Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2018"

Moscow, May 30—June 2, 2018

RUSSE'2018: WORD SENSE INDUCTION AND DISAMBIGUATION METHOD BASED ON CONTEXT-BASED LISTS

Sreya Mittal (sreya.mittal@students.iiit.ac.in),
Pratibha Rani (pratibha_rani@research.iiit.ac.in)

Data Sciences and Analytics Center, Kohli Center on Intelligent Systems, International Institute of Information Technology, Hyderabad, India

This paper reports the participation of IIITHDSAC team in the shared task on word sense induction and disambiguation (WSID) for the Russian language in RUSSE'2018. The method adopted is semi-supervised and knowledge-free which does not use any knowledge resource like dictionary or Wiki. It only uses the sense tagged and untagged data provided by the task organizers as training data and builds the WSID model using the concept of context-based lists from words of training data converted into root form. Context-based lists enables to cluster words and senses based on contexts and hence, provides a way to use context of an unseen target word to find its sense even if it is absent in the training data. We have used the root form of training set words because the test set words were given in root form otherwise our method is generic and would work for normal form of words also.

Keywords: word sense induction, word sense disambiguation, context-based lists, semi-supervised method, sense-tagged corpus, Knowledge-free method

1. Introduction

Word sense induction (WSI) is a Natural language processing (NLP) problem which concerns with the automatic identification of the senses of a word. Word sense disambiguation (WSD) is the problem of identifying which sense of a word is used in a sentence, when the word has multiple meanings associated with it. In this paper we present a word sense induction and disambiguation (WSID) method which is application and extension of a generic domain independent WSD approach proposed by authors of [2] evaluated in the framework of the RUSSE'18 WSID shared task challenge [1][1].

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

The WSD method proposed in [2] extracts *context based lists* (CBL) from a small sense-tagged and untagged training data without using domain knowledge and uses these lists to predict the senses of test words. Hence, this is a knowledge-free method for the given problem. On the contrary, knowledge-rich methods use dictionaries and other domain knowledge/resources to arrive at the result.

In our method first we convert all the training data words into their root form using a morphological analyzer[2] and then find CBLs from this converted training data. Then we use these CBLs to cluster senses based on contexts which gives a WSID model in which we can use context of a target word to find its sense even if it is absent in the training data. So this model will work for WSD task of both seen and new unseen target words and using the association between context, sense and target words it can also perform the word sense induction (WSI) task of new unseen target words.

Please note that we have build our model using root form of training set words because the test set words were given in root form otherwise our method is generic and would work for normal form of words also.

Rest of this paper is organized as follows: Section 2 describes some of the previous works related to the WSD/WSID task. Section 3 outlines the evaluation methodology. Section 4 presents a brief discussion of the proposed algorithm, followed by the key results in Section 5 and conclusions in Section 6.

2. Related Work

Knowledge based WSD techniques [3] use knowledge structures like, WordNet [5] [6] or machine-readable dictionaries [7] to build WSD models in which words and their senses are directly associated.

Supervised WSD approaches need domain expertise for creating and selecting features for the algorithms in which machine learning [12] and statistical methods [8] are applied on manually created sense-tagged training corpus. Domain expertise is also required in the form of rules and details for preprocessing and transforming the training data into the form required for designing the algorithms [3] [8].

Unsupervised WSID methods require large amount of raw untagged training corpus [13] [14] to find word clusters which discriminate the senses of the words in different clusters. These methods also use multilingual parallel corpora [9] [20], a knowledge resource like WordNet [18] [19] or multilingual dictionary [10] to build the WSD models.

Semi-supervised WSID approaches use both sense-tagged and untagged data in different proportions with different methods like, co-training with multilingual parallel corpora [15], bootstrapping [11], neural network [16] [17] and word sense induction [23].

Analyzing the existing WSD and WSID techniques we find that all the WSD techniques require domain knowledge in some or other form for designing the algorithms along with sense-tagged data and knowledge resources and can't work with low amount of data while unsupervised WSID techniques require raw untagged data in huge amounts and need domain expertise to convert the results which come in the form of cluster of words and contexts to valid sense IDs.

[2] https://pypi.python.org/pypi/pymystem3/0.1.1

The WSD method proposed in [2] is a semi-supervised method which utilizes the contextual similarity property based on one sense per collocation hypothesis [21] and works with low amount of data without requiring domain knowledge or knowledge resources except a moderate/small size sense-tagged corpus. Our proposed WSID method builds upon the method proposed in [2] and utilizes their advantages available for low resource languages to convert it into a WSID method.

3. Evaluation

The evaluation measure used by the RUSSE'18 WSID shared task organizers to rank the submissions of the teams was Adjusted Rand Index[3] (ARI). The Rand Index (RI) (from sklearn[4]) computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings.

The raw RI score is then "adjusted for chance" into the ARI score using the following scheme:

$$\text{ARI} = (\text{RI} - \text{Expected\_RI}) \,/\, (\max(\text{RI}) - \text{Expected\_RI})$$

Detail of datasets provided by the task organizers is presented in **Table 1**.

Table 1: RUSSE'2018 WSID Shared Task Dataset Details

Dataset	Inventory	Corpus	Split	Num. of words	Num. of senses	Avg. num. of senses	Num. of contexts
wiki-wiki	Wikipedia	Wikipedia	train	4	8	2.0	439
wiki-wiki	Wikipedia	Wikipedia	test	7			638
active-dict	Active Dict.	Active Dict.	train	85	312	3.7	2,073
active-dict	Active Dict.	Active Dict.	test	168			3,729
bts-rnc	Gramota.ru	RNC	train	30	96	3.2	3,491
bts-rnc	Gramota.ru	RNC	test	51			6,556

It should be noted that test set words of the provided data was in "root" form while the training set words were in normal form. Hence, we have used a morphological analyzer to convert all the words of the training set to their root form.

4. Method

The method which we have used to get the results for the WSID task of RUSSE'2018 is built upon the one proposed in [2] for word sense disambiguation (WSD). Their method uses the concept of *context based list* (CBL) proposed by the authors of [4]

[3] http://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html

[4] http://scikit-learn.org/

for POS tagging task. The authors of [4] utilize the contextual similarity property based on one sense per collocation hypothesis [21] and call the list of words occurring in a particular context as CBL and use association rule mining [22] for obtaining effective context based POS tagging rules from the set of tagged and raw untagged training data. The authors of [2] extend their idea by supplementing CBL with the concepts of *extended context list*, *context based sense list* and *context based word list* (please refer to paper [2] for details). Their method handles the data sparsity related problems of WSD task. Their method uses raw untagged data and concept of *extended context list* to handle the issue of non-availability of matching contexts for a word in the given sense-tagged training data. They use raw untagged data and CBLs to deal with the problem of absence of target words in the sense-tagged training set. They also define proper threshold parameters and use CBLs to handle the large imbalance in frequencies of senses associated with the words of training set.

To make use of the above mentioned properties available in the WSD method of [2] for WSID task we have extended the method by adding *context based word sense induction* step and converted the algorithm into WSID algorithm. Also, to be able to use the training data provided by the task organizers we have added the step of converting all the words of the training and test sets to their root forms using a morphological analyzer[5].

To be able to understand our WSID algorithm we need to revisit following terminologies defined by the authors of [2] (please refer to paper [2] for detailed explanations): firstly, they define notion of **context** as a word pair and use the left and right immediate neighboring words of a word/sense ID in a sentence/phrase as its context.

Then **Context Based Word List** is defined as the list of word instances from a text collection sharing the same context. Same way **Context Based Sense List** is defined as a list of sense ID instances from a sense-tagged text collection sharing the same context. They also define **Single Sense Word List** as a list of word instances having only one sense ID associated them in the sense-tagged training data. The count ***TotalSenseSupport*** gives the total count of a particular sense ID in the sense-tagged training set.

The defined **Extended Context List** is used to find contexts similar to a given context (W_l, W_r). A count value **ExtContextCount** is associated with each context present in the *extended context list* which shows how frequently it can be obtained from the available CBLs. For a given context, (W_l, W_r) of a word W_m, if *PreListSet* is the set of words obtained from those *context based word lists* which have left context W_l in their word list, *PostListSet* is the set of words obtained from those *context based word lists* which have right context W_r in their word list and *extended context list* is the set of all contexts (W_{pre}, W_{post}) obtained by taking W_{pre} from *PreListSet* and W_{post} from *PostListSet*. Count value *ExtContextCount* of a context will show how many word combinations from *PreListSet* and *PostListSet* generated that context (see paper [2] for details).

Along with these terms and concepts there are also some simple parameters and threshold values defined in paper [2] which are required for the algorithm. They can be easily understood by going through the paper. Our proposed WSID method follows steps given in **Table 2** to find suitable sense IDs for the test words:

[5] https://pypi.python.org/pypi/pymystem3/0.1.1

Table 2: WSID Method using CBL

1. **Training Phase**:

1.0 Convert all words to their root form using the morphological analyzer.

1.1 Using a sliding window of size three, collect all the *context based word lists*, *context based sense lists*, *single sense word list*, word and sense counts from the sense-tagged and raw untagged text collection in a single iteration, taking care of the sentence boundaries.

2. **Testing Phase**:

2.0 Convert all words to their root form using the morphological analyzer.

WSD_Present_test_word:

2.1 If test word $\mathbf{W_t}$ is present in *single sense word list* then directly output the associated sense ID.

2.2 If test word $\mathbf{W_t}$ is present in sense-tagged text collection then find its context $\mathbf{(W_{tl}, W_{tr})}$ from test sentence and apply **Algo 1** to find and return the best sense ID.

WSI_Absent_test_word:

2.3 If test word $\mathbf{W_t}$ is not present in sense tagged text collection then find its context $\mathbf{(W_{tl}, W_{tr})}$ from test sentence and apply **Algo 2** to find a list **ProbSenList** of all probable senses for $\mathbf{W_t}$.

WSD_Absent_test_word:

2.4 If the list **ProbSenList** of probable senses for $\mathbf{W_t}$ does not contain ***NOEXISTSEN*** then return the sense ID with highest ***TotalSenseSupport*** value.

Algo 1:

If test word $\mathbf{W_t}$ is present with its context $\mathbf{(W_{tl}, W_{tr})}$ as trigram $\mathbf{(W_{tl}, W_t, W_{tr})}$ in sense-tagged text collection then find the corresponding sense IDs of $\mathbf{W_t}$ from *context based sense list* of $\mathbf{(W_{tl}, W_{tr})}$ and return the sense ID with highest $\mathbf{W_t}$ count.

If test word $\mathbf{W_t}$ is present without its context $\mathbf{(W_{tl}, W_{tr})}$ in sense-tagged data then using *context based word lists* obtained from sense-tagged data find *extended context list* of contexts similar to $\mathbf{(W_{tl}, W_{tr})}$ in which $\mathbf{W_t}$ is present. From this list select the context with highest ***ExtContextCount*** value. For this context, using *context based sense lists* find the associated sense IDs satisfying all the defined parameters and thresholds and return the one with highest highest $\mathbf{W_t}$ count.

If no output is found in step 2 and if *context based word list* of $\mathbf{(W_{tl}, W_{tr})}$ from raw untagged data contains test word $\mathbf{W_t}$ then, using *context based sense lists* obtained from sense-tagged data find the sense IDs associated with $\mathbf{(W_{tl}, W_{tr})}$ satisfying all the defined parameters and thresholds and return the one with highest $\mathbf{W_t}$ count.

If no output is found in step 3 then return the sense ID having highest $\mathbf{W_t}$ count.

Algo 2:

Using *context based word lists* obtained from sense-tagged data find *extended context list* of contexts similar to (W_{tl}, W_{tr}) and from this select the context with highest ***ExtContextCount*** value. For this context find *context based word list* from sense-tagged data and using *context based sense lists* find the associated sense IDs satisfying all the defined parameters and thresholds and return the list of sense IDs.

If step 1 does not produce any output then find *context based word list* of (W_{tl}, W_{tr}) from raw untagged data in which test word W_t is present and *context based word list* of (W_{tl}, W_{tr}) from sense-tagged data. If these two lists satisfy some word matching criteria, some parameters and thresholds then using *context based sense lists* find the associated sense IDs satisfying all the defined parameters and thresholds and return the list of sense IDs.

If step 2 does not produce any output then using *context based word lists* obtained from raw untagged data find *extended context list* of contexts similar to (W_{tl}, W_{tr}) and from this select the context with highest ***ExtContextCount*** value. For this context find *context based word list* from sense-tagged data and using *context based sense lists* find the associated sense IDs satisfying all the defined parameters and thresholds and return the list of sense IDs.

If step 3 does not produce any output then return ***NOEXISTSEN***.

In training phase of our proposed WSID method all the counts and CBLs from sense-tagged and raw untagged data are computed and collected for performing the WSD and WSI tasks. Then in testing phase, if the test word is present in sense-tagged data then step 2.1 uses *single sense word list* and step 2.2 uses **Algo 1** to perform the WSD task. **Algo 1** gives highest priority to the immediate context of test word present with the test word in sense-tagged data and applies the concept of *extended context list* to find indirectly available similar contexts from sense-tagged data when the actual context is not present with test word in the sense tagged data and lastly uses *context based sense list* of test context only if *context based word list* of the test context obtained from raw untagged data contains the test word. It uses the defined count values and thresholds to select the best possible sense ID (refer to paper [2] for detailed pseudocode and explanation).

If the test word is not present in sense-tagged data, then in testing phase step 2.3 uses **Algo 2** to perform the WSI task and returns a list of possible sense IDs for the unseen test word. **Algo 2** also uses the concept of *extended context list* to find the indirectly available similar contexts from the sense tagged and raw untagged data in priority order to use them for finding the probable sense IDs. Using this concept and raw untagged data it is able to find senses for new unseen target words also if target word's context information is available. The defined support and threshold parameters help it in selecting sense IDs with confidence. If we need to perform WSD task for the test word not present in sense-tagged data then step 2.4 can select the best possible sense ID for this test word based on count value from the list of senses found in step 2.4. As this method does not use any knowledge resource or domain knowledge so, it is a generic domain independent knowledge-free method.

The task organizers mention that there will be some target test words which would be different from the training datasets. As explained above, our method is able

to deal with unseen new test words by using the concept of *extended context list* to find the sense IDs. Steps 1 and 3 of **Algo 2** allows to find sense IDs of those test words also which are not present in training datasets and step 2.3 performs WSI task for such words and step WSD task for them.

5. Result

In this section we discuss evaluation results of our approach. Table 3 presents the ARI score obtained by our submissions for the three public and the three private datasets.

Table 3: Results of CBL method for RUSSE'2018 WSI/D Shared Task Datasets

Dataset	Our Method		Rank 1	
	Training data size	ARI Score	Training data size	ARI Score
Wiki-wiki (public)	365 kB	0.0249	365 kB + x GB	1.0000
Bts-rnc (public)	1.2 MB	0.0123	1.2 MB + x GB	0.3508
active-dict (public)	351 kB	0.0271	351 kB + x GB	0.2643
Wiki-wiki (private)	—	0.1287	—	0.9625
Bts-rnc (private)	—	0.0268	—	0.3384
active-dict (private)	—	0.0166	—	0.2477

Our method gives ARI scores lower than the rank 1 submission and many other submissions but the point to be noted is that their method uses pretrained sense corpora to classify sentences using the sense clusters, while our method has been built on minimal data, that is, just the data provided by the organizers of the shared task. Hence, our method can be used for resource-poor languages as it can work on small dataset. The fact that the training data size matters for resource-poor languages makes our method more valuable. One more point to be noted is that our method is very fast since it deals with a very small amount of data.

For the given datasets as first of our WSID algorithm we used the Python wrapper of the Yandex Mystem 3.0 morphological analyzer—pymystem3[6] to convert all the words of training data (both sense tagged and untagged) into their "root" form and then we applied the remaining steps of the algorithm described in Table 2 on the converted data to get the results.

From the dataset details given in Table 1 it is clear that the size of training data (sense-tagged and untagged both) is very very small and so for all the existing WSD methods it would be very difficult to obtain a WSD model without using supporting domain knowledge resources and additional data. But our approach is able to obtain a working WSD model using only the available small training data. Our results can always be improved by using more raw untagged data.

[6] https://pypi.python.org/pypi/pymystem3/0.1.1

6. Conclusion

Our approach, despite being knowledge-free and without using any domain knowledge based resources, gives acceptable ARI scores on all three datasets. Our approach is special and very different from other knowledge-free methods because it is able to obtain a working WSD model using only the available small sense-tagged and untagged training data without using any additional resources. So our approach is especially suitable for resource-poor languages and can also be used as initial WSD/WSI tool to help the language annotators by suggesting them probable senses of words to be annotated.

References

1. *Alexander Panchenko, Anastasiya Lopukhina, Dmitry Ustalov, Konstantin Lopukhin, Aleksey Leontyev, Nikolay Arefyev, Natalia Loukachevitch* (2018), RUSSE'2018: A Shared Task on Word Sense Induction and Disambiguation for the Russian Language, In Proceedings of the 24rd International Conference on Computational Linguistics and Intellectual Technologies (Dialogue'2018), May 30—June 2, Moscow, Russia.
2. *Pratibha Rani, Vikram Pudi, Dipti Misra Sharma* (2017), Semisupervised Data Driven Word Sense Disambiguation for Resource-poor Languages, In proceedings of the 14th International Conference on Natural Language Processing (ICON 2017), December, pp. 503-512, https://ltrc.iiit.ac.in/icon2017/proceedings/icon2017/pdf/W17-7561.pdf
3. *Roberto Navigli* (2009), Word Sense Disambiguation: A Survey, ACM Computing Survey, 41(2):10:110:69.
4. *Pratibha Rani, Vikram Pudi, and Dipti Misra Sharma* (2016), A semi-supervised associative classification method for POS tagging, International Journal of Data Science and Analytics, Vol. 1(2), pp. 123–136.
5. *Christiane Fellbaum, editor* (1998), WordNet: An Electronic Lexical Database, MIT Press.
6. *Satanjeev Banerjee and Ted Pedersen* (2002), An adapted Lesk algorithm for word sense disambiguation using WordNet, In International Conference on Intelligent Text Processing and Computational Linguistics, pp. 136–145. Springer.
7. *Michael Lesk* (1986), Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone, In Proceedings of the 5th annual international conference on Systems documentation, pp. 24–26, ACM.
8. *Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli* (2016), Embeddings for Word Sense Disambiguation: An Evaluation Study, In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, Vol. 1.
9. *Sudha Bhingardive, Samiulla Shaikh, and Pushpak Bhattacharyya* (2013), Neighbors Help: Bilingual Unsupervised WSD Using Context, In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL, Vol. 2, Short Papers, pp. 538–542.

10. *Mitesh M. Khapra, Salil Joshi, and Pushpak Bhattacharyya* (2011), It Takes Two to Tango: A Bilingual Unsupervised Approach for Estimating Sense Distributions using Expectation Maximization, In Proceedings of 5th International Joint Conference on Natural Language Processing, pp. 695–704.
11. *David Yarowsky* (1995), Unsupervised Word Sense Disambiguation Rivaling Supervised Methods, In ACL, pp. 189–196.
12. *Mikael Kågebäck and Hans Salomonsson* (2016), Word sense disambiguation using a bidirectional LSTM, CoRR, abs/1606.03568.
13. *Ted Pedersen and Rebecca Bruce* (1997), Distinguishing Word Senses in Untagged Text, In eprint arXiv: cmp-lg/9706008.
14. *Dekang Lin* (1998), Automatic retrieval and clustering of similar words, In Proceedings of the 17th international conference on Computational linguistics, Vol. 2, pp. 768–774.
15. *Mo Yu, Shu Wang, Conghui Zhu, and Tiejun Zhao* (2011), Semi-supervised learning for word sense disambiguation using parallel corpora, In 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Vol. 3, pp. 1490–1494, IEEE.
16. *Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf* (2016), Semi-supervised word sense disambiguation with neural models, arXiv preprint arXiv:1603.07012.
17. *Kaveh Taghipour and Hwee Tou Ng* (2015), Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains, In HLT-NAACL, pp. 314–323.
18. *Ping Chen, Wei Ding, Chris Bowes, and David Brown* (2009), A Fully Unsupervised Word Sense Disambiguation Method Using Dependency Knowledge, In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 28–36.
19. *Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen* (2007), UMND1: Unsupervised Word Sense Disambiguation Using Contextual Semantic Relatedness, In Proceedings of the 4th International Workshop on Semantic Evaluations, pp. 390–393.
20. *Nancy Ide, Tomaz Erjavec, and Dan Tufis* (2002), Sense Discrimination with Parallel Corpora, In Proceedings of the ACL-02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions, Vol. 8, pp. 61–66.
21. *David Yarowsky* (1993), One sense per collocation, In Proceedings of the workshop on Human Language Technology, pp. 266–271.
22. *Rakesh Agrawal, Tomasz Imieli´nski, and Arun Swami* (1993), Mining Association Rules Between Sets of Items in Large Databases, In SIGMOD'93, pp. 207–216.
23. *Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov* (2015), Breaking sticks and ambiguities with adaptive skip-gram, arXiv preprint arXiv:1502.07257.
24. *Osman Baskaya and David Jurgens* (2016), Semi-supervised Learning with Induced Word Senses for State of the Art Word Sense Disambiguation, Journal of Artificial Intelligence Research, Vol. 55(1), pp. 1025–1058.