paradigm is summarized in Algorithm 1:

Algorithm 1 Pattern discovery algorithm

Input: $\mathcal{C}, \mathcal{P}, k \geq 1, d \in \{\uparrow, \downarrow, \updownarrow\}$)
Output: Claim patterns for subgraph matching

$\mathcal{G} \leftarrow \emptyset$
for c_i in $\mathcal{C}$ **do**
  $\text{parses}_i \leftarrow \bigcup_{\text{Parser} \in \mathcal{P}} \text{Parser}(\textsc{Text}(c_i))$
  $\mathcal{D}_i \leftarrow \text{CreateDigraph}(\text{parses}_i)$
  $t \leftarrow \bigcup \textsc{Spans}(c_i)$
  $g \leftarrow \text{neighborhood}(t, \mathcal{D}_i, k, d)$
  add g to $\mathcal{G}$
end for
patterns $\leftarrow \text{Mining}(\mathcal{G})$
return patterns

6.1 SPMiner

SPMiner[9] (Ying et al., 2020b) is the first and only neural approach we are aware of to extract frequent subgraphs from a collection of graphs. SPMiner uses a GNN encoder to embed graphs in an order embedding space, and is trained to enforce a partial ordering such that subgraphs reside to the lower-left of their super-graphs in this space. This neural matching subroutine is used in a search (greedy search, beam search or Monte Carlo tree search; we only explored the first of these) that identifies frequent motifs of size k by iteratively expanding nodes and edges of candidate motifs (starting with seed nodes at random), and selecting those that retain the most points to their top right in the embedding space (i.e. the motifs with highest frequency).

A challenge of applying SPMiner to our NLP domain is that it was developed with molecular graphs in mind that have more variable connectivity patterns than parses, while allowing for at most one label per node or edge (unlike the token nodes in our setting). To mitigate this mismatch, we expand out the graphs so that each attribute is encoded by a synthetic node-edge connection, cf. Figures 5 and 6. The transformation is invertible, such that the expanded graph can be unambiguously collapsed into the original. Given the unreliability of SPMiner's default GNN encoder for our graphs, in the search procedure we swap out the neural (batched) subgraph isomorphism with the much slower VF2 algorithm to ensure that the random walk's results are not confounded by the encoder's performance. This substantially slows down our exploration and is a point for future work (cf. §6.3). Finally, instead of arbitrary seed nodes, we force SPMiner to start the growth with the token nodes for the claimer and inner claim, so as to force the resulting motifs to contain the full claim information.

6.2 Mining for Claims

In another proof-of-concept evaluation, we create a silver corpus of high-precision within-sentence claims from LDC corpus LDC2021E11 (10 documents with 1219 sentence total) that we parsed for dependency syntax and MDP and annotated with the human-curated "seed" patterns discussed in §4. We then use SPMiner to mine for syntax-only patterns from the silver claims. We obtain 100 patterns, which we then apply over the same corpus, examining the quality of the predicted claims. Many of the mined patterns are very simple (high-recall, low-precision), resulting in numerous spurious matches[10]. However, after filtering out any patterns that found over 1000 matches, we are left with syntactic claim patterns that are interpretable and that found reasonable claims not identified by our MDP-constrained seedling patterns. Figure 7 visualizes a non-trivial mined pattern that recovers the overall *ccomp* structure (with some admittedly superfluous edges), Sentence (7) is a claim from

[8] Our implementation supports composing sentence-level syntactic dependency parses (DP) and abstract meaning representation (AMR) (Banarescu et al., 2013), and document-level modal and temporal dependency parses (MDP and TDP) (Zhang and Xue, 2018; Zhang, 2020), into a composite digraph that is "held together" at the token nodes, since every parse type includes them (AMR nodes can be aligned to tokens). We can use any relevant subset of {DP, AMR, MDP, TDP} as our input to the pattern mining procedure.

[9] http://snap.stanford.edu/frequent-subgraph-mining

[10] 449536 total matches, 53630 unique matches

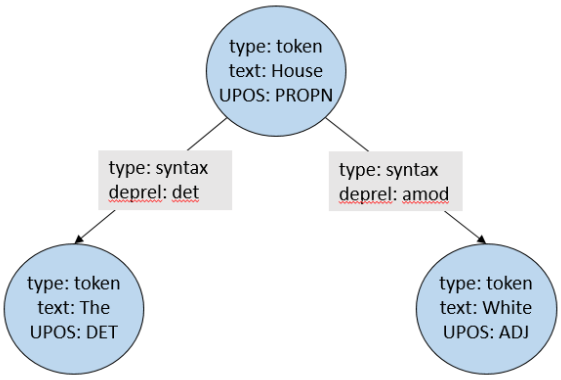

Figure 5: Unexpanded dependency syntax graph for "The White House"

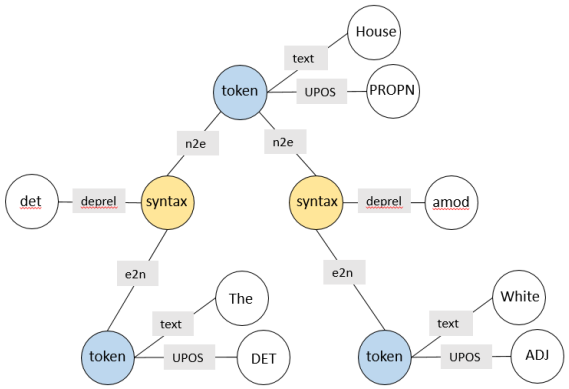

Figure 6: Expansion of dependency syntax graph for "The White House" into graph with a single attribute per each node and edge. *n2e* stands for *node-to-edge* and *e2n* for *edge-to-node*

the silver corpus and Sentence (8) is a novel claim extracted by the mined pattern:

(7) [Tim Trevan, a biological safety expert based in Maryland $_{\text{Claimer}}$], said [most countries had largely abandoned $_{\text{Inner Claim}}$] their bioweapons research after years of work proved fruitless.

(8) [Richard Ebright, a professor of chemical biology at Rutgers University $_{\text{Claimer}}$], said earlier this year in an interview with The Washington Post: ["Based on the virus genome and properties, there is no indication whatsoever that it was an engineered virus." $_{\text{Inner Claim}}$]

Mining the silver corpus after parsing it for AMR yielded 21 AMR-only claim patterns, and after parsing it for both AMR and dependency syntax yielded 29 composite DP-AMR claim patterns.

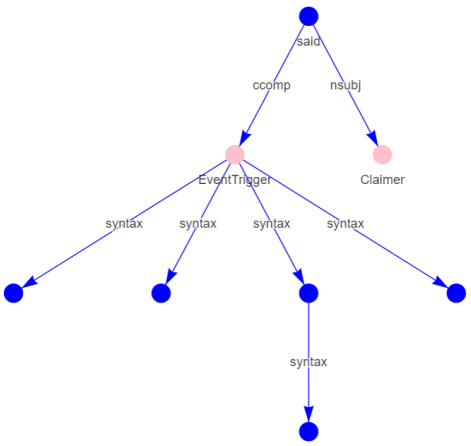

Figure 7: Over 10 documents, this dependency syntax pattern detected 46 claims, 27 of which were not captured by our original DP+MDP patterns

6.3 Challenges and Future Work

The subgraphs returned by SPMiner do not come equipped with node- or edge-match functions as defined in §4.6. This works out in the case of the expanded single-attribute-per-node/edge graph input to SPMiner, as we can simply require that the functions match on all attributes and trust that the mining algorithm will exclude irrelevant/non-predictive attributes from its frequent motifs. We leaving mining for frequent attributes jointly with frequent motifs to future work.

We have yet to explore training NLP-specific GNN encoders such as discussed in §2.2 for accurate neural subgraph isomorphism to speed up the mining procedure and allow for a thorough hyperparameter search that is not prohibitively slow.

Finally, we would like to explore this approach at different linguistic levels, including discourse-level argumentation structures such as that of Stab and Gurevych (2017) or Rhetorical Structure Theory (RST) (Mann and Thompson, 1987).

7 Conclusion

We demonstrate the viability of a simple paradigm for extracting and learning KE structures from a variety of parses. We outline avenues to make this approach more efficient and robust, and surmise that as linguistic representations and parsers continue to improve in scope and in accuracy, the NLP community will benefit from interpretable graph-based techniques over them.

8 Acknowledgements

This research was developed with funding from the Air Force Research Lab (AFRL) and Defense Advanced Research Projects Agency (DARPA), via Contract No.: FA8750-18-C-0001. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. This report has been Approved for Public Release, Distribution Unlimited.

References

Ammar Arbaaeen and Asadullah Shah. 2020. Natural language processing based question answering techniques: A survey. In *2020 IEEE 7th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pages 1–8. IEEE.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. *arXiv preprint arXiv:1704.04675*.

Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 355–362.

Luigi P Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. 2004. A (sub) graph isomorphism algorithm for matching large graphs. *IEEE transactions on pattern analysis and machine intelligence*, 26(10):1367–1372.

Michael Glass and Alfio Gliozzo. 2018. A dataset for web-scale knowledge base population. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pages 256–271. Springer.

Eduard Hovy. 2020. Active interpretation of disparate alternatives (aida). *Defense Advanced Res. Projects Agency*, 2020.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. Hover: A dataset for many-hop fact extraction and claim verification. *arXiv preprint arXiv:2011.03088*.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. Wice: Real-world entailment for claims in wikipedia. *arXiv preprint arXiv:2303.01432*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

William C Mann and Sandra A Thompson. 1987. *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*.

Jordan K. Matelsky, Elizabeth P. Reilly, Erik C. Johnson, Jennifer Stiso, Danielle S. Bassett, Brock A. Wester, and William Gray-Roncal. 2021. DotMotif: an open-source tool for connectome subgraph isomorphism search and graph queries. *Scientific Reports*, 11(1).

Thien Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

B Onyshkevych. 2017. Active interpretation of disparate alternatives.

Kota Prudhvi, A Bharath Chowdary, P Subba Rami Reddy, and P Lakshmi Prasanna. 2020. Text summarization using natural language processing. In *Intelligent System Design: Proceedings of Intelligent System Design: INDIA 2019*, pages 535–547. Springer.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Revanth Gangi Reddy, Sai Chetan Chinthakindi, Zhenhailong Wang, Yi Fung, Kathryn Conger, Ahmed Elsayed, Martha Palmer, Preslav Nakov, Eduard Hovy, Kevin Small, et al. 2022. Newsclaims: A new benchmark for claim detection from news with attribute knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6002–6018.

Benjamin Rozonoyer. 2021. *Graph Convolutional Encoders for Syntax-aware AMR Parsing*. Ph.D. thesis, Brandeis University.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Meagan Vigus, Jens EL Van Gysel, and William Croft. 2019. A dependency structure annotation for modality. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 182–198.

Jiarui Yao, Haoling Qiu, Jin Zhao, Bonan Min, and Nianwen Xue. 2021. Factuality assessment as modal dependency parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1540–1550.

Rex Ying, Zhaoyu Lou, Jiaxuan You, Chengtao Wen, Arquimedes Canedo, and Jure Leskovec. 2020a. Neural subgraph matching. *ArXiv*, abs/2007.03092.

Rex Ying, A Wang, Jiaxuan You, and Jure Leskovec. 2020b. Frequent subgraph mining by walking in order embedding space. In *Proc. Int. Conf. Mach. Learn. Workshops*.

Yi Zhang, Zachary Ives, and Dan Roth. 2020. "who said it, and why?" provenance for natural language claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4416–4426.

Yuchen Zhang. 2020. *Temporal Dependency Structure Modeling*. Ph.D. thesis, Brandeis University.

Yuchen Zhang and Nianwen Xue. 2018. Neural ranking models for temporal dependency structure parsing. *arXiv preprint arXiv:1809.00370*.

A Subgraph Isomorphism vs Subgraph Monomorphism

We conducted our subgraph matching experiments with the NetworkX subgraph isomorphism matcher, unaware of the subtle difference between subgraph isomorphism and subgraph monomorphism. Our implementation of the *relaxed_ccomp* pattern (cf. §4.8), where we deliberately did not define any edge between α and β to generalize their distance from each other, kept failing to match claims in which α was the parent of β (i.e. exactly 1-hop above it). This led us to implement the pattern $ccomp_{1\text{-}hop}$ in addition to $ccomp_{>1\text{-}hop}$ to get full coverage; the latter pattern having no edge between α and β (as intended), while the former containing the edge corresponding to the 1-hop distance between those two nodes.

We refer the reader to https://networkx.org/documentation/stable/reference/algorithms/isomorphism.vf2.html#subgraph-isomorphism, from which we cite, for a mathematical definition of subgraph isomorphism and monomorphism: "*to say that G1 and G2 are graph-subgraph isomorphic is to say that a subgraph of G1 is isomorphic to G2*".

The key point to note is that in the NetworkX VF2-based subgraph isomorphism, "subgraph" always refers to *node-induced subgraph*:

- If $G' = (N', E')$ is a node-induced subgraph, then:
  - N' is a subset of N, E' is **the subset of edges** in E relating nodes in N'
- If $G' = (N', E')$ is a monomorphism, then:
  - N' is a subset of N, E' is **a subset of the set of edges** in E relating nodes in N'

The node-induced subgraph requirement of isomorphism necessitates the *complete* subset of edges connecting nodes in N'. This explains the failure of the $relaxed\_ccomp$ pattern to match the case where α is a parent of β, since the pattern subgraph does not contain the edge between α and β. By contrast, monomorphism *does* yield a match in this case, as it requires the pattern to define merely *a subset* of the set of edges connecting nodes in N': "*if G' is a node-induced subgraph of G, then it is always a subgraph monomorphism of G, but the opposite is not always true, as a monomorphism can have fewer edges.*"

B SPMiner Hyperparameters

We used the following hyperparameters for SPMiner:

Parameter	Value
node_anchored	true
n_neighborhoods	3000
n_trials	100
min_pattern_size	10
max_pattern_size	50
min_neighborhood_size	10
max_neighborhood_size	60
search_strategy	greedy

Table 4: SPMiner hyperparameters

https://github.com/snap-stanford/neural-subgraph-learning-GNN/blob/master/subgraph_mining/config.py

C Visualizations

Figure 8: Syntactic dependency parse for "*The World Health Organization also warned Monday that the virus had not suddenly become less lethal.*"

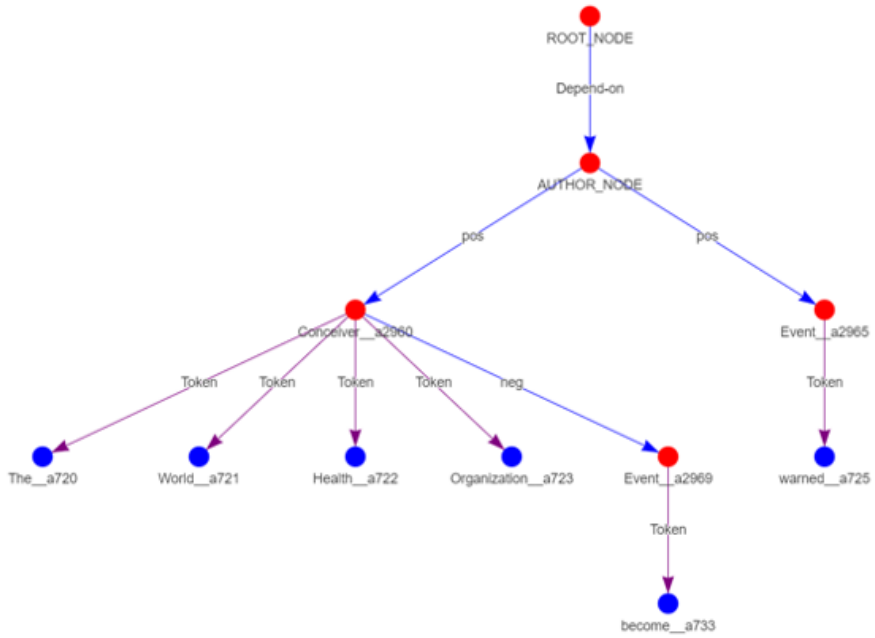

Figure 9: Modal dependency parse for "*The World Health Organization also warned Monday that the virus had not suddenly become less lethal.*"

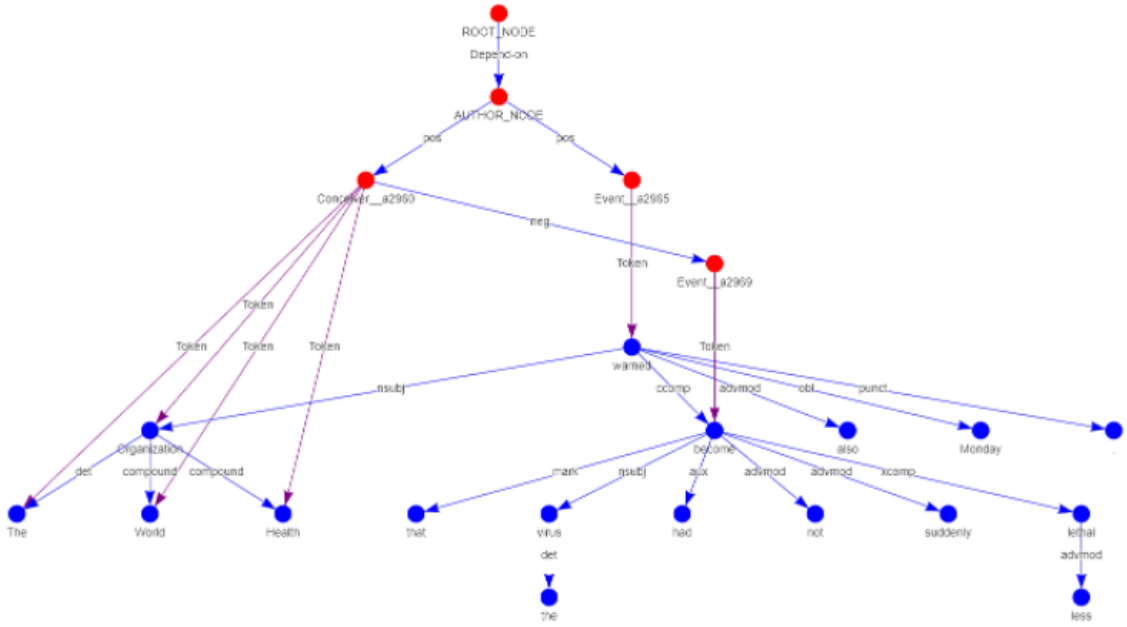

Figure 10: Composed modal and syntactic parse for "*The World Health Organization also warned Monday that the virus had not suddenly become less lethal.*"

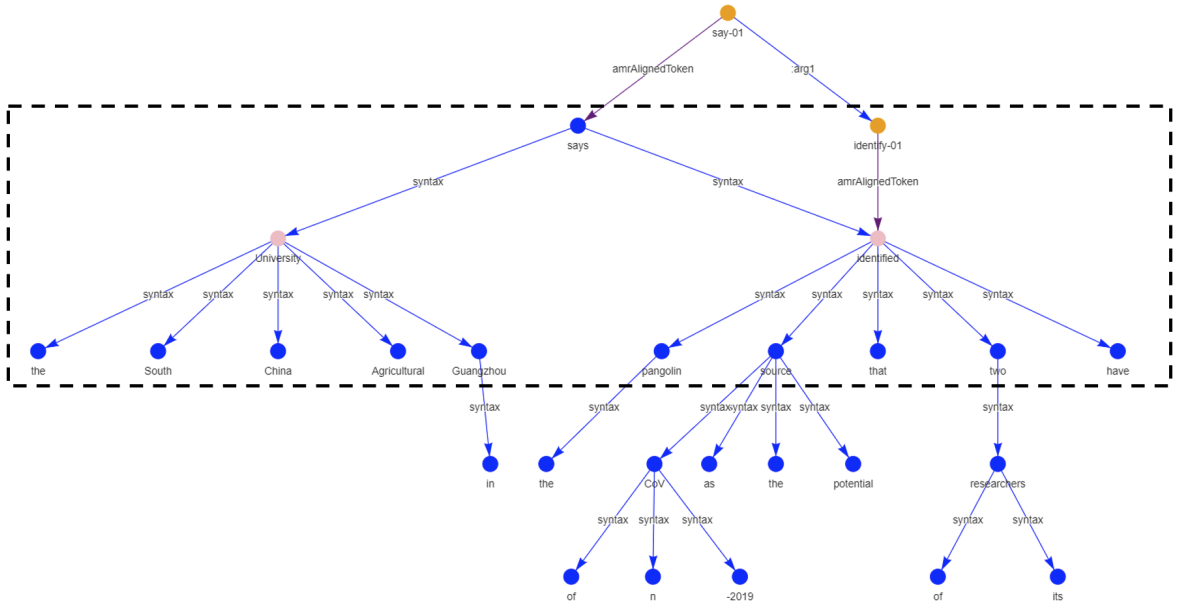

Figure 11: Composed syntactic and AMR parse for "*The Guangzhou South China Agricultural University says that two of its researchers have identified the pangolin as the potential source of COVID-19.*", with 1-hop neighborhood around claimer and inner claim head tokens. Note `amrAlignedToken` edges that connect an AMR node to the token it has been aligned to

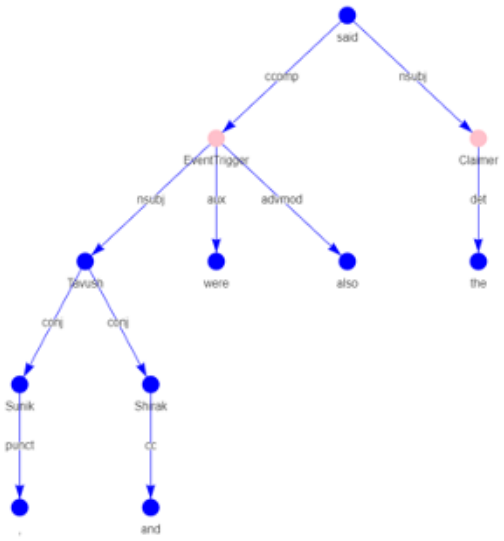

Figure 12: Example local neighborhood graph for dependency syntax parse of a claim

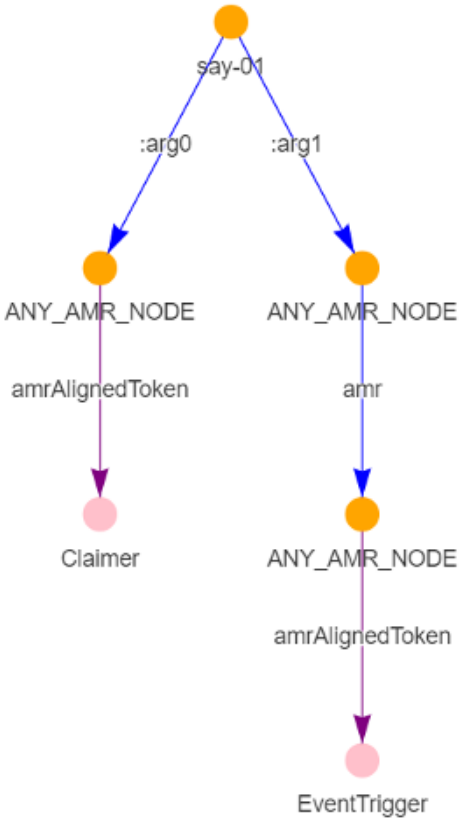

Figure 13: AMR pattern example, analogous to dependency syntax example in Figure 7