# i2 TextChart Studio

Welcome to i2 TextChart Studio. This software enables you to tune TextChart processing for your organization by customizing or creating the LxBases and GxBases that TextChart uses. Users can apply the new or updated LxBases and GxBases to both TextChart and TextChart Premium

**Note:** i2 TextChart Studio is only supported with the Google Chrome browser. Other browsers may cause unusual errors when using TextChart Studio.

## Administering TextChart Studio

This part of the documentation describes how to install, configure, and secure access to i2 TextChart Studio.

## Installing TextChart Studio

i2 TextChart Studio is a Java application that you can install on Microsoft Windows, Apple macOS, and Linux operating systems. The procedure in this topic describes the installation process.

1. Double-click the `Studio-Installer.jar` file to begin, and then click **Next**.

   **Note:** You can also type `java -jar Studio-Installer.jar` at a command prompt to start the installer.
2. Read the license agreement, select **I accept the terms of this license agreement**, and click **Next** to continue.
3. Provide the installer with an installation path, and click **Next** to continue.

   By default, TextChart Studio is installed in a directory under `C:\Program Files` on Windows, under `/Applications` on macOS, and under `/opt` on Linux.

   **Important:** Always install TextChart Studio into a new directory. Do not install the software over previous installations.
4. On the **Select Path** page, choose a directory in which to store user data, and click **Next** to continue.
5. On the **User Data** page, select the amount of memory to devote to the Java virtual machine running the TextChart Studio process and, if desired, modify the network port on which to connect. Click **Next** to continue.

   **Note:** To change these settings after installation, you can configure the JVM memory in the `runtime.properties` file, and the HTTP port in the `studio.properties` file.
6. The installer displays a summary of your chosen settings. Click **Next** to continue.
7. The installation process begins. As it proceeds through each phase, click **Next** to continue, until you see the **Installation Finished** page.
8. Click **Done** to finish and exit the installer.

### User data

TextChart Studio stores user data (including configuration settings, the LxBase, and regression points) in an operating-system-specific user area. Uninstalling TextChart Studio doesn't delete this data, and reinstalling TextChart Studio doesn't overwrite it.

To examine or delete TextChart Studio user data, see the following locations:

- Windows: The `%APPDATA%/TextChartStudio` directory
- Linux: The `$HOME/.textchartstudio` directory
- macOS: The `$HOME/Library/Application Support/TextChart Studio` directory

## Starting and stopping TextChart Studio

i2 TextChart Studio is a web application that runs on the (included) Jetty web server, on the user's workstation. The application must be running before the user can access it through their web browser.

The commands for starting and stopping the TextChart Studio application are slightly different, depending on whether the user has a Windows, Linux, or macOS workstation.

**Note:** On all platforms, to restart the TextChart Studio application, use the "stop" and "start" commands in sequence.

### Windows

To start TextChart Studio on Microsoft Windows:

- From the Start menu, select **TextChart Studio Start**. Alternatively, open a command window, navigate to the installation directory, and run the following command:

  ```
  rosokastudio.bat start
  ```

To stop TextChart Studio:

- From the Start menu, select **TextChart Studio Stop**. Alternatively, run the following command:

  ```
  rosokastudio.bat stop
  ```

### Linux

To start TextChart Studio on Linux:

- From the installation directory, run the following command:

  ```
  ./rosokastudio.sh start
  ```

To stop TextChart Studio:

- From the installation directory, run the following command:

  ```
  ./rosokastudio.sh stop
  ```

### macOS

On Apple macOS, you can use the same commands for starting and stopping TextChart Studio as on Linux. i2 also provides commands that you can run from the Finder.

To start TextChart Studio on macOS:

- Double-click the `StartStudio.command` file in the installation directory.

To stop TextChart Studio:

- Double-click the `StopStudio.command` file in the installation directory.

## Managing TextChart Studio users

A fresh installation of i2 TextChart Studio comes with a pre-populated user account named **rosoka** that has administrator privileges. The initial password is the same as the username.

To add users to the system, or to change the password for a user, you use a command-line utility from the TextChart Studio installation directory. On Microsoft Windows, the utility is `rosokastudio.bat`; on Apple macOS and Linux, the utility is `rosokastudio.sh`.

To manage TextChart users, navigate to the TextChart Studio installation directory and run the command for your operating system, providing three parameters:

- The command name (**passwd**)
- The name of the user to add or modify
- The new password for the named user

For example, to create a user named **joe** and set their password to `xx7799`, run this command on Windows:

```
rosokastudio.bat passwd joe xx7799
```

Or this command on macOS and Linux:

```
./rosokastudio.sh passwd joe xx7799
```

On completion, the command provides the following output:

```
The new encrypted password for the user 'joe' is

   CRYPT:joKVahDE8hMXY

Copy this into the file 'conf/auth.properties' on the appropriate
line for the user 'joe'. Here is the full line for a user:

joe: CRYPT:joKVahDE8hMXY,rosokauser
```

Follow the instructions in the command output:

1. Using a text editor, open the `auth.properties` file, which is located in the `conf` directory under the TextChart Studio installation directory.
2. Paste the text from the output into the file, either replacing a line (to change the password for an existing user) or adding a line to the end (to add a new user).

   For example, after adding **joe** to the default user provided with TextChart Studio, the edited file would look like this:

   ```
   rosoka: CRYPT:roVsWZ.bWXgsk,rosoka
   joe: CRYPT:joKVahDE8hMXY,rosoka
   ```
3. Save the file, and then start or restart the TextChart Studio server.

On completion of the above steps, the new or modified user becomes available for logging in to i2 TextChart Studio.

## Configuring user access through LDAP

If your organization uses LDAP (Lightweight Directory Access Protocol) as a site-wide authentication and authorization system, then you can configure TextChart Studio to use LDAP in place of its own user management system.

1. To configure TextChart Studio to access the LDAP server, you need to edit the supplied `conf/ldap.conf` file.

   Because interfacing with LDAP can be complex, i2 recommends contacting your IT department for assistance in setting the values in the LDAP configuration file.

**Note:** To use TextChart Studio with an *LDAPS* (LDAP with SSL) server, see Configuring SSL for Studio for information about creating and managing SSL certificates and key stores.

2. To set up LDAP authentication, open the `conf/studio.properties` file in a text editor, and set the `authType` property to `ldap`:

   ```
   authType=ldap
   ```

3. To restrict access to the TextChart Studio user interface to members of a particular LDAP group, also set the `authGroup` property to the name of that group. For example:

   ```
   authGroup=TextChartUsers
   ```

   Alternatively, to allow all LDAP users to access TextChart Studio, either don't set the `authGroup` property or set it to the default value, `**`.

4. When your changes to `conf/ldap.conf` and `conf/studio.properties` are complete, restart TextChart Studio to make them take effect.

## Changing the default HTTP port

By default, users connect to i2 TextChart Studio through HTTP on port 8080. If this port is already in use in your organization - or for any other reason - you can configure TextChart Studio to use a different port.

1. Open the `conf/studio.properties` file in a text editor.
2. Find the line that contains `httpPort=8080`, and change the port number to the new value. For example, to change the port to 8081, the line would look like this:

   ```
   httpPort=8081
   ```

3. Save your changes and close the file.

To verify your changes, restart TextChart Studio, add the new port number to the URL in your browser, and reopen the application.

## Configuring SSL access to TextChart Studio

If you will access TextChart Studio only from the the workstation it's installed on, then setting up access through SSL is probably unnecessary. However, if others will use TextChart Studio from other workstations, i2 recommends configuring SSL connections.

**Note:** These instructions assume that you are familiar with the general use of SSL certificates and key stores. Following the instructions sets up the Jetty server that runs TextChart Studio to use SSL with a self-signed certificate. For more advanced use, see Managing SSL keys and certificates.

Enabling SSL for TextChart Studio involves creating a certificate *keystore* file, setting passwords for the keystore and the keystore manager, and enabling SSL functionality in the `conf/studio.properties` file:

1. Use the `keytool` program from the Java JDK to create a new keystore. Navigate to the `etc` directory and run the following command:

   ```
   keytool -keystore keystore -alias studio -genkey -keyalg RSA
   ```

   Answer the questions appropriately for your site, and make a note of the password that you provide for the certificate generation process. When the command finishes, it generates a file named `keystore` file in the same directory.

   **Note:** On Microsoft Windows, you might need to add the JDK binary directory to your path in order to run the `keytool` command. You can do so with a command like this:

```
set PATH=%PATH%;"c:\Program Files\Java\jdk1.8.0_65\bin"
```

Adjust the path and the version number to match your JDK installation.

2. Next, you must add the password that you entered during the certificate creation process to the TextChart Studio properties file. So that the password is not visible in plain text, you obfuscate it.

   Run the following command at the command line, substituting the password you entered during the certificate creation process in place of `<password>`:

   ```
   java -cp RosokaStudio.jar org.eclipse.jetty.util.security.Password
    <password>
   ```

   **Note:** Run this command from the TextChart Studio installation directory.

   The command outputs the obfuscated password to the console. Copy it (including the initial `OBF:`) for use in the next step.

3. Open the `conf/studio.properties` file in a text editor, remove the # signs from the lines below, and enter the port you want to use, the path to the keystore, and the obfuscated password as values for the following settings:

   ```
   httpsPort=8443
   keyStorePath=
   keyStorePassword=
   keyManagerPassword=
   ```

   To prevent unsecured HTTP access to TextChart Studio, comment out or remove the `httpPort` setting, and then save the modified file.

4. Restart TextChart Studio.

After making these changes, you can access TextChart Studio on the local workstation at `https://localhost:8443/RosokaStudio`. (If you changed the port number, you'll need to make the same change to the URL.)

Using a self-signed certificate like this generates a warning when you access TextChart Studio through a web browser. You can instruct your browser to ignore the error and proceed to the page, but to remove it completely you must get your certificate signed by a valid certification authority. See Managing SSL keys and certificates for more information.

## Managing SSL keys and certificates

This topic contains additional information about generating and managing SSL keys and certificates for use with deployments of i2 TextChart Studio.

### Generating keys and certificates with keytool

The simplest way to generate keys and certificates is to use the `keytool` application that comes with the JDK, as it generates keys and certificates directly into the keystore.

If you already have keys and certificates, jump to Loading keys and certificates to load them into a Java Secure Socket Extension (JSSE) keystore. That section also applies if you have a renewal certificate to replace one that is expiring. The examples below generate only basic keys and certificates.

The following command generates a key pair and a certificate directly into a file named `keystore`:

```
keytool -keystore keystore -alias jetty -genkey -keyalg RSA
```

The command prompts for information about the certificate, and for passwords to protect both the keystore and the keys within it. The only mandatory response is to provide the fully qualified host name of the server at the "first and last name" prompt.

You now have the minimum requirements for creating an SSL connection, and you could proceed directly to configuring one. However, the browser will not trust the certificate you generated, and will prompt the user to this effect. What you have at this point is sufficient for testing, but not for production.

If you want to use only a self-signed certificate, you can add `-validity <days>` to the `keytool` call above to extend the validity of the certificate beyond a month, which is the default period.

You can also use the SAN extension to set one or more names that the certificate applies to:

```
keytool -keystore keystore -alias jetty -genkey -keyalg RSA -sigalg
 SHA256withRSA --ext 'SAN=dns:jetty.eclipse.org,dns:*.jetty.org'
```

### Requesting a trusted certificate

To obtain a certificate that most common browsers will trust, you need to ask a well-known certificate authority (CA) to sign it. Trusted CAs include AddTrust, Entrust, GeoTrust, RSA Data Security, Thawte, VISA, ValiCert, Verisign, and beTRUSTed.

Each CA has its own instructions (look for information about "JSSE" or "OpenSSL"), but all involve a step that generates a certificate-signing request (CSR).

### Generating a CSR with keytool

The following command uses `keytool` to generate the file `jetty.csr` for a key or certificate that's already in the keystore:

```
keytool -certreq -alias jetty -keystore keystore -file jetty.csr
```

### Loading keys and certificates

When a CA has sent you a certificate, or if you generated your own certificate without `keytool`, you need to load it into a JSSE keystore.

**Note:** You need both the private key and the certificate in the JSSE keystore. You should load the certificate into the keystore that was used to generate the CSR with `keytool`. If your key pair is not already in a keystore (for example, because it was generated with OpenSSL), you need to use the PKCS12 format to load both the key and certificate.

#### Loading certificates with keytool

You can use `keytool` to load a certificate in PEM format directly into a keystore. The PEM format is a text encoding of certificates; it is produced by OpenSSL, and is returned by some CAs.

The following command loads a PEM-encoded certificate in the `jetty.crt` file into a JSSE keystore:

```
keytool -keystore keystore -import -alias jetty -file jetty.crt -
trustcacerts
```

If the certificate that you receive from the CA is not in a format that `keytool` understands, you can use the `openssl` command to convert formats:

```
openssl x509 -in jetty.der -inform DER -outform PEM -out jetty.crt
```

#### Loading keys and certificates via PKCS12

If you have a key and certificate in separate files, you need to combine them into a PKCS12 format file to load into a new keystore. The certificate can be one you generated yourself or one returned from a CA in response to your CSR.

The following OpenSSL command combines the keys in `jetty.key` and the certificate in the `jetty.crt` file into a file named `jetty.pkcs12`:

```
openssl pkcs12 -inkey jetty.key -in jetty.crt -export -out jetty.pkcs12
```

If you have a chain of certificates because your CA is an intermediary, build the PKCS12 file as follows:

```
cat example.crt intermediate.crt [intermediate2.crt] ... rootCA.crt > cert-
chain.txt
openssl pkcs12 -export -inkey example.key -in cert-chain.txt -out
 example.pkcs12
```

The order of certificates must be from server to rootCA, as described in RFC2246 section 7.4.2.

OpenSSL asks for an *export password*. A non-empty password is required to make the next step work. Then load the resulting PKCS12 file into a JSSE keystore with `keytool`:

```
keytool -importkeystore -srckeystore jetty.pkcs12 -srcstoretype PKCS12 -
destkeystore keystore
```

### Renewing certificates

If you are updating your configuration to use a newer certificate because the old one is expiring, just load it as described in Loading keys and certificates.

If you originally imported the key and certificate using the PKCS12 method, use an `alias` of `1` rather than `jetty`, because that is the alias the PKCS12 process enters into the keystore.

# Using TextChart Studio

This part of the documentation describes what i2 TextChart Studio does, how it works, and how to use it.

## Getting started

When you start to use i2 TextChart Studio for the first time, you need to log in, provide your license information, and perform some initial customization.

### Logging in

To log in to TextChart Studio, enter your username and password on the front page. A development system includes a default account with the following credentials:

- **Username:** rosoka
- **Password:** rosoka

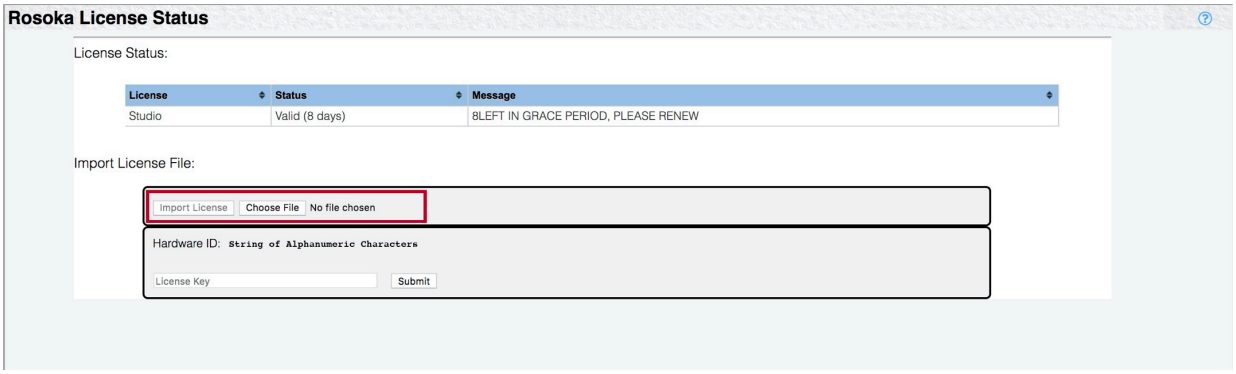
### Accessing settings

To access TextChart Studio settings (including license provision), use the toolbar on the left of the application window. The button at the top of the column opens the **Licenses & LxBase Settings** section of the toolbar:

The first button in the open section allows you to **Manage Licenses**; the second lets you **Edit Property Settings**.

### Providing a license key

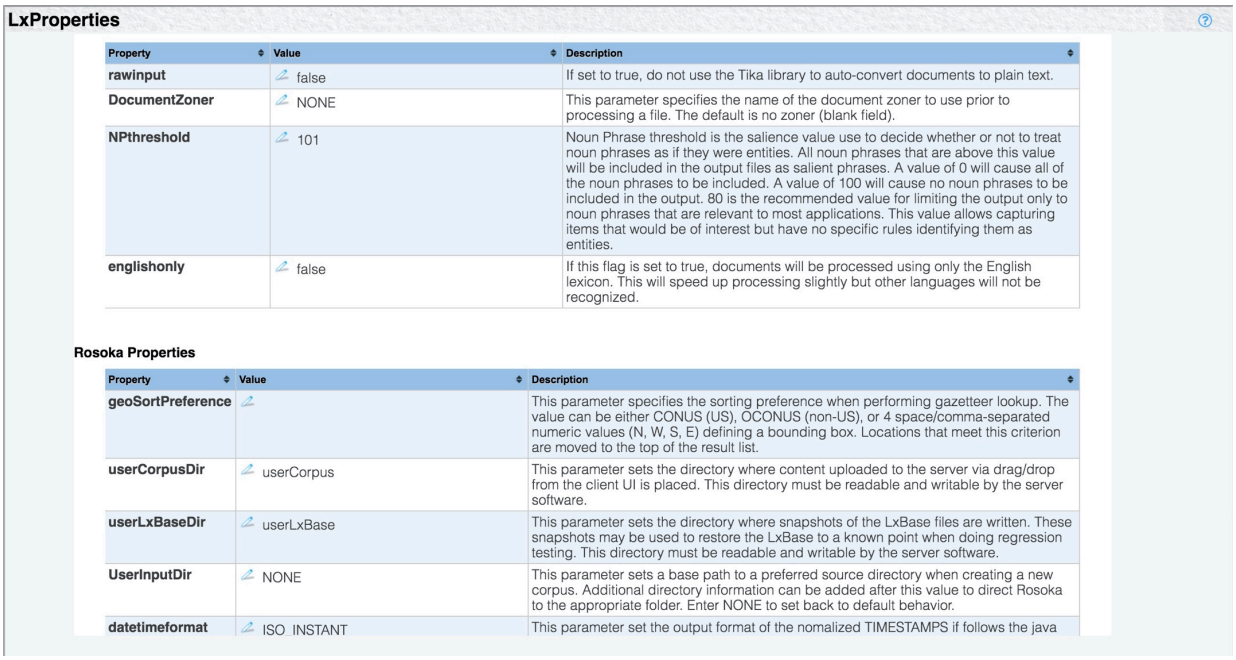
To provide TextChart Studio with a license key, click **Manage Licenses** to open the **Rosoka License Status** page.




The top part of the page indicates how long a particular license is valid for. The bottom part allows you to provide a new license key. Click **Choose File** to locate your `licenseKeys.xml` file, and then click **Import License**.

### Changing settings

To change TextChart Studio settings, click **Edit Property Settings** to open the **LxProperties** page.




Some of the most frequently modified properties are the following:

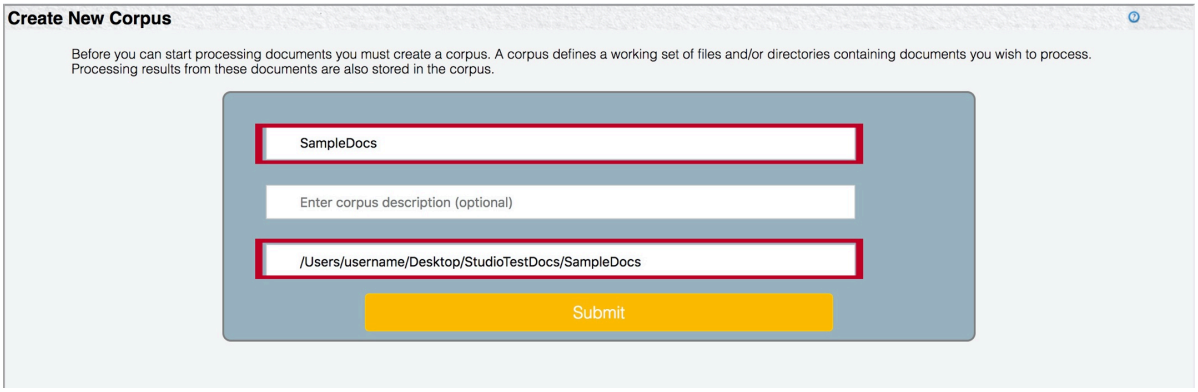
- **proximityrelationship**

  When `proximityrelationship` is `true`, relationship extraction includes relationships based on the entities' proximity to one another.

- **UserInputDir**

  The `UserInputDir` setting specifies the location of the directory where TextChart Studio looks for sets of documents (*corpora*) by default.

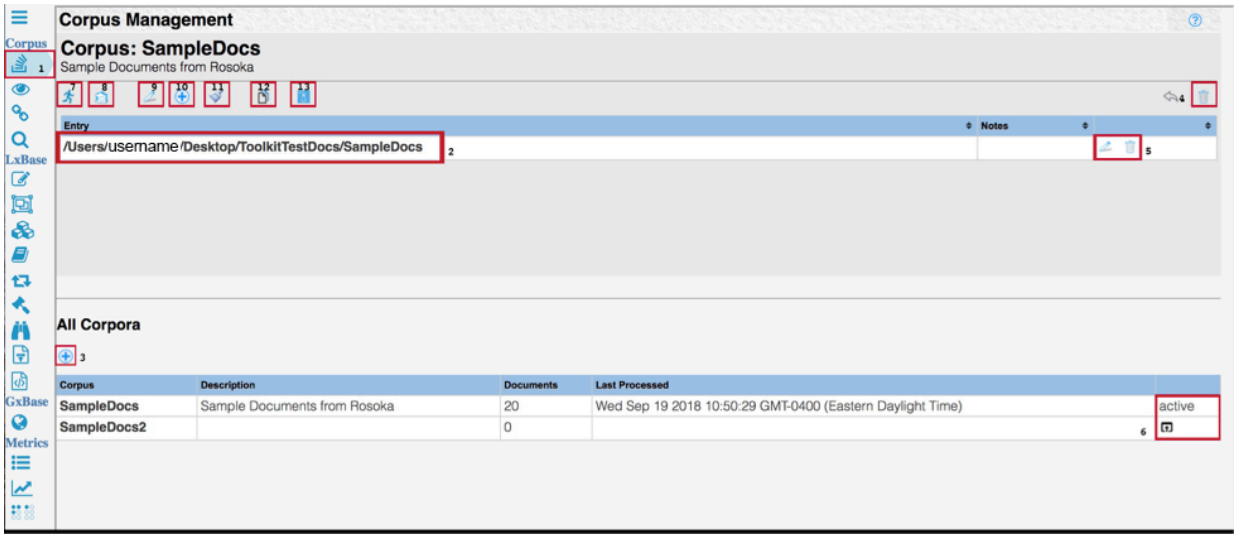
  The directory that you provide here is used to pre-populate the path on the **Create New Corpus** page.



# Corpus management

When you first log in to i2 TextChart Studio, you're prompted to create an initial corpus of documents that you'll use to evaluate the changes you make to the LxBase. optionally, you can create multiple, named corpora with different sets of documents.

The **Corpus Management** page in TextChart Studio allows you to add, remove, and process corpora; to toggle between different saved corpora; and to perform a variety of other document management functions.




1. To open the the **Corpus Management** page, click the **View Corpus List** icon located towards the top of toolbar.
2. The list at the top of the page indicates the *active* corpus. Corpus management commands act only on the active corpus.
3. To return to the **Create New Corpus** page, click the **Add a New Corpus** button above the **All Corpora** list at the bottom of the page.
4. To remove the active corpus from the **Corpus Management** page, click **Delete Corpus**.
5. To edit the path of the active corpus, use the **Edit Entry** and **Delete Entry** buttons.
6. To make a different corpus active, click the **Open Corpus** button on the right of the **All Corpora** list.
7. To make TextChart Studio process all the documents in the active corpus, click **Process all Documents** in the horizontal toolbar.
8. To take a snapshot of the LxBase in its current form for later comparisons, click **Create New Regression Point**.

   You can use TextChart Studio's regression testing feature to measure the impact that your changes to the LxBase have on the output. For more information, see Regression Testing.
9. To change the name or the description of the active corpus, click **Edit Corpus Name/Description**.
10. To add more documents to the active corpus, click **Add More Files**.
11. If you modify any dictionaries or linguistic rules after you process the active corpus, you must clear the previous results before you reprocess the corpus. To do so, click **Clear Processing Results**.

12. TextChart Studio automatically removes duplicate documents upon processing. Click **Show Duplicate Documents** to see a list of these duplicates. Raw documents are documents that are the same before Tika processing; content duplicates are documents that are the same after Tika processing.
13. Sometimes, TextChart finds words that it does not understand. Typically, these words belong to subject-specific vocabularies, or they pertain to proprietary industry information, or they are non-English words for which TextChart has no translation.

    To download a list of these words, potentially to add them to a custom LxBase, click **Download Unknown Words**

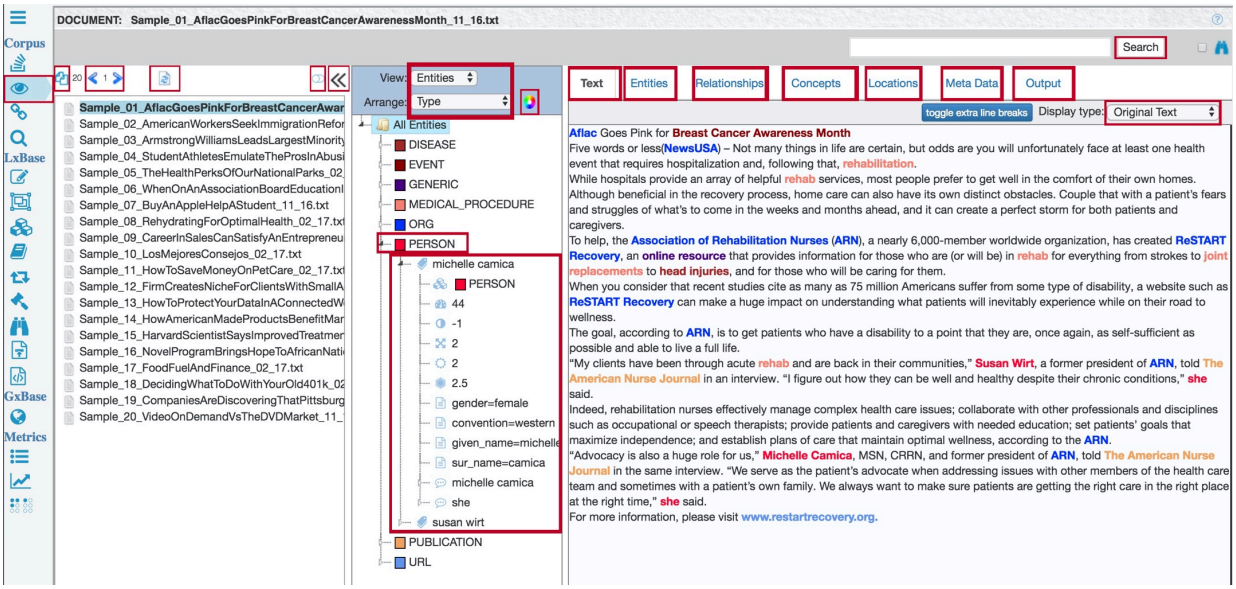
    

    above the entry information line.

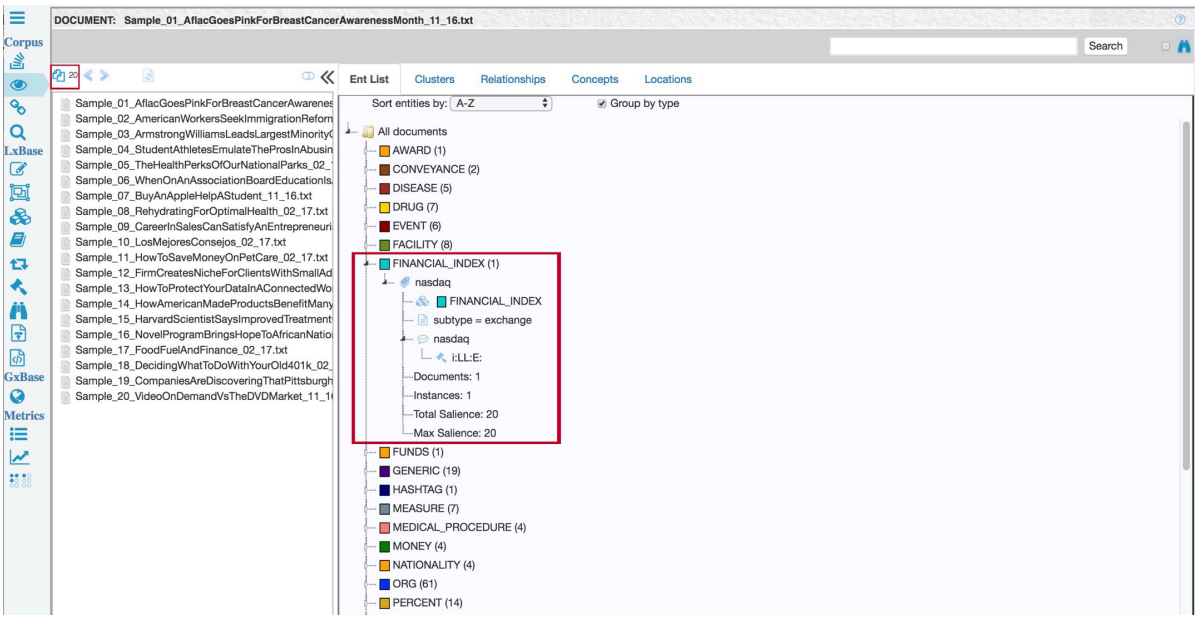
    Select which language to download an unknown word list for, and then click **Submit** to compile the list, which can take several minutes. When you see a notification, click the icon to download the list as a ZIP file.

## Viewing corpus results

By default, the **Active Corpus** page contains information about a single document in that corpus. Here, you can view a single document with hit highlights, select certain entities to view, explore the lexical items, and view rule traces.

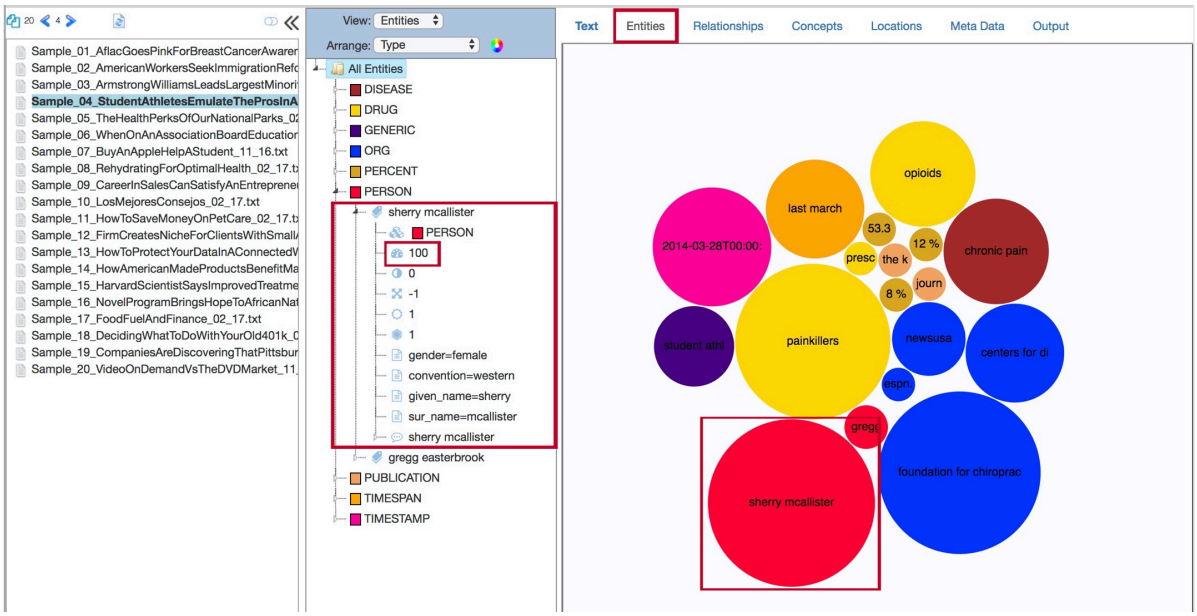



1. To open the **Active Corpus** page, click the **View Active Corpus** icon near the top of the vertical toolbar.
2. To see a summary of the extraction results for the whole corpus, click **Aggregate View**.




Each entity type that TextChart Studio found in the corpus is listed, along with its total number of extraction instances. You can open entries in the list to see individual extraction results and their corresponding metadata.

3. To view the results for each document in the corpus, click the **Next Document** and **Previous Document** buttons. TextChart Studio displays the text for the document on the right of the page, with highlights corresponding to the extracted entities.
4. When you make changes to the LxBase, and you want to see the effect of those changes on a single document without reprocessing the whole corpus, click **Reprocess the Current Document**.
5. Instead of navigating through the documents in the corpus, click **Switch to Search Mode** to view the results for a specific document by name.
6. When you select a single document in the corpus, the tree view in the center of the **Active Corpus** page displays the extraction results for that document. Use the **View** and **Arrange** lists to change how the results are represented.
7. On the right of the **Active Corpus** page, the **Text** tab contains the text of the selected document with hit highlights corresponding to the extracted entities.
8. The **Entities** tab contains a visualization of extracted entities and their overall importance within the selected document.




The larger the circle that represents the entity, the more important (or *salient*) that entity is to the document. Salience is scored from 0 to 100, with 100 representing the most entity.

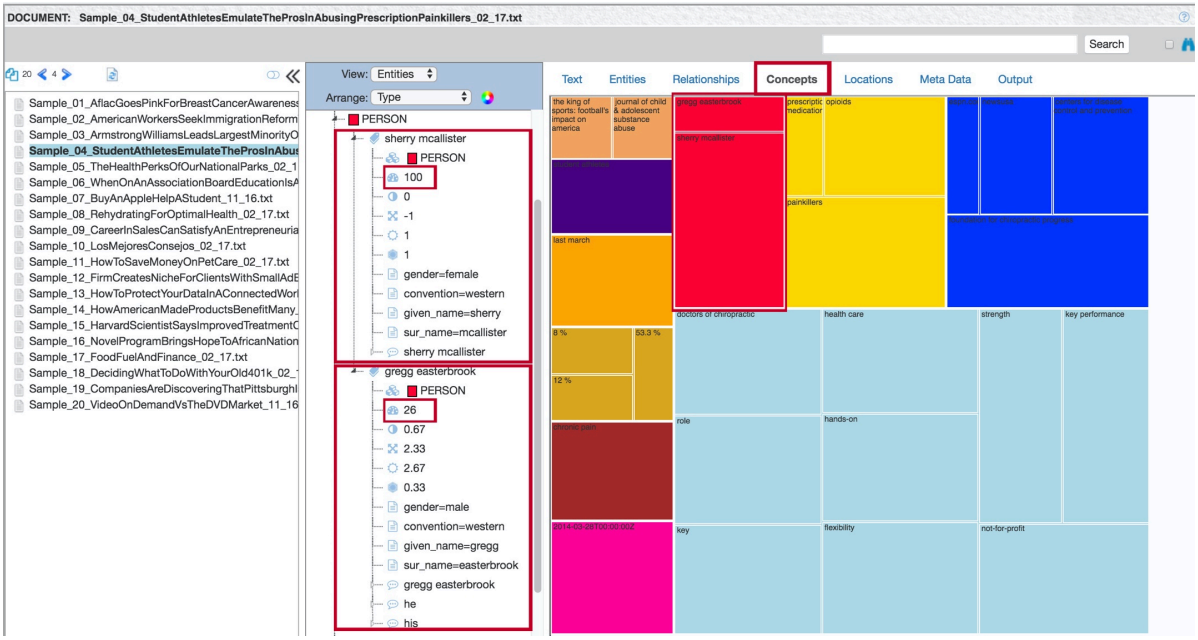
The **Entities** tab is also available in the aggregate view, where its contents represent the salience of extracted entities to to the corpus as a whole.

9. The **Relationships** tab contains a visualization of extracted entities and their relationships to each other in the selected document.


You can interact with the contents of the **Relationships** tab to gain information about the subjects, objects, and predicates that prompted each extracted relationship.

The tab is also available in the aggregate view, where its contents represent relationships between entities across the whole corpus.

10. The **Concepts** tab contains a visualization of extracted entities, as well as other words or phrases that TextChart has identified as salient by their overall importance to the selected document.

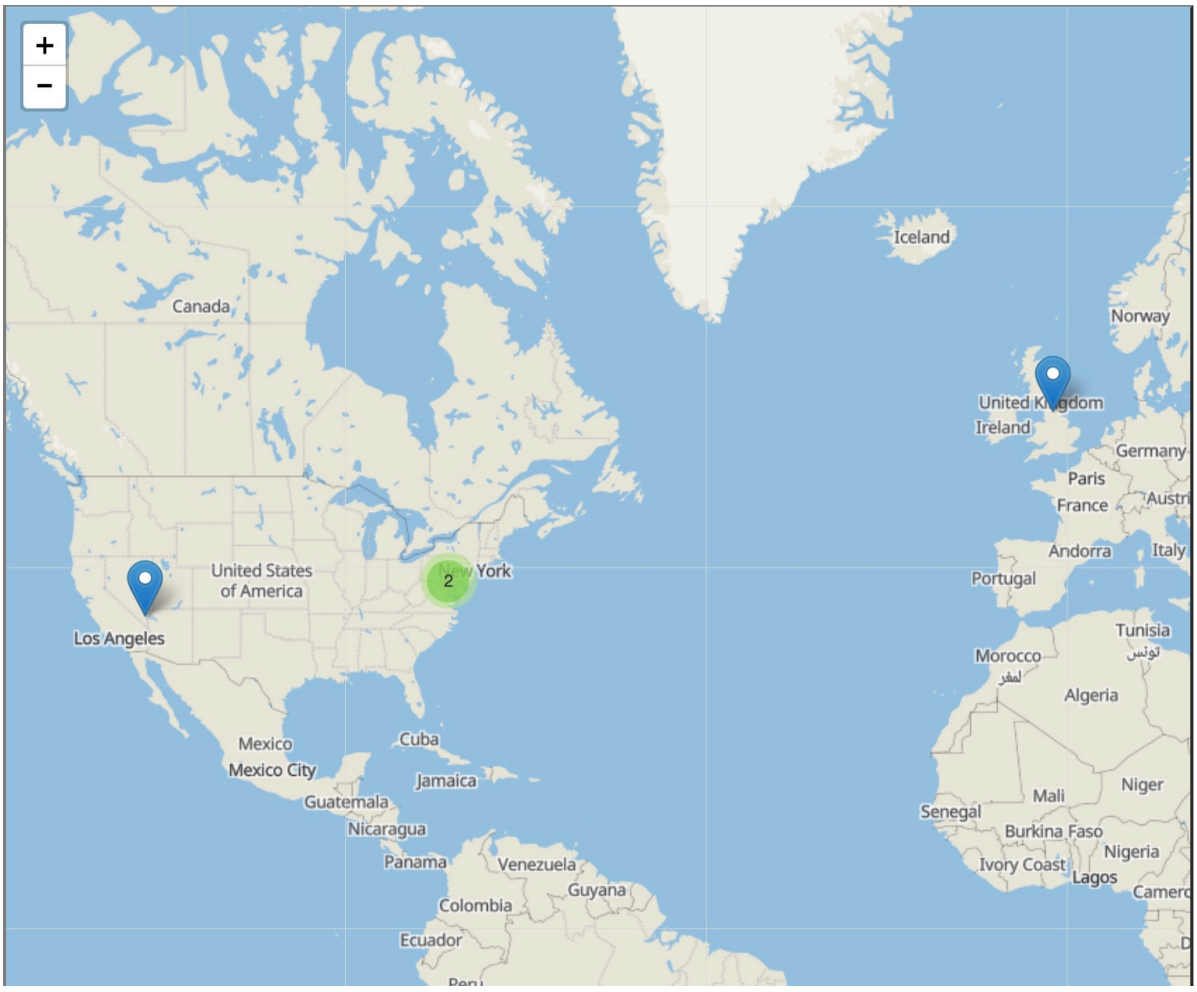



The larger the box that represents the entity, the more important (or *salient*) that entity is to the document.

The brightly colored boxes represent individually extracted entities, while the light blue boxes represent salient words or phrases. You can use this information to see if there are any additional items that need to be modified to reach your extraction goals.

The **Concepts** tab is also available in the aggregate view, where its contents represent the salience of words and phrases to the corpus as a whole.

11. The **Locations** tab contains a visualization of extracted entities and their geographic locations from the selected document.




This tab is also available in the aggregate view, where its contents represent the locations of entities from across the corpus as a whole.

12. The **Metadata** tab presents the user with document-level metadata, including overall sentiment scores and the languages identified in the selected document.

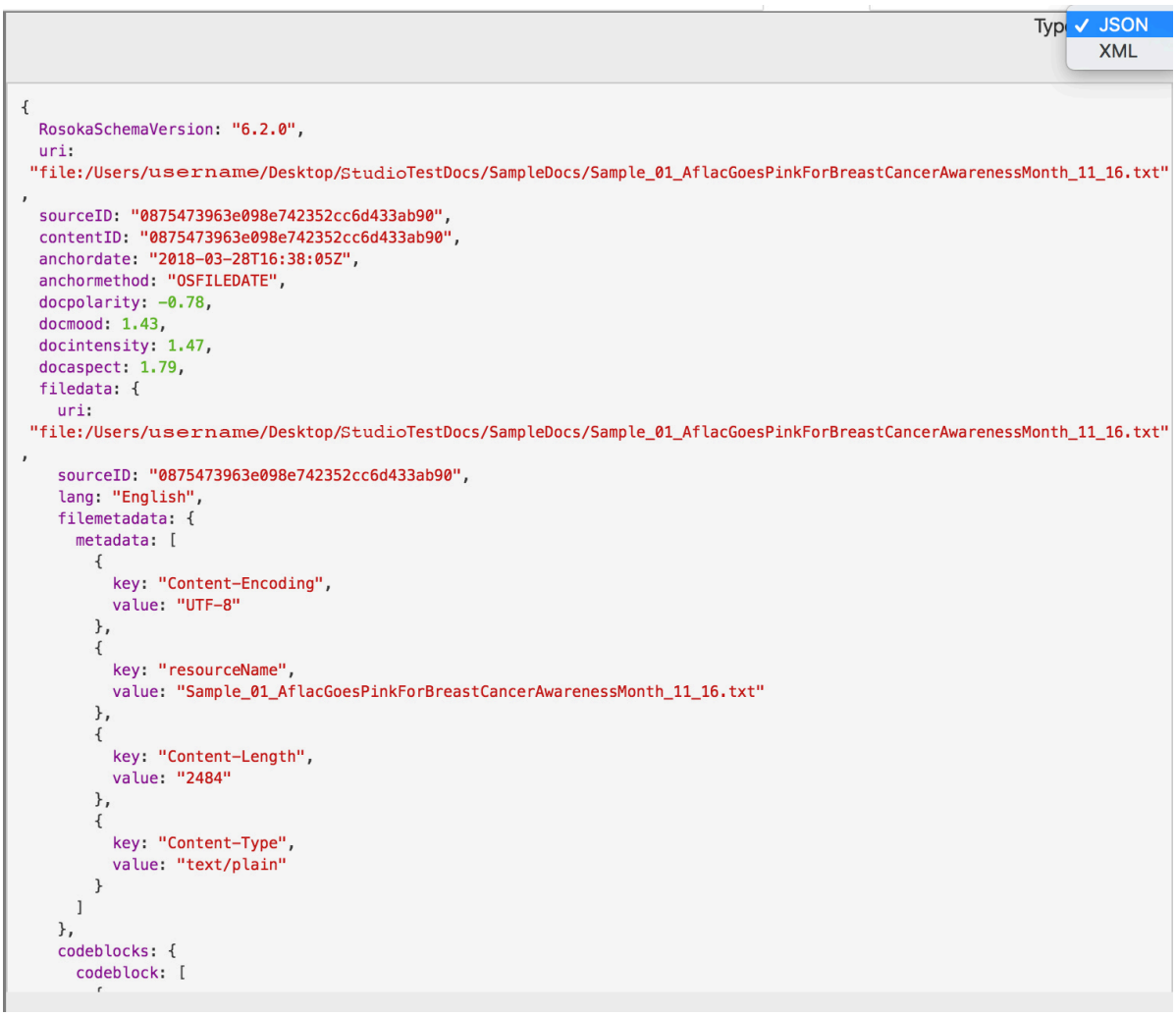
| **Document** | |
|---|---|
| Anchor Date | 2018-03-28T16:38:05Z |
| Anchor Method | OSFILEDATE |
| | |
| Polarity | -0.78 |
| Mood | 1.43 |
| Intensity | 1.47 |
| Aspect | 1.79 |

| **Languages** | |
|---|---|
| English | 100 |
| French | 7 |

| **Codeblock** | **Count** |
|---|---|
| Basic_Latin | 2469 |
| General_Punctuation | 14 |

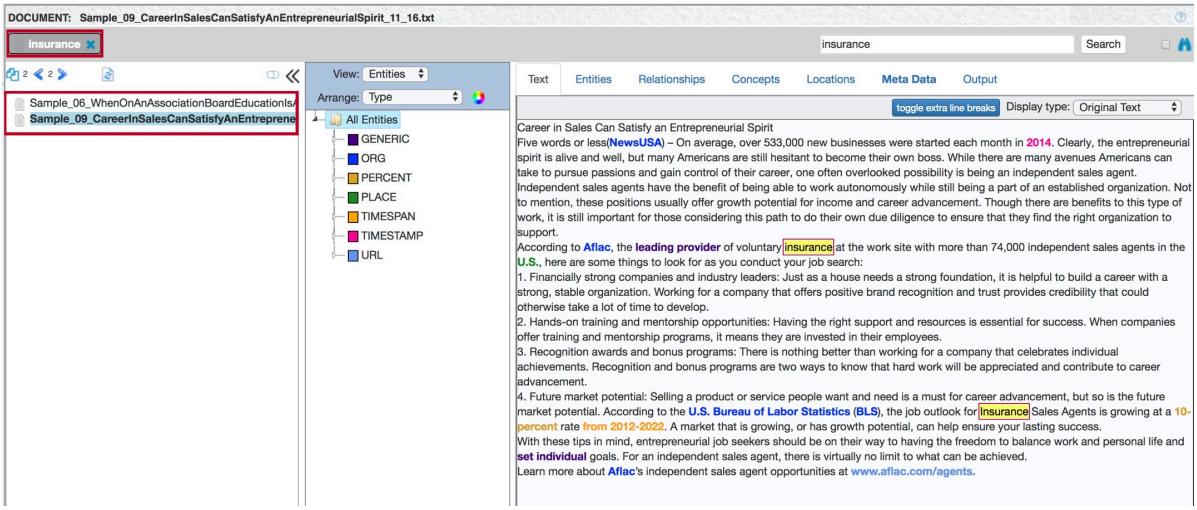
| **Field** | **Value** |
|---|---|
| Content-Encoding | UTF-8 |
| resourceName | Sample_01_AflacGoesPinkForBreastCancerAwarenessMonth_11_16.txt |
| Content-Length | 2484 |
| Content-Type | text/plain |

**13.** The **Output** tab contains the raw output generated from the extraction process for the selected document, in JSON or XML format.




This information is what would be sent to the database if the current LxBase had processed the document in a production environment. You can view the output in its source language or with an English gloss, and get a breakdown of the individual tokens, entities, sentiment scores, and relationship extraction results.

14. Use the **Search** function to find all the documents in the corpus that contain a particular word or phrase, effectively filtering the document list.



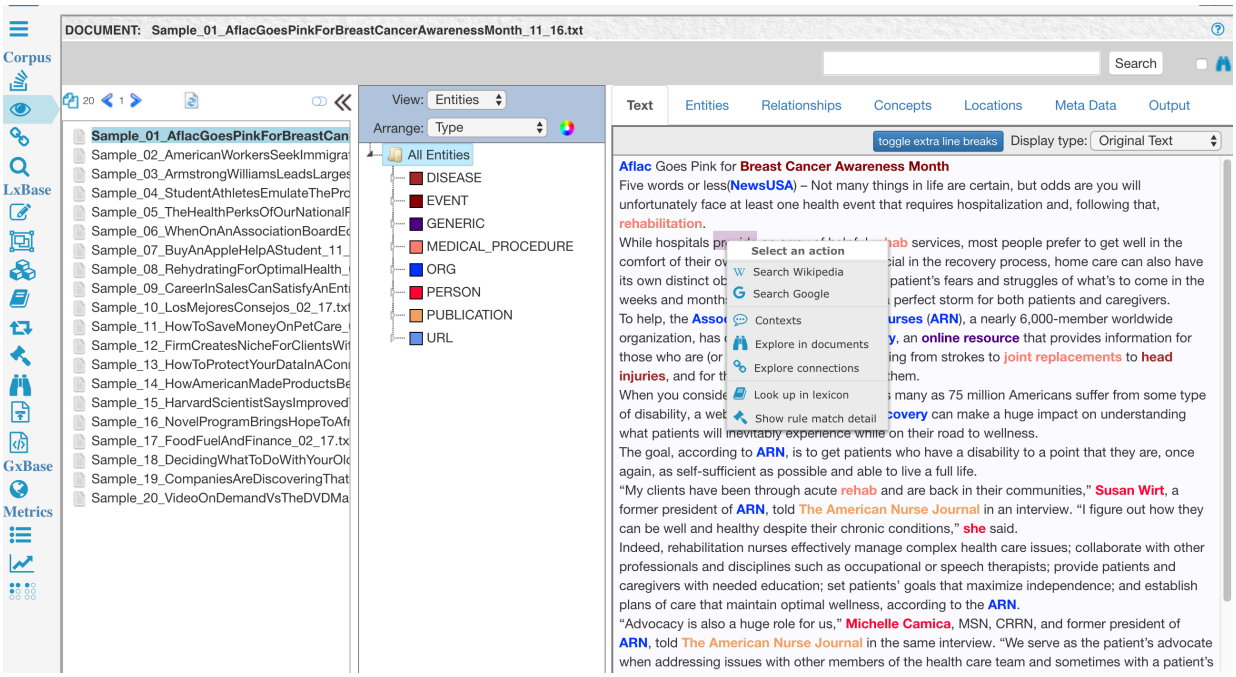

When you select a document from search results, the contents of the **Text** tab highlight not only extracted entities but also the terms that you searched for.

15. When you're viewing a single document, you can use the **Display type** options to view the original text in its source language, with an English gloss, or both.

    In the side-by-side view, hovering over an element on either side highlights the corresponding information in both views.

## Exploring processed text

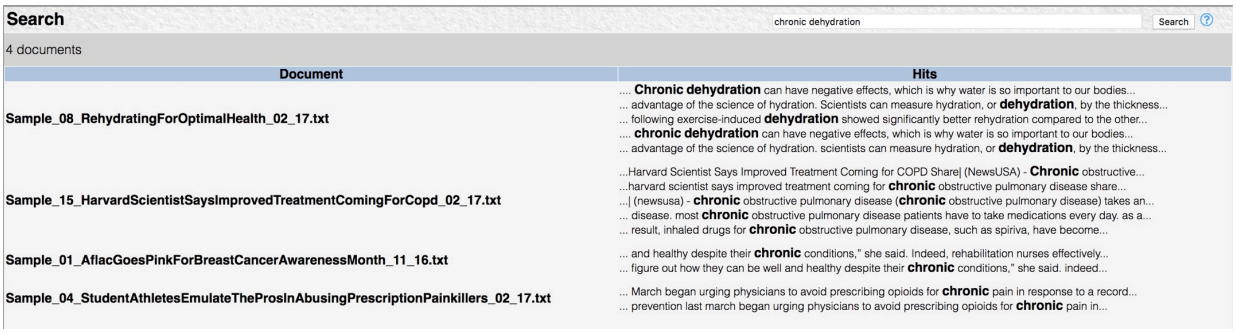
TextChart Studio's **Text** tab allows you to modify the current LxBase directly, by interacting with the highlighted extraction results. Right-click any result to see a pop-up menu with a list of the available options.




Among other things, the menu enables you to search for the result in both Wikipedia and Google. You can also open different views in TextChart Studio, as well as modify the associated semantic vectors.

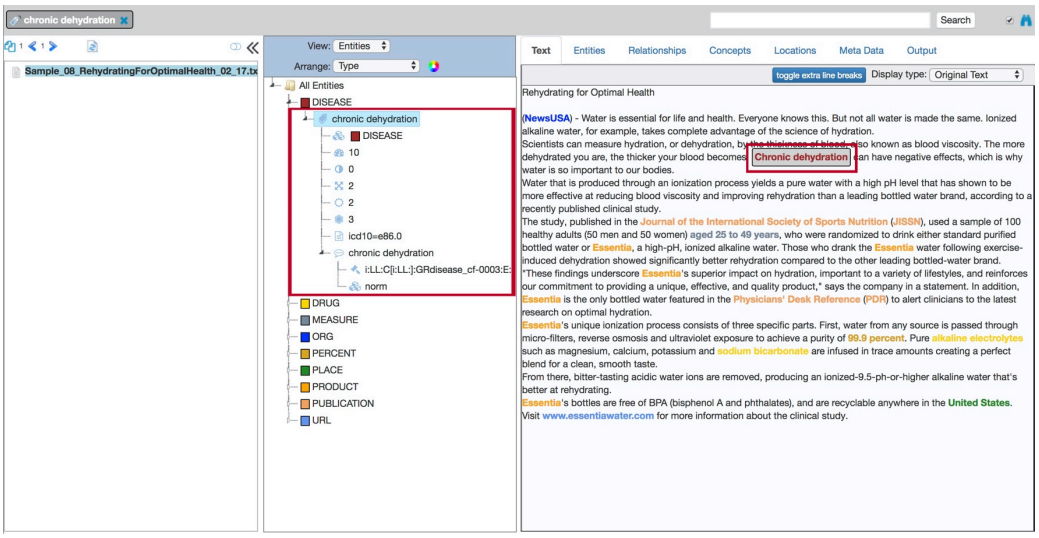
## Document search

Select **Document search** from the pop-up menu to perform a fuzzy search for the extraction result across all documents in the corpus. The results are a list of the documents that matched the search, along with a brief sample of the context in which the result was found.



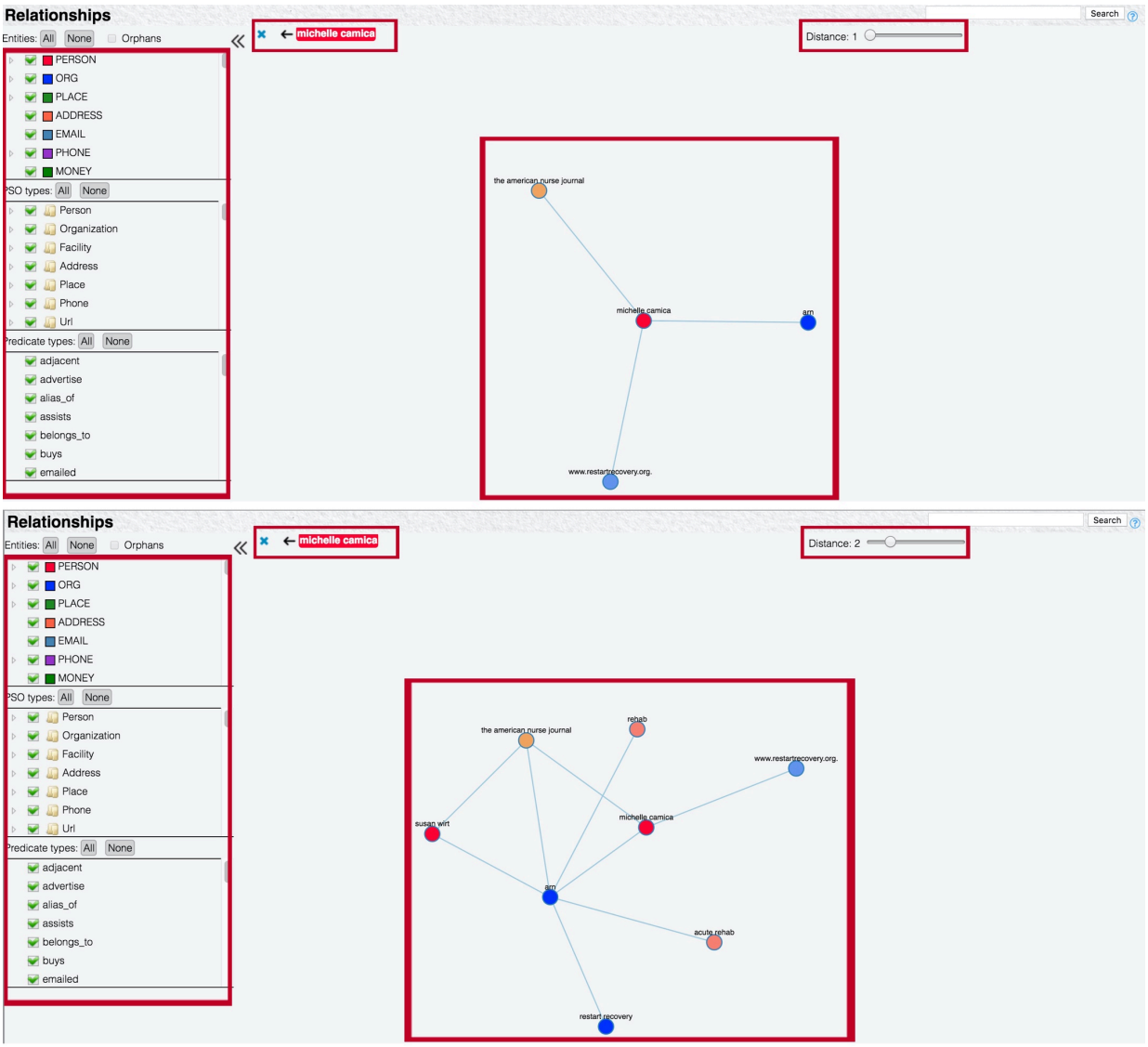

## Explore in documents

Select **Explore in documents** from the pop-up menu to generate a list of all the documents that contain the extraction result. The contents of the **Text** tab change to display the first document in the list, with the first matching result selected.




## Explore connections

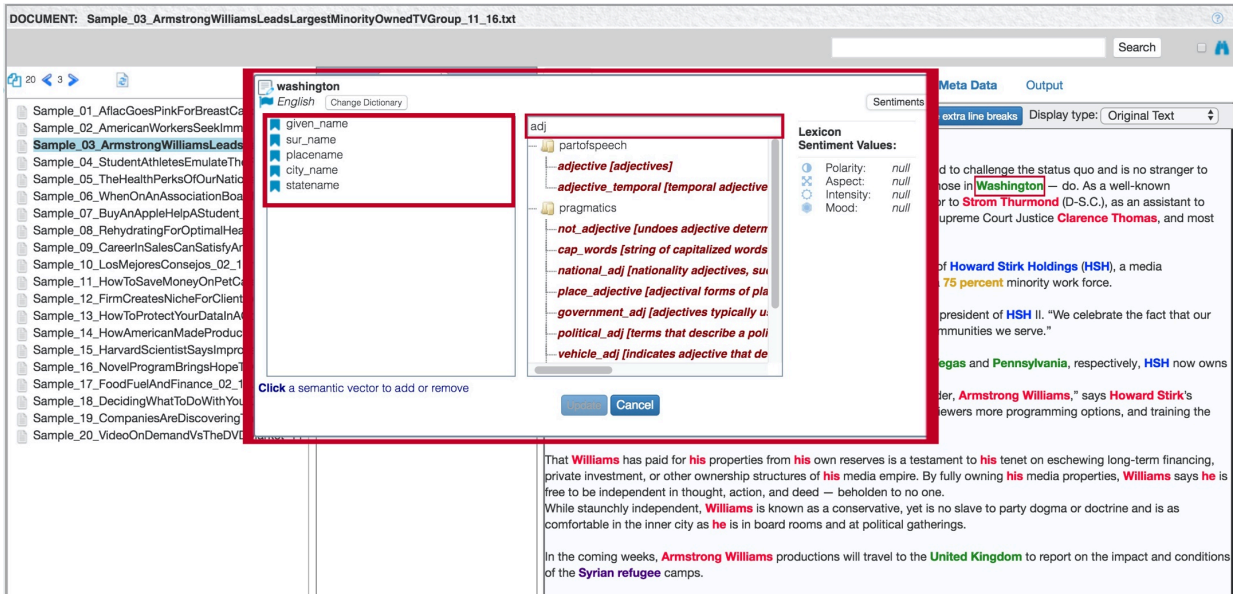
Select **Explore connections** from the pop-up menu to switch to the **Relationships** page, which contains information about the relationships in which the selected entity is involved.




You can adjust the **Distance** setting to display relationships at up to six degrees of separation from the selected entity.

### Look up in lexicon

Select **Look up in lexicon** from the pop-up menu to display a dialog where you can modify the highlighted term.



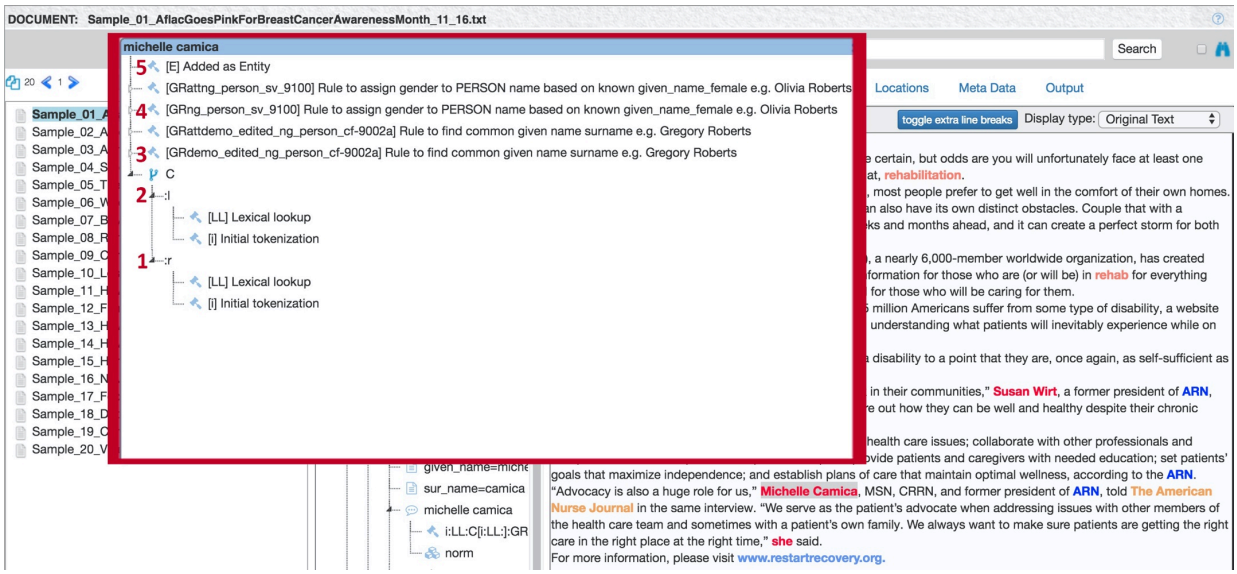

The dialog shows the term, the language of the dictionary that TextChart found it in, and a list of any semantic vectors (SVs) that are associated with it.

To add a new semantic vector to the term, click **Add SV** and then use the box on the right to find the semantic vector that you want to add. Alternatively, click an existing semantic vector to remove it from the term.

When your changes are complete, click **Update** to make the TextChart engine recognize them, and then reprocess either the individual document or the entire corpus.

### Show rule match detail

Select **Show rule match detail** from the pop-up menu to display a dialog that contains the steps the TextChart engine took in order to extract (or not extract) a particular result. For example, to extract the PERSON entity "Michelle Camica", the engine took five steps:

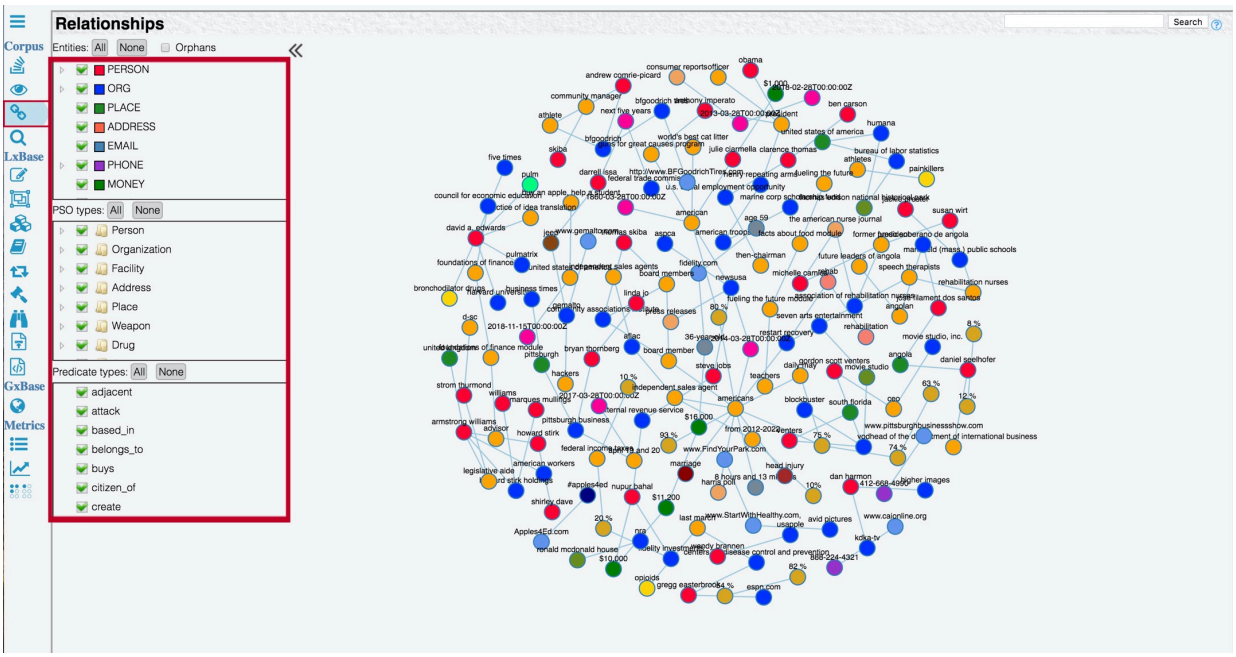



1. The TextChart engine tokenized the term "Michelle" and looked it up in the dictionaries.
2. The TextChart engine tokenized the term "Camica" and looked it up in the dictionaries.
3. The engine executed a linguistic rule whose description contains "...to find common given name and surname..."
4. The engine executed a second linguistic rule whose description contains "...to assign gender to PERSON name based on known given name female..."
5. The engine assigned the two tokens "Michelle" and "Camica" to one PERSON entity, based on the linguistic rules that it executed.

To modify a rule, you can double-click it in the list to open the rule editor.

## Relationships

The **Relationships** page allows you to interact with relationships between the entities in your extraction results. To open the page, click the **View Relationships** button in the vertical toolbar.



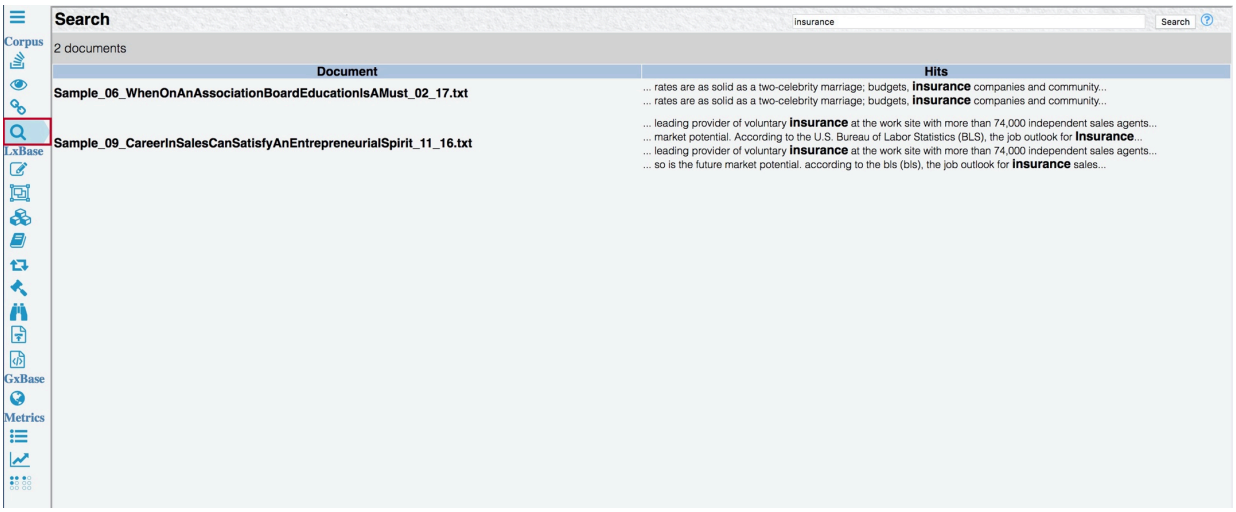

Use the check boxes on the left of the page to select and deselect entity and PSO types in order to customize the visualization.

From a single entity in the visualization, you can step out from first-level connections to see a more complete picture by using the **Distance** setting.

You can also see connections in other views throughout TextChart Studio by right-clicking entities and selecting **Explore connections**.

## Search page

You can use the **Search** page in TextChart Studio to perform a fuzzy search across all the documents in a corpus. To open it, click the **Search** button in the vertical toolbar.




This list of results is presented on the left, with a brief summary of the context surrounding the search result on the right. Click a document in the results to open it in the **Text** tab.

## Token definition editor

TextChart Studio's **Token Definition Editor** page enables you to add or modify semantic vectors in the LxBase. To open the page, click the **Edit Token Definitions** button in the vertical toolbar.

The page presents the current set of semantic vectors in separate tabs according their type.

### Semantic vector types

After processing with TextChart, the terms in a document are associated with one or more semantic vectors. An LxBase defines each semantic vector as one of the following: an *entity*, a *no-output entity*, a *part of speech*, *pragmatic*, or *relational*.

- **Entities**

  A term that's associated with an entity semantic vector such as PERSON, ORG, or PLACE is extracted as an entity and included in the output.

- **No-output entities**

  A term that's associated with a no-output entity semantic vector *is* extracted as an entity but *is not* included in the output.

  By changing a semantic vector from "entity" to "no-output entity", you can prevent entities of the affected type from appearing in the output.

- **Parts of speech**

  A term that's associated with a part-of-speech semantic vector has been identified as a part of speech, such as a verb or an adjective.

- **Pragmatic**

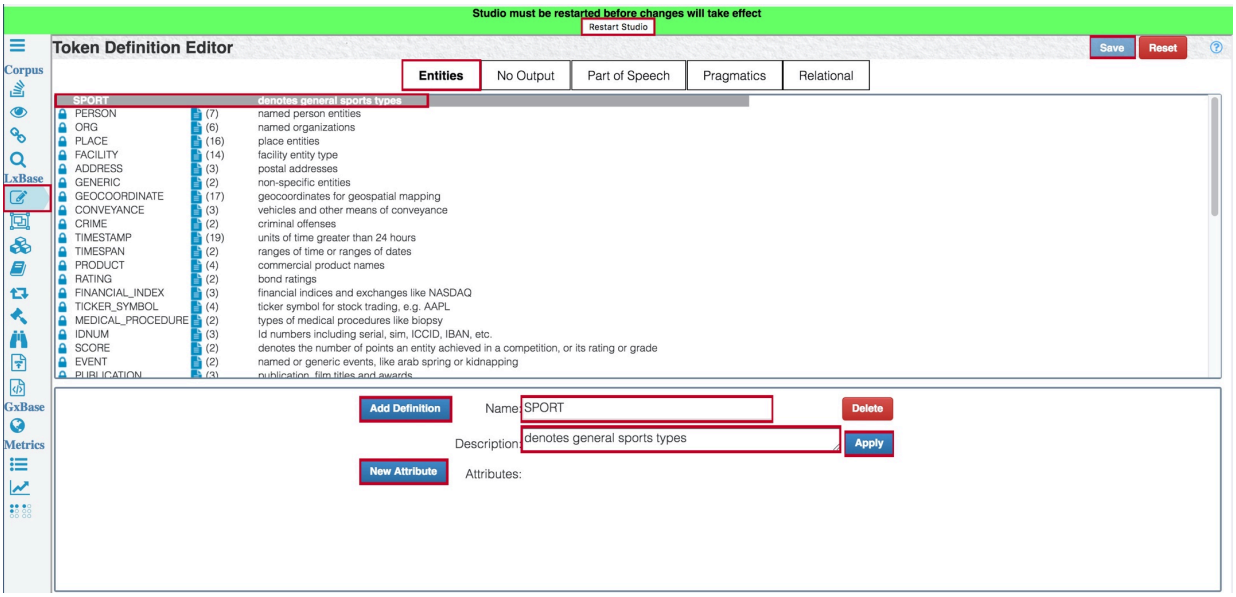
  A term that's associated with a pragmatic semantic vector has been identified as belonging to a pragmatic category such as "given name" or "possible identification number".

- **Relational**

  A term that's associated with a relational semantic vector such as "interviewed" or "identified as" is used as the predicate in a predicate-subject-object (PSO) relationship.

## Editing entity types

To add to the default set of entity types in an LxBase, or to modify one of the existing entity types, you use the **Entities** tab in the token definition editor.

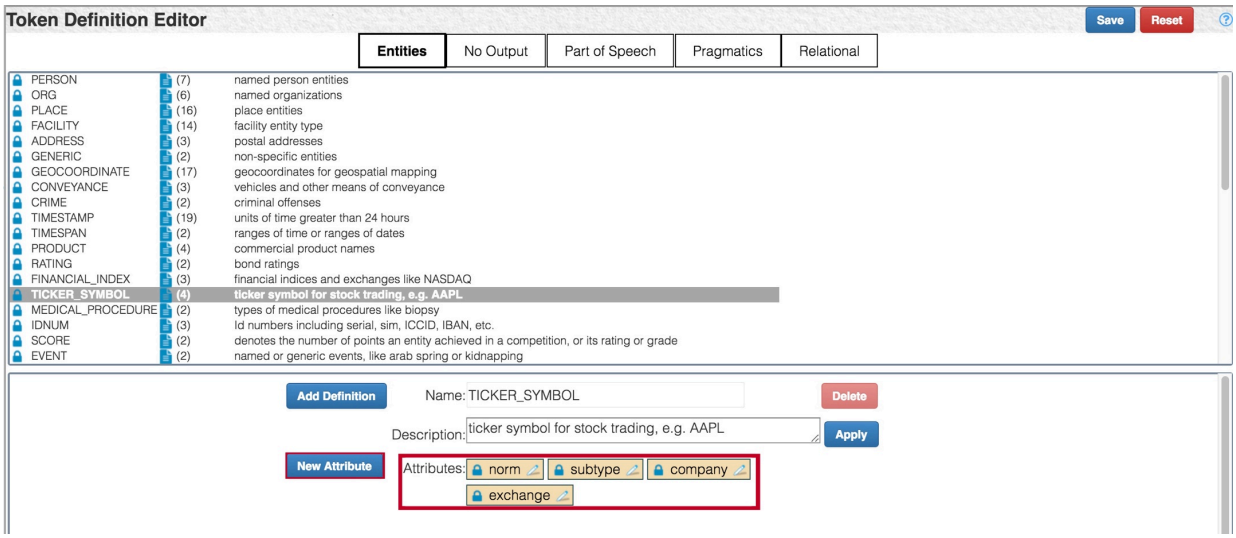



To add a new entity type, click **Add Definition**. To add an attribute to a new or existing entity type, click **New Attribute**.

### Adding and modifying attributes

Clicking **New Attribute** displays a dialog where you can select attributes to add to the entity type. To modify the behavior of an attribute that you've already added - to change its default value, for example - click the pencil icon

next to its name.




**Note:** Semantic vectors and attributes with a lock

next to their name are a core part of TextChart and cannot be deleted. You can, however, add attributes to a locked semantic vector.

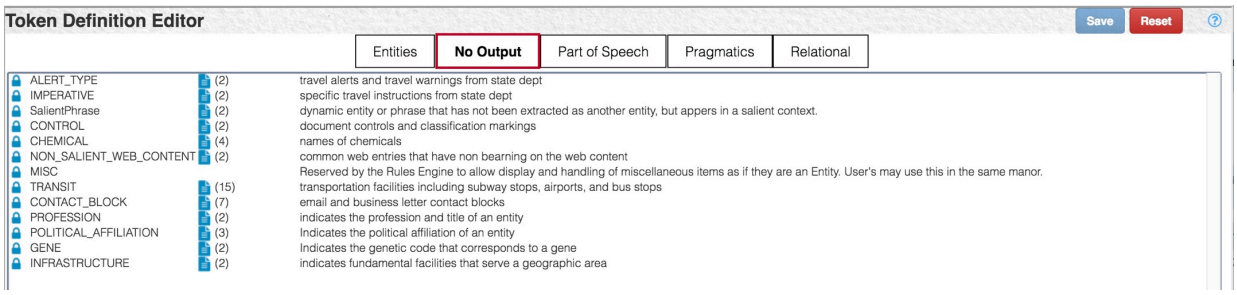
### Updating LxBase

To commit your changes to the LxBase, click **Apply** and then **Save**. Then, since modifying entity types is a schema change, you must restart TextChart Studio, clear any previous processing results from the **Corpus Management** page, and reprocess the corpus.

**Important:** When you add a new entity type, you must associate lexical entries or linguistic rules with it in order for TextChart to extract entities of that type from documents. You should create any additional semantic vectors that will inform that extraction at the same time.

## No-output entity types

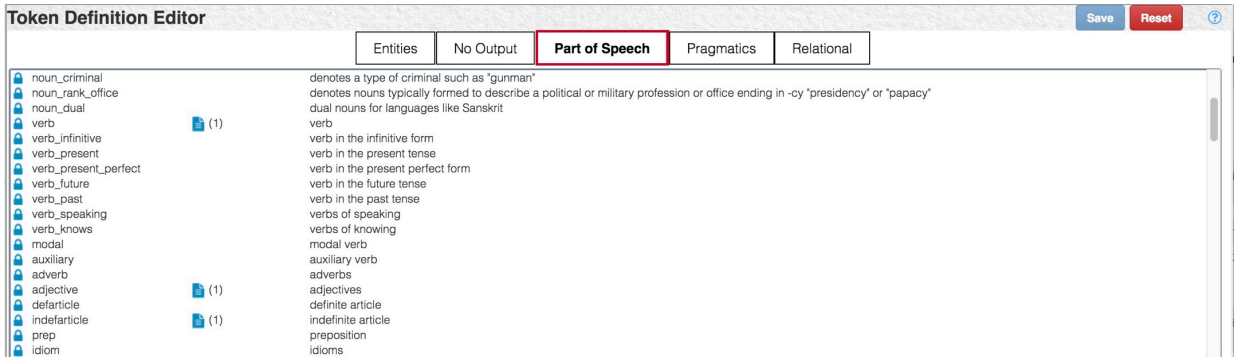
By default, the LxBase is configured to suppress entities of certain entity types. Those types still have linguistic rules and lexical entries associated with them, and TextChart still identifies entities with those types, but they do not appear in extraction results.




If your extraction results contain entities of types that you're not interested in, or if you want entities of suppressed types to appear in your results, you can move entity types between these categories. To do so, drag an entity type from the **Entities** tab to the **No Output** tab, or vice versa.

## Parts of speech

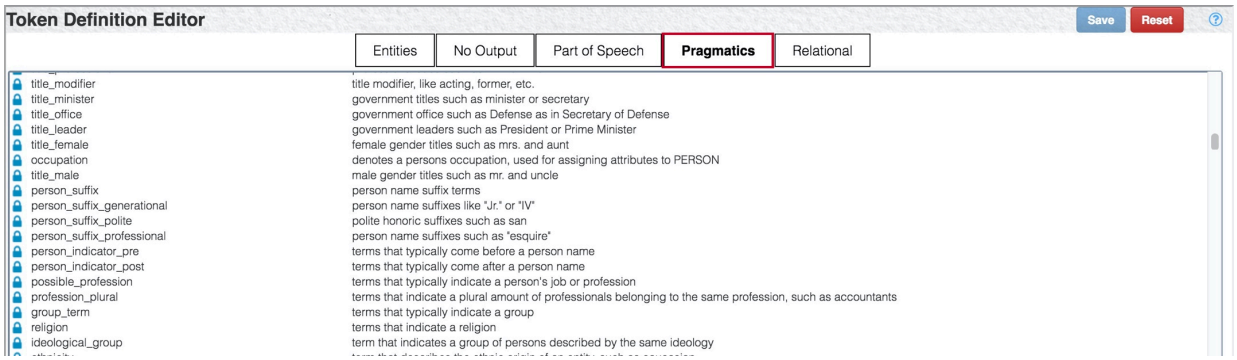
Part-of-speech semantic vectors assign syntactic meaning such as "noun", "verb", and "adjective" to lexical entries. Semantic vectors of this type appear on lexical entries and in linguistic rules to help establish linguistic context.




## Pragmatics

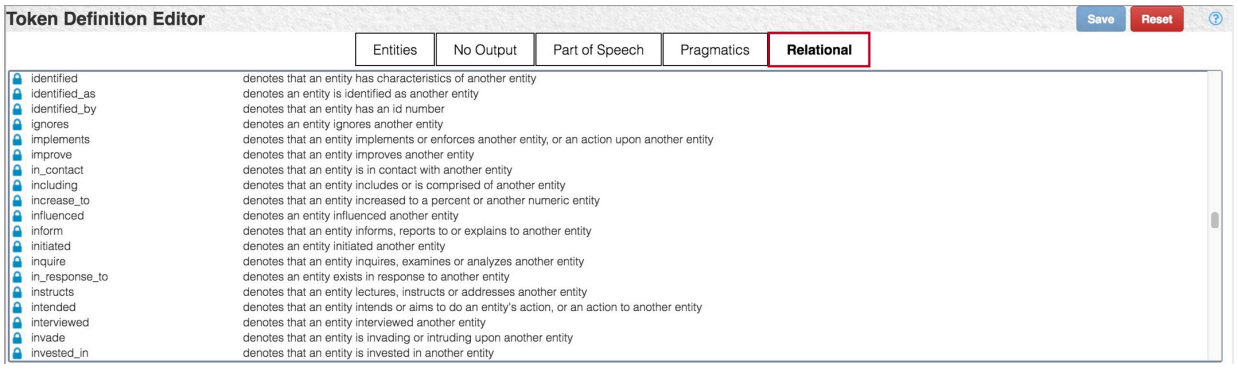
Pragmatic semantic vectors assign meanings like "given name", "city name", or "office title" to lexical entries. As such, they are more specific than their part-of-speech counterparts.

Semantic vectors of this type appear on lexical entries and in linguistic rules to make use of linguistic context and find entity-specific context.




## Relationals

Relational semantic vectors assign predicates like "identified as", "alias of", or "influenced" to a word or a small phrase. Semantic vectors of this type appear on lexical entries, where they're used to prompt extraction of a relationship between two entities.




## Modifying relationships

TextChart extracts relationships between entities through the use of predicate-subject-object (PSO) types that identify those relationships. It uses relational semantic vectors to designate words or small phrases as predicates.

For example, the TextChart dictionaries might associate a relational semantic vector like "interviewed" with a number of lexical entries:

```
<lex><word>interview</word><sv><interviewed/></sv></lex>
<lex><word>interviews</word><sv><interviewed/></sv></lex>
<lex><word>interviewing</word><sv><interviewed/></sv></lex>
<lex><word>interviewed</word><sv><interviewed/></sv></lex>
<lex><word>interrogate</word><sv><interviewed/></sv></lex>
<lex><word>interrogates</word><sv><interviewed/></sv></lex>
<lex><word>interrogating</word><sv><interviewed/></sv></lex>
<lex><word>interrogated</word><sv><interviewed/></sv></lex>
```

If TextChart tags all of these words in a document with the "interviewed" semantic vector, then a relationship will be extracted any time one of them appears as a predicate in context with two entities.

*During **David Frost**'s **interview** of former President **Richard Nixon**, Frost very bluntly asked why he didn't burn the tapes.*

In the example phrase above, both "David Frost" and "Richard Nixon" are extracted as PERSON entities. A PERSON to PERSON relationship is also extracted, because the predicate "interview" is tagged with the "interviewed" relational semantic vector.

In PSO terms, the *subject* of the relationship is "David Frost", the *object* is "Richard Nixon", and the *predicate* is "interview".

## Modifying PSO relationship types

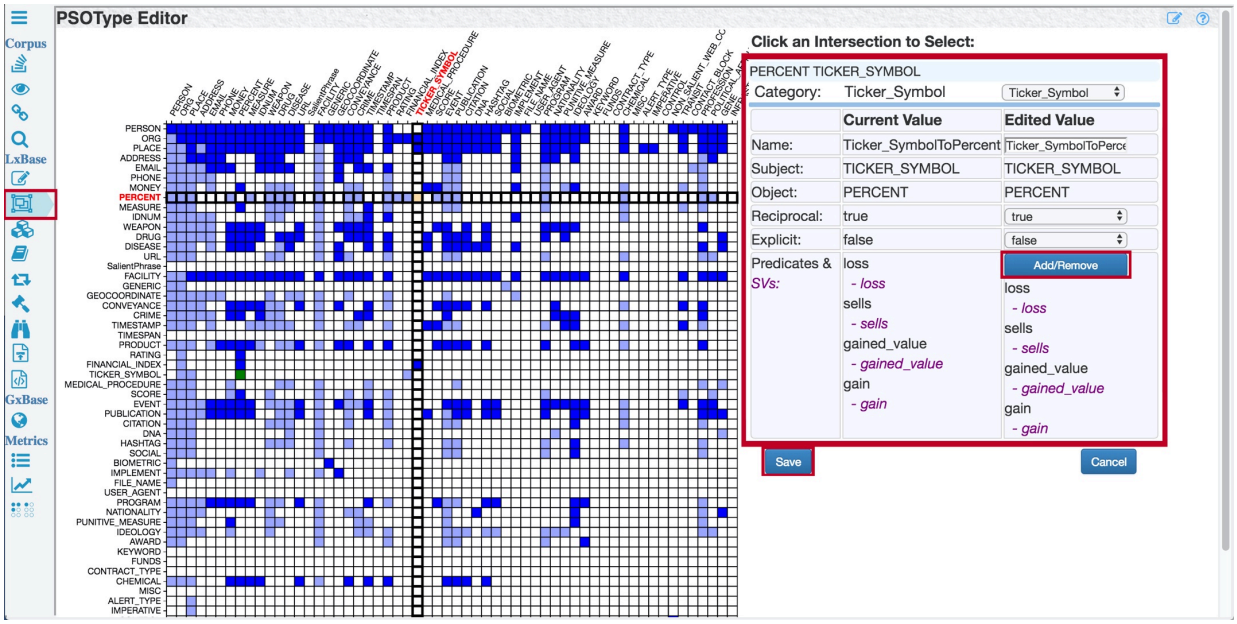
To create or edit PSO relationship types in TextChart Studio, click **Edit PSO types** in the **LxBase** section of the vertical toolbar. You can make modifications through a graphical interface, or by modifying an XML file directly.

The sections below demonstrate using both approaches to add a relational semantic vector named `closed_at` to the list of semantic vectors that are associated with relationships between TICKER_SYMBOL and PERCENT entities.

Before you begin, make sure that the `closed_at` semantic vector is present in the list in the **Relational** tab in TextChart Studio. If it's not there, create it, and then follow either of the following procedures.

### GUI

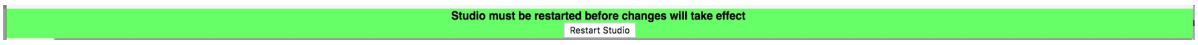
The **PSOType Editor** page displays the graphical user interfaces for modifying relationship types by default.




1. In the grid, click on the intersecting point between TICKER_SYMBOL and PERCENT.

2. In the section on the right, click **Add/Remove** to open the **Predicate Editor** dialog.

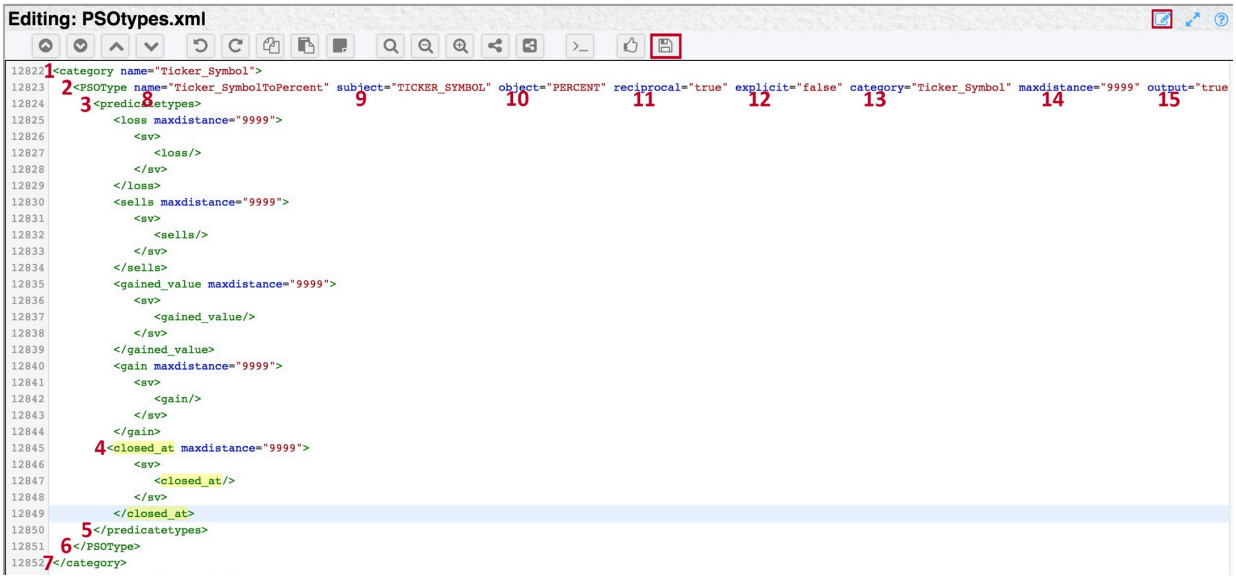

3. In the bar at the top, type a predicate that you'll associate with the new semantic vector, and then click **Add**. In this example, the predicate is also `closed_at`.
4. To associate a semantic vector with the new predicate, click the new **closed_at** button. In the new window on the right, type `closed_at` in the field at the top of the **Semantic Vectors** list. Select `closed_at` from the list.
5. Close all the new windows, and then click **Save**. TextChart Studio displays a message at the top of the application window, prompting a restart. Click **Restart Studio**.




6. Add the `closed_at` semantic vector to relevant entries in the lexicon.

### XML

To modify PSO relationship types directly, click **Edit PSO types in XML Mode** in the top right corner of the **PSOType Editor** window.




Then, you need to find the definition of the relationship type that you want to modify, and add a block of XML like this to it:

```
<users_predicate maxdistance="9999">
  <sv>
    <users_semantic_vector/>
  </sv>
</users_predicate>
```

For example, to add a predicate named `closed_at` (and its associated semantic vector) to the Ticker_SymbolToPercent relationship type, find the `<PSOType>` element for that type in the file, and then add the following before the `</predicatetypes>` tag:

```
<closed_at maxdistance="9999">
  <sv>
    <closed_at/>
  </sv>
</closed_at>
```

After you edit the file, click the **Save** button, and restart TextChart Studio.

### Creating a PSO relationship type

To create a relationship type when a category for one of the entity types involved is not present in the XML file, you need to add a complete `<category>` element.

For example, to create a type for relationships between TICKER_SYMBOL and PERCENT entities in the case where neither has its own category, you'd need to add (and modify) the following code:

```
<category name="Entity_Type1">
  <PSOType name="Entity_Type1ToEntity_Type2" subject="Entity_Type1"
           object="Entity_Type2" reciprocal="true" output="true"
           explicit="false" category="Entity_Type1">
    <predicatetypes>
      <users_predicate maxdistance="9999">
        <sv>
```

```
          <users_semantic_vector/>
        </sv>
      </users_predicate>
    </predicatetypes>
  </PSOType>
</category>
```

On the <PSOType> element, the `reciprocal` attribute is `"true"` by default, which means that in our example, TextChart extracts both TICKER_SYMBOL to PERCENT relationships and PERCENT to TICKER_SYMBOL relationships. Setting it to `"false"` means extracting only TICKER_SYMBOL to PERCENT relationships.

Conversely, the `explicit` attribute is `"false"` by default, which makes TextChart extract relationships that are explicitly defined as TICKER_SYMBOL to PERCENT, *and* other relationships that involve entities of those types, as in both examples below:

1. AAPL **closed at** a 3% increase yesterday.
2. AAPL, +3%, 11-01-2018.

Setting the attribute to `"false"` means extracting only explicitly TICKER_SYMBOL to PERCENT relationships, as in the first example above.

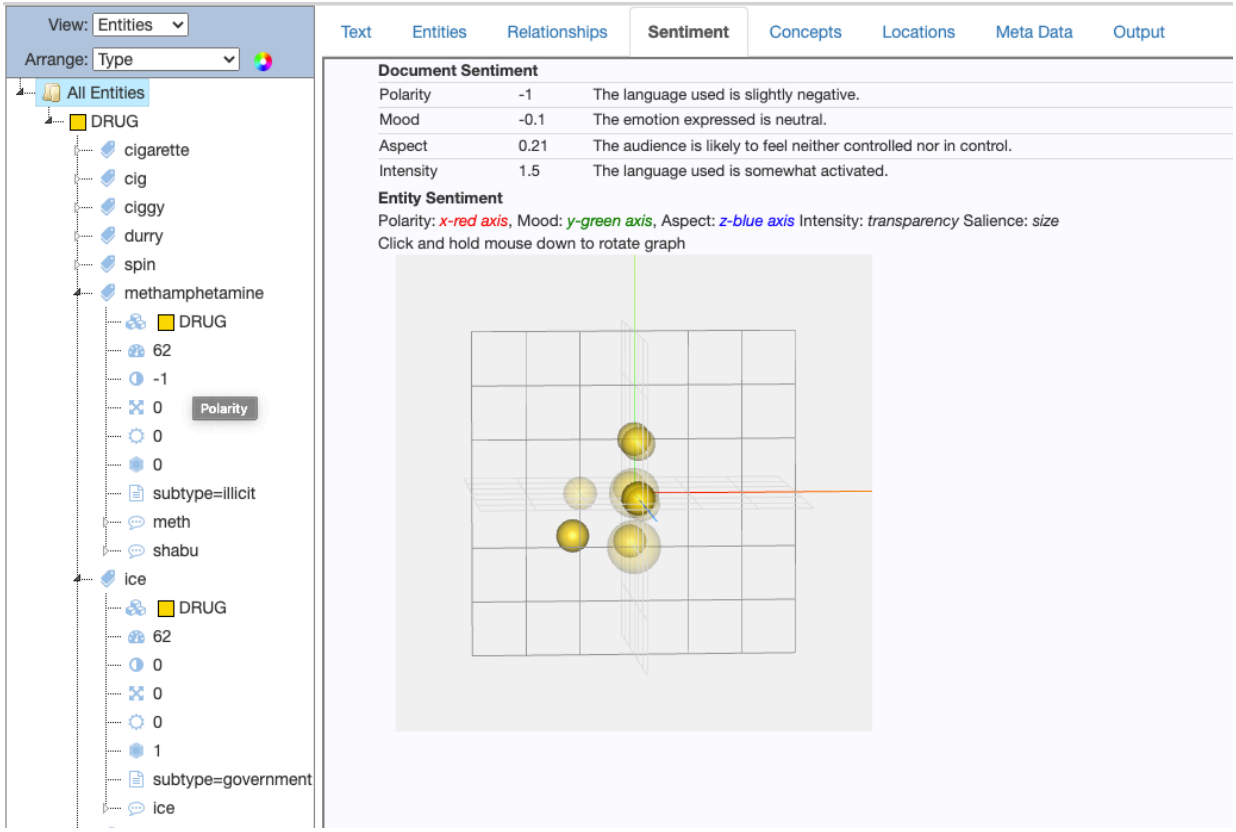
Finally, setting the `output` attribute to `"true"` means that the result of extracting this relationship will be output.

## Modifying sentiment scores

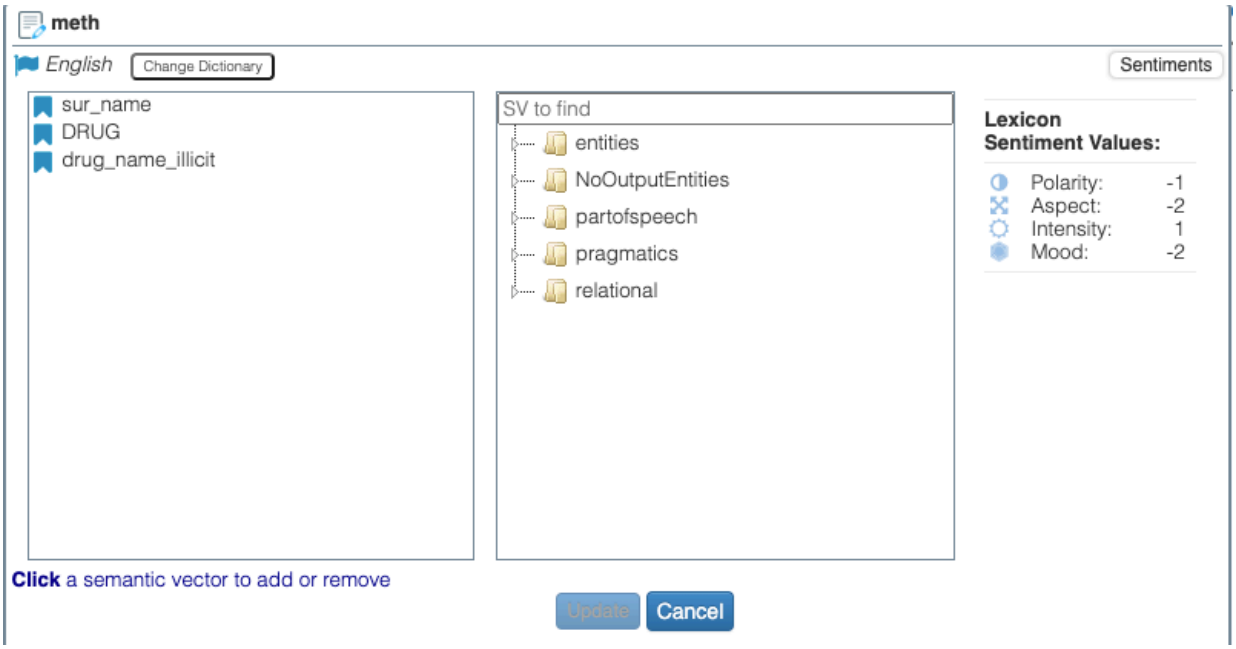
i2 TextChart performs sentiment analysis by assigning scores to the polarity, mood, aspect, and intensity of each dictionary entry. You can use TextChart Studio to modify the sentiment scores of individual dictionary entries manually.

In the following example, a list of slang terms for DRUG entities has been added to a new dictionary file using the word import tool.

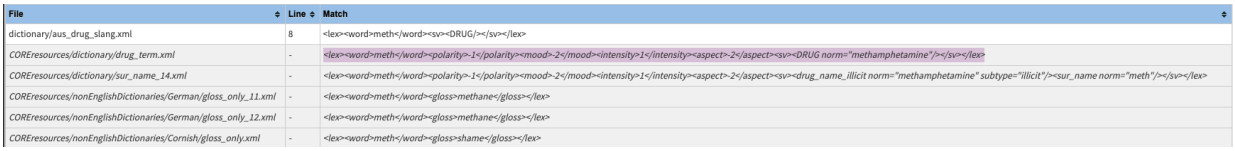
The next image shows the **Sentiment** tab displaying results for these DRUG terms, as well as the individual sentiment results for "methamphetamine" and "ice." In this case, "methamphetamine" has a negative polarity score out-of-the-box, while "ice" does not have a sentiment score.




To see what out-of-the-box information is associated with a particular term (and to view the XML of the dictionary entry), right-click its name in the document view and select **Look up in lexicon** to display the following dialog.

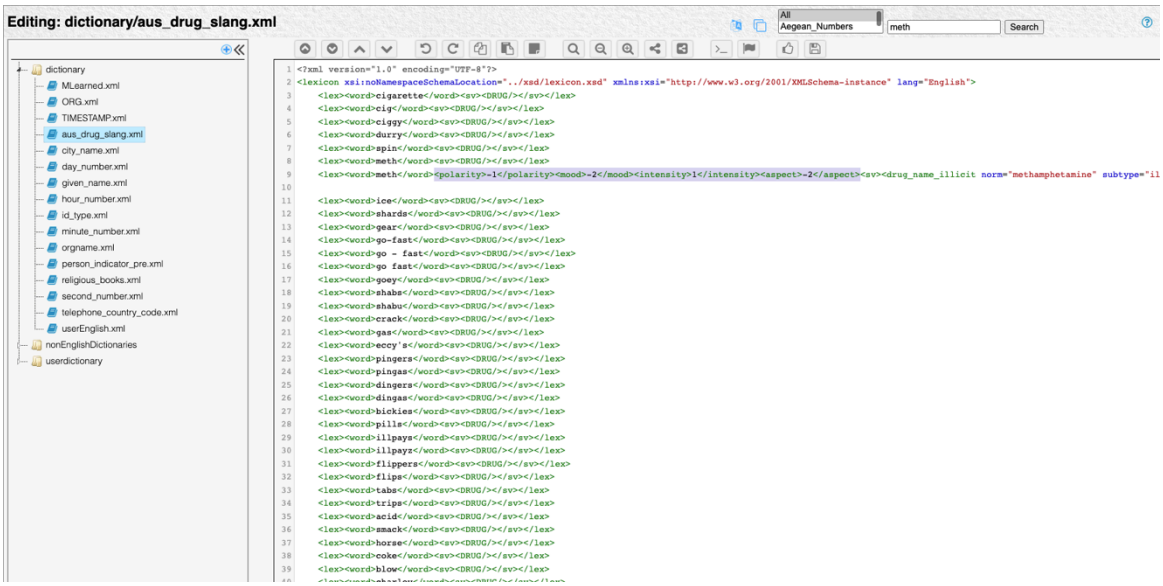



Click the "Notepad" icon in the upper left corner of the the dialog to display the individual dictionary entries associated with the selected result. Then, you can simply copy the relevant information. In this example, the XML for "meth" has been highlighted and copied.

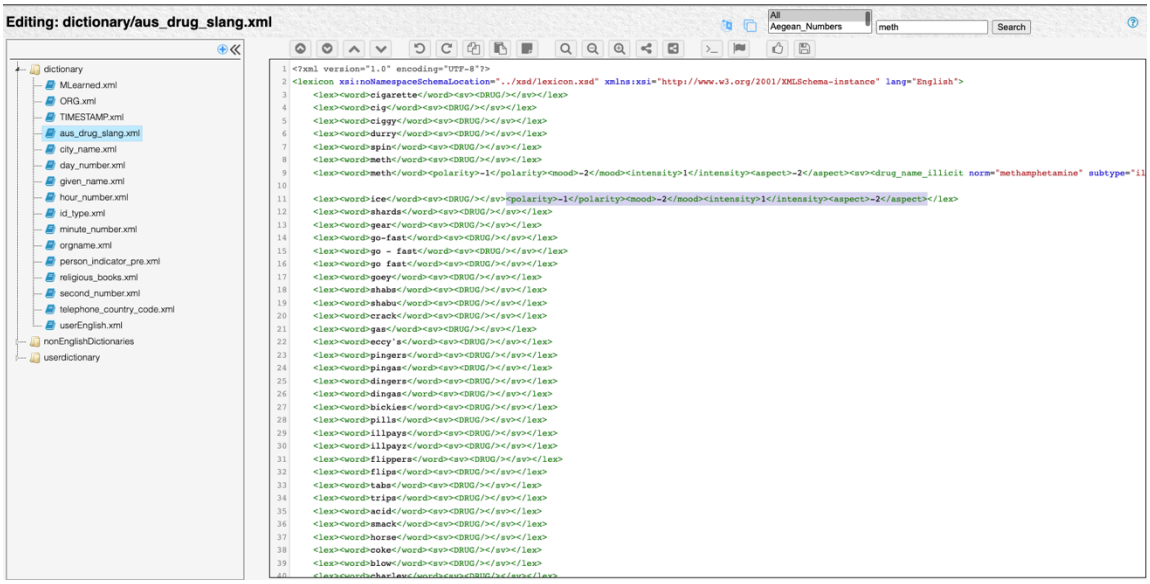


| File | Line | Match |
|---|---|---|
| dictionary/aus_drug_slang.xml | 8 | <lex><word>meth</word><sv><DRUG/></sv></lex> |
| CORE resources/dictionary/drug_term.xml | - | <lex><word>meth</word><polarity>-1</polarity><mood>-2</mood><intensity>1</intensity><aspect>-2</aspect><sv><DRUG norm="methamphetamine"/></sv></lex> |
| CORE resources/dictionary/sur_name_14.xml | - | <lex><word>meth</word><polarity>-1</polarity><mood>-2</mood><intensity>1</intensity><aspect>-2</aspect><sv><drug_name_illicit norm="methamphetamine" subtype="illicit"/><sur_name norm="meth"/></sv></lex> |
| CORE resources/nonEnglishDictionaries/German/gloss_only_11.xml | - | <lex><word>meth</word><gloss>methane</gloss></lex> |
| CORE resources/nonEnglishDictionaries/German/gloss_only_12.xml | - | <lex><word>meth</word><gloss>methane</gloss></lex> |
| CORE resources/nonEnglishDictionaries/Cornish/gloss_only.xml | - | <lex><word>meth</word><gloss>shame</gloss></lex> |

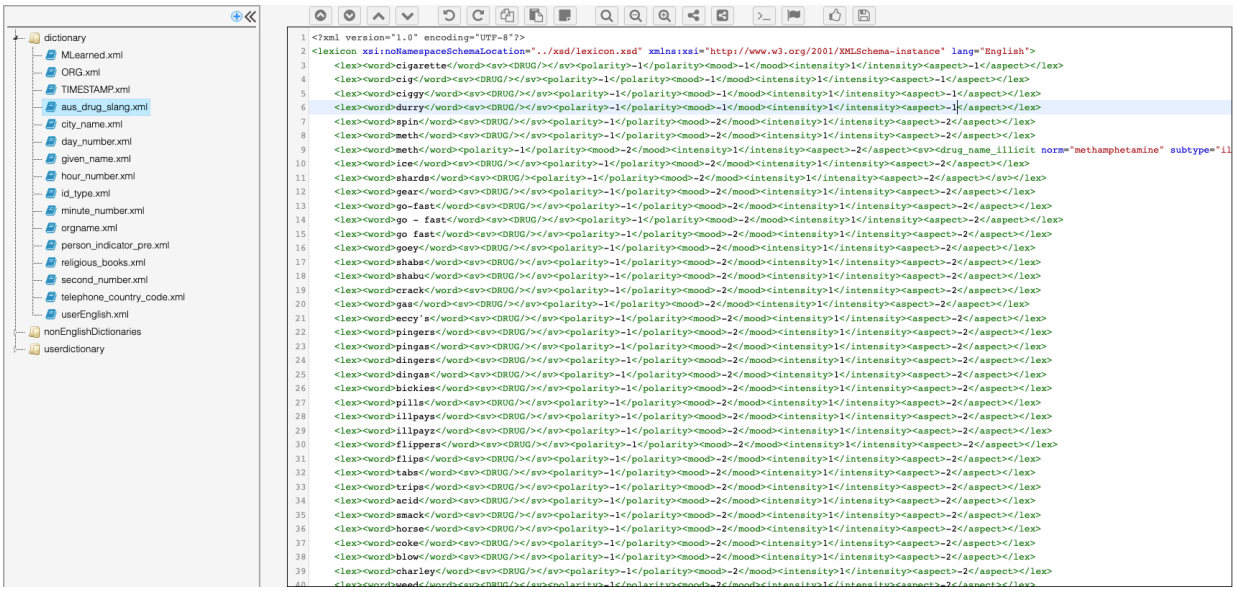
Now you can navigate to another dictionary file and paste the XML where you choose. In this example, the XML for "meth" has been pasted into the drug slang dictionary.



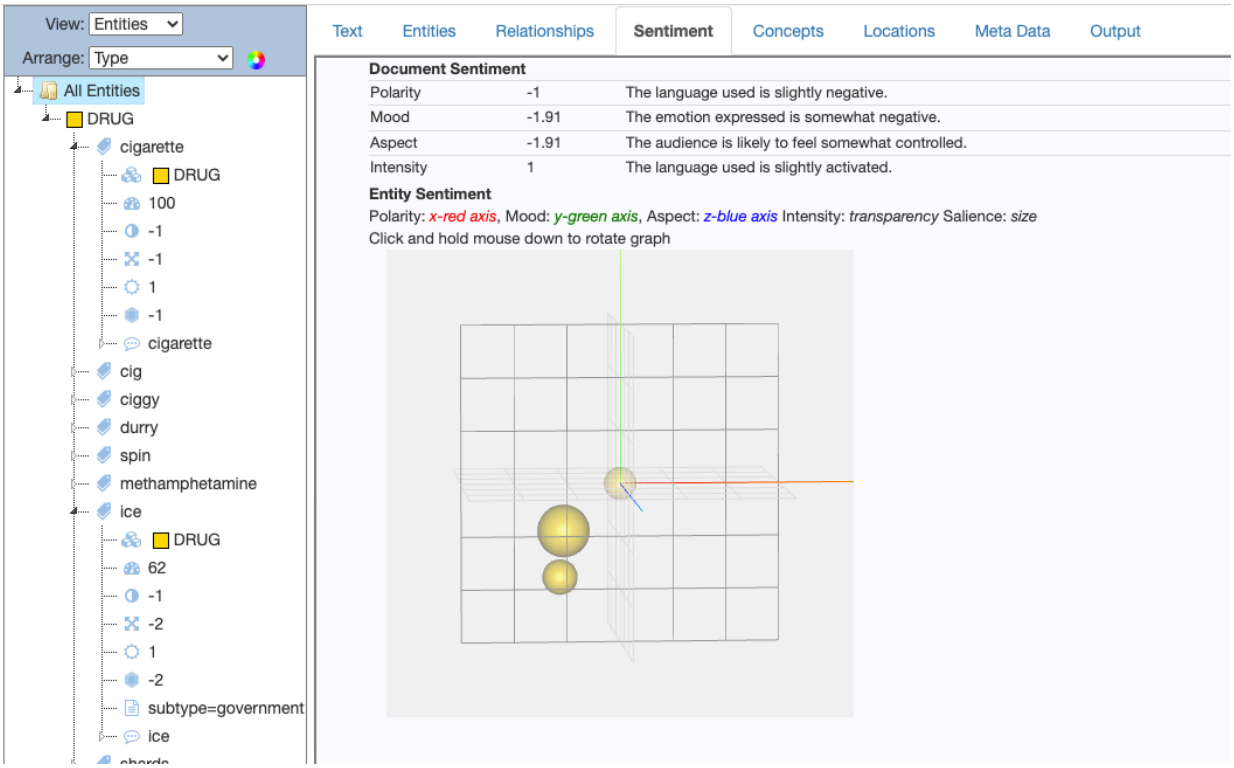

You can add the same sentiment scores to other terms in the same file just by copying the XML from one term to another.




In the next example, all of the drug entries are associated with the same scores, except for the tobacco-type terms, which have less intense scores.




After you reprocess the document, the scores in the **Sentiment** tab are different, as are the scores for individual DRUG entity results.

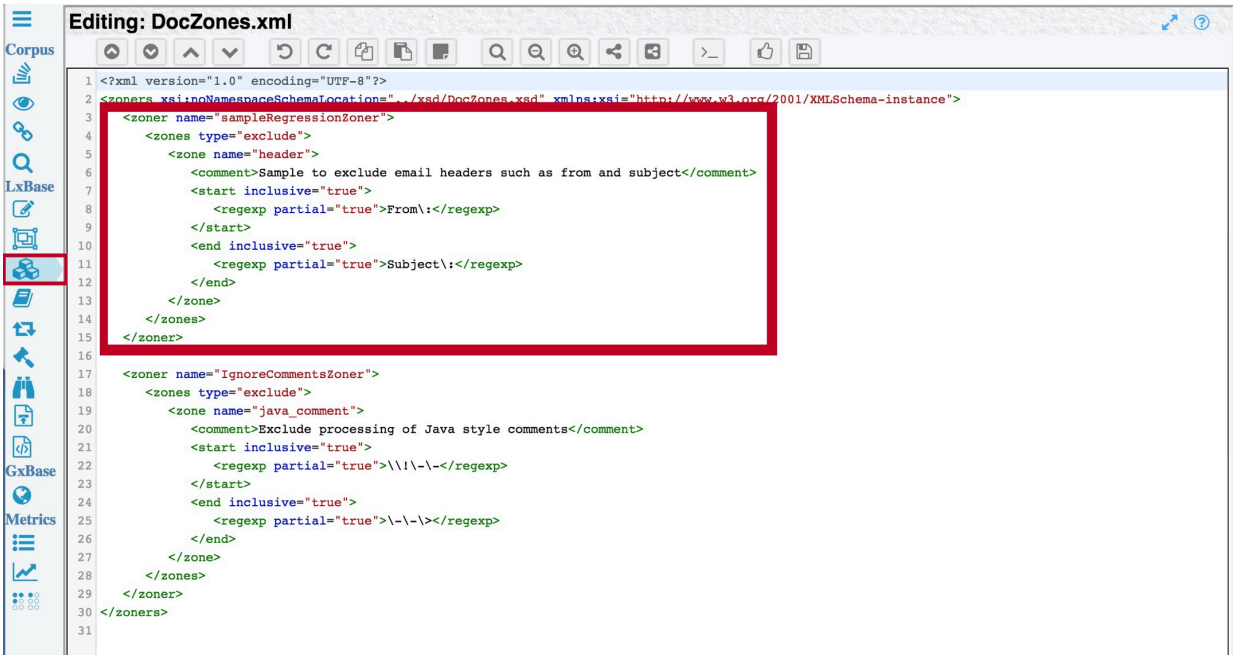


## Document zoning

If the corpora that you process with TextChart regularly contain documents with sections that you want to skip, you can configure LxBase to ignore them. TextChart Studio enables this behavior through *document zoning*.

In zoning, you provide the definitions of some text that marks the beginning and the end of zones, and say whether you want to include those zones in processing. These definitions appear in an XML file that you can edit through TextChart Studio.

To view or modify the document zoning XML file, click **Configure document zoning** in the **LxBase** section of the vertical toolbar.




The default zoning file contains the definitions of two *zoners*: collections of zone definitions that have similar aims. In this file, each zoner contains a single zone.

The zone definitions themselves contain regular expressions describing the text that starts and ends the document zones that you want to control processing for.

The first zone definition in the file demonstrates a way of ignoring some email headers during processing. As it appears above, the zone is excluded from processing by a setting in its enclosing `<zones>` element:

```
<zones type="exclude">
```

To include the headers in processing, you can permanently delete the zone definition, or temporarily edit the element:

```
<zones type="include">
```

For the `<start>` and `<end>` elements that define the start and end of document zones, you can use the `inclusive` attribute to say whether the text that matched the regular expression is a part of the zone. `inclusive="true"` means that it is; `inclusive="false"` means the opposite.

For the regular expressions, you can specify whether a match effectively selects only the matching text (`<regexp partial="true">`), or the whole line that contains the matching text (`<regexp partial="false">`). In other words, you control whether the document zone can start or end in the middle of a line, or if it always includes whole lines.

### XML zoners

If your corpora include documents in XML format, then you can use zoners to target specific elements for inclusion and exclusion from processing, instead of using regular expressions.

You define XML zoners in the same file as the zoners that use regular expressions, although only one zoner can be active at a time.

To create an XML zoner, use `<includedXml>` and `<excludedXml>` elements in place of the `<zones>` element in the `<zoner>` definition:

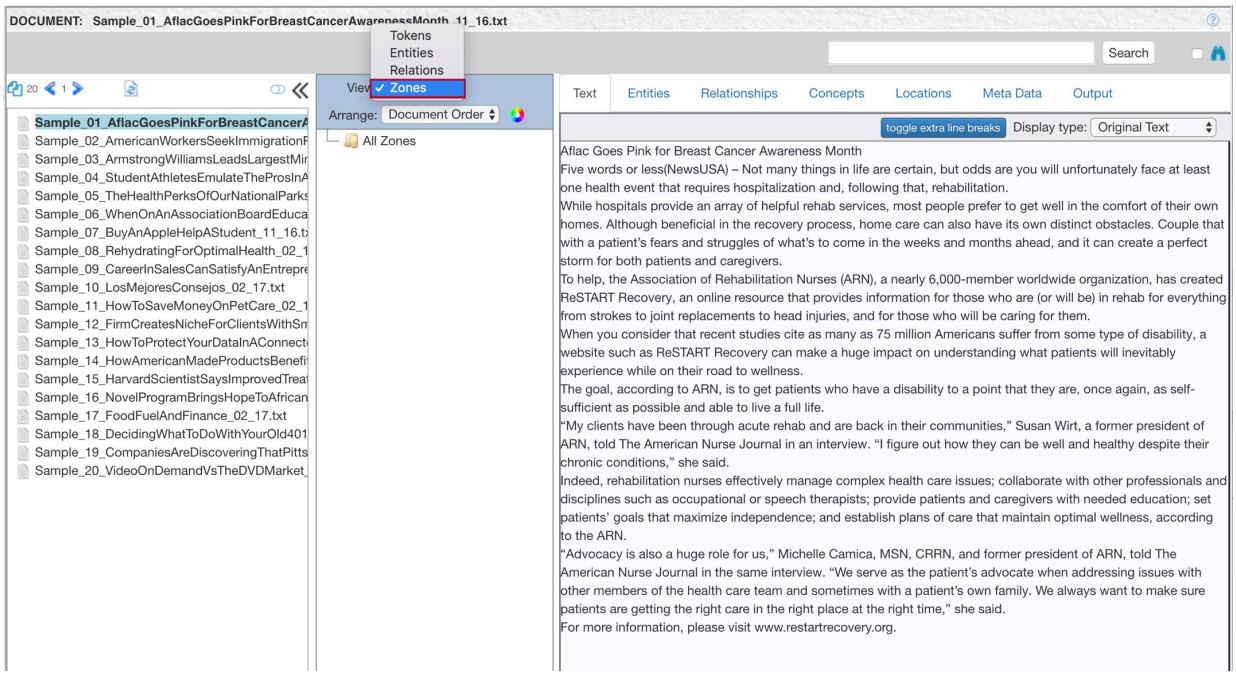
```
<zoner name="datastream">
  <includedXml>
    <tag include="false">legis-body</tag>
  </includedXml>
  <excludedXml>
    <tag>section</tag>
  </excludedXml>
</zoner>
```

With this zoner definition, TextChart processing creates document zones for the `<legis-body>` elements in an XML document. However, if a zone contains a `<section>` element, then its contents are excluded, effectively splitting the zone into two or more pieces.

The `include` attribute of the `<tag>` element in the definition controls whether the opening and closing tags of the specified element are included in the zone. In general, you'll use `include="false"` for inclusion, but `include="true"` for exclusion.

For the XML zoner to work, it needs XML. For the moment, set the LxProperties item "rawinput" to "true" to avoid Tika processing.

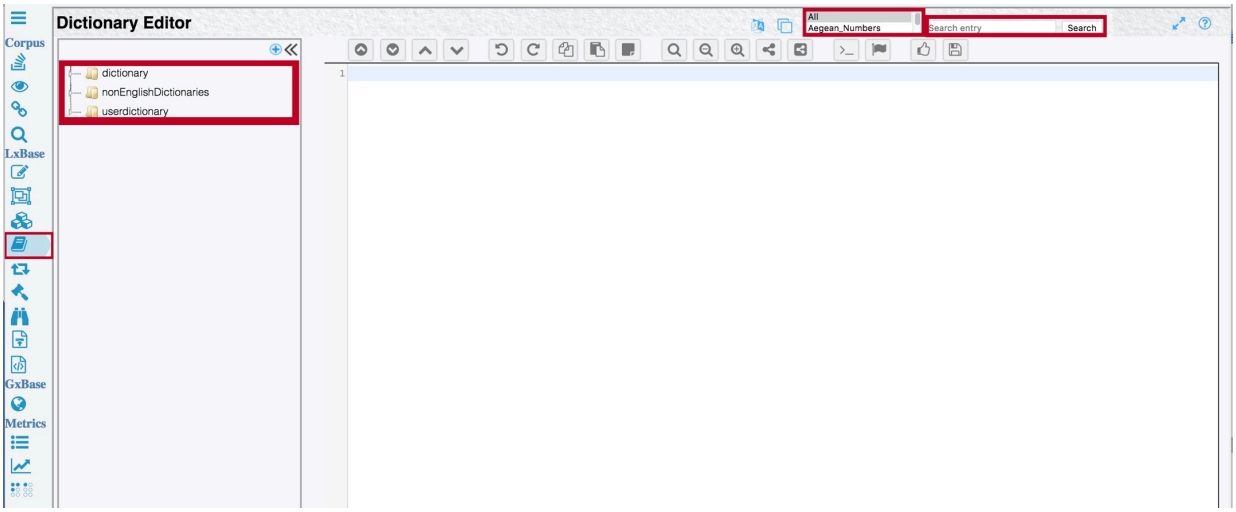
To view the parts of a document that match a particular zone definition, use the **Zones** setting in the single document view.




## Modifying lexicons

In TextChart studio, you can modify lexical items directly in the lexicon, or from the document view. To open the dictionary editor, click **Edit lexicon** in the **LxBase** section of the vertical toolbar.

The **Dictionary Editor** page includes a search box at the top for searching in specific dictionaries or the lexicon as a whole. Lexical items that are not in the core can be modified from this view.

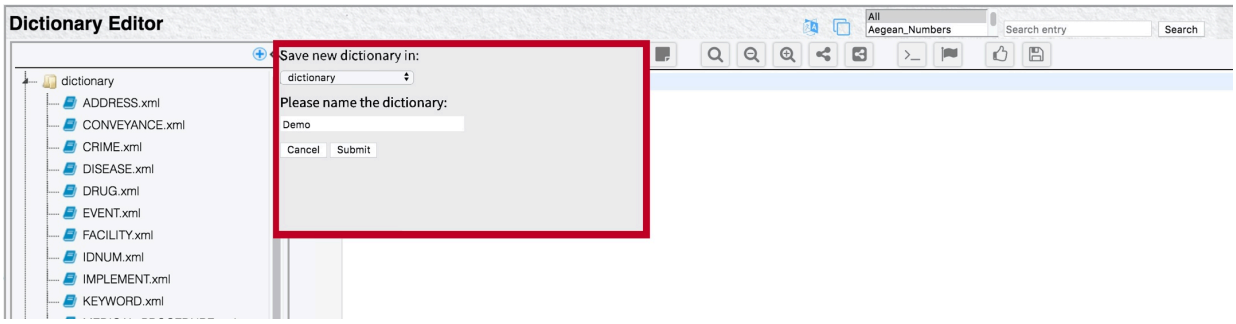


The toolbar below the search box provides standard features to assist with navigating and customizing dictionaries. The buttons next to the search box enable you to enable or disable lexical entries, and to check for duplicate entries across multiple dictionaries.

## Creating dictionaries

i2 TextChart can only extract information from documents when they contain words and terms that appear in its dictionaries. If your organization regularly uses industry-specific terms that are not present in the standard TextChart dictionaries, you can create your own dictionary and add those terms to it.

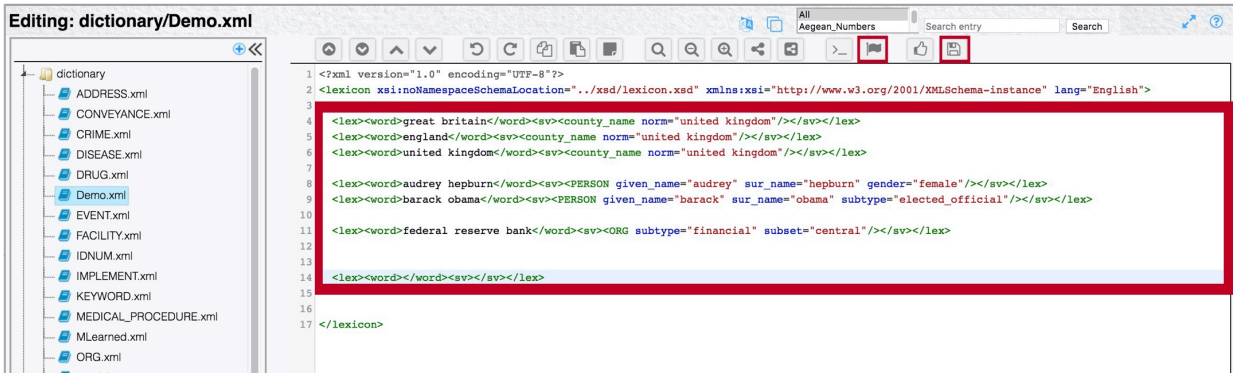
To create a dictionary, clicking the "plus" icon above the tree view in the dictionary editor. TextChart Studio displays a dialog where you can provide the name and location of the new dictionary.




To add a lexical entry to the new dictionary, click the "flag" button in the toolbar to generate an XML template.

```
<lex><word></word><sv></sv></lex>
```

As well as the term itself, which you type in the `<word>` element, there are additional attributes that you can use to enrich your extraction results.




For example, if you wanted to normalize processing so that all references in documents to "England", "Great Britain", or "United Kingdom" resolve to just "United Kingdom", you can use the `norm` attribute on the `country_name` semantic vector:

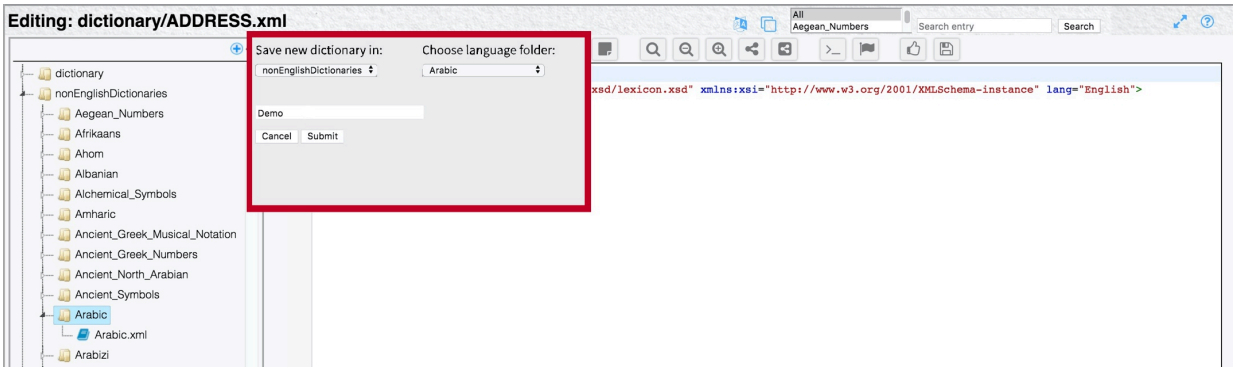
```
<lex><word>great britain</word><sv><country_name norm="united kingdom"/></
sv></lex>
```

```
<lex><word>england</word><sv><country_name norm="united kingdom"/></sv></
lex>

<lex><word>united kingdom</word><sv><country_name norm="united kingdom"/></
sv></lex>
```

Alternatively, you can configure lexical entries so that terms to be extracted as PERSON entities always produce results with the same form. For example, to extract "Audrey Hepburn" as "Audrey Hepburn" and not "V. Audrey Hepburn", or to arrange that "Barack Obama" always has the subtype "elected official":

```
<lex><word>audrey hepburn</word><sv><PERSON given_name="audrey"
 sur_name="hepburn" gender="female"/></sv></lex>

<lex><word>barack obama</word><sv><PERSON given_name="barack"
 sur_name="obama" subtype="elected_official"/></sv></lex>
```

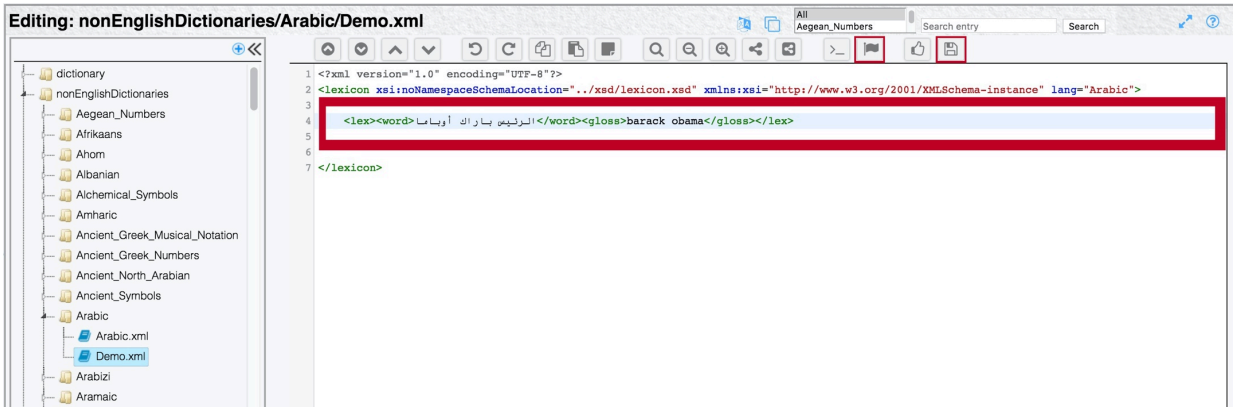
You can also create non-English, industry-specific dictionaries. When you click the "plus" icon above the tree view in the dictionary editor, TextChart Studio displays a dialog where you can select the appropriate language for your dictionary.




You can industry-specific knowledge to any available non-English dictionary. To do so, click the "flag" button in the toolbar to generate an XML template:

```
<lex><word></word><gloss></gloss></lex>
```

To add an entry to a non-English dictionary, add the term (in the dictionary's language) to the `<word>` element, and the English gloss to the `<gloss>` element.

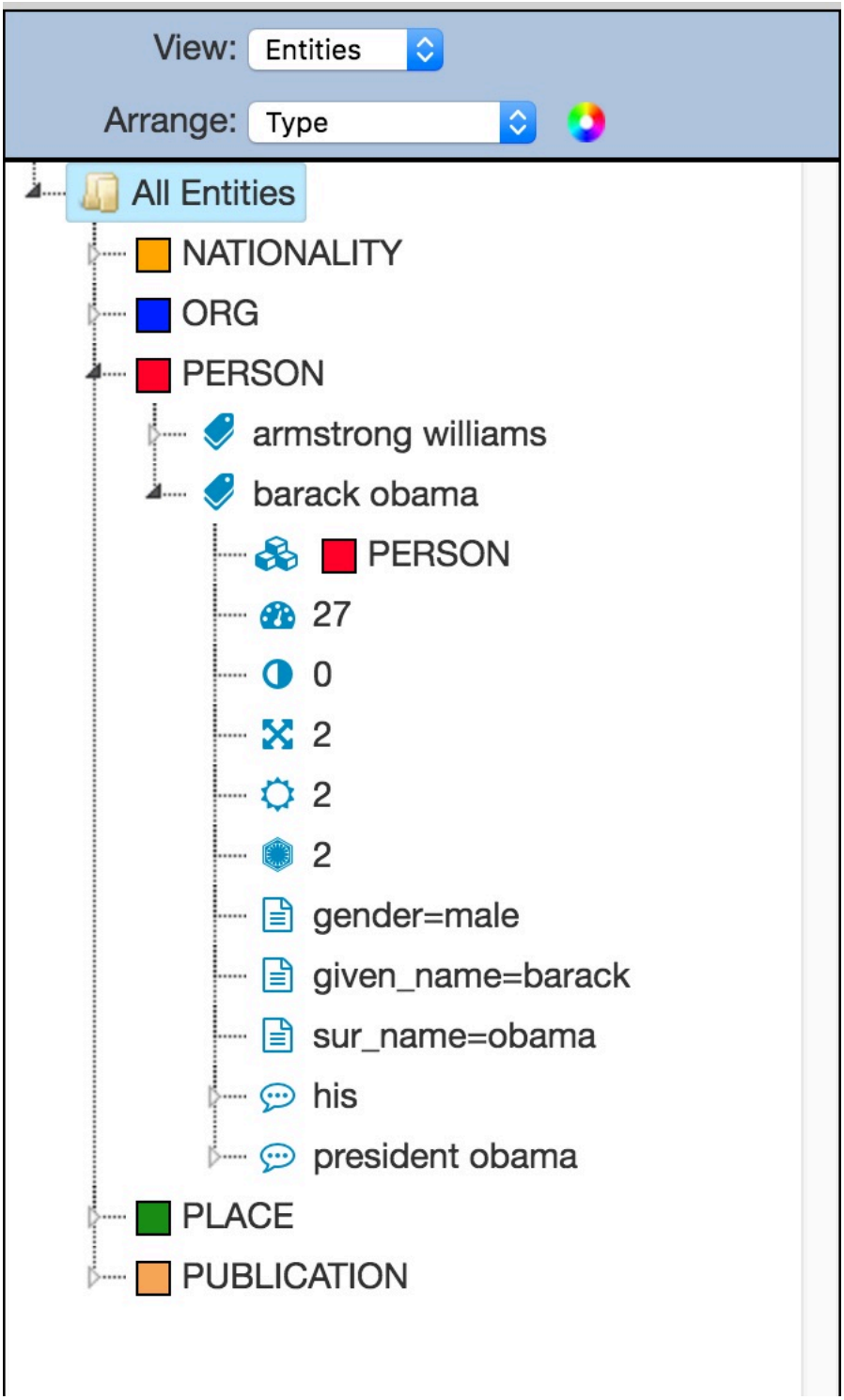



If the English gloss is already an entry in an English dictionary, TextChart uses the semantic vectors that are attached to the English entry. Optionally, you can also add semantic vectors to the non-English entry.

## Anaphora resolution

Anaphora refers to words that have already been used. If a document mentions a person by name in one place, it's likely that the same person is referred to again, later in the document, through the use of pronouns or in a different variant.

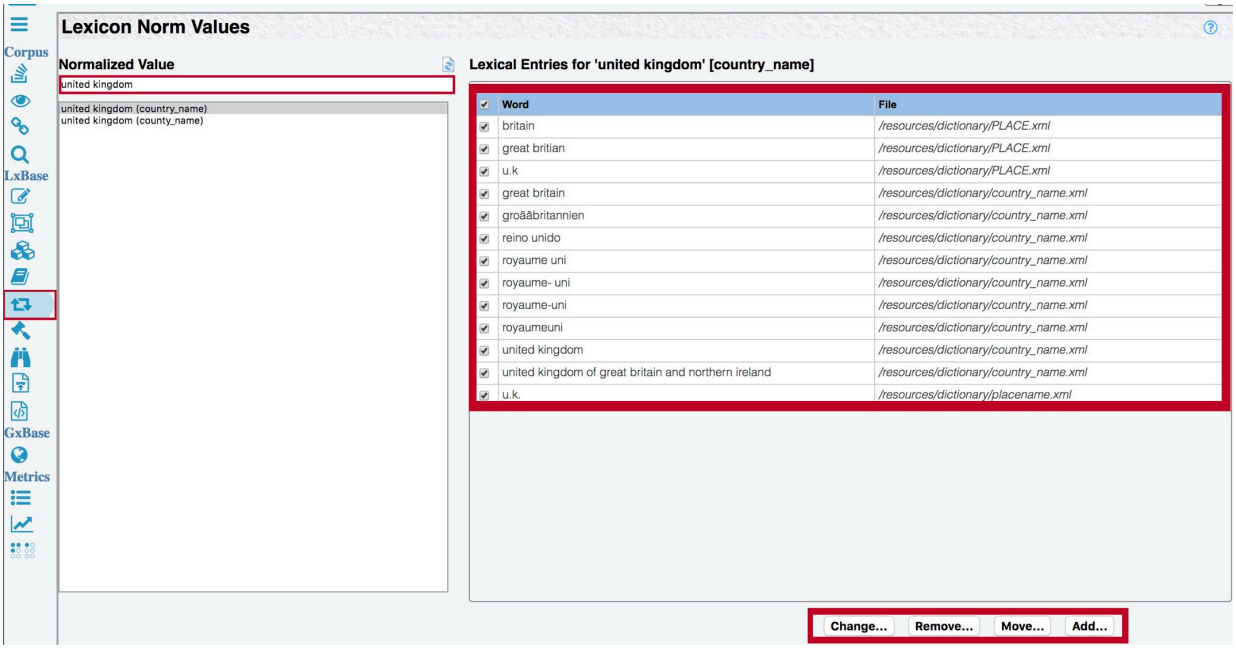
TextChart uses algorithms to determine which anaphora refer to which entities.



## Managing normalization

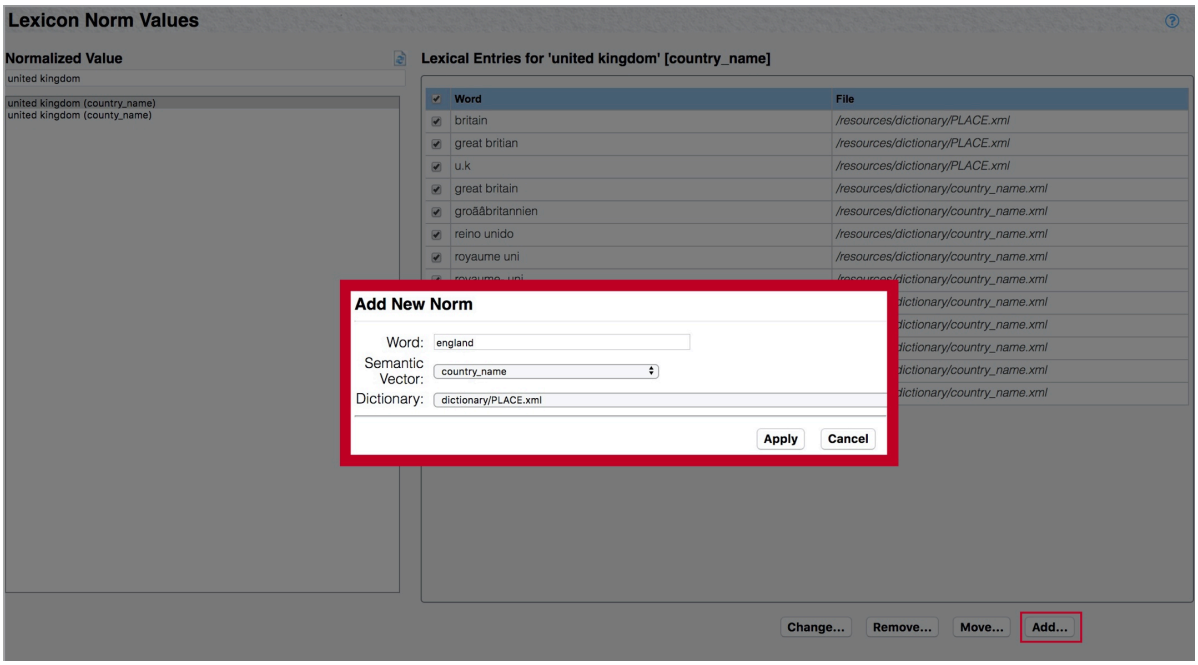
When you create dictionaries in TextChart Studio, you can arrange for lexical entries to be *normalized* to a standard entry. TextChart Studio also provides the ability to manage all such normalization in one place.

To see a list of the lexical entries that have been normalized, click **Edit lexical norm values** in the **LxBase** section of the vertical toolbar to open the **Lexical Norm Values** page.

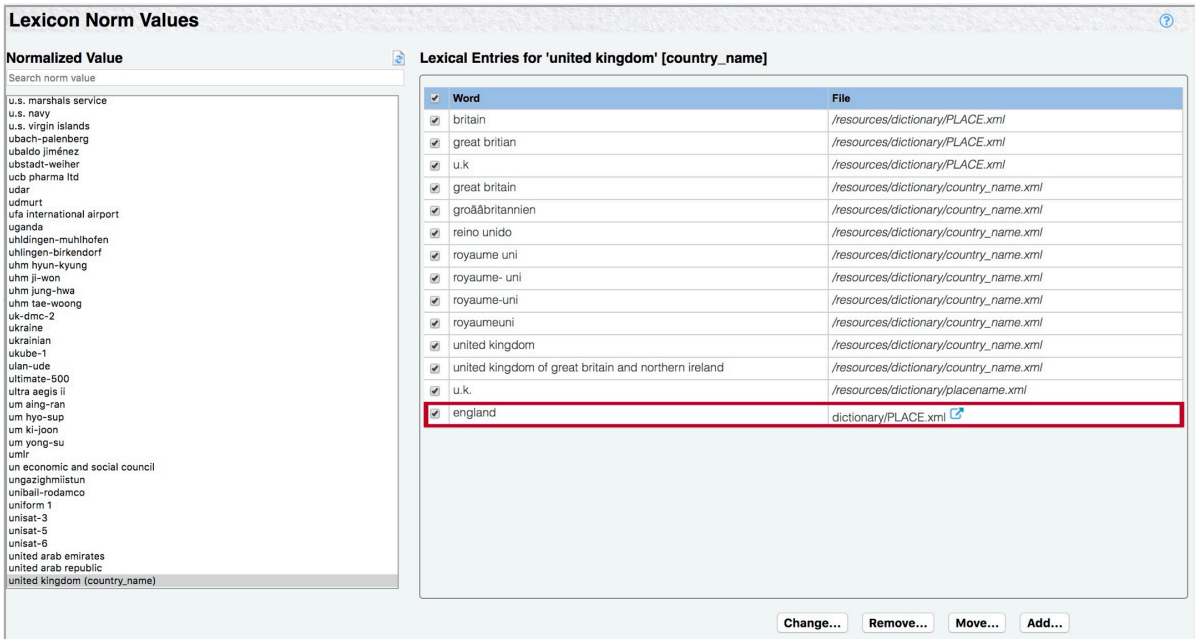



Selecting a normalized value on the left of the page populates the list on the right with a list of the lexical entries that normalize to it. The buttons below the list provide the following features:

- **Change** changes the normalization value for all the selected entries.
- **Remove** removes the selected entries.
- **Move** moves the value from one semantic vector to another in the same entry.
- **Add** creates a lexical entry that uses this normalized value. You can specify the term, and assign it a semantic vector and a dictionary. For example:



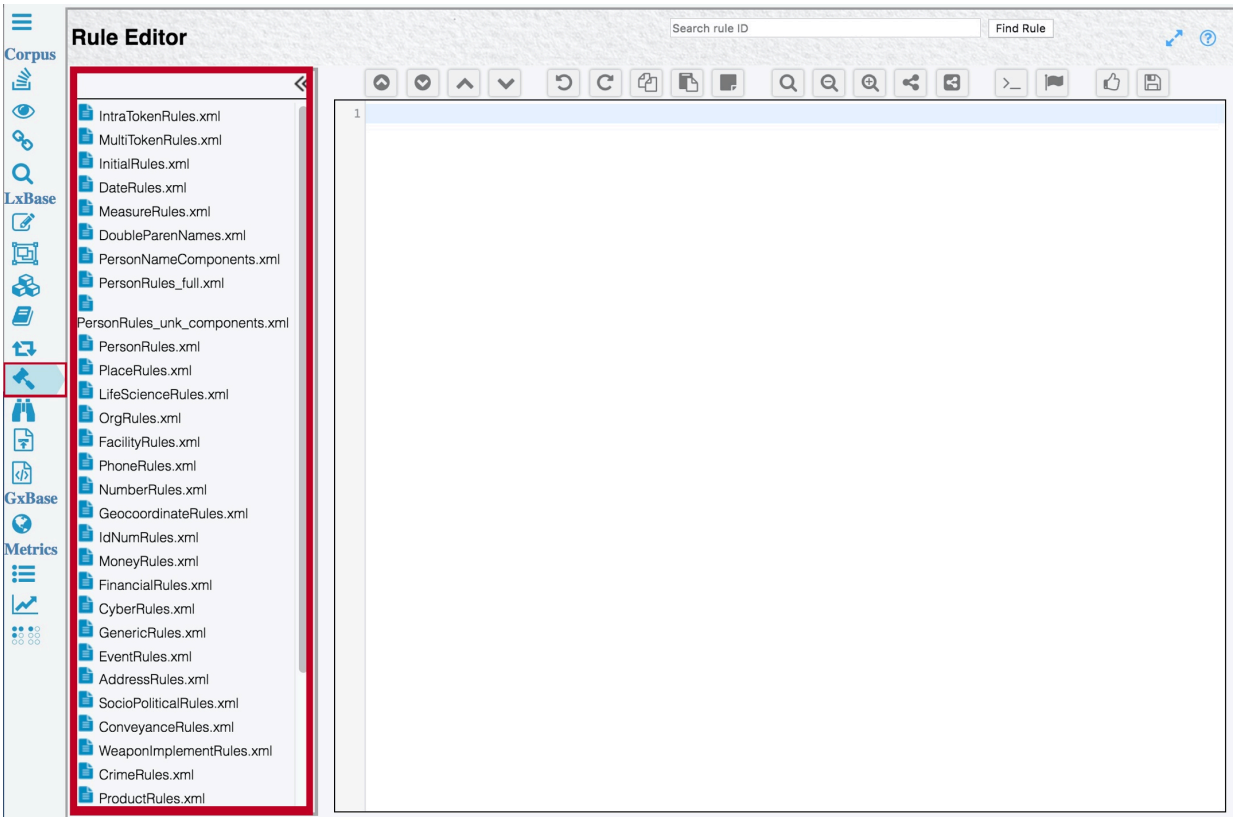

When you click **Apply**, the new entry appears in the list:



# Editing TextChart rules

i2 makes all of the rules that control extraction available in a set of XML files. You can use TextChart Studio to modify the rules in the supplied set, or to write new rules of your own.

TextChart rules specify the linguistic patterns that entities must match in order to be extracted. TextChart reads and applies rule files in the order they appear in `RuleFileList.xml`. To access this file, click **Manage LxBase** in the **LxBase** section of the vertical toolbar.




Within each rule file, TextChart applies the rules in the order in which they occur. To view and edit a particular rule file, click the **Edit rules** button in the menu.

To add a new rule to the open rule file, click the "Flag" button in the horizontal toolbar. TextChart Studio adds an XML template for the new rule to the file:

```
<Rule ID="ADD ID">
  <description>ADD DESCRIPTION</description>
  <comment>ADD ANY NECESSARY COMMENTS</comment>
  <example>ADD AN EXAMPLE OF THE DESIRED RESULT</example>
  <result>
    <combine></combine>
    <sv></sv>
    <attributes></attributes>
    <nolonger></nolonger>
  </result>
  <when>
    <T offset="0">
```

```
      <IS><sv></sv></IS>
      <ISNOT><sv></sv></ISNOT>
    </T>
  </when>
</Rule>
```

Every TextChart rule has a unique identifier and a description. You can also provide additional comments, and an example of the desired extraction result.

The rule's logic is composed of a `<result>` clause and a `<when>` clause. The `<result>` clause includes the semantic vector (`<sv>`) that the rule creates. For example, the matching pattern might be a PERSON entity or a `sur_name` semantic vector.

The `<result>` element also optionally includes `<combine>`, `<nolonger>`, and `<attributes>` elements:

- Use `<combine>` to merge multiple tokens into the same resulting semantic vector. The value, which can be positive or negative, is the number of tokens to combine together as a match, starting with a count of 0. If the value is negative, combination happens backwards from the 0 token. This is used to look backwards for recursion.
- Use `<nolonger>` to remove semantic vectors from tokens that match the pattern.
- Use `<attributes>` to assign attributes to a particular token in a match.

The `<when>` element contains the pattern itself. For each token in the match, semantic vectors that must match are specified in the `<IS>` element, while semantic vectors that must not match are specified in the `<ISNOT>` element.

If an `<IS>` element contains two semantic vectors, they're ORed together during processing. The following example illustrates a token that must be a given name or a surname, but not a verb:

```
<T offset="0">
  <IS><sv><given_name/><sur_name/></sv></IS>
  <ISNOT><sv><verb/></sv></ISNOT>
</T>
```

To AND semantic vectors, the `<when>` element must contain instances of tokens with the same offset. In this example, the token must be both a surname and a word that starts with a capital letter:

```
<T offset="0">
  <IS><sv><sur_name/></sv></IS>
</T>

<T offset="0">
  <IS><sv><cap_word/></sv></IS>
</T>
```

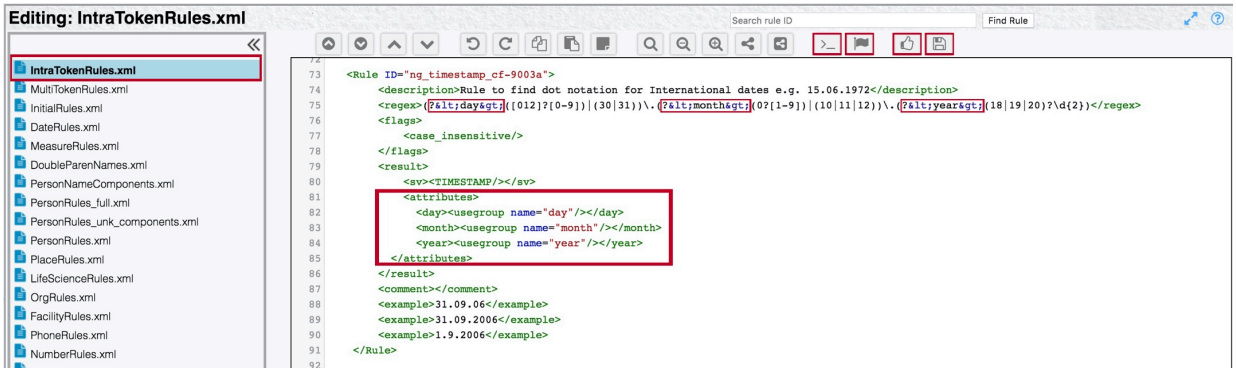
You can use a negative offset to find a match in the rule, without including that particular token in the extraction result.

## Regular expression rules

Intra-token rules and multi-token rules are used to define character-based regular expressions. The `IntraTokenRules.xml` file should only include regular expressions without character type change. Regular expressions *with* character type change should be multi-token rules. For example, if a pattern includes both letters and numbers, it should be a multi-token rule.

## Intra-token rules

The image below contains an intra-token rule that adds attribute information to the extraction result. The regular expression and the `<attribute>` element work together to enable the rule to enrich the result with metadata.




The regular expression in the rule matches attributes for "day", "month", and "year", and automatically adds the attributes to the extraction result:

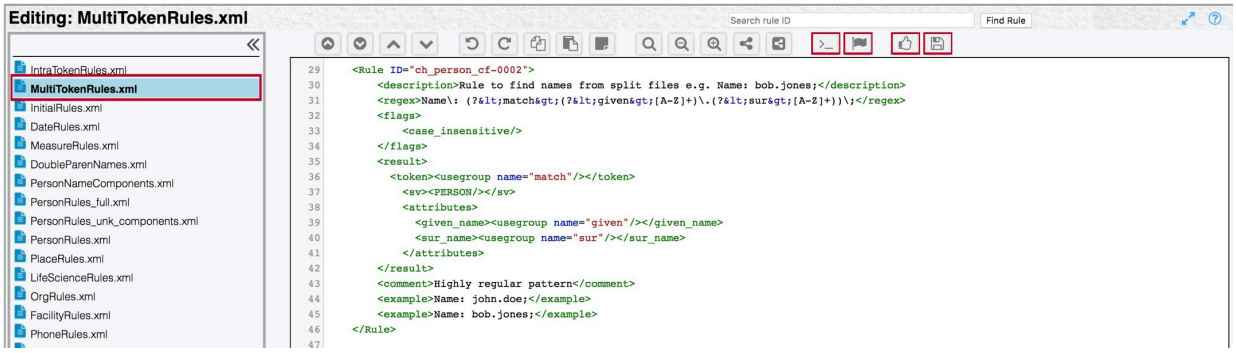
```
(?&lt;day&gt;([012]?[0-9])|(30|31))\.(?&lt;month&gt;(0?[1-9])|(10|11|12))\.
(?&lt;year&gt;(18|19|20)?\d{2})
```

The following attribute information is then completed within the rule:

```
<attributes>
  <day><usegroup name="day"/></day>
  <month><usegroup name="month"/></month>
  <year><usegroup name="year"/></year>
</attributes>
```

## Multi-token rules

The image below presents an example of a multi-token rule, where there is a character type change, and given name and surname attributes are automatically added to the extraction result. The rule is listed after the image.

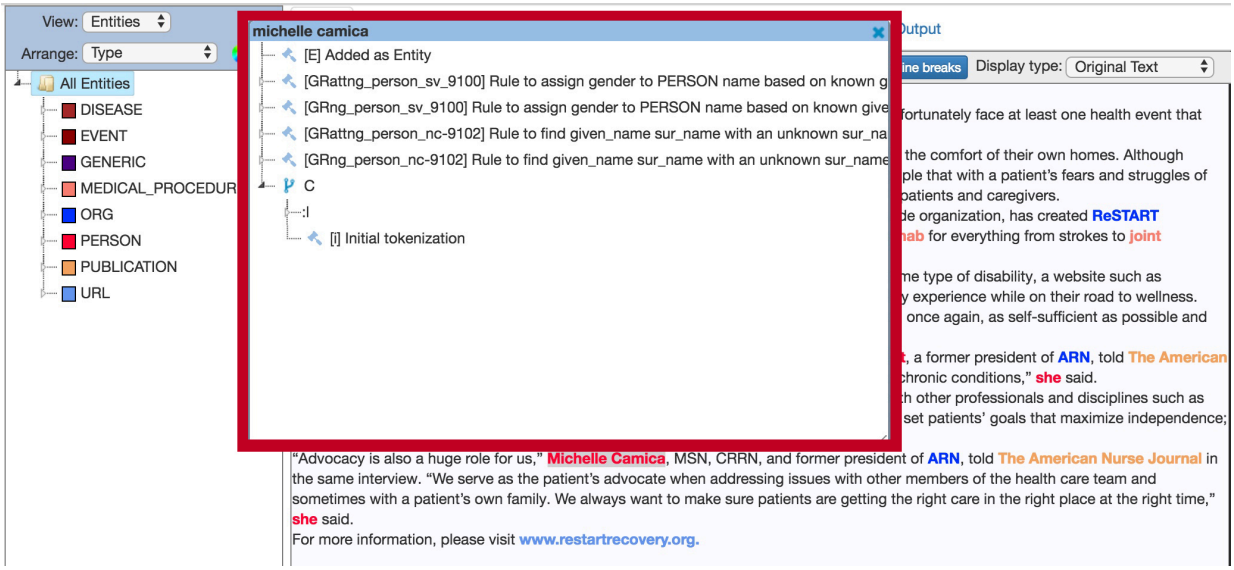


```
<Rule ID="ch_person_cf-0002">
  <description>Rule to find names from split files, such as Name:
 bob.jones;</description>
  <regex>Name\: (?&lt;match&gt;(?&lt;given&gt;[A-Z]+)\.(?&lt;sur&gt;[A-
Z]+))\;</regex>
  <flags>
    <case_insensitive/>
  </flags>
  <result>
    <token><usegroup name="match"/></token>
    <sv><PERSON/></sv>
    <attributes>
      <given_name><usegroup name="given"/></given_name>
      <sur_name><usegroup name="sur"/></sur_name>
    </attributes>
  </result>
  <comment>Highly regular pattern</comment>
  <example>Name: john.doe;</example>
  <example>Name: bob.jones;</example>
</Rule>
```

## Tracing TextChart rules

In TextChart Studio, you can use the document view to see which rules were responsible for an entity being extracted. Highlight the entity, right-click, and select **Show rule match detail**.

TextChart Studio displays a window that lists the order in which the TextChart engine executed lexical lookups and rules. From here, double-clicking a rule opens the corresponding rule file in the rule editor.




The rule trace contains the sequence of processing steps that took place when a string or document was processed. The following list provides a summary of how to interpret the steps:

- `:` = Delimiter between rule actions
- `i` = Initial tokenization
- `LL` = Lexical Lookup

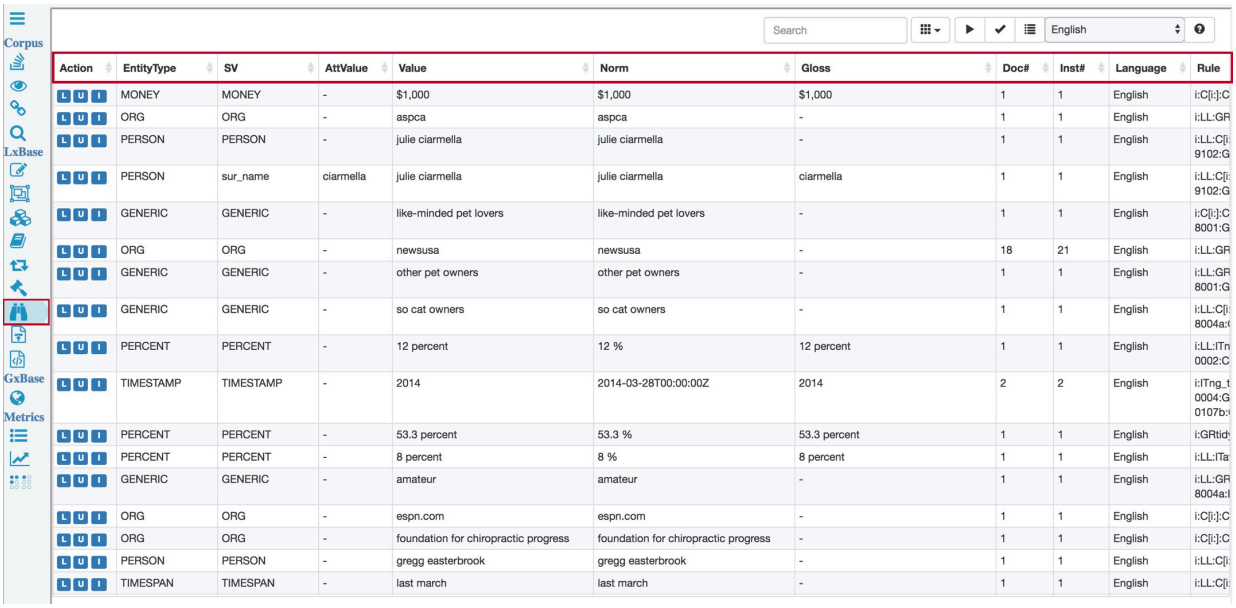
- `C` = The token was combined
- `[...]` = The contents of the brackets after the `C` includes the rule trace of the combined tokens
- `MT` = Multi-token rule
- `IT` = Intra-token rule
- `GR` = General rules
- `HC` = Hard-coded rule
- `BCE` = Back chaining entity detection
- `att` = Attribute portion of a rule assigning attribution to a token

## Adding terms in bulk

TextChart Studio provides two ways of adding terms to the lexicon in bulk. Through machine discovery, you can let the TextChart engine alert you to likely new terms in your documents so that you can add them. Alternatively, if you have a list of terms that you know TextChart does not understand, you can import and classify those terms by hand.

## Machine discovery

In addition to matching lexical items and patterns, TextChart uses *machine discovery* to extract other entities and lexical entries based on their linguistic context. Using their known semantic vectors, you can quickly add them to a lexicon as another meaningful semantic vector or entity, as well as unlearn or ignore them.




| Action | EntityType | SV | AttValue | Value | Norm | Gloss | Doc# | Inst# | Language | Rule |
|---|---|---|---|---|---|---|---|---|---|---|
| L U I | MONEY | MONEY | - | $1,000 | $1,000 | $1,000 | 1 | 1 | English | i:C[i:]:C |
| L U I | ORG | ORG | - | aspca | aspca | - | 1 | 1 | English | i:LL:GR |
| L U I | PERSON | PERSON | - | julie ciarmella | julie ciarmella | - | 1 | 1 | English | i:LL:C[i: 9102:G |
| L U I | PERSON | sur_name | ciarmella | julie ciarmella | julie ciarmella | ciarmella | 1 | 1 | English | i:LL:C[i: 9102:G |
| L U I | GENERIC | GENERIC | - | like-minded pet lovers | like-minded pet lovers | - | 1 | 1 | English | i:C[i:]:C 8001:G |
| L U I | ORG | ORG | - | newsusa | newsusa | - | 18 | 21 | English | i:LL:GR |
| L U I | GENERIC | GENERIC | - | other pet owners | other pet owners | - | 1 | 1 | English | i:LL:GR 8001:G |
| L U I | GENERIC | GENERIC | - | so cat owners | so cat owners | - | 1 | 1 | English | i:LL:C[i: 8004a: |
| L U I | PERCENT | PERCENT | - | 12 percent | 12 % | 12 percent | 1 | 1 | English | i:LL:ITn 0002:C |
| L U I | TIMESTAMP | TIMESTAMP | - | 2014 | 2014-03-28T00:00:00Z | 2014 | 2 | 2 | English | i:ITng_t 0004:G 0107b: |
| L U I | PERCENT | PERCENT | - | 53.3 percent | 53.3 % | 53.3 percent | 1 | 1 | English | i:GRtid |
| L U I | PERCENT | PERCENT | - | 8 percent | 8 % | 8 percent | 1 | 1 | English | i:LL:ITa |
| L U I | GENERIC | GENERIC | - | amateur | amateur | - | 1 | 1 | English | i:LL:GR 8004a:I |
| L U I | ORG | ORG | - | espn.com | espn.com | - | 1 | 1 | English | i:C[i:]:C |
| L U I | ORG | ORG | - | foundation for chiropractic progress | foundation for chiropractic progress | - | 1 | 1 | English | i:C[i:]:C |
| L U I | PERSON | PERSON | - | gregg easterbrook | gregg easterbrook | - | 1 | 1 | English | i:LL:C[i: |
| L U I | TIMESPAN | TIMESPAN | - | last march | last march | - | 1 | 1 | English | i:LL:C[i: |

After TextChart has processed a corpus, you can view the entities that it found through machine discovery by clicking **Perform discovery** in the LxBase section of the vertical toolbar.

Click **Discovery settings**

to select which entities or semantic vectors to learn.

Perform discovery by clicking **Start discovery**

.

You can choose to learn, unlearn, or ignore items from machine discovery by selecting the corresponding checkbox for each item, or by pressing the *L*, *U*, or *I* keys. You can also press *Space* to move to the next item without making a selection.

To add your selections to the lexicon, click **Commit actions**

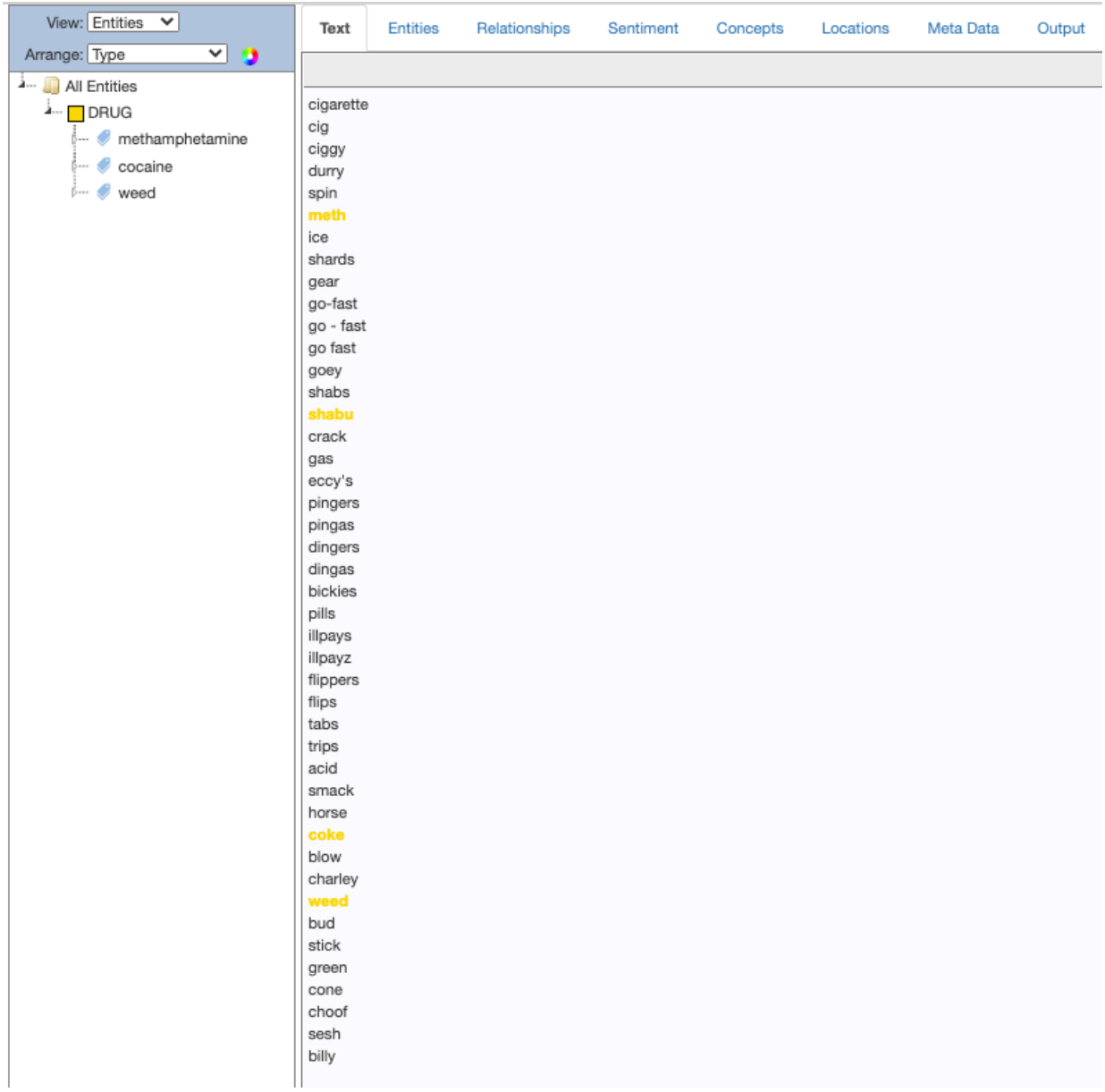
.

**Note:** Double-clicking an item in the **Value** or **Norm** column opens a window containing the surrounding context for each item. As well as the text, this context also includes the rule trace and the associated semantic vectors for each token.

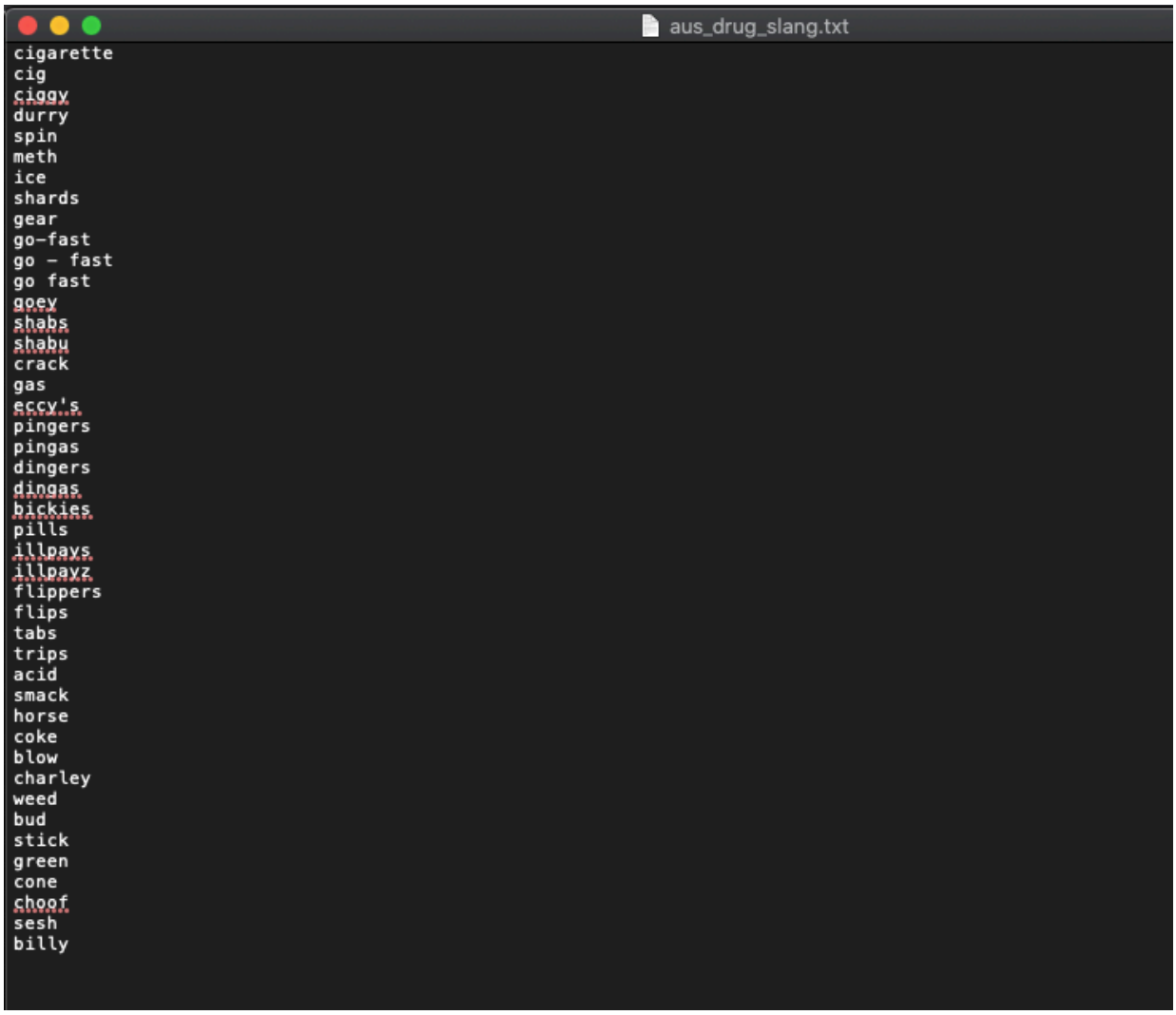
## Importing words

Another way of adding lexical items to LxBase is through TextChart Studio's word import tool, which you open by clicking **Import word list** in the vertical toolbar. The tool allows you to add a large list of lexical entries by appending them to an existing file, overwriting an existing file, or creating an additional file.

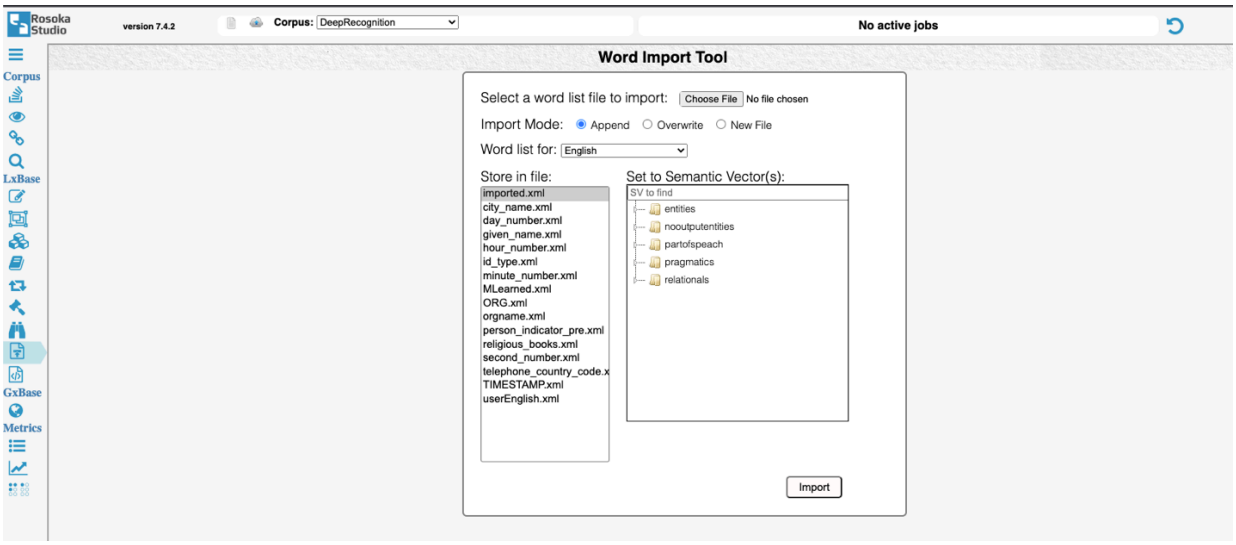
For example, the next image shows a list of drug slang terms that the TextChart engine has processed. Some of the results are highlighted in yellow, which represents an extracted DRUG entity result. The others do not have this "hit highlighting" and were therefore not extracted as an entity.



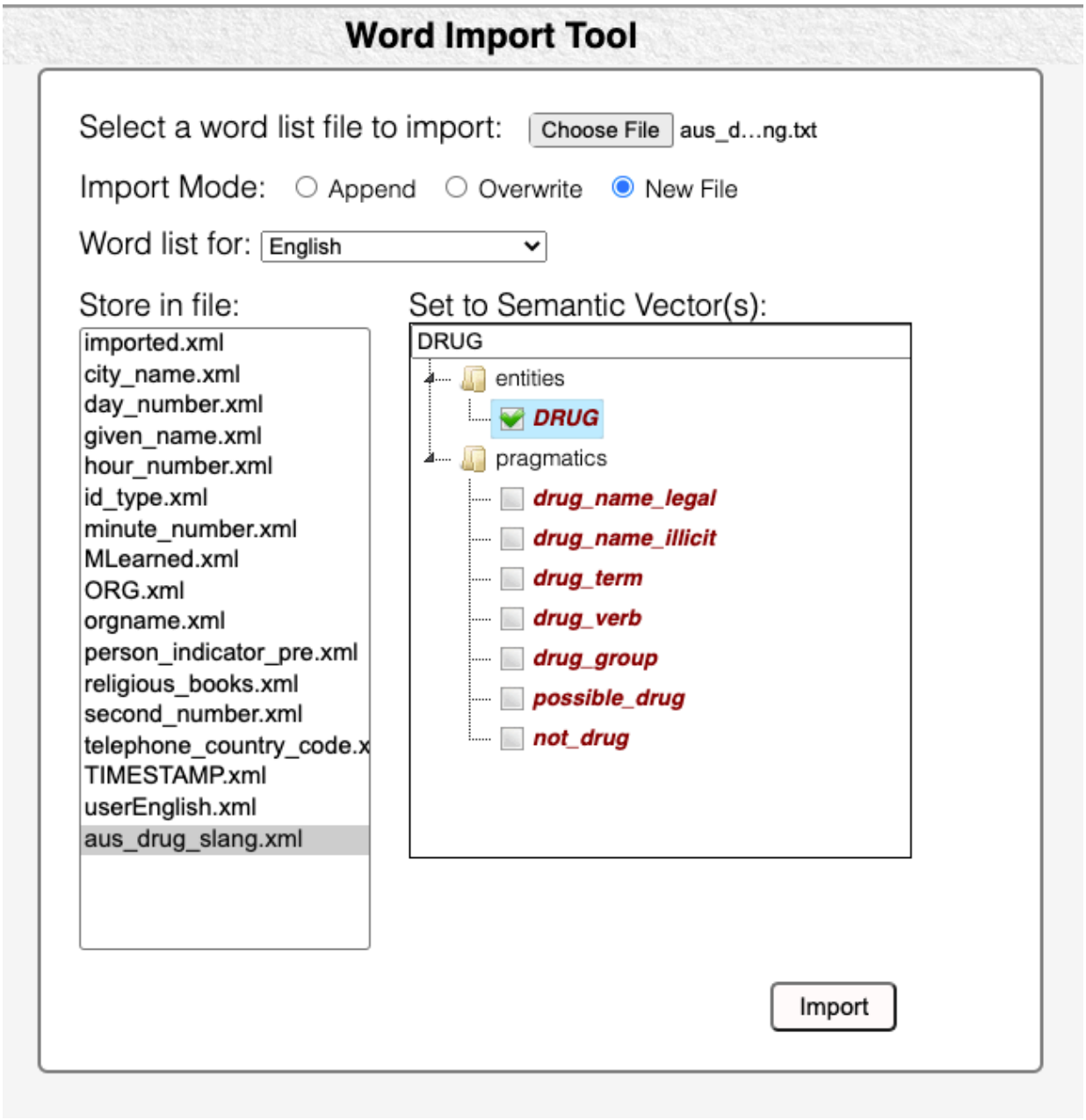

To add some or all of the words that were not extracted to the LxBase, you can create a text file that contains them. Entries in the text file must be formatted with a line break between each entry, as in the next image.



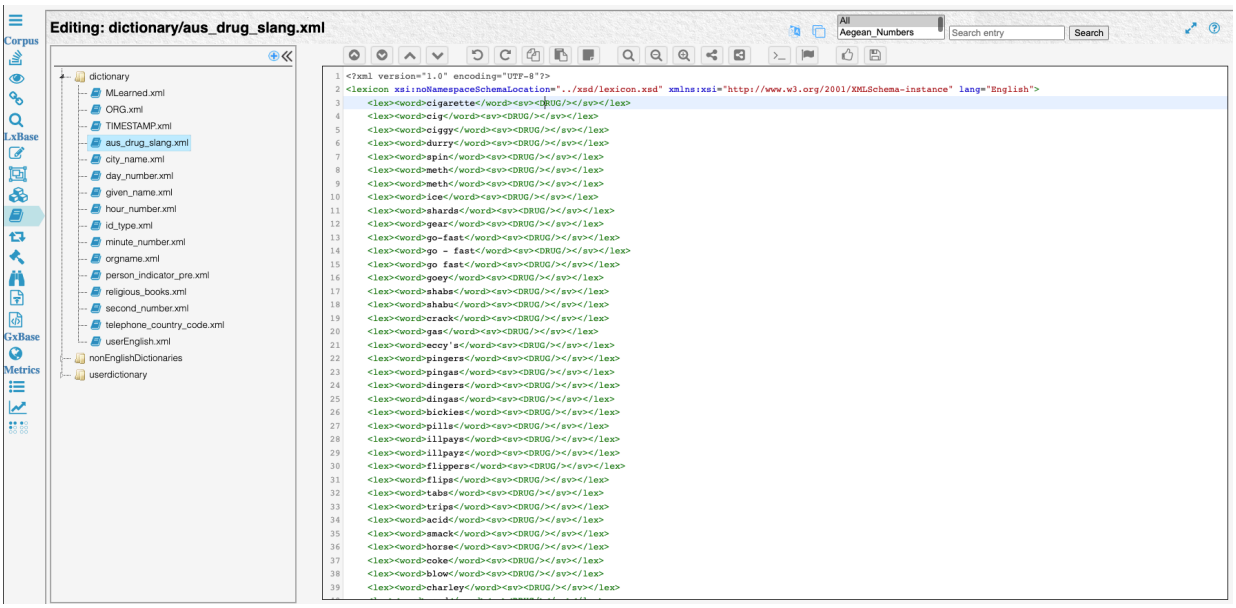

When you open the word import tool, you can click **Choose File** to locate the text file that you created, and then choose the import mode.



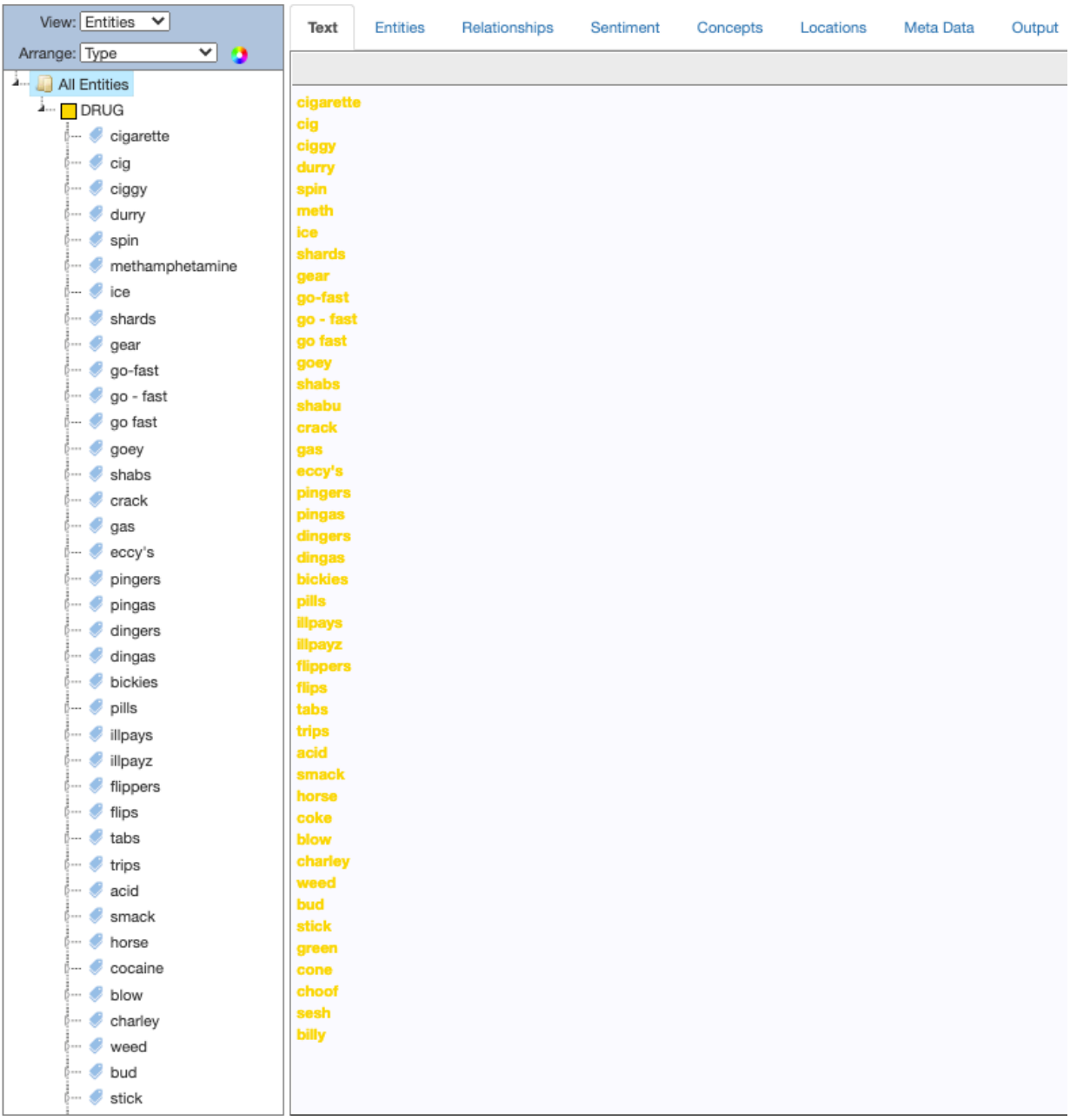

In the next image, the word import tool is configured to create a file for storing the new terms, and to associate each term with the DRUG entity type.




A few seconds after you click **Import**, the TextChart engine creates a dictionary file, complete with the appropriate XML. If you want to see the file, click the "Book" icon in the LxBase section of the vertical toolbar.

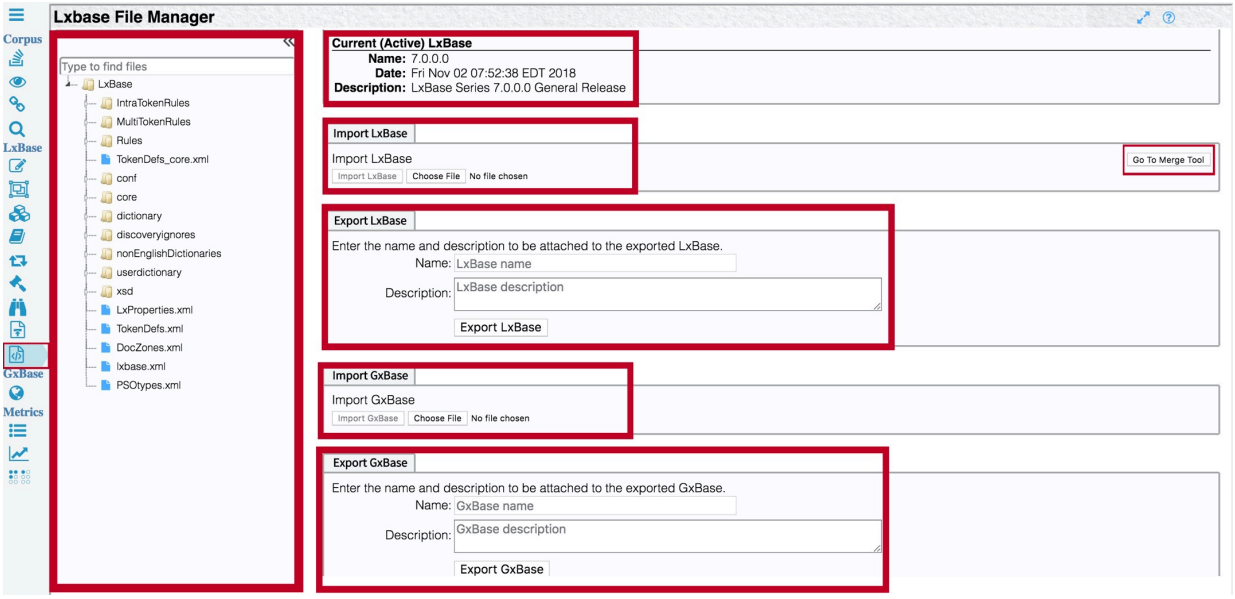



If you now reprocess the original source document inside TextChart Studio, all the terms are displayed in yellow, indicating that they are being extracted as entities of type DRUG.




## Importing and exporting LxBases

TextChart Studio provides a tool for managing LxBase files, and for importing, exporting, and merging LxBases. To open the **LxBase File Manager** page, click **Manage LxBase Files** in the LxBase section of the vertical toolbar.




The tree view on the left of the page contains a searchable list of LxBase files, while the right side provides information about the current LxBase and a series of actions that you can perform on it.

### Import LxBase

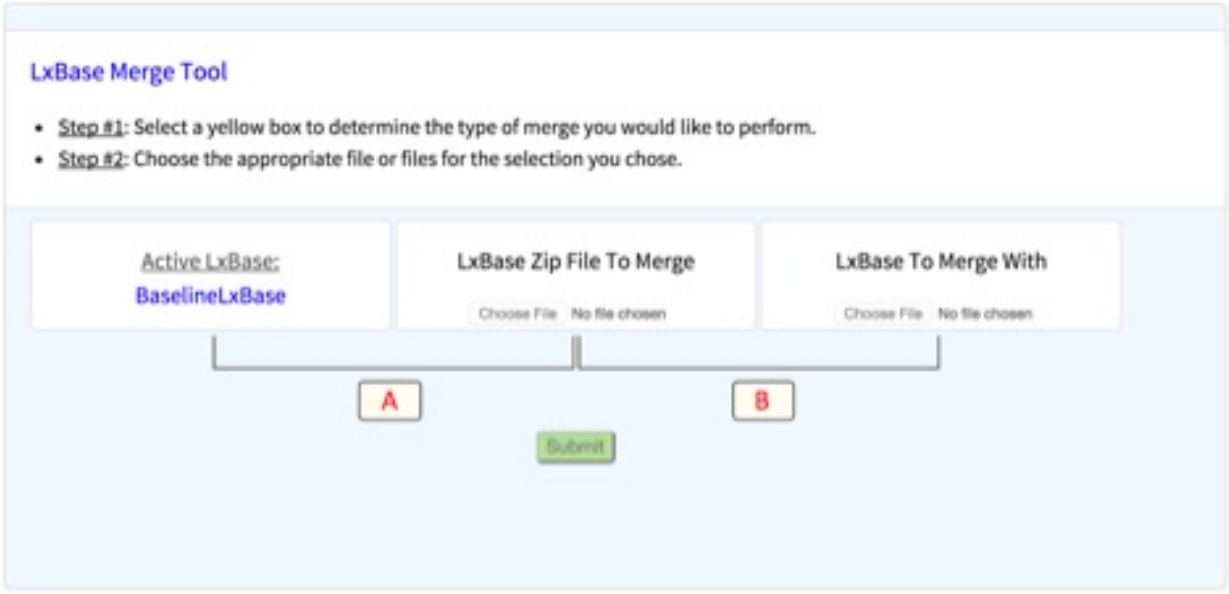
To import a new LxBase, click **Choose File** in the **Import LxBase** panel, locate the ZIP file containing the LxBase you want to import, and then click **Import LxBase**.

### Export LxBase

To export an LxBase, provide a name and a description in the **Export LxBase** panel, and then click **Export LxBase** to download a ZIP file containing the LxBase along with necessary metadata.

### Merge LxBases

To consolidate the work of multiple users, or to incorporate an update to the default LxBase into a customized one, TextChart Studio can merge two LxBases together. To begin, click **Go To Merge Tool** in the **Import LxBase** panel.




Here, you can either merge a single LxBase ZIP file into the active LxBase (**A**), or merge two LxBase ZIP files together (**B**).

**Important:** To restore an existing LxBase after completing a merge, you must save a regression checkpoint beforehand.

After you select one or two LxBase ZIP files, TextChart Studio displays a table of their contents, organized by category. For each category, the table indicates how many files need merging, how many are unique to each LxBase, and how many are identical. To start merging, click in the leftmost column in the first row.

| | Total | Pending | UniqueA | UniqueB | Identical | Resolved |
|---|---|---|---|---|---|---|
| Conf Files | 5 | 3 | 0 | 1 | 1 | 0 |
| MultiTokenRules Files | 1 | 1 | 0 | 0 | 0 | 0 |
| IntraTokenRules Files | 1 | 1 | 0 | 0 | 0 | 0 |
| Rules Files | 21 | 17 | 0 | 3 | 1 | 0 |
| Dictionary Files (English) | 18 | 1 | 5 | 12 | 0 | 0 |
| Non-English Dictionary Files | 217 | 3 | 0 | 0 | 214 | 0 |
| User Dictionary Files | 1 | 1 | 0 | 0 | 0 | 0 |
| Discovery Ignores Files | 1 | 0 | 0 | 0 | 1 | 0 |

- **Pending**

  Pending files are not identical between the two LxBases, and therefore you need to resolve their differences.

  Click **Edit** to open a dialog that allows you to view the differences between the files. You can decide whether the merged LxBase should use the contents of one of the LxBases in the merge, or you can edit to create a hybrid. Click **Back** to return to the table without saving changes, or **Submit** to validate your changes and save them if validation was successful.

- **Unique**

  Unique files appear in one of the LxBases being merged, but not the other. To include a unique file in the result, click **Include**; to exclude it, click **Exclude**. Either decision resolves the merge process for that file.

- **Identical**

  Identical files appear in both of the LxBases being merged, and have the same name and the same contents. You don't need to do anything to identical files, apart from clicking **Resolve** to acknowledge them.

  To resolve all identical files at once, click **Resolve All Identicals**.

To change which LxBase appears in which pane on the screen, click **Reset Pane**. To abandon all current merges and delete the selected ZIP files, click **Reset Merge**.

When you've resolved all the files in one category, you can move on to the next. When you've resolved all the files in all the categories, click **Final Validation** to make sure that your completed file is valid. Provided that it is, you can then complete the merge operation.

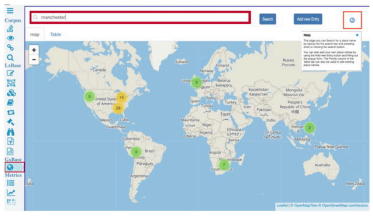
# GxBase

TextChart uses GxBase to retrieve names and geocoordinates for the places that it finds in documents in your corpora. The standard GxBase includes the National Geospatial Agency's (NGA) GNS, the United States Geological Survey's (USGS) GNIS, and i2's own internally developed gazetteer to provide worldwide coverage.

In TextChart Studio, you can search a GxBase, customize it, and add features that are specific to your domain. To begin, click **GxBase** in the **GxBase** section of the vertical toolbar..

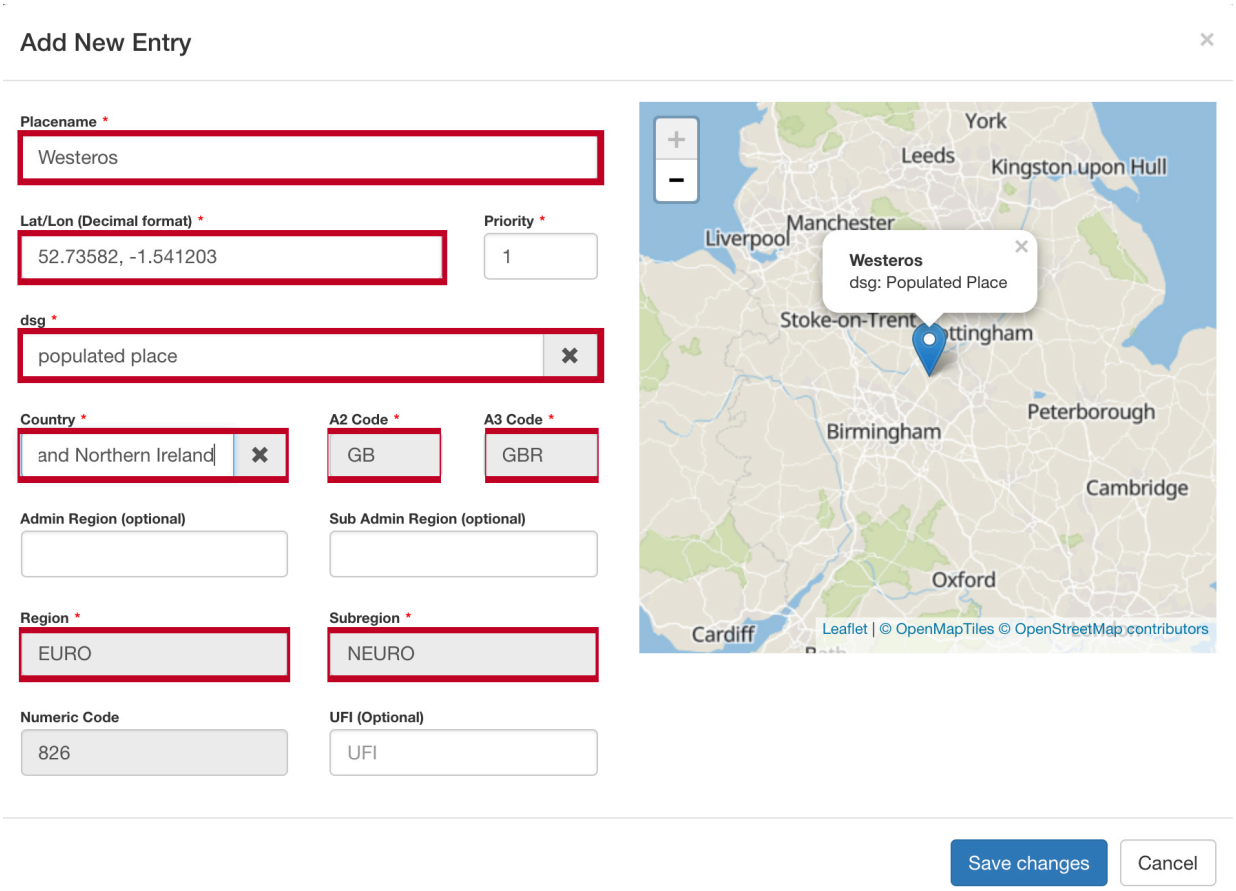
## Searching for places

To search for a place in GxBase, type its name into the search bar and press *Enter*. All instances of the name that exist in the gazetteer are shown on the map, and you can zoom in and out as you need.

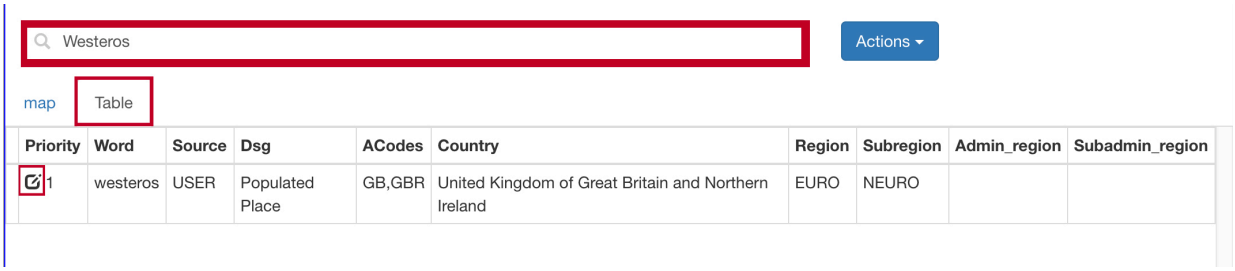


## Adding a place name

To add a place to GxBase, type its name ito the search bar and click **Add new Entry** to display the **And New Entry** dialog. Complete the required information, and then click **Save changes**.

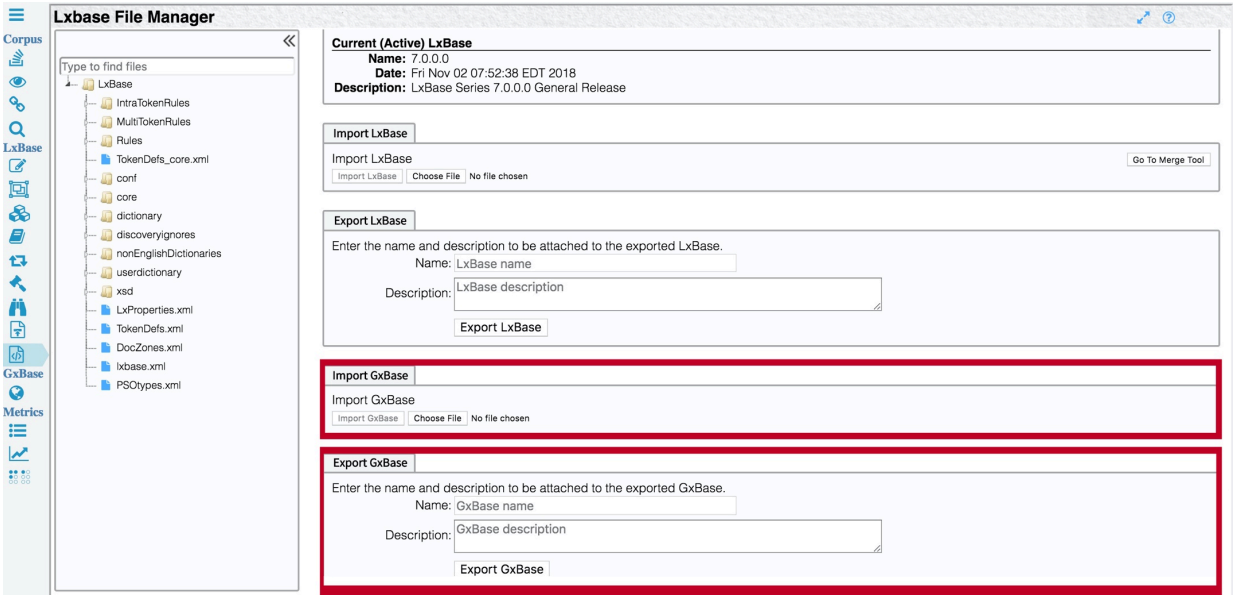



After you add the new place, you can open the **Table** tab to search for and view the entry for it in GxBase. To make adjustments, click the button in the **Priority** column.




## Importing and exporting GxBases

You can import and export customized GxBases through the same interface that you use to import and export LxBases. Open the **LxBase File Manager** page by clicking **Manage LxBase Files** in the LxBase section of the vertical toolbar, and use the **Import GxBase** and **Export GxBase** panels according to your needs.

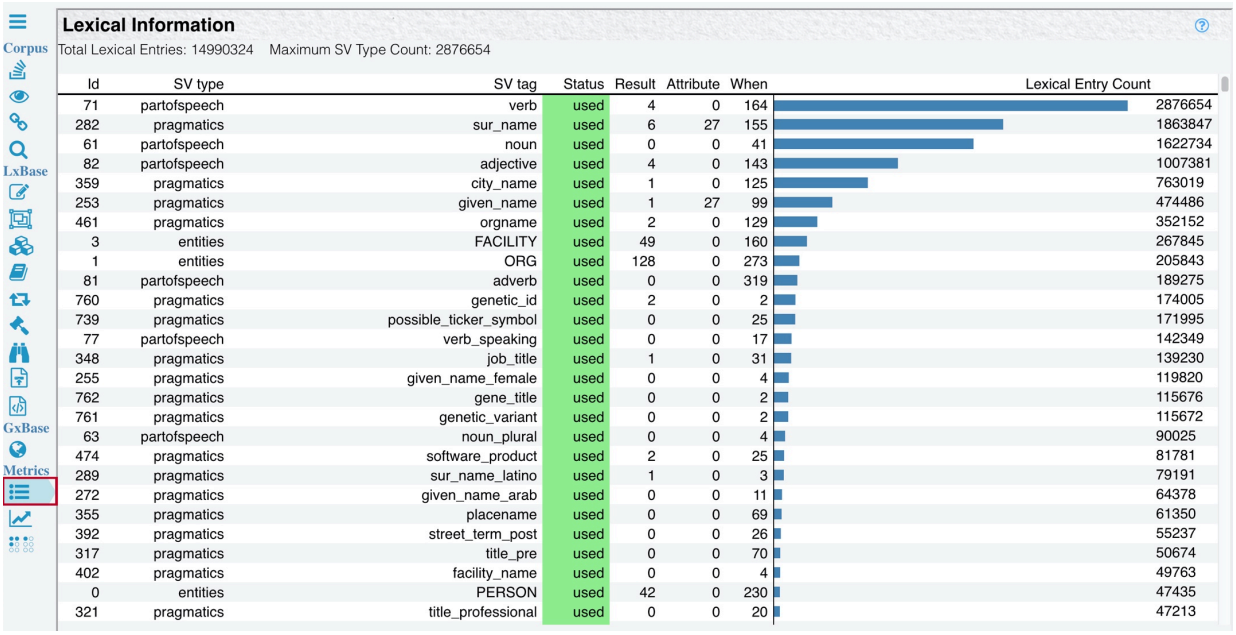



# Metrics

TextChart Studio provides metrics about the active LxBase that you can review and act upon to improve the efficiency of the system. The items in the **Metrics** menu reveal information about lexical entries (including semantic vectors) and LxBase rules.

## Lexical information

Click **Analyze lexicon** in the **Metrics** section of the vertical toolbar to open the **Lexical information** page. TextChart Studio displays information including the total number of lexical entries in all dictionaries, and the counts of the semantic vector types that occur most frequently.

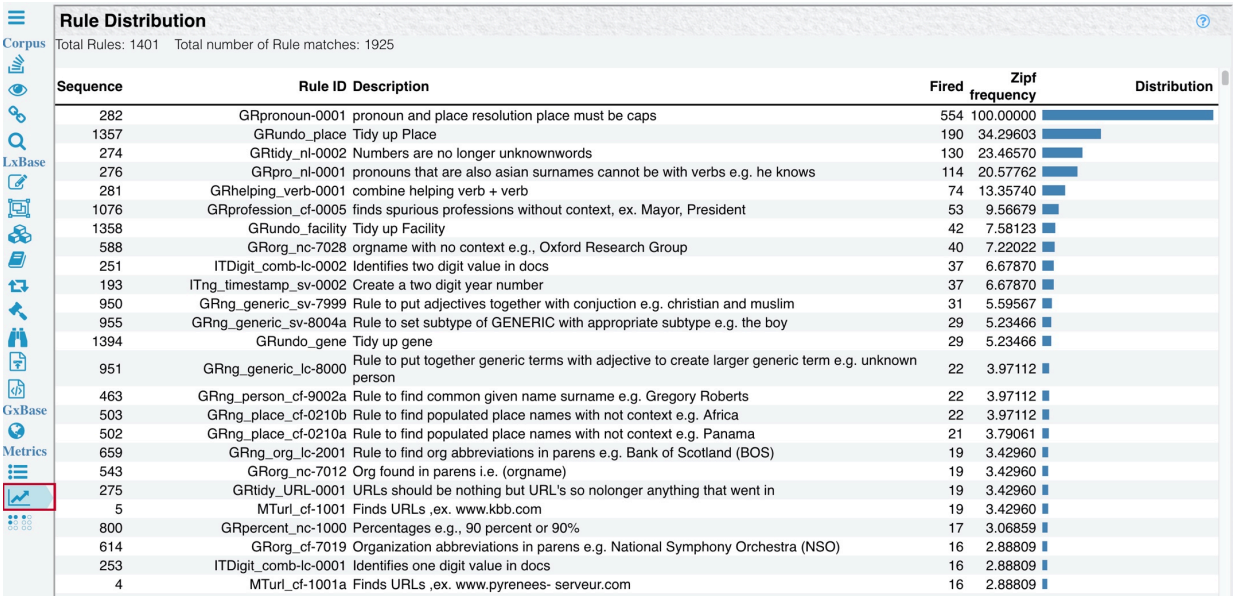



| Id | SV type | SV tag | Status | Result | Attribute | When | Lexical Entry Count |
|---|---|---|---|---|---|---|---|
| 71 | partofspeech | verb | used | 4 | 0 | 164 | 2876654 |
| 282 | pragmatics | sur_name | used | 6 | 27 | 155 | 1863847 |
| 61 | partofspeech | noun | used | 0 | 0 | 41 | 1622734 |
| 82 | partofspeech | adjective | used | 4 | 0 | 143 | 1007381 |
| 359 | pragmatics | city_name | used | 1 | 0 | 125 | 763019 |
| 253 | pragmatics | given_name | used | 1 | 27 | 99 | 474486 |
| 461 | pragmatics | orgname | used | 2 | 0 | 129 | 352152 |
| 3 | entities | FACILITY | used | 49 | 0 | 160 | 267845 |
| 1 | entities | ORG | used | 128 | 0 | 273 | 205843 |
| 81 | partofspeech | adverb | used | 0 | 0 | 319 | 189275 |
| 760 | pragmatics | genetic_id | used | 2 | 0 | 2 | 174005 |
| 739 | pragmatics | possible_ticker_symbol | used | 0 | 0 | 25 | 171995 |
| 77 | partofspeech | verb_speaking | used | 0 | 0 | 17 | 142349 |
| 348 | pragmatics | job_title | used | 1 | 0 | 31 | 139230 |
| 255 | pragmatics | given_name_female | used | 0 | 0 | 4 | 119820 |
| 762 | pragmatics | gene_title | used | 0 | 0 | 2 | 115676 |
| 761 | pragmatics | genetic_variant | used | 0 | 0 | 2 | 115672 |
| 63 | partofspeech | noun_plural | used | 0 | 0 | 4 | 90025 |
| 474 | pragmatics | software_product | used | 2 | 0 | 25 | 81781 |
| 289 | pragmatics | sur_name_latino | used | 1 | 0 | 3 | 79191 |
| 272 | pragmatics | given_name_arab | used | 0 | 0 | 11 | 64378 |
| 355 | pragmatics | placename | used | 0 | 0 | 69 | 61350 |
| 392 | pragmatics | street_term_post | used | 0 | 0 | 26 | 55237 |
| 317 | pragmatics | title_pre | used | 0 | 0 | 70 | 50674 |
| 402 | pragmatics | facility_name | used | 0 | 0 | 4 | 49763 |
| 0 | entities | PERSON | used | 42 | 0 | 230 | 47435 |
| 321 | pragmatics | title_professional | used | 0 | 0 | 20 | 47213 |

The page also displays a chart that lists all entities and semantic vectors, as well as associated usage information. You can find the the number of times a semantic vector occurs in different parts of a rule (results clause, attribute clause, and when clause), and the number of lexical entries that contain that semantic vector.

### Rule distribution

Clicking **View rule distribution** in the **Metrics** section of the vertical toolbar to open the **Rule Distribution** page. TextChart Studio displays information about the frequency and distribution of rule matches.




| Sequence | Rule ID | Description | Fired | Zipf frequency | Distribution |
|---|---|---|---|---|---|
| 282 | GRpronoun-0001 | pronoun and place resolution place must be caps | 554 | 100.00000 | |
| 1357 | GRundo_place | Tidy up Place | 190 | 34.29603 | |
| 274 | GRtidy_nl-0002 | Numbers are no longer unknownwords | 130 | 23.46570 | |
| 276 | GRpro_nl-0001 | pronouns that are also asian surnames cannot be with verbs e.g. he knows | 114 | 20.57762 | |
| 281 | GRhelping_verb-0001 | combine helping verb + verb | 74 | 13.35740 | |
| 1076 | GRprofession_cf-0005 | finds spurious professions without context, ex. Mayor, President | 53 | 9.56679 | |
| 1358 | GRundo_facility | Tidy up Facility | 42 | 7.58123 | |
| 588 | GRorg_nc-7028 | orgname with no context e.g., Oxford Research Group | 40 | 7.22022 | |
| 251 | ITDigit_comb-lc-0002 | Identifies two digit value in docs | 37 | 6.67870 | |
| 193 | ITng_timestamp_sv-0002 | Create a two digit year number | 37 | 6.67870 | |
| 950 | GRng_generic_sv-7999 | Rule to put adjectives together with conjuction e.g. christian and muslim | 31 | 5.59567 | |
| 955 | GRng_generic_sv-8004a | Rule to set subtype of GENERIC with appropriate subtype e.g. the boy | 29 | 5.23466 | |
| 1394 | GRundo_gene | Tidy up gene | 29 | 5.23466 | |
| 951 | GRng_generic_lc-8000 | Rule to put together generic terms with adjective to create larger generic term e.g. unknown person | 22 | 3.97112 | |
| 463 | GRng_person_cf-9002a | Rule to find common given name surname e.g. Gregory Roberts | 22 | 3.97112 | |
| 503 | GRng_place_cf-0210b | Rule to find populated place names with not context e.g. Africa | 22 | 3.97112 | |
| 502 | GRng_place_cf-0210a | Rule to find populated place names with not context e.g. Panama | 21 | 3.79061 | |
| 659 | GRng_org_lc-2001 | Rule to find org abbreviations in parens e.g. Bank of Scotland (BOS) | 19 | 3.42960 | |
| 543 | GRorg_nc-7012 | Org found in parens i.e. (orgname) | 19 | 3.42960 | |
| 275 | GRtidy_URL-0001 | URLs should be nothing but URL's so nolonger anything that went in | 19 | 3.42960 | |
| 5 | MTurl_cf-1001 | Finds URLs ,ex. www.kbb.com | 19 | 3.42960 | |
| 800 | GRpercent_nc-1000 | Percentages e.g., 90 percent or 90% | 17 | 3.06859 | |
| 614 | GRorg_cf-7019 | Organization abbreviations in parens e.g. National Symphony Orchestra (NSO) | 16 | 2.88809 | |
| 253 | ITDigit_comb-lc-0001 | Identifies one digit value in docs | 16 | 2.88809 | |
| 4 | MTurl_cf-1001a | Finds URLs ,ex. www.pyrenees- serveur.com | 16 | 2.88809 | |

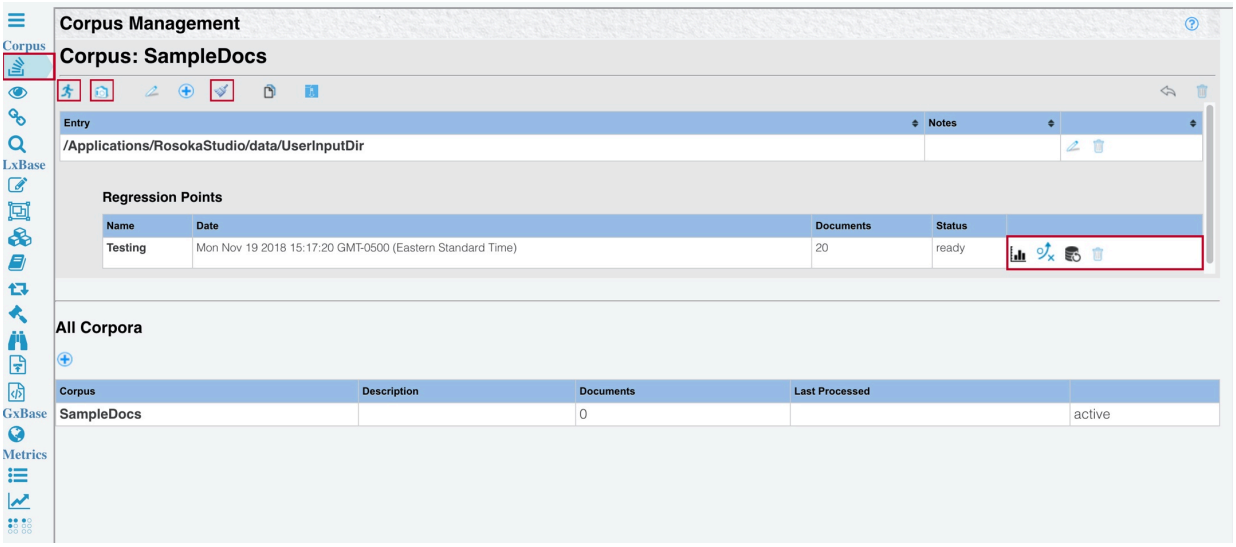
The chart on this page lists all rules in the LxBase, the number of times the rule matched, and the Zipf frequency of the rule.

## Regression testing

TextChart Studio has a built-in regression testing feature for measuring the impact that changes to the LxBase have on the output. When a corpus is active, all saved regression points are listed in a chart below the corpus entry in the **Corpus Management** page.

To create a new regression point, click **Create New Regression Point**

in the horizontal toolbar above the corpus entry.




After you make lexical or rule modifications to the LxBase, you can score a current run against a regression point. First, click **Clear Processing Results** to remove any previous results from the comparison. Then you need to reprocess the corpus.

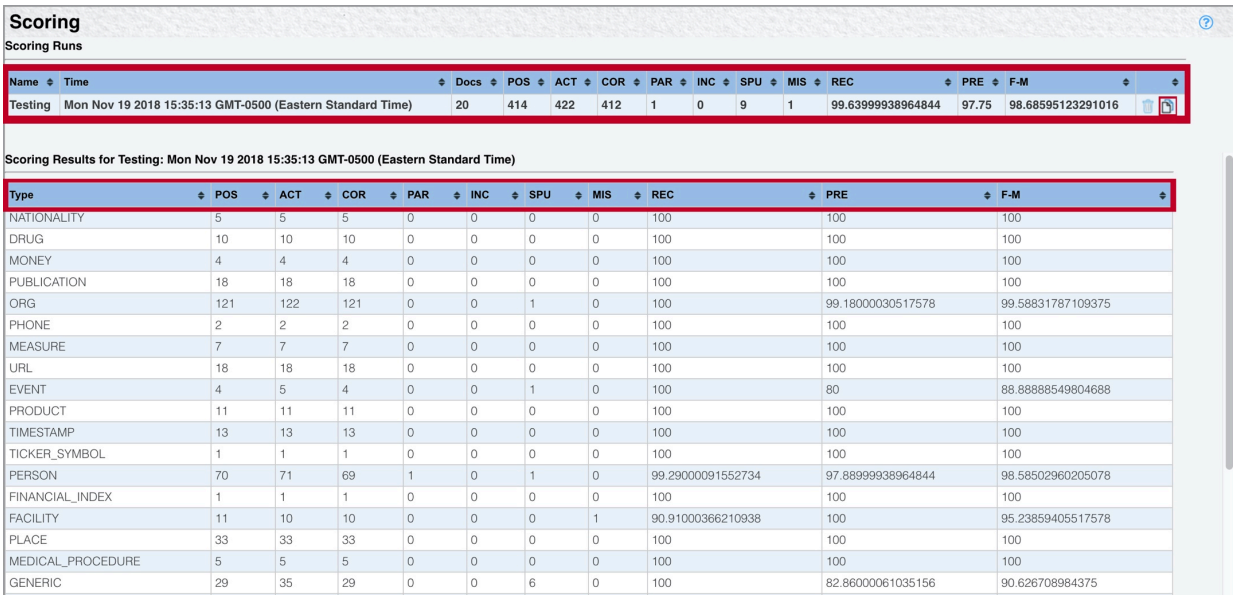
To compare the results of the current run with a regression point, click **Score against current results**

in the appropriate row of the corpus list, or **Compare and score results** in the **Metrics** section of the vertical toolbar. You can save multiple regression points and test against any of them.

TextChart Studio retains regression points and scoring results until you delete them. To view scoring results without prompting a new run, click **View Score Results** in the appropriate row. You can also revert the active LxBase to an earlier version by clicking **Restore saved LxBase**.

### Interpreting results

As you view the scoring results of a regression run, you can click the table at the top of the **Scoring** page to see a breakdown for each entity type.



**Scoring**

**Scoring Runs**

| Name | Time | Docs | POS | ACT | COR | PAR | INC | SPU | MIS | REC | PRE | F-M | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Testing | Mon Nov 19 2018 15:35:13 GMT-0500 (Eastern Standard Time) | 20 | 414 | 422 | 412 | 1 | 0 | 9 | 1 | 99.63999938964844 | 97.75 | 98.68595123291016 | |

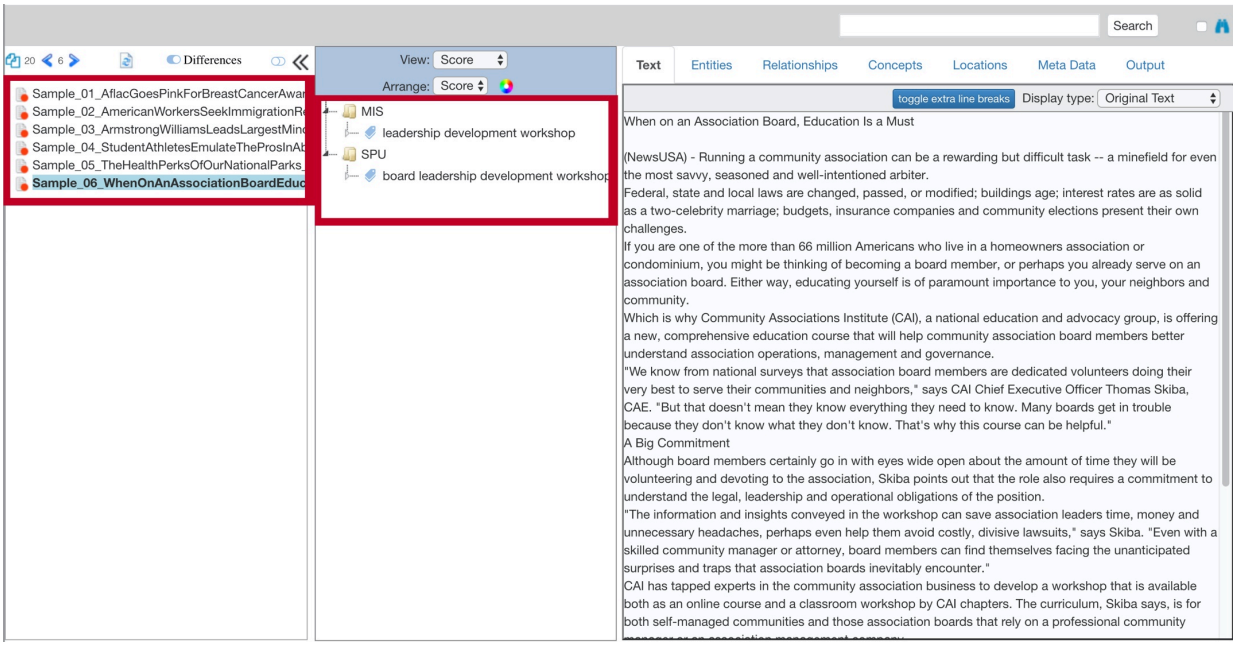
**Scoring Results for Testing: Mon Nov 19 2018 15:35:13 GMT-0500 (Eastern Standard Time)**

| Type | POS | ACT | COR | PAR | INC | SPU | MIS | REC | PRE | F-M |
|---|---|---|---|---|---|---|---|---|---|---|
| NATIONALITY | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| DRUG | 10 | 10 | 10 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| MONEY | 4 | 4 | 4 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| PUBLICATION | 18 | 18 | 18 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| ORG | 121 | 122 | 121 | 0 | 0 | 1 | 0 | 100 | 99.18000030517578 | 99.58831787109375 |
| PHONE | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| MEASURE | 7 | 7 | 7 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| URL | 18 | 18 | 18 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| EVENT | 4 | 5 | 4 | 0 | 0 | 1 | 0 | 100 | 80 | 88.88888549804688 |
| PRODUCT | 11 | 11 | 11 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| TIMESTAMP | 13 | 13 | 13 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| TICKER_SYMBOL | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| PERSON | 70 | 71 | 69 | 1 | 0 | 1 | 0 | 99.29000091552734 | 97.88999938964844 | 98.58502960205078 |
| FINANCIAL_INDEX | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| FACILITY | 11 | 10 | 10 | 0 | 0 | 0 | 1 | 90.91000366210938 | 100 | 95.23859405517578 |
| PLACE | 33 | 33 | 33 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| MEDICAL_PROCEDURE | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 100 | 100 | 100 |
| GENERIC | 29 | 35 | 29 | 0 | 0 | 6 | 0 | 100 | 82.86000061035156 | 90.626708984375 |

The scoring run produces the following results:

- POS (possible) - The number of possible entities (based on key)
- ACT (actual) - The number of entities extracted during the current run.
- COR (correct) - The number of correct entities (matching the key)
- PAR (partial) - The number of partial entities, where a portion of the string overlaps with an entity from the key.
- INC (incorrect) - The number of incorrect entities, where the string overlaps exactly with an entity in the key, but the entity type is different in the current run.
- SPU (spurious) - The number of new entities not in the key.
- MIS (missing) - The number of missing entities, where the entity *is* in the key, but not in the current run.
- REC (recall) - The proportion of actual entities (from the key) that are extracted as entities (in the current run).
- PRE (precision) - The proportion of postulated entities (from the current run) that are actual entities (from the key).
- F-M (F-measure) - A weighted average of precision and recall, between 0 and 1, with 1 being a perfect score. The formula uses # = 1, meaning that precision and recall are weighted equally.

As an alternative to the breakdown table, you can click **View Detailed Score Results in Documents** in the top table to view a list of the documents that have extraction differences.




If there are no differences, the view is empty. However, a document with a red dot next to its name means that TextChart Studio has detected an extraction change.

Click each document to review the extraction changes. Click the **Differences** button above the list of documents to move between the original extraction results and the new ones. This view displays scoring changes in the center column.

# Glossary

The definitions in this list of terms that appear in i2 TextChart Studio apply only to it (and to other TextChart software). Some of the terms have different meanings in other i2 software.

### Corpus

A *corpus* is a set of documents. TextChart Studio allows a user to save the file paths to multiple corpora on the **Corpus Management** page.

### Entity

*Entities* are the important items such as persons, places, and events that TextChart finds within a document. The linguistic context of the document determines what words or phrases are extracted as entities.

Users have the ability to modify entity extraction results and to apply their own real-world knowledge. The LxBase documentation includes a list of all the types of entities that TextChart can extract.

### GxBase

TextChart uses *GxBase* to retrieve the place names and geocoordinates that it finds in a set of documents. It uses the National Geospatial Agency's GNS and the United States Geological Survey's

GNIS, as well as i2's own, internally developed gazetteer, to provide world-wide coverage and use linguistic context to decipher between ambiguous location names.

Through TextChart Studio, users can customize and import their own client-specific features.

### LxBase

The *LxBase* is the set of underlying linguistic rules and dictionaries that TextChart uses as the foundation for entity and relationship extraction.

TextChart Studio allows users to modify and add to the LxBase in order to fine tune for industry-specific extraction goals.

### Normalized form

Whether one term is equivalent to another is sometimes a choice for an individual user. For example, it might or might not be appropriate for the terms "Britain" and "England" to be considered equivalent to "United Kingdom".

When they do judge terms to be equivalent, users can arrange for TextChart to extract entities in the same, *normalized form*.

The standard TextChart dictionaries already contain many normalized lexical entries. Through TextChart Studio, users can modify and add normalized forms of their own. See Normalization management for more information.

### Relationship (or PSO)

A predicate-subject-object statement, or *PSO*, is a *relationship* between two entities established by the linguistic context.

Relationships have names of the form EntityToEntity. For example, PersonToPerson is the name of a relationship between one person entity and another.

In TextChart, every relationship has a *predicate* that describes the nature of that relationship. For example, a PersonToPerson relationship might have the predicate "interviewed". For a full list of predicate types, see the LxBase documentation.

### Semantic vector (or SV)

*Semantic vectors* represent a vector space of possible meanings for individual terms or phrases, allowing the same term to have various meanings depending on the linguistic context.

For example, a term such as "Washington" has many different semantic vectors associated with it: it might be a city name, a given name, or a surname. Some linguistic rules might even determine that it is a place.

In TextChart Studio, you can find (and modify) a list of semantic vectors and their corresponding definitions through the Token Definition Editor.

### Token

A *token* is the smallest unit of meaning that TextChart can extract from a document.

For example, a TextChart dictionary might include the term "bank" as a noun, while the term "Bank of America" is an ORG. TextChart then considers both terms as one token, because they are both listed in the dictionary.

If "Bank of America" was not listed in the dictionary, then each unit of meaning would be parsed individually, resulting in three unique tokens: "Bank", "of", and "America".

You can modify this in TextChart Studio.