The ELEXIS System for Monolingual Sense Linking in Dictionaries

John P. McCrae[1], Sina Ahmadi[1], Seung-Bin Yim[2], Lenka Bajčetić[2]

[1] Data Science Institute, NUI Galway
[2] Austrian Academy of Sciences
E-mail: john@mccr.ae, sina.ahmadi@insight-centre.org,
seung-bin.yim@oeaw.ac.at, lenka.bajcetic@oeaw.ac.at

Abstract

Sense linking is the task of inferring any potential relationships between senses stored in two dictionaries. This is a challenging task and in this paper we present our system that combines Natural Language Processing (NLP) and non-textual approaches to solve this task. We formalise linking as inferring links between pairs of senses as exact equivalents, partial equivalents (broader/narrower) or a looser relation or no relation between the two senses. This formulates the problem as a five-class classification for each pair of senses between the two dictionary entries. The work is limited to the case where the dictionaries are in the same language and thus we are only matching senses whose headword matches exactly; we call this task Monolingual Word Sense Alignment (MWSA). We have built tools for this task into an existing framework called Naisc and we describe the architecture of this system as part of the ELEXIS infrastructure, which covers all parts of the lexicographic process including dictionary drafting. Next, we 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. We then look at methods that can exploit non-textual information about the senses in a meaningful way. Afterwards, we describe 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 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. The combination of these tools provides a highly flexible implementation that can link senses between a wide variety of input dictionaries and we demonstrate how linking can be done as part of the ELEXIS toolchain.

Keywords: sense linking; lexicography; natural language processing; linked data; tools

1. Introduction

Monolingual word sense alignment is the task of finding the equivalent or related senses among two dictionary entries with the same headword from two different dictionaries. In this paper, we present our framework and tool for creating such a mapping between two dictionaries, called Naisc McCrae & Buitelaar (2018)[1]. This architecture is intended as an experimental framework into which many components can be integrated. In this paper, we give an overview of this system and examples of some of the methods that that can be integrated into this framework. For this work, we focus on only the monolingual word sense alignment task, but many of the techniques discussed here can also be used to create multilingual linking between dictionaries and also linking between other kinds of datasets.

We understand that there are three major aspects to consider when building a linking system in the framework provided by Naisc. Firstly, we have the task of textual similarity, which takes the textual content of each sense, principally the definition and estimates the similarity between them. Secondly, we have non-textual similarity, an iterative process that can be used to link dictionaries that contain links between entries, such as WordNet. These tools become especially useful in the context of linking to external encyclopaedic resources such as Wikipedia or Wikidata. Finally, we look at linking as a holistic step, where we consider the linking task as one of predicting one of four relationships between senses: equivalent, narrower, broader or partially related. This turns the task into a

[1] https://github.com/insight-centre/naisc

five-class classification task (with 'unrelated' as the fifth class), but in addition there are constraints that logically follow, and we formalise this and show how we can generate an optimal overall mapping between senses.

These elements are all being integrated into the framework and we present some preliminary results about the individual component performance as well as insight into the motivations of the architecture and the design of the system. In addition, we also summarise the development of a benchmark dataset for this task (Ahmadi et al., 2020). The rest of this paper is structured as follows. In Section 2 we present the overall architecture of the Naisc system. We then look at textual features in Section 3, non-textual features in Section 4 and constraints for linking in Section 5. Finally, we describe the development of a benchmark dataset in Section 6 and conclude in Section 7.

2. Architecture

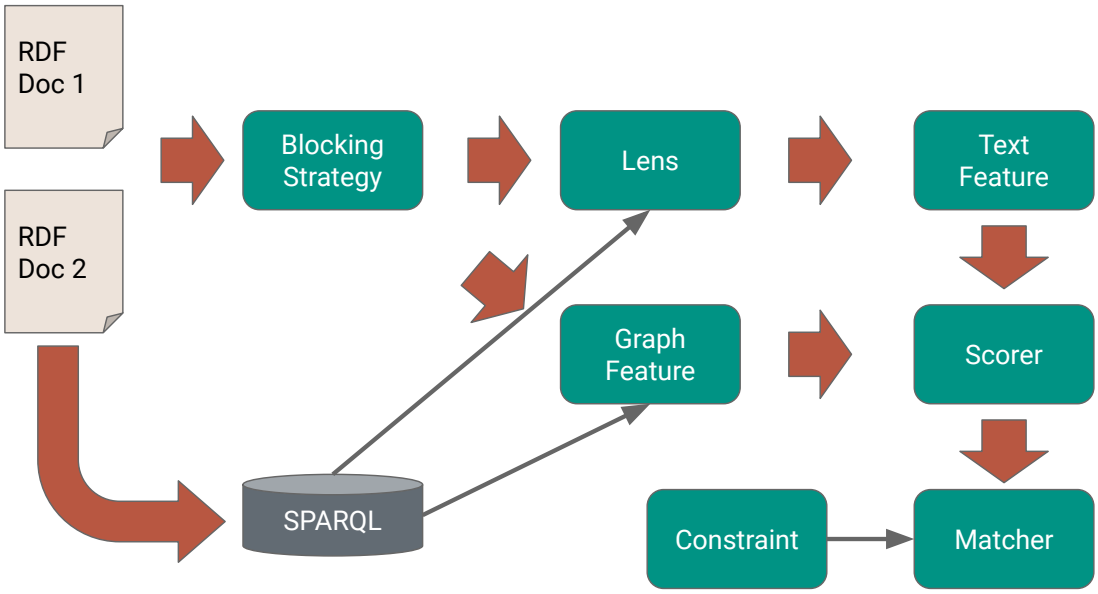

Figure 1: The architecture of the Naisc system for sense linking

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 et al., 2017; Cimiano et al., 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 linked. 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 based 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) ⇒ (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' 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
scorers:
- name: scorer.LibSVM
  modelFile: models/default.libsvm
matcher:
  name: matcher.BeamSearch
  constraint:
```

```
    name: constraint.Taxonomic
description: The default setting for processing two OntoLex 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 et al., 2014a). The scores for each property type are calculated using LibSVM (Chang & Lin, 2011) and finally the overall linking is calculated using the taxonomic constraints, which will be defined later in this document.

3. Text Similarity Methods

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 for 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 $= \frac{2|AB|}{|A|+|B|}$, Containment $= \frac{|A \cap B|}{\min(|A|,|B|)}$

Sentence Length Ratio The ratio of the length of the sentences as $\text{SLR}(x, y) = 1 - \frac{min(|x|,|y|)}{max(|x|,|y|)}$

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 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).

Jaro-Winkler, Levenshtein Standard string similarity functions, we use the Apache Commons Text implementations.

Monge-Elkan This is defined as follows where sim is a word similarity function (we use either Jaro-Winkler of Levenshtein) $\text{ME}(s,t) = \frac{1}{|s|}\sum_{i=1}^{|s|}\max_{j=1,\ldots t} sim(s_i, t_j)$

In addition, we implement the following approach based on using GloVe vectors (Pennington et al., 2014b), 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 & 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, cross-references, synonyms and antonyms, senses are unique in the sense that they are more descriptive but also highly contextualised. 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 the focus of previous research (Ahmadi & McCrae, 2021). However, 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 analysing 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 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 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.

Feature Extraction

In this step, we extract sense instances from the MWSA datasets (Ahmadi et al., 2020), as $t = (p, s_i, s_j, r_{ij})$. This instance is interpreted as sense s_i has relation r_{ij} 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.

#	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_1/2`	number of space-separated tokens in s_1 and s_2	$\mathbb{N}$
3	`s_len_1/2`	number of space-separated tokens in s_1 and s_2 without function words	$\mathbb{N}$
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`}

Table 1: 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.

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 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 possibly be 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 standardised by scaling each feature to the range of [0-1].

Feature learning and classification

A Restricted Boltzmann Machine (RBM) is a generative model representing a probability distribution given a set of observations (Fischer & Igel, 2012). An RBM is composed of two layers: a visible one where the data instances are provided according to the manually created features, 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. 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.

3.3 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 significantly lower 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 the 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. Furthermore, incorporating knowledge bases and external language resources such as corpora can be beneficial in improving the ability of the system to address sense ambiguity for polysemous entries.

Language	Metric	Baseline	Binary	All	SKOS	RBM-Binary	RBM-all	RBM-SKOS
Basque	Accuracy	78.90	78.79	58.47	49.77	70.37	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	**72.08**	**58.83**	**44.80**	67.94	55.62	29.46
Bulgarian	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	**65.11**	**31.79**	**75.11**	40.76	29.69
Danish	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	**75.02**	34.43	28.12	**58.02**	63.80	**43.00**
Dutch	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	**87.60**	**78.90**	32.22	63.33	52.54	41.88
English	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	34.51
	F-measure	0.00	**82.61**	**74.68**	**42.66**	72.17	70.65	42.57
Estonian	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	**62.85**	**42.35**	**39.05**	54.43	27.83
German	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	**64.01**	**62.25**	42.79	35.63	39.30	**44.63**
Hungarian	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	**34.85**	**62.05**	**66.09**	27.84
Irish	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	**84.44**	42.95	37.68	63.06	**47.83**	27.61
Italian	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	**58.69**	47.32	**34.22**	56.49	**55.32**	35.32
Serbian	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	**57.15**	**33.26**	**75.43**	42.40	30.99
Slovenian	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	**77.82**	30.62	**35.56**	54.45	**32.51**	25.65
Spanish	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	-	**80.07**	**54.10**	36.31	68.47	27.83	**39.07**
Portuguese	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	**63.74**	**53.49**	**64.01**	33.78	39.26
Russian	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	**77.29**	52.39	31.31	70.71	**72.04**	**38.82**

Table 2: Results of the classification results with and without an RBM.

3.4 Deep Learning Methods

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

Recently, transformer-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 unlabelled text by jointly conditioning on both the 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., 2019).

The MWSA task can be ultimately regarded as sentence pair classification task and BERT can easily be 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., 2019).

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)

3.4.1 Fine-tuning transformer models

A 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.

Preprocessing

Representation of word senses The 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. The Hugging Face transformers library provides tokenisers 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.

Model training

Training Environment

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

Hyperparameters

Our early explorations with the pre-trained 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., 2019) or RoBERTa performing better than the 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 limitations. The original BERT models were trained with 512 sequence lengths, 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.

Parameter	value set	English	German
used model	BERT English(Large) German BERT(deepset.ai, DBMDZ cased)	RoBERTa(Large)	DBMDZ German BERT
label weights		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
max-seq-length	64, 128, 256, 512	256	256
train-batch-size	8, 16, 32	16	32
num-train-epochs	2,3,5,7,10,15	2	7
weight-decay	0.3, 0.5	0.3	0.3
learning-rate	1e-6, 8e-6, 9e-6, 1e-5, 3e-5, 4e-5,5e-5	9e-6	3e-5

Table 3: Language model and Hyperparameters used for fine-tuning NNLM to MWSA

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.

Model Evaluation

For evaluation of the trained models, we use weighted the Matthews correlation coefficient (Matthews, 1975), F1-measure and balanced accuracy, to take data imbalance into account. We also monitored the three metrics during training to determine when the model starts to overfit and adjusted the hyperparameters for further tuning.

Comparison of the fine-tuned models were not only done in regard 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 transformer-based models were able to classify some of the "narrower"

relations correctly, where feature-based models failed. The general performance of the models leaves room for improvements, and data imbalance probably plays a significant role in improving them.

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	0.766	0.612	0.533	0.570
	BERT Large	0.654	0.467	0.850	0.602
	RoBERTa	0.763	0.619	0.782	0.691
German	Baseline	0.777	0.000	0.000	0.000
	Feature-based	0.777	0.709	0.448	0.549
	BERT	0.798	0.738	0.608	0.667

Table 4: Evaluation of RoBERTa and BERT models on the MWSA benchmark for English and German

4. Non-textual Linking Methods

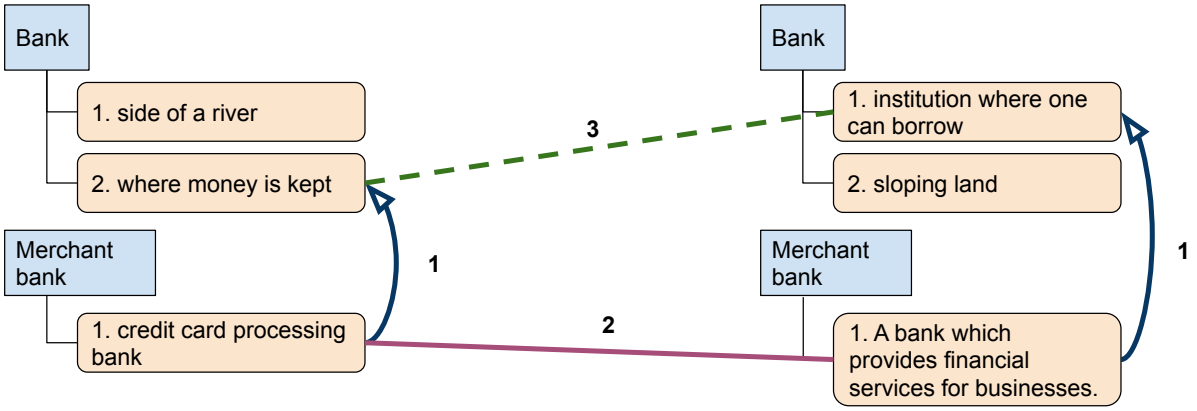

Figure 2: An example of the use of non-textual features for linking. Here the two senses of bank are distinguished by the hypernym links (1) and an inferred hapax legomenon link (2), so that the correct sense (3) can be selected.

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 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. In another work (McCrae & Cillessen, 2021) we 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%) or

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 the 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.

5. 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.

5.1 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, $\{s_1, \ldots, s_n\}$ and target senses $\{t_1, \ldots, t_m\}$, an alignment, $A = \{a_{ij}\}$ is optimal given a score function, $s(a_{ij})$. If the following hold:

$$\forall i \in \{1, \ldots, n\} \nexists j \in \{1, \ldots, m\}, j' \in \{1, \ldots, m\}, j \neq j' : a_{ij} \in A \wedge a_{ij'} \in A$$
$$\forall j \in \{1, \ldots, m\} \nexists i \in \{1, \ldots, n\}, i' \in \{1, \ldots, n\}, i \neq i' : a_{ij} \in A \wedge a_{i'j} \in A$$
$$\forall i \in \{1, \ldots, n\} \exists j \in \{1, \ldots, m\} a_{ij} \in A \text{ if } n \leq m$$
$$\forall j \in \{1, \ldots, m\} \exists i \in \{1, \ldots, n\} a_{ij} \in A \text{ if } m \leq n$$

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

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).

5.2 b-Matching

WBbM, or b-matching, is one of the widely studied classical problems in combinatorial optimisation for modelling data management applications, e-commerce and resource allocation systems (Ahmed et al., 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-maximised score $\sum_{e \in E'} W(e)$

```
E' = ∅
Sort E by descending W(e)
for e to E do
    if H = ((U,V)), E' ∪ {e}, W) does not violate L and B then
        E' = E' ∪ {e}
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.

5.3 Taxonomic

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.

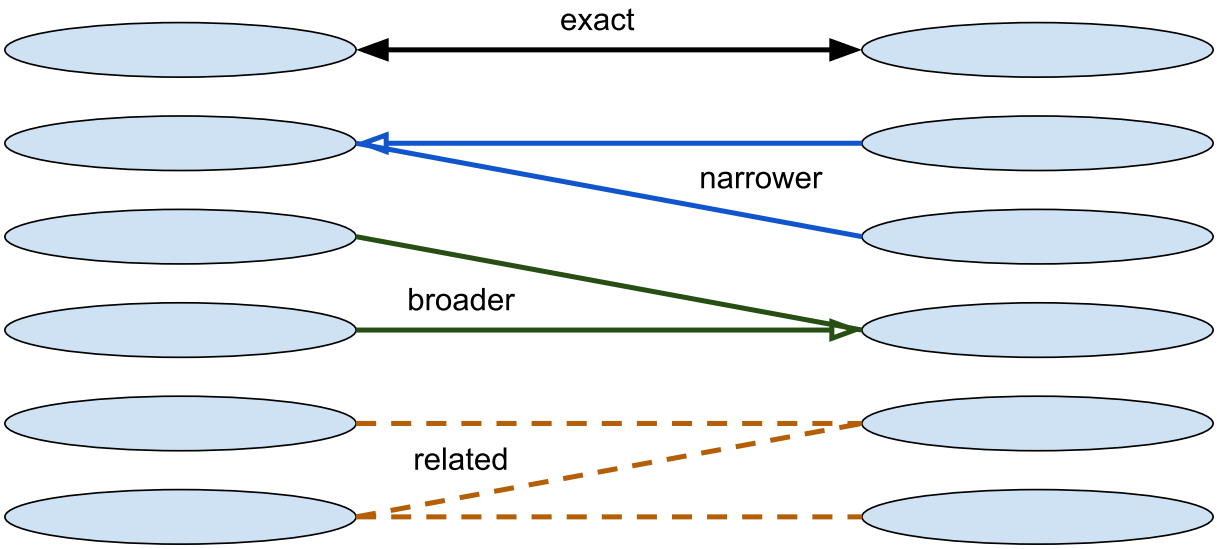

Figure 3: An example of a valid taxonomic linking according to the constraints. No further links could be added between any of the elements

- 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 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.

6. 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 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 organisation system reference (SKOS) (Miles & Bechhofer, 2009), namely, broader, narrower, related, exact and none.

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

Language	# Entries	# SKOS	# SKOS+none	# All
Basque	256	813	3661	4382
Bulgarian	1000	1976	3708	5656
Danish	587	1644	16520	18164
Dutch	161	622	20144	20766
English	684	1682	9269	10951
Estonian	684	1142	2316	3426
German	537	1211	4975	6185
Hungarian	143	949	15774	16716
Irish	680	975	2816	3763
Italian	207	592	2173	2758
Serbian	301	736	5808	6542
Slovenian	152	244	1100	1343
Spanish	351	1071	4898	5919
Portuguese	147	275	2062	2337
Russian	213	483	3376	3845

Table 5: Basic statistics of the datasets. # refers to the number

data that was required for the annotation, each of the participating institutes provided their data in some form providing the following:

- An entry identifier, that locates the entry in the resource
- 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 senses 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 these senses.

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. The datasets are available in JSON format and external keys such as meta_ID and external_ID 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 5 also provides basic statistics of the datasets such as number of entries and sense alignments. #Entries and #SKOS refer to the number of entries and senses with a relationship within SKOS. In addition, the senses within the two resources which belong to the same lemma but are not annotated with a SKOS relationship, are included with a *'none'* relationship.

Given that the datasets are publicly available, we carried out a shared task on the task of monolingual word sense alignment across dictionaries as part of the GLOBALEX 2020 – Linked Lexicography workshop at the 12th Language Resources and Evaluation Conference (LREC 2020) which took place on Tuesday, May 12, 2020 in Marseille (France).

7. Conclusions

In this paper, we have defined the monolingual word sense alignment task and a framework for solving this called Naisc. We looked at textual similarity and there are a large number of methods that are effective for estimating similarity, however the task of distinguishing between exactly equivalent senses and broader/narrower senses is still a challenging one. We then looked at non-textual linking methods that are effective for a few kinds of dictionary linking tasks, especially with large-scale knowledge graphs such as Wikidata. Finally, we examined the constraints that can be used to find the best overall linking between senses and showed how these can be solved. Further, we showed the development of a new benchmark and are working on the integration of all these tools into a single workflow that will form part of the ELEXIS dictionary infrastructure.

Acknowledgements

This work has received funding from the EU's Horizon 2020 Research and Innovation programme through the ELEXIS project under grant agreement No. 731015.

8. References

Agirre, E., Banea, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G. & Wiebe, J. (2016). SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 497–511. URL https://www.aclweb.org/anthology/S16-1081.

Ahmadi, S. & McCrae, J.P. (2021). Monolingual Word Sense Alignment as a Classification Problem. In *Proceedings of the 11th Global Wordnet Conference*. pp. 73–80.

Ahmadi, S., McCrae, J.P., Nimb, S., Khan, F., Monachini, M., Pedersen, B.S., Declerck, T., Wissik, T., Bellandi, A., Pisani, I., Troelsgård, T., Olsen, S., Krek, S., Lipp, V., Váradi, T., Simon, L., Gyorffy, A., Tiberius, C., Schoonheim, T., Moshe, Y.B., Rudich, M., Ahmad, R.A., Lonke, D., Kovalenko, K., Langemets, M., Kallas, J., Dereza, O., Fransen, T., Cillessen, D., Lindemann, D., Alonso, M., Salgado, A., Sancho, J., Ureña-Ruiz, R., Zamorano, J.P., Simov, K., Osenova, P., Kancheva, Z., Radev, I., Stankovic, R., Perdih, A. & Gabrovsek, D. (2020). A Multilingual Evaluation Dataset for Monolingual Word Sense Alignment. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani,

H. Mazo, A. Moreno, J. Odijk & S. Piperidis (eds.) *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*. European Language Resources Association, pp. 3232–3242. URL https://www.aclweb.org/anthology/2020.lrec-1.395/.

Ahmed, F., Dickerson, J.P. & Fuge, M. (2017). Diverse Weighted Bipartite b-Matching. In C. Sierra (ed.) *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. ijcai.org, pp. 35–41. URL https://doi.org/10.24963/ijcai.2017/6.

Chang, C. & Lin, C. (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3), pp. 27:1–27:27. URL https://doi.org/10.1145/1961189.1961199.

Chaslot, G., Bakkes, S., Szita, I. & Spronck, P. (2008). Monte-Carlo Tree Search: A New Framework for Game AI. In C. Darken & M. Mateas (eds.) *Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference, October 22-24, 2008, Stanford, California, USA*. The AAAI Press. URL http://www.aaai.org/Library/AIIDE/2008/aiide08-036.php.

Cimiano, P., McCrae, J.P. & Buitelaar, P. (2016). Lexicon Model for Ontologies: Community Report. URL https://www.w3.org/2016/05/ontolex/.

Devlin, J., Chang, M.W., Lee, K. & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. URL https://www.aclweb.org/anthology/N19-1423.

Fischer, A. & Igel, C. (2012). An Introduction to Restricted Boltzmann Machines. In L. Álvarez, M. Mejail, L.G. Déniz & J.C. Jacobo (eds.) *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - 17th Iberoamerican Congress, CIARP 2012, Buenos Aires, Argentina, September 3-6, 2012. Proceedings*, volume 7441 of *Lecture Notes in Computer Science*. Springer, pp. 14–36. URL https://doi.org/10.1007/978-3-642-33275-3_2.

Hinton, G.E. (2012). A Practical Guide to Training Restricted Boltzmann Machines. In G. Montavon, G.B. Orr & K. Müller (eds.) *Neural Networks: Tricks of the Trade - Second Edition*, volume 7700 of *Lecture Notes in Computer Science*. Springer, pp. 599–619. URL https://doi.org/10.1007/978-3-642-35289-8_32.

Kuhn, H.W. (1955). The Hungarian Method for the Assignment Problem. In M. Jünger, T.M. Liebling, D. Naddef, G.L. Nemhauser, W.R. Pulleyblank, G. Reinelt, G. Rinaldi & L.A. Wolsey (eds.) *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*. Springer, pp. 29–47. URL https://doi.org/10.1007/978-3-540-68279-0_2.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692. URL http://arxiv.org/abs/1907.11692. 1907.11692.

Lofgren, P., Banerjee, S., Goel, A. & Comandur, S. (2014). FAST-PPR: scaling personalized pagerank estimation for large graphs. In S.A. Macskassy, C. Perlich, J. Leskovec, W. Wang & R. Ghani (eds.) *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*. ACM, pp. 1436–1445. URL https://doi.org/10.1145/2623330.2623745.

Matthews, B. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2), pp. 442–451. URL https://www.sciencedirect.com/science/article/pii/0005279575901099.

McCrae, J.P., Bosque-Gil, J., Gracia, J., Buitelaar, P. & Cimiano, P. (2017). The OntoLex-Lemon Model: development and applications. In *Proceedings of eLex 2017*. pp. 587–597. URL https://elex.link/elex2017/wp-content/uploads/2017/09/paper36.pdf.

McCrae, J.P. & Buitelaar, P. (2018). Linking Datasets Using Semantic Textual Similarity. *Cybernetics and Information Technologies*, 18(1), pp. 109–123. URL http://www.cit.iit.bas.bg/CIT_2018/v-18-1/10_paper.pdf.

McCrae, J.P. & Cillessen, D. (2021). Towards a Linking between WordNet and Wikidata. In *Proceedings of the 11th Global Wordnet Conference*. University of South Africa (UNISA): Global Wordnet Association, pp. 252–257. URL https://www.aclweb.org/anthology/2021.gwc-1.29.

McCrae, J.P., Rademaker, A., Bond, F., Rudnicka, E. & Fellbaum, C. (2019). English WordNet 2019 – An Open-Source WordNet for English. In *Proceedings of the 10th Global Wordnet Conference*. Wroclaw, Poland: Global Wordnet Association, pp. 245–252. URL https://www.aclweb.org/anthology/2019.gwc-1.31.

Miles, A. & Bechhofer, S. (2009). SKOS Simple Knowledge Organization System Reference. W3C Recommendation, World Wide Web Consortium. URL https://www.w3.org/TR/skos-reference/.

Pennington, J., Socher, R. & Manning, C. (2014a). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. URL https://www.aclweb.org/anthology/D14-1162.

Pennington, J., Socher, R. & Manning, C.D. (2014b). Glove: Global Vectors for Word Representation. In A. Moschitti, B. Pang & W. Daelemans (eds.) *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, pp. 1532–1543. URL https://doi.org/10.3115/v1/d14-1162.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. & Polosukhin, I. (2017). Attention is All you Need. In I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan & R. Garnett (eds.) *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. pp. 5998–6008. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.