Benefit Graph Extraction from Healthcare Policies

Vanessa Lopez[1(✉)], Valentina Rho[1], Theodora S. Brisimi[1], Fabrizio Cucci[1], Morten Kristiansen[2], John Segrave-Daly[2], Jillian Scalvini[3], John Davis[3], and Grace Ferguson[3]

[1] IBM Research Dublin, Dublin, Ireland
vanlopez@ie.ibm.com
[2] IBM Watson Health, GHHS, Dublin, Ireland
[3] Payment Integrity (Truven Health Analytics), Ann Arbor, USA

Abstract. With healthcare fraud accounting for financial losses of billions of dollars each year in the United States, the task of investigating regulation adherence is key to reduce the impact of Fraud, Waste and Abuse (FWA) on the healthcare industry. Providers rendering services to patients typically submit claims to healthcare insurance agencies. Such claims must follow specific compliance criteria specified by state and federal policies. This paper presents an ontology-based system that aims to support the FWA claim investigation process by extracting graph-based actionable knowledge from policy text describing those compliance criteria. We discuss the process of creating a domain-specific ontology to model human experts' conceptualisations and to incorporate early-on the feedback of FWA investigators, who are the early adopters of our solution. We explore whether the ontology is expressive and flexible enough to model the diverse compliance processes and complex relationships defined in policy documents. The ontology is then used, in combination with natural language understanding and semantic techniques, to guide the extraction of a Knowledge Graph (KG) from policies. Our solution is validated in terms of correctness and completeness by comparing the extracted knowledge to a ground truth created by investigators. Lastly, we discuss further challenges our deployed semantic system needs to tackle in this novel scenario, with the prospect of supporting the investigation process.

1 Introduction and Business Scenario

The National Health Care Anti-Fraud Association estimates that the financial losses due to health care fraud in the US are in the tens of billions of dollars each year [1]. According to Truven Health research, approximately $125 to $175 billion is wasted each year on healthcare fraud and abuse [2]. The Health Care Fraud and Abuse Control

V. Lopez, V. Rho, T. S. Brisimi and F. Cucci—Equal research contribution. We would like to **acknowledge** Conor Cullen, Carlos Alzate, Spyros Kotoulas, Martin Stephenson, Pierpaolo Tommasi, Marco Sbodio, Denisa Moga and our OM: Tim Cooper, Mark Gillespie and Mark Goodhart for their support and insights.

C. Ghidini et al. (Eds.): ISWC 2019, LNCS 11779, pp. 471–489, 2019.
https://doi.org/10.1007/978-3-030-30796-7_29

Program (HCFAC), established under the Health Insurance Portability and Accountability Act (HIPAA), directs federal and state agencies to audit healthcare expenditure with the objective of improving the quality of care and recover tax payer dollars.

Medicare and Medicaid have been designated as high-risk federal programs [3], because of their size, complexity and susceptibility to improper payments. The *Program Integrity* investigation units established under the HCFAC aim to assert that the ***correct payment*** has been made for the ***correct member*** for the ***correct service*** to the ***correct provider***. Healthcare providers (hospitals, pharmacies, clinics etc.) submit claims to state and federally-administered health insurance agencies (such as Medicare or Medicaid) for services rendered to a patient. Policy guidelines set out which claims are permissible based on eligibility criteria for a particular service and generally accepted medical practices. Invalid claims are those that infringe policy criteria either intentionally (fraudulent) or unintentionally (providing services that are unnecessary, inefficient or inconsistent with accepted medical practices). FWA investigators need to prioritize investigations based on likelihood of recovery (dollars) and maximum return on investigation resources. However, understanding policy, consisting of hundreds of text pages describing *compliance criteria* that investigators have to review and refer to in an investigation for further recovery actions, is a manual and labor-intensive task. Investigation does not guarantee recovery, since the policy may turn out to be too vague to be enforced, or the recoverable amounts too low to warrant action – any of which take scarce investigation resources away from other recovery opportunities. Comprehensively understanding policy is a key step to ensuring recovery of inappropriately paid claims.

We present a semantic solution that extracts compliance knowledge from healthcare policy documents. This knowledge can facilitate FWA investigations in several ways - for example, helping in the development of claims-inspection algorithms. Semantics play a key role in extracting machine-readable knowledge about Benefit Rules from the human-oriented policy documents. ***Benefit Rules (BRs)*** describe structured compliance criteria, such as: eligible service providers (e.g. role: physician, nurse); eligible places of service (e.g., home, hospital); maximum billable units of service or equipment per-patient in a given period; services that should not be billed together for a patient on a single date, services (in)appropriate for a patient's age or gender, etc. An experienced team of ***FWA Investigator*** consultants, working with the state and federal government to help them meet their obligations under the HCFAC and to shape policy, acted as early adopters of our solution, providing robust evaluation feedback and ground truth data along the way.

Our **contributions** in this paper are twofold. First, we describe the lessons learned and the best practices adopted while working with investigators throughout the entire lifecycle: validating the value proposition; modelling a domain ontology with the purpose of supporting claims investigations; and capturing experts' feedback to build a **Ground Truth** (GT) on BRs, i.e., knowledge that an investigator would learn from policy text, enabling us to provide performance metrics that validate the accuracy and completeness of our solution's automatically extracted knowledge. Secondly, we describe the research challenges, design choices and the approach to build a semantic-based system that applies natural language understanding techniques to policy text to transform it into relevant, semantic, graph-based BRs, guided by the ontology.

The rest of the paper is organised as follows. Section 2 presents related work. Section 3 discusses the advantages as well as the challenges of using semantic technologies in our solution, while Sect. 4 gives an overview on the technical implementation. Section 5 presents a validation of the system *in-use* with policies from two different domains. We conclude with discussing and summarizing the ongoing work in Sect. 5.

2 Background and Related Work

2.1 Medical Claims Audit

The role of analytics to identify FWA in healthcare insurance claims has been explored in [4] through different analytical approaches on top of claims data (e.g., text mining, social network analysis, time series analysis).

Supervised and unsupervised data mining approaches to support fraud detection on claims data are presented in the surveys [5, 6]. For example, they are applied to detect anomalies in the utilisation of certain procedure codes, or to create a risk profile about providers to report to third-party payers (such as health insurance organizations like Medicaid/Medicare).

A significant differentiator of our approach with respect to claim-based state of the art analytical approaches is that this is the first system, that we are aware of, that aims to interpret unstructured policy with the purpose of supporting policy investigators' work. Investigator time is precious, and policy is vast, hard to understand and hard to relate to claims. Our goal is to extract BRs that can facilitate policy comprehension to support investigators on the analysis of potentially-inappropriate payments.

2.2 Knowledge Base Population

Ontology guided Information Extraction (IE) [7] and Knowledge Base Population (KBP) from text, has been addressed by both the computational linguistics and semantic web communities for several years (for a survey see [8]). For instance, the Text Analysis Conference (TAC) has a Cold Start KB evaluation track to build a KB from scratch, using just a predefined schema and a corpus of text [9]. Effective systems in these competitions combine many approaches such as rule-based relation extraction and distantly supervised linear and neural network extractors. Domain-independent relation extraction has been studied by a wide range of approaches, however relation extraction and KBP from text often requires building IE analytics to discover facts about entities in text for the domain, as generic models rarely work well on the customer specific data.

Statistical supervised IE approaches, based on term frequency and co-occurrence of specific terms, require substantial effort from domain experts to manually label each mention of an entity or relation on hundreds of documents. Background knowledge can alleviate the need for human supervision for domain adaptation. A *knowledge and linguistic-based* approach is presented in [10] to extract first, medical entities from sentences to determine their categories, and second, semantic relations between a pair

of entities by using lexical patterns built semi-automatically using a corpus (PubMed) and six relations types from UMLS. *Distance supervision* approaches do not require manual data labeling, instead training data is provided in the form of entity pairs belonging to a specific relation [11]. For example, [12] exploits a partially populated KB and a corpus of text to train a set of deep learning classifier to find paraphrases, i.e. different expressions with similar meaning in text, and to augment and extend a partially populated KG. With the exception of [12], most state of the art approaches are only able to recognize explicit pairwise relations within the same sentence [13]. [13] explores a cross- sentence neural architecture for *n-ary* relation extraction, by building paths connecting two identified arguments through related entities in a biomedical domain.

An *ontology guided IE* approach is presented in [14], based on the linguistic platform GATE entities are annotated in documents (e.g., to capture facts about a company) and mapped into ontology concepts, and then documents that refer to the same entity (e.g., company) are cluster together using on a cosine similarity vector representation. PIKES is a frame-based framework to extract instances and relationships in text [15], each frame is a reified object connecting instances through properties describing their semantic role based on the FrameBase ontology. Semantics are often applied in the healthcare domain to integrate data from heterogeneous sources, model diverse business process, and to build declarative rules to capture measures on the quality of care expressing complex relationships [16].

We believe that recognizing the many explicit and implicit *N-ary* relationships needed to extract multiple BRs from a policy document requires substantial domain background knowledge and the ability to perform inference. In this paper we describe a first implementation that combines NLP, knowledge representation and ontology-guided reasoning, to automatically capture the complex BR relationships into a KG. Labelled data is required for evaluation only.

3 Semantics in Practice

3.1 Advantages and Challenges of an Ontology-Based Solution

Ontologies serve as explicit, conceptual knowledge models to share a common understanding of the information in a domain and make that knowledge available to information systems [17]. This knowledge includes machine-interpretable definitions of concepts and relations in the domain, makes domain assumptions explicit and separates the domain knowledge from the operational knowledge. In our scenario, the role of the ontology is central to guide the IE process and for visually representing auto-extracted knowledge to investigators for curation.

In the following, we describe the **advantages** to using a domain ontology as a foundation for our solution.

- **Interoperability:** the ontology is the only domain-dependent resource. It contains the schema of the relations and entities to be extracted from policy text, and the labels (surface forms) needed to match ontological resources to entities and relations in the policy. It also acts as central hub to link to other relevant domain sources. In

particular: (1) medical codes used during the billing process to describe clinical procedures, such as the American Medical Association's CPT (Current Procedural Terminology) code set; (2) body parts, e.g. in the dental domain a procedure may be only applicable on a subset of teeth; (3) healthcare programs available for a specific State; and (4) eligible places of service. The ontology also links to medical taxonomies such as UMLS to define diagnoses or treatments, that are typically referred to by patient's *medical history* or *high-risk status*.

- **Flexible and incremental model:** it is not feasible to define a complete domain model a priori. We started by identifying high-value BR types with our investigators (for example, a Service Limitation subtype is shown in Table 1). We then extended the ontology incrementally to cover more BR subtypes, as well as new policy domains. As the coverage of the ontology increased, the extractor's ability to capture more-relevant knowledge and to infer missing relations also increased. Using semantics there is no need to impose a fixed BR template. Extractors can automatically instantiate a BR in the KG using any combination of criteria known in the ontology (property-values) as long as they are semantically consistent.

Table 1. Table with three BR examples based on a *Service Limitation* template, which describes unit or dollar limits for services for a single beneficiary on a date of service.

Policy text	Template	Ground truth (BRs)
Dental prophylaxis (i.e., teeth cleanings) is recommended every 6 months, and may be reimbursed twice per 12 months per member	Members who [*qualifying criteria*] can receive up to [*max units/monetary amount*] of [*list of applicable services*] per [*body part*] in [*applicable time period*] requires [*requirements*] unless [*list of non-applicable services*] and [*exclusions*]	[*qualifying criteria*] - all-members [*max units*] - 2 [*applicable services*] - prophylaxis [*time period*] - 12 month
Members determined to be at high risk for periodontal disease or high risk for caries (decay) is eligible for additional services. These services include [..] up to four (4) prophylaxis procedures		[*applicable services*] - prophylaxis [*member - high risk of*] - umls-caries, umls-periodontal-disease [*max units*] - 4
Fluoride rinse is not an acceptable treatment for any child member and will not be reimbursed		[*qualifying criteria - min age*] - 0 [*qualifying criteria - max age*] - 21 [*non-applicable services*] - fluoride-rinse

- **Semantically sound:** specific domain constraints can be defined in the ontology to ensure that consistent and meaningful BRs are extracted from a portion of text when consolidating multiple BRs and identifying information conflicts. For example, the

content of a BR can span across different sentences in the same paragraph and/or other connected portions of the policy, e.g. section headings. In this scenario, the constraints in the ontology help in understanding when a BR can be enriched with contextual information or merged with another BR extracted from a connected sentence (see Sect. 4.2).

An ontological model can faithfully represent an expert Investigator's conceptualization and be sufficiently flexible to capture diverse compliance processes in knowledge graphs. However, a significant challenge is that **the knowledge graphs cannot be understood or curated by an Investigator**. Two important goals in our scenario were to enable investigators to curate the extracted knowledge and to create GT through an approachable user interface (UI). To achieve this goal, graph BRs are transformed in a user-friendly *flat*-representation (see Fig. 3), hiding the complexity of the underlying graph ontology (see Sect. 4.1). To help users understand extracted conditions, descriptive labels were added to the ontology for each field (i.e. condition) to be displayed. Users can curate the user-friendly representation of a BR by modifying, deleting or adding new fields and values.

Keeping track of the provenance of each BR is also a key requirement, not only to link the BRs in the KG to the original text in the policy, but also to reason about the origin of the information, e.g. which extractor extracted the BR, and to keep track of ontology updates. Ontology maintenance to reflect updates in policies and generalizability of the ontology is a challenge, e.g., context dependent default values, like a "fiscal year" may have different start and end date based on the state the policy applies to.

3.2 Ontology Definition and Ground Truth Collection with Investigators

Investigators expertise is crucial to understand the business area, to validate whether technical representations of BRs reflect the original policy accurately, and to assess the generalisability of this approach (schema) across policies from different geographies.

When processing unstructured data (text), the same information can often be represented and interpreted in many different ways. To collect a formal, abstract representation of domain knowledge from the Investigators, we adopted the following strategy:

- Investigators highlighted sentences containing BRs in the policy text and associated them with *BR templates.* The templates are abstract definitions of common BR patterns, expressed as a set of entities and relations. The templates are intended to be *transferable* - i.e. to generalize well to other policy areas and geographies.
- Guided by these templates, we (manually) created a first-draft of a formal ontological model (classes, relations and some instances) and began an iterative process of modeling, reviewing and incrementally improving the ontology with the Investigators. Investigators also helped identify other domain-related data sources, e.g. procedures codes, healthcare programs, body parts, places of service, etc.
- Using these templates, the investigators created a set of ground truth BRs from policies – a standard against which extractor output quality could be validated.

Table 1 showcases an example of a Service Limitation BR template and GT from policy text. These templates play the same role as *competency questions* (i.e., the set of questions that an ontology must be able to answer) typically used for ontology development, as they describe the ontology requirements to model different types of BRs.

4 Approach and Technical Components

We will introduce the components of the system, illustrated in Fig. 1 through the typical processing workflow. The first step when dealing with a new policy document, in PDF format, is to process it (1) together with its metadata in order to transform it in a machine-friendly tree structure in which the content is hierarchically organised, e.g. sections, paragraphs. After the document is ingested, the extraction service triggers the following steps: concepts annotation (2), based on the content of the BRs ontology and/or on external named entities annotators (NER) (Sect. 4.2); BRs extraction (3, 4), performed with different available extractors, currently WatsonX-based (3a) and SystemT-based (3b) (Sect. 4.2); across extractors consolidation and filtering (5) in order to merge the extracted rules and remove potential noise; BRs conversion (6) from the KG representation to a more user-friendly representation that can be easily displayed in the UI (8) to allow both the internal team and the investigators to inspect the extracted BRs, collect GT data and analyse the performance metrics computed on different extraction configurations (7). It's worth noticing that all the described components are domain independent, as they rely on the ontology to retrieve all the needed domain information. Existent KGs, e.g. some of the relevant types in the UMLS semantic network, are linked with the main ontology. External data in tabular form, e.g. relevant procedure codes, are normalised and lifted automatically (based on the creation of a file providing mappings between tabular columns and ontological entities) into a KG (0), following W3C recommendations [18], and linked to the ontology. All mentioned components are implemented as microservices and deployed on IBM Cloud. The ontology is currently loaded in-memory and accessed through the Jena Ontology APIs.

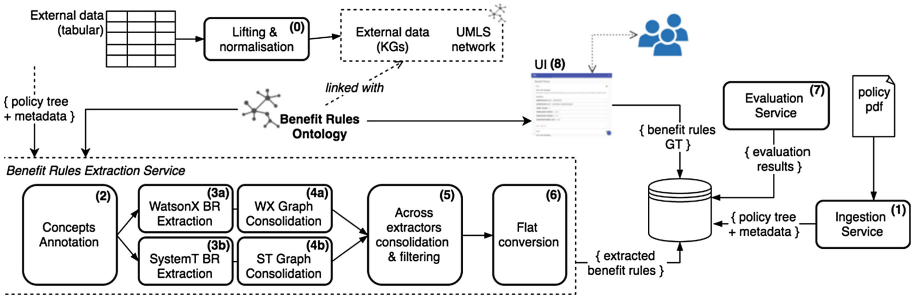

Fig. 1. Architecture of the system

4.1 Benefit Rules Ontology and Knowledge Graph Definition

The ontology has been iteratively and incrementally built pursuing two main goals: represent as correctly and unambiguously as possible the domain of interest, i.e. the BRs that are expressed in the policy text, and that is fit for purpose for the automatic IE, i.e. capturing the connections between the entities of interest and how they appear in text. The ontology was built based on the methodology described in [19] and the collaboration with a team of investigators. Based on the corresponding templates manually identified by the investigators (Sect. 3.2), we started by defining the class hierarchy, the object and data properties and semantic constraints that the ontology needs to cover. Different data sources, e.g. procedure codes, policy programs, places of service, have been identified and lifted in the ontology to populate the instances space.

Since the templates and the ontology are a commercially sensitive asset and they cannot be reasonably shared, we will focus the remaining of this section on a subset of it (see Fig. 2) that is meant to model the information encoded in the sentence *"Adult members may receive up to $1,000 in dental benefits per year (July1 through June 30)"*. The subset of the ontology used to model Service Limitations contains 21 object and 8 datatype properties of interest, 31 classes, 1034 individuals. The *Policy* class is the root node in the ontology. A *Policy* individual represents a document and may be associated with multiple *BenefitRule* individuals. A subtype of *BenefitRule* is created for each independent rule template we want to address, e.g. *BrServiceLimitation*. A subtype inherits all the properties of a generic BR class while at the same time allows us to capture the semantics particular to each. The principal BR properties modelled in Fig. 2 are: *service limitation,* that is meant to model a monetary or a unit limit range for a specific service under certain circumstances; *applicable services,* that model the services the BR applies to; *applicable time period*, represented in this case with *start* and *end date*, but that in other examples can be modelled as a frequency, e.g. "every 6 months"; *member eligibility*, to model all the eligibility criteria regarding the group of patients affected by the BR, in our example the only mentioned criteria was regarding the *age group* of the patient, but other criteria like the *enrolment* in a State plan, or the *history* of a particular disease, are covered as well. The ontology contains some pre-populated instances to model predefined nodes with default values, e.g. *adult* as an instance of *AgeRange* with predefined *min* and *max ages*.

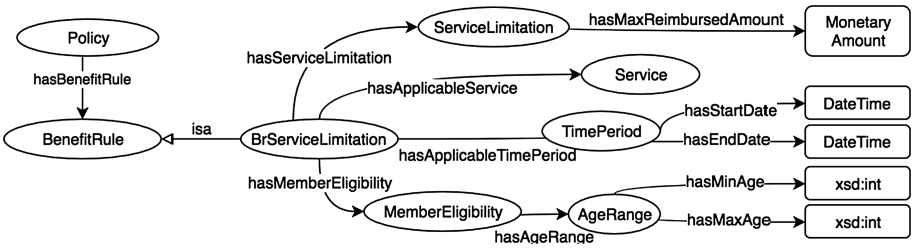

Fig. 2. Subset of the ontology to model the information in the policy text *"Adult members may receive up to $1,000 in dental benefits per year (July 1 through June 30)."*.

Different ontological constraints can be expressed on the ontology properties and will be used during the extraction process and enforced in the extracted KG, e.g. *owl:propertyDisjointWith*, *owl:minCardinality*, *owl:maxCardinality* or *owl:FunctionalProperty*. User-defined datatypes, associated to *entity types* extracted with the concept annotators ((2) in Fig. 1), are created to represent additional constraints that restrict the range of allowed datatype values to help in the disambiguation task, e.g., the string "$1,000" is annotated by the concept annotators as *MonetaryAmount* which is valid range for the datatype property *hasMaxReimbursedAmount*.

We divide classes and properties in the ontology in: *root,* e.g. *BenefitRule, hasMemberEligibilit; intermediate,* e.g. *TimePeriod, hasAgeRange;* or *leaf,* e.g. *Service, hasMinAge, hasService.* The user-friendly flat representation (Fig. 3) of a BR is created by taking all and only the leaf properties, also called *conditions*, in the BR with the corresponding range *values*. In order to be able to convert the KG into a flat representation without leading to ambiguities, the portion of the ontology that describes a BR type, i.e. the subgraph rooted in the *benefit rule type* class and generated by following the domain-property-range relations, must be in the shape of a tree; the proper ontological constraints are also added in the ontology to enforce the tree-shape of the KG as well. Given an instantiated BR, if the same *condition* has more than one different range value, these values are considered to be in disjunction with each other, e.g., if the BR mentions multiple *applicable services* we will consider the union of them. In contrast, two different conditions are considered to represent a conjunction, e.g. if both an *age* and a *history* of a particular disease are mentioned as eligibility criteria, the rule applies only to patients that are satisfying both criteria. A confidence score is assigned to each relevant KG statement. It is calculated based on the reliability of the applied extraction approach and the considered contextual information. Confidence information, as well as the approach used to extract each statement, are stored as reified statements.

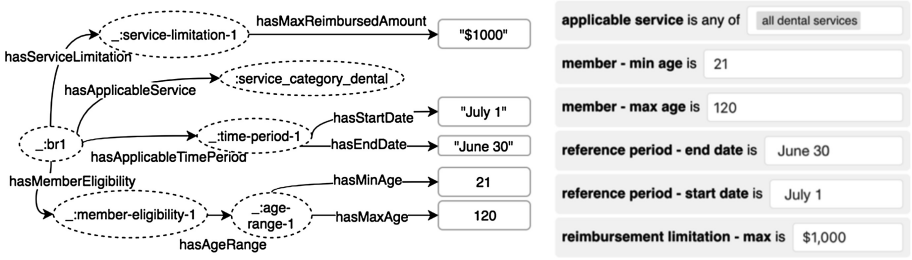

Fig. 3. KG of the BR (left) and the user-friendly flat representation (right) for sentence "Adult members may receive up to $1,000 in dental benefits per year (July 1 through June 30)."

4.2 Ontology-Based Information Extraction

Two extractors were implemented on top of two different NLP technologies WatsonX [20] and SystemT [21]. Due to space limitations we cannot give an in-depth description of the two, but we give an overview of how they work and discuss their capabilities in

what follows. Currently, both extractors work at a sentence level, taking as input a set of concept annotations computed by entity annotators, e.g. UMLS annotator, and a custom concept annotator that exploits the surface forms present in the ontology, and returning a KG of BRs. Internally, a first graph consolidation step is performed in order to verify that the KGs are consistent with the ontology definitions.

WatsonX Approach. WatsonX is built as a classical NLP deep parsing pipeline implemented as an Apache UIMA application. Originally it was part of the IBM Watson system that won the Jeopardy challenge against human experts in 2011 [20]. It receives as input a sentence and identifies syntactic, morphological and semantic elements of the sentence, building a dependency parse tree (see example in Fig. 5). In a dependency tree, nodes are dependant of other nodes on the tree and that dependency is a labelled edge representing grammatical relations like *nominal subject, direct object, object of a preposition.* Dependency representations are useful for relation extraction because they can connect terms even if they are not subsequent in the original sentence.

From the dependency tree, a set of linguistic-based subtree extraction rules are executed in order to identify potentially interesting linguistic PAS (Predicate Argument Structure) tuples. These tuples represents relationships across the textual entities, in the form of <subject, predicate, object, object modifiers>, and other functional dependencies that can be expressed as linguistic rules over the dependency tree, such as a noun with object prepositions or adjective modifiers, e.g. in Fig. 5 <member, receive, $1000, up to>, <adult, member>, <$1000, benefits, dental>, <$1000, benefits, year>. Then, an ontology reasoning component translates functional dependencies in the sentences (PAS) to semantic relations, i.e. ontological statements. First, the textual entities in the PAS tuple are matched to ontological entities, based on a search over the entity labels. Second, PAS tuples are matched into a Graph Patterns (GPs).

The search of GPs across the combination of relevant entities and datatypes within a PAS is guided by domain-independent pattern templates. Given any of the combinations between the candidate entities matched in a PAS, the system searches for the patterns (or combinations) that better translate the PAS tuple, i.e. cover most of the terms in the tuple, and if the found GPs are semantically compliant with the ontology it adds them to the output KG. A pattern consists of variables (preceded by "?") that must bind to an ontological resource, parameters to substitute by the candidate matches of the type sought, e.g., a class, property, instance or datatype (in between <>) and the target variable (*?target*) to instantiate. In our example, for the PAS tuple <adult, member>, the pattern fired between the matched instance *adult* (of type *AgeRange*) and the class *MemberEligibility* is: *?target rdf:type <Class>. ?target ?property <Instance>*, that identifies *hasAge* as a valid binding for *?property*, resulting in the instantiated pattern *?target rdf:type <MemberEligibility>. ?target <hasAge> <adult>*. For other PAS tuples multiple combinations of patterns can be executed and intermediate nodes may be inferred in order to find a path between two resources connected in the ontology. A BR is then created by joining together all GPs obtained from the connected PAS tuples, i.e. those that have a *join* term. The resulting BR (shown in Fig. 3) can be consolidated with other BRs created from other subtrees in the sentence or across sentences.

SystemT Approach. SystemT is an industrial-strength declarative rule-based IE system. Borrowing ideas from database systems, commonly used text operations are abstracted away as built-in operators […] and exposed through a formal declarative language called AQL (Annotation Query Language) [21]. The output of the execution of an AQL query is a set of tuples in tabular form (see example in Fig. 4). As a first step, the implemented extractor explores the ontology structure and annotations to dynamically generate, for each property of interest in the ontology, a corresponding set of extraction queries in AQL based on multiple property templates. These extraction queries aim at extracting candidate ontology pairs, i.e. <property, range>, based on the annotations resulting from the available concepts annotators, ((2) in Fig. 1). For example, given the annotations in Fig. 5, <receive, dental benefits> may be selected as a candidate pair for the *hasApplicableService* property. These queries combine different extraction approaches based on the characteristics of the target property, e.g. property type, range types, polarity, as well as the syntactic and semantic information available for the examined sentence. The property-range extraction approaches can be divided into two main categories: (1) semantic-based approaches, that make use of the results of a shallow semantic parsing of the input text[1], and they reason over semantic roles, actions and contextual information, (2) distance-based approaches, used as fall-back strategies when the semantic information is partial or missing, e.g. due to incomplete or grammatically incorrect sentences, and are based on the sequence and distance between a property-trigger and a corresponding candidate range annotation. For example Fig. 5, in a strategy of type (1) extracts the condition *'applicable service: dental benefits'* by taking into account the main action, its polarity and the connected theme (see Fig. 4), while a strategy of type (2) extracts the condition *'max reimbursement: 1000$'* by considering the proximity of the property's trigger *"up to"* and the candidate range value *"1000$"*.

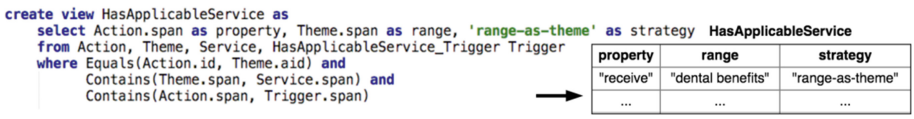

Fig. 4. Example of simplified AQL query and resulting tabular output

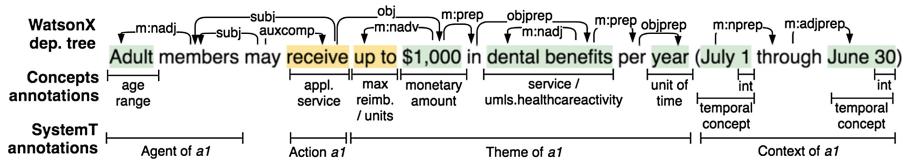

Fig. 5. Example of annotations extracted from the Concepts Annotation component and from the WatsonX and SystemT tools.

[1] The shallow semantic parsing of the sentence is performed through a natural language understanding capability of SystemT, currently under development, that computes and exposes information regarding the semantic roles present in the sentence, e.g. actions, agents, themes and contextual information of those actions, together with information regarding voice, polarity, etc.

All the applicable extraction queries are executed for each property of interest producing a set of candidate ontology triples where the domain can be inferred due to the previously described constraints on the ontology. The confidence score of each triple is dependent on the approach that generated the triple and on the information taken into account, e.g. semantic approaches are usually stronger than distance-based approaches and an explicit property trigger is stronger than an inferred trigger. As a last step, the extracted candidate pairs are then filtered and ranked, based on the associated confidence scores, in order to enforce the intuition that a specific span of text in the sentence can be associated only to a single condition in the resulting benefit rule, e.g. "1" can be either part of a date range or of a unit limitation. For each sentence, a BR is populated with all the selected triples.

Filtering and Consolidation. A fundamental step is the consolidation and filtering of the KGs created by the different extractors. This phase try to accomplish various goals: (1) enforce all the constraints expressed in the ontology, e.g. disjointness between two properties, min and max cardinality; (2) consolidate BRs extracted from different portions of the policy text, currently different sentences of the same paragraph; (3) consolidate BRs extracted by different extractors on the same policy text; (4) discard BRs that are noisy. We give an overview of the approaches adopted in the following.

Ontology Constraints Enforcement. The consolidation strategies that fall in this category are meant to find and solve ontological constraints violations, e.g. *max cardinality* constraints to enforce, for example, the presence of a single *age range* per BR. These are the main strategies executed in the internal consolidation stage for each extractor and are based on the statements' confidence scores.

Consolidation Across Sentences. These consolidation strategies are meant to merge information extracted from different sentences in a policy. Consider the example *"Adult members may receive up to $1,000 in dental benefits per year (July 1 through June 30). Emergency and denture benefits are not subject to this limit."* we would like to capture the information that *emergency benefits* and *denture benefits* are not covered by this rule. As a first implementation, assuming that the extractors capture the relevant information in two BRs, one per sentence, these BRs are merged if the resulting BR does not violate any ontological constraint. In our example, the merge will succeed extending the BR in Fig. 3 with the additional condition *excluded services*. Instead, if considering the following as second sentence in our example "*Emergency and denture benefits are not covered for children."* the merge would fail due to the conflict between the different *age range* conditions, i.e. *adults* and *children*. Currently, only the sentences belonging to the same paragraph of text are considered as candidate for this kind of consolidation, but we aim to improve the results of this strategy, e.g. by looking at explicit co-references between sentences.

Consolidation Across Extractors. When different extractors generate a BR for the same policy text, we want to output a consolidated result, if possible, to avoid repetitions and duplication. As a baseline strategy all BRs that are a subset of another BR from the same sentence are merged together. More elaborated strategies can be implemented i.e. to merge BRs that share most of the properties and can be merged

without leading to (ontology) conflicts. More work is needed to implement strategies able to handle the disagreement between extractors, e.g., based on the confidence scores.

Noise Filtering. Finally, we implemented some filtering strategies to remove BRs whose structure are too simple to be valid, as such we filter out: (a) BRs with only one *root* property, e.g. describing only a *member eligibility requirement*; (b) sentences, and corresponding BRs, that do not contain any explicit *property trigger* (i.e., no ontological relation was detected across any of the entities mentioned in the sentence) for the specific BR type; (c) BRs that contain less than n condition values. Different strategies can be selected depending on if we want to maximise precision or recall.

5 Evaluation

Here we describe the ground truth, first set of metrics and evaluation framework used to assess the quality of the KGs extracted from policy text. To obtain a Ground Truth (GT), expert Investigators sampled pages from Medicaid policy documents and modelled an 'expected' set of structured BRs for them. The BRs were modelled using the same user-friendly 'flat' UI representation described in Sect. 4.1 (see Fig. 2). Guided by the ontology, this UI enables investigators to express policy knowledge by selecting structured combinations of conditions and entities or datatype values (to define BRs). Two investigators and a senior investigator lead worked together to create and agree on the GT, which was subsequentially peer-reviewed by a wider investigation team. To measure how well our approach generalises across different policy areas, Ground Truth was created for two areas: Physical Therapy and Dental Services. Once the investigators input the GT in the system, the evaluation framework is fully automated (configurable to use one or both extractors and different consolidation strategies). This enables us to incorporate and evaluate new policy domains.

Standard Precision(P) and Recall(R) metrics are adapted to measure the quality of extracted knowledge against the GT. However, in this scenario we cannot focus only on the quality of an individual pair of *condition* and *value*, but we need to consider the overall quality of the BRs extracted which combines multiple conditions and values. We focus our performance evaluation at both an overall knowledge-extraction (BR) level (how well the extracted BRs match the Investigators' expectations), as well as the contribution of individual elements (condition and values) to overall performance.

5.1 Performance Metrics and Results

The evaluation metrics are calculated by comparing automatically-extracted BRs to investigator-defined BRs (GT) on the same policy text. BRs are always compared in their 'flattened' form and are sorted based on the order they appeared in the policy text.

Ground truth BRs $R^{GT} = \left\{R_j^{GT}\right\}, j = 1, \ldots, n_{GT}$ are paired with those automatically extracted from the same policy text $R^E = \left\{R_i^E\right\},\ i = 1, \ldots, n_E$. They are identified as (a) Exact matches, if all condition-value pairs $\{c_k : v_k\}, k = 1, \ldots, L_i$ are identical; (b) Partial matches, if some conditions are missing or values of identical conditions differ, or (c) Not matched, if no identical condition was identified or if there was no rule coming from the same policy text. For every pair of partial matches BR $\left(R_j^{GT}, R_i^E\right)$, a similarity score S_{ji} is calculated (1) and the GT BR is matched to the extracted BR with the highest *similarity score*. This pair is then removed from each set before the process continues.

$$S_{ji} = \frac{\min\left(L_i, L_j\right)}{\max\left(L_i, L_j\right)} * \frac{1}{L_j} * \sum\nolimits_{k=1}^{L_j} score_{c_k} \tag{1}$$

where L_j and L_i are the sizes of R_j^{GT} and R_i^E correspondingly (i.e., how many conditions each BR consists of), $\frac{\min\left(L_i, L_j\right)}{\max\left(L_i, L_j\right)}$ represents a penalizing factor when the sizes of the two BR are not the same (rule length similarity). The score for each condition value pair $\{c_k : v_k\}$ is calculated (2)

$$score_{c_k} = \begin{cases} 0, & \text{if } c_k \text{ is captured in } R_j^{GT}, \text{but not in } R_i^E \\ 1, & \text{if } c_k \text{ is captured in both } R_j^{GT}, R_i^E \text{ and } v_k \text{ is the same} \\ 0.5 + 0.5 * f_1 * \left(1 - \frac{1}{Ca_k}\right), & \text{if } c_k \text{ is captured in both } R_j^{GT}, R_i^E \text{ and } v_k \text{ differ} \end{cases} \tag{2}$$

Here, f_1 is the harmonic P-R mean generated by comparing the values of c_k in R_j^{GT} and R_i^E and $\{Ca_k\}$ is the number of semantically compatible candidate values a condition may have in the ontology (i.e., instances of the same type, such as all known medical programs), for datatypes Ca_k *is* 1.

Precision (P) measures the proportion of extracted rules that match the GT. Recall (R) measures the proportion of GT rules correctly extracted. f_1 combines these two. Specifically, they are defined as follows (3):

$$P = \frac{n^\circ \textit{ exact matches} + n^\circ \textit{ partial matches}}{n^\circ \textit{ extracted rules}} \quad R = \frac{n^\circ \textit{ exact matches} + n^\circ \textit{ partial matches}}{n^\circ \textit{ GT rules}}$$

$$f_1 = 2 * \frac{P * R}{P + R} \tag{3}$$

In the GT, we have 50 manually created BRs extracted from 27 pages of Dental policy, and 25 manually created BRs extracted from 14 pages of Physical Therapy policy. The evaluation metrics for each scenario are presented in Table 2.

Table 2. Evaluation results: P, R, f_1, n° of exact and partial matches, average $score_{ck}$, n° of extracted BRs not matched to the GT (FP) and n° GT BRs not matched by extracted BRs (FN)

	P	R	f_1	n° exact	n° partial	Avg. s_c	FP	FN
Approach (dental policy)								
WatsonX only	**0.84**	0.54	0.66	4	23	0.5	5	23
SystemT only	0.70	0.66	0.68	5	28	0.56	14	17
Consolidated (subsets)	0.75	**0.82**	**0.78**	6	35	0.54	14	9
Consolidated (non-conflicts)	0.78	0.64	0.70	6	26	0.61	9	18
Approach (Physical Therapy)								
WatsonX only	**0.94**	0.64	**0.76**	6	10	0.63	1	9
SystemT only	0.64	0.72	0.68	4	14	0.56	10	7
Consolidation (subsets)	0.64	**0.84**	0.73	7	14	0.60	12	4
Consolidation (non-conflicts)	0.69	0.80	0.74	7	13	0.60	9	5

As expected, WatsonX deep-parsing, yields more precise BRs but also misses relations that were not captured from the dependency tree (e.g. because of ill-formed or complex sentences). SystemT applies a shallower NLP approach as well as different techniques to maximize recall. Two consolidation strategies are compared, the first one merges all BRs that are subset of another and the second one merges BRs if it doesn't lead to ontology conflicts. They offer a good compromise between (P) and (R). Combining both extractors increases (R) with a relatively small impact on (P). Further work could look at more sophisticated consolidation strategies that are able to leverage the confidence scores assigned by the extractors when merging statements across BRs, while detecting unresolved conflicts that require investigator input.

The evaluation framework also measures the contribution of individual elements to overall performance – specifically, P and R for each condition-value pair (see examples in Fig. 6), which helps with iterative enhancement and debugging of the extractors.

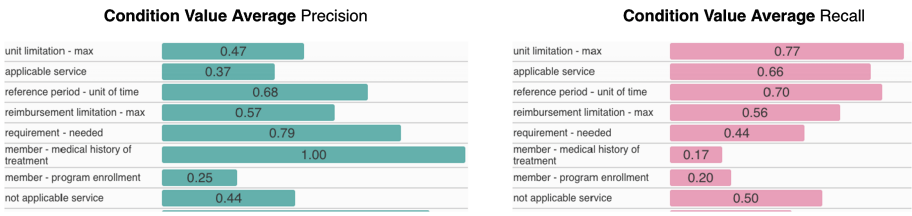

Fig. 6. Visualisation of P/R results on a subset of conditions when considering condition values.

5.2 Discussion

Here we look at areas where further work is likely to improve performance:

Automatic Extraction of Condition-Value Pairs: (1) a parse tree may not capture implicit semantic relationships present in policy text which can lead to failures capturing values. For example, a date range in brackets at the end of a sentence about

waiver programs describes the period when that program is active. If the parse tree does not pick up the dependency, no values will be extracted for the relevant date range (in order to balance P & R, our extractors avoid making relational inferences in these cases); (2) Contextual values may be set outside of the sentence or paragraph being extracted - e.g. by titles like "Orthotics for Adults" or introductory sentences, or table headings (for text inside table cells). (3) Parsing errors can be introduced during the policy PDF ingestion which in turn lead to incorrect value extraction downstream in the extractors.

Mismatches Between GT and Extracted Condition-Value Pairs: (1) Some fields may have alternative valid representations leading to them being incorrectly measured as misses during the automatic evaluation - e.g. when the GT contains *'1 year'* but an extractor gets *'12 months'*; (2) Investigators modelling expected BRs (GT) may include knowledge that does not come from the policy text - e.g. modelling "high risk for caries" in the GT when they see a type of tooth surface in policy text that they happen to know is prone to caries. While extractors cannot make these inferences per-se, some progress may be achievable via an ontology hierarchy (specifically subclassOf and partOf). This could be used to infer some relationships, such as 'anterior teeth' from a reference 'canine' or 'incisor'. More work is needed here, in particular for temporal expressions.

Invalid Condition-Value Pairs and BRs: (1) Paragraphs that mention relevant entities (e.g. a healthcare service and a program, or body parts) but do not describe limitations or other policy knowledge may still result in an BR being extracted, we describe these BRs as false positives or 'noise' and measure them via Precision (P). (2) Different extractors may produce conflicting BRs that cannot be merged. When this happens, one is selected and the other is measured as 'noise'. For performance measurement, the one with the best-matching similarity score to the GT is selected; (3) A sentence may lead to two different BRs, for example, an 'orthotic' policy, expressing different service coverage for adults and children will be extracted as one BR for adult orthotics and another BR for child orthotics. Similarly two BRs may be consolidated into a 'logical' BR. For example, a unit limitation might refer to either a combination of procedures, implying the need to create only one BR covering all procedures, or to each procedure individually, in which case a separate BR limitation should be created for each. Due to ambiguity in text, two BRs may be have been erroneously consolidated into one (e.g., "Members [] are eligible for any combination of up to four (4) prophylaxes or up to four (4) periodontal maintenance visits"). We aim to address this challenge firstly, by more advanced BR consolidation strategies utilising confidence scores, and secondly by exploiting information redundancy in policies.

Human/Machine 'Co-reasoning': The goal of our work is to enable Investigators to collaborate and 'co-reason' with tools like these, not merely to automate knowledge extraction from policy text. A central element of this is empowering human Investigators to interact with, curate and use the extracted knowledge. This was the rationale for creating the UI (see Fig. 2) and 'flat' KG representation early on. These have been central to both iteratively reviewing extractor output with expert Investigators, as well as helping them to construct a GT to support formal performance measurement.

Informal feedback from the Investigators about the UI representation has been very positive. In particular, we were gratified to find that they could take new policy areas and rapidly construct high-quality GT for them after only a few hours of acclimatising to the UI tool. In large part, this is due to the ontology (and hence the UI) being driven by concepts and structure derived from on their own BR templates. In future, we hope to use this approach to speed up the process of obtaining formal GT for measuring performance on new policy domains. Specifically, by automatically extracting BRs and having Investigators manually curate them into a formal GT (rather than creating GT manually by hand).

Impact on the Investigators' Workflow: Computable policies in the form of benefit rules enables a wide range of downstream benefits that can have a significant improvement to the investigators workflow. Examples of this include:

- Investigators always have a large backlog of investigations and they lack objective data on which to prioritize the opportunity landscape. Automatically constructed benefit rules could be executed against claims data to quantify systematic policy violations and support prioritization.
- Investigators need to provide strong evidence to support allegations of policy violations particularly for legal proceedings. Automatically constructed BRs can explicitly tie invalid claims to the policy constraints that they are violating.
- Additionally, through curation of the automatically constructed BRs we are building institutional knowledge on correct and complete policy BRs relevant for investigation cases. This enables consistency in policy reviews. The BR representation serve as a means for policy data insights, validation and sharing of knowledge across team members with varied levels of expertise and diverse skillsets. As such the BRs can inform development of new algorithms or enable modification of existing algorithms to make them more precise, targeted and complete.

6 Conclusion and Future Directions

We have developed a semantic system to extract a KG of *actionable* BRs from healthcare policies. The ontology is designed to balance expressiveness of the extracted knowledge with the ability to represent it in a simple, unambiguous, human-readable way to support policy comprehension and curation. The engagement with our target users (investigators) early in the development and throughout the continuous delivery process was key for the successful adoption of our semantic solution.

We presented a first validation of the semantic solution with investigators and showed solid progress in two vertical domains. Most of the effort required to generalise is on identifying external data linked to the ontology (i.e., instance data) that is state-specific, such as programs and codes that are not part of the federal code set. Nonetheless, we found a strong degree of re-usability in the core concepts between the two domains (i.e., same BR modelling was applicable), making this an excellent domain for the application of SW technologies. More BR types are being incrementally added for the next version of the system, thus incrementally improving the scope of

information available to Investigators doing policy research. By tying together these BRs and the policy text from which they are derived, investigators can build the evidence necessary to make a case for recovery of inappropriately paid claims.

Further planning is in process to cover more policy areas and assessing both the value and viability of this technology for large-scale deployment across several domains. We aim to provide quantitative metrics on usability, increased productivity in the context of investigations (e.g., not just on time-saved but on whether this solution supports our users' prioritization of investigations that are likely to result in additional money recovery) and generalisability, scaled across policy domains. To this end, we hope to transition from manually-created GT to automatically extracted and manually-curated GT, which we expect to be considerably more efficient.

There is much room to improve performance, such as the ability to induce domain specific reasoning patterns. We aim to investigate approaches for classifying policy paragraphs that contain BRs (using labelled data collected via the UI), as this will reduce noise BRs by filtering out irrelevant paragraphs; as well as approaches for learning patterns not yet be explicitly captured in the ontology. We aim to experiment with unsupervised approaches to find paraphrases and to augment partially-populated KG. Finally, our hope is that extracting high-quality, computable representations of policy knowledge will ultimately lead to new, policy-informed ways of analysing claims data.

References

1. https://www.nhcaa.org/resources/health-care-anti-fraud-resources/the-challenge-of-health-care-fraud.aspx. Accessed Apr 2019
2. https://truvenhealth.com/media-room/press-releases/detail/prid/127/truven-health-analytics-professionals-receive-accredited-health-care-fraud-investigator. Accessed Apr 2019
3. https://www.gao.gov/key_issues/medicaid_financing_access_integrity/issue_summary. Accessed Apr 2019
4. Chandola, V., Sukumar, S.R., Schryver, J.C.: Knowledge discovery from massive healthcare claims data. In: Proceedings of the KDD, pp. 1312–1320 (2013)
5. Joudaki, H., Rashidian, A., Minaei-Bidgoli, B., Mahmoodi, M., et al.: Using data mining to detect health care fraud and abuse: a review of literature. Glob. J. Health Sci. **7**(1), 194–202 (2015)
6. Waghade, S.S., Karandikar, A.M.: A comprehensive study of healthcare fraud detection based on machine learning. J. Appl. Eng. Res. **13**(6), 4175–4178 (2018)
7. Wimalasuriya, D., Dou, D.: Ontology-based information extraction: an introduction and a survey of current approaches. J. Inf. Sci. **36**(3), 306–323 (2010)
8. Martinez-Rodriguez, J.L., Hogan, A., Lopez-Arevalo, I.: Information extraction meets the Semantic Web: a survey. Semant. Web 1–81 (2018, pre-press)
9. https://tac.nist.gov/2017/KBP/ColdStart/index.html. Accessed Apr 2019
10. Ben Abacha, A., Zweigenbaum, P.: Automatic extraction of semantic relations between medical entities: a rule based approach. J. Biomed. Semant. **2**(5), S4 (2011)
11. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of ACL and AFNLP, vol. 2, pp. 1003–1011 (2009)

12. Glass, M., Gliozzo, A., Hassanzadeh, O., Mihindukulasooriya, N., Rossiello, G.: Inducing implicit relations from text using distantly supervised deep nets. In: Vrandečić, D., et al. (eds.) ISWC 2018. LNCS, vol. 11136, pp. 38–55. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00671-6_3
13. Peng, N., Poon, H., Quirk, C., Toutanova, K., Yih, W.: Cross-sentence N-ary relation extraction with graph LSTMs. Trans. Assoc. Comput. Linguist. **5**, 101–115 (2017)
14. Saggion, H., Funk, A., Maynard, D., Bontcheva, K.: Ontology-based information extraction for business intelligence. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 843–856. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_61
15. Corcoglioniti, F., Rospocher, M., Aprosio, A.P.: Frame-based ontology population with PIKES. IEEE Trans. Knowl. Data Eng. **28**(12), 3261–3275 (2016)
16. Piro, R., et al.: Semantic technologies for data analysis in health care. In: Groth, P., et al. (eds.) ISWC 2016. LNCS, vol. 9982, pp. 400–417. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46547-0_34
17. Grimm, S., Abecker, A., Völker, J., Studer, R.: Ontologies and the semantic web. In: Domingue, J., Fensel, D., Hendler, J.A. (eds.) Handbook of Semantic Web Technologies, pp. 507–579. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-540-92913-0_13
18. W3C Recommendation. https://www.w3.org/TR/csv2rdf/. Accessed Apr 2019
19. Noy, N., McGuinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Medical Informatics Technical Report SMI-2001–0880 (2001)
20. Kalyanpur, A., Boguraev, B., Patwardhan, S., Murdock, J.W., et al.: Structured data and inference in DeepQA. IBM J. Res. Dev. **56**(3), 10 (2012)
21. Chiticariu, L., Danilevsky, M., Li, Y., Reiss, F., Zhu, H.: Systemt: declarative text understanding for enterprise. In: NAACL-HLT, pp. 76–83 (2018)