MEDINFO 2021: One World, One Health – Global Partnership for Digital Innovation
P. Otero et al. (Eds.)

doi:10.3233/SHTI220081

Learning Insurance Benefit Rules from Policy Texts with Small Labeled Data

Gabriele Picco[a], Hoang Thanh Lam[a], Vanessa Lopez[a], John Segrave-Daly[b], Miao Wei[b], Marco Sbodio[a], Inge Vejsbjerg[a], Seamus Brady[b] and Morten Kristiansen[b]

[a] IBM Research Europe, Dublin, Ireland
[b] IBM Watson Health, Dublin, Ireland

Abstract

To protect vital health program funds from being paid out on services that are wasteful and inconsistent with medical practices, government healthcare insurance programs need to validate the integrity of claims submitted by providers for reimbursement. However, due the complexity of healthcare billing policies and the lack of coded rules, maintaining "integrity" is a labor-intensive task, often narrow-scope and expensive. We propose an approach that combines deep learning and an ontology to support the extraction of actionable knowledge on benefit rules from regulatory healthcare policy text. We demonstrate its feasibility even in the presence of small ground truth labeled data provided by policy investigators. Leveraging deep learning and rich ontological information enables the system to learn from human corrections and capture better benefit rules from policy text, beyond just using a deterministic approach based on pre-defined textual and semantic pattterns.

Keywords:

health policy, deep learning, ontology

Introduction

Health and Social care Programs have a major impact on outcomes for vulnerable citizens. To ensure resources are distributed fairly to citizens, and are not lost to Fraud, Waste or Abuse (FWA), these Programs typically have large bodies of policy text describing benefit rules. In practice, however, only a fraction of these rules get automated and the lack of coded rules creates major problems down-stream. Firstly, regulators tasked with protecting the Program's financial integrity cannot get an overview of the "compliance landscape" to identify FWA and prioritize investigations. Secondly, FWA detection is typically based on statistical and data-analytic approaches [1][2], making difficult for investigators to determine which (if any) policy may have been violated. Finally, those being regulated (Providers) have no fast/automatic way to check the compliance of their claims, leading to friction, cost and avoidable errors.

Recently, the "Rules as Code" movement [3] has emerged as a response to these issues. It calls for policy rules to be published as "digital twins"— in both human and machine-consumable form— making them amenable to automated compliance-checking, fairness-checking, loophole identification, and what-if analysis, as well as building trust through transparency.

In this vein, our prior work [4][5] describes a deterministic approach to extracting "benefit rules" from healthcare policy, in a form that is both human and machine consumable. Each rule is associated to the policy text from where it was extracted, which facilitates easy review/validation and correction by medically-aware users before use.

In this paper, we move this approach forward by proposing a model to (1) learn from the addition/correction of new rules and (2) identify new elements, beyond the "vocabulary" on which the model was trained.

To achieve this, we combine neural and symbolic approaches. A neural NLP model learns which spans, i.e., one or more consecutive words, in policy text are rule conditions— e.g. services that are "mutually exclusive" (not payable when billed together). A domain ontology built with investigators [5], supports both human understanding/oversight and acts as a blueprint for constructing semantically-meaningful rules from the conditions labeled by the neural model— e.g., determining if a complete "mutually exclusive" rule can be formed. To train the NLP model, we use a small set of 141 benefit rules identified by professional investigators in dental policy documents from two US states1. For each rule, we annotate the rule text with ontology-aligned labels describing the different condition types. Two models are then trained: first, a classifier model to assess which paragraphs contain benefit rules (many do not), and second a model to predict and label specific spans.

We evaluate results against a gold standard of policy rules obtained from professional investigators and compare the results against the existing deterministic baseline, presented in [5] and find that this fusion of neural and symbolic approaches (1) improves extraction performance, with a small up-front labeling cost and a surprisingly small set of benefit rules provided by investigators; and (2) correctly labels previously-unseen elements in rule conditions— e.g. healthcare services that are not (yet) present in our ontology or terminology. In short, this approach shows good promise for lowering extraction cost and improving rule quality, ultimately helping investigators to review health and social care policies and support the extraction of actionable rules.

Methods

Benefit Rule Extraction Pipeline

In this section, we briefly describe the existent pipeline to extract benefit rules from healthcare policy. First, an off-the-shelf PDF conversion tool [6] is used to obtain an HTML

[1] Ontology and benefit rules to be made available in https://github.com/IBM/rules_extraction_from_healthcare_policy

representation with headings, paragraphs and sentences. These paragraphs are filtered via a BERT-based classifier. No other text pre-processing or cleaning is performed. The classifier simply filters out policy paragraphs that do not appear to contain any rules, thereby reducing noise, time, and computational load for the rest of the pipeline.

Each paragraph may contain zero, one, or more benefit rules. An example of a paragraph with two different benefit rules types can be seen in Figure 1. While the wording of policy texts can differ significantly between geographical regions or policy topics, all policies set out similar guidelines based on common compliance concepts such as eligible patients (e.g., based on age, medical history), billable places of service (e.g., home, hospital), maximum billable units of service or monetary amounts per member in a given period, services that should not be billed together, etc. This shared conceptualization behind benefit rules is captured in a domain ontology. The ontology is used to ensure semantically-meaningful rules are extracted from the policy text. For example, the property “applicable service” expects a type of billed service as its value, or that there can only be one applicable time period per benefit rule (max cardinality of 1). The ontology also links to other data sources with relevant terminology, such as codes for billable dental services [8] or places of service [9], and guides the annotation and extraction of relevant domain entities, relationships and the logical constraints that underpin them.

Expert-labeled datasets are expensive to develop (and necessarily small), therefore a deterministic approach to extract rules from policy texts was proposed in [5], where after annotating the mentions of entities and relations based on the ontology, NLP tools are applied to identify functional dependencies between the annotated ontological terms as well as their semantic role in the sentence (actions, agent, theme of the action, polarity, etc.). Then, based on a combination of linguistic rules and domain-independent semantic patterns to reason over the ontology, textual dependencies are translated into meaningful benefit rules. For example, a number will be interpreted as a unit, amount of time, or age limitation by considering its semantic proximity to an annotated property for an age range, a unit limit, or time period (both expecting a number as the range).

The extracted benefit rules are presented to the user in the form of editable condition – entity/value pairs, that enable investigators to review the extracted rule against the policy text and correct extraction errors and omissions. These validated benefit rules form a shared store of high-quality, machine and human readable rules, making for a more transparent rule creation and consumption. However, the deterministic extraction is unable to learn from those human corrections, and the approach is limited by, first, the coverage of the ontology annotators and the need to map policy text to known ontology entities; and second, the use of a rule-based approach to identify linguistic dependencies and semantic patterns among those relevant entities that can be transformed into semantically meaningful connections to build a benefit rule.

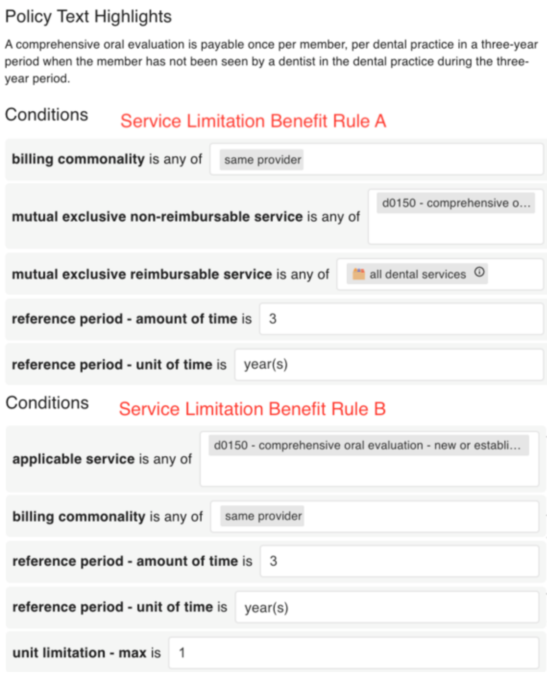

Figure 1: Benefit rules extracted from a paragraph in a dental policy [7]. On top, a Mutually Exclusive rule on services that cannot be billed together in a given period. On the bottom, a Service Limitation rule on the units of service a provider may bill per member over a time period

We propose a deep learning approach to predict textual spans that constitute conditions of a benefit rule. That is, not just saying that a “comprehensive oral evaluation” is an instance of a service, but that in the context of the sentence it has two distinct roles simultaneously: an applicable service and a mutually exclusive non-reimbursable service, for a Service Limitation rule and Mutually Exclusive rule respectively.

Benefit Rule Span Prediction

We consider the rule extraction as a span prediction problem. Let $\boldsymbol{s} = \boldsymbol{t_1 t_2 \ldots t_m}$ be a sentence with m tokens and V the set of labels. Each token t is associated with one or more labels from V. Assume that we have k labels in our problem, i.e., $|V| = k$, e.g., for our policy rule extraction, we have k=27 labels with examples provided in Figure 2 which shows sentences with associated labels.

A span can have more than one label. Therefore, we use BERT [10] as a backbone network and we add a multi-class classification head on the top of it to enable multi-class classification. We trained the models with Adam optimization [13], mini-batch size 8, learning rate 0.00001 and 100 epochs.

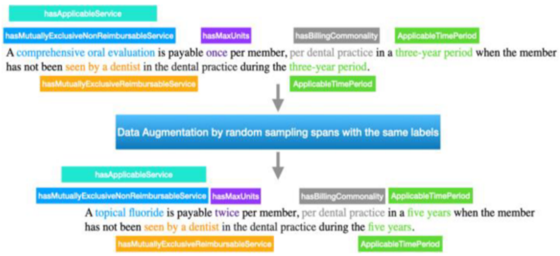

Figure 2 Data augmentation: given an input labeled sentence, a new sentence is created by randomly sampling the spans from the set of spans with the same labels in V.

Data Augmentation. Investigators created benefit rules that were used to manually label span fragments in 141 policy paragraphs using the 27 labels in V, one for each benefit rule condition defined in our ontology, that our deep learning models can work on. The label schema was selected to fully cover all relevant conditions that may be included in a benefit rule, but also that is simple enough for human annotators. While this process can be automated to some extent, human supervision is needed to match the values in a benefit rule to spans in the sentence. Our main challenge during the labeling process is the difference in the way annotators interpret the concepts in an unfamiliar domain. Time-consuming cross-checks were carried out to ensure the consistency between labels provided by different annotators in the team. We ensured each label was reviewed by at least two annotators based on a set of annotation gudeliness to ensure consistency. For example, the label *billing-commonality* covers the span "per dental practice" (including the preposition "per"), as billing-commonality is the condition used by domain experts to indicate that the claims need to be billed by the same provider.

Given the fact that acquisition of labeled data for a new policy is expensive, we consider enriching the available labeled data using data augmentation. In a data augmentation algorithm, an input sentence is perturbed to create a new sentence that is expected to have the same meaning or preserving the semantic structure of the sentence but may accept a slightly difference in lexical representations. Generally speaking, data augmentation for text is a hard problem because a small perturbation of the sentence can create a sentence with a completely new meaning or with a different semantic structure. Therefore, we propose a controlled data augmentation to preserve the main semantic structure of the rules where the spans are perturbed but the labels are persisted.

For each label v in V, let Span(v) be the collection of all the spans that are labeled as v in our training data. For every input labeled sentence among our 141 paragraphs, we look at each span with label v in the sentence and randomly replace the given span with a random span sampled from the set Span(v) to create a new sentence used for training purposes. This data augmentation method is simple but it is very effective as demonstrated in the experiments. Figure 2 shows an example with an input sentence and a newly generated sentence using this method. Since the structure of the sentence is preserved, swapping spans between sentences helps the model generalize better as it is forced to learn the hidden structure of the input rather than remembering the actual span contents.

Model fine-tuning. Our work relies on pretrained language models such as BERT. Since BERT is trained on public domain texts, a popular practice when dealing with domain specific text is to fine-tune these models with domain specific texts. We consider two approaches to this. The first one fine-tunes BERT using all policy texts available in our data. The second method only fine-tunes BERT on the rules. Figure 3 shows the span prediction F1 scores of different approaches. The combination of BERT, data augmentation and fine-tuning with rule text yields the best result so it is the default choice in our experiments.

Methods	5-fold crossvalidation F1
BERT	0.77
BERT + Fine-tuned with all texts	0.79
BERT + Fine-tuned with rule texts	0.78
BERT + Augmentation	0.79
BERT + Augmentation + Fine-tuned with rule texts	**0.8**
BERT + Augmentation +Fine-tuned with all texts	0.78

Figure 3 F1 score of different approaches for span prediction in a 5-fold cross-validation settings. The combination of BERT, data augmentation and fine-tuning with rule text yields consistent improvement over BERT alone.

Ontology-based span prediction to rules

The predicted labels, which capture the structural information of the rules, are combined with the entity annotations to construct the benefit rules. For each sentence, all the labels predicted by the span predictor that are compatible with a type of benefit rule (e.g., those corresponding to potential conditions in a Mutually Exclusive rule as defined in the ontology) are clustered. Subsequently, we identify for each cluster the entity annotations that overlap with each label and whose type is compatible with the range of the predicted label, which corresponds to a condition defined in the ontology.

At this point, for each label there will be one or more matching annotations that are used as a value to build the benefit rule. If there is no match, the text covered by the label can be used to infer a new entity of the type expected by the condition label. In our example there are two clusters, corresponding to a Mutually Exclusive and a Service Limitation benefit rule. The span predictor predicts the condition label "hasMutuallyExclusiveNonReimbursableService" that covers the text "comprehensive oral evaluation" (Figure 2) and is compatible with a Mutually Exclusive rule. The ontology annotators identify "d0150 - comprehensive oral evaluation" as an instance of a "Procedure Code" type and the annotated span overlaps with the span of the predicted label. Moreover, the type of the instance is compatible with the expected range of the condition in the ontology, therefore, the condition (whose display name in the ontology is defined as "mutually exclusive - non-reimbursable service" as shown in Figure 1) is assigned the instance value "d0150 - comprehensive oral evaluation" and will be one of the conditions of the built benefit rule.

In the example, the textual span "seen by a dentist" is labeled as the non-reimbursable service, while "seen by a dentist" obviously does not correspond to an actual instance of a billed service in the ontology, in the context of this sentence investigators interpret it as "any dental service". The prediction correctly recognizes that pattern from similar labeled rules. The extractor can optionally be configured to allow the creation of benefit rules with predicted values even if unrecognized in the ontology, while this may introduce inaccurate results (affecting precision) it does favour recall. Investigators can then validate unknown values while validating the benefit rule and either match them to existent ones (eg., in this case the service "all dental services") or optionally update the ontology with new instances and/or lexicalizations.

Results

Experimental Setup

We evaluate the system's accuracy in terms of precision/recall when extracting rules from text [5]. The evaluations are carried out with respect to a gold-standard of 141 rules created in consultation with our policy investigators for two US healthcare policies in the dental domain and compared with respect to the rule-based approach used as baseline. Given the small number of training instances, we obtained span prediction for 141 gold sentences in a 5-fold cross-validation setting. This technique guarantees that predictions are always made without overlapping between training and test sets. The span predictions for sentences that do not overlap with the 141 gold sentences (i.e., those that do not contain a benefit rule and are not filtered out by the classifier) are computed using a model trained with all the gold sentences. Instead, the span prediction for sentences *s* that overlap with 141 gold sentences are computed using a model trained without *s*.

Evaluation

We compare results with two configurations: default and optimal. Modifiable configuration includes selecting whether or not to add values not recognized by the ontology, choosing to use the classifier, filters, and rule-based consolidation strategies predefined in the pipeline [5]. The default configuration was reported in [5], while the optimal configuration is identified by optimizing for F1, using the Optuna package [11].

In the first experiment, using the default configuration, we compare the learning methods with the rule-based approaches. Table 1 reports the results for both Learn+Uv and Learn methods, which correspond respectively to the variant of the method with and without adding unknown spans to the final benefit rule. We also report results for execution with (+C) and without the classifier.

In the second experiment we compare the rule-based method (Rule) with the deep learning approach (Learn), reporting the maximum performances obtained by both methods, optimizing the pipeline configuration using the Optuna package and selecting the best execution from 200 trials in terms of F1 score. This experiment analyzes the two systems to the best of their performance and allows comparison of the approaches independently of the default configuration.

Table 1 – Method comparison using the default configuration: Learning method (Learn) and Rule-based baseline (Rule). (+Uv) Indicates where the option to keep spans unrecognized in the ontology was used (+C) Indicated if the classifier was active in the pipeline

Method	Precison	Recall	F1
Learn	69.85	58.23	62.26
Learn+Uv	56.48	**69.17**	61.53
Learn+C	**78.1**	54.78	62.82
Learn+C+Uv	67.19	66.36	**66.73**
Rule	60.57	67	63.34
Rule+C	69.55	63.72	66.47

Table 2 – Method comparison using the optimal configuration: Learning method (Learn). Learning method combined with Rule-based method (Learn+Rule) and Rule-based method (Rule).

Method	Precison	Recall	F1
Learn	**79.56**	66	72.03
Rule	76.54	67.37	71.57
Learn+Rule	70.77	**73.75**	**72.22**

Discussion and future work

Results reported in Table 1 and Table 2 demonstrate that the neural approach achieves comparable performance compared to the rule-based method, learning correlations from a surprisingly small set of benefit rules provided by investigators. Table 1 shows the effect of adding or ignoring values that are not part of the ontology in the extracted benefit rules and the trade-off between precision and recall. Intuitively, adding values that are not part of the ontology increases the recall of the system, extracting a greater number of rules present in the gold-standard, while reducing the precision of the system. The classifier, on the other hand, has the opposite effect, as it increases precision but reduces recall. This behavior is expected since the classifier in the pipeline operates as a filter, removing the paragraphs with low probability of containing benefit rules.

Table 2 shows the results of various methods in the optimal configuration environment. The precision of the learning method is higher than that of the rule-based method, and the recall values of the two methods are roughly similar. Although the precision of the Learning method combined with the Rule-based method has decreased, the recall has increased significantly from 67% to 73.75%. The F1 score is 72.22% which shows that the combination of the Learning method and the Rule-based method can obtain excellent comprehensive performance.

Furthermore, in Table 1, the use of the span prediction together with the classifier (Learn+C+Uv) achieved the best result due to the presence of entities that are not captured yet in the domain ontology and/or textual patterns that are notably hard to extract based on pre-defined linguistic and semantic patterns to build a benefit rule. The result demonstrated that machines are able to learn from examples of benefit rules validated by our investigators, without the need of linguistic and semantic experts to add more hand-crafted linguistic rules or reasoning patterns in the system [5].

The learning method presented requires an initial modeling phase, where the main concepts and relationships are defined and formalized in an ontology. While many of the concepts are shared and the ontology contains valid common concepts when applied in the same domain, as in the case of insurance policy texts of different US states, in order to apply the method to a new domain it is necessary to repeat the modeling phase. In future work we could analyze the use of state-of-art ontology learning system, such as [12], combined with the BERT-based extractor to extract rules in a different domain.

Conclusions

Governments and businesses everywhere are automating policy enforcement. When they do, the resulting rules becomes the "effective policy" that most citizens actually experience in their lives. That comes with great opportunities for fairness — both in enabling policy to be applied consistently at scale and in defending scarce resources from wasteful practices not compliant to policy. This latter point is often missed but it is critical in ensuring that vulnerable people can access the services they need. However, organizations that automate policy enforcement have a responsibility to ensure that there is no “translation of intent” errors when translating from policy, to business requirements, to rules, to code. A recent OECD report [3] on this issue identifies several ways to tackle this, some visionary (simultaneous code and policy development) and others practical (use AI and automation to shorten the route from policy to code). Our system takes this latter approach. This meant developing a structured representation of the benefit rules that feels natural and understandable to policy-aware users and is grounded to specific policy text. This core representation is captured by the ontology and is critical in ensuring that non-technical policy-aware individuals can understand and correct the rules.

In this paper we proposed a learning approach to predict benefit rule conditions from curated, small labeled examples that can be added by policy-aware users, and that can be assembled into actionable benefit rules using the ontology as a blueprint. We determined that the learning approach achieves similar performance to a pattern-based approach. While the training set is currently small, due to the need for domain experts and the cost of manual policy labeling, it is expected this will expand as investigators use the system to review and curate more benefit rules in any given policy domain.

We believe this work shows the potential impact of leveraging NLP and AI technologies with expert knowledge in an area where human understanding and control are important AI design concerns, such as safeguarding the integrity of government healthcare insurance programs.

This combination empowers investigators to be more effective and consistent in formalizing and validating human and machine interpretable rules at scale, and enables the system to learn from those corrections and improve its performance over time.

References

[1] J. Li, K.Y. Huang, J. Jin, et al. A survey on statistical methods for health care fraud detection. Health care management science (2008), 11(3): 275-87.

[2] H. Joudaki, A. Rashidian, B. Minaei-Bidgoli, et al. Using data mining to detect health care fraud and abuse: a review of literature. *Global journal of health science* (2015), 7(1):194.

[3] J. Mohun, A. Roberts. Cracking the code: Rulemaking for humans and machines. *OECD Working Papers on Public Governance*, No. 42, OECD Publishing (2020)

[4] T.S. Brisimi, V. Lopez, V. Rho, M. Sbodio, G. Picco, M. Kristiansen, J Segrave-Daly, C Cullen. Ontology-guided Policy Information Extraction for Healthcare Fraud Detection. *Medical Informatics Europe conference*, MIE 2020.

[5] V., Lopez, V., Rho, T.S., Brisimi, F., Cucci, M., Kristiansen, J., Segrave-Daly, J., Scalvini, J., Davis, G., Ferguson. Benefit graph extraction from healthcare policies. *International Semantic Web Conference,* ISWC 2019

[6] https://cloud.ibm.com/docs/compare-comply

[7] Iowa Department of Human Services (DHS) *Dental Services Provider Manual* (2013) Retrieved from https://dhs.iowa.gov/sites/default/files/Dental.pdf?012520210943

[8] Centers for Medicare & Medicaid Services (U.S.) *Healthcare Common Procedure Coding System (HCPCS)*. Centers for Medicare & Medicaid Services (2020).

[9] Centers for Medicare & Medicaid Services (U.S.). n.d *Place of Service Code* (2020). https://www.cms.gov/Medicare/Coding/place-of-service-codes

[10] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. ACL

[11] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework SIGKDD 2019.

[12] A. Khadir, H. Aliane and AGuessoum. 2021. Ontology learning: Grand tour and challenges. Computer Science Review. 39. 100339. 10.1016/j.cosrev.2020.100339.

[13] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).

Address for correspondence

Gabriele Picco

IBM Reseach Europe, Dublin, Ireland

Email address: gabriele.picco@ibm.com