*Proceedings of the Globalex Workshop on Linked Lexicography*, pages 69–75
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020


# MWSA Task at GlobaLex 2020: RACAI's Word Sense Alignment System using a Similarity Measurement of Dictionary Definitions

**Vasile Păiș, Dan Tufiș, Radu Ion**
Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy
CASA ACADEMIEI, 13 "Calea 13 Septembrie", Bucharest 050711, ROMANIA
{vasile, tufis, radu}@racai.ro

**Abstract**

This paper describes RACAI's word sense alignment system, which participated in the Monolingual Word Sense Alignment shared task organized at GlobaLex 2020 workshop. We discuss the system architecture, some of the challenges that we faced as well as present our results on several of the languages available for the task.

**Keywords:** word sense alignment, semantic similarity, BERT, gloss similarity

## 1. Introduction

The "Monolingual Word Sense Alignment" (MWSA) task aimed at identifying a degree of similarity between word definitions across multiple dictionaries, in the same language. For this purpose, a corpus (Ahmadi et al., 2020) was provided for multiple languages. For each language, word senses from two distinct dictionaries were extracted and participating systems had to classify the relationship between the senses in one of five categories: "exact", "broader", "narrower", "related" or "none".

Each provided entry in the evaluation set contains the following information: the lemma associated with the two definitions (the definiendum), the part of speech, two fields corresponding to the first and second dictionary entries (the definientia). Additionally, in the training set the relationship label is also provided.

Given this information, the task can be seen either as a word sense disambiguation problem, considering the sense of the definiendum in each of the definitions, or as a sentence similarity problem, considering the relatedness of the two definitions if they were sentences.

Word sense disambiguation (WSD) is the ability to identify the meaning of words in context in a computational manner (Navigli, 2009). This is an extremely hard problem, previously described as an AI-complete problem (Mallery, 1988), equivalent to solving central problems of artificial intelligence. This happens because difficult disambiguation issues can be resolved only based on knowledge. For the purpose of the MWSA task, a WSD approach will consider at each step the definiendum and its two contexts as expressed by the dictionary definitions.

Sentence similarity aims at computing a similarity measure between two sentences based on meanings and semantic content. For this purpose, the two definitions are treated like sentences and their meaning is compared. In this case the definiendum is not directly used, only the meaning expressed by the definiens being considered.

The present paper presents our system developed in the context of the MWSA shared task. We start by presenting related research, then continue with the implementation of our system and finally present concluding remarks.

## 2. Related Work

Word sense disambiguation is a very old task in natural language processing. Already in 1940s it is viewed as a fundamental task of machine translation (Weaver, 1949). Early systems employed manually created lists of disambiguation rules (Rivest, 1987). The power of these systems was demonstrated in the first Senseval competition (Kilgarriff, 2000), where decision lists were the most successful techniques employed (Yarowsky, 2000).

One of the earliest attempts at using additional digital resources in the form of machine-readable dictionaries is known as the Lesk algorithm, after its author (Lesk, 1986). In this case, the dictionary sense of a word having the highest overlap with its context (the most words in common) is considered to be the correct one. A Lesk-based similarity measure can also be computed for entire sentences. A survey of different semantic text similarity methods is given in Islam and Inkpen (2008).

With the introduction of the unsupervised distributional representation of words, new sentence similarity measures have become available. These representations are also known as "word embeddings" and include GloVe (Pennington et al., 2014), Skip-gram and CBOW (Bengio et al., 2003) and further refinements such as those described in Bojanowski et al. (2016). In all of these variants, a unique representation is computed for each word based on all the contexts it appears in. This is not directly usable for WSD since the representation remains the same regardless of the word context. However, short text or sentence similarity measures can be computed by using the word embeddings representation of each word (Kenter and Rijke, 2015). One of the advantages of using word embeddings representations is the availability of such pre-computed vectors for many languages (Grave et al., 2018), trained on a mixture of Wikipedia and Common Crawl data. Additionally, on certain languages there are pre-computed vectors available computed on more language representative corpora, such as (Păiș and Tufiș, 2018).

A more recent representation of words is represented by their contextual embeddings. Well-known models of this type are ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019). They provide a word representation in context. Therefore, as opposed to previous embedding models, the word representation is not fixed, but determined based on the actual context the word appears in at runtime.

Currently such pre-trained representations are not yet available for all languages, but multilingual models do exist, covering multiple languages in the same model, such as (Artetxe and Schwenk, 2019). Recent studies have confirmed that BERT multilingual models seem to create good representations usable in a large number of experiments, even though concerns have been expressed regarding certain language pairs (Pires et al., 2019).

Sentence-BERT (Reimers and Gurevych, 2019) is a system for determining sentence embeddings. These are representations of entire sentences that can be used to assess sentence similarity.

## 3. Dataset and Basic Processing

The dataset proposed for the MWSA task is comprised of training and test data for 15 languages. For each of the languages, a tab separated file is available for evaluation containing 4 columns (lemma, part-of-speech, first definition, second definition) with one additional column in the training data (the relatedness of the two definitions). The definitions come from two distinct sources and are related to the word presented in the first column.

As mentioned in the introduction, the definition similarity issue can be considered a sentence similarity problem. However, definitions are usually not regular sentences. Considering the "English_nuig" portion of the dataset, which consists of definitions taken from the Princeton English WordNet (Miller, 1995) and the Webster's 1913 dictionary, the following types of definitions can be identified:

- A list of synonyms (example: "a pennant; a flag or streamer", "a wing; a pinion")
- One or more expressions detailing the word (example: "not having a material body", "wild or intractable; disposed to break away from duty; untamed")
- Entire sentences (example: "a tower built by Noah's descendants (probably in Babylon) who intended it to reach up to heaven; God foiled them by confusing their language so they could no longer understand one another").

Other characteristics of definitions include:

- Further clarifications given in parentheses (example: "(Genesis 11:1-11)", "(probably in Babylon)", "(approximately)")
- Definitions tend to use a simpler language, out of more common words (usually explaining a less common word by means of common words)
- There can be additional clarifications or examples at the end of the definitions starting with "--" (example: "-- usually used of people, especially women;", "-- contrary to")
- For things like proper names or historical events there can be years or periods given in parentheses (example: "(1805)", "(1909-1984)").

For other languages in the dataset similar observations can be made. Nevertheless, some specifics can also be identified. For example, in the Dutch part of the corpus first definitions usually start with a number (example: "1.a/|\Van personen", "II.6.c/|\(Onz.) Zonder nadere bep.").

Given these corpus characteristics, a first phase before any actual algorithm implementation must consist in cleaning the definitions and pre-processing towards obtaining actual definientia. Since in most cases a single definition text actually groups together multiple simpler definitions our goal for pre-processing is to actually split them into individual ones (will also reference to them as "sub-definitions"). A first step is to split the definition text by ";" characters. However, since some of the sub-definitions may still be complex, we followed some of the approaches for sentence decomposition described in Haussmann (2011). We paid special attention to cases where multiple alternatives were given in the definition text, usually by means of coordinating conjunctions.

Taking an example definition "of plain or coarse features; uncomely; ugly; -- usually used of people, especially women" this would be expanded into 4 sub-definitions: "of plain features", "of coarse features", "uncomely" and "ugly". The final part, after the "--" is removed during the cleaning phase. Even though this final part could provide some information, it appears only in one of the definition pairs and therefore it was deemed not useful for the analysis algorithms.Further primary processing operations include lemmatization and part-of-speech tagging. Given the observations presented previously and the examples shown, we considered that a regular annotation pipeline would not produce good results, since these are usually trained on regular text, containing complete sentences. Therefore, we decided to employ a statistical based annotation, considering the most frequent lemma and part-of-speech that appears in a large enough corpus. For this purpose, we used the Open American National Corpus (Ide and Macleod, 2001) for the English language, the Spoken Dutch Corpus (Corpus Gesproken Nederlands – CGN) (Hoekstra et al., 2000) for the Dutch language, the PAISA corpus (Lyding et al., 2014) for the Italian language and the available Universal Dependencies treebanks for the Spanish language.

The choice of the aforementioned resources for lemmas and part-of-speech was justified by their public availability online as well as the relatively short timeframe allocated for the purpose of the MWSA task.

Dataset structure for the languages in which our system participated is presented in Tables 1-4 for the training part and in Table 5 for the test part. The part of speech is associated with the defined word and the relation categories "exact", "narrower", "broader", "related" and "none" are presented as they appear in the training set.

| **POS** | **Exact** | **Narr.** | **Broad.** | **Rel.** | **None** | ***Total*** |
|---|---|---|---|---|---|---|
| Noun | 409 | 143 | 11 | 16 | 2115 | *2694* |
| Verb | 230 | 100 | 19 | 25 | 4381 | *4755* |
| Adj | 149 | 58 | 7 | 8 | 588 | *810* |
| Adv | 12 | 9 | 2 | 2 | 53 | *78* |

Table 1. Dataset structure for the English training set

| **POS** | **Exact** | **Narr.** | **Broad.** | **Rel.** | **None** | ***Total*** |
|---|---|---|---|---|---|---|
| Noun | 264 | 14 | 40 | 24 | 8616 | *8958* |
| Verb | 77 | 9 | 7 | 7 | 4664 | *4766* |
| Adj | 93 | 5 | 4 | 3 | 4013 | *4118* |
| Adv | 10 | 1 | 0 | 4 | 1363 | *1378* |

Table 2. Dataset structure for the Dutch training set

| POS | Exact | Narr. | Broad. | Rel. | None | *Total* |
|---|---|---|---|---|---|---|
| Noun | 161 | 43 | 22 | 23 | 773 | *1022* |
| Verb | 120 | 66 | 11 | 54 | 695 | *946* |

Table 3. Dataset structure for the Italian training set

| POS | Exact | Narr. | Broad. | Rel. | None | *Total* |
|---|---|---|---|---|---|---|
| Noun | 350 | 72 | 50 | 38 | 1718 | *2228* |
| Verb | 129 | 24 | 22 | 10 | 865 | *1051* |
| Adj | 160 | 29 | 19 | 16 | 767 | *991* |
| Adv | 20 | 0 | 0 | 1 | 50 | *71* |
| Conj. | 2 | 1 | 0 | 3 | 22 | *28* |
| Adp. | 4 | 0 | 0 | 1 | 44 | *49* |
| Affix | 5 | 1 | 0 | 1 | 27 | *34* |
| Interj. | 1 | 0 | 0 | 0 | 0 | *1* |

Table 4. Dataset structure for the Spanish training set

| | Noun | Verb | Adjective | Adverb | Total |
|---|---|---|---|---|---|
| English | 177 | 262 | 100 | 5 | 544 |
| Dutch | 834 | 0 | 90 | 0 | 924 |
| Italian | 136 | 69 | 0 | 0 | 205 |
| Spanish | 171 | 119 | 150 | 4 | 444 |

Table 5. Dataset structure for test sets

Some common observations can be extracted from the above tables. In all the analyzed languages the predominant parts-of-speech associated with the entries are nouns and verbs, in both training and test sets. Additional part of speech words present are usually adjectives and adverbs. For the Italian dataset only nouns and verbs are provided while the Spanish data set also has a few entries (a total of 112) with other part of speech tags, present only in the training set: conjunction, adposition, affix, interjection.

Considering the English dataset alone, the nouns and verbs together total 7449 entries while the rest account for only 888 entries. From this point of view, it is expected that any system trained on the training set and making use of part-of-speech information will probably work better on nouns and verbs.

With regard to relationship classes, for all datasets it seems the "none" class is the most used, followed by the "exact" class. For the English dataset, the "none" class accounts for 7137 entries, the "exact" class has 800 entries and all the other classes account for 400 entries. Given this huge difference between the available examples associated with each class, it is expected that a system trained on this dataset will perform better on "none" and "exact" and less on the other classes.

## 4. System Architecture

The overall system is constructed as a series of modules that can be turned on or off depending on what resources are available for a certain language. Each module produces one or more features that can be finally fed into a decision tree or random forest classifier, thus producing the final result. The overall system diagram is presented in Figure 1.

The first two modules "Cleanup" and "Definition decomposition" were already presented in the previous section. Their functionality is about obtaining clean sub-definitions. The following modules usually make use of these sub-definitions, but there are also features computed on the entire definition directly after the cleanup pre-processing. Modules using sub-definitions, as detailed below, will compute a score for each sub-definition pair. Finally, the scores are combined by selecting the maximum score between all sub-definition pairs.

The first series of features is based on variants of the Lesk algorithm. We use three types of algorithms based on complete words, lemmas and stems. For each sub-definition pair (the first taken from the first definition and the second from the second definition) we compute a score based on the common indicators between the two. Finally, the algorithm keeps the maximum number of words in common as well as the maximum and minimum number of words in the sub-definitions corresponding to the first and second definition. For stemming we used a Porter stemmer algorithm (Rijsbergen, 1980; Porter, 1980).

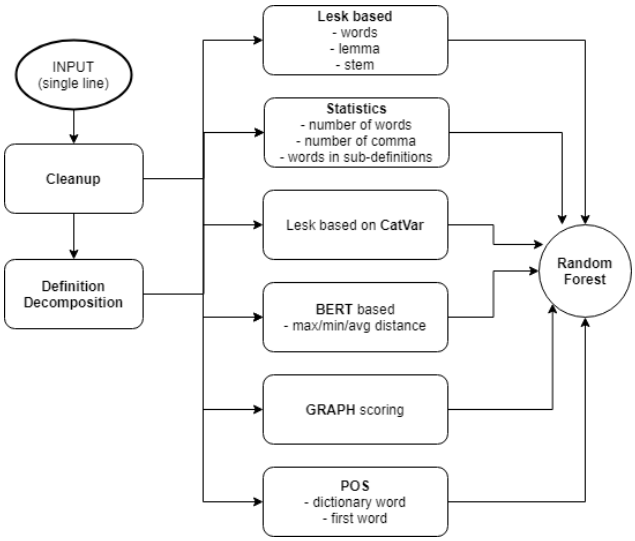


Figure 1. System architecture

An additional enhancement was realized by implementing a Lesk algorithm variant by incorporating the cluster information from the Categorial Variation Database (Catvar) (Habash and Dorr, 2003). Catvar is a database of clusters of uninflected words (lexemes) and their categorial (i.e. part-of-speech) variants.

As mentioned in the "Related work" section, BERT is a word embeddings model allowing for word representation in context and this representation was used in Sentence-BERT (Reimers and Gurevych, 2019) for obtaining sentence-level representations. We exploited this by incorporating a series Sentence-BERT based features. Thus, for each sub-definition pair we computed the Sentence-BERT representation and obtained the cosine distance between those. Finally, the minimum, maximum and average distances were computed and used as features. Also, a complete embedding was computed on the entire definition and the cosine distance between the two definitions was used as another feature.

A novel algorithm was implemented using a graph representation. For each sub-definition pair, the component words were added to the graph. Then, the lemmas of the words were added. Finally, synonyms and related words (see below) were added as well. These were extracted from WordNet. The extraction process involves a further sense disambiguation in order to detect relevant synsets. This was achieved using a basic Lesk-based

disambiguation algorithm between the synset definition available in WordNet and the input sub-definition. In order to exploit the word order within the sub-definitions and allow for missing words, additional edges were added between adjacent words in the sub-definitions. An example is given in Figure 2 for the sub-definitions "refuse to accept" and "refuse to receive". This is a very simple example in which a word appears in both sub-definitions and the remaining words are actually detected as being synonyms.

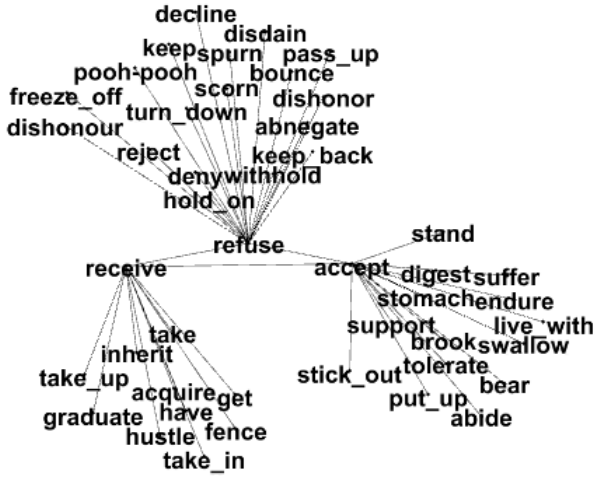


Figure 2. Example graph-based representation for "refuse to accept" and "refuse to receive"

Finally, a score was computed based on the distance between words belonging to the two sub-definitions.

Since all these algorithms make use of statistics or pre-trained word vectors without further optimization on the training corpus, we present results from each algorithm alone in Table 6.

| **Algorithm** | **Accuracy 5-class** |
|---|---|
| Lesk words | 0.8502 |
| Lesk lemma | 0.8501 |
| Lesk stem | 0.8496 |
| Lesk Catvar | 0.8221 |
| Graph | 0.8539 |
| BERT avg. sub-definitions | 0.8676 |

Table 6. Accuracy results from different algorithms on the English training set

From Table 6 it can be seen that the BERT average calculation on the sub-definitions seems to produce the best accuracy score. However, by comparing the different algorithms it seems that each algorithm produces good results in different contexts (considering the observations from section 3, above). Therefore, the final classification module becomes very important, especially combined with other features that could allow a decision between different scores.

Statistical features which were computed included the total number of words, minimum and maximum number of words in sub-definitions, number of comma characters.

Furthermore, from several manual investigations on the training data it was deemed useful to have a comparison between the first words of sub-definitions having the same part of speech as the defined word. This comparison is realized by means of synonyms and is further used as a feature. For example, let's consider the sub-definitions associated with the word "holograph" which has the indicated part of speech "noun": "handwritten book" and "a document". In this case we are interested in comparing "book" and "document" since these have the same part of speech ("noun") as the defined word.

Furthermore, considering the observations regarding definition structure from section 3, an additional feature was created with 3 possible values: 0, if both sub-definitions are single word (not considering stop words); 1, if one of the sub-definitions is a single word and the other is a more complex expression; 2, if both sub-definitions are complex expressions.

A total of 17 features were finally used in a Random Forest Classifier (Ho, 1995). The classifier hyperparameters were trained and optimized using a grid search approach with cross validation on the training set.

The final cross validation measurement of mean accuracy on the training set indicated a value of 0.881 with a variation of +/- 0.02. This is above the score obtained on the test set, thus indicating some potentially significant variations in the data used. Nevertheless, our system obtained a final score of 0.798 on the 5-class accuracy evaluation, thus positioning the system on the first place for the English language competition.

For the other languages in which we participated (Dutch, Italian and Spanish) we deactivated the modules using WordNet based synonyms. We acknowledge the existence of wordnets for the aforementioned languages, however due to the short amount of time available for the task we were not able to technically integrate these resources into our system. Nevertheless, this was an exercise proving the modularity of the developed system and the possibility to adapt to different available resources. Furthermore, even with this disadvantage, the system was able to be on the first place for the Dutch language and on second place for Italian and Spanish.

## 5. System Evaluation

Once the test set annotations were released, we were able to evaluate our system, including all the other algorithms on the final data. Table 5, above, already contains an analysis of the test dataset part-of-speech structure. Distribution of available gold annotations in the test dataset are presented in tables 7-10 for the English, Dutch, Italian and Spanish languages.

| **POS** | **Exact** | **Narr.** | **Broad.** | **Rel.** | **None** | ***Total*** |
|---|---|---|---|---|---|---|
| Noun | 39 | 18 | 0 | 2 | 118 | *177* |
| Verb | 31 | 11 | 1 | 10 | 209 | *262* |
| Adj | 14 | 0 | 2 | 4 | 80 | *100* |
| Adv | 1 | 0 | 0 | 0 | 4 | *5* |

Table 7. Dataset structure for the English test set

| **POS** | **Exact** | **Narr.** | **Broad.** | **Rel.** | **None** | ***Total*** |
|---|---|---|---|---|---|---|
| Noun | 40 | 1 | 10 | 1 | 782 | *834* |
| Adj | 3 | 0 | 3 | 0 | 84 | *90* |

Table 8. Dataset structure for the Dutch test set

| POS | Exact | Narr. | Broad. | Rel. | None | *Total* |
|---|---|---|---|---|---|---|
| Noun | 23 | 6 | 2 | 8 | 97 | *136* |
| Verb | 5 | 9 | 3 | 1 | 51 | *69* |

Table 9. Dataset structure for the Italian test set

| POS | Exact | Narr. | Broad. | Rel. | None | *Total* |
|---|---|---|---|---|---|---|
| Noun | 29 | 8 | 4 | 1 | 129 | *171* |
| Verb | 17 | 5 | 0 | 0 | 97 | *119* |
| Adj | 24 | 12 | 5 | 3 | 106 | *150* |
| Adv | 2 | 0 | 0 | 0 | 2 | *4* |

Table 10. Dataset structure for the Spanish test set

Test dataset similarity tags follow a distribution like that of the training set. However, the distinction between "exact" and "none" classes is emphasized even more. In the English, Dutch and Spanish datasets there are cases where the number of "narrower", "broader" or "related" tags is equal to zero for certain parts of speech. By looking at the total numbers of tags in each category in the English data set, it can be observed that there are only three of type "broader". Similarly, for the other languages analyzed there are tags for which the total number is equal to or less than 5.

The official evaluation was performed using the CodaLab website[1]. Results on the test datasets for our system are presented in Table 11. This evaluation contains 4 indicators: accuracy (the percentage of scores for which the predicted label matches the reference label, considering all five classes), precision, recall and F-measure (taking into account accuracy in predicting the link but not the type of the link, thus considering only 2 classes: none and non-none).

| | 5-Class Accuracy | 2-Class Precision | 2-Class Recall | 2-Class F-measure |
|---|---|---|---|---|
| English | 0.798 | 0.746 | 0.353 | 0.480 |
| Dutch | 0.944 | 0.846 | 0.190 | 0.310 |
| Italian | 0.761 | 0.760 | 0.333 | 0.463 |
| Spanish | 0.786 | 0.667 | 0.655 | 0.661 |

Table 11. System evaluation on the test dataset

Our system obtained first place for the English and Dutch accuracy score (considering all 5 classes) and second place for the Italian and Spanish accuracy. Probably the lower score for Italian and Spanish is due to the fewer language resources that we used and thus to the fewer modules of the system that were involved, as described in section 4.

Looking at the 2-class measures, our system reached high precision and was on the first place for English and Dutch and on the second place for Italian and Spanish. Compared to other systems our recall was lower resulting in a F-measure that situated our system on second and third place with regard to this metric.

Similar to the individual algorithm evaluation provided in Table 6 on the training set, we provide accuracies on the test set for the English language in Table 12.

As mentioned in section 4, these algorithms are not dependent on the training set, being statistical in nature, therefore we would expect seeing similar scores. However, a slightly lower score than the one on the training set could be attributed to a potential difference between the two sets. Tables 1 and 7 provide comparison between the training and test sets for the English language and one of the possible differences is the high number of nouns in the training set as compared to the more balanced number of nouns and verbs in the test set. Another difference is the reduced number of "narrower", "broader" and "related" definitions.

| Algorithm | Accuracy 5-class |
|---|---|
| Lesk words | 0.6985 |
| Lesk lemma | 0.6912 |
| Lesk stem | 0.6930 |
| Lesk Catvar | 0.6415 |
| Graph | 0.7445 |
| BERT avg. sub-definitions | 0.7096 |

Table 12. Accuracy results from different algorithms on the English test set

The addition of a Random Forest classifier combining all the available features improved the overall accuracy from 0.744 (in the case of the Graph-based algorithm, which obtained the highest individual score) to 0.798, which was the final score achieved by our system on the English language.

## 6. Conclusions and Future Work

This paper presented our system proposal[2] for the Monolingual Word Sense Alignment 2020 shared task. The system is composed of multiple modules which can be enabled or not depending on the linguistic resources available for a particular language. Finally, a random forest classifier is trained on the provided training dataset using the features produced by the different modules. The system was able to achieve state-of-the-art performance for the English language, by using all the implemented modules, as described in section 4 above. Furthermore, with a reduced set of modules, due to the resources available to us in the short amount of time for this competition, we were able to achieve first place in the Dutch language competition and second place in the Italian and Spanish competitions.

The overall system contains both language independent modules (like some of the Lesk based approaches and purely statistical features) and modules requiring the presence of language resources. In the second case, these range from basic resources (synonyms, stemming algorithms) to more advanced resources (WordNet, lemmatization, part of speech tagging) and even the presence of a BERT model (either multilingual or language specific).

Having a modular architecture means the system can be used on any language and it can adapt itself (also its results) to the available resources. As always, having more language resources available translates into a better system performance. Of course, integrating resources for additional languages requires manual intervention on the system to allow it to process the new resources in their respective formats. This also explains our limited participation in the task's languages since we had to integrate different resources (with different formats) available for the different languages.

[1] https://competitions.codalab.org/competitions/22163

[2] https://github.com/racai-ai/MWSA2020

Implemented modules can be used individually, even without a training set. This set was needed in the last stage when training the final classifier together with additional statistical features. Therefore, it is our hope that this implementation can be adapted for Romanian language as well. Currently a large annotated Reference Corpus of Contemporary Romanian Language (CoRoLa) (Mititelu et al., 2018) is available for our research together with the Romanian WordNet (Tufiș et al, 2008). Currently, as far as we know, there is no monolingual BERT model available for Romanian language. However, multilingual models, similar to the one used for the purpose of the MWSA task, are available. Finally, we envisage to further include such a system in the RELATE platform (Păiș et al., 2019) dedicated to processing Romanian language.

## 7. Acknowledgements

Part of this work was conducted in the context of the ReTeRom project. Part of this work was conducted in the context of the Marcell project.

## 8. Bibliographical References

Ahmadi, S., McCrae, P.J., Nimb, S., Troelsgård, T., Olsen, S., Pedersen, S.B., Declerck, T., Wissik, T., Monachini, M., Bellandi, A., Khan, F., Pisani, I., Krek, S., Lipp, V., Váradi, T., Simon, L., Győrffy, A., Tiberius, C., Schoonheim, T., Moshe, B.Y., Rudich, M., Ahmad, A.R., Lonke, D., Kovalenko, K., Langemets, M., Kallas, J., Dereza, O., Fransen, T., Cillessen, D., Lindemann, D., Alonso, M., Salgado, A., Sancho, L.J., Ureña-Ruiz, R., Simov, K., Osenova, P., Kancheva, Z., Radev, I., Stanković, R., Krstev, C., Lazić, B., Marković, A., Perdih, A. and Gabrovšek, D. (2020). A Multilingual Evaluation Dataset for Monolingual Word Sense Alignment. Proceedings of the 12th Language Resource and Evaluation Conference (LREC 2020).

Artetxe, M. and Schwenk, H. (2019). Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. Transactions of the Association for Computational Linguistics. 7. 597-610. 10.1162/tacl_a_00288.

Bengio, Y., Ducharme, R., Vincent, P. (2003). A neural probabilistic language model, Journal of Machine Learning Research, 3, pp.1137–1155.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2016). Enriching Word Vectors with Subword Information, arXiv:1607.04606.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A. and Mikolov, T. (2018). Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, arXiv:1802.06893.

Devlin J., Chang, M.W., Lee, K. and Toutanova K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186.

Habash, N. and Dorr, B. (2003). A Categorial Variation Database for English. In *Proceedings of the North American Association for Computational Linguistics*, Edmonton, Canada, pp. 96 -102.

Haussmann, E. (2011). Contextual sentence decomposition with applications to semantic full-text search. Master's thesis, University of Freiburg.

Ho, T. K. (1995). Random Decision Forests. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, QC, 14–16 August 1995. pp. 278–282.

Hoekstra, H., M. Moortgat, I. Schuurman & T. van der Wouden (2000). Syntactic Annotation for the Spoken Dutch Corpus Project (CGN). In W. Daelemans, K. Sima'an, J. Veenstra & J. Zavrel (Eds.), Computational Linguistics in the Netherlands 2000. 73-87. Amsterdam: Rodopi.

Ide, N., Macleod, C. (2001). The American National Corpus: A Standardized Resource of American English. Proceedings of Corpus Linguistics 2001, Lancaster UK.

Islam, A. and Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. ACM Trans. Knowl. Discov. Data. 2, 2, Article 10 (July 2008), 25 pages.

Kenter, T. and Rijke, M. (2015). Short text similarity with word embeddings. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pp 1411-1420.

Kilgarriff, A., Palmer, M. (eds., 2000): Senseval98: Evaluating Word Sense Disambiguation Systems, vol. 34 (1–2). Kluwer, Dordrecht, the Netherlands.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th SIGDOC* (New York, NY). 24–26.

Lyding, V., Stemle, E., Borghetti, C., Brunello, M., Castagnoli, S., Dell'Orletta, F., Dittmann, H., Lenci, A., Pirrelli, V. (2014): "The PAISÀ Corpus of Italian Web Texts" In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, Association for Computational Linguistics, Gothenburg, Sweden, April 2014. pp. 36-43.

Mallery, J. C. (1988). Thinking about foreign policy: Finding an appropriate role for artificial intelligence computers. Ph.D. dissertation. MIT Political Science Department, Cambridge, MA.

Miller, G.A. (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-4.

Mititelu, B.V., Tufiș, D. and Irimia, E. (2018). The Reference Corpus of Contemporary Romanian Language (CoRoLa). In Proceedings of the 11th Language Resources and Evaluation Conference – LREC'18, Miyazaki, Japan, European Language Resources Association (ELRA).

Navigli, R. (2009). Word Sense Disambiguation: A Survey. ACM Computing Surveys. Vol. 41, No. 2.

Păiș, V., Tufiș, D. (2018). Computing distributed representations of words using the COROLA corpus. In Proceedings of the Romanian Academy, Series A, Volume 19, Number 2/2018, pp. 403–409.

Păiș, V., Tufiș, D. and Ion, R. (2019). Integration of Romanian NLP tools into the RELATE platform. In Proceedings of the International Conference on Linguistic Resources and Tools for Processing Romanian Language – CONSILR 2019, pages 181-192.

Pennington, J., Socher, R. and Manning C.D. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pp 1532-1543.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L. (2018). Deep

contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1, pp. 2227-2237.

Pires, T., Schlinger, E. and Garette, D. (2019). How multilingual is Multilingual BERT? arXiv:1906.01502.

Porter, M.F. (1980). An algorithm for suffix stripping, Program, 14(3) pp 130−137.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp 3982-3992.

Rijsbergen, C.J., Robertson, S.E. and Porter, M.F. (1980). New models in probabilistic information retrieval. London: British Library. British Library Research and Development Report, no. 5587.

Rivest, R. L. (1987). Learning decision lists. Mach. Learn. 2, 3, 229–246.

Tufiș, D., Ion, R., Bozianu, L. Ceaușu, A. and Ștefănescu, D. (2008). Romanian Wordnet: Current State, New Applications and Prospects. In *Proceedings of the 4th Global WordNet Conference, GWC-2008*, pp. 441-452.

Weaver, W. (1949). Translation. In Machine Translation of Languages: Fourteen Essays (written in 1949, published in 1955), W. N. Locke and A. D. Booth, Eds. Technology Press of MIT, Cambridge, MA, and John Wiley & Sons, New York, NY, 15–23.

Yarowsky, D. (2000). Hierarchical decision lists for word sense disambiguation. Comput. Human. 34, 1-2, 179–186.