POTATO: exPlainable infOrmation exTrAcTion framewOrk

Ádám Kovács

TU Wien
adam.kovacs@tuwien.ac.at

TUW NLP seminar
1/18/2022
XAI - interpretability, explainability

- We should be able to explain the decisions of machine learning systems.
- Explainable systems have the following traits (Doshi-Velez and Kim, 2017):
  - **Fairness** - unbiased predictions
  - **Privacy** - no information leakage
  - **Reliability** - small changes in the input do not affect heavily the output
  - **Trust, Auditability** - we can trust XAI systems better than black-box models
There are interpretable machine learning systems e.g. Logistic Regression, Decision trees, Naive bayes, etc..

- feature importance can directly correlate with the decisions

State-of-the-art models are usually complex Deep Learning architectures with billions of parameters

- GPT3 has 175B parameters (Brown et al., 2020)
- BERT-large has 340M parameters (Devlin et al., 2019)
Interpreting ML models

- There are ways to explain complex ML models
- Model-agnostic methods → can work with any ML model
  - example based explanations → provide examples for decisions
  - global model-agnostic methods → explain the behaviour of the model (Apley and Zhu, 2020)
  - local model-agnostic methods → explain individual predictions (LIME, (Ribeiro et al., 2016), SHAP (Lundberg and Lee, 2017))

- Model-specific methods
  - use attention as explanation (Fukui et al., 2019; Wang et al., 2016; Lee et al., 2017; Ghaeini et al., 2018)
LIME (Ribeiro et al., 2016)

Prediction probabilities

- **Atheism**: 0.58
- **Christian**: 0.42

**Text with highlighted words**

From: johnchad@triton.unm.edu (jchadwic)
Subject: Another request for Darwin Fish
Organization: University of New Mexico, Albuquerque
Lines: 11

**NNTP-Posting-Host**: triton.unm.edu

Hello Gang,

There have been some notes recently asking where to obtain the DARWIN fish. This is the same question I have and I have not seen an answer on the net. If anyone has a contact please post on the net or email me.
SHAP (Lundberg and Lee, 2017)

4.98 = LSTAT
6.575 = RM
0.538 = NOX
1 = RAD
296 = TAX
0.006 = CRIM
4.09 = DIS
15.3 = PTRATIO
65.2 = AGE
4 other features

\[ f(x) = 24.019 \]

\[ E[f(X)] = 22.533 \]
Attention as explanation

- We can look at the local weights for each prediction.
- The weights can serve as an explanation for that specific decision.

(a) the aspect of this sentence: service

(b) the aspect of this sentence: food
DL models

▶ limited explainability
  (Serrano and Smith, 2019; Wiegreffe and Pinter, 2019; Jain and Wallace, 2019; Pruthi et al., 2020)

▶ prone to bias
  (De-Arteaga et al., 2019; Kurita et al., 2019; Bender et al., 2021)

▶ prone to solving datasets rather than solving problems \(\sim\) artefacts
  (Glockner et al., 2018; Gururangan et al., 2018; McCoy et al., 2019; Rychalska et al., 2018; Chen et al., 2016; Jia and Liang, 2017)
Rule-based systems

Pros
▶ Rule-based systems are interpretable and explainable by design
▶ Are popular in “real-world” applications
▶ Fully-customizable and can be debugged

Cons
▶ Hard to maintain
▶ Worse performance on benchmarks
▶ Domain expertise is needed
▶ Time-consuming to maintain and to develop

Combine ML and rule-systems: Learn rules!
Relation extraction

- We will use an example from the Semeval 2010 relation extraction dataset (Hendrickx et al., 2010)
- Relation extraction (RE) is the task of extracting semantic relationship between entities from a text
- Usually between two or more entities
- Semantic categories (e.g. Destination, Component, Employed by, Founded by, etc..)
- Example for the **Entity-Destination** label:
  - The diamond ring was dropped into a trick-or-treater’s bag.
Rules

The diamond `<entity1>ring</entity1> was dropped into a trick-or-treater’s `<entity2>bag</entity2>.

- A rule can be a simple regex
  
  \n  r" entity1 .* dropped into .* entity2"
  
- More advanced like spaCy’s `TokenMatcher` or the Holmes Extractor

  pattern = [
    { 'POS': 'VERB' },
    { 'LOWER': 'into' },
    { 'TEXT': { 'REGEX': '.*' } },
    { 'LOWER': 'entity2' }
  ]
Syntactic, Semantic graphs

After graduation John moved to Paris

Universal dependency graph (UD)

4lang Kornai (2019)

AMR Banarescu et al. (2013)
Graph rules

- Rules on graphs could utilize the underlying graph structure of texts
- SpaCy’s **DependencyMatcher** module
  - Can be used to match rules on dependency trees.
  - But only works on UD structures
  - Complex structure
- Our own solution in [https://github.com/recski/tuw-nlp](https://github.com/recski/tuw-nlp)\(^1\)
  - Works with networkx
  - Can be used with arbitrary graph structures
  - Currently works with AMR ([Banarescu et al., 2013](https://pypi.org/project/tuw-nlp/)), 4lang ([Kornai, 2019](https://pypi.org/project/tuw-nlp/)), and Stanza ([Qi et al., 2020](https://pypi.org/project/tuw-nlp/))
Input: *The diamond* `<entity1>`*ring*<entity1>*was dropped into a trick-or-treater’s* `<entity2>`*bag*<entity2>.

```python
pattern = [
    {'RIGHT_ID': 'anchor_verb',
     'RIGHT_ATTRS': {'TEXT': {'REGEX': '.*'}}}
],

{'LEFT_ID': 'anchor_verb',
 'REL_OP': '>',
 'RIGHT_ID': 'entity2',
 'RIGHT_ATTRS': {'LOWER': 'entity2', 'DEP': 'nmod'}
},

{'LEFT_ID': 'entity2',
 'REL_OP': '>',
 'RIGHT_ID': 'into',
 'RIGHT_ATTRS': {'LOWER': 'into', 'DEP': 'case'}
},

{'LEFT_ID': 'anchor_verb',
 'REL_OP': '>',
 'RIGHT_ID': 'diamond',
 'RIGHT_ATTRS': {'LEMMA': 'diamond'}
},

{'LEFT_ID': 'diamond',
 'REL_OP': '>',
 'RIGHT_ID': 'entity1',
 'RIGHT_ATTRS': {'LOWER': 'entity1'}
}
]
Input: *The diamond* `<entity1>`*ring*`<entity1>`*was dropped into a trick-or-treater’s* `<entity2>`*bag*`<entity2>`.

Rule in penman format:

\[
(u_15 / into :2 (u_2 / entity2) \\
:1 (u_3 / .* :2 (u_4 / entity1)))
\]

Retrieved examples:

- The man placed the entity1 into the entity2.
- Industries have pushed entity1 into fragile marine entity2.
- I am putting the entity1 into a MySQL entity2.
- The entity1 were released into the entity2.
Patterns with AMR in our system

Rule:

Input: *The Chinese virus kills everyone*
POTATO is a human-in-the-loop XAI framework

We provide

- a unified `networkx` interface for multiple graph libraries (4lang, stanza, AMR)
- a python package for learning and evaluating interpretable graph features as rules
- a human-in-the-loop (HITL) UI framework built in streamlit
- a REST-API to use extracted features for inference in production mode

---

Collaborators
All of our components are open-source under MIT license and can be installed with pip.

Library to build and use graphs:
https://github.com/recski/tuw-nlp

xpotato: https://github.com/adaamko/potato

---

3 pip install tuw-nlp
4 pip install xpotato
Human-in-the-loop learning (HITL) of rules

- Idea → use subgraphs as features for training simple classifiers (LogReg, Random Forest, etc.)
- Generate subgraphs only up to a certain edge number (to avoid large number of features)
- Suggest rules to users based on feature importance
- User can accept, reject, edit, combine patterns
- Subgraphs may have regexes as node or edge labels
- Underspecified subgraphs can be refined
Workflow

- Dataset -> Generate graphs and subgraphs -> POTATO backend
- POTATO backend -> Load graphs -> Manual rule modification
- Manual rule modification -> HITL supervised learning
- HITL supervised learning -> Get automatic rule suggestions
- Get automatic rule suggestions -> Evaluation
- Evaluation -> Is it good enough?
- Is it good enough? -> NO: Annotated Data? -> Advanced mode
- YES: Annotated Data? -> POTATO UI
- POTATO UI -> YES: Inference mode
- NO: Annotated Data? -> Advanced mode
- Advanced mode -> YES: Inference mode
- Inference mode -> Human expert
POTATO UI

Rule chooser and modifier

First, choose class you want to use to build rules

Entity-Destination(e1,e2)

You can modify any rule you want to

Remember, we use the PENMAN notation to describe a rule. You can find more information about the rules in the README of our repository.

<table>
<thead>
<tr>
<th>rules</th>
<th>negated_rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>(u_3 / to :2 (u_2 / entity2))</td>
<td></td>
</tr>
<tr>
<td>(u_15 / into :2 (u_2 / entity2))</td>
<td></td>
</tr>
<tr>
<td>(u_264 / place :2 (u_25 / entity1))</td>
<td></td>
</tr>
<tr>
<td>(u_24 / in :2 (u_2 / entity2))</td>
<td></td>
</tr>
<tr>
<td>(u_1200 / give :2 (u_2 / entity1))</td>
<td></td>
</tr>
<tr>
<td>(u_414 / put :2 (u_25 / entity1))</td>
<td></td>
</tr>
<tr>
<td>(u_3 / to :2 (u_2 / entity2) :1 (u_694 / send))</td>
<td></td>
</tr>
<tr>
<td>(u_966 / add :2 (u_25 / entity1))</td>
<td></td>
</tr>
<tr>
<td>(u_4 / COORD :2 (u_25 / entity1) :0 (u_414 / put))</td>
<td></td>
</tr>
<tr>
<td>(u_3 / to :1 (u_2628 / donate))</td>
<td></td>
</tr>
<tr>
<td>(u_3 / to :1 (u_1200 / give))</td>
<td></td>
</tr>
<tr>
<td>(u_35 / into :2 (u_2 / entity2) :1 (u_3 / :2 (u_4 / entity1)))</td>
<td></td>
</tr>
</tbody>
</table>

After you modified any rule, click on save updates button to save your changes.

Graph viewer and evaluator

Choose from the rules

(u_15 / into :2 (u_2 / entity2))

Result of using all the rules: Precision: 0.762, Recall: 0.628, Fscore: 0.689

The rule’s result: Precision: 0.762, Recall: 0.628, Fscore: 0.689, True positives: 407, False positives: 127

Show validation data

Select the graphs you want to view

True Positive graphs
### Inspect rules

Tick to box next to the rules you want to accept, then click on the `accept_rules` button.

Unaccepted rules will be deleted.

<table>
<thead>
<tr>
<th>feature</th>
<th>precision</th>
<th>recall</th>
<th>fscore</th>
<th>TP</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>(u_2628 / donate)</td>
<td>0.857</td>
<td>0.019</td>
<td>0.036</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>(u_103 / pour)</td>
<td>0.848</td>
<td>0.060</td>
<td>0.112</td>
<td>39</td>
<td>7</td>
</tr>
<tr>
<td>(u_264 / place,:2 (u_25 / entity1))</td>
<td>0.792</td>
<td>0.059</td>
<td>0.109</td>
<td>38</td>
<td>10</td>
</tr>
<tr>
<td>(u_1412 / spread)</td>
<td>0.583</td>
<td>0.022</td>
<td>0.042</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>(u_1200 / give,:2 (u_25 / entity1))</td>
<td>0.533</td>
<td>0.022</td>
<td>0.042</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>(u_414 / put)</td>
<td>0.486</td>
<td>0.082</td>
<td>0.140</td>
<td>53</td>
<td>56</td>
</tr>
<tr>
<td>(u_2109 / export)</td>
<td>0.474</td>
<td>0.014</td>
<td>0.027</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>(u_264 / place)</td>
<td>0.418</td>
<td>0.079</td>
<td>0.132</td>
<td>51</td>
<td>71</td>
</tr>
<tr>
<td>(u_3 / to,:1 (u_1200 / give))</td>
<td>0.381</td>
<td>0.024</td>
<td>0.042</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>(u_14 / in,:2 (u_2 / entity2))</td>
<td>0.118</td>
<td>0.088</td>
<td>0.101</td>
<td>57</td>
<td>428</td>
</tr>
</tbody>
</table>
Select the graphs you want to view

True Positive graphs

Tick the box next to the graphs you want to see. The rule that applied will be highlighted in the graph.

The penman format of the graph will be also shown, you can copy any of the part directly from the penman format if you want to add a new rule.

<table>
<thead>
<tr>
<th>id</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>entity1 in the text associated concepts was brought into the working entity2 in an attempt to resolve the violation.</td>
</tr>
<tr>
<td>30</td>
<td>Finally, we injected entity1 into the entity2.</td>
</tr>
<tr>
<td>66</td>
<td>Then after the concert, he stuffed the entity1 into a entity2 under his bed where they remained for 40 years.</td>
</tr>
<tr>
<td>133</td>
<td>The manager has added background text entity1 into the existing PDF entity2.</td>
</tr>
<tr>
<td>166</td>
<td>He accidentally dropped the entity1 into the wrong entity2.</td>
</tr>
<tr>
<td>212</td>
<td>An American entity1 fell drunkenly into the city's Main entity2.</td>
</tr>
<tr>
<td>242</td>
<td>The man placed the entity1 into the entity2.</td>
</tr>
<tr>
<td>253</td>
<td>Industries have pushed entity1 into fragile marine entity2.</td>
</tr>
<tr>
<td>264</td>
<td>I am putting the entity1 into a MySQL entity2.</td>
</tr>
<tr>
<td>296</td>
<td>The entity1 arrived into this entity2 with gifts and talents.</td>
</tr>
<tr>
<td>297</td>
<td>We removed the sharp entity1 into the entity2.</td>
</tr>
<tr>
<td>312</td>
<td>New entity1 are manually added into phone entity2.</td>
</tr>
</tbody>
</table>

Sentence: Finally, we injected entity1 into the entity2.

Sentence ID: 30

Gold label: Entity-Destination(e1,e2)

TP: 483
POTATO advanced mode

- Our framework can be used with limited data
- Annotate some data
- Get suggestions from our simple ML model
- Define, modify the rules
- Annotate the data with the rules
- Iterate recursively
POTATO advanced mode

Annotation/Dataset browser:

Annotate samples here:

Currently the following rules are applied:

```
0 : "(\s_1 / shame)"
```

<table>
<thead>
<tr>
<th>index</th>
<th>text</th>
<th>label</th>
<th>applied_rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>Look the seriousness of BJP... Nation is dying of covid and they going to dharna in entire nation... Shame on BJP RaviMurariMoosooi (URL)</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Shame on you RJD</td>
<td>Speak4ShahabuddinTejvai</td>
<td>Shahabuddin</td>
</tr>
<tr>
<td>36</td>
<td>ModifailedIndia</td>
<td>IndiaCovideCrisis</td>
<td>heartbreaking report</td>
</tr>
<tr>
<td>38</td>
<td>[USER] How can you sugarcoat this? No courage to actually let your readers know why she is suspended! The only thing she was ever vocal was about Islamophobia &amp; is preaching Hinduva hate! Shame on Filmfare that cant speak the truth in spite of all the hate, death &amp; carnage in India</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>It is unconscionable that Australia is stonewalling the TRIPS waiver. To add insult to injury, it has banned its own citizens from repatriation mid escalating crisis. For shame</td>
<td>OFF</td>
<td></td>
</tr>
</tbody>
</table>

Annotate

Samples you have already annotated:

<table>
<thead>
<tr>
<th>index</th>
<th>text</th>
<th>label</th>
<th>applied_rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>82</td>
<td>nah, do not FUCKING piss me off [URL]</td>
<td>OFF</td>
<td></td>
</tr>
</tbody>
</table>
Results and use-cases
### HASOC 2020 - English

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rules</td>
<td>95.3</td>
<td>74.6</td>
<td>83.7</td>
</tr>
<tr>
<td>BERT</td>
<td>90.2</td>
<td>90.5</td>
<td>90.3</td>
</tr>
</tbody>
</table>

### HASOC 2020 - German

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rules</td>
<td>92.4</td>
<td>28.3</td>
<td>43.4</td>
</tr>
<tr>
<td>BERT</td>
<td>66.6</td>
<td>81.7</td>
<td>73.4</td>
</tr>
</tbody>
</table>
Rule extraction from textual building regulations of the City of Vienna
Presented previously by Eszter Iklódi on this seminar.

<table>
<thead>
<tr>
<th>Planzeichen</th>
<th>Precision%</th>
<th>Recall%</th>
<th>F1%</th>
<th>BERT Precision%</th>
<th>Recall%</th>
<th>F1%</th>
<th>RULES Precision%</th>
<th>Recall%</th>
<th>F1%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planzeichen</td>
<td>83</td>
<td>90</td>
<td>86</td>
<td>96</td>
<td>85</td>
<td>90</td>
<td>85</td>
<td>90</td>
<td>89</td>
</tr>
<tr>
<td>Dachart</td>
<td>88</td>
<td>84</td>
<td>86</td>
<td>95</td>
<td>84</td>
<td>89</td>
<td>91</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>BegruenungDach</td>
<td>90</td>
<td>78</td>
<td>84</td>
<td>87</td>
<td>91</td>
<td>89</td>
<td>91</td>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>AnFluchtlinie</td>
<td>81</td>
<td>71</td>
<td>76</td>
<td>89</td>
<td>70</td>
<td>79</td>
<td>70</td>
<td>79</td>
<td>79</td>
</tr>
<tr>
<td>VorkehrungBepflanzung</td>
<td>100</td>
<td>95</td>
<td>98</td>
<td>100</td>
<td>90</td>
<td>95</td>
<td>90</td>
<td>95</td>
<td>95</td>
</tr>
<tr>
<td>GebaeudeBautyp</td>
<td>100</td>
<td>52</td>
<td>69</td>
<td>100</td>
<td>66</td>
<td>80</td>
<td>66</td>
<td>80</td>
<td>80</td>
</tr>
</tbody>
</table>
Medical Relation extraction

On the CrowdTruth data ([Dumitrache et al., 2017][5]

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rules</td>
<td>91.3</td>
<td>32.3</td>
<td>47.7</td>
</tr>
<tr>
<td>BERT</td>
<td>64.7</td>
<td>81.4</td>
<td>70.4</td>
</tr>
</tbody>
</table>

[5]: github.com/CrowdTruth/Medical-Relation-Extraction
## Tone analysis for chatbots

Sparse data, no labels $\rightarrow$ bootstrapping of rules and annotation

<table>
<thead>
<tr>
<th>text</th>
<th>label</th>
<th>applied_rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>warum werden mir 3. $\rightarrow$ vom Konto abgezogen??? Stol</td>
<td>OFF</td>
<td>[]</td>
</tr>
<tr>
<td>Das ist mir keine Hilfe!</td>
<td>OFF</td>
<td>[&quot;(u1 / hilf.* :nmod (u_37 / kein.*))&quot;]</td>
</tr>
<tr>
<td><em>Firstname</em> du bist unnütz!</td>
<td>OFF</td>
<td>[&quot;(u1 / unnuetz.*)&quot;]</td>
</tr>
<tr>
<td>ich huss $ jetzt. nimmt der passwort nicht mehr</td>
<td>OFF</td>
<td>[&quot;(u1 / hass.*)&quot;]</td>
</tr>
<tr>
<td>danke, verarschen kann ich mich selber</td>
<td>OFF</td>
<td>[&quot;(u1 / <em>arsch.</em>)&quot;]</td>
</tr>
<tr>
<td>Ich bin sehr unzufrieden mit Eure Kontaktmöglichkeiten.</td>
<td>OFF</td>
<td>[&quot;(u1 / unzufrieden)&quot;]</td>
</tr>
<tr>
<td>Mir wurde versprochen das man um mein Anlegen sich k</td>
<td>OFF</td>
<td>[]</td>
</tr>
<tr>
<td>du bist keine hilfe</td>
<td>OFF</td>
<td>[&quot;(u1 / hilf.* :nmod (u_37 / kein.*))&quot;]</td>
</tr>
</tbody>
</table>
Thank you!


