

Lexical Chains Finder Web Service

Table of Contents

1	BASIC INFORMATION.....	2
1.1	Web Service name	2
1.2	Overview and purpose of the Web Service.....	2
1.3	A short description of the service	2
2	TECHNICAL INFORMATION	2
2.1	WSDL code for invoking the web-service	2
2.2	Software dependencies and system requirements (if any)	2
2.3	Execution instructions.....	3
2.4	Input / Output data formats	3
2.5	Integration with external tools	3
3	CONTENT INFORMATION	4
3.1	An usage example with associated data.....	4
3.2	An example of the output data	4
3.3	Approximation of the time necessary to process the test input file	4
4	ADMINISTRATIVE INFORMATION	4
4.1	Contact	4
5	REFERENCES	4

1 BASIC INFORMATION

1.1 Web Service name

The WebService is called **LexChains** which is the short for “Lexical Chains”.

1.2 Overview and purpose of the Web Service

LexChains generates lexical chains between words in English or English Princeton WordNet 3.0 (Fellbaum, 1998) (PWN) concepts, using the PWN structure. It is developed in C# as a REST web service accessible at:

<http://khufu.racai.ro:8001/lexchains.ashx>

1.3 A short description of the service

Lexical Chains as returned by this service are described in Ion and Ștefănescu (2011) and Ștefănescu (2010).

The term “lexical chain” refers to a set of words which, in a given context (sentence, paragraph, section and so on), are semantically related to each other and are all bound to a specific topic. For instance, words like “tennis”, “ball”, “net”, “racket”, “court” all may form a lexical chain if it happens that a paragraph in a text contains all of them.

Moldovan and Novischi (2002) used an extended version of the PWN (XWN¹) to derive lexical chains between the meanings of two words by finding relation paths in the PWN hierarchy.

In a similar fashion, the LexChains Web Service exploits the PWN structure returning such lexical chains for a given pair of words or concepts. In this paradigm, a lexical chain is not simply a set of topically related words, but becomes a path of synsets in the PWN hierarchy. This is the derived definition that we have used to implement our version of lexical chains as meaning paths through PWN.

2 TECHNICAL INFORMATION

2.1 WSDL code for invoking the web-service

This is a *REpresentational State Transfer* (REST) Web Service, for which there is no WSDL code.

2.2 Software dependencies and system requirements (if any)

There is no need for local installations and there are no software dependencies or system requirements for this tool, since it is hosted by a remote machine (khufu.racai.ro).

¹ <http://xwn.hlt.utdallas.edu/>

2.3 Execution instructions

Since this is a REST Web Service, it can be directly accessed by users or applications. This can be done in multiple ways:

- i. given 2 words as input
e.g.: <http://khufu.racai.ro:8001/lexchains.ashx?w1=tree&w2=leaf>
The user must use GET parameters *w1* and *w2*
- ii. given 2 words and their part of speech as input
e.g.: <http://khufu.racai.ro:8001/lexchains.ashx?w1=farmer&pos1=n&w2=grow&pos2=v>
The user must use GET parameters *w1*, *w2*, *pos1* and *pos2*
- iii. given 2 concepts as input, by using the Inter Lingual Indexes (ILIs) as codification
e.g.: <http://khufu.racai.ro:8001/lexchains.ashx?ili1=09450163-n&ili2=09394007-n>
The user must use GET parameters *ili1* and *ili2*

The user has also the possibility to limit the number of chains returned by the service by using GET parameter *max*; e.g.:

<http://khufu.racai.ro:8001/lexchains.ashx?ili1=09450163-n&ili2=09394007-n&max=10>

2.4 Input / Output data formats

The Input requires the GET parameters to be correctly set.

The Output has the following format (see Fig. 1):

<word_1/concept_1> (<PWN_relation_1>) <intermediary_concept_1> (<PWN_relation_2>)
<intermediary_concept_2> ... (<PWN_relation_k>) <intermediary_concept_k>
<word_2/concept_2>

<concept_i> and <intermediary_concept_i> have the following format (see Fig. 1):

ILI[literal_1#sense_no(pos), ... , literal_n#sense_no(pos)]

O

```
09450163-n[sun#1(n),Sun#1(n)] (member_holonym) 09439433-n[solar_system#1(n)] (member_holonym) 09394007-n[planet#1(n),major_planet#1(n)]
09450163-n[sun#1(n),Sun#1(n)] (member_holonym) 09439433-n[solar_system#1(n)] (member_meronym) 09394007-n[planet#1(n),major_planet#1(n)]
09450163-n[sun#1(n),Sun#1(n)] (member_holonym) 09439433-n[solar_system#1(n)] (member_meronym) 09381480-n[outer_planet#1(n)] (hyponym) 09
09450163-n[sun#1(n),Sun#1(n)] (member_meronym) 09439433-n[solar_system#1(n)] (member_holonym) 09394007-n[planet#1(n),major_planet#1(n)]
09450163-n[sun#1(n),Sun#1(n)] (member_holonym) 09439433-n[solar_system#1(n)] (member_holonym) 09381480-n[outer_planet#1(n)] (hyponym) 09
09450163-n[sun#1(n),Sun#1(n)] (member_holonym) 09439433-n[solar_system#1(n)] (member_meronym) 09355623-n[minor_planet#1(n),planetoid#1(n)
n[planet#1(n),major_planet#1(n)]
09450163-n[sun#1(n),Sun#1(n)] (member_holonym) 09439433-n[solar_system#1(n)] (member_meronym) 09355623-n[minor_planet#1(n),planetoid#1(n)
n[planet#1(n),major_planet#1(n)]
09450163-n[sun#1(n),Sun#1(n)] (member_holonym) 09439433-n[solar_system#1(n)] (member_meronym) 09394007-n[planet#1(n),major_planet#1(n)]
n[planet#1(n),major_planet#1(n)]
```

Figure 1: LexChains Output format

2.5 Integration with external tools

LexChains is fully self-contained.

3 CONTENT INFORMATION

3.1 An usage example with associated data

Examples of how to use this Web Service are given in Section 2.3

3.2 An example of the output data

An Output example for query (see also Fig. 1):

<http://khufu.racai.ro:8001/lexchains.aspx?ili1=09450163-n&ili2=09394007-n&max=5>

```
09450163-n[sun#1(n),Sun#1(n)] (member_holonym) 09439433-n[solar_system#1(n)]
(member_holonym) 09394007-n[planet#1(n),major_planet#1(n)]
09450163-n[sun#1(n),Sun#1(n)] (member_holonym) 09439433-n[solar_system#1(n)]
(member_meronym) 09394007-n[planet#1(n),major_planet#1(n)]
09450163-n[sun#1(n),Sun#1(n)] (member_holonym) 09439433-n[solar_system#1(n)]
(member_meronym) 09381480-n[outer_planet#1(n)] (hyponym) 09394007-
n[planet#1(n),major_planet#1(n)]
09450163-n[sun#1(n),Sun#1(n)] (member_meronym) 09439433-n[solar_system#1(n)]
(member_holonym) 09394007-n[planet#1(n),major_planet#1(n)]
09450163-n[sun#1(n),Sun#1(n)] (member_holonym) 09439433-n[solar_system#1(n)]
(member_holonym) 09381480-n[outer_planet#1(n)] (hyponym) 09394007-
n[planet#1(n),major_planet#1(n)]
```

3.3 Approximation of the time necessary to process the test input file

LexChains Web Service is hosted on a 8-core Intel(R) Xeon(R) x64 CPU E5504 @ 2.00GHz and 8 GB of RAM. Running time for a query is limited to maximum 100 miliseconds.

4 ADMINISTRATIVE INFORMATION

4.1 Contact

For further information, please contact Dan ȘTEFĂNESCU (<http://www.racai.ro/~danstef/>; danstef@racai.ro).

5 REFERENCES

Fellbaum, C. (1998). WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

Ion, R., Ștefănescu, D. (2011). Unsupervised Word Sense Disambiguation with Lexical Chains and Graph-Based Context Formalization. In Zygmunt Vetulani (ed.): LTC 2009, Lecture Notes in Artificial Intelligence, Volume 6562/2011, pp. 435—443, Springer, Heidelberg

Moldovan, D., Novischi, A. (2002). Lexical chains for question answering. In: Proceedings of the 19th International Conference on Computational Linguistics, August 24 – September 01, 2002, Taipei, Taiwan, pp. 1-7.

Ștefănescu, D. (2010). Intelligent Information Mining from Multilingual Corpora. PhD thesis (in Romanian). Romanian Academy, Bucharest.