Introduction

Legal document analysis is a rich field (classification algorithms, analysis of language and formatting use etc.).

However practical tools available to legal professionals are limited, although computational intelligence has been harnessed efficiently for other professionals (e.g. AutoCad for architects and engineers).

Many possible formal representations of legal documents (e.g. deontic logic) that allow one to reason about them, but missing techniques to go from text to formal logic automatically.

We detail a proof-of-concept tool that leverages NLP tools for the formal analysis of contracts, while also providing other useful features.
Tool Architecture

- Contract
- Tool
- Contract-Support Algorithms
  - Identify Parties
  - Expand Queries
  - Formalize Contract
- NLP techniques and Databases
  - Named Entity Recognizer
  - WordNet
  - Syntactic Parser
User Interface

- Integrated in Word
- Every feature presented as a task pane, or through a context menu.

Features:
- Legal Document Cross-referencing
- National Company Database Search
- Related Word Search
- Conflict Analysis Search
5. All Confidential Information, without limitation, shall remain the personal and proprietary property of Discloser. Recipient shall not acquire any licence or other intellectual property interest in any Confidential Information disclosed to it by Discloser. Further disclosure of Confidential Information shall not result in any obligation to grant Recipient any right in or to said Confidential Information.

6. At any time upon request by Discloser, Recipient shall immediately return to Discloser all Confidential Information it has received from Discloser, including but not limited to all notes and reports which may have been made regarding said Confidential Information, and all copies thereof. Recipient may retain one copy of each item of Confidential Information received by it hereunder and notes regarding the same, provided that said copy shall be retained and used solely for compliance purposes and shall be held in Recipient’s confidential and secured files under the same standards.
5. All Confidential Information disclosed to the Recipient by Discloser shall remain the personal and proprietary property of Discloser. The Recipient undertakes not to disclose any licence or other intellectual property interest in any Confidential Information disclosed to it by Discloser.

6. At any time upon request by Discloser, the Recipient will return to Discloser all Confidential Information, all notes and reports made by Recipient, and all other information or documents concerning Confidential Information, and also all copies of each item of Confidential Information.
These features are supported through a number of off-the-shelf NLP packages:

- Syntax tree parser (Stanford parser)
- Lexical database (WordNet)
- Keyword Extraction
- Named Entity Recognizer
- Legal Documents & Company Database

Dependency injection is used to allow for these to be easily updated by the user with any other implementation, without the need to edit any source code.
The logic behind each feature is also encapsulated in separate modules, with the same benefits.

- Contract Structure/Clause Recognizer
- Query Expansion
- Related Laws and Company Search
- Formal Analysis Module
Provided the passenger satisfies any local airport regulations and checks, when at the boarding gate, the passenger should board the plane. If this is violated, they should return to the check-in desk.

If the passenger is at the boarding gate, the passenger should not be carrying any weapons, otherwise the passenger should leave the airport.
Formal Analysis - Representation

- Two or more parties involved.
- List of definitions and clauses.
- Clauses can be obligations, permissions or prohibitions (i.e. norms), e.g. renter is obliged to pay rent.
- Clause can be structured in a certain way, that places certain conditions on their activation (e.g. renter is obliged to pay rent, on the first day of the month).
Given an action $a$ and a party $p$:

- Obligations $\rightarrow O_p(a)$
- Permissions $\rightarrow P_p(a)$
- Prohibitions $\rightarrow F_p(a)$
Given an action $a$ and a party $p$:

- **Obligations** → $O_p(a)$
- **Permissions** → $P_p(a)$
- **Prohibitions** → $F_p(a)$

- **Conditionals** → $[a]C$
- **Sequential Composition** → $C \triangleright C'$ [If $C$ is satisfied then $C'$ is active]
- **Reparation** → $C \triangleright C'$ [If $C$ is violated then $C'$ is active]
Formal Analysis - Representation

Given an action $a$ and a party $p$:

- **Obligations** → $O_p(a)$
- **Permissions** → $P_p(a)$
- **Prohibitions** → $F_p(a)$

- **Conditionals** → $[a]C$
- **Sequential Composition** → $C \triangleright C'$ [If $C$ is satisfied then $C'$ is active]
- **Reparation** → $C \triangleright C'$ [If $C$ is violated then $C'$ is active]

- **Satisfied Contract** → $\top$
- **Violated Contract** → $\bot$
Formal Analysis - ‘Translation’ Algorithm

- Entity recognition -> Party names
- Constituency parser -> Structure of sentence
- Tree patterns -> Formal clauses
The passenger should not be carrying any weapons.
The passenger should not be carrying any weapons.
Formal Analysis - ‘Translation’ Example

$F_{passenger}(carryWeapons)$
Provided the passenger satisfies any local airport regulations and checks, when at the boarding gate, the passenger should board the plane. If this is violated, they should return to the check-in desk.

If the passenger is at the boarding gate, the passenger should not be carrying any weapons, otherwise the passenger should leave the airport.
Two contracts are said to be in conflict if there is no trace that satisfies both at the same time. The conflict relation is denoted by $\bot$, so that that $C$ and $C'$ are conflicting is denoted by $C \bot C'$. Note also that we denote two mutually exclusive actions as $a \triangleleft a'$.

Axioms:

\[ \vdash P_p(a) \bot F_p(a) \quad (1) \]
\[ \vdash O_p(a) \bot F_p(a) \quad (2) \]
\[ a \triangleleft a' \vdash O_p(a) \bot O_p(a') \quad (3) \]
\[ a \triangleleft a' \vdash O_p(a) \bot P_p(a') \quad (4) \]
\[ C \bot C' \vdash C' \bot C \quad (5) \]
\[ C \bot C' \land C' \equiv C'' \vdash C \bot C'' \quad (6) \]
Provided the passenger satisfies any local airport regulations and checks, when at the boarding gate, the passenger should board the plane. If this is violated, they should return to the check-in desk.

\((\text{atGate}) O_p(satisfies) \triangleright O_p(boardPlane)) \triangleright O_p(desk)\)

If the passenger is at the boarding gate, the passenger should not be carrying any weapons, otherwise the passenger should leave the airport.

\((\text{atGate}) O_p(noWeapons)) \triangleright O_p(leave)\)
## Evaluation

<table>
<thead>
<tr>
<th>Contract</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$</th>
<th>$F_{0.5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C14</td>
<td>14</td>
<td>33</td>
<td>5</td>
<td>2</td>
<td>0.739</td>
<td>0.875</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C21</td>
<td>9</td>
<td>170</td>
<td>56</td>
<td>0</td>
<td>0.139</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C41</td>
<td>16</td>
<td>61</td>
<td>9</td>
<td>0</td>
<td>0.64</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C69</td>
<td>12</td>
<td>37</td>
<td>4</td>
<td>0</td>
<td>0.75</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C199</td>
<td>5</td>
<td>37</td>
<td>18</td>
<td>10</td>
<td>0.217</td>
<td>0.333</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.497</td>
<td>0.842</td>
<td>0.625</td>
<td>0.541</td>
</tr>
</tbody>
</table>

**Table:** Formalizing norms evaluation.
Future Work

- Formalisation can be augmented using dependency parsing, and more types of analyses can be employed (e.g. detecting useless clauses).
- Results can also be made precise through detecting definitions in document.
- More useful features can be added, e.g. template management, versioning system (WIP) etc.
Conclusions

- Law is an area of the humanities where NLP tools have yet to make a noticeable impact.
- We implemented a contract-drafting aid tool as a Word add-in, integrating both natural language and contract-focused analyses within it.
- We show how syntactic parsing can be employed to (naively) translate a contract text into a deontic logic model.
- Tool was tested with lawyers and notaries, with positive feedback on legal documents and company cross-referencing.