Our Quest for Interpretable Natural Language Processing

Mihai Surdeanu

April 2020
Recent ML developments: a deal with the devil

- Interpretability
- Performance
Interpretability

• Interpretability is an overloaded term in machine learning (ML)
• But we can classify it roughly in two classes
1. Post hoc interpretability (Ribeiro et al., 2016)

Figure 3: Toy example to present intuition for LIME. The black-box model’s complex decision function $f$ (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using $f$, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.
2. Global interpretability:
Converting the statistical model into something interpretable
(aka "knowledge distillation")

(Craven and Shavlik, 1996) (Hinton et al., 2015)
Comparison

• Post hoc interpretability
  – May provide only an approximate explanation (Netflix lies to you 😊)
  – You can’t fix a problem with the original model when identified

• Global interpretability
  – May lose some performance in the conversion
  – Allows corrections to the model when problems are discovered
Comparison

- Post hoc interpretability
  - May provide only an approximate explanation (Netflix lies to you 😊)
  - You can’t fix a problem with the original model when identified

- Global interpretability
  - May lose some performance in the conversion
  - Allows corrections to the model when problems discovered
Comparison

- **Post hoc interpretability**
  - May provide only an approximate explanation (Netflix lies to you)
  - You can’t fix a problem with the original model when identified

- **Global interpretability**
  - May lose some performance in the conversion
  - Allows corrections to the model when problems discovered

This is our focus!
Globally interpretable models for natural language processing

1. Humans can understand the model
2. Humans can change the model
Academia vs. industry

Implementations of Entity Extraction

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Interpretable (rule-based)</td>
<td>3.5%*</td>
<td>45%*</td>
<td>67%*</td>
</tr>
<tr>
<td>Hybrid</td>
<td>21%$</td>
<td>22%$</td>
<td>17%$</td>
</tr>
<tr>
<td>Machine Learning</td>
<td>75%$</td>
<td>33%$</td>
<td>17%$</td>
</tr>
</tbody>
</table>
Why do we care about global interpretability?

• Most of today’s meaningful projects are interdisciplinary, e.g., ML + medicine

• We can’t expect an expert in another domain to understand (and fix!) our statistical classifiers

• But we need to iterate quickly...
The [any specific topic or issue here] seems to be buggy and consistently produces incorrect events. Some examples (I can provide more if needed):

- Expression of SIRT1, SIRT2, and acetylated (K120)-p53 in gastric cancer cell lines
  -> SIRT1 is acetylated
- Sirtuin deacetylase and can deacetyltransferins in order to regulate cell function.
  -> DNA helicase is deacetylated
- This investigation correlation between phosphorylation dependent and ubiquitinated cytoplasmic forms of Tax and deacetylated nuclear forms
  -> Tax is sumoylated
- the expression levels of herp, which increases 8 increased those of Bcl-2, Bcl-xL, and phosphorylated Akt significantly
  -> Bcl-2 is phosphorylated
- Moreover, accumulation of p62 and polyubiquitinated proteins has been reported in murine cells deficient in autophagy (Atg-5 or Atg-7 deficient mice)
  -> p62 is ubiquitinlated
USE CASE: MACHINE READING FOR CANCER RESEARCH
Why cancer?

$200$ billion have been invested in cancer research since (R. Barzilay, NAACL 2016)
Why cancer?

Cancer Death Rates* Among Women, US, 1930 – 2005

*Age-adjusted to the 2000 US standard population.
National Center for Health Statistics, Centers for Disease Control and Prevention, 2008.
Why cancer?

Cancer Death Rates* Among Men, US, 1930 – 2009

*Age-adjusted to the 2000 US standard population.
National Center for Health Statistics, Centers for Disease Control and Prevention.
Why the slow progress?

Publications indexed by PubMed each year since 1995
Why the slow progress?

90% are never cited!¹

¹ http://www.smithsonianmag.com/smart-news/half-academic-studies-are-never-read-more-three-people-180950222/?no-ist
“A knotty puzzle may hold a scientist up for a century, when it may be that a colleague has the solution already and is not even aware of the puzzle that it might solve.”

– Isaac Asimov, *The Robots of Dawn*
We need machine reading

• If humans can’t process this much information, then machines must help!
Machine reading for biomedical literature

The phosphorylation of Hdm2 by MK2 promotes the ubiquitination of p53.

Reading  Assembly

Reasoning
Predict, explain, test, curate, etc.
We hypothesize that decreased PTPN13 expression enhances its phosphorylation.
REACH (REading and Assembling Contextual and Holistic mechanisms)

We hypothesize that decreased PTPN13 expression enhances its phosphorylation.
We hypothesize that decreased PTPN13 expression enhances its phosphorylation.
REACH (REading and Assembling Contextual and Holistic mechanisms)
We hypothesize that decreased PTPN13 expression enhances its phosphorylation.
REACH (REading and Assembling Contextual and Holistic mechanisms)

Preprocessing

Entity Extraction

Simple Event Extraction

Nested Event Extraction

Polarity

Hedging

We hypothesize that decreased PTPN13 expression enhances its phosphorylation.
REACH (REading and Assembling Contextual and Holistic mechanisms)

- Preprocessing
- Entity Extraction
- Simple Event Extraction
- Nested Event Extraction
- Polarity
- Coreference

We hypothesize that decreased PTPN13 expression enhances its phosphorylation.

... EphrinB1 ... decreased PTPN13 expression enhances its phosphorylation.
REACH (REading and Assembling Contextual and Holistic mechanisms)

We hypothesize that decreased PTPN13 expression enhances its phosphorylation.

... EphrinB1 ... decreased PTPN13 expression enhances its phosphorylation.
## Few grammar rules

<table>
<thead>
<tr>
<th>Type</th>
<th>Syntax</th>
<th>Surface</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entities</td>
<td>0</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Generic entities</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Modifications</td>
<td>0</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Mutants</td>
<td>0</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td><strong>Total entities</strong></td>
<td>0</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Simple events</td>
<td></td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>Binding</td>
<td></td>
<td></td>
<td>37</td>
</tr>
<tr>
<td>Hydrolysis</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Translocation</td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>Positive regulation</td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>Negative regulation</td>
<td></td>
<td></td>
<td>17</td>
</tr>
<tr>
<td><strong>Total events</strong></td>
<td></td>
<td></td>
<td>122</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td>154</td>
</tr>
</tbody>
</table>

These rules were manually written. Later I will discuss our work on using learn such rules using representation learning.
How well does machine reading work for a complete reading task?

Human domain experts are around here
But is machine reading *actually* useful?
Mutual exclusivity intuition
Mutex insight: If a tumor “wants” to disable a mechanism, it will mutate something upstream, but it generally won’t “pay” for two mutations that do the same thing. So mutually exclusive mutations plus a good model can tell us which mechanisms the tumor disables.

Y is a “kill switch” that prevents proliferation.
How Mutex works

• Mutex does a graph search on the signaling network to find subgraphs of genes that
  – are altered in mutually exclusive manner, and
  – have a common downstream signaling target.

Patient data

Machine reading contributes here!
Breast (BRCA)

Legend for gene alteration frequency:
- post-translational control (protein-level)
- expression control (RNA-level)
- new discovery
Machine reading suggests novel hypotheses that are missed by the authors of the individual publications.
LEARNING INTERPRETABLE MODELS
Motivation for rule learning

• It took us 1 – 2 person months to build the grammar of 150+ rules for the biomedical domain

• In many cases, one does not have this time
  • Scenario 1: But training data exists
  • Scenario 2: And training data does not exist either...
Motivation

• It took us 1 – 2 person months to build the grammar of 154 rules for the biomedical domain

• In many cases, one does not have this time
  • **Scenario 1: But training data exists (2 papers)**
  • **Scenario 2: And training data does not exist either...**
SnapToGrid: From Statistical to Interpretable Models for Biomedical Information Extraction

Marco A. Valenzuela-Escárcega, Gus Hahn-Powell, Dane Bell, Mihai Surdeanu
University of Arizona
Tucson, AZ 85721, USA
{marcov, hahnpowell, dane, msurdeanu}@email.arizona.edu
A three-step process

1. Train a statistical classifier
   a) Aggressive feature selection using regularization

2. Convert model to rules
   a) Convert features to rules
   b) “Snap to grid”: throw away most statistics by discretizing feature weights

3. Model editing
CD2 signaling induces phosphorylation of CREB in primary lymphocytes.
BioNLP 2009 Shared Task

CD2 signaling induces phosphorylation of CREB in primary lymphocytes.

Input
Protein

Output
+Reg
Phosphorylation
Classifier #1: detect event triggers
BioNLP 2009 Shared Task

CD2 signaling induces phosphorylation of CREB in primary lymphocytes.

Classifier #1: detect event triggers
Classifier #2: detect relations between triggers and proteins
Step 1: train statistical classifiers

- We use logistic regression for both classifiers
- Feature selection through regularization
  - L1 regularization: aggressive feature selection; slightly lower performance
Machine learning features

**Triggers**
- **Token**
  - Word, lemma, gazetteer
- **Surface**
  - Token features for a window around token of interest
  - Bigrams
- **Syntax**
  - Dependency paths up to depth 2
  - Token features for token at the end of each path
- **Bag-of-words and entity count**
  - For whole sentence
  - For window surrounding token of interest

**Relations**
- **Path**
  - Shortest dependency path between nodes of interest
  - With and without lexicalization
  - Path length
- **Surface**
  - Words surrounding and between nodes of interest
- **Consistency**
  - Soft constraints on edges between trigger and arguments (e.g., only regulations have causes)
- **N-gram**
  - Dependency, token, dependency
  - Token, dependency, token
Step 2: Convert model features to rules

- Features are just patterns
- We simply rewrote them using Odin syntax

“Passive subject of a phosphorylation trigger that is a protein”
Step 2: Convert model features to rules

- type: dependency
  label: Phosphorylation
  pattern: |
    trigger: Phosphorylation
    theme: Protein = >nsubjpass

Diagram:

```
NN MEK gets auxpass VBN phosphorylated.
```

```
MEK
  \text{gets} \quad \text{phosphorylated.}
```
Step 2: Convert model features to rules

• We know have a decision list classifier. But:
  • Feature weights are unbounded continuous values
    – Useful for resolving conflicts
    – But nearly impossible to understand/modify

• Rules still need to vote
  – We normalize and discretize weights (“votes”) using Scott’s rule (used for the generation of bins in histograms)
Step 2: Convert model features to rules
Step 2: Convert model features to rules

• Binding
  – PROTEIN recruits PROTEIN

• Localization
  – PROTEIN is recruited to the cytoplasm

# vote: +2
- name: Binding_1
  label: Binding
  type: token
  action: countMentions
  pattern: |
    [lemma=recruit & tag=/(V|N|J)/]

# vote: +1
- name: Localization_1
  label: Localization
  type: token
  action: countMentions
  pattern: |
    [lemma=recruit & tag=/(V|N|J)/]
Step 3: Model editing

- Two linguists were given the task of improving the generated rules

- Constraints:
  - Only have access to the model. No peeking at the training data
  - Approximately one hour to work on the task
Expert recommendations

• Generalize syntactic patterns, e.g., participants in events may be heads or modifiers of noun phrases
  – E.g.: “K-Ras” or “the K-Ras protein”
• Eliminate trigger rules that were not sufficiently discriminative
• Make rules robust to common parsing mistakes

• (22 total specific recommendations)
Results on BioNLP 2009

1.19M features
Results on BioNLP 2009

![Bar chart showing results on BioNLP 2009 with two bars: one for 1.19M features and one for 10K features.]
Results on BioNLP 2009

- 1.19M features
- 10K features
- 10K rules
Results on BioNLP 2009

Nearly the same performance. But with an interpretable model!

This is still a large number of rules. We can reduce them by > 1 order of magnitude using automata theory.
Summary

• Converting statistical models into a deterministic decision list classifier does impact performance negatively

• But keeping the human in the loop allows us to recover almost all the lost performance. And we end up with an interpretable model!
Fine. But how do you do this with neural networks that do not have explicit features?
Exploring Interpretability in Event Extraction: Multitask Learning of a Neural Event Classifier and an Explanation Decoder

Zheng Tang *, Gus Hahn-Powell †, Mihai Surdeanu *
* Department of Computer Science
† Department of Linguistics
University of Arizona
{zhengtang, hahnpowell, msurdeanu}@email.arizona.edu
Intuition: jointly training for prediction and interpretability!

1. Use neural classifiers instead of the LRs in the previous paper 😊

2. Decode rules from natural language texts (reusing ideas from machine translation)
   - Source language – original texts
   - “Target language” – grammar rule that matches

3. Train them jointly
Encoder-decoder neural architectures for machine translation

A neural take on the interlingua representation

Je mange du gâteau .

Figure 1: Our model reads an input sentence “ABC” and produces “WXYZ” as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

I eat cake .
Our “source” and “target languages”

(source)

PKCAlpha regulates phosphorylation of NF-kappaB p65 at Ser276.

(target)

Phosphorylation_syntax_1a_noun:

trigger = [lemma=/phosphorylation/ !word=/(?:i)^{de|auto}/]

theme:BioChemicalEntity = prep_of appos? /nn|conj_(and|or|nor)|cc/{,2}
Trigger = [context^D_2]

LSTM Decoder

Rule

Task 2: Rule Generation

Trigger Classification

Entity Attention

BiLSTM Encoder

Token Representation

Data Preparation

word: ... the phosphorylation of PKC by ...

relative position: ... -3 -2 -1 0 1 ...
Data

• For event classification: BioNLP 2013 (similar to the data used in the previous paper)

• For rule decoding: pairs of (sentence, rule) extracted by our machine reading system with manually written rules
  – Some come from the BioNLP dataset and are aligned with the gold annotations in BioNLP 2013
  – Approximately 15K pairs come from other publications ("silver" data)
Results for event classification

We extract 3 types of events

- Rule baseline – system with manually written rules
- T1 – supervised neural event classifier
- T1 + Silver – semi-supervised neural event classifier
- T1 + Silver + T2 - semi-supervised neural event classifier trained jointly with the rule decoder

<table>
<thead>
<tr>
<th></th>
<th>Phosphorylation (P)</th>
<th>Localization (L)</th>
<th>Gene Expression (GE)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>Rule baseline</td>
<td>92.68</td>
<td>48.12</td>
<td>63.35</td>
</tr>
<tr>
<td>T1</td>
<td>87.78</td>
<td>49.38</td>
<td>63.20</td>
</tr>
<tr>
<td>T1 + Silver</td>
<td>62.75</td>
<td>82.50</td>
<td>71.28</td>
</tr>
<tr>
<td>T1 + Silver + T2</td>
<td>84.38</td>
<td>68.75</td>
<td>75.77</td>
</tr>
</tbody>
</table>
Results for event classification

<table>
<thead>
<tr>
<th></th>
<th>Phosphorylation (P)</th>
<th>Localization (L)</th>
<th>Gene Expression (GE)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>Rule baseline</td>
<td>92.68</td>
<td>48.12</td>
<td>63.35</td>
</tr>
<tr>
<td>T1</td>
<td>87.78</td>
<td>49.38</td>
<td>63.20</td>
</tr>
<tr>
<td>T1 + Silver</td>
<td>62.75</td>
<td>82.50</td>
<td>71.28</td>
</tr>
<tr>
<td>T1 + Silver + T2</td>
<td>84.38</td>
<td>68.75</td>
<td>75.77</td>
</tr>
</tbody>
</table>

Semi-supervised learning helps!
## Results for event classification

<table>
<thead>
<tr>
<th></th>
<th>Phosphorylation (P)</th>
<th></th>
<th>Localization (L)</th>
<th></th>
<th>Gene Expression (GE)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>Rule baseline</td>
<td>92.68</td>
<td>48.12</td>
<td>63.35</td>
<td>66.13</td>
<td>44.44</td>
<td>53.16</td>
</tr>
<tr>
<td>T1</td>
<td>87.78</td>
<td>49.38</td>
<td>63.20</td>
<td>100.00</td>
<td>4.04</td>
<td>7.77</td>
</tr>
<tr>
<td>T1 + Silver</td>
<td>62.75</td>
<td>82.50</td>
<td>71.28</td>
<td>54.55</td>
<td>34.34</td>
<td>42.15</td>
</tr>
<tr>
<td>T1 + Silver + T2</td>
<td><strong>84.38</strong></td>
<td><strong>68.75</strong></td>
<td><strong>75.77</strong></td>
<td><strong>76.60</strong></td>
<td><strong>39.39</strong></td>
<td><strong>52.03</strong></td>
</tr>
</tbody>
</table>

Jointly training for prediction and interpretability helps prediction!
## Results for rule decoding

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>Exact Matches</th>
<th>Non-exact Explainable Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>93.80</td>
<td>86.11</td>
<td>2/15</td>
</tr>
<tr>
<td>L</td>
<td>83.78</td>
<td>84.33</td>
<td>1/9</td>
</tr>
<tr>
<td>GE</td>
<td>78.99</td>
<td>76.45</td>
<td>10/43</td>
</tr>
</tbody>
</table>
## Rule error analysis

<table>
<thead>
<tr>
<th>Hand-written Rule</th>
<th>Decoded Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>trigger = [ lemma = /phosphorylate/ &amp; ! word = /(?!)(de</td>
<td>auto)/ &amp; tag = /^VJJ)/ &amp; !</td>
</tr>
<tr>
<td>mention = ModificationTrigger ]</td>
<td>ModificationTrigger ]</td>
</tr>
<tr>
<td>theme : BioChemicalEntity = &gt; nsubjpass prep_of ? /conj_(and</td>
<td>or</td>
</tr>
<tr>
<td>trigger = [ lemma = /detect</td>
<td>localiz</td>
</tr>
<tr>
<td>outgoing = /prep_(by</td>
<td>of)/</td>
</tr>
<tr>
<td>trigger = [ lemma = /phosphorylation/ &amp; ! word = /(?!)( de</td>
<td>auto )/ &amp; !</td>
</tr>
<tr>
<td>outgoing = /prep_(by</td>
<td>of)/ ]</td>
</tr>
<tr>
<td>theme : BioChemicalEntity = &lt; /conj_ ( and</td>
<td>or</td>
</tr>
<tr>
<td>theme : BioChemicalEntity = &lt; /conj_ ( and</td>
<td>or</td>
</tr>
</tbody>
</table>

**Green** – missed text in the decoded rule  
**Red** – hallucinated text
Summary

• A neural approach that jointly trains for prediction and interpretability
• The joint training improves prediction!
Motivation

• It took us 1 – 2 person months to build the grammar of 154 rules for the biomedical domain

• In many cases, one does not have this time
  • Scenario 1: But training data exists
  • Scenario 2: And training data does not exist either...
Lightly-supervised Representation Learning with Global Interpretability

Andrew Zupon, Maria Alexeeva, Marco A. Valenzuela-Escárcega, Ajay Nagesh, and Mihai Surdeanu
University of Arizona, Tucson, AZ, USA
{zupon, alexeeva, marcov, ajaynagesh, msurdeanu}@email.arizona.edu
Architecture and walkthrough example

Preprocessing

Entity Extraction

Simple Event Extraction

Nested Event Extraction

Polarity

Hedging

Coreference

Shown for this task. But a similar algorithm works for event extraction.
Traditional rule bootstrapping algorithm for entity classification

Problem: brittle statistics, estimated only with respect to known information!

Seeds

LOC: San Francisco
LOC: city of ___
LOC: Los Angeles
LOC: mayor of ___

✓

city of San Francisco

✓

of ___

of ___
Learning to read

We want to combine:

• The advantages of representation learning, aka “word embeddings”
  – Neural network language models handle unsupervised data well

• The interpretability of our current approach
  – Produce patterns in the end

• Keep the human in the loop, but minimally
  – Just a few examples
Starting point: distributional hypothesis

• Distributional hypothesis:
  – By looking at a word’s context, one can infer its meaning (Harris, 1954)
  – You shall know a word by the company it keeps (Firth, 1957)
Example

- tasty $X$
- $X$ with butter
- $X$ and coffee
- greasy $X$
Example

- tasty $X$
- $X$ with butter
- $X$ and coffee
- greasy $X$
Starting point: word2vec, skip-gram

You shall know a word by the company it keeps.
- Firth, 1957

You shall know the company by the word it keeps.
- Word2vec, skip-gram
Starting point: word2vec, skip-gram

The city of Tucson is the place to vacation!

\[
P(\text{city} | \text{Tucson}) \quad ++
\]
\[
P(\text{place} | \text{Tucson}) \quad ++
\]

...  

\[
P(\text{dog} | \text{Tucson}) \quad --
\]

...
Two important changes

• We will learn embeddings for *both named entities and patterns*
  – An entity’s context is defined by the patterns that match it

• Added supervision in the objective function, to incorporate human-provided information
  – We have “seed” names in each category
The city of Tucson is the place to vacation!

P(city of __ | Tucson) ++
P(__ is the place) ++
...
P(the __ barks | Tucson) --
...

We can export the patterns closest to a category to be used in our grammars!

Just n-grams...
Objective function

Unsupervised, directly “inherited” from skip-gram

Light supervision from a few seed examples, iteratively expanded
EmBoot

SGD on the previous objective function

Entities

Seeds
LOC: San Francisco
LOC: Los Angeles

embedding vectors for entities and patterns

Promote entities closest to the seeds
Visualization of the training procedure

• On the CoNLL-2003 dataset
  – PER = purple
  – LOC = blue
  – ORG = green
  – MISC = red

• Human contribution: seed set with 10 entities in each category
  – PER: Clinton, Dole, ...
  – LOC: U.S., Germany, ...
  – ORG: Reuters, U.N., ...
  – MISC: Russian, German, ...
Mostly performers

Mostly politicians

Participants:
- PER
- MISC
- LOC
- ORG

Keywords:
- Most popular terms related to performers, politicians, locations, and organizations.
Results, CoNLL dataset (4 classes)
Results, OntoNotes (11 classes)

OVERALL : Ontonotes

- EPB
- EPB<int
- Emboot
- Emboot<int
- LP
Summary

• A lightly-supervised approach that jointly learns representations for entities and patterns that extract them

• State-of-the-art results for semi-supervised learning

• The rules can be edited by domain experts, and this leads to further improvements in performance (not shown)
Take-home message

• For large, inter-disciplinary projects we need to move beyond “black-box” methods to approaches that produce globally interpretable models

• We can produce such interpretable models using deep learning (best of both worlds?)


Many thanks to my collaborators!
THANK YOU!
QUESTIONS?
Conflict of interest disclosure

• M. Surdeanu discloses a financial interest in Lum.ai. This interest has been disclosed to the University of Arizona Institutional Review Committee and is being managed in accordance with its conflict of interest policies.