An Unsupervised Word Sense Disambiguation System for Under-Resourced Languages

Dmitry Ustalov[*†], **Denis Teslenko**[†], **Alexander Panchenko**[‡], **Mikhail Chernoskutov**[†], **Chris Biemann**[‡], **Simone Paolo Ponzetto**[*]

[*] Data and Web Science Group, University of Mannheim, Germany
[†] Ural Federal University, Russia
[‡] Universität Hamburg, Department of Informatics, Language Technology Group, Germany
{dmitry,simone}@informatik.uni-mannheim.de, teslenkoden@gmail.com,
mikhail.chernoskutov@urfu.ru, {panchenko,biemann}@informatik.uni-hamburg.de

Abstract

In this paper, we present Watasense, an unsupervised system for word sense disambiguation. Given a sentence, the system chooses the most relevant sense of each input word with respect to the semantic similarity between the given sentence and the synset constituting the sense of the target word. Watasense has two modes of operation. The sparse mode uses the traditional vector space model to estimate the most similar word sense corresponding to its context. The dense mode, instead, uses synset embeddings to cope with the sparsity problem. We describe the architecture of the present system and also conduct its evaluation on three different lexical semantic resources for Russian. We found that the dense mode substantially outperforms the sparse one on all datasets according to the adjusted Rand index.

Keywords: word sense disambiguation, system, synset induction

1. Introduction

Word sense disambiguation (WSD) is a natural language processing task of identifying the particular word senses of polysemous words used in a sentence. Recently, a lot of attention was paid to the problem of WSD for the Russian language (Lopukhin and Lopukhina, 2016; Lopukhin et al., 2017; Ustalov et al., 2017). This problem is especially difficult because of both linguistic issues – namely, the rich morphology of Russian and other Slavic languages in general – and technical challenges like the lack of software and language resources required for addressing the problem.

To address these issues, we present Watasense, an unsupervised system for word sense disambiguation. We describe its architecture and conduct an evaluation on three datasets for Russian. The choice of an unsupervised system is motivated by the absence of resources that would enable a supervised system for under-resourced languages. Watasense is not strictly tied to the Russian language and can be applied to any language for which a tokenizer, part-of-speech tagger, lemmatizer, and a sense inventory are available.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the Watasense word sense disambiguation system, presents its architecture, and describes the unsupervised word sense disambiguation methods bundled with it. Section 4 evaluates the system on a gold standard for Russian. Section 5 concludes with final remarks.

2. Related Work

Although the problem of WSD has been addressed in many SemEval campaigns (Navigli et al., 2007; Agirre et al., 2010; Manandhar et al., 2010, *inter alia*), we focus here on word sense disambiguation *systems* rather than on the research methodologies.

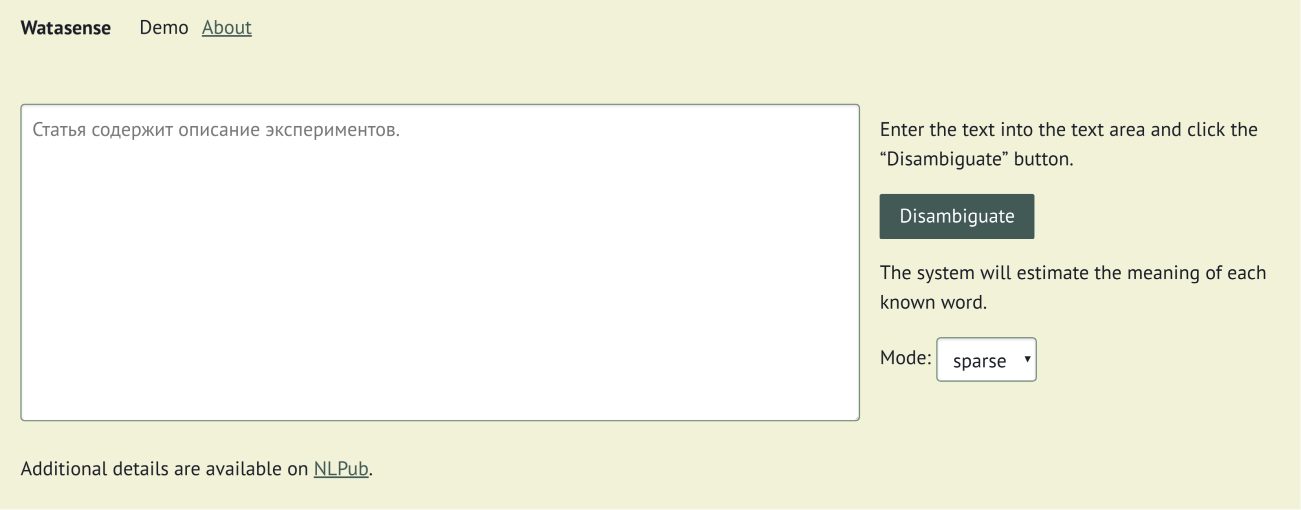
Among the freely available systems, IMS ("It Makes Sense") is a supervised WSD system designed initially for the English language (Zhong and Ng, 2010). The system uses a support vector machine classifier to infer the particular sense of a word in the sentence given its contextual sentence-level features. Pywsd is an implementation of several popular WSD algorithms implemented in a library for the Python programming language.[1] It offers both the classical Lesk algorithm for WSD and path-based algorithms that heavily use the WordNet and similar lexical ontologies. DKPro WSD (Miller et al., 2013) is a general-purpose framework for WSD that uses a lexical ontology as the sense inventory and offers the variety of WordNet-based algorithms. Babelfy (Moro et al., 2014) is a WSD system that uses BabelNet, a large-scale multilingual lexical ontology available for most natural languages. Due to the broad coverage of BabelNet, Babelfy offers entity linking as part of the WSD functionality.

Panchenko et al. (2017b) present an unsupervised WSD system that is also knowledge-free: its sense inventory is induced based on the JoBimText framework, and disambiguation is performed by computing the semantic similarity between the context and the candidate senses (Biemann and Riedl, 2013). Pelevina et al. (2016) proposed a similar approach to WSD, but based on dense vector representations (word embeddings), called SenseGram. Similarly to SenseGram, our WSD system is based on averaging of word embeddings on the basis of an automatically induced sense inventory. A crucial difference, however, is that we induce our sense inventory from synonymy dictionaries and not distributional word vectors. While this requires more manually created resources, a potential advantage of our approach is that the resulting inventory contains less noise.

3. Watasense, an Unsupervised System for Word Sense Disambiguation

Watasense is implemented in the Python programming language using the scikit-learn (Pedregosa and others,

[1] https://github.com/alvations/pywsd



Watasense Demo About

Статья содержит описание экспериментов.

Enter the text into the text area and click the "Disambiguate" button.

Disambiguate

The system will estimate the meaning of each known word.

Mode: sparse ▾

Additional details are available on NLPub.

Figure 1: A snapshot of the online demo, which is available at http://watasense.nlpub.org/ (in Russian).

2011) and Gensim (Řehuřek and Sojka, 2010) libraries. Watasense offers a Web interface (Figure 1), a command-line tool, and an application programming interface (API) for deployment within other applications.

3.1. System Architecture

A sentence is represented as a list of *spans*. A span is a quadruple: (w, p, l, i), where w is the word or the token, p is the part of speech tag, l is the lemma, i is the position of the word in the sentence. These data are provided by tokenizer, part-of-speech tagger, and lemmatizer that are specific for the given language. The WSD results are represented as a map of spans to the corresponding word sense identifiers.

The sense inventory is a list of synsets. A synset is represented by three bag of words: the synonyms, the hypernyms, and the union of two former – the *bag*. Due to the performance reasons, on initialization, an inverted index is constructed to map a word to the set of synsets it is included into.

Each word sense disambiguation method extends the `BaseWSD` class. This class provides the end user with a generic interface for WSD and also encapsulates common routines for data pre-processing. The inherited classes like `SparseWSD` and `DenseWSD` should implement the `disambiguate_word(...)` method that disambiguates the given word in the given sentence. Both classes use the *bag* representation of synsets on the initialization. As the result, for WSD, not just the synonyms are used, but also the hypernyms corresponding to the synsets. The UML class diagram is presented in Figure 2.

Watasense supports two sources of word vectors: it can either read the word vector dataset in the binary Word2Vec format or use Word2Vec-Pyro4, a general-purpose word vector server.[2] The use of a remote word vector server is recommended due to the reduction of memory footprint per each Watasense process.

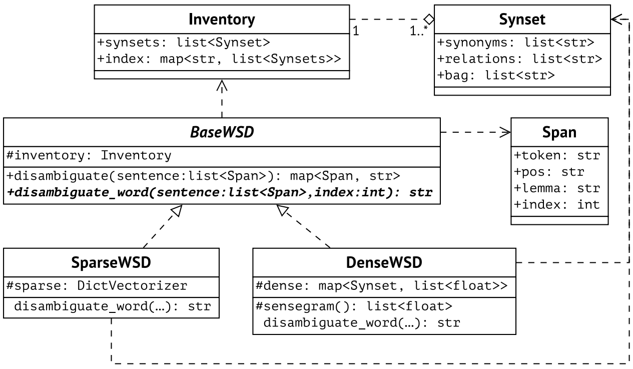

Inventory
+synsets: list<Synset>
+index: map<str, list<Synsets>>
1
1..*
Synset
+synonyms: list<str>
+relations: list<str>
+bag: list<str>
BaseWSD
#inventory: Inventory
+disambiguate(sentence:list<Span>): map<Span, str>
+disambiguate_word(sentence:list<Span>,index:int): str
Span
+token: str
+pos: str
+lemma: str
+index: int
SparseWSD
#sparse: DictVectorizer
disambiguate_word(...): str
DenseWSD
#dense: map<Synset, list<float>>
#sensegram(): list<float>
disambiguate_word(...): str

Figure 2: The UML class diagram of Watasense.

3.2. User Interface

Figure 1 shows the Web interface of Watasense. It is composed of two primary activities. The first is the text input and the method selection (Figure 1). The second is the display of the disambiguation results with part of speech highlighting (Figure 3). Those words with resolved polysemy are underlined; the tooltips with the details are raised on hover.

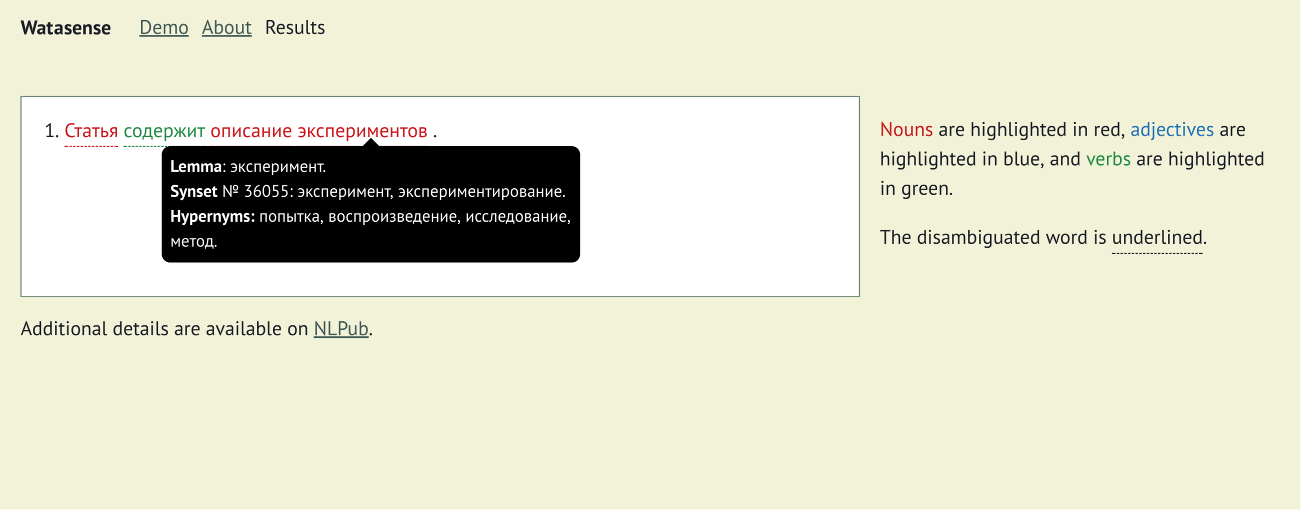
3.3. Word Sense Disambiguation

We use two different unsupervised approaches for word sense disambiguation. The first, called 'sparse model', uses a straightforward sparse vector space model, as widely used in Information Retrieval, to represent contexts and synsets. The second, called 'dense model', represents synsets and contexts in a dense, low-dimensional space by averaging word embeddings.

Sparse Model. In the vector space model approach, we follow the sparse context-based disambiguated method (Faralli et al., 2016; Panchenko et al., 2017b). For estimating the sense of the word w in a sentence, we search for such a synset $\hat{w}$ that maximizes the cosine similarity to the sentence vector:

$$\hat{w} = \arg\max_{S \ni w} \cos(S, T), \quad (1)$$

[2] https://github.com/nlpub/word2vec-pyro4



1. Статья содержит описание экспериментов .

Lemma: эксперимент.
Synset № 36055: эксперимент, экспериментирование.
Hypernyms: попытка, воспроизведение, исследование, метод.

Nouns are highlighted in red, adjectives are highlighted in blue, and verbs are highlighted in green.

The disambiguated word is underlined.

Additional details are available on NLPub.

Figure 3: The word sense disambiguation results with the word “experiments” selected. The tooltip shows its lemma “experiment”, the synset identifier (36055), and the words forming the synset “experiment”, “experimenting” as well as its hypernyms “attempt”, “reproduction”, “research”, “method”.

where S is the set of words forming the synset, T is the set of words forming the sentence. On initialization, the synsets represented in the sense inventory are transformed into the tf–idf-weighted word-synset sparse matrix efficiently represented in the memory using the compressed sparse row format. Given a sentence, a similar transformation is done to obtain the sparse vector representation of the sentence in the same space as the word-synset matrix. Then, for each word to disambiguate, we retrieve the synset containing this word that maximizes the cosine similarity between the sparse sentence vector and the sparse synset vector. Let $w_{\max}$ be the maximal number of synsets containing a word and $S_{\max}$ be the maximal size of a synset. Therefore, disambiguation of the whole sentence T requires $O(|T| \times w_{\max} \times S_{\max})$ operations using the efficient sparse matrix representation.

Dense Model. In the synset embeddings model approach, we follow SenseGram (Pelevina et al., 2016) and apply it to the synsets induced from a graph of synonyms. We transform every synset into its dense vector representation by averaging the word embeddings corresponding to each constituent word:

$$\vec{S} = \frac{1}{|S|} \sum_{w \in S} \vec{w}, \tag{2}$$

where $\vec{w}$ denotes the word embedding of w. We do the same transformation for the sentence vectors. Then, given a word w, a sentence T, we find the synset $\hat{w}$ that maximizes the cosine similarity to the sentence:

$$\hat{w} = \arg\max_{S \ni w} \cos\left(\frac{\sum_{u \in S} \vec{u}}{|S|}, \frac{\sum_{u \in T} \vec{u}}{|T|}\right). \tag{3}$$

On initialization, we pre-compute the dense synset vectors by averaging the corresponding word embeddings. Given a sentence, we similarly compute the dense sentence vector by averaging the vectors of the words belonging to non-auxiliary parts of speech, i.e., nouns, adjectives, adverbs, verbs, etc. Then, given a word to disambiguate, we retrieve the synset that maximizes the cosine similarity between the dense sentence vector and the dense synset vector. Thus, given the number of dimensions d, disambiguation of the whole sentence T requires $(|T| \times w_{\max} \times d)$ operations.

4. Evaluation

We conduct our experiments using the evaluation methodology of SemEval 2010 Task 14: Word Sense Induction & Disambiguation (Manandhar et al., 2010). In the gold standard, each word is provided with a set of instances, i.e., the sentences containing the word. Each instance is manually annotated with the single sense identifier according to a pre-defined sense inventory. Each participating system estimates the sense labels for these ambiguous words, which can be viewed as a clustering of instances, according to sense labels. The system’s clustering is compared to the gold-standard clustering for evaluation.

4.1. Quality Measure

The original SemEval 2010 Task 14 used the V-Measure external clustering measure (Manandhar et al., 2010). However, this measure is maximized by clustering each sentence into his own distinct cluster, i.e., a ‘dummy’ singleton baseline. This is achieved by the system deciding that every ambiguous word in every sentence corresponds to a different word sense. To cope with this issue, we follow a similar study (Lopukhin et al., 2017) and use instead of the adjusted Rand index (ARI) proposed by Hubert and Arabie (1985) as an evaluation measure.

In order to provide the overall value of ARI, we follow the addition approach used in (Lopukhin et al., 2017). Since the quality measure is computed for each lemma individually, the total value is a weighted sum, namely

$$\mathrm{ARI} = \frac{1}{\sum_w |I(w)|} \sum_w \mathrm{ARI}_w \times |I(w)|\,, \tag{4}$$

where w is the lemma, $I(w)$ is the set of the instances for the lemma w, ARI_w is the adjusted Rand index computed for the lemma w. Thus, the contribution of each lemma to the total score is proportional to the number of instances of this lemma.

4.2. Dataset

We evaluate the word sense disambiguation methods in Watasense against three baselines: an unsupervised approach for learning multi-prototype word embeddings called AdaGram (Bartunov et al., 2016), same sense for all the instances per lemma (One), and one sense per instance (Singletons). The AdaGram model is trained on the combination of RuWac, Lib.Ru, and the Russian Wikipedia with the overall vocabulary size of 2 billion tokens (Lopukhin et al., 2017).

As the gold-standard dataset, we use the WSD training dataset for Russian created during RUSSE'2018: A Shared Task on Word Sense Induction and Disambiguation for the Russian Language (Panchenko et al., 2018). The dataset has 31 words covered by 3 491 instances in the *bts-rnc* subset and 5 words covered by 439 instances in the *wiki-wiki* subset.[3]

The following different sense inventories have been used during the evaluation:

- **WATLINK**, a word sense network constructed automatically. It uses the synsets induced in an unsupervised way by the WATSET[CW$_{\text{nolog}}$, MCL] method (Ustalov et al., 2017) and the semantic relations from such dictionaries as Wiktionary referred as *Joint+Exp+SWN* in Ustalov (2017). This is the only automatically built inventory we use in the evaluation.

- **RuThes**, a large-scale lexical ontology for Russian created by a group of expert lexicographers (Loukachevitch, 2011).[4]

- **RuWordNet**, a semi-automatic conversion of the RuThes lexical ontology into a WordNet-like structure (Loukachevitch et al., 2016).[5]

Since the *Dense* model requires word embeddings, we used the 500-dimensional word vectors from the Russian Distributional Thesaurus (Panchenko et al., 2017a).[6] These vectors are obtained using the Skip-gram approach trained on the `lib.rus.ec` text corpus.

4.3. Results

We compare the evaluation results obtained for the *Sparse* and *Dense* approaches with three baselines: the AdaGram model (AdaGram), the same sense for all the instances per lemma (One) and one sense per instance (Singletons). The evaluation results are presented in Table 1. The columns bts-rnc and wiki-wiki represent the overall value of ARI according to Equation (4). The column Avg. consists of the weighted average of the datasets w.r.t. the number of instances.

We observe that the SenseGram-based approach for word sense disambiguation yields substantially better results in every case (Table 1). The primary reason for that is the implicit handling of similar words due to the averaging of dense word vectors for semantically related words. Thus, we recommend using the dense approach in further studies. Although the AdaGram approach trained on a large text corpus showed better results according to the weighted average, this result does not transfer to languages with less available corpus size.

Table 1: Results on RUSSE'2018 (Adjusted Rand Index).

Method		**bts-rnc**	**wiki-wiki**	**Avg.**
AdaGram		0.22	0.39	0.23
WATLINK	*Sparse*	0.01	0.07	0.01
	Dense	0.08	0.14	0.08
RuThes	*Sparse*	0.00	0.17	0.01
	Dense	0.14	0.47	0.17
RuWordNet	*Sparse*	0.00	0.11	0.01
	Dense	0.12	0.50	0.15
One		0.00	0.00	0.00
Singletons		0.00	0.00	0.00

5. Conclusion

In this paper, we presented Watasense,[7] an open source unsupervised word sense disambiguation system that is parameterized only by a word sense inventory. It supports both sparse and dense sense representations. We were able to show that the dense approach substantially boosts the performance of the sparse approach on three different sense inventories for Russian. We recommend using the dense approach in further studies due to its smoothing capabilities that reduce sparseness. In further studies, we will look at the problem of phrase neighbors that influence the sentence vector representations.

Finally, we would like to emphasize the fact that Watasense has a simple API for integrating different algorithms for WSD. At the same time, it requires only a basic set of language processing tools to be available: tokenizer, a part-of-speech tagger, lemmatizer, and a sense inventory, which means that low-resourced language can benefit of its usage.

6. Acknowledgements

We acknowledge the support of the Deutsche Forschungsgemeinschaft (DFG) under the project "Joining Ontologies and Semantics Induced from Text" (JOIN-T), the RFBR under the projects no. 16-37-00203 mol_a and no. 16-37-00354 mol_a, and the RFH under the project no. 16-04-12019. The research was supported by the Ministry of Education and Science of the Russian Federation Agreement no. 02.A03.21.0006. The calculations were carried out using the supercomputer "Uran" at the Krasovskii Institute of Mathematics and Mechanics.

7. Bibliographical References

Agirre, E., de Lacalle, O. L., Fellbaum, C., Hsieh, S.-K., Tesconi, M., Monachini, M., Vossen, P., and Segers, R. (2010). SemEval-2010 Task 17: All-words Word Sense Disambiguation on a Specific Domain. In *Proceedings*

[3] http://russe.nlpub.org/2018/wsi/

[4] http://www.labinform.ru/pub/ruthes/index_eng.htm

[5] http://www.labinform.ru/pub/ruwordnet/index_eng.htm

[6] https://doi.org/10.5281/zenodo.400631

[7] https://github.com/nlpub/watasense

of the 5th International Workshop on Semantic Evaluation, SemEval '10, pages 75–80, Los Angeles, CA, USA. Association for Computational Linguistics.

Bartunov, S., Kondrashkin, D., Osokin, A., and Vetrov, D. P. (2016). Breaking Sticks and Ambiguities with Adaptive Skip-gram. *Journal of Machine Learning Research*, 51:130–138.

Biemann, C. and Riedl, M. (2013). Text: now in 2D! A framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1):55–95.

Faralli, S., Panchenko, A., Biemann, C., and Ponzetto, S. P. (2016). Linked Disambiguated Distributional Semantic Networks. In *The Semantic Web – ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II*, pages 56–64, Cham, Germany. Springer International Publishing.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.

Lopukhin, K. A. and Lopukhina, A. A. (2016). Word Sense Disambiguation for Russian Verbs Using Semantic Vectors and Dictionary Entries. In *Computational Linguistics and Intellectual Technologies: Papers from the Annual conference "Dialogue"*, pages 393–404, Moscow, Russia. RSUH.

Lopukhin, K. A., Iomdin, B. L., and Lopukhina, A. A. (2017). Word Sense Induction for Russian: Deep Study and Comparison with Dictionaries. In *Computational Linguistics and Intellectual Technologies: Papers from the Annual conference "Dialogue". Volume 1 of 2. Computational Linguistics: Practical Applications*, pages 121–134, Moscow, Russia. RSUH.

Loukachevitch, N. V., Lashevich, G., Gerasimova, A. A., Ivanov, V. V., and Dobrov, B. V. (2016). Creating Russian WordNet by Conversion. In *Computational Linguistics and Intellectual Technologies: papers from the Annual conference "Dialogue"*, pages 405–415, Moscow, Russia. RSUH.

Loukachevitch, N. V. (2011). *Thesauri in information retrieval tasks*. Moscow University Press, Moscow, Russia. In Russian.

Manandhar, S., Klapaftis, I., Dligach, D., and Pradhan, S. (2010). SemEval-2010 Task 14: Word Sense Induction & Disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 63–68, Uppsala, Sweden. Association for Computational Linguistics.

Miller, T., Erbs, N., Zorn, H.-P., Zesch, T., and Gurevych, I. (2013). DKPro WSD: A Generalized UIMA-based Framework for Word Sense Disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Sofia, Bulgaria.

Moro, A., Raganato, A., and Navigli, R. (2014). Entity Linking meets Word Sense Disambiguation: A Unified Approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Navigli, R., Litkowski, K. C., and Hargraves, O. (2007). SemEval-2007 Task 07: Coarse-Grained English All-Words Task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 30–35, Prague, Czech Republic. Association for Computational Linguistics.

Panchenko, A., Ustalov, D., Arefyev, N., Paperno, D., Konstantinova, N., Loukachevitch, N., and Biemann, C., (2017a). *Human and Machine Judgements for Russian Semantic Relatedness*, pages 221–235. Springer International Publishing, Cham, Germany.

Panchenko, A., Marten, F., Ruppert, E., Faralli, S., Ustalov, D., Ponzetto, S. P., and Biemann, C. (2017b). Unsupervised, Knowledge-Free, and Interpretable Word Sense Disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 91–96, Copenhagen, Denmark. Association for Computational Linguistics.

Panchenko, A., Lopukhina, A., Ustalov, D., Lopukhin, K., Leontyev, A., Arefyev, N., and Loukachevitch, N. (2018). RUSSE'2018: A Shared Task on Word Sense Induction and Disambiguation for the Russian Language. In *Computational Linguistics and Intellectual Technologies: Papers from the Annual conference "Dialogue"*, Moscow, Russia. RSUH.

Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pelevina, M., Arefiev, N., Biemann, C., and Panchenko, A. (2016). Making Sense of Word Embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 174–183, Berlin, Germany. Association for Computational Linguistics.

Ustalov, D., Panchenko, A., and Biemann, C. (2017). Watset: Automatic Induction of Synsets from a Graph of Synonyms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1579–1590, Vancouver, Canada. Association for Computational Linguistics.

Ustalov, D. (2017). Expanding Hierarchical Contexts for Constructing a Semantic Word Network. In *Computational Linguistics and Intellectual Technologies: Papers from the Annual conference "Dialogue". Volume 1 of 2. Computational Linguistics: Practical Applications*, pages 369–381, Moscow, Russia. RSUH.

Řehuřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *New Challenges for NLP Frameworks Programme: A workshop at LREC 2010*, pages 51–55, Valetta, Malta. European Language Resources Association (ELRA).

Zhong, Z. and Ng, H. T. (2010). It Makes Sense: A Wide-Coverage Word Sense Disambiguation System for Free Text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83, Uppsala, Sweden. Association for Computational Linguistics.