Article
Benjamin De Boe · Jun 21, 2016 7m read

Open Exchange

iKnow demo apps (part 4) - Rules Builder Demo

This is the fourth article in a series on iKnow demo applications, showcasing how the concepts and context provided through iKnow's unique bottom-up approach can be used to implement relevant use cases and help users be more productive in their daily tasks. Previous articles discussed the Knowledge Portal, the Set Analysis Demo and the Dictionary Builder Demo, each of which gradually implemented slightly more advanced interactions with what iKnow gleans from unstructured data.

This week, we'll look into one more demo application, the **Rules Builder Demo**, in which we'll build on previous work but again climb a step on the level ladder, implementing a more high-level use case than in the previous ones. The idea came from an opportunity where we were asked to help the customer in the finance sector make sense of vast volumes of contract data. They wanted to semi-automate the extraction of logical rules from that text (in fluent legalese!), so they could be fed into other systems. While this was an exciting use case to work on (and more on it in this GS2016 presentation), we've also used it in other cases, for example to extract mentions of ejection fraction from Electronic Health Records.

Installation & Setup

This demo is available for download from a GitHub repository, where you can either check it out using your preferred Git client, or download it as a zip outright. You should be able to import and compile all classes in any release of Caché starting with 2016.1.

Like the Set Analysis demo, the application consists of a single (CSP) page with AngularJS-infused HTML and associated JavaScript code that calls a REST interface. To make the application work, you'll need to set up a web application that redirects to the Demo.RulesBuilder.RestHandler class, either manually or through theDemo.RulesBuilder.Utils:CreateRestWebApp() method included in the package.

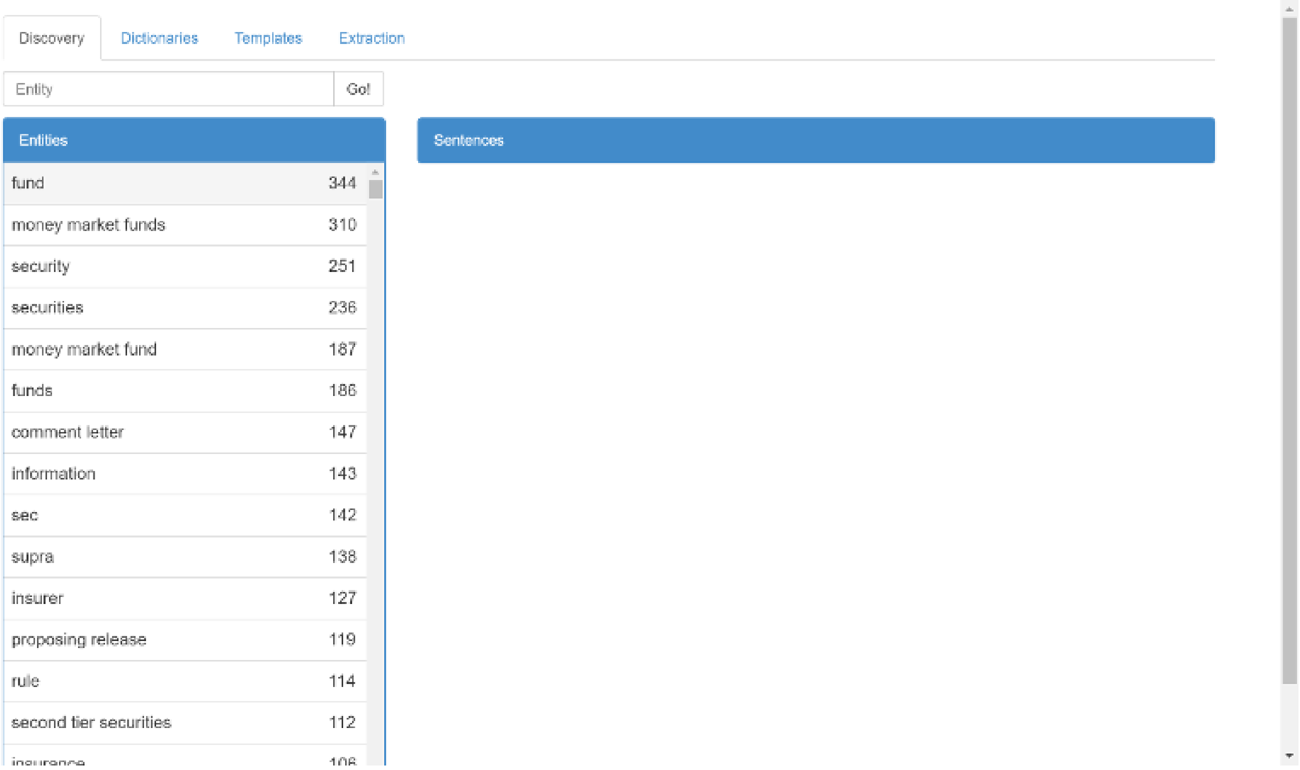
This demo comes with a sample domain that has no data by itself, but a few dictionaries with relevant terminology from the financial domain (to the untrained author!). However, it should work with any domain, as long it is editable.

Domains managed by domain definition classes by default can only be updated through that class and will refuse attempts made by the demo GUI. You can enable editing the domain by setting the top-level allowCustomUpdates attribute to true in the XML. Separately, the demo also expects a metadata field named "DocumentName".

The GUI itself can be accessed through the following URL, using the domain ID as the sole URL request parameter (here: 1) http://localhost:57772/csp/rulesbuilder/RulesBuilderDemo.csp?1.

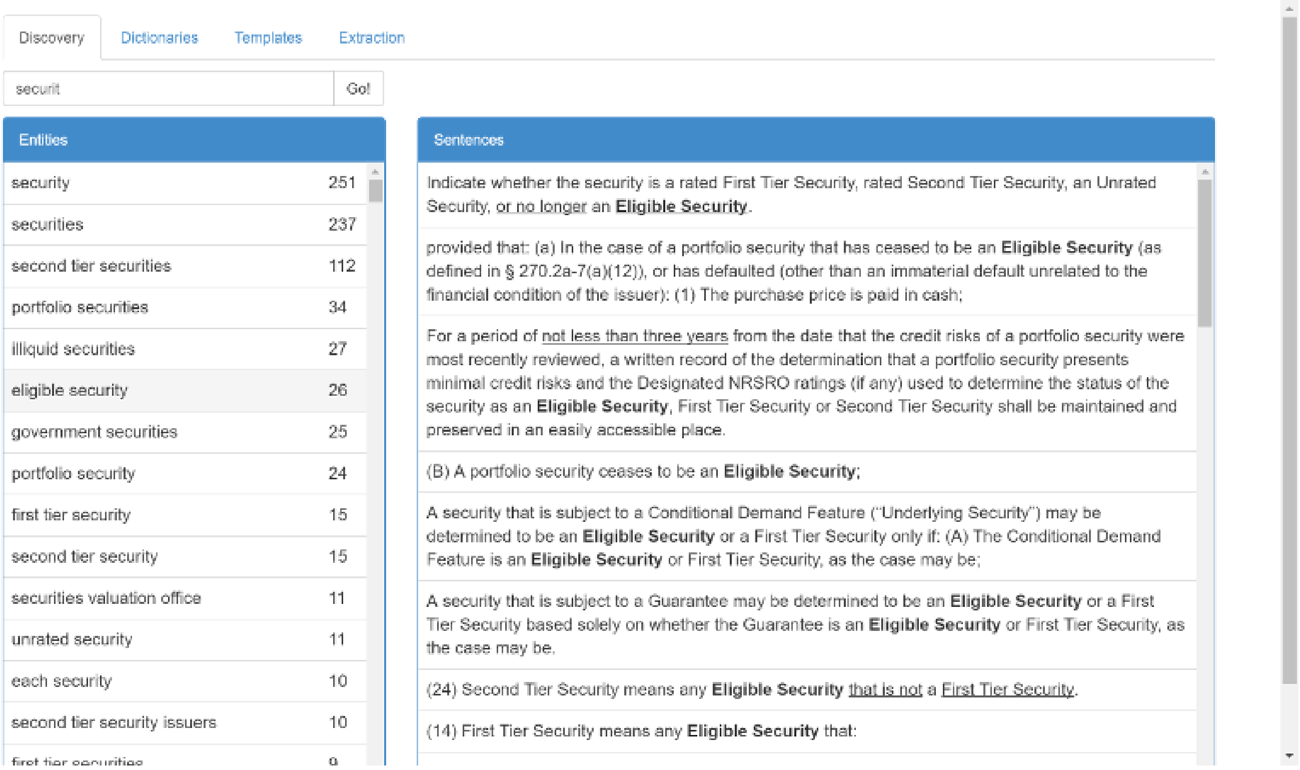
Exploring the domain

As usual, we make sure our demo interfaces have some basic exploration capabilities, giving us the opportunity to showcase what iKnow's output looks like and how these entities differ from classic words. When you click the **Discovery** tab, you'll end up in a very condensed form of the Knowledge Portal, offering its basic capabilities for browsing entities and the sentences listing them. For the screenshots, we've used some sample data from which we've removed all customer-identifiable references. Note that this tab will look rather empty until you start adding data to your domain.



Entities	
fund	344
money market funds	310
security	251
securities	236
money market fund	187
funds	186
comment letter	147
information	143
sec	142
supra	138
insurer	127
proposing release	119
rule	114
second tier securities	112
insurance	106

You can use the input box at the top to search for entities similar to a particular string. When clicking an entity, all sentences listing it will show up on the right. In the screenshot below, you'll already see that iKnow, itself not knowing anything about finance, identifies how word groups are formed that express a particular concept, where an approach based on individual words would have a hard time identifying what the author alludes to.



Entities	
security	251
securities	237
second tier securities	112
portfolio securities	34
illiquid securities	27
eligible security	26
government securities	25
portfolio security	24
first tier security	15
second tier security	15
securities valuation office	11
unrated security	11
each security	10
second tier security issuers	10
first tier securities	9

Sentences

Indicate whether the security is a rated First Tier Security, rated Second Tier Security, an Unrated Security, or no longer an **Eligible Security**.

provided that: (a) In the case of a portfolio security that has ceased to be an **Eligible Security** (as defined in § 270.2a-7(a)(12)), or has defaulted (other than an immaterial default unrelated to the financial condition of the issuer): (1) The purchase price is paid in cash;

For a period of not less than three years from the date that the credit risks of a portfolio security were most recently reviewed, a written record of the determination that a portfolio security presents minimal credit risks and the Designated NRSRO ratings (if any) used to determine the status of the security as an **Eligible Security**, First Tier Security or Second Tier Security shall be maintained and preserved in an easily accessible place.

(B) A portfolio security ceases to be an **Eligible Security**;

A security that is subject to a Conditional Demand Feature ("Underlying Security") may be determined to be an **Eligible Security** or a First Tier Security only if: (A) The Conditional Demand Feature is an **Eligible Security** or First Tier Security, as the case may be;

A security that is subject to a Guarantee may be determined to be an **Eligible Security** or a First Tier Security based solely on whether the Guarantee is an **Eligible Security** or First Tier Security, as the case may be.

(24) Second Tier Security means any **Eligible Security** that is not a First Tier Security.

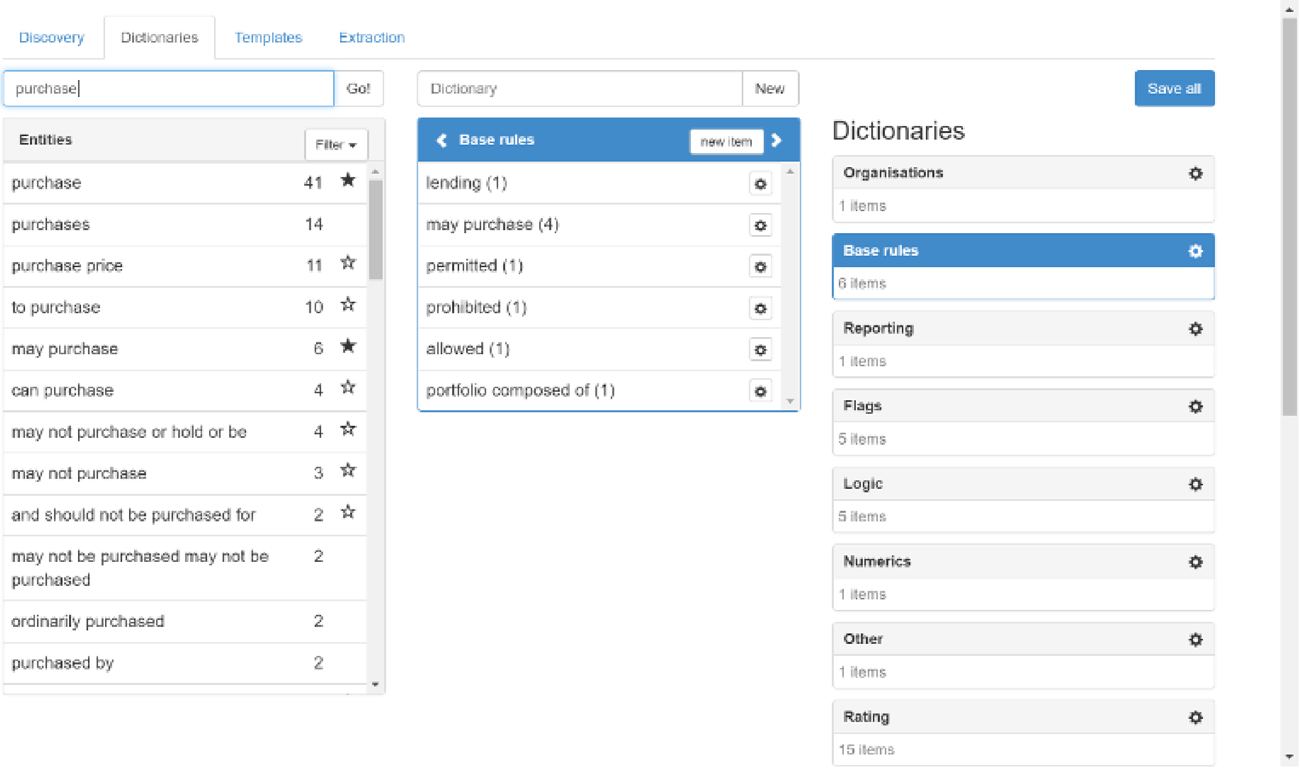
(14) First Tier Security means any **Eligible Security** that:

These are of course still the very basics, so let's move on to the next tab.

Managing dictionaries

In the Dictionaries tab, you can manage the terminology that is of interest to your particular domain. These will be the building blocks for the rule templates we're trying to match to new contracts, so it makes sense to build them in an exploratory way, leveraging iKnow's output for a representative set of contracts.

This tab is literally a nested version of the Dictionary Builder Demo we described last week, so please refer to that article for more details on how to use the interface.



purchase

Entities		
purchase	41	★
purchases	14	
purchase price	11	☆
to purchase	10	☆
may purchase	6	★
can purchase	4	☆
may not purchase or hold or be	4	☆
may not purchase	3	☆
and should not be purchased for	2	☆
may not be purchased may not be purchased	2	
ordinarily purchased	2	
purchased by	2	

Base rules

- lending (1)
- may purchase (4)
- permitted (1)
- prohibited (1)
- allowed (1)
- portfolio composed of (1)

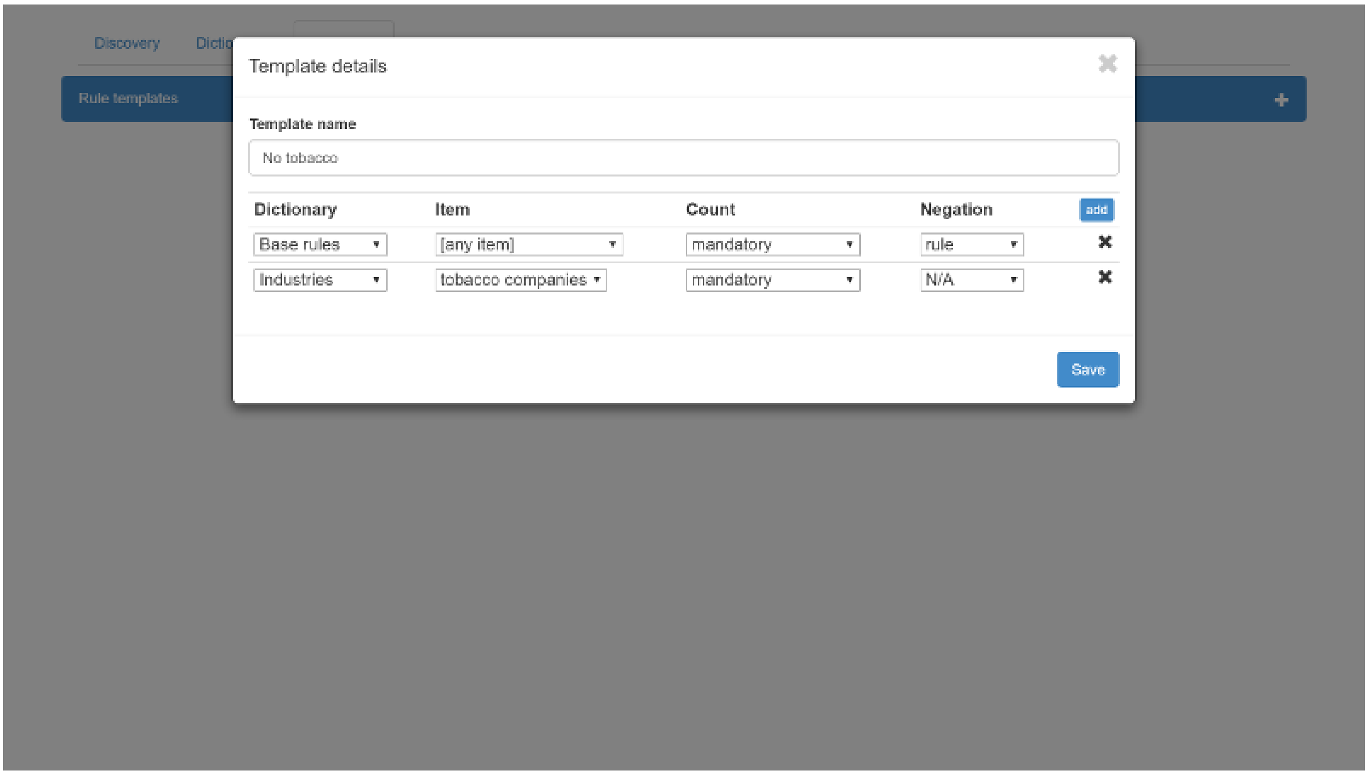
Dictionaries

Dictionary	Items
Organisations	1 items
Base rules	6 items
Reporting	1 items
Flags	5 items
Logic	5 items
Numerics	1 items
Other	1 items
Rating	15 items

Please note you'll have to press the **Save** button in the top right corner in order to save your changes before they can be used in the next tab.

Managing rule definitions

In the **Templates** tab, we'll be joining the building blocks we saved as Dictionaries into sequences that we wish to identify in new contracts. When you first open the tab, you'll see an empty table. Click the plus icon to open the new template dialog.



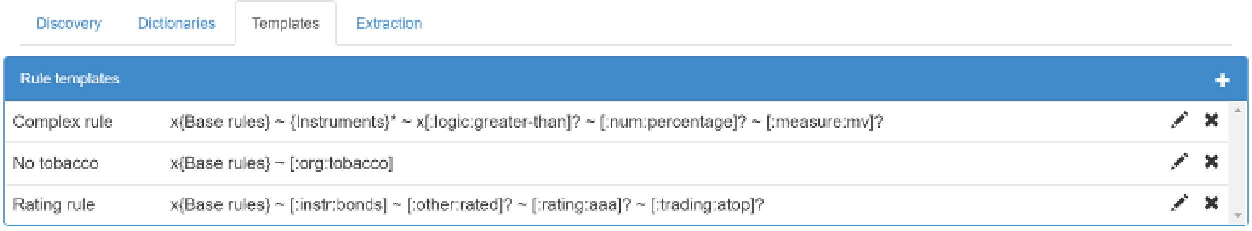
Template name: No tobacco

Dictionary	Item	Count	Negation	add
Base rules ▾	[any item] ▾	mandatory ▾	rule ▾	✖
Industries ▾	tobacco companies ▾	mandatory ▾	N/A ▾	✖

Through this interface, you can assemble a particular sequence of dictionary items (either specific or "any item" of a dictionary) and define what constitutes to a match using a few flags. The *Count* flag expresses how often the item should or may appear in the text, with mandatory and optional the typical values (note that your rule should have at least one mandatory element). With the *Negation* flag, you can express whether that element of the rule should appear in a strictly negative context, an affirmative one (no negation affecting it), or whether that negation context actually is part of the rule itself.

For example, if you have a dictionary with terms like "purchase", "acquire", etc, you can use the "rule" negation flag value to make sure your rule triggers on any occurrence of an entity like purchase and will retain the negation context, so sentences like "Portfolio manager may not purchase Khadaffi inc stock" will properly retain the point made by the author, namely that this particular stock is not to be invested in.

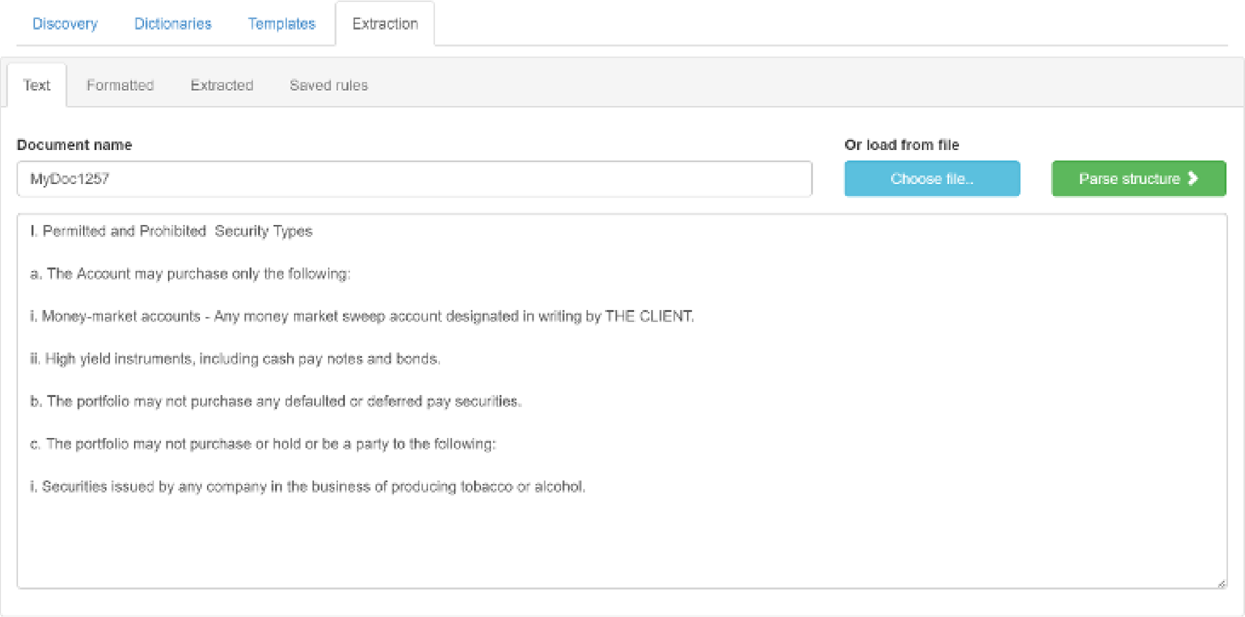
Adding a couple of rules, you'll see how the table gets populated. A shorthand representation of the rule summarizes its elements, using {} for dictionaries, [] for items; a prefix for every element to express the negation flag and a suffix to express the count flag (for example with ? meaning optional).



Complex rule	x{Base rules} ~ {Instruments}* ~ x[:logic:greater-than]? ~ [:num:percentage]? ~ [:measure:mv]?
No tobacco	x{Base rules} ~ [:org:tobacco]
Rating rule	x{Base rules} ~ [:instr:bonds] ~ [:other:rated]? ~ [:rating:aaa]? ~ [:trading:atop]?

Extracting rules

After defining templates for the rules we wish to extract, we can finally start feeding new contracts into the tool and ask it to match the templates against them. Choose the **Extraction** tab to access this part of the interface. You can start by typing a bit of sample text, or loading it straight from a file using the upload button.



Document name: MyDoc1257

I. Permitted and Prohibited Security Types

a. The Account may purchase only the following:

i. Money-market accounts - Any money market sweep account designated in writing by THE CLIENT.

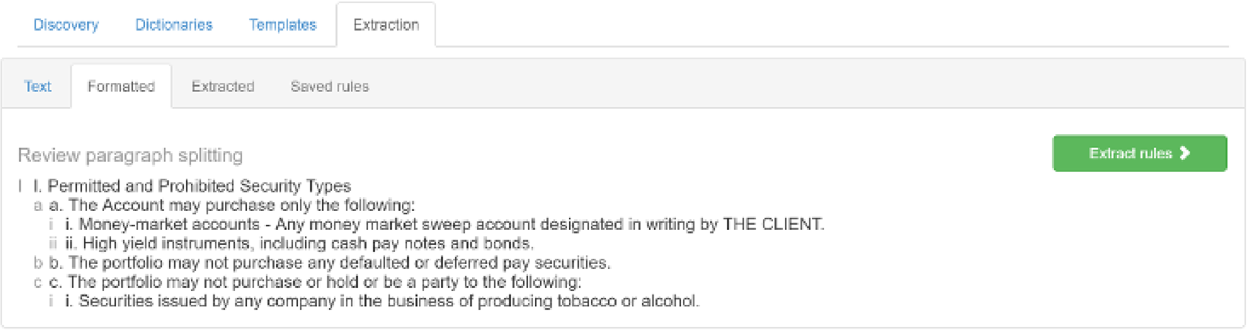
ii. High yield instruments, including cash pay notes and bonds.

b. The portfolio may not purchase any defaulted or deferred pay securities.

c. The portfolio may not purchase or hold or be a party to the following:

i. Securities issued by any company in the business of producing tobacco or alcohol.

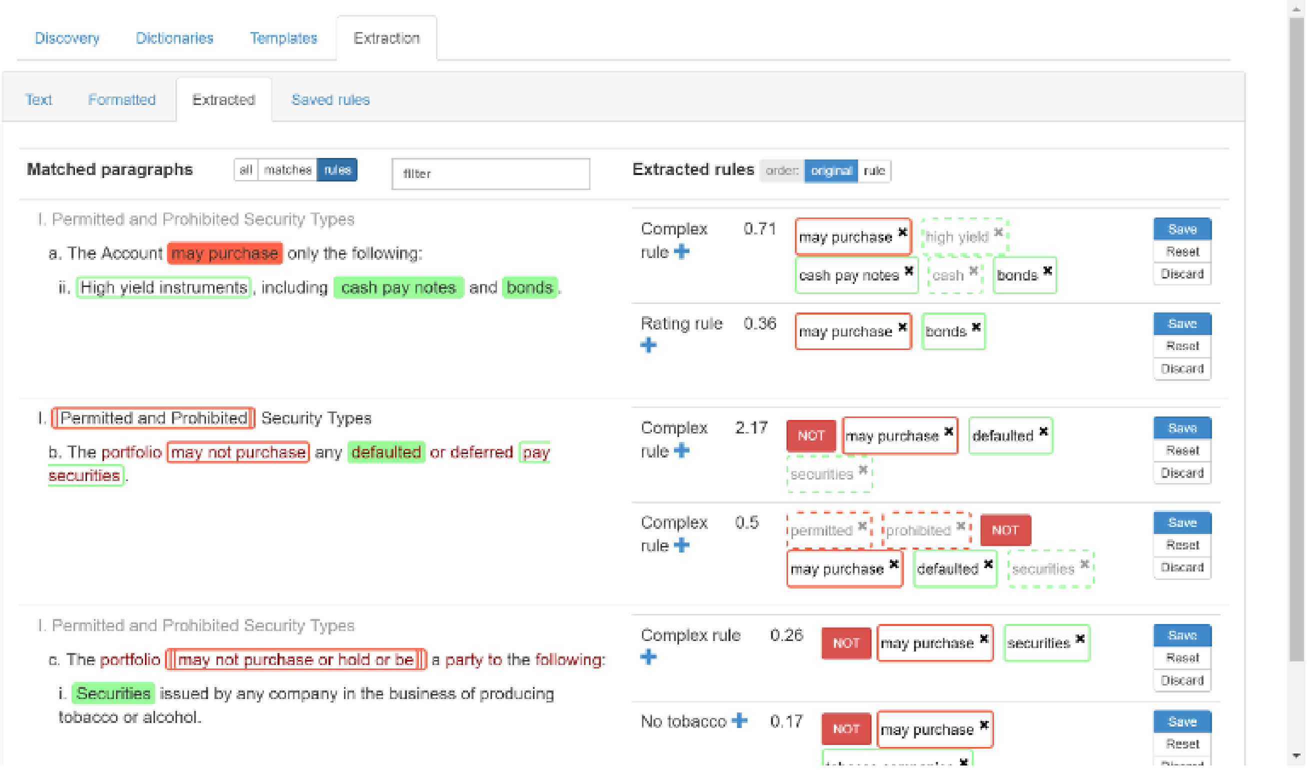
In the particular context for which this demo was originally designed, the paragraph structure of the original document was very important. Therefore, the first step of the extraction process involved trying to capture that structure. This is not a functionality of iKnow (which concentrates on natural language sentences and not document structure), so we implemented something basic but effective in plain COS. If you click the **Parse Structure** button above the text area, you'll proceed to the next stage of this tab-based wizard, and can review if the structure was appropriately parsed. If it's not, for example because an OCR issue messed up the section numbering (there's some tolerance to that) you can simply go back and edit the text, press the parse button again and re-evaluate.



Review paragraph splitting

- I I. Permitted and Prohibited Security Types
  - a a. The Account may purchase only the following:
    - i i. Money-market accounts - Any money market sweep account designated in writing by THE CLIENT.
    - ii ii. High yield instruments, including cash pay notes and bonds.
  - b b. The portfolio may not purchase any defaulted or deferred pay securities.
  - c c. The portfolio may not purchase or hold or be a party to the following:
    - i i. Securities issued by any company in the business of producing tobacco or alcohol.

Then, use the **Extract Rules** button to index all paragraphs one by one. Actually, we're doing it two by two, as in the above example sections I.a. and I.a.i. need to be considered together, as are I.a. and I.a.ii., nicely pairing up every paragraph with its parent. The results may look a little confusing at first if your first set of rules were very relaxed and yield a lot of false positives, but they'll soon start to make more sense as you tune your rules or use the buttons to filter your view.



I. Permitted and Prohibited Security Types
a. The Account may purchase only the following:
ii. High yield instruments, including cash pay notes and bonds.

Complex rule	0.71	may purchase ✖ high yield ✖ cash pay notes ✖ cash ✖ bonds ✖
Rating rule	0.36	may purchase ✖ bonds ✖

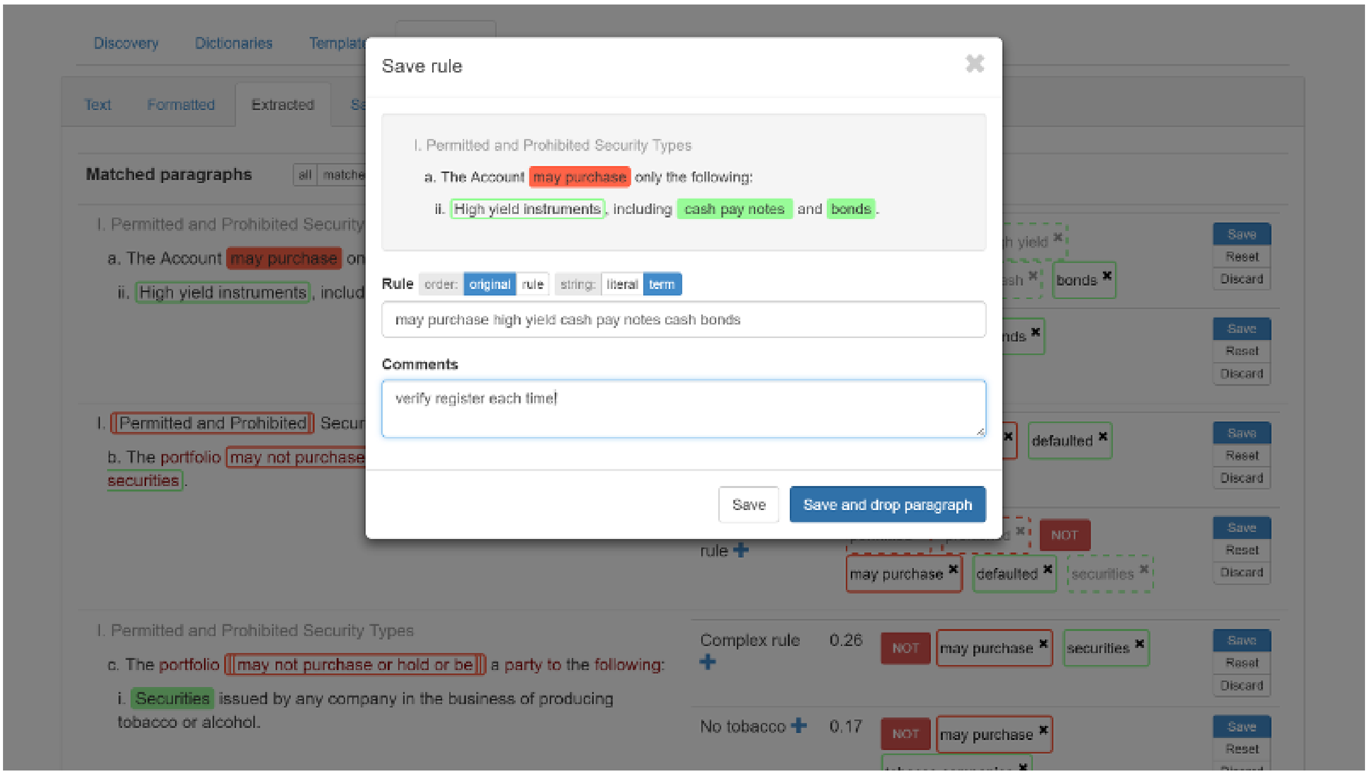
I. Permitted and Prohibited Security Types
b. The portfolio may not purchase any defaulted or deferred pay securities.

Complex rule	2.17	NOT may purchase ✖ defaulted ✖ securities ✖
Complex rule	0.5	permitted ✖ prohibited ✖ NOT may purchase ✖ defaulted ✖ securities ✖

I. Permitted and Prohibited Security Types
c. The portfolio may not purchase or hold or be a party to the following:
i. Securities issued by any company in the business of producing tobacco or alcohol.

Complex rule	0.26	NOT may purchase ✖ securities ✖
No tobacco	0.17	NOT may purchase ✖

For the first paragraph, the app identified two rule template matches: one for the "Complex rule" and one for the "Rating rule", having a score of 0.71 and 0.36 respectively. The formula for these scores is rather complex, but it takes into account factors such as the quality of the individual element's dictionary match (full or partial) and how *compact* the overall combination of matches was, favouring template matches that only span a single sentence. If you hover over the match element overview next to the score, the text fragment on the left will be highlighted according to the same color scheme, which can be helpful if you have a complex rule. You can also discard elements that were not relevant using the x icon and, once you're fine with a candidate match, press the save button to retain the rule and "save" it so it can be forwarded to the external system (as demonstrated in this GS2016 session).



Save rule

I. Permitted and Prohibited Security Types

a. The Account may purchase only the following:

ii. High yield instruments, including cash pay notes and bonds.

Rule: order: original; string: term

may purchase high yield cash pay notes cash bonds

Comments

verify register each time!

The last tab of our extraction wizard allows you to download all "saved" rules into a CSV file. Please note that "save" only means save for the duration of the session, as we're not storing any of that in a table. It's still a demo, after all... ;o)

Wrapping up

This was the fourth and by far most "real" demo application we've covered in this series so far. We hope you enjoyed walking through them and have further ideas for your own applications or how we could make these even better and reusable. We also hope you saw how they gradually grew more complex, starting from just browsing indexing results (Knowledge Portal demo), organizing content according to its contents (Set Analysis demo) over formalizing your domain knowledge based on real-world terminology (Dictionary Builder demo) into this versatile rules building demo application.

#InterSystems Natural Language Processing (NLP, iKnow)
Check the related application on InterSystems Open Exchange

Source URL:https://community.intersystems.com/post/iknow-demo-apps-part-4-rules-builder-demo