Implementation of Supervised Training Approaches for Monolingual Word Sense Alignment: ACDH-CH System Description for the MWSA Shared Task at GlobaLex 2020

Bajčetić Lenka, Yim Seung-Bin
Austrian Centre for Digital Humanities and Cultural Heritage
Vienna
{lenka.bajcetic, seung-bin.yim}@oeaw.ac.at

Abstract

This paper describes our system for monolingual sense alignment across dictionaries. The task of monolingual word sense alignment is presented as a task of predicting the relationship between two senses. We will present two solutions, one based on supervised machine learning, and the other based on pre-trained neural network language model, specifically BERT. Our models perform competitively for binary classification, reporting high scores for almost all languages.

Keywords: Monolingual Word Sense Alignment, Lexicography, BERT, Gloss Similarity

1. Introduction

This paper presents our submission for the shared task on monolingual word sense alignment across dictionaries as part of the GLOBALEX 2020 – Linked Lexicography workshop at the 12th Language Resources and Evaluation Conference (LREC). Monolingual word sense alignment (MWSA) is the task of aligning word senses across resources in the same language.

Lexical-semantic resources (LSR) such as dictionaries form valuable foundation of numerous natural language processing (NLP) tasks. Since they are created manually by experts, dictionaries can be considered among the resources of highest quality and importance. However, the existing LSRs in machine readable form are small in scope or missing altogether. Thus, it would be extremely beneficial if the existing lexical resources could be connected and expanded.

Lexical resources display considerable variation in the number of word senses that lexicographers assign to a given entry in a dictionary. This is because the identification and differentiation of word senses is one of the harder tasks that lexicographers face. Hence, the task of combining dictionaries from different sources is difficult, especially for the case of mapping the senses of entries, which often differ significantly in granularity and coverage. (Ahmadi et al., 2020)

There are three different angles from which the problem of word sense alignment can be addressed: approaches based on the similarity of textual descriptions of word senses, approaches based on structural properties of lexical-semantic resources, and a combination of both. (Matuschek, 2014)

In this paper we focus on the similarity of textual descriptions. This is a common approach as the majority of previous work used some notion of similarity between senses, mostly gloss overlap or semantic relatedness based on glosses. This makes sense, as glosses are a prerequisite for humans to recognize the meaning of an encoded sense, and thus also an intuitive way of judging the similarity of senses. (Matuschek, 2014)

The paper is structured as follows: we provide a brief overview of related work in Section 2, and a description of the corpus in Section 3. In Section 4 we explain all important aspects of our model implementation, while the results are presented in Section 5. Finally, we end the paper with the discussion in Section 6 and conclusion in Section 7.

2. Related Work

Similar work in monolingual word sense alignment has previously been done mostly for one language in mind, for example (Henrich et al., 2014), (Sultan et al., 2015) and (Caselli et al., 2014).

Researchers avoid modeling features according to a specific resource pair, but aim to combine generic features which are applicable to a variety of resources. One example is the work of (Matuschek and Gurevych, 2014) on alignment between Wiktionary and Wikipedia using distances calculated with Dijkstra-WSA, an algorithm which works on graph representations of resources, as well as gloss similarity values.

Recent work in monolingual corpora linking includes (McCrae and Buitelaar, 2018) which utilizes state-of-the-art methods from the NLP task of semantic textual similarity and combines them with structural similarity of ontology alignment.

Since our work is focusing on similarity of textual descriptions, it is worth mentioning that there have been lots of advances in natural language processing with pre-trained contextualized language representations relying on large corpora (Devlin et al., 2018), which have been delivering improvements in a variety of related downstream tasks, such as word sense disambiguation (Scarlini et al., 2020) and question answering (Yang et al., 2019). However, we could not find any related work leveraging the newest advances with neural network language models (NNLM) for monolingual word sense alignment. For this reason we have chosen to implement our classifiers based on two approaches: one which is feature-based, and the other one using pre-trained NNLMs.

3. Dataset

The dataset used to train and test our models was compiled specifically with this purpose in mind (Ahmadi et al., 2020). The complete corpus for the shared task consists of sixteen datasets from fifteen European languages.[1] The gold standard was obtained by manually classifying the level of semantic similarity between two definitions from two resources for the same lemma.

The data was given in four columns: lemma, part-of-speech (POS) tag and two definitions for the lemma. The fifth column which the system aims to predict contains the semantic relationship between definitions. This falls in one of the five following categories: EXACT, BROADER, NARROWER, RELATED, NONE.

The data was collected as follows: a subset of entries with the same lemma is chosen from the two dictionaries and a spreadsheet is created containing all the possible combinations of definitions from the entries. Experts are then asked to go through the list and choose the level of semantic similarity between each pair. This has created a huge number of pairs which have no relation, and thus the dataset is heavily imbalanced in favor of NONE class. Two challenges caused by the skewness of data were identified. Firstly, the models should be able to deal with underrepresented semantic relations. Secondly, evaluation metrics should consider the imbalanced distribution.

Table 1 displays the distribution of relations between two word definitions and the imbalance of the labels in the training data. We have implemented several ways to battle this, such as undersampling and oversampling, as well as doubling the broader, narrower, exact and related class by relying on their property of symmetry, or applying ensemble learning methods, such as random forest.

4. System Implementation

We aimed to explore the advantages of two different approaches, so we created two different versions of our system. One is the more standard, feature-based approach, and the other is a more novel approach with pre-trained neural language models, specifically BERT (Devlin et al., 2018). The novel approach was used for English and German dataset, in addition to the feature based approach.

4.1. Feature-based models

4.1.1. Preprocessing

Firstly, we loaded the datasets and mitigated imbalanced distribution of relation labels by swapping the two definitions and thus doubling the data samples for related labels, i.e. BROADER, NARROWER, EXACT, RELATED. For example, one English data sample for English head word *follow* has the definition pair *"keep to"* and *"to copy after; to take as an example"* and the relation *"narrower"*. We swap the order of definition pair and change the relation to *"broader"*. An outcome of this swapping process is the generalisation of the dataset. Since two definitions are from different dictionaries, features derived by comparing the two sets of definitions is dependent on the dictionaries. By swapping the definitions, more general features can be calculated, since the columns contain definitions of two dictionaries, instead of one. This aspect could make the trained feature-based models more robust against new dictionaries. After doubling the data samples, we applied upsampling to match the number of samples of NONE category.

For linguistic preprocessing, the definitions were tokenized using Spacy[2] for English and German, and NLTK[3] for other languages. For languages other than English and German, stopwords were removed from the definitions, in order to create word embedding models. Word vectors included in Spacy language models were used for English and German. We have compiled stopword lists for all languages using several resources found on the Web.[4]

4.1.2. Feature Extraction

Since many of the languages in the dataset have very few open-source resources and tools, and of uncertain quality, the features used are mostly based on word embeddings. The word embeddings were trained using the sets of definitions provided and the Word2Vec(Mikolov et al., 2013) model from *gensim(Řehůřek and Sojka, 2010)* Python library. To calculate the vector of a definition we used the average of word embeddings of consisting tokens. Sentence similarity was calculated with different similarity measures, namely cosine distance, Jaccard similarity, and word mover distance (WMD). For English and German, we used Spacy's built-in language models for word embeddings. The English language model used, *en_core_web_lg* has 685k unique vectors over 300 dimensions, while the German model, *de_core_news_md* has 20k unique vectors over 300 dimensions. Additionally, similarity calculation based on contextualized word representation ELMo (Peters et al., 2018) was used for English to model semantic differences depending on the context.

We selected a different set of features for each classification model from the features described below. Complete list of features used by each classification model is shown in Table 4.

Overall, we used the following features:

- Statistical features: Difference in length of definitions was added as a feature.
- Similarity measures based features: In addition to the word embedding comparisons between the word definition pair, we calculated similarity of the most similar word to the headword by calculating cosine similarity for list of word embeddings of tokens of definitions excluding stopwords and headword word embedding.
- Part-of-speech based features: We included one-hot encoded POS of the headword, as well as difference in POS count of two definitions as features. The POS count was not done for most languages as we were not certain in the quality of existing POS-taggers.

[1] The dataset is still growing, and the current version can be found here: https://github.com/elexis-eu/MWSA

[2] https://spacy.io/

[3] https://www.nltk.org/

[4] https://github.com/Xangis/extra-stopwords and https://www.rdocumentation.org/packages/stopwords/versions/0.1.0

Language	Broader	Narrower	Exact	Related	None	Total	None %
Basque	82	124	359	170	2496	3231	77%
Bulgarian	153	151	522	275	2256	3357	67%
Danish	172	302	1007	32	14271	15784	90%
Dutch	51	29	444	40	18656	19220	97%
English	39	310	800	51	7137	8337	85%
Estonian	92	105	921	6	1077	2201	49%
German	381	281	321	106	3322	4411	75%
Irish	62	40	664	117	1729	2612	66%
Italian	33	109	281	77	1468	1968	75%
Portuguese	3	32	178	22	1176	1411	83%
Russian	107	11	265	61	2757	3201	86%
Serbian	101	56	413	173	5052	5795	87%
Slovene	176	433	408	105	5595	6717	83%

Table 1: Label distribution of training datasets

- Lexico-syntactic features: One feature exploiting the structure of definitions was to compare the first token of definitions for equality. We also counted matching lemma in the pair of sentences and normalized by the combined length of sentences. Normalization was applied, because we wanted how much overlap exists between two definitions with respect to the length. Without normalization, longer definitions might tend to have higher number of matching lemma. Depth of dependency tree was calculated to add information about structural complexity of definitions. Occurrences of semicolons were also added, since lots of definitions were comprised of multiple short definitions concatenated by semicolon. Additionally, Root word of dependency trees were compared for each definition pair.
- Word sense based features: WordNet[5] was used to count the number of synsets of headwords. Average count of synsets were also added as feature. It was calculated by simply counting synsets for each token of definitions in wordnet and taking the average. These features were used for English only, due to the availability of its primary resource, WordNet.

Standardization was applied for some features,*length difference, pos count difference, and cosine simlarities* prior to training some machine learning models in order to bring the features to similar scale to the other features. Standardization was done by applying Scikit-learn Standard-scaler, which calculates the standardized value of feature by taking the difference of the feature value to the mean value and dividing it by standard deviation.

4.1.3. Classification Models

We tried several machine learning models, mostly from *scikit learn*[6] library for Python: logistic regression, support vector machine, random forest classifier, and decision tree. Classification models were trained by tuning hyperparameters with grid search over 5-fold cross-validation. The hyperparameters used for the submitted models are listed in Table 6. Due to imbalanced nature of the datasets, we have used balanced accuracy and weighted f1-measure for model evaluation. For languages other than English and German, we have ultimately settled for the random forest classifier as it has consistently given the best results.

4.2. Fine-tuning of Pre-trained Neural Network Language Models

For English and German, we additionally fine-tuned pre-trained neural network language models(NNLM), BERT(Devlin et al., 2018) and RoBERTa(Liu et al., 2019) in particular, using simpletransformers [7] on top of pre-trained models provided by transformers python[8] libraries on Google Cloud Platform [9].

In general, applications of pre-trained language models to downstream tasks can be categorized into feature-based and fine-tuning based approaches. Recently, BERT (Devlin et al., 2018), which stands for Bidirectional Encoder Representations from Transformers, have been proven to be beneficial for improving different downstream NLP tasks. 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). Sun et al. (2020) present different approaches to fine-tune BERT for downstream tasks, including pre-training on in-domain data, multi-task fine-tuning and different layers and learning rates.

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 pair effectively includes bidirectional cross attention between two sentences. We follow the fine-tuning approach presented in the original paper (Devlin et al., 2018), and adapt our definition pairs as input sequence $[CLS], x_1, ..., x_n[SEP]y_1, ..., y_n[EOS]$ and use [CLS] representation for classification layer.

[5]https://wordnet.princeton.edu/

[6]https://scikit-learn.org/stable/

[7]https://github.com/ThilinaRajapakse/simpletransformers

[8]https://huggingface.co/transformers/index.html

[9]https://cloud.google.com/

Language	5-class Accuracy	2-class Precision	2-class Recall	2-class F-Measure
Baseline	**0.789**	0.211	0.050	0.081
Basque	0.407	**0.223**	**0.738**	**0.342**
Baseline	**0.728**	0.250	0.011	0.020
Bulgarian	0.395	**0.331**	**0.842**	**0.475**
Baseline	**0.817**	**0.300**	0.023	0.043
Danish	0.522	0.253	**0.756**	**0.379**
Baseline	0.936	0.000	0.000	0.000
Dutch	**0.940**	**0.636**	**0.241**	**0.350**
Baseline	0.752	0.000	0.000	0.000
English	**0.766**	0.612	0.533	0.570
English BERT Large	0.654	0.467	**0.850**	0.602
English RoBERTa	0.763	**0.619**	0.782	**0.691**
Baseline	0.482	0.545	0.093	0.159
Estonian	**0.565**	**0.707**	**0.806**	**0.754**
Baseline	0.777	0.000	0.000	0.000
German	0.777	0.709	0.448	0.549
German BERT	**0.798**	**0.738**	**0.608**	**0.667**
Basline	**0.583**	**0.680**	0.185	0.291
Irish	0.549	0.631	**0.891**	**0.739**
Baseline	**0.693**	0.000	0.000	0.000
Italian	0.537	**0.418**	**0.719**	**0.529**
Baseline	**0.921**	0.083	0.024	0.037
Portuguese	0.870	**0.311**	**0.762**	**0.441**
Baseline	**0.754**	**0.438**	0.179	0.255
Russian	0.606	0.372	**0.821**	**0.512**
Baseline	**0.853**	0.000	0.000	0.000
Serbian	0.599	**0.190**	**0.464**	**0.269**
Baseline	**0.834**	0.100	0.009	0.017
Slovene	0.442	**0.173**	**0.587**	**0.268**
Average	0.615	0.413	0.694	0.414

Table 2: Comparison of evaluation Results of MWSA from the final evaluation

We have experimented with different pre-trained models, such as BERT Base, BERT Large and RoBERTa for English, which claims to have improved original BERT models by tweaking different aspects of pre-training, such as bigger data and batches, omitting of next sentence prediction, training on longer sequences and changing the masking pattern (Liu et al., 2019). For German, we used the models published by deepset.ai[10] and Bavarian State Library [11]. The training was done on NVIDIA Tesla P100 GPU, different parameter settings have been tried out to find the best performing model for each NNLM. Due to the size of the pre-trained language models and limitations in computation powers, we were only able to explore hyperparameter 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. Evaluation of the models were done by comparing Matthews Correlation Coefficient, accuracy and cross entropy. We monitored the three metrics also during training to determine when the model starts to overfit and adjusted hyperparameters for further tuning. It quickly turned out that bigger pre-trained 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. 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. Complete list of parameter values tested and the values of the submitted models are shown in Table 5.

$$w_c = \frac{total\ \#\ of\ samples}{\#\ labels \times \#\ datasamples\ of\ c} \quad (1)$$

With appropriate hyperparameters, English and German classifiers based on BERT (German) and RoBERTa (English) showed convergence with repsect to the Cross-

[10]https://deepset.ai/german-bert

[11]https://github.com/dbmdz/berts

entropy loss function. Classes were weighted according to the distribution for loss calculation. The weight for label class C, w_c is determined inversely proportional to label frequencies shown in equation 1. The values used for training is listed in Table 5

5. Results

Results of our MWSA models are presented in Table 2, including baseline models for each language provided by the organizers. In this section we explain the evaluation measures proposed by the organizers for model evaluation and review the results of the two approaches we have explored, feature-based MWSA and fine-tuning NNLM.

5.1. Evaluation Measures

The final submission was evaluated in terms of five class prediction accuracy, as well as binary classification scored with precision, recall, and F-measure. Binary evaluation metrics are calculated by considering relation labels BROADER, NARROWER, RELATED and EXACT as one class of label and NONE classified pairs as the other class. In addition, the organizers provide an average grade over all languages participated in. Our system participated for all languages excluding Hungarian and Spanish, and the results can be seen in Table 1. We argue that due to the imbalanced datasets, 5-class accuracy without balancing cannot adequately represent the model qualities and should only be interpreted holistically together with binary evaluation measures. For example, English baseline model has 5-class accuracy of 0.752, but 2-class F1-measure of 0.0 which indicates that the model is classifying the most of the definition pairs as none-related. The ratio of none related pairs in English training dataset(85%) supports this interpretation. While our both English models show similar 5-class accuracy with respect to the base classifier, they have higher 2-class f1-score, thus higher 2-class precision and recall. Table 3 additionally shows the result of our feature-based English model and RoBERTa based model in comparison with NONE classifier, which classifies all pairs as NONE. It shows that all three models have similar (5-class) accuracy with 0.76, 0.77 and 0.76. Thus, the measure is not sufficient to represent the difference in quality of the models, which can be assumed to exist when looking into the precision and recall for each label. Macro averaged or weighted averaged metrics show that our models perform better. We argue that for future work of MWSA weighted f1-measure or balanced accuracy should be used for adequate evaluation of imbalanced 5-class datasets.

5.2. Result Interpretation and Model Comparison

Our interpretation of the evaluation metrics indicates that our monolingual word sense alignment models show best overall performance for majority of languages. English and German pre-trained NNLM based models perform particularly well, while feature-based models delivered competitive overall results.

Feature-based models showed good results especially in terms of binary recall and f1-measure. However, they perform poorly when it comes to binary precision and the results vary for five-class accuracy. Aside from the peculiar aspect of 5-class accuracy for this task described above, there are several reasons for this variety in results. All the models are dependent on the quality and size of their corresponding datasets. Also our sampling strategies to deal with imbalanced data may have caused the models to overfit certain patterns of definitions pairs having some kind of relations(BROADER, NARROWER, EXACT, RELATED) and classified some of NONE-related pairs as being related, which could explain high recall and low precision. Another important aspect is the availability and quality of tools for semantic parsing and lexical resources for all the languages. To investigate the results in more detail we present precision, recall, f1-measure for label predictions of English model in Table 3. We can see that the model fails in detecting BROADER, NARROWER, and RELATED class, while performing moderately in detecting EXACT relations.

The BERT based models for English and German performed well in all binary evaluation measures, with English RoBERTa model placing first out of five teams in all three binary evaluation measures. There was no submission from other teams for German, thus no detailed analysis was possible. Nevertheless the German BERT based model outperformed the base model and achieved relatively high scores in binary precision and f-measure. For both languages the neural language model based approaches outperformed feature-based classifiers in all binary evaluation metrics. The English RoBERTa model is on par with the random forest classifier in terms of 5-class accuracy and precision, but outperforms it when it comes to binary recall and binary 2-class f-measure by significant margins. Different to the feature-based classifier, the NNLM based model manages to classify some of the NARROWER relations correctly(Table 3, but precision and recall are still very low. Confusion matrix showed that the model tends to classify NARROWER relations as EXACT. In contrast to English random forest model, German feature-based classifier cannot compete with the neural language model in all evaluation metrics, lack of more sophisticated features used by English feature-based classifier, such as ELMo sentence embedding or wordnet based features are possible reasons. However, the pre-trained German language model is pre-trained on smaller dataset (16GB of data) than English (RoBERTa: 160GB), thus it is to assume there might be room for improvement of both approaches.

For English models, which we have investigated more in detail, we can clearly see the correlation between number of data samples in each category and the performance of the models on those categories. BROADER and RELATED relations were only trained on 10 and 20 samples respectively, which we believe is too little to model pattern variety of complex natural language expressions.

6. Discussion

As previously mentioned, an important property of the provided datasets is the extreme imbalance in the favor of NONE class. For future work, it would be useful to acquire more examples of the classes less represented in the dataset. Since classifiers are prone to overfitting, it would be useful to expand the datasets with definitions extracted from more dictionaries. This way it would be easier to get a more gen-

	NONE classifier			Features-based			RoBERTa-based			
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Support
BROADER	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3
NARROWER	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.17	0.16	29
EXACT	0.00	0.00	0.00	0.44	0.60	0.51	0.47	0.74	0.58	85
RELATED	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	16
NONE	0.76	1.00	0.86	0.86	0.89	0.87	0.92	0.84	0.88	411
accuracy			0.76			0.77			0.76	
macro avg.	0.15	0.20	0.17	0.26	0.30	0.28	0.31	0.35	0.32	
weighted avg.	0.57	0.76	0.65	0.71	0.77	0.74	0.78	0.76	0.76	

Table 3: Evaluation results of test set prediction by English models. NONE classifier predicts all labels to NONE

eral and robust classifier. Our feature-based models showed that differentiating exact semantic relation is a difficult task, especially NARROWER and EXACT relations get mixed up by the English model, more work on methodologies to distinguish these relations will help to improve 5-class accuracy. A different idea to consider would be to opt for specific classifiers for each pairing of two dictionaries, where features used could be dictionary-dependant and possibly more precise, e.g. numbers of semicolons or other formatting aspects which are dictionary-specific.

Another possible issue we identified for this task is that dictionary definitions have different or atypical language usage in terms of structure of sentences, term occurrences, additional information expressed with symbols, such as semicolons, hyphens. For this reason, we think that building language models based on multiple dictionaries might help to further increase accuracy of the models.

For German and English we demonstrated that fine-tuning neural network language models outperform the feature-based approaches. Considering that the pre-trained models were trained on more general corpora, further studies involving pre-training on dictionary data and further fine-tuning different aspects described in (Sun et al., 2020) might lead to improvements of the models.

7. Conclusion

In this paper we describe our system submission for the Monolingual Word Sense Alignment shared task at Globalex 2020. Our solution consists of a separate random forest classifier trained for each language, while a BERT-based solution is implemented for English and German. The feature-based classifiers perform competitively for binary classification and employing fine-tuning of pre-trained BERT models for monolingual word sense alignment is showing promising results and should be investigated further.

8. Acknowledgements

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015.

9. References

Ahmadi, S., McCrae, J. P., Nimb, S., Troelsgard, T., Olsen, S., Pedersen, B. S., Declerck, T., Wissik, T., Monachini, M., Bellandi, A., Khan, F., Pisani, I., Krek, S., Lipp, V., Varadi, 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. L., Urena-Ruiz, R.-J., Simov, K., Osenova, P., Kancheva, Z., Radev, I., Stankovic, R., Krstev, C., Lazic, B., Markovic, A., Perdih, A., and Gabrovsek, D. (2020). A Multilingual Evaluation Dataset for Monolingual Word Sense Alignment. In *Proceedings of the 12th Language Resource and Evaluation Conference (LREC 2020)*.

Caselli, T., Strapparava, C., Vieu, L., and Vetere, G. (2014). Aligning an italian wordnet with a lexicographic dictionary: Coping with limited data.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

Henrich, V., Hinrichs, E., and Barkey, R. (2014). Aligning word senses in germanet and the dwds dictionary of the german language.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Matuschek, M. and Gurevych, I. (2014). High performance word sense alignment by joint modeling of sense distance and gloss similarity. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*.

Matuschek, M. (2014). *Word Sense Alignment of Lexical Resources*. Ph.D. thesis, Technischen Universitat Darmstadt.

McCrae, J. P. and Buitelaar, P. (2018). Linking datasets using semantic textual similarity. *Cybernetics and information technologies*, 18.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, et al., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In

Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta, May. ELRA. http://is.muni.cz/publication/884893/en.

Scarlini, B., Pasini, T., and Navigli, R. (2020). SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation.

Sultan, M. A., Bethard, S., and Sumner, T. (2015). Feature-rich two-stage logistic regression for monolingual alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Sun, C., Qiu, X., Xu, Y., and Huang, X. (2020). How to Fine-Tune BERT for Text Classification? *arXiv:1905.05583 [cs]*, February. arXiv: 1905.05583.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention Is All You Need. *arXiv:1706.03762 [cs]*, December. arXiv: 1706.03762.

Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M., and Lin, J. (2019). End-to-End Open-Domain Question Answering with BERTserini. *Proceedings of the 2019 Conference of the North*, pages 72–77. arXiv: 1902.01718.

Feature	EU	BG	DA	NL	EN	ET	DE	GA	IT	PT	RU	SR	SL
cosine sim	O		O	O		O	O	O		O		O	
jaccard sim	O		O	O		O	O	O		O		O	
tfidf similarity	O	O	O	O		O	O	O	O	O	O	O	O
elmo similarity					O								
similarity diff to target					O								
first word same	O	O	O	O		O	O	O	O	O	O	O	O
root word same	O	O	O	O		O	O	O	O	O	O	O	O
length difference	O	O	O	O	O	O	O	O	O	O	O	O	O
pos count difference				O	O		O						
target pos		O	O	O	O	O	O	O	O	O	O	O	O
lemma match count	O	O	O	O	O	O	O	O	O	O	O	O	O
pos count				O	O		O						
dep. tree depth					O								
target word synset count					O								
average synset count					O								
semicolon count					O								

Table 4: Features used for each classifier, with language codes according to ISO 639-1

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

Parameter	EU	BG	DA	NL	EN	ET	DE	GA	IT	PT	RU	SR	SL
max-features	3	3	3	auto	log2	2	auto	3	3	3	3	3	3
max-depth	10	10	10	30	10	10	30	10	10	7	10	10	10
min-samples-leaf	3	3	5	5	2	3	3	4	3	3	3	3	3
min-samples-split	10	2	10	8	5	2	8	2	8	5	2	5	8
n-estimators	100	100	100	500	300	50	500	100	200	50	50	100	100

Table 6: Hyperparameters used for Random Forest Classifier