EURALEX XIX

Congress of the European Association for Lexicography

Lexicography for inclusion

7-11 September 2021

Ramada Plaza Thraki
Alexandroupolis, Greece

www.euralex2020.gr

Proceedings Book
Volume 1

Edited by Zoe Gavriilidou, Maria Mitsiaki, Asimakis Fliatouras

EURALEX Proceedings

ISSN 2521-7100

ISBN 978-618-85138-1-5

Edited by: Zoe Gavriilidou, Maria Mitsiaki, Asimakis Fliatouras

English Language Proofreading: Lydia Mitits and Spyridon Kiosses

Technical Editor: Kyriakos Zagliveris

2020 Edition

Making Dictionaries Visible, Accessible, and Reusable: The Case of the Greek Conceptual Dictionary API

Giouli V., Sidiropoulos N.F.

Institute for Language and Speech Processing/ "Athena" Research Centre, Greece

Abstract

Language resources of any type are of paramount importance to several Natural Language Processing applications; developing and maintaining, however, quality lexical semantic resources is still a laborious and costly task that presents various challenges. In this respect, there is an ever-growing demand for resources that are visible, easily accessible, inter-operable and re-usable. The paper presents work in progress aimed at the development of a web service and the integration of a semantic lexical resource for Modern Greek in it, with a view to enabling robust 'search and retrieve' case scenarios. Given a lexeme, the intended service returns lexical semantic information encoded in the conceptual dictionary. The web service and the dictionary jointly form an infrastructure that can be exploited not only by researchers interested in studying the lexicon of the Modern Greek language, but also in application scenarios involving deep semantic information.

Keywords: conceptual dictionary; RESTful API; web service; accessing, querying, and re-using dictionaries

1 Introduction

Lexicographic data stored in databases constitute valuable resources that might be useful not only to human end-users but also to researchers and application developers alike. In this respect, the overwhelmingly big datasets of today ask for robust and efficient tools and services that will facilitate easy access to all sorts of language data. Currently, a variety of similar services and tools exist for well-resourced languages since most key players in the Lexicography industry have already opted to provide access to their data under a variety of licences and business models. However, it is still difficult to spot similar infrastructures for less-resourced languages. This paper presents a web Application Programming Interface (API) that enables robust "search and retrieve" case scenarios over a conceptual dictionary of Modern Greek (MG).

The remainder of the paper is structured as follows: In section (2) we provide a brief overview of trends in the development of semantic lexical resources that can be exploited for Natural Language Processing (NLP) applications and the initial endeavours of stakeholders in the industry of lexicography to provide quality data. The conceptual lexical resource that constitutes the basis of the intended infrastructure is described in section (3); the approach taken towards developing the dictionary API is presented in section (4) along with extended examples of the functionalities provided. Finally, our conclusions and plans for future research and development are outlined in section (5).

2 Background and Objectives

Over the last decades, semantic representation at word, phrase and sentence level has been the focus of attention in the field of NLP. In this context, the development of lexical resources (semantic lexica, thesauri, ontologies) coupled with information about the words and their meaning take linguistic theories of semantic representation into account. Over the last years, machine-readable hand-crafted datasets have been created primarily for the English language. Among these, WordNet (Fellbaum 1998), Verbnet (Kipper-Schuler 2005), and FrameNet (Fillmore et al. 2001) have been successfully employed in applications, whereas similar projects world-wide have resulted in resources in other languages as well.

From another perspective, the ontological approach to the representation of lexical semantics (ontology-driven lexical semantics), is largely based on the principles of Artificial Intelligence. In this context, an ontology formally specifies the concepts based on their referential status as well as the relationships that hold between these concepts. Therefore, the ontological approach to the lexical meaning is aimed at representing formally our knowledge of the entities and the relations that hold between them. In a sense, this approach to the representation of the lexical meaning reflects our grasping of entities and abstract concepts that surround us in the world as we perceive it. Prominent examples of ontologies include the OntoWordNet Project (Gagnemi et al. 2003) which has defined relations between WordNet synonym sets (synsets) as well as a set of logical, ontological and contextual commitments; similarly, SUMO (Suggested Upper Merged Ontology) ontology (Niles & Pease 2001; Pease 2011) consists of a basic upper-level ontological scheme and a number of domain-specific ontologies; it is thus one of the largest standard ontologies used in a variety of NLP applications. Similarly, Mikrokosmos (Mahesh & Nirenburg 1995a; Mahesh & Nirenburg 1995b) is a language independent ontology that is organized around the upper concepts Event, Object and Property.

Moreover, crowdsourcing techniques have been extensively employed with a view to boosting the creation of large-scale language resources and catering, thus, for the so-called knowledge acquisition bottleneck (Gale et al. 1992). The multi-lingual wiktionary dictionary (Meyer & Gurevych 2012) is developed in accordance with the principles set by Wikipedia; similarly, DBpedia is a crowd-sourced community effort to extract structured content from the information created in various Wikimedia projects. This structured information resembles an open knowledge graph which is available for everyone on the Web (Morsey et al. 2011). The resources thus constructed are freely available for research purposes via dedicated APIs.

Still, the quest for hand-crafted quality language data that are made accessible and open to interested parties is still present. In this context, stakeholders in the industry of Lexicography provide their data to prospect developers; Oxford and Meriam Webster Dictionaries APIs are pioneers in this respect.

In this framework, Greek is still under-represented, and therefore, our objective in creating the current API was two-fold: on the one hand, we wanted to render the conceptual dictionary of MG that we are developing open, visible, accessible and re-usable; on the other hand, we wanted to provide high quality lexicographic data to prospect application developers.

3 The Conceptual Dictionary of Modern Greek

The core of the infrastructure is a computational dictionary of Modern Greek (Fotopoulou & Giouli 2015). It is a conceptually organized dictionary that builds on the format proposed by an already existing lexical resource model (Markantonatou & Fotopoulou 2008). In the current implementation, the initial model was appropriately extended and modified where needed. As a result, the new architecture was defined along the following axes: (a) to ensure compatibility and interoperability with standardized tools and resources, (b) to define a robust – yet extensible – taxonomical system that could be applied consistently throughout the dictionary, (c) to account for the efficient description of the semantic properties of words and the relations that hold between them, and (d) to ensure the functionality of the final resource and the user-friendliness to the lexicographer while encoding. In this respect, the whole ontology was re-designed and extended. Initially, we adopted widely accepted standards for lexicon description (namely, the Lexical Markup Framework ISO (TC37/SC4)) and the appropriate mappings were performed through-out the resource.

In our approach, we have chosen the ontological approach to the representation of lexical semantics. Following the initial model adopted, lexical entries are represented in the dictionary as instances modelling the Saussurian notion of the linguistic SIGN and its two inseparable facets, namely, the SIGNIFIER and the SIGNIFIED (Markantonatou & Fotopoulou 2008). The final resource forms a linguistic ontology in which the linguistic SIGN is instantiated as an instance in the ontology that is represented as the unique combination of a word and a lexical concept. In this approach, words (word forms) are instances in the SIGNIFIER class; these are further specified for (a) morphosyntactic properties: grammatical category or Part of Speech (POS), gender, argument structure, and word specific information; (b) lexical relations such as word families, allomorphs, syntactic variants etc.; and (c) features pertaining to lexical semantic relations (i.e., synonymy, antonymy). Values for these features are assigned to both single- and multi-word entries in the lexicon. Finally, following common lexicographic practices, all entries are coupled with rich linguistic information, that is, gloss, one or more usage examples, register, and prior polarity information. The afore-mentioned lexicographic information is encoded via a set of relations (i.e., has_gender, has_pos, has_allomorph, has_synonym, has_antonym, etc).

3.1 The Hierarchical Ontological Schema: the SIGNIFIED Class

Similarly, word meanings (or lexical concepts) are instances in the SIGNIFIED class. Each instance in the SIGNIFIER class is mapped onto a concept, the latter represented as an instance in the SIGNIFIED class. Moreover, depending on the kind of relations defined thereof, concepts in the SIGNIFIED class are organized under two main classes: the class IS-A and the class ABOUT-A. In the current implementation, the two classes were re-designed and defined from scratch. The former class comprises a set of sub-classes that define a taxonomy hierarchically organized; concepts in this taxonomy are primarily connected via the hyponymy-hyperonymy or Is-a relation. Meronymy or part-whole relations are also encoded in this class as well. The latter class comprises a schema adopted from well-defined thesauri, namely Roget's Thesaurus (Hullen 2003) and Onomastikon (Vostantzoglou 1962). Sub-classes in the ABOUT-A class are semantically homogenous; they are defined via and populated by groups of concepts that are related via a set of semantic and pragmatic relations. In this way, words are inter-linked via a dense semantic net within or across POS categories and classes. The ontological hierarchy provides relations of inheritance, from general classes to classes that are lower in the hierarchy. As a result, the organization attempted provides for a functional and effective encoding interface, that facilitates the encoding process. The upper ontology defined within the current project is depicted in Figure 1.

Being defined as two-place predicates involving two lexical items, all relations between concepts in the ontology are further defined as inverse relations as well. For example, the relation is_the_quality_of has been paired with the inverse relation has_the_quality, and vice versa, as depicted in (1):

(1) is_the_quality_of ↔ has_the_quality

As a result, lexical instances like *εξυπνάδα* (=intelligence) and *έξυπνος* (=intelligent) are interlinked via both relations:

(2) is_the_quality_of(*εξυπνάδα*,*έξυπνος*)

(3) has_the_quality(*έξυπνος*,*εξυπνάδα*)

As noted above, the resource developed builds on existing models for semantic representation and extends the existing language resource infrastructure by integrating, harmonizing, and extending already existing lexical resources. However, the approach taken differs from them in several ways. Being primarily an ontological approach to the representation of meaning, the relations depicted are extended beyond mere lexical semantic ones. In this regard, the resource at hand differs from WordNet, a large lexical database of English in which nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. As a result, our dictionary also differs from lexical resources that were built following the WordNet paradigm for other languages – including Greek. Similarly, the conceptual dictionary adopts a FrameNet-like cognitive approach to the representation of meaning – yet the relations defined are not limited to the ones entailed by the argument structure of the predicates encoded.

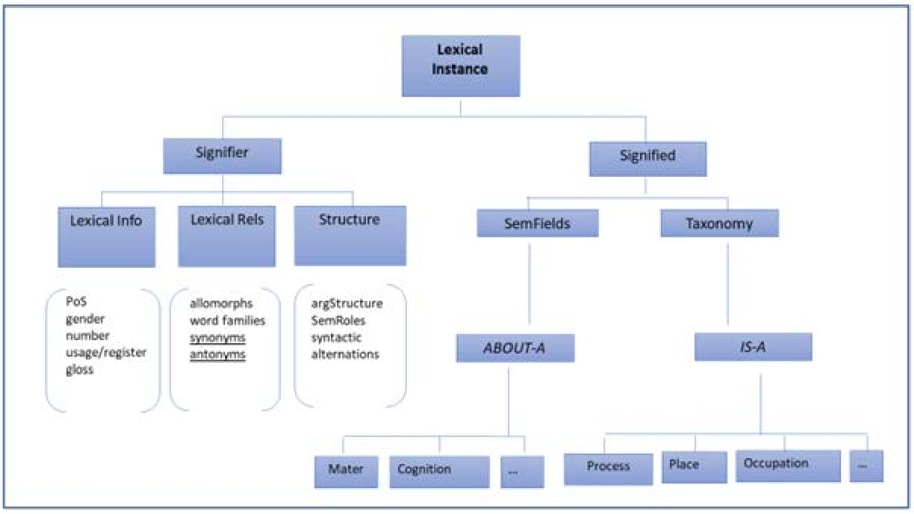

Figure 1: The search for the Greek word άνθρωπος.

3.2 The Dictionary in Numbers

The conceptual dictionary comprises lexical entries that belong to the grammatical categories Adjective, Adverb, Noun and Verb. It features both single and multi-word lexical entries, defined as lexical items which are composed by more than one lexical items and are characterized by semantic non-compositionality, syntactic non-modifiability, and the non-substitutability of components by semantically similar words. MWEs are further coupled with rich linguistic information that is useful for their robust representation in the lexical resource. To date, the dictionary comprises c. 26,5 lexical instances (entries) which have been classified under c. 250 classes (semantic fields). These instances are mapped onto ~13K concepts, whereas a set of c. 120 relations defines a dense network of linked lexical instances. The integration and handling of the relations and lexical instances developed in the API is still in progress.

POS	Single	MWEs	Total
Adjective	5214	70	5284
Adverb	1068	439	1507
Noun	13659	933	14592
Verb	3939	1191	5130
Total	23880	2633	26513

Table 1: Distribution of lexical instances per Part of Speech (PoS).

4 Creating the Dictionary API

An application programming interface (API) is seen as a computing interface which defines interactions between multiple software intermediaries. In this regard, a dictionary API is the intermediate between developers who might be in need of language data and the data providers. Therefore, an API defines the kinds of calls or requests that can be made to the data at hand, the format of the data, the conventions to follow, etc. In this section, we describe the work towards building the dictionary API and the searches foreseen so far.

4.1 Data Preparation

For the purpose of creating an interoperable tool between computer systems which would exploit the conceptually organized lexical resource, we selected the RESTFul API architecture (Fielding 2000) as an easy way to enable robust and flexible “search and retrieve” case scenarios. Since there was no need to manipulate the resource data, but only make

them available, RESTful architecture seemed ideal.
As the dictionary was built on Protégé and later moved to WebProtégé, a web application running on Tomcat Apache server in Ubuntu 14, we had to find a way to provide access to dictionary data. WebProtégé uses MongoDB, a non-relational database, making access difficult to implement, since MongoDB uses its own query language. So, we decided to find a way to migrate the data to a relational database. PHP provided the RAP-RDF API which enabled that migration. The process involved the exportation from WebProtégé of a text file in RDF/XML format. The file was next loaded into a PHP scripting page, running in an IIS web server, and then imported in a MySQL 5.5 relational database using RAP-RDF API. The main dictionary information was saved in a three-field table called 'statements' with corresponding columns "subject", "predicate" and "object". This structure is in accordance with the triplets' structure that an ontology dictionary is based on.

4.2 API Functionality

In order to develop a RESTful API we used the same IIS Web Server running on Windows. PHP was the scripting language while the use of Slim framework for PHP provided the RESTful functionality. The framework simply translates plain URL paths following the RESTful logic to corresponding SQL queries to the MySQL database, while the response to each search is provided in JSON format. Since the dictionary material is important as a digital asset that needs to be protected and regularly maintained and updated, the RESTful API will be running under an API Gateway Platform, Tyk serving also other API services. Tyk is only regulating the use of the Conceptual Dictionary API and not the data per se; this was important in view of ensuring a proper and sufficient use of the service. The Tyk platform provides the use of basic authentication parameters (username and password) to discourage improper use. For this paper and for testing our API, we have excluded the Tyk Gateway, until we analyze its usage and determine the running parameters corresponding to the load of and behavior towards our server.
A JSON result consists of blocks of data in a form of attribute-value pairs, namely the search parameter name and its corresponding value as extracted from the dictionary. In case a value consists of several value data, then a sub-block of data appears in the value place. This node-like structure permits a limitless depth of data presentation following a hierarchical presentation. Examples of JSON results will be presented in the following section where we describe all our API searches that are available as functionalities.

4.2.1 Searching the Dictionary

As mentioned above, our API provides several functionalities (searches) in the form of simple URL requests, as RESTful architecture demands. All our API requests are performed with GET method, so there is no need of sending parameters except the ones included in our URL path. There is a general word search that provides all information related to the word matches but also specific searches that retrieve part of information according to our needs.
The most simple or general search of a word is performed by a simple URL:

GET http://www.xanthi.ilsp.gr/apis/polytropon/word/<word_text>

where the <word_text> is the Greek word we search for. So, our API URL for the word άνθρωπος (anθropos, =man) will assume the following form:

GET http://www.xanthi.ilsp.gr/apis/polytropon/word/άνθρωπος

The above URL performs a search in the dictionary for the word "άνθρωπος" along with other matches of words starting with the same characters. Each matching result is returned as a block of information containing the text of the word and all dictionary information related to that word (category type, synonyms, antonyms, morphologically related words). The response is formatted in JSON (Figure 2) so it can be easily used in every environment.
To provide error control, we have the "error" parameter which appears in each result set and has a "false" value whenever the search result is successful. This is very helpful for developers that will use the API to check the status of their request result and conduct error handling. Moreover, a "message" parameter includes all necessary data retrieved by our dictionary. Each match is presented is a form of JSON block that has the same structure. This structure contains a set of parameters that contain the information encoded in the dictionary. In the current release of the dictionary API, the following parameters can be retrieved:
• **wordtxt**: Contains the word string on which search matching is performed. The number in parenthesis notes the number of similar word records with different meaning.
• **params**: Includes all possible dictionary information.
• **word_type**: Returns the value "single" for a single word and "mwe" for a phraseological unit or multi-word entry that includes the searched word.
• **PoS**: Returns the Part of Speech characterization of the matching word. Returned values are No (Noun), Ad (Adverb), Aj (Adjective), Pp (Preposition) and Vb (Verb).
• **wordstring**: Returns the plain word string.
• **number**: Returns the value "Sg" for singular and "Pl" for plural. This information in the dictionary is retained only for lexical instances that belong to the grammatical category Noun and are only plural.
• **gender**: Returns the value "Ma" for Masculine, "Fe" for feminine and "Ne" for neuter. This information is valid for lexical instances that belong to the grammatical category Noun.

• **degree**: Returns the value “Ba” for basic, “Co” for comparative and “Su” for superlative. As expected, this information is valid only for lexical instances that belong to the grammatical category Adjective or Adverb.
• **morphologically_related**: Returns an array of words that are morphologically related to the search word. It should be noted that lexical instances that are encoded as morphologically related are linked via derivation relations.
• **synonyms**: Returns an array of words characterized as synonyms of the matching word. In the current implementation, only lexical items that are absolute synonyms as opposed to near synonyms (Cruse 1986) have been encoded. To this end, the dictionary specifications cater for the distinction of the two types of synonyms, and stylistic, expressive, and structural variations are taken into account.
• **antonyms**: Returns an array of words characterized as antonyms of the matching word. According to the lexicographic specifications set, both gradable and reciprocal antonymy relations have been encoded in the dictionary.
• **instantiates:** returns lexical concepts which the word form instantiates. In case a word is polysemous, multiple lexical concepts are returned.


```json
{
    "error": false,
    "message": [
        {
            "wordtxt": "άνθρωπος (1)",
            "params": [
                {
                    "name": "word_type",
                    "value": "single"
                },
                {
                    "name": "PoS",
                    "value": "No"
                },
                {
                    "name": "wordstring",
                    "value": "άνθρωπος"
                },
                {
                    "name": "number",
                    "value": "Sg"
                },
                {
                    "name": "gender",
                    "value": "Ma"
                },
                {
                    "name": "morphologicaly_related",
                    "value": [
                        "ανθρωπάκι",
                        "ανθρωπιά",
                        "ανθρωπινά",
                        "ανθρωπινός",
                        "ανθρωπισμός",
                        "ανθρωπιστικός",
                        "ανθρωποειδές",
                        "ανθρωποκεντρικός",
                        "ανθρωποκεντρισμός",
                        "ανθρωπότητα",
                        "ολιγάνθρωπος"
                    ]
                }
            ]
        }
    ]
}
```


Figure 2: JSON output of a general Dictionary API search for the Greek word *άνθρωπος*.

The dictionary information structure contains only the existing parameters for each matching record depending in its PoS.

4.2.2 Specific Searches

Moreover, the dictionary provides several specific search functionalities. These specific request types are used to perform faster searches and retrieve small amount of data. Such a specific functionality is the type search. In this case the URL path is:

GET http://www.xanthi.ilsp.gr/apis/polytropon/word/<word_text>/<type>

where <word_text> is the greek word we search for, and <type> must have one of the following values: synonyms, antonyms, morpho and instantiates.
For synonyms of the word καλός (kalós, =good) we perform the following request:

GET http://www.xanthi.ilsp.gr/apis/polytropon/word/καλός/synonyms

The API result is presented in Figure 3. In this use case, the service returns the word αγαθός (aγaθós, =naive) as synonym.

```
{
    "error": false,
    "message": [
        {
            "wordtxt": "καλός (1)",
            "type": "synonyms",
            "words": [
                [
                    "αγαθός (1)"
                ]
            ]
        }
    ]
}
```


Figure 3: An example of synonym search result.

For the same word καλός (kalós, =good), the antonym search is performed by the URL:

GET http://www.xanthi.ilsp.gr/apis/polytropon/word/καλός/antonyms

As expected, the API returns the code in the following figure.


```
{
    "error": false,
    "message": [
        {
            "wordtxt": "καλός (1)",
            "type": "antonyms",
            "words": [
                [
                    "κακός (1)"
                ]
            ]
        }
    ]
}
```


Figure 4: An example of antonym search result.

An example of a search for only instantiates of the word βαφή (vafí, =paint) is the following URL:

GET http://www.xanthi.ilsp.gr/apis/polytropon/word/βαφή/instantiates

This request results are presented in Figure 5.


```
{
    "error": false,
    "message": [
        {
            "wordtxt": "βαφή (1)",
            "type": "instantiates",
            "words": [
                [
                    "βάψιμο",
                    "χρωμάτισμα"
                ]
            ]
        },
        {
            "wordtxt": "βαφή (2)",
            "type": "instantiates",
            "words": [
                [
                    "βάψιμο",
                    "χρωμάτισμα"
                ],
                [
                    "μπογιά"
                ]
            ]
        }
    ]
}
```


Figure 5: An example of instantiate search result.

Finally, a search example for morphologically related words of the word καρδιά (karðjá, =heart) will be performed by the URL:

GET http://www.xanthi.ilsp.gr/apis/polytropon/word/καρδιά/morpho

Such a request will result a JSON code presented in Figure 6:

```
{
    "error": false,
    "message": [
        {
            "wordtxt": "καρδιά (1)",
            "type": "morphologicaly_related",
            "words": [
                [
                    "ekfrassi_tzortzi6_Class84",
                    "καρδιά (2)",
                    "καρδιαγγειακός",
                    "καρδιογράφημα",
                    "καρδιογράφος",
                    "καρδιογραφία",
                    "καρδιολογία",
                    "καρδιολογικός",
                    "καρδιολόγος",
                    "καρδιοπάθεια",
                    "καρδιοπαθής",
                    "μυοκάρδιο",
                    "περικάρδιο"
                ]
            ]
        },
        {
            "wordtxt": "καρδιά (2)",
            "type": "morphologicaly_related",
            "words": [
                [
                    "ekfrassi_tzortzi6_Class84",
                    "καρδιά (2)",
                    "καρδιαγγειακός",
                    "καρδιογράφημα",
                    "καρδιογράφος",
                    "καρδιογραφία",
                    "καρδιολογία",
                    "καρδιολογικός",
                    "καρδιολόγος",
                    "καρδιοπάθεια",
                    "καρδιοπαθής",
                    "μυοκάρδιο",
                    "περικάρδιο"
                ],
                [
                    "καρδιά (1)"
                ]
            ]
        }
    ]
}
```


Figure 6: JSON result for morphologically related word search.

In order to provide advanced functionality as a resource beyond a simple dictionary, we have also opted for implementing a collective search. As it has already been mentioned, concepts in the dictionary are linked via hyperonymy/hyponymy relations. Being a transitive relation, hyperonymy and its inverse relation, hyponymy, link lexical instances in an hierarchical order and under a single node. This is information is deemed useful for many applications that call for reasoning and inference-making. In this regard, searches for words that are encoded as hyponyms or hypernyms of a selected word have also been enabled returning a chain of lexical instances that are linked via the hyperonymy/hyponymy relation at any depth.
This functionality is illustrated in the following URL that provides a repetitive search for hyponyms of the word "χρώμα" (chroma, =colour):

GET http://www.xanthi.ilsp.gr/apis/polytropon/word/χρώμα/hyponyms

The proposed search will return all words marked as hyponyms of the word 'χρώμα' and will return all entries denoting a colour, namely: "κόκκινο" (kokino, =red), "πράσινο" (prasino, =green), "καφέ" (kafe, =brown).The process will go on searching for hyponyms of every matching word in a depth of 2 levels by default from the initial word. The outcome is one or more lexical chains comprising the hyponyms of the searched word (Figure 7).
In case we want to perform a quicker search, we can set the depth level by submitting it in the URL request as a number, in our case:

GET http://www.xanthi.ilsp.gr/apis/polytropon/word/χρώμα/hyponyms/1

```json
{
    "error": false,
    "message": [
        {
            "word": "χρώμα",
            "level": 0,
            "hypos": [
                {
                    "word": "άσπρο χρώμα",
                    "level": 1,
                    "hypos": "No hyponyms"
                },
                {
                    "word": "χρυσό χρώμα",
                    "level": 1,
                    "hypos": "No hyponyms"
                },
                {
                    "word": "μπλε χρώμα",
                    "level": 1,
                    "hypos": "No hyponyms"
                },
                {
                    "word": "μαύρο χρώμα",
                    "level": 1,
                    "hypos": "No hyponyms"
                },
                {
                    "word": "παραλλαγή βασικού χρώματος",
                    "level": 1,
                    "hypos": "No hyponyms"
                },
                {
                    "word": "κόκκινο χρώμα",
                    "level": 1,
                    "hypos": "No hyponyms"
                },
                {
                    "word": "καφέ χρώμα",
                    "level": 1,
                    "hypos": "No hyponyms"
                },
                {
                    "word": "γκρι χρώμα",
                    "level": 1,
                    "hypos": "No hyponyms"
                },
                {
                    "word": "κίτρινο χρώμα",
                    "level": 1,
                    "hypos": "No hyponyms"
                },
                {
                    "word": "πράσινο χρώμα",
                    "level": 1,
                    "hypos": "No hyponyms"
                },
                {
                    "word": "μοβ_1 χρώμα",
                    "level": 1,
                    "hypos": "No hyponyms"
                },
                {
                    "word": "μπορντό",
                    "level": 1,
                    "hypos": "No hyponyms"
                }
            ]
        }
    ]
}
```


Figure 7: Hyponyms of word χρώμα in JSON format.

This search will provide only the words been marked as direct hyponyms of the word "χρώμα" (=colour) and them. The higher the number the heavier the search is, so a maximum of 3 level search has been proposed to be used.

Similarly, the dictionary can be searched for all relations encoded in the dictionary, and for each entry hypernyms, allomorphs, synonyms, near-synonyms, antonyms, etymologically related and semantically related words can be retrieved. A set of pre-defined relations enables semantic searches. For this kind of search we use a URL of the following syntax:

GET http://www.xanthi.ilsp.gr/apis/polytropon/word/<greek_word>/relations/<relation_string>

The <relation_string> string must use one of the values as presented on Table 2.

Relation string	Description	Relation string	Description
affliction	Disease that afflicts	has_parts	What are its parts
afflicted	Part of body that gets afflicted	is_part_of	Where do this part belongs to
affliction_entity	The one that gets afflicted	has_participant	Who is taking part
entity_affliction	He gets afflicted by	participates_in	Where is he participating
drives	What does he drives	treats	What diseases this doctor treats
driven_by	It is driven by	is_treated_by	Which doctor treats this disease
has_garment	He has that garment	practices	What does he practice
garment_used_by	From whom this garment is used for	is_practiced_by	Who practices it
has_agent	Who is the agent of this action	location_of	What takes place in this
is_agent_of	What is the action of this agent	takes_place	Where is this taking place in
has_habitat	What is the habitat of this person	workplace_of	Who works here
is_habitat_of	Whom is this habitat of	works_in	Where is he working here
has_bodypart	What are the body parts of this body part	used_by	What tool does he use
is_bodypart_of	Which body part contains this part	uses_tool	Who uses this tool
has_fruit	Which is the fruit that is being produced by	used_in	Which action uses this tool
is_fruit_of	Where is this fruit originated from	needs_tool	What tool is used in this action
has_colour	What is its color	effect	Effect
is_colour_of	Whom is this color	cause	Cause
has_member	What are its members	uses_vehicle	What vehicle does he use
is_member_of	Where do this member belongs to	vehicle_used_by	Who uses this vehicle
has_office	What is this person's office	has_property	What property does he have
is_office_of	Whom office is this	property_of	Whose property is this

Table 2: Pre-defined relation types used in API.

For example if we want to search the Greek word "ορνιθοτροφείο" (orniθotrofío, =poultry farm) with the "location_of" relation and find as the description says ("What takes place in this"), we should perform the following URL:

GET http://www.xanthi.ilsp.gr/apis/polytropon/word/ορνιθοτροφείο/relations/location_of

This request will result the following JSON code (Figure 8) where the related word "εκτροφή" (ektrofí ,=breeding) is returned from our dictionary.


```
{
    "error": false,
    "message": [
        {
            "wordtxtword": "ορνιθοτροφείο",
            "relation": "location_of",
            "related_words": [
                "εκτροφή"
            ]
        }
    ]
}
```


Figure 8: Relation type result in JSON format.

5 Conclusion

We have presented an infrastructure, currently under development, for exploiting a variety of lexical resources. Currently, the infrastructure will facilitate easy and robust access to a conceptual dictionary of MG. The service is targeted to researchers of MG and to application developers in need of linguistically aware lexical resources. In the future, we envisage the creation of an interface so that the resource is usable by end-users as well. In this context, we are planning to enable interactive access to the data and extending the resources and services via crowdsourcing. Finally, the integration of other lexical resources into the infrastructure is also planned; in this context, linking of the resource at hand with other language data will be examined.

6 References

Fellbaum, C. (ed.). (1998). *WordNet: An electronic lexical database*. The MIT Press.
Fielding, R. (2000). Architectural styles and the design of network-based software architectures, Ph.D. thesis, University of California.
Fillmore, C., Wooters, C. & Baker, K. (2001). Building a Large Lexical Databank Which Provides Deep Semantics. In *Proceedings of the Pacific Asian Conference on Language, Information and Computation*. Hong Kong. Accessed at: http://framenet.icsi.berkeley.edu/papers/dsemlex16.pdf [21 /06/ 2007].
Fotopoulou, A., Giouli, V. (2015). From Ekfrasis to Polytropon: conceptual design of Lexical Resources. In *Proceedings of the 12th International Conference in Greek Linguistics* (ICGL12). Berlin, Germany.
Gale, W. A., Church, K.W. & Yarowsky. D. (1992). A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5-6),415.
Gangemi A., Navigli R. & Velardi P. (2003). The OntoWordNet Project: Extension and Axiomatization of Conceptual Relations in WordNet. In R. Meersman, Z. Tari, D.C. Schmidt (eds) *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003*. Lecture Notes in Computer Science, vol 2888. Springer, Berlin, Heidelberg.
Guarino, N., Welty, C. (2004). An Overview of OntoClean. In S. Staab, R. Studer (eds.) *The Handbook on Ontologies*. pp. 151-172. Berlin:Springer-Verlag.
Hullen, W. (2003). *A History of Rogets Thesaurus: Origins, Development, and Design*. Oxford University Press.
Kipper-Schuler, K. (2005). VerbNet: A broad-coverage, comprehensive verb lexicon. PhD thesis, Computer and Information Science Dept., University of Pennsylvania, Philadelphia, PA.
Mahesh, K., Nirenburg, S. (1995a). A situated ontology for practical NLP. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence* (IJCAI-95), Montreal, Canada.
Mahesh, K., Nirenburg, S. (1995b). Semantic classification for practical natural language processing. In *Proceedings of the 6th ASIS SIG/CR Classification Research Workshop: An Interdisciplinary Meeting*, Chicago, IL.
Markantonatou S., Fotopoulou, A. (2007). The tool Ekfrassi. In *Proceedings of the 7th International Conference on Greek Linguistics*, The Lexicography Workshop, September 8-10, 2007, University of Ioannina, Greece [in Greek].
Meyer, C. M., Gurevych, I. (2012). Wiktionary: A new rival for expert-built lexicons? Exploring the possibilities of collaborative lexicography. In S. Granger, M. Paquot (Eds.): *Electronic Lexicography*, Oxford: Oxford University Press, pp. 259–291.
Navigli, R., Ponzetto, S. (2012). BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. Artificial Intelligence, 193, Elsevier, pp. 217-250.
Niles, I., Pease, A. (2001). Toward a Standard Upper Ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001).*
Pease, A. (2011). *Ontology: A Practical Guide*. Articulate Software Press, Angwin, CA.
Vostantzoglou, Th. (1962). *Antilexicon i Onomasticon tis neoellinikis*, Athens.

Acknowledgements

The authors would like to thank the anonymous reviewers for their comments that contributed to improving the manuscript. The research leading to the results presented here was partially funded by the project "Computational Science & Technologies: Data, Content and interaction" (MIS 5002437), which was co-financed by Greece and the EU, and the project "AIO IEL – Strategic Planning and Infrastructures of ILSP".