

D4.4

DICTIONARY  
ENHANCEMENT  
MODULE



Author(s): Vojtěch Kovář, Adam  
Rambousek, Miloš Jakubiček

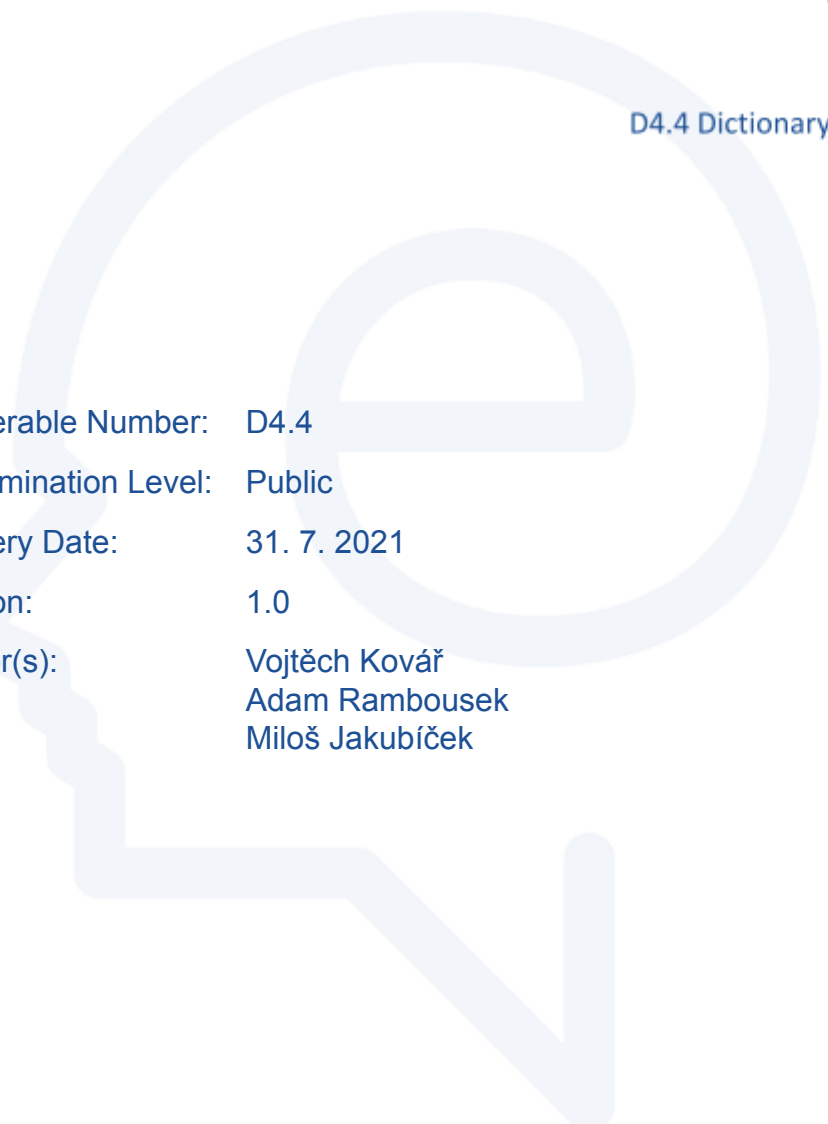
Date: 31. 7. 2021

H2020-INFRAIA-2016-2017

Grant Agreement No. 731015

ELEXIS - European Lexicographic Infrastructure

D4.4 Dictionary enhancement module



Deliverable Number: D4.4  
Dissemination Level: Public  
Delivery Date: 31. 7. 2021  
Version: 1.0  
Author(s): Vojtěch Kovář  
Adam Rambousek  
Miloš Jakubíček



Project Acronym: ELEXIS  
Project Full Title: European Lexicographic Infrastructure  
Grant Agreement No.: 731015

### Deliverable/Document Information

Project Acronym: ELEXIS  
Project Full Title: European Lexicographic Infrastructure  
Grant Agreement No.: 731015

### Document History

Version Date	Changes/Approval	Author(s)/Approved by
03/07/2021	First draft	Vojtěch Kovář
13/07/2021	Screenshots added	Adam Rambousek
31/07/2021	Final version	Miloš Jakubíček

## Introduction

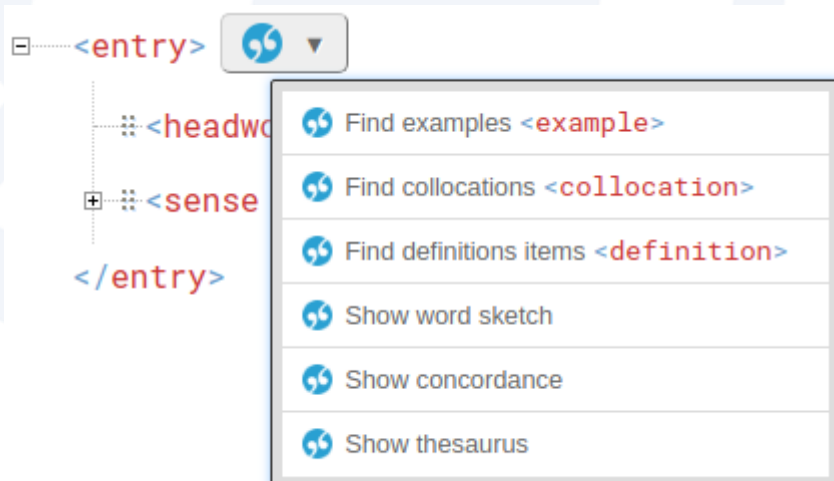
This document describes the Dictionary Enhancement Module in the ELEXIS infrastructure. The module is implemented as a series of enhancements and integration within the Lexonomy dictionary writing system and is fully integrated with other Lexonomy functionalities.

In the following text, we describe the individual improvements to the Lexonomy dictionary writing system, explain their connection and interplay with other parts of the system.

## Enhancing dictionaries with corpus resources

The Dictionary Enhancement Module provides a range of options how a lexicographer can connect with a corpus in Sketch Engine and pull raw corpus data directly into the dictionary entry they are writing. Namely, Sketch Engine can offer good dictionary examples selected by the GDEX function, corpus collocations and distributional thesaurus.

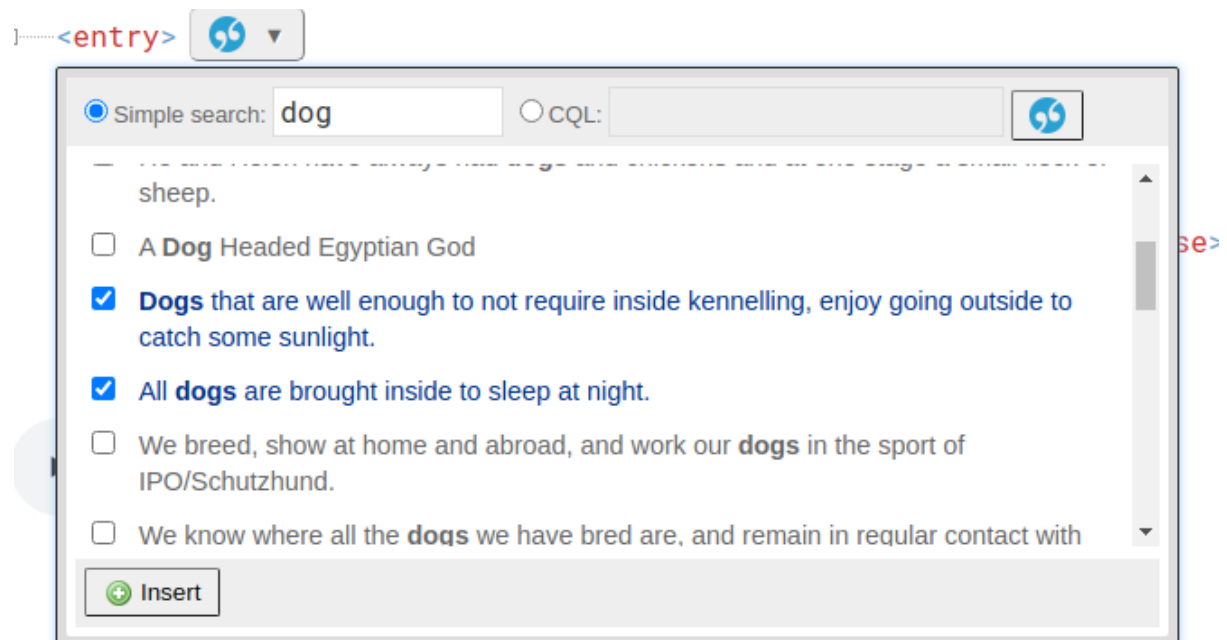
To set up connection with the Sketch Engine, users need to provide their Sketch Engine API key. Key setting is handled automatically when users log in to Lexonomy via Sketch Engine Single Sign-on feature. Consequently, users are able to select different corpus for each dictionary (e.g., based on dictionary language or domain) and map entry elements to various information from the corpus (e.g., which element in entry stores usage examples).



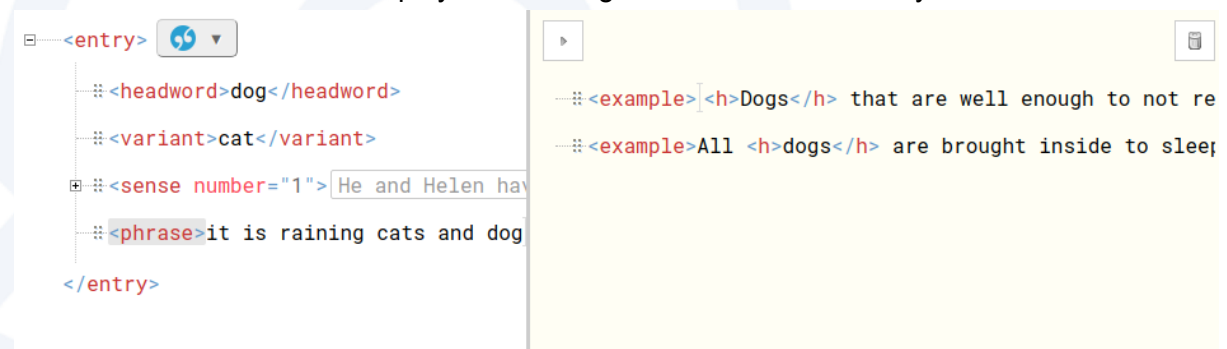
When the corpus setup is complete, users will find Sketch Engine icon in the entry editor with options to retrieve corpus data or display various information about the word.

## Corpus examples

When users want to include example sentences from the corpus, they select the “Find examples” option and the entry headword is used as a search query for GDEX function in the Sketch Engine. Users are able to use full CQL query if they need more specific examples. Users check the sentences from the list that they want to include.



All selected sentences are displayed on the right hand side of the entry editor.



Now users may move each example to sense where it's most suitable.

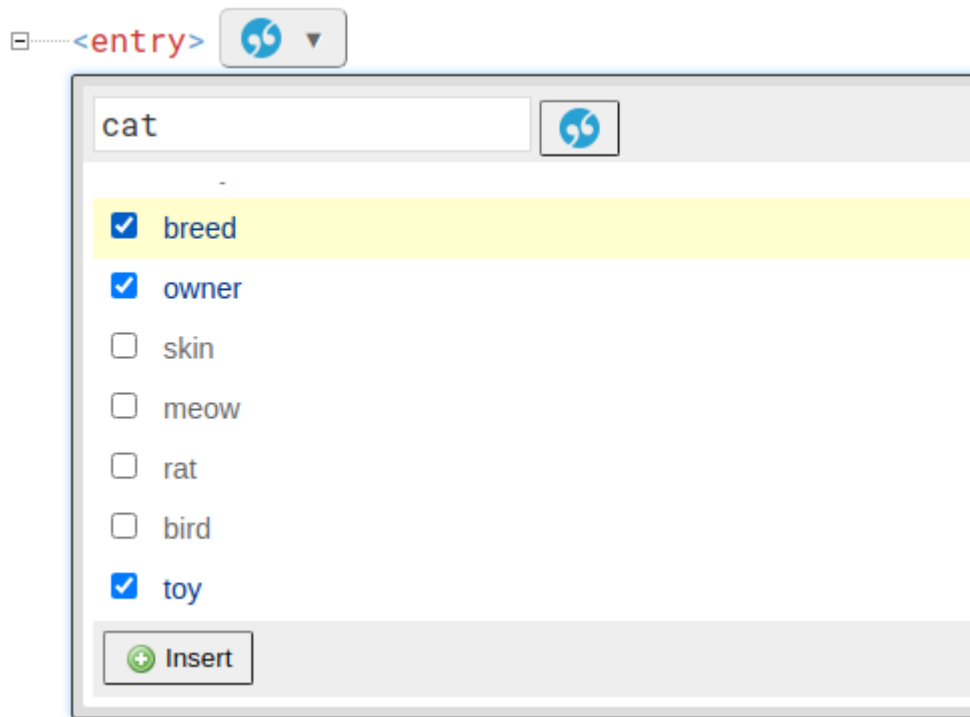
```

<entry>
  <headword>dog</headword>
  <variant>cat</variant>
  <sense number="1">
    <example>All <h>dogs</h> are brought inside to sleep at night.</example>
    <example>He and Helen have always had <h>dogs</h>
      and chickens and at one stage a small flock of sheep.</example>
  </sense>
</entry>

```

## Corpus collocations and thesaurus items

When users want to include word collocations or words from thesaurus (related words) from the corpus, they select the “Find collocations” or “Find thesaurus items” option and the entry headword is used as a search query in the Sketch Engine. Users check the collocations or thesaurus items from the list that they want to include.



Selected words are displayed in the right hand side panel of the entry editor and users move the collocations or related words to the appropriate senses.

```

<entry>
  <headword>cat</headword>
  <partOfSpeech norm="NOUN">n</partOfSpeech>
  <sense number="1">an animal with s
  <sense number="2">any of a family
  <sense number="3">
    <definition>a strong tackle use
    <collocation>breed</collocation>
  </sense>
  <sense number="4">
    <definition>a malicious woman e
    <collocation>toy</collocation>
  </sense>
  <collocation>owner</collocation>

```

## Corpus definitions

For supported languages and corpora, users are able to retrieve definitions of a word from the corpus.

The screenshot shows the entry editor interface. At the top, there is a search bar with the word "cat" and a search icon. Below the search bar, a list of definitions is displayed, each with a checkbox. The first definition is checked: "Cats are obligate carnivores: their physiology has evolved to efficiently process meat, and they have difficulty digesting plant matter." The second definition is unchecked: "Domestic cats are a major predator of wildlife in the United States, killing an estimated 1.4–3.7 billion birds and 6.9–20.7 billion mammals annually." The third definition is unchecked: "Cats are common pets in Europe and North America, and their worldwide population exceeds 500 million." The fourth definition is checked: "Feral cats are domestic cats that were born in or have reverted to a wild state." The fifth definition is unchecked: "For example, the word " cat " consists of three letters , , and , in which represents the". At the bottom of the list, there is an "Insert" button.

Selected definition sentences are displayed in the right hand side panel of the entry editor and users move the definitions to the appropriate sense for post-editing.



## Push model: one-click dictionary

It is possible to create a new dictionary and fill it with data from the Sketch Engine interface. Users will start in the Sketch Engine and its OneClick Dictionary tool. Depending on language support and user selection, the process generates a headword list with part-of-speech labels, provides candidates for example sentences, collocations, synonyms, or definitions. Subsequently, all the data are pushed into Lexonomy, where the new dictionary is created. Users are able to extend or edit the dictionary during the post-editing phase, thus saving time.

## Linking entries and senses to other dictionaries

The new linking mechanism in Lexonomy supports links between any entry elements in any dictionary. As a first step, users have to specify which entry elements should serve as the link point and how each element is identified. For example, “entry” may serve as a link point and each entry is uniquely identified with “headword + PoS”, or “sense” may be used as a link point and each sense is uniquely identified with “headword + PoS + sense number”.

### Manual linking between entries

Elements listed here can be used as target of cross-reference link. For each element, specify unique identifier in the form of placeholders ‘%(element)’. Eg. element entry can have identifier ‘%(lemma) - %(pos)’, element sense can have identifier ‘%(number)’. Optionally, specify element you want to show as preview when selecting links.

<code>&lt;entry&gt;</code>	Identifier	<input type="text" value="%(lemma)-%(pos)"/>	Preview	<input type="text"/>	<input type="button" value="✘"/>
<code>&lt;sense&gt;</code>	Identifier	<input type="text" value="%(lemma)-%(pos)-%(number)"/>	Preview	<input type="text"/>	<input type="button" value="✘"/>

When users are editing an entry, they have the option to add or view links at corresponding entry elements. When they want to add a new link, they select the target dictionary, choose which element to use in the target entry, and search for a particular link target. Source and target elements may be on a different level in an entry structure. For example, it is possible to create a link between a full entry and one sense.



The screenshot shows a user editing a dictionary entry. A link creation dialog is open over a definition element. The dialog has a 'dictionary' dropdown set to 'linking2' and a 'target' input field containing 'ar|'. A 'Link' button is visible. Below the dialog, a list of target senses is shown: 'entry: arm', 'sense: arm-1 (One of the upper limbs, from shoulder to wrist.)', and 'sense: arm-2 (A weapon.)'.

When browsing the dictionary, links are also displayed in the entry preview.

## Herrgott

*Gott, und zwar auch i. S. v. Christus und Hostie, als éin Begr. gedacht, nicht da Volke nicht gebraucht, sondern entw. «der lieb Gott» oder dann eben «der Herr*

## Incoming links

44126\_1 ← [elexis-oeaw-schranka : Herrgott : sense : 25621\\_1 \(1\)](#)

## Automatic linking of entries: NAISC

For integration with automatic linking tools, Lexonomy provides API interface to work with the cross-links. As of now, the NAISC tool (<https://github.com/insight-centre/naisc>) is available for automatic linking directly from Lexonomy. Although the process was developed with the NAISC tool, it may be easily extended to work with other tools.

The process automated link creation uses the following steps:

- user selects source and target dictionary,
- both dictionaries are converted to the OntoLex RDF format required by NAISC,
- NAISC detects the links,
- output from NAISC is converted to the internal Lexonomy format and stored in the database,
- links are available, and users may post-edit the results in Lexonomy editor.

To provide better overview, all automatic links can be displayed in single list:

elexis ZRC SAZU JSV

Edit Configure Download

## Outgoing links

### → elexis-zrcsazu-pletersnik

- [lorber](#) (3825\_1) → [lorber](#) (sense 30798\_1)
- [truplo](#) (8047\_1) → [truplo](#) (sense 88698\_1)
- [nadražiti](#) (4361\_1) → [nadražiti](#) (sense 36387\_1)
- [počasi](#) (5572\_1) → [počasi](#) (sense 54372\_1)
- [enak](#) (1885\_1) → [enak](#) (sense 11727\_1)
- [coprati](#) (1403\_1) → [coprati](#) (sense 5822\_1)
- [korar](#) (3391\_1) → [korar](#) (sense 25367\_1)
- [skriven](#) (7077\_1) → [skriven](#) (sense 77350\_1)
- [purman](#) (6415\_1) → [purman](#) (sense 69957\_1)
- [odlašati](#) (4910\_1) → [odlašati](#) (sense 44291\_1)
- [raznesen](#) (6530\_1) → [raznesen](#) (sense 71567\_1)
- [zvezati](#) (8952\_1) → [zvezati](#) (sense 101342\_1)
- [nerodovit](#) (4590\_1) → [nerodovit](#) (sense 40237\_1)
- [oblegati](#) (4807\_1) → [oblegati](#) (sense 42189\_1)
- [premilostljiv](#) (6095\_1) → [premilostljiv](#) (sense 64046\_1)
- [oves](#) (5216\_1) → [oves](#) (sense 103661\_1)
- [izdati](#) (2741\_1) → [izdati](#) (sense 18140\_1)
- [zakleti](#) (8608\_1) → [zakleti](#) (sense 96157\_1)
- [leški](#) (3711\_1) → [leški](#) (sense 29718\_1)
- [natočiti](#) (4475\_1) → [natočiti](#) (sense 38483\_1)
- [pastirica](#) (5331\_1) → [pastirica](#) (sense 103667\_1)
- [tam](#) (7753\_1) → [tam](#) (sense 85912\_1)
- [drokniti](#) (1758\_1) → [drokniti](#) (sense 10731\_1)
- [ris 1](#) (6668\_1) → [ris 1.](#) (sense 73498\_1)

## Enhancing dictionaries with multimedia

Multimedia can be an important part of a dictionary entry -- an image can be better and more intuitive than any definition, an audio pronunciation is definitely better than an IPA transcription.




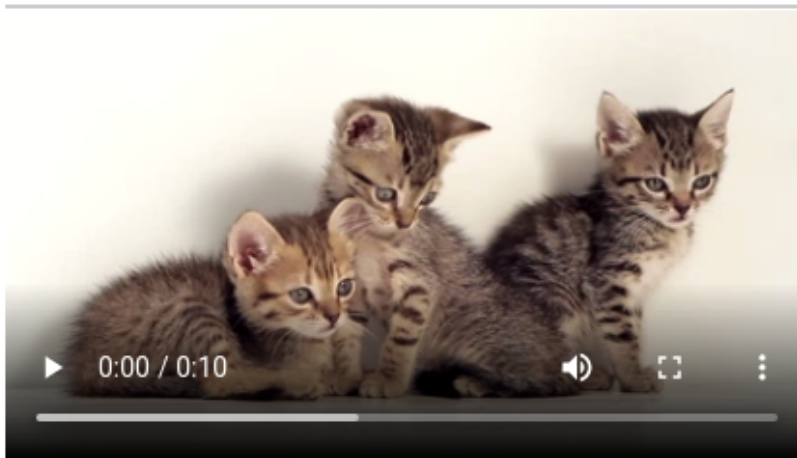
## Adding multimedia - images, video, audio

When setting up the entry structure, users may select a special element type “media” for elements that should contain multimedia data. It is not necessary to select the exact media type. When editing the entry, users only need to insert a URL link to the media file. Lexonomy will detect the media type from the URL and preview the file appropriately - i.e. images are displayed as thumbnails, video files in the video player, and audio files in the audio player.

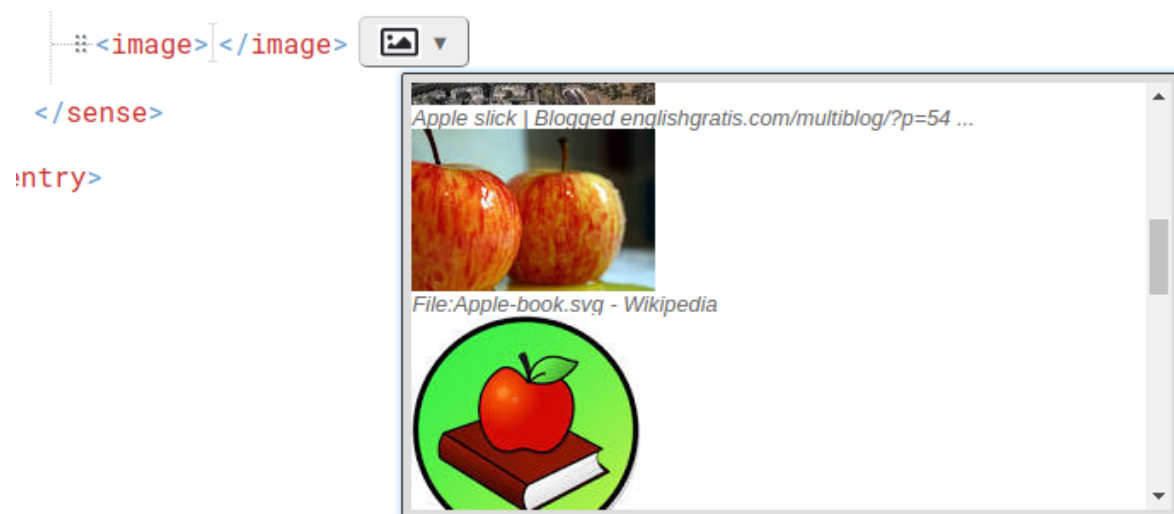
- 1 an animal with soft fur, a long thin tail, and whiskers, that people ke  
A young cat is called a kitten.

 *Birches Serviced Apartments is a pet-friendly accommodation that dog and cat .*



## Adding images automatically, while editing

To save time with getting the right images for an entry, Lexonomy now supports several image search services - Wikidata, Pixabay, and Google Image. After users configure their API keys in the settings, a new icon for image search will be displayed in the entry editor. Users may also select which licence they prefer. When users click on the image search icon, Lexonomy will search for images with selected licence and display the previews. Users only click on the selected image and the corresponding URL is inserted into the entry.



Apart from the selected image search services, more services may be added later based on the user feedback.

## Adding images automatically, whole dictionary

For users who need to add images to each entry in their dictionary, Lexonomy supports extended function to add images automatically. This feature is using the same configuration as for adding image while editing (configured API keys and licence type). Users select the element to store image links and number of images they want to add to each entry. For each entry in the dictionary, Lexonomy will search for images in all configured image services and randomly select chosen number of images. Image licence is also stored for follow-up inspection. After all entries are enhanced, users may simply delete the images they don't want.

### Auto download images to each entry



If you want to add images to each entry automatically, Lexonomy can do that for you. First, go to Entry structure and add element with content type *media*. When you're ready, select element and number of images you want to add.

Image element to add:  Add X images:  [Add images](#)

## Adding audio automatically

For selected languages, it is possible to enhance dictionaries with automatic speech synthesis. Users need to configure their VoiceRSS API key and select the language. After configuration, each entry in the dictionary will contain an audio player with automatic speech synthesis of the entry headword.

**able** *adj*

- 1 If someone is able to do something, they can do it.  
 *I'm busy tomorrow, so I won't be **able** to see you.*
- 2 If a person is able, they are good or skillful at what they do.  
 *She is an **able** teacher.*

▶ 0:01 / 0:01 ———▶ 🔊 ⋮

VoiceRSS services are used for testing and other speech synthesis services with a working API interface may be added based on the user feedback.

## Enhancing dictionaries with diachronic data

Team at the Insight SFI Research Centre for Data Analytics is preparing corpora for several languages with the diachronic data about word usage. To present the data and enhance dictionaries, Insight Centre is also creating a widget for Lexonomy that will automatically retrieve diachronic data for each headword.

## Conclusion

The Dictionary Enhancement Module within the Lexonomy dictionary writing system provides new powerful tools that enable lexicographers to enrich the entries in the existing dictionaries, in an easy and efficient way. We have described the individual parts of the module and explained how they work on the back-end, as well as for the user.

