https://doi.org/10.14618/ids-pub-11200

Dorielle Lonke/Ilan Kernerman/Vova Dzhuranyuk

LEXICAL DATA API

Abstract This API provides data from various dictionary resources of K Dictionaries across 50 languages. It is used by language service providers, app developers, and researchers, and returns data as JSON documents. A basic search result consists of an object containing partial lexical information on entries that match the search criteria, but further in-depth information is also available. Basic search parameters include the source resource, source language, and text (lemma), and the entries are returned as objects within the *results* array. It is possible to look for words with specific syntactic criteria, specifying the part of speech, grammatical number, gender and subcategorization, monosemous or polysemous entries. When searching by parameters, each entry result contains a unique entry ID, and each sense has its own unique sense ID. Using these IDs, it is possible to obtain more data – such as syntactic and semantic information, multiword expressions, examples of usage, translations, etc. – of a single entry or sense. The software demonstration includes a brief overview of the API with practical examples of its operation.

Keywords API; lexical data; search; dictionary

1. Introduction

We present a Web RESTful API [1] that offers expansive lexical data originating from diverse lexicographic resources of K Dictionaries (KD) across 50 languages, including monolingual cores as well as bilingual pairs and numerous multilingual combinations. The target users include language service providers for their translation, localization and lemmatization tasks, application developers for language learning, games and word puzzles, and researchers in the academia and industry for enhancing NLP features and training machine learning models.

The idea for this API was initially conceived in the context of LDL4HELTA – Linked Data Lexicography for High-End Language Technology Application – a Eureka bilateral project involving KD and Semantic Web Company (2015–2017) that was aimed at offering dictionary content as linked data for NLP purposes ([2], [3]), and it continued to evolve and materialize as part of a Horizon 2020 project Lynx – Legal Knowledge Graph for Multilingual Compliance Services (2017–2021, [4]) for the development of a domain-specific multilingual knowledge management platform. In its alpha phase, the API was available online only for pre-approved users who received credentials for testing and provided input on the interface and functionality. Next (2018), it was launched in a public website, allowing access to registered users on a freemium basis. This launch served as a beta phase, in which conclusions were drawn regarding the usage pattern and engagement of registered users. Despite only minimal promotion, the API attracted interested users based on keyword search. In 2019, a market-ready version was released, including new user plans and an integrated payment processing system.

Prior to release, a comprehensive market research was performed to assess the availability of similar products and services available for commercial use. It was discovered that although several other renowned dictionaries offered access to their data in similar configurations, there were limited options for comprehensive lexical data APIs that provide full coverage of a lexicographic entry [5]. Many other services existed, offering well-structured results of linguistic data mining or machine-translated keywords [6]. These types of services

This paper is part of the publication: Klosa-Kückelhaus, Annette/Engelberg, Stefan/Möhrs, Christine/Storjohann, Petra (eds.) (2022): Dictionaries and Society. Proceedings of the XX EURALEX International Congress. Mannheim: IDS-Verlag. https://doi.org/10.14618/phpy-6r66

gained popularity in recent years, as computational methods for linguistics have become more prevalent. They offer an advantage of quickly collected data in great amounts that can be used for technical applications. However, this type of data is often automatically generated and as such it has limited use for more comprehensive linguistic research. Today, there are still only a handful of services providing reliable, human-curated data that can be used for linguistics as well as computational purposes, even more so for under-resourced languages.

The API returns data as JSON documents. A basic search result consists of an object containing partial lexical information on entries that match the search criteria, but further in-depth information is also available. It is possible to look for words with specific syntactic criteria, specifying the part of speech, grammatical number, gender and subcategorization, semantic and syntactic components, multiword expressions, and monosemous or polysemous entries.

In section 2 we describe the data resources and formats, section 3 presents the technical infrastructure and functionality, and the modelling is explained in section 4. section 5 presents our dissemination modes and conclusions.

2. Data

2.1 Resources

The API offers data from three different lexicographic resources, which are regularly updated and enriched, including:

(A) A series of extensive dictionary datasets for 25 languages, all adhering to the same macrostructure and microstructure and created from scratch (except two languages that rely on exterior resources, and which were adapted accordingly). Most of these sets are multi-layered, that is containing a monolingual base with translations in bilingual and multilingual levels. [7]

(B) A series of English bilingual learner's dictionaries including versions for 45 languages. [8]

(C) A legacy English monolingual dictionary. [9]

In addition, the API incorporates privately developed lists of inflected forms with morphology techniques that enable looking up word inflections and obtaining results from their main lemmas.

2.2 Format

Each of the three resources has its own data structure.

The first (A) consists of 25 lexical datasets, each representing a different source language and altogether including nearly a hundred bilingual versions. The entry in each one has two parts: headword block and sense block. The former provides details of alternative spelling and scripts, pronunciation, grammatical details, and inflected forms; the latter conveys the senses of the entry, each usually including a definition and related semantic labels (e.g. synonym, antonym, domain, register, etc.), example(s) of usage, (and multiword expressions,) and translation equivalents for each sense, example and expression. Some translations include additional information on grammatical gender and number, geographical usage, and

This paper is part of the publication: Klosa-Kückelhaus, Annette/Engelberg, Stefan/Möhrs, Christine/Storjohann, Petra (eds.) (2022): Dictionaries and Society. Proceedings of the XX EURALEX International Congress. Mannheim: IDS-Verlag. *https://doi.org/10.14618/phpy-6r66*

irregular inflected forms. The microstructure of expressions resembles that of the entry's senses and may also include various additional lexical details as described above, as well as be divided into different senses.

The second (B) stems from a single dataset that holds together the full English entries along with a translation for each sense in each of the other 45 target languages. The entry consists of a headword container and sense containers, which roughly include similar components as those of (A), with the exception of translations for the examples of usage.

The third (C) provides rich lexical details also including headword variants, geographical and biographical names with corresponding information, notes on language, spelling and grammar, and etymology.

3. Structure and functions

3.1 Infrastructure

The API backend relies on Elasticsearch [10], which enables efficient text searches and uses stemmers for several languages, making it possible to find words when searching their inflected forms. Stemmers create a stem word from the given word, although such stem does not have to be a valid word per se. For example, *argued* or *arguing* might be stemmed to the form *argu*. Stemming helps to find the appropriate headword and is a very convenient tool for searching.

The documents are stored as objects, whose structure is defined according to the mappings provided to Elasticsearch. Each data resource has its own mappings, and those in turn can differ based on language. Other object fields such as part of speech, grammatical gender, grammatical number, inflections (included as an integral part of the source data, usually as irregular forms, not those generated by language stemmers used by Elasticsearch), are used for the search.

The backend part is hosted on Amazon Web Services (AWS), while Elasticsearch is hosted on ElasticCloud. Both platforms are highly reliable and scalable, and have undergone massive testing and validation by our team.

3.2 Functionality

The API endpoint is located at [11]. There are two main methods for querying the API. The first, GET /search, allows the user to search for entries by specifying parameters such as language and headword text. This call returns a JSON object that contains an array of results, which is a list of entries that match the search criteria. Each entry in the result array contains the unique entry ID, the headword text and part of speech, and a list of its different senses, which, in turn, include their own unique identifiers and the definition text for disambiguation purposes. The lexical information provided by this call is partial.

A basic query requires headword text and a language code, but it is possible to further specify the search by adding optional parameters such as part of speech, grammatical number, gender or subcategorization, or narrowing the search to only monosemous or polysemous entries. The language codes mostly adhere to the ISO 639-1 code convention, but it is possible to obtain a list of all language codes by querying GET /languages. Since there can be multiple entries with the same headword text (differing in part of speech, for example), this preliminary result allows the user to select the relevant option prior to obtaining the entire entry information.

The second method is GET /entries (GET /senses), which enables users to query the entire entry (sense) database across all languages, using the unique entry IDs obtained through the first method. This querying method posits that the user had already obtained the entry ID and returns various syntactic and semantic information, expressions, usage examples, translations, and more – of that single entry (or sense). The result object contains the following information:

- **id** (string) – the unique dictionary entry ID
- **source** (string) – the lexicographic resource from which the entry is taken
- **language** (string) – a two-character string that is the language code
- **version** (number) – the version of the dictionary the entry is taken from
- **related entries** (array of strings) – an array containing the IDs of the related entries
- **headword** (object/array of objects) – contains extensive phonetic and syntactic information of the headword(s)
- **senses** (array of objects) – contains an elaborate disambiguation of the headword into sense(s), including phonetic, syntactic and semantic information

The structure of the complete dictionary entry consists of two layers: the headword layer, which contains grammatical information pertaining to the entry regardless of its meaning, and the senses layer, which contains a semantic disambiguation of the word into its different meanings. Information about the usage of the word in context, such as example phrases, semantic category, register, synonyms and antonyms, and, of course, translations – which almost always depend on the particular meaning – is stored in the sense layer. To that extent, there exists a collection of senses, apart from the entry collection, within the API core, and they can be queried individually using the unique ID obtained in the initial search.

The API also includes two more complex functionalities designed to increase the number of results per search, *morph* and *analyzed.* These are Boolean parameters added to a GET / search query, and they expand the pool of results for a given word by including inflected forms or stems, disregarding diacritics and vocalization (for example in Arabic and Hebrew), and removing case-sensitivity (uppercase/lowercase). The parameters operate as follows:

- ***morph*** (boolean) – searches for the text in both headwords and inflections, including in our supplemental morphological lists. This is based on existing human-curated data and semi-automatically generated morphological lists, e.g., querying for the word *doors* will return the entry “door” (noun).
- ***analyzed*** (boolean) – this relies on a stemmer algorithm that strips words to their stem, disregarding diacritics and case (uppercase/lowercase), e.g., querying for the word *working* will return the entries *working* (adjective), *work* (verb), *work* (noun), *hard-working* (adjective), *working class* (noun), *work on* (verb), and any other entry with the stem *work* in its headword.

These functionalities were added after noticing that the previous search mechanism, which matched the exact text that was queried by the user to only the headwords in our data, was too restrictive. Opening the search for inflections and related entries consisting of the same lemma enabled a flexible query and proved economic for the users in terms of numbers of queries needed to obtain the desired results.

4. Modelling

The structure that was chosen for the API response is convenient for integrating the API in external applications, while bearing similarity to the hierarchic structure of the original XML entry due to its nested form. The motivation behind this response structure was to encapsulate all the different components of a single dictionary entry. This allows the user to choose from the various information provided for a single headword, including the equivalents in the target languages. The JSON design was initially meant to facilitate the RDF conversion from XML to JSON-LD, which had several iterations from 2014 until its culmination in 2019 (cf. [12], [13], [14], [15], [16]), but eventually proved to be useful for the API users as well.

Upon examination of user behavior, it became evident that there were benefits for providing complete dictionary information. For one, it sets this API apart from other services that focus on one aspect of a headword, like translations, examples or grammatical information. This is much better utilized for linguistic research in which syntactic and semantic information are as relevant as translations or phrases, but also carry full potential for users who need specific information, which can be easily selected and filtered from the response in the user's application. Furthermore, the single-entry response structure enabled flexible search, as it provided the user with extensive information about a particular entry, and for that entry only. In this way, users are exposed to the entirety of information that is available per entry, without having to purchase a full corpus of words, and can select only those relevant for their needs and parse for the relevant information while still receiving the fullest extent of a dictionary entry.

The decision to provide a single entry at a time, but enable access to all of the entry components, stemmed from the thought that there is no added value in otherwise providing mass amounts of headwords or translations. Services offering large corpora of headwords or translations already exist, and a similar result can be achieved by more tech-savvy users by mining existing corpora and applying machine translation tools on their own. However, while the obvious advantage of large sets of headwords or translations is the amount of data a user receives, they usually lack the crucial information needed for linguistic research or language learning applications, and also might be reduced in quality. Our lexicographic resources are compiled based on frequency lists and automatic spellcheckers, but they undergo meticulous editing by human lexicographers who select the most relevant headwords, perform additional spelling revisions and grammatical corrections, and consolidate the headword list to a uniform configuration. More details, such as part of speech, grammatical gender and number, and geographical usage are then added, further enriching the dataset, and any additional information is allocated to the corresponding components in a systematic way. This ensures that the information provided inside the *Grammatical Gender* field, for example, is indeed grammatical gender information, and the correct one for that matter, as it has been added by a competent editor. The API structure allows the user to receive an abundance of information in one place, rather than later relying on separate resources to obtain further information for a headword, and to have the assurance that the available information has been manually curated. Instead of receiving huge lists of single values, users choose which words to access and receive the entirety of grammatical, syntactic, and semantic information related to that word. In that sense, we value quality over quantity.

5. Dissemination and conclusions

The API was advertised mainly in inner circles of the lexicographic community, a couple of articles appeared in print and online [17], [18], and several hands-on workshops and software demonstrations were held in conjunction with lexicography conferences such as at EuraLex (2018, [19]) and eLex and AsiaLex (2019, [20], [21]), or the LTI summit (2019, [22]). Additionally, some SEO (search engine optimization) work was put into promoting the API and further market research pertaining to related keywords searched by potential users. During that process we discovered that public interest tended to focus more on dictionary APIs, particularly bilingual or multilingual ones, rather than on lexical data. To that extent, we invested in highlighting the lexicographic structure of the data alongside its modularity and functionality as broader lexical data to be used in other linguistics contexts. In terms of applications, it was evident that this type of product was sought after by various users, and the complete dictionary entry format proved to be useful in multiple contexts, further reinforcing the decision to model API responses as complete dictionary entries. Our user base consisted mainly of businesses who relied on the data for their own service, but there were occasionally individual users who utilized the data for their own research. This is congruent with our B2B, *Data as a Product*, business model data, but also showed that this type of dictionary data can be useful for smaller-scale products, as its modular character of a single-entry response allows for scaling up or down at the user's end. The provision of single entries per search was another strength point for the API, as it eliminated the need to acquire entire corpora or datasets and let the user select only relevant information (see section 4). For dissemination purposes, this was an advantage, as it allowed us to highlight the benefits of a full dictionary entry as a product with intrinsic value. This also allowed us to reach multiple market sectors: those interested in specific components, such as word lists or translations, could rely on these services to extract the information relevant to them; those seeking an array of lexicographic information were able to obtain a wealth of components in one place, rather than from separate services.

Future work that was considered is further developing the search mechanism to allow searching in particular components of the dictionary, such as examples, multiword expressions, or translations to a selected target language. This entails adding collections of particular components across dictionaries, which shifts the hierarchy from the traditional lexicographic sorting to a component-based filter applied on all datasets as one. A point in favor of this is the discovery that many users were only interested in certain aspects of a dictionary entry, as reflected by their queries. This would be another step in the direction of a lexical data API, as the extensive lexical information currently being offered would be accessible through a modular search structure that is not limited by the current structure of a dictionary entry result. As of now, the entirety of the lexical data is provided per dictionary and per entry, and particular information can be easily accessed by the user on their end by parsing the entry as they please. Adding other search mechanisms would not be changing the data but rather the means of delivery, possibly facilitating the usage for some users who are looking for non-lexicographic language solutions. This type of structural change reflects a shift from providing dictionary data to offering pieces of lexical information as per the user's request, without imposing the lexicographic configuration. While we are interested in exploring this direction, it is evident that there is still high interest in a dictionary API, with all the benefits of the complete lexicographic structure provided as a whole.

This paper is part of the publication: Klosa-Kückelhaus, Annette/Engelberg, Stefan/
Möhrs, Christine/Storjohann, Petra (eds.) (2022): Dictionaries and Society.
Proceedings of the XX EURALEX International Congress. Mannheim: IDS-Verlag.
https://doi.org/10.14618/phpy-6r66

References

[1] https://api.lexicala.com/.

[2] Kaltenböck, M./Kernerman, I. (2017): Introducing LDL4HELTA: Linked data lexicography for high-end language technology application. Kernerman Dictionary News 25. https://lexicala.com/review/#no25.

[3] Turdean, T./Joshi, S. (2017): Triplifying a dictionary: some learnings. In: Kernerman Dictionary News 25. https://lexicala.com/review/#no25.

[4] https://lynx-project.eu/.

[5] https://programmableweb.com/category/dictionary/api.

[6] https://programmableweb.com/news/10-popular-apis-words/brief/2021/03/12.

[7] GLOBAL series: Arabic, Chinese (Simplified and Traditional), Czech, Danish, Dutch, English, French, German, Greek, Hebrew, Hindi, Italian, Japanese, Korean, Latin, Norwegian, Polish, Portuguese (Brazilian and European), Russian, Spanish, Swedish, Thai, Turkish.

[8 PASSWORD English semi-bilingual dictionaries.

[9] Random House Webster's college dictionary.

[10] https://en.wikipedia.org/wiki/Elasticsearch.

[11] https://lexicala1.p.rapidapi.com.

[12] Klimek, B./Brümmer, M. (2015): Enhancing lexicography with semantic language databases. Kernerman Dictionary News 23. https://lexicala.com/review/#no23.

[13] Bosque-Gil, J./Gracia, J./Gómez-Pérez, A. (2016): Linked data in lexicography. In: Kernerman Dictionary News 24. https://lexicala.com/review/#no24.

[14] Aguado-de-Cea, G./Montiel-Ponsoda, E./Kernerman, I./Ordan, N. (2016): From dictionaries to cross-lingual lexical resources. In: Kernerman Dictionary News 24. https://lexicala.com/review/#no24.

[15] Bosque-Gil, J./Gracia, J./Montiel-Ponsoda, E. (2017): Towards a module for lexicography in OntoLex. In: Kernerman Dictionary News 25. https://lexicala.com/review/#no25.

[16] Lonke, D./Bosque-Gil, J. (2020): Applying the OntoLex-lemon lexicography module to K Dictionaries' multilingual data. In: Kernerman Dictionary News 28. https://lexicala.com/review/2020/lonke-bosque-gil-ontolex-lemon-lexicog/.

[17] Alper, M. (2017): KD API. In: Kernerman Dictionary News 25. https://lexicala.com/review/#no25.

[18] Kernerman, I./Lonke, D. (2019): Lexicala API: a new era in dictionary data. In: Kernerman Dictionary News 27. https://lexicala.com/review/#no27.

[19] https://euralex2018.cjvt.si/.

[20] https://elex.link/elex2019/.

[21] https://asialex2019.istanbul.edu.tr/en/_.

[22] https://www.lt-innovate.org/award.

This paper is part of the publication: Klosa-Kückelhaus, Annette/Engelberg, Stefan/Möhrs, Christine/Storjohann, Petra (eds.) (2022): Dictionaries and Society. Proceedings of the XX EURALEX International Congress. Mannheim: IDS-Verlag. *https://doi.org/10.14618/phpy-6r66*

Contact information

Dorielle Lonke
K Dictionaries
dorielle@kdictionaries.com

Ilan Kernerman
K Dictionaries
ilan@kdictionaries.com

Vova Dzhuranyuk
K Dictionaries
vova@kdictionaries.com

This paper is part of the publication: Klosa-Kückelhaus, Annette/Engelberg, Stefan/
Möhrs, Christine/Storjohann, Petra (eds.) (2022): Dictionaries and Society.
Proceedings of the XX EURALEX International Congress. Mannheim: IDS-Verlag.
https://doi.org/10.14618/phpy-6r66