

D2.3

Lexical Resource

Linking Service

Author(s):

John P. McCrae (NUIG)

Sina Ahmadi (NUIG)

Lenka Bajčetić (ÖAW)

Seung-bin Yim (ÖAW)

Date: 31. 01. 2021

H2020-INFRAIA-2016-2017

Grant Agreement No. 731015

ELEXIS - European Lexicographic Infrastructure

### D2.3 Lexical Resource Linking Service

Deliverable Number: D2.3

Dissemination Level: Public

Delivery Date: 31. 1. 2021

Version: 1.0

Author(s): John P. McCrae (NUIG),  
Sina Ahmadi (NUIG),  
Lenka Bajčetić (ÖAW),  
Seung-bin Yim (ÖAW)



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
---------	------	------------------	-----------------------





# Table of Contents

Executive Summary.....	1
1 Introduction .....	2
2 Architecture .....	4
3 Text Similarity Documents .....	7
3.1 Basic Methods.....	7
3.2 Beyond similarity .....	8
3.2.1 Approach.....	9
3.2.1.1 Feature Extraction.....	9
3.2.1.2 Feature learning and classification .....	11
4 Experiments .....	12
5 Deep learning methods.....	14
5.1.1 Fine-tuning Transformer models .....	14
5.1.1.1 Preprocessing.....	15
5.1.1.2 Model training.....	15
5.1.1.2.1 Training Environment .....	15
5.1.1.2.2 Hyperparameters.....	15
5.1.1.2.3 Loss Function .....	16
5.1.1.3 Model Evaluation .....	16
6 Non-textual Linking Methods .....	18
7 Linking Constraints.....	20
7.1 Bijection .....	20
7.2 b-Matching.....	21
7.3 Taxonomic.....	22
8 Benchmarks and Shared Task .....	24
9 Summary .....	27
10 Bibliography .....	28





## Executive Summary

This deliverable describes the development of a system for linking between two lexical resources of the same language in a task that we call Monolingual Word Sense Alignment (MWSA). We have developed a benchmark for this task, which covers 15 languages and contains on average about 1,000 annotations for each language. We have also developed a toolkit for linking called ‘Naisc’, which provides a flexible and extensible architecture for sense linking. We have investigated methods for sense similarity based on comparing the definition text, using both traditional frequency-based features and state-of-the-art machine learning approaches including restricted Boltzmann machines and the BERT transformer model. We have also looked into exploiting the graph and non-textual features to improve linking. Finally, we look at the problem of linking holistically, ensuring that the linking between a set of senses does not introduce any logical contradictions.



## 1 Introduction

Sense linking is the task of inferring from the list of senses in two dictionaries any potential relationships between them. For this task, we categorize the linking into exact equivalence, partial equivalence (broader/narrower) and any other relation between senses. This formulates the problem as a five-class<sup>1</sup> classification for each pair of senses between the two dictionary entries. For this deliverable, the work is limited to the case where the dictionaries are in the same language and thus that we are only matching senses whose headword matches exactly, we call this task Monolingual Word Sense Alignment (MWSA). The multilingual case will be tackled in D2.4.

We have built in tools for this task into an existing framework called Naisc<sup>2</sup> and we will first describe the architecture of this system, which will interact with the rest of the project through the interfaces defined in D2.2. Next, we will look at methods of linking that rely on the text of the definitions to link, firstly looking at some basic methodologies and then implementing methods that use deep learning models such as BERT (Devlin et al. 2018). We will then look at methods that can exploit non-textual information about the senses in a meaningful way. The next step is to look at the challenge of inferring links holistically, taking into account that the links inferred by direct comparison of the definitions may lead to logical contradictions, e.g., multiple senses being equivalent to a single target sense. Finally, we will document the creation of a test set for this MWSA task that covers 17 dictionary pairs in 15 languages and some results for our systems on this benchmark.

### Publications documented in this deliverable:

- McCrae, J.P. and Buitelaar, P. (2018). Linking Datasets Using Semantic Textual Similarity. John P. McCrae and Paul Buitelaar, *Cybernetics and Information Technologies*, 18(1), pp 109-123.
- Ahmadi, S., Arcan, M., & McCrae, J. (2019, May). Lexical sense alignment using weighted bipartite b-matching. In the 2nd *Conference on Language, Data and Knowledge (LDK 2019)*, Germany.

---

<sup>1</sup> equal, broader, narrower, related, unrelated

<sup>2</sup> <https://github.com/insight-centre/naisc>

'naisc' means links in Irish and is pronounced 'nashk'



---

### D2.3 Lexical Resource Linking Service

- Ahmadi, S., McCrae, J. P., Nimb, S., Khan, F., Monachini, M., Pedersen, B. S., ... & Gabrovsek, D. (2020, May). A multilingual evaluation dataset for monolingual word sense alignment. In *Proceedings of the 12th Language Resource and Evaluation Conference (LREC 2020)*, France.
- Salgado, A., Ahmadi, S., Simões, A., McCrae J.P. and Costa, R. (2020, May) Challenges of Word Sense Alignment: Portuguese Language Resources., *Proceedings of the 7th Workshop on Linked Data in Linguistics: Building tools and infrastructure at LREC 2020*, pp 45-51.
- Bajčetić, L. and Yim, S.B., (2020). Implementation of Supervised Training Approaches for Monolingual Word Sense Alignment:ACDH-CH System Description for the MWSA Shared Task at GlobalLex 2020. *Proceedings of the Globalex Workshop on Linked Lexicography (@LREC 2020)*.
- Ahmadi, S. McCrae, J. P. (2021, January). Word Sense Alignment as a Classification Problem. In *Proceedings of the 11th International Global Wordnet Conference (GWC)*, South Africa.



## 2 Architecture

Blabla

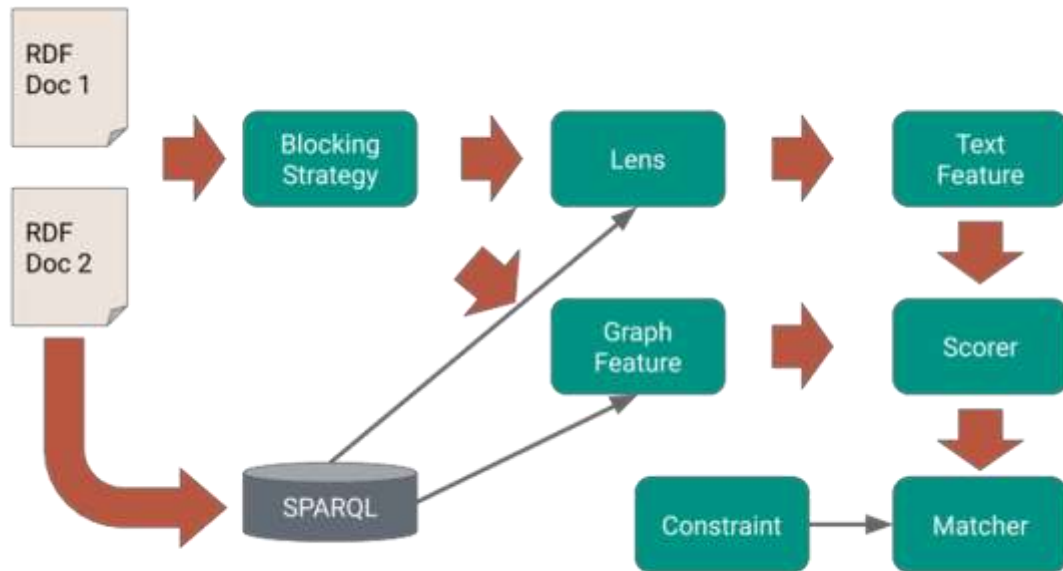


Figure 1: The Architecture of the Naisc system for sense linking

The Naisc architecture is depicted in Figure 1. The architecture of Naisc was originally designed for linking any RDF datasets and this can be applied to the MWSA task by converting the dictionaries into an RDF format such as OntoLex (McCrae, Buitelaar, and Cimiano 2017; Cimiano, McCrae, and Buitelaar 2016). The process of linking is broken down into a number of steps that are described as follows:

- **Blocking:** The blocking step finds the set of pairs that are required to be linking. For more general linking tasks and for the multilingual linking task this is quite challenging and error-prone. However, for the MWSA task we only link on matching headwords so the blocking task has a single implementation that simply finds matching headwords and outputs every sense pair between these two entries.

**Signature:** (Dataset, Dataset)  $\Rightarrow$  (Sense, Sense)\*
- **Lens:** The lens examines the data around the sense pair to be linked and extracts text that can be compared for similarity. Clearly, the most important lens for this task extracts the senses'

### D2.3 Lexical Resource Linking Service

definitions. However, other information such as examples can also be extracted here.

**Signature:** (Sense, Sense)  $\Rightarrow$  (Text, Text)

- **Text features:** The text features extract a set of similarity judgements about the texts extracted with the lenses and are described in more detail in the following section.

**Signature:** (Text, Text)  $\Rightarrow \mathbb{R}^*$

- **Graph features:** Graph (or non-textual) features do not rely on the text in the dataset but instead look at other features. They are described in more detail later in the document.

**Signature:** (Sense, Sense)  $\Rightarrow \mathbb{R}^*$

- **Scorer:** From a set of features extracted either from the text or from other graph elements, a score must be estimated for each of the sense pairs. This can be done in either a supervised or unsupervised manner and we implement standard methods for supervised classification such as SVMs and unsupervised classification using voting.

**Signature:**  $\mathbb{R}^* \Rightarrow [0,1]^*$  - *Output corresponds to a probability distribution over the relation classes*

- **Matcher and Constraint:** There are normally some constraints that we wish to enforce on the matching and these are applied by the matcher

**Signature:** (Sense, Sense,  $[0,1]^*$ )\*  $\Rightarrow$  (Sense, Sense)\* - *Output is a subset of the input*

Naisc is implemented in Java and the configuration of each run can be specified by giving a JSON description of the components that can be used. For example, this is a default configuration for the MWSA task (presented using YAML syntax):

```
blocking:
  name: blocking.OntoLex
lenses:
- name: lens.Label
  property:
  - http://www.w3.org/2004/02/skos/core#definition
  id: label
textFeatures:
- name: feature.BasicString
  wordWeights: models/idf
  ngramWeights: models/ngidf
  labelChar: true
- name: feature.WordEmbeddings
  embeddingPath: models/glove.6B.100d.txt
```



---

### D2.3 Lexical Resource Linking Service

#### scorers:

```
- name: scorer.LibSVM
  modelFile: models/default.libsvm
```

#### matcher:

```
  name: matcher.BeamSearch
  constraint:
    name: constraint.Taxonomic
```

```
description: The default setting for processing two OntoLex-compliant dictionaries
```

This configuration assumes that the dictionary is in the OntoLex format for blocking and processes it as such, it then extracts the definitions using the ‘Label’ lens and applies both some basic string text features as well as text features based on GloVe vectors (Pennington, Socher, and Manning 2014). The scores for each property type are calculated using LibSVM (Chang and Lin 2011) and finally the overall linking is calculated using the taxonomic constraints, which will be defined later in this document.



### 3 Text Similarity Documents

The comparison of the definitions of the lexical entries is the most obvious and effective method for establishing similarity between senses in two dictionaries and is the primary method that humans would use. As such, it makes sense to focus our efforts on developing an artificial intelligence approach to the task of estimating the similarities of definitions, which is a kind of Semantic Textual Similarity (STS) as explored in tasks at SemEval (Agirre et al. 2016). We have explored three main approaches to this, firstly using simple text features to provide a baseline for the task. Secondly, we use deep learning methods including BERT and finally we move beyond simple similarity to also predict the taxonomic type of the relationship between senses.

#### 3.1 Basic Methods

The basic methods use frequency and surface forms of the strings to compute features, the following methods are implemented by the Naisc tool. Most of these methods can work on words or on characters.

- **Longest common subsequence:** The longest subsequence of words (characters) that match between the two strings as a ratio to the average length between the two strings.
- **Longest common prefix/suffix:** The longest subsequence of words (characters) from the start/end of each string, as a ratio to the average length.
- **N-gram:** The number of matching subsequences of words (characters) of length  $n$  between the two strings as a ratio to the average maximum number of  $n$ -grams that could match (e.g. length of string minus  $n$  plus one)
- **Jaccard/Dice/Containment:** The match between the words of the two definitions using the Jaccard and Dice coefficients. Let  $A$  and  $B$  be the set of words in each definition:  

$$Jaccard = \frac{|A \cap B|}{|A \cup B|}, Dice = \frac{2|A \cap B|}{|A| + |B|}, Containment = \frac{|A \cap B|}{\min(|A|, |B|)}$$
- **Sentence Length Ratio:** The ratio of the length of the sentences as  $SLR(s, t) = 1 - \frac{\min(|x|, |y|)}{\max(|x|, |y|)}$
- **Average Word Length Ratio:** The ratio of the average word length in each sentence normalized to the range  $[0, 1]$  as for SLR
- **Negation:** Whether either both sentences contain negation words<sup>3</sup> or both don't (1 if true, 0 if false)
- **Number:** If both sentences contain numbers do these numbers match (1 if all numbers match)

<sup>3</sup> The negation words for English are: not, never, neither, nor, nobody, nothing, don't, won't, can't, doesn't, aren't, isn't, haven't



### D2.3 Lexical Resource Linking Service

- **Jaro-Winkler, Levenshtein:** Standard string similarity functions, we use the Apache Commons Text implementations<sup>4</sup>.
- **Monge-Elkan:** This is defined as follows where *sim* is a word similarity function (we use either Jaro-Winkler or Levenshtein)

$$ME(s, t) = \frac{1}{|s|} \sum_{i=1}^{|s|} \max_{j=1, \dots, |t|} sim(s_i, t_j)$$

In addition, we implement the following approach based on using GloVe vectors (Pennington, Socher, and Manning 2014), where we calculate the word embeddings for each word in the two definitions and then compare pairwise the words of each definition. These are turned into a single feature using methods described in McCrae and Buitelaar (McCrae and Buitelaar 2018).

## 3.2 Beyond similarity

Dictionaries are valuable resources which document the life of words in a language from various points of view. Senses, or definitions, are important components of dictionaries where dictionary entries, i.e. lemmata, are described in plain language. Therefore, unlike other properties such as references, comparisons (*cf.*), synonyms and antonyms, senses are unique in the sense that they are more descriptive but also highly contextualized. Moreover, unlike lemmata which remain identical through resources in the same language, except in spelling variations, senses can undergo tremendous changes based on the choice of the editor, lexicographer and publication period, to mention but a few factors. Therefore, the task of word sense alignment (WSA) will facilitate the integration of various resources and the creation of inter-linked language resources.

Considering the literature, various components of the WSA task have been matters of research previously. However, a few of the previous papers address WSA as a specific task on its own. As a preliminary study, our focus is on providing explainable observations for the task of WSA using manually-extracted features and analyzing the performance of traditional machine learning algorithms for word sense alignment as a classification problem. Despite the increasing popularity of deep learning methods in providing state-of-the-art results in various NLP fields, we believe that

---

<sup>4</sup> <https://commons.apache.org/proper/commons-text/>





### D2.3 Lexical Resource Linking Service

evaluating the performance of feature-engineered approaches is an initial and essential step to reflect the difficulties of the task and also, the expectations from the future approaches.

We define our task of *WSA* and semantic induction as the detection of the semantic relationship between a pair of senses in two monolingual resources, as follows:

$$rel = sem(p, s_i, s_j)$$

where  $p$  is the part-of-speech of the lemma,  $s_i$  and  $s_j$  are senses belonging to the same lexemes in two monolingual resources and  $rel$  is a semantic relation, namely exact, broader, narrower, related and none. Our goal is to predict a semantic relation, i.e.  $rel$  given a pair of senses. Therefore, we define three classification problems based on the relation:

- **Binary classification** which predicts if two senses can possibly be aligned together. Otherwise, none is selected as the target class.
- **SKOS classification** which predicts a label among exact, broader, narrower and related semantic relationships.
- **SKOS+none classification** which predicts a label given all data instances. This is similar to the previous classifier, with none as a target class.

#### 3.2.1 Approach

Assuming that the textual representation of senses as in definitions can be useful to align them, we define a few features which use the lengths of senses along with their textual and semantic similarities. In addition, we incorporate word-level semantic relationships to determine the type of relation that two senses may possibly have. Our features are defined in Table 1.

##### 3.2.1.1 Feature Extraction

In this step, we extract sense instances from the *MWSA* datasets (Ahmadi et al. 2020), as  $t = (p, s_i, s_j, r_{i_j})$ . This instance is interpreted as sense  $s_i$  has relation  $r_{i_j}$  with sense  $s_j$ . Therefore, the order of appearance is important to correctly determine the relationship. It should also be noted that both senses belong to the same lemma with the part-of-speech  $p$ . Table 2 provides the basic statistics of the senses and their semantic relationships in various languages. *#Entries* and *#SKOS* refer to the number of entries and senses with a relationship within SKOS. In addition, the senses within the two



## D2.3 Lexical Resource Linking Service

resources which belong to the same lemma but are not annotated with a SKOS relationship, are included with a 'none' relationship.

#	feature	definition	possible values
1	POS_tag	part of speech of the headword	a one-hot vector of {N, V, ADJ, ADV, OTHER}
2	s_len.no_func.l/2	number of space-separated tokens in $s_1$ and $s_2$	N
3	s_len.l/2	number of space-separated tokens in $s_1$ and $s_2$ without function words	N
4	hypernymy	hypernymy score between tokens	sum of weights in CONCEPTNET
5	hyponymy	hyponymy score between tokens	sum of weights in CONCEPTNET
6	relatedness	relatedness score between tokens	sum of weights in CONCEPTNET
7	synonymy	synonymy score between tokens	sum of weights in CONCEPTNET
8	antonymy	antonymy score between tokens	sum of weights in CONCEPTNET
9	meronymy	meronymy score between tokens	sum of weights in CONCEPTNET
10	similarity	similarity score between tokens	sum of weights in CONCEPTNET
11	sem.sim	semantic similarity score between senses using word embeddings	averaging word vectors and cosine similarity [0-1]
12	sem.sim.no.func	semantic similarity score between senses without function words	averaging word vectors and cosine similarity excluding function words [0-1]
13	sem_bin_rel	target class	1 for alignable, otherwise 0
14	sem_rel.with.none	target class	{exact, narrower, broader, related, none}
15	sem_rel	target class	{exact, narrower, broader, related}

Figure 2: Manually extracted features for semantic classification of sense relationships

Given the class imbalance where senses with a 'none' relationship are more frequent than the others, we carry out a data augmentation technique based on the symmetric property of the semantic relationships. By changing the order of the senses, also known as relation direction, in each data instance, a new instance can be created by semantically reversing the relationship. In other words, for each  $t = (p, s_i, s_j, r_{ij})$  there is a  $t' = (p, s_j, s_i, r'_{ij})$  where  $r'_{ij}$  is the inverse of  $r_{ij}$ . Thus, exact and related as symmetric properties remain the same, however, the asymmetric property of the broader and narrower relationships yields narrower and broader, respectively.

---

### D2.3 Lexical Resource Linking Service

Once the senses are extracted, we create data instances using the features in Table 1. Features 2 and 3 concern the length of senses and how they are different. Intuitively speaking, this regards the wordings used to describe two concepts and their semantic relationship. In features 4 to 11, we calculate this with and without function words, words with little lexical meaning. One additional step is to query ConceptNet<sup>5</sup> to retrieve semantic relations between the content words in each sense pair. For instance, the two words “gelded” and “castrated” which appear in two different senses are synonyms and therefore, the whole senses can be possibly synonyms. In order to measure the reliability of the relationships, we sum up the weights, also known as *assertions*, of each relationship according to ConceptNet. Finally, features 12 and 13 provide the semantic similarity of each sense pair using word embeddings. The data instances are all standardized by scaling each feature to the range of [0-1].

#### 3.2.1.2 Feature learning and classification

Restricted Boltzmann machine (RBM) is a generative model representing a probability distribution given a set of observations (Fischer and Igel 2012). An RBM is composed of two layers, a visible one where the data instances according to the manually-created features are provided, and a latent one where a distribution is created by the model by retrieving dependencies within variables. In other words, the relation of the features in how the target classes are predicted is learned in the training phase. We follow the description of (Hinton 2012) in implementing and using an RBM for learning further features from our data instances. Regarding the classification problem, instead of training our models using the data instances described in the previous section, we train the models using the latent features of an RBM model. These new features have binary values and can be configured and tuned depending on the performance of the models.

For this supervised classification problem, we use support vector machines (SVMs) using various hyper-parameters, as implemented in Scikit<sup>6</sup>. After a preprocessing step, where the datasets are shuffled, normalized and scaled, we split them into train, test and validation sets with 80%, 10% and 10% proportions, respectively.

---

<sup>5</sup> <https://conceptnet.io/>

<sup>6</sup> <https://scikit-learn.org>



## 4 Experiments

Table 2 provides the evaluation results of our classification approach for MWSA. Despite the high accuracy of the baseline systems for most languages, they do not perform equally efficiently for all languages in terms of precision and recall. Although our classifiers outperform the baselines for all the relation prediction tasks and perform competitively when trained for the binary classification and also given all data instances, there is a significant low performance when it comes to the classification of SKOS relationships. This can be explained by the lower number of instances available for these relations. Moreover, distinguishing certain types of relationships, such as related versus exact, is a challenging task even for an expert annotator. Regarding the performance of RBM, we do not observe a similar improvement in the results of all classifiers.

One major limitation of the current approach is the usage of crafted features. We believe that as a future work further techniques can be used, particularly thanks to the current advances in word representations and neural networks. In addition, incorporating knowledge bases and external language resources such as corpora can be beneficial in improving to address sense ambiguity for polysemous entries.



## D2.3 Lexical Resource Linking Service

Language	# Entries	# SKOS	# SKOS+none	# All	Metric	Baseline	Binary	All	SKOS	RBM-Binary	RBM-all	RBM-SKOS
Basque	256	813	3661	4382	Accuracy	78.90	78.79	58.47	49.77	70.57	54.17	28.85
					Precision	21.10	71.40	59.21	43.65	62.14	59.08	20.73
					Recall	5.00	72.78	58.45	46.01	74.93	52.55	50.87
					F-measure	8.10	<b>72.08</b>	<b>58.83</b>	<b>44.80</b>	67.94	55.62	29.46
Bulgarian	1000	1976	3708	5656	Accuracy	72.80	70.60	65.91	34.05	73.51	63.38	36.47
					Precision	25.00	68.75	64.79	31.75	77.46	34.46	36.85
					Recall	1.10	69.32	65.44	31.83	72.91	49.87	24.86
					F-measure	2.00	69.03	<b>65.11</b>	<b>31.79</b>	<b>75.11</b>	40.76	29.69
Danish	587	1644	16520	18164	Accuracy	81.70	66.47	34.82	27.87	73.85	50.08	29.67
					Precision	3.00	74.54	23.70	36.49	60.59	60.96	30.47
					Recall	2.30	75.51	62.90	22.87	55.66	66.92	73.04
					F-measure	4.30	<b>75.02</b>	34.43	28.12	<b>58.02</b>	63.80	<b>43.00</b>
Dutch	161	622	20144	20766	Accuracy	93.60	82.55	59.99	24.75	83.90	51.47	36.34
					Precision	0.00	86.97	78.59	31.38	59.78	77.82	30.66
					Recall	0.00	88.24	79.22	33.10	67.33	39.65	66.03
					F-measure	0.00	<b>87.60</b>	<b>78.90</b>	32.22	63.33	52.54	41.88
English	684	1682	9269	10951	Accuracy	75.20	89.00	81.00	49.00	80.16	65.03	48.57
					Precision	0.00	82.35	73.03	39.31	64.36	63.67	55.53
					Recall	0.00	82.87	76.41	46.63	82.13	79.35	34.31
					F-measure	0.00	<b>82.61</b>	<b>74.68</b>	<b>42.66</b>	72.17	70.65	42.57
Estonian	684	1142	2316	3426	Accuracy	48.20	78.98	58.92	46.11	75.96	62.75	47.82
					Precision	54.50	76.06	68.83	40.81	63.53	60.67	36.63
					Recall	9.30	20.76	57.82	44.02	28.18	49.35	22.44
					F-measure	15.90	32.62	<b>62.85</b>	<b>42.35</b>	<b>39.05</b>	54.43	27.83
German	537	1211	4975	6185	Accuracy	77.77	73.14	61.99	49.58	77.97	43.23	44.21
					Precision	0.00	77.72	64.74	41.89	80.44	66.34	40.99
					Recall	0.00	54.41	59.95	43.73	22.88	27.92	48.99
					F-measure	0.00	<b>64.01</b>	<b>62.25</b>	42.79	35.63	39.30	<b>44.63</b>
Hungarian	143	949	15774	16716	Accuracy	94.00	79.65	58.40	22.95	81.46	36.27	15.20
					Precision	5.30	49.96	30.14	23.41	68.50	59.80	26.58
					Recall	1.20	54.47	37.95	68.08	56.72	73.85	29.23
					F-measure	2.00	52.12	33.60	<b>34.85</b>	<b>62.05</b>	<b>66.09</b>	27.84
Irish	680	975	2816	3763	Accuracy	58.30	75.00	55.75	26.27	79.61	60.84	24.75
					Precision	68.00	84.42	46.58	31.84	79.03	42.52	30.25
					Recall	18.50	84.46	39.85	46.15	52.47	54.65	25.40
					F-measure	29.10	<b>84.44</b>	42.95	37.68	63.06	<b>47.83</b>	27.61
Italian	207	592	2173	2758	Accuracy	69.30	59.08	55.43	44.48	77.23	46.26	43.01
					Precision	0.00	52.55	42.98	28.80	75.69	46.31	40.56
					Recall	0.00	66.47	52.64	42.16	45.05	68.67	31.27
					F-measure	0.00	<b>58.69</b>	47.32	<b>34.22</b>	56.49	<b>55.32</b>	35.32
Serbian	301	736	5808	6542	Accuracy	59.90	80.05	32.53	27.55	82.35	41.43	32.96
					Precision	19.00	76.78	48.57	43.06	73.51	37.70	21.49
					Recall	46.40	65.73	69.40	27.10	77.46	48.45	55.53
					F-measure	26.90	70.83	<b>57.15</b>	<b>33.26</b>	<b>75.43</b>	42.40	30.99
Slovenian	152	244	1100	1343	Accuracy	44.20	84.29	36.13	26.13	78.93	39.57	31.63
					Precision	17.30	73.08	23.19	46.98	78.62	38.59	20.97
					Recall	58.70	83.22	45.07	28.61	41.64	28.09	33.02
					F-measure	26.80	<b>77.82</b>	30.62	<b>35.56</b>	54.45	<b>32.51</b>	25.65
Spanish	351	1071	4898	5919	Accuracy	-	73.79	54.67	30.28	80.71	54.38	58.48
					Precision	-	79.78	55.07	33.21	79.40	42.54	39.57
					Recall	-	80.37	53.15	40.04	60.18	20.68	38.59
					F-measure	-	<b>80.07</b>	<b>54.10</b>	36.31	68.47	27.83	<b>39.07</b>
Portuguese	147	275	2062	2337	Accuracy	92.10	71.31	66.62	51.71	73.14	55.69	42.87
					Precision	8.30	49.29	58.23	53.52	77.72	69.41	40.45
					Recall	2.40	37.47	70.41	53.47	54.41	22.32	38.15
					F-measure	3.70	42.57	<b>63.74</b>	<b>53.49</b>	<b>64.01</b>	33.78	39.26
Russian	213	483	3376	3845	Accuracy	75.40	60.88	58.90	37.75	75.80	59.76	33.10
					Precision	43.80	72.92	63.83	27.28	73.38	73.77	32.71
					Recall	17.90	82.21	44.43	36.74	68.23	70.39	47.75
					F-measure	25.50	<b>77.29</b>	52.39	31.31	70.71	<b>72.04</b>	<b>38.82</b>

Figure 3: Basic statistics of the datasets and the best classification results with and without an RBM. #refers to the number



## 5 Deep learning methods

Besides employing feature-based approaches, we additionally utilize fine-tuned pre-trained neural network language models (NNLM), Bidirectional Encoder Representations from Transformers (Devlin et al. 2018) and Robustly optimized BERT pretraining approach (RoBERTa) (Liu et al. 2019). This is done by using the huggingface transformers<sup>7</sup> library, which provides the API for finetuning of transformer models.

Recently, transformers architecture based approaches have been proven to be beneficial for improving different downstream NLP tasks. For this reason we have decided to explore how well those models are suited for the MWSA task.

BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers and is trained on masked word prediction and next sentence prediction tasks. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks (Devlin et al. 2018).

MWSA task can be ultimately regarded as sentence pair classification task and BERT can be easily fine-tuned for it, since its use of self-attention mechanism (Vaswani et al. 2017) to encode concatenated text pairs effectively includes bidirectional cross attention between two sentences. We have followed the fine-tuning approach presented in the original paper (Devlin et al. 2018).

In order to get the best results, we have experimented with different pre-trained models, such as BERT Base, BERT Large and RoBERTa for English. RoBERTa is a variation of BERT created by tweaking different aspects of pre-training, such as bigger data and batches, omitting next sentence prediction, training on longer sequences and changing the masking pattern (Liu et al. 2019).

### 5.1.1 Fine-tuning Transformer models

Transformer based approach was conducted for English and German. Different parameter settings have been tried out to find the best performing model for both languages. Due to the size of the pre-trained language models and limitations in computation powers, we were only able to explore hyper-parameter combinations selectively. Different pre-trained language models were used and were evaluated in the early phase of the experiments, to limit the parameter exploration space.

---

<sup>7</sup> <https://huggingface.co/transformers/index.html>



## D2.3 Lexical Resource Linking Service

### 5.1.1.1 Preprocessing

- **Representation of word senses:** Transformers architecture requires input to be in certain structures depending on the pretrained models used. For our MWSA task, which we basically regard as sentence pair classification, transformer models require two sentences concatenated by separation token, and a preceding classification token. Transformers library by Huggingface provides Tokenizers for different pre-trained models.
- **Labels and class weight:** Labels are one hot encoded and class weights are calculated to mitigate the class imbalance problem.

### 5.1.1.2 Model training

#### 5.1.1.2.1 Training Environment

The training was done on NVIDIA Tesla P100 GPU hosted on Google Cloud Platform.

#### 5.1.1.2.2 Hyperparameters

Our early explorations with the pretrained models quickly showed that bigger models deliver better results. The tendency that bigger pre-trained models perform better on MWSA is in line with observations made by the original BERT paper authors by comparing BERT Base and Large for different downstream tasks (Devlin et al. 2018), or RoBERTa performing better than original BERT on selected downstream tasks (Liu et al. 2019). For this reason, we have conducted more hyperparameter test combinations for those models (RoBERTa Large for English, and DBMDZ for German). When using bigger models, such as RoBERTa or BERT Large, smaller train-batch-size was selected due to resource limitation. The original BERT models were trained with 512 sequence length, but since the MWSA datasets mostly have short sentence pairs, we experimented with shorter sequence length of 128 and 256 to save memory usage and be more flexible with respect to batch size.

The hyperparameters and their values explored are shown in the table below.

Parameter	value set	English	German
<i>used model</i>	BERT English(Large) German BERT(deepset.ai, DBMDZ cased)	RoBERTa(Large)	DBMDZ German BERT
<i>label weights</i>		NONE: 0.23 EXACT: 2.08 BROADER: 42.05 NARROWER:5.37 RELATED:32.69	NONE: 0.27 EXACT: 2.74 BROADER: 2.31 NARROWER:3.13 RELATED:8.32
<i>max-seq-length</i>	64, 128, 256, 512	256	256
<i>train-batch-size</i>	8, 16, 32	16	32
<i>num-train-epochs</i>	2,3,5,7,10,15	2	7
<i>weight-decay</i>	0.3, 0.5	0.3	0.3
<i>learning-rate</i>	1e-6, 8e-6, 9e-6, 1e-5, 3e-5, 4e-5,5e-5	9e-6	3e-5

Figure 4: Set of hyperparameters explored and selected

## D2.3 Lexical Resource Linking Service

### 5.1.1.2.3 Loss Function

As the MWSA task is a multi-class classification task, we use categorical cross entropy as our loss function for fine-tuning the models.

### 5.1.1.3 Model Evaluation

For evaluation of the trained models, we use weighted Matthews correlation coefficient, F1-measure and balanced accuracy, to take data imbalance into account. We monitored the three metrics also during training to determine when the model starts to overfit and adjusted hyperparameters for further tuning.

Comparison of the fine-tuned models were not only done in regards to different hyperparameter settings, but also with respect to feature-based classification models, which we took as the baseline models.

With appropriate hyperparameters, English and German classifiers based on BERT (German) and RoBERTa (English) showed convergence with respect to the categorical cross-entropy loss function. Classes were weighted according to the distribution for loss calculation. Both models selected deliver better results than feature based models. Noteworthy is that transformers based models were able to classify some of the “narrower” relations correctly, where feature based models failed. The general performance of the models leave room for improvements, data imbalance probably plays a significant role in improving the models.

Language	Model	5-class accuracy	2-class precision	2-class recall	2-class F-measure
English	Baseline	0.752	0.000	0.000	0.000
	Feature-based	<b>0.766</b>	0.612	0.533	0.570
	BERT Large	0.654	0.467	<b>0.850</b>	0.602
	RoBERTa	0.763	<b>0.619</b>	0.782	<b>0.691</b>



---

 D2.3 Lexical Resource Linking Service

German	Baseline	0.777	0.000	0.000	0.000
	Feature-based	0.777	0.709	0.448	0.549
	BERT	<b>0.798</b>	<b>0.738</b>	<b>0.608</b>	<b>0.667</b>



## 6 Non-textual Linking Methods

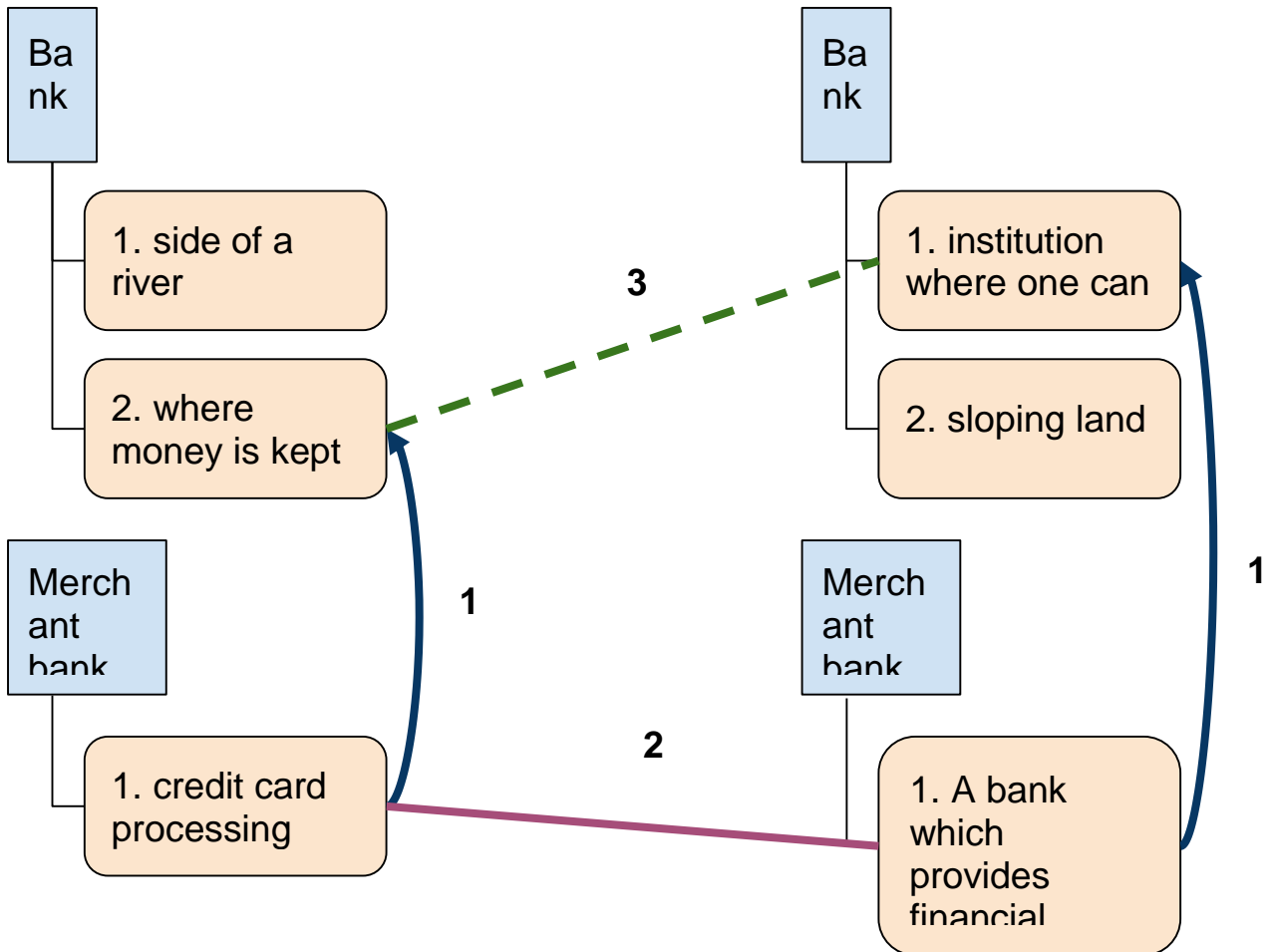


Figure 5: An example of the use of non-textual features for linking. Here the two senses of bank are distinguished by the hypernym links (1) and an inferred hapax legomenon link (2), so that the correct sense (3) can be selected.

In addition, to using textual similarity methods, a number of non-textual methods can be used that are useful for linking dictionaries. There are two principal methods that can be used here: firstly, Naisc supports linking by means of *property overlap*, which creates a feature if two properties of a lexical entry are the same. These properties might be part-of-speech values or may be something more sophisticated such as register or other usage values. The second main method is graph-based similarity, which relies on there being a graph relating the senses of an entry and so is primarily used in the case of WordNet linking. Naisc implements the FastPPR method (Lofgren et al. 2014) to find graph similarity. In the case of wordnet linking, graph similarity cannot be naively applied as there are not generally links between the graphs of the two wordnets, instead we rely on the *hapax legomenon*

---

### D2.3 Lexical Resource Linking Service

links, which are links that are created when there is only one sense for the lemma in both dictionaries. These links allow us to create a graph between the two graphs as shown in Figure 2. We (McCrae and Cillessen 2021) explored this method in the context of linking English WordNet (McCrae et al. 2019) with Wikidata, where we used the Naisc system to find equivalent senses of WordNet synsets and entities in the Wikidata database. In this paper, we found that 67,569 (55.3%) of WordNet's synsets have a matching lemma in Wikidata, of which 16,452 (19.5%) counted as *hapax legomenon* links. We directly evaluated the accuracy of the *hapax legomenon* links and found that accuracy, when applying some simple filters, was 96.1% based on an evaluation of two annotators, who had a Cohen's kappa agreement of 81.4%. We then evaluated using the non-textual methods along with simple textual methods from the previous section and found that there was a 65-66% accuracy of the Naisc system in predicting links between WordNet and Wikidata. Divided by the prediction scores, those links predicted with a confidence of less than 60% by the system were all incorrect (0.0% accuracy), those with a 60-80% accuracy were correct 23/39 times (59.0% accuracy) and those with a greater than 80% confidence were correct 42/49 times (85.7% accuracy), indicating that the system's confidence was a good predictor of the accuracy of links



## 7 Linking Constraints

Linking is a task that cannot only be achieved by looking at pairs of definitions by themselves but instead a **holistic** approach looks at all the links being generated and considers whether this leads to a good overall linking. It is clear that mapping multiple senses to the same senses or generating many more or fewer links than the number of senses is not ideal. In this section, we will look at the methods for solving the problem of sense linking holistically that are implemented in Naisc.

### 7.1 Bijection

The simplest constraint called **bijection** states that the senses for each dictionary entry should be marked as equivalent to at most one sense on the target side and that all senses should be linked for whichever dictionary entry has the fewest entries. This problem is known more generally as the **assignment problem** and can be formally stated for a set of source senses,  $\{s_1, \dots, s_n\}$  and target senses  $\{t_1, \dots, t_m\}$ , an alignment,  $A = \{a_{ij}\}$  is optimal given a score function,  $s(a_{ij})$ . If the following hold:

$$\forall i \in \{1, \dots, n\} \neg \exists j \in \{1, \dots, m\}, j' \in \{1, \dots, m\}, j \neq j' : a_{ij} \in A \wedge a_{ij'} \in A$$

$$\forall j \in \{1, \dots, m\} \neg \exists i \in \{1, \dots, n\}, i' \in \{1, \dots, n\}, i \neq i' : a_{ij} \in A \wedge a_{i'j} \in A$$

$$\forall i \in \{1, \dots, n\} \exists j \in \{1, \dots, m\} a_{ij} \in A \text{ if } n \leq m$$

$$\forall j \in \{1, \dots, m\} \exists i \in \{1, \dots, n\} a_{ij} \in A \text{ if } m \leq n$$

We can weight this problem by assuming that the score is given by  $\sum_{a_{ij} \in A} s(a_{ij})$  and this problem can be solved in cubic time by the Hungarian algorithm (Kuhn 1955). To apply this we use the output probabilities from the classifiers described in the previous section and then:

$$s(a_{ij}) = \log p(a_{ij})$$

Given the high variance in the classifiers we normally further smooth this value as follows:

$$s(a_{ij}) = \log [p(a_{ij}) + \lambda]$$

Where  $\lambda \approx 0.5$ . This allows the system to choose answers rejected by the classifier without an extreme penalty.



### D2.3 Lexical Resource Linking Service

For the purpose of sense linking, the Hungarian algorithm is efficient as the problem can be divided into linking problems for each of the senses. However, for more complex cases the Hungarian algorithm can be very slow and so we have also investigated the use of approximate solvers, such as a simple greedy solver, a beam-search-based solver and the Monte-Carlo tree search algorithm (Chaslot et al. 2008).

## 7.2 b-Matching

WBbM, or shortly b-matching, is one of the widely studied classical problems in combinatorial optimization for modeling data management applications, e-commerce and resource allocation systems (Ahmed, Dickerson, and Fuge 2017). WBbM is a variation of the weighted bipartite matching, also known as assignment problem. In the assignment problem, the optimal matching only contains one-to-one matching with the highest weight sum. This bijective mapping restriction is not realistic in the case of lexical resources where an entry may be linked to more than one entry. Therefore, WBbM aims at providing a more diversified matching where a node may be connected to a certain number of nodes.

---

#### Algorithm 1: Greedy WBbM

---

**Input:**  $G = ((U, V), E, W)$ , bounds  $L$  and  $B$

**Output:**  $H = ((U, V), E', W)$  satisfying bound constraints with a greedily-maximized score  $\sum_{e \in E'} W(e)$

```

1  $E' = \emptyset$ 
2 Sort  $E$  by descending  $W(e)$ 
3 for  $e$  to  $E$  do
4   if  $H = ((U, V), E' \cup \{e\}, W)$  does not violate  $L$  and  $B$  then
5      $E' = E' \cup \{e\}$ 
6 return  $H = ((U, V), E', W)$ 

```

---

Algorithm 1 presents the WBbM algorithm with a greedy approach where an edge is selected under the condition that adding such an edge does not violate the lower and the upper bounds, i.e.  $L$  and  $B$ .



### 7.3 Taxonomic

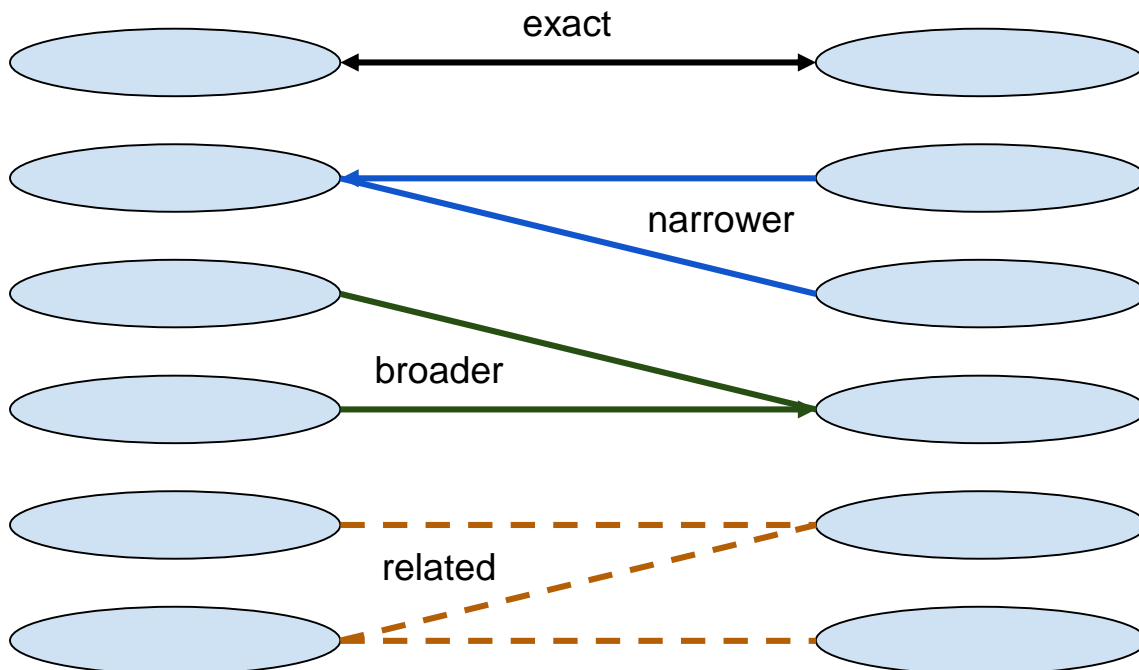


Figure 6: An example of a valid taxonomic linking according to the constraints. No further links could be added between any of the elements.

The most typical case of sense linking consists of not only exact matches as considered in the bijective and b-matching case, but also broader, narrower and related links. As such we have investigated the use of a ‘taxonomic’ constraint that can be stated as follows:

- **Exact** links should be bijective (as defined above). Any sense that is the source or target of an exact link should not be the source or target of any other link.
- **Broader/narrower** links should be *surjective/injective*. This means that if a source sense is part of a broader link it may be part of other broader links, but the target sense cannot be the target of another broader link. Similarly the converse holds for narrower links
- All link types are **exclusive**, that is if the source or target sense of any element is linked by one of the four relation types (exact, broader, narrower, related), then neither the source or target can be involved in a link of any other type.
- A **threshold** can be applied to ensure that only links of a certain quality are generated by the system.

An example of the links that are valid for these constraints is shown in Figure 3. With this more complex constraint, it is not clear whether there exists a polynomial-time algorithm to solve these

### D2.3 Lexical Resource Linking Service

constraints, and while, even for the small size of problems that are seen in sense linking, validating an optimal solution is not feasible, we have also observed that the greedy solver mostly returns the optimal or a near-optimal solution. As such, we simply rely on the approximate methods of linking, including the greedy solver, for this task.



## 8 Benchmarks and Shared Task

One major limitation regarding previous work was with respect to the nature of the data used for the WSA task. Expert-made resources, such as the Oxford English Dictionary, require much effort to create and therefore, are not as widely available as collaboratively-curated ones like Wiktionary<sup>8</sup> due to copyright restrictions. On the other hand, the latter resources lack domain coverage and descriptive senses. To address this, we present a set of 17 datasets containing monolingual dictionaries in 15 languages, annotated by language experts within the ELEXIS volunteers and partners with five semantic relationships according to the simple knowledge organization system reference (SKOS) (Miles and Bechhofer 2009), namely, broader, narrower, related, exact and none.

```

{
  "lemma": "splenetic",
  "POS_tag": "adjective",
  "gender": "",
  "meta_ID": "",
  "resource_1_senses": [
    {
      "#text": "of or relating to the spleen",
      "external_ID": "splenic.a.01"},
    {
      "#text": "very irritable",
      "external_ID": "bristly.s.01*"}
  ],
  "resource_2_senses": [
    {
      "#text": "affected with spleen; malicious;
      ↳ spiteful; peevish; fretful.",
      "external_ID": ""}
  ],
  "alignment": [
    {
      "sense_source": "very irritable",
      "sense_target": "affected with spleen;
      ↳ malicious; spiteful; peevish;
      ↳ fretful.",
      "semantic_relationship": "exact"}
  ]
}

```

Figure 7: An example of the structure of senses and their alignments in the datasets

The main goal of creating datasets for MWSA is to provide semantic relationships between two sets of senses for the same lemmas in two monolingual dictionaries. The actual annotation was implemented by means of dynamic spreadsheets that provide a simple but effective manner to complete the annotation. This also had the added advantage that the annotation task could be easily completed from any device. In order to collect the data that was required for the annotation, each of the participating institutes provided their data in some form providing the following data:

- An entry identifier, that locates the entry in the resource

<sup>8</sup> [www.wiktionary.org](http://www.wiktionary.org)



---

### D2.3 Lexical Resource Linking Service

- A sense identifier marking the sense in the resource, for example the sense number
- The lemma of the entry
- The part-of-speech of the entry
- The sense text, including the definition

One of the challenges is that sense granularity between two dictionaries is rarely such that we would expect one-to-one mapping between the senses of an entry. In this respect, we followed a simple approach such as that in SKOS providing different kinds of linking predicates which is described as follows:

- exact: The sense are the same, for example the definitions are simply paraphrases
- broader: The sense in the first dictionary completely covers the meaning of the sense in the second dictionary and is applicable to further meanings
- narrower: The sense in the first dictionary is entirely covered by the sense of the second dictionary, which is applicable to further meanings
- related: There are cases when the senses may be equal but the definitions in both dictionaries differ in key aspects
- none: There is no match for this sense

While it is certainly not easy to decide which relationship is to be used, we found that this methodology was broadly effective and we believe will simplify the development of machine-learning-based classifiers for sense alignment prediction.

Figure 4 presents the structure of the datasets in JSON format. External keys such as *meta\_ID* and *external\_ID* will enable future lexicographers to integrate the annotations in external resources. Given that some of the semantic relationships, such as narrower and broader, are not symmetric, *sense\_source* and *sense\_target* are important classes in determining the semantic relationship correctly. Table 2 also provides basic statistics of the datasets such as number of entries and sense alignments.

Given that the datasets are publicly available<sup>9</sup>, we carried out a shared task<sup>10</sup> on the task of monolingual word sense alignment across dictionaries as part of the GLOBALEX 2020 – Linked

---

<sup>9</sup> <https://github.com/elexis-eu/MWSA>

<sup>10</sup> <https://competitions.codalab.org/competitions/22163>



### D2.3 Lexical Resource Linking Service

Lexicography workshop at the 12th Language Resources and Evaluation Conference (LREC 2020) which took place on Tuesday, May 12 2020 in Marseille (France).



## 9 Summary

The MWSA task that we have defined here is interesting and there is still much to be done in terms of solving the task effectively. In order to solve this issue, we have introduced the first benchmark for this task, which was supported by a shared task. We have investigated the use of a number of methods for solving this task and we found that we have achieved strong performance with state-of-the-art methods such as BERT and RBMs. We have also introduced a number of features including graph-based methods that can be very useful for linking certain datasets including wordnets. Finally, we looked at making a holistic linking using constraint-based solvers, which ensure that the output makes sense to the lexicographers who will use the downstream results.



## 10 Bibliography

- Agirre, E., C. Banea, D. M. Cer, and M. T. Diab. 2016. "SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation." *SemEval@ NAACL*, 497–511.
- Ahmadi, Sina, John P. McCrae, Sanni Nimb, Fahad Khan, Monica Monachini, Bolette S. Pedersen, Thierry Declerck, et al. 2020. "A Multilingual Evaluation Dataset for Monolingual Word Sense Alignment." In *Proceedings of the 12th Language Resource and Evaluation Conference (LREC 2020)*. National University of Ireland Galway. <https://www.library.ucg.ie/handle/10379/15920>.
- Ahmed, Faez, John P. Dickerson, and Mark Fuge. 2017. "Diverse Weighted Bipartite B-Matching." *arXiv [cs.DS]*. arXiv. <http://arxiv.org/abs/1702.07134>.
- Chang, Chih-Chung, and Chih-Jen Lin. 2011. "LIBSVM: A Library for Support Vector Machines." *ACM Trans. Intell. Syst. Technol.* 2 (3): 27:1–27:27.
- Chaslot, Guillaume, Sander Bakkes, Istvan Szita, and Pieter Spronck. 2008. "Monte-Carlo Tree Search: A New Framework for Game AI." In *AIIDE*. <https://www.aaai.org/Papers/AIIDE/2008/AIIDE08-036.pdf>.
- Cimiano, Philipp, John P. McCrae, and Paul Buitelaar. 2016. "Lexicon Model for Ontologies: Community Report." W3C. <https://www.w3.org/2016/05/ontolex/>.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." *arXiv [cs.CL]*. arXiv. <http://arxiv.org/abs/1810.04805>.
- Fischer, Asja, and Christian Igel. 2012. "An Introduction to Restricted Boltzmann Machines." In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, 14–36. Springer Berlin Heidelberg.
- Hinton, Geoffrey E. 2012. "A Practical Guide to Training Restricted Boltzmann Machines." In *Neural Networks: Tricks of the Trade: Second Edition*, edited by Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller, 599–619. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Kuhn, H. W. 1955. "The Hungarian Method for the Assignment Problem." *Naval Research Logistics Quarterly* 2 (1-2): 83–97.
- Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." *arXiv [cs.CL]*. arXiv. <http://arxiv.org/abs/1907.11692>.
- Lofgren, Peter A., Siddhartha Banerjee, Ashish Goel, and C. Seshadhri. 2014. "FAST-PPR: Scaling Personalized Pagerank Estimation for Large Graphs." In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1436–45. KDD '14. New York, NY, USA: Association for Computing Machinery.
- McCrae, John P., and Paul Buitelaar. 2018. "Linking Datasets Using Semantic Textual Similarity." *Cybernetics and Information Technologies* 18 (1): 109–23.
- McCrae, John P., Paul Buitelaar, and Philipp Cimiano. 2017. "The OntoLex-Lemon Model: Development and Applications." In *Proceedings of eLex 2017*.



---

### D2.3 Lexical Resource Linking Service

- McCrae, John P., and David Cillessen. 2021. "Towards a Linking between WordNet and Wikidata." In *Proceedings of the 11th Global WordNet Conference*.
- McCrae, John P., Alexandre Rademaker, Francis Bond, Ewa Rudnicka, and Christiane Fellbaum. 2019. "English Wordnet 2019--an Open-Source Wordnet for English." In *Wordnet Conference*, 245.
- Miles, Alistair, and Sean Bechhofer. 2009. "SKOS Simple Knowledge Organization System Reference." *W3C Recommendation*, August. <https://www.escholar.manchester.ac.uk/uk-ac-man-scw:66505>.
- Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. "Glove: Global Vectors for Word Representation." In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–43.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." *arXiv [cs.CL]*. arXiv. <http://arxiv.org/abs/1706.03762>.

