Natural Language Processing (2024), 1–24
doi:10.1017/nlp.2024.31

ARTICLE

Word sense disambiguation corpus for Kashmiri

Tawseef Ahmad Mir[1] and Aadil Ahmad Lawaye[2]

[1] Alliance School of Advanced Computing, Alliance University, Bangalore, India and [2] Department of Computer Science, Baba Ghulam Shah Badshah University, Rajouri, India
Corresponding author: Tawseef Ahmad Mir; Email: tawseefmir1191@gmail.com

(Received 25 December 2022; revised 11 December 2023; accepted 13 February 2024)

Special Issue on '**Natural Language Processing Applications for Low-Resource Languages**'

Abstract

Ambiguity is considered an indispensable attribute of all natural languages. The process of associating the precise interpretation to an ambiguous word taking into consideration the context in which it occurs is known as word sense disambiguation (WSD). Supervised approaches to WSD are showing better performance in contrast to their counterparts. These approaches, however, require sense annotated corpus to carry out the disambiguation process. This paper presents the first-ever standard WSD dataset for the Kashmiri language. The raw corpus used to develop the sense annotated dataset is collected from different resources and contains about 1 M tokens. The sense-annotated corpus is then created using this raw corpus for 124 commonly used ambiguous Kashmiri words. Kashmiri WordNet, an important lexical resource for the Kashmiri language, is used for obtaining the senses used in the annotation process. The developed sense-tagged corpus is multifarious in nature and has 19,854 sentences. Based on this annotated corpus, the Lexical Sample WSD task for Kashmiri is carried out using different machine-learning algorithms (J48, IBk, Naive Bayes, Dl4jMlpClassifier, SVM). To train these models for the WSD task, bag-of-words (BoW) and word embeddings obtained using the Word2Vec model are used. We used different standard measures, viz. accuracy, precision, recall, and F1-measure, to calculate the performance of these algorithms. Different machine learning algorithms reported different values for these measures on using different features. In the case of BoW model, SVM reported better results than other algorithms used, whereas Dl4jMlpClassifier performed better with word embeddings.

Keywords: information extraction; machine learning; sense annotation; word sense disambiguation

1. Introduction

Word sense disambiguation (WSD), an exigent task in natural language processing (NLP), means assigning definite sense to an ambiguous word in the given text (Zhang *et al.* 2023; Haouassi *et al.* 2023). Understanding the meaning of words is very easy for humans when they communicate; however, machines find it very difficult to interpret natural language data. Understanding the natural languages by machines makes it necessary to solve the WSD problem. WSD is not a new research problem in NLP but has a long history. It has been classified as an AI-Complete problem (Navigli 2009) in NLP. Different NLP tasks like Machine Translation (Wang *et al.* 2023), Information Retrieval (Agirre and Edmonds 2007), Information Extraction (Abderrahim and Abderrahim 2022), Question Answering (Rahman and Borah 2022), Sentiment Analysis (Kharate and Patil 2021), Text Summarization (Sert *et al.* 2023), Discourse Analysis (Saeed *et al.* 2019) are dependent on it for better performance. WSD problem when catered carefully, the performance of these NLP tasks gets boosted. However, the lack of high-performance WSD systems hinders the creation of effective NLP systems for these NLP tasks.

Although a number of researchers have put their efforts into solving this problem in the last many decades, these efforts cover a few languages like English, Catalan, Romanian, Italian, Spanish, Chinese, Swedish, Japanese, and Korean. Little attention was given to other languages, in particular to South Asian languages (Rouhizadeh *et al.* 2022; Saeed *et al.* 2019). There is still a vacuum to work out the WSD problem for low-resource languages like Kashmiri.

WSD task in the research perspective is investigated into two forms; Lexical Sample WSD task and All-Words WSD task. A set of pre-selected ambiguous words are disambiguated in the Lexical Sample WSD task whereas the focus of the All-Words WSD task is to decipher all ambiguous words existing in the given input text. The Lexical Sample version of the WSD task has been explored more often in comparison to the All-Words WSD task by researchers as the All-Words WSD task is more challenging (Rouhizadeh *et al.* 2022). Kashmiri language, however, lacks research on both versions of the WSD task due to the unavailability of linguistic resources. The main objective of this research work is to develop the first standard sense-tagged dataset for the Lexical Sample WSD task which is a prerequisite to use data-driven approaches to solve the WSD problem.

Different categories of approaches developed so far to solve the WSD task are (1) Knowledge-based approaches (2) Corpus-based approaches and (3) Hybrid approaches (Nodehi and Charkari 2022). Knowledge-based WSD approaches use the clues available in machine-readable dictionaries etc. to learn the disambiguation process. Corpus-based approaches need a large dataset to infer the exact sense of a dubious word based on context. Corpus-based Approaches are of two types (a) Supervised and (b) Unsupervised approaches.

Supervised WSD approach means learning the WSD task from a sense-annotated corpus; unsupervised approaches on the other hand learn the disambiguation task using untagged corpora. Hybrid approaches to WSD take advantage of multiple resources like Wordnet (Miller 1995) and tagged corpora. When the performance of these different approaches is analyzed, it is observed that supervised approaches are superior (Mihalcea 2007; Park *et al.* 2022). The use of supervised WSD approaches for languages like English, French, Japanese, Dutch, etc. has shown steady progress. On the other hand, WSD research has lagged behind for low-resource languages due to the unavailability of sense annotated datasets. In this research work, we introduce, (Kmri-WSD-Dataset[a]), a novel WSD dataset for the Kashmiri language. The dataset so developed has a total of 19854 manually sense annotated instances for 124 ambiguous Kashmiri words.

Kashmiri, classified as an Indo-Aryan language, is one of the official languages as per the Indian constitution. Mainly spoken by residents of Jammu and Kashmir and Kashmiri diaspora in various parts of the world, Kashmiri is known for its rich literary heritage. Key characteristics of the Kashmiri language include its complex phonological system, great inflection in grammar, rich vocabulary, and dialect diversity (Koul and Wai 2015). It is a highly inflected language with verb-second (V2) word order. From the computational resources perspective, Kashmiri is considered a low-resource language in comparison to other Indian languages like Hindi, Punjabi, Telugu, etc. It has not been explored much by researchers in the NLP domain. Some of the important research works to develop computational resources for the Kashmiri language that are useful in NLP tasks are discussed here. Gold standard raw corpus for Kashmiri has been developed at the Central Institute of Indian Languages (Ramamoorthy *et al.* 2019). The corpus has 466,054 words obtained from different books, magazines and newspapers. The corpus has text from two domains Aesthetics (85.93%) and Social Sciences (14.7%). A raw corpus of 2.3 million words was also created as a part of the project EMILLE supported by the universities of Lancaster and Baker *et al.* (2022). The purpose of the project was to develop a 67-million-word corpus for different Indic

[a] https://github.com/Tawseef-Mir/Kmri_WSD_Dataset

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

languages. In Mehdi and Kak (2018) paradigm based approach is capitalized to develop a morphological analyzer for Kashmiri. The authors first identified Part-of-Speech (PoS) of the lexical terms and then created different paradigms for different PoS categories for morphological analysis. Kak *et al.* (2017) presents the challenges and issues related to the development of Kashmiri Wordnet. Lawaye and Purkayastha (2014) proposed an automatic PoS tagger for Kashmiri using Conditional Random Field (CRF). In Mehdi and Lawaye (2011) research work, the first Unicode font for Kashmiri called "AFAN Koshur Naksh" is designed. Kak *et al.* (2009) presented a tagset consisting of 26 tags for Kashmiri PoS tagging. Banday *et al.* (2009) presented a trilingual (English-Hindi-Kashmiri) sense-based e-dictionary. The dictionary contains different senses of a word based on its environment, hence can be used for resolving the sense ambiguity. Recently Research Center for Technical Development of Punjabi language, Punjabi University, Patiala, has decided to develop linguistic tools such as corpus, storage code converters, transliteration tools, typing tools, digital dictionaries, word predictors for four languages namely; Kashmiri, Pashto, Balochi and Dari. To the best of our knowledge, no standard Kashmiri sense-annotated corpus is available freely for research purposes. This is the primary motivation behind the research work presented in this paper. This work is actually an extension of research work Mir *et al.* (2023) to create a sense-tagged corpus for the development of a supervised WSD system for the Kashmiri language.

Key contributions of the paper are:

- **Development of maiden standard corpus for Kashmiri Lexical Sample WSD task:** This is the first standard corpus for the Kashmiri Lexical Sample WSD task that we are aware of. The advancement of Kashmiri language study in the NLP domain would be aided by the inclusion of this dataset. It will also assist scholars in developing comparable resources for other languages with limited resources.
- **Setting criteria for future Kashmiri Lexical Sample WSD research:** A collection of machine learning WSD algorithms is run on the developed corpus to give a baseline for evaluating future Kashmiri Lexical Sample WSD systems. The analysis of the results produced by various algorithms run on the dataset is also provided.
- **Utility of the developed corpus in assessing other Kashmiri NLP tasks:** The developed corpus is tokenized, POS-labeled, and sense-tagged. As a result, it may be used as a standard for preprocessing tools such as tokenizers and PoS taggers.

The paper is delineated into different parts: Section 2 discusses the important sense-tagged datasets developed for different languages; Section 3 is about the procedure followed to develop the sense-tagged corpus; Section 4 discusses the methods used to explore WSD task in Kashmiri; Section 5 presents the results and error analysis of the different methods used to carry out the WSD task and the Conclusion is presented in Section 6.

2. Related work

Sense-tagged datasets are commonly used in NLP tasks, particularly in WSD and related tasks like semantic role labeling and machine translation. These datasets are labeled with information about the specific sense or meaning of words in context. Research for developing sense-annotated corpora started a long time ago and a number of sense-annotated corpora have been developed for various natural languages. They are often used for training and evaluating WSD systems. The annotated corpora have been developed for handling both the Lexical Sample WSD task as well as the All-Words WSD task. Important WSD datasets developed for foreign and Indian languages are discussed below.

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

2.1 WSD datasets for foreign languages

English being the most commonly used language worldwide has been the hotspot for NLP research. Different WSD datasets have been developed till now for English with different sizes and characteristics. SemCor (Miller *et al.* 1994) is the largest available manually curated English sense-tagged dataset aligned with WordNet senses. It has about 234,000 tokens attached with syntactic category and appropriate sense. It has been extensively used by researchers for different NLP tasks. Some important research works in which SemCor is used include Song *et al.* (2021); Luan *et al.* (2020); Nodehi and Charkari (2022). DSO Corus (Ng and Lee 1996) is another WSD corpus with 192800 instances for a set of 191 target words among which 121 are nouns and 72 are verbs. This corpus has also been employed by different researchers for evaluating WSD systems. Patankar and Devane (2017) utilized the DSO corpus to analyze the WSD problem in depth and its role to resolve the multilingual translation problem. Itankar and Raza (2020) also utilized the DSO corpus for analyzing the role of WSD in machine translation by carrying out experiments on fifteen ambiguous words. MASC (Ide *et al.* 2010), a multilayered annotated dataset, stands for Manually Annotated Sub-Corpus. The corpus has 500000 lexical items with named entity annotations, part-of-speech tags, syntactic structure, etc. making it applicable in multiple NLP tasks like WSD, syntactic parsing, named entity recognition, etc. Wijeratne *et al.* (2017) utilized MASC for WSD task for emoji sense discovery. de Lacalle and Agirre (2015) also used data from MASC to train the It Makes Sense (IMS) (Zhong and Ng 2010) WSD model and produced good results. In İlgen *et al.* (2012), a Lexical Sample Dataset for Turkish is developed. The dataset comprises of noun and verb sets with each 15 highly ambiguous words. For each ambiguous word, the dataset contains at least 100 instances collected from various online sources. In Rakho *et al.* (2012), WSD corpus is created for 20 polysemous French verbs. The corpus created has been annotated using four different sense inventories taking into account various types of evidences. OMSTI dataset (Taghipour and Ng 2015) acronym for One Million Sense Tagged Instances is a huge sense tagged dataset and is a valuable asset for computational linguistics research. The dataset consists of one million English words, each of which has been sense-tagged according to its meaning in context. Its diversity makes it fruitful for resolving the English WSD problem. Le *et al.* (2018) utilized this dataset to carry out WSD task using LSTM (Hochreiter and Schmidhuber 1997) algorithm. Kokane and Babar (2009) used the OMSTI dataset to carry out the supervised WSD task based on neural networks. Kokane *et al.* (2023) utilized the OMSTI dataset along with other two datasets to carry out the WSD experiments based on the adaptive WSD model. The results showed that the proposed WSD model produced better results when utilizing the OMSTI dataset. Authors in Scarlini *et al.* (2020) also created large-scale multilingual WSD datasets for five different languages to mitigate the knowledge-acquisition bottleneck faced in multilingual WSD. The dataset released contains about 15 million instances with sense annotations for five languages (English, German, Spanish, French, and Italian). It also contains five English datasets belonging to distinct semantic domains. The dataset is annotated automatically instead of using the manual annotation method and when tested on supervised methods showed better performance compared to other automatically created corpora. In Pasini *et al.* (2021), cross-lingual evaluation framework for WSD, XL-WSD, covering sense-tagged datasets for 18 languages. The datasets so developed are then utilized for carrying out the WSD task implementing knowledge-based and neural network approaches. The results produced by these approaches are quite encouraging. In Zheng *et al.* (2021), Lexical-Sample WSD dataset is developed for Chinese. The developed corpus is then utilized to enhance the performance of the Chinese WSD task by utilizing word-formation knowledge. Kirillovich *et al.* (2022) created a sense-tagged Russian corpus. The corpus is created manually with raw data obtained from OpenCorpora[b] and senses from RuWordNet.[c] It contains

[b] http://opencorpora.org/
[c] https://www.ruwordnet.ru

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

a total of 6751 sentences with 109,893 lemmas. The corpus is then utilized for testing the unsupervised WSD system. In Laba *et al.* (2023), a fine-tuned pretrained language model is presented for Ukrainian WSD. A WSD evaluation dataset is created to achieve this objective. The dataset has 2882 samples each having lemmas, possible meanings with examples for each meaning.

2.2 WSD datasets for Indian languages

In comparison to foreign languages, Indian languages have not been explored much in the NLP domain, and there are no such standard linguistic resources available for these languages. Although researchers have made several attempts to tackle the WSD problem in Indian languages, but the datasets used are usually small in size and private. Here we will discuss some important WSD datasets available in Indian languages and important attempts to handle the WSD problem in these languages. WSD in Hindi, the national language of India, has been explored by different researchers. A Lexical Sample Hindi WSD dataset is discussed in Singh and Siddiqui (2016). The dataset has only 7506 for 60 ambiguous words. For this corpus, instances are reaped from Hindi Corpus[d] and Internet and senses from Hindi WordNet.[e] Based on this corpus, Hindi WSD task has been analyzed in research efforts presented in Singh and Siddiqui (2015a), Singh and Siddiqui (2015b); Singh *et al.* (2014). Singh and Siddiqui (2015a) used two supervised algorithms, one based on the conditional probability of co-occurring words and another one is the Naïve Bayes algorithm to analyze the impact of karaka relations on Hindi WSD. Singh and Siddiqui (2015b) compared the performance of three WSD algorithms trained on this dataset. The first algorithm used is corpus-based Lesk (1986), the second algorithm is based on conditional probability and co-occurring words and the third algorithm is based on a classification information model. Singh *et al.* (2014) used this dataset to develop Hindi WSD based on the Naïve Bayes algorithm. In this different feature combinations have been explored to check their impact on the performance of the WSD system. In Sarmah and Sarma (2016), WSD for Assamese is carried out based on supervised methodology. Naïve Bayes WSD model is trained on a manually created WSD corpus having 2.7K sentences, and it yielded 71% accuracy. Next, the size of the WSD corpus is increased to 3.5K sentences which enhanced the results by 7%. The WSD corpus is created utilizing Assamese corpus (Sarma *et al.* 2012) and Assamese WordNet (Sarma *et al.* 2010). In Walia *et al.* (2018), supervised methodology to WSD is adopted with reference to Punjabi language. The researchers created a WSD corpus for 100 ambiguous Punjabi words utilizing Punjabi Corpus obtained from ELRA[f] and Punjabi WordNet (Kaur *et al.* 2010). The same dataset has been utilized by other researchers for Punjabi WSD (Walia *et al.* 2020). Singh and Kumar (2020) also created a sense-tagged corpus and employed deep learning techniques for WSD. The corpus created has 9251 instances for 66 ambiguous Punjabi words with 25.7 average number of instances per sense. In Saeed *et al.* (2019), a benchmark Lexical Sample WSD corpus, ULS-WSD-18 Corpus, is developed for Urdu. UrMono Corpus Jawaid *et al.* (2014), the largest available raw Urdu corpus is utilized to obtain instances, and Urdu Lughat Board (2008) is used as sense inventory. For each target word, there are 75 + 15n instances (n gives the number of senses for the target word) in the developed WSD corpus. The applicability of the developed dataset for the development and evaluation of the Lexical Sample WSD system for Urdu is then checked by using different supervised WSD algorithms, viz. Naïve Bayes, SVM, ID3 Quinlan (1986), k-NN. Different types of features like bag-of-words (BoW), most-frequent sense, part-of-speech, and word embeddings were used to train these models so as to analyze their role in WSD. In Sruthi *et al.* (2022), unsupervised LDA-based WSD disambiguation methodology is employed to Malayalam. A WSD corpus with 1147 contexts is created

[d] http://www.cfilt.iitb.ac.in/Downloads.html

[e] https://www.cfilt.iitb.ac.in/wordnet/webhwn/

[f] http://www.elra.info/en/about/elda/

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

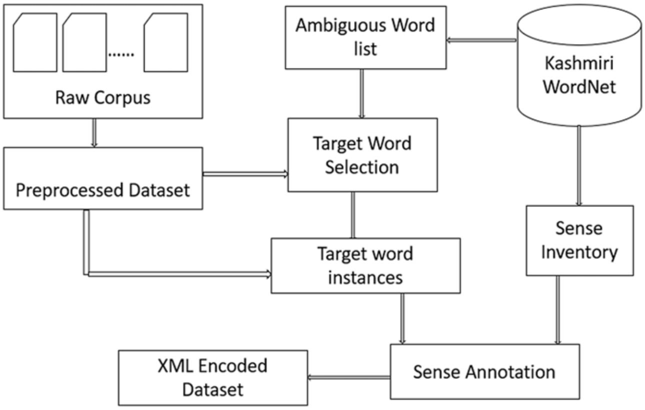


Figure 1. Architecture for sense annotated dataset preparation process.

for target words to carry out experiments. The experiments have shown better performance utilizing co-occurrence information and synonyms as features. In Das Dawn *et al.* (2023), Bengali sense-annotated corpus for the Lexical Sample WSD task is created. The dataset covers a total of hundred ambiguous Bengali words. There are three to four senses used in the annotation process for each ambiguous word. For each sense of a particular word used in the annotation, there are ten paragraphs in the developed corpus. The authors also proposed a generic architecture for utilizing this dataset to analyze the WSD task in Bengali. In Patil *et al.* (2023), WSD for Marathi is investigated using BERT Tenney *et al.* (2019). The model is evaluated on the WSD dataset having 5285 instances for 282 dubious terms.

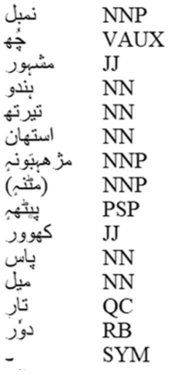
3. Dataset preparation process

This section is devoted to the systematic technique adopted to develop the sense-tagged dataset. In Subsection 3.1, data collection is discussed, Subsection 3.2 discusses the preprocessing used to clean raw corpus, Subsection 3.3 is devoted to the procedure followed to select the target words, Subsection 3.4 discusses the sense inventory selection procedure, Subsection 3.5 discusses the extraction of instances for the target words, Subsection 3.6 discusses the sense annotation process, and Subsection 3.7 discusses the encoding used for final corpus prepared. Various steps involved in the corpus development are depicted in the diagram below (Figure 1) and discussed in the following subsections.

3.1 Raw corpus

Due to the deficiency of data in digital form for the Kashmiri language, we faced a huge challenge in managing the dataset to carry out this research work. We explored different resources available both online and offline to prepare the dataset for smooth conduction of research work. PoS tagged dataset produced by Lawaye and Purkayastha (2014) was obtained which contains about 120K tokens. Other options available for obtaining data include Kashmiri WordNet (Kak *et al.* 2017), trilingual (English-Hindi-Kashmiri), and E-dictionary (Banday *et al.* 2009). In addition, we synthesized primary data wherever we found the deficiency. The corpus compiled from different resources is generic and has text from various domains like history, sports, politics, religion, and news.

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press



نمبل	NNP
چھُ	VAUX
مشہور	JJ
ہندو	NN
تیرتھ	NN
استھان	NN
مژھہبَونہٕ	NNP
(مٹنہٕ)	NNP
پیٹھہٕ	PSP
کھوور	JJ
پاس	NN
میل	NN
تار	QC
دوٗر	RB
۔	SYM

Figure 2. PoS tagged instance extracted from PoS tagged dataset.

3.2 Data preprocessing

The raw corpus written in the Perso-Arabic script, we collected in the first step which is not PoS tagged was passed through a few preprocessing steps so as to make it handy for further processing. The first step in the preprocessing we did was to tokenize it and divide it into sentences as it was too messy and the sentences were not properly separated. Using NLPTK library the dataset was tokenized. We stored the dataset in a separate text file (8 MB) with one sentence per line. As the data were collected from multiple resources, and there is no standard writing system in Kashmiri, the same word has different spelling in different resources. For example, the words حال ژاٹہٕ and ژاٹہاٖلہ seem different but are the same with different spellings. After analyzing the raw corpus carefully, we resolved all such cases by taking spellings from Kashmiri WordNet as standard. PoS tagger developed by Lawaye and Purkayastha (2014) is employed on the non-PoS tagged data for PoS tagging with an accuracy of 94%. The file containing the PoS-tagged dataset has one token per line along with its PoS tag. A sample of PoS data is shown in Figure 2.

3.3 Target word selection

The ultimate product of the dataset preparation process is the sense-tagged dataset for the Kashmiri Lexical Sample WSD task. In order to achieve this objective target words are selected. We extracted ambiguous-word list present in the Kashmiri WordNet. The PoS-tagged dataset is then analyzed for the presence of these ambiguous words. We calculated the frequency of ambiguous words present in the dataset. Out of the ambiguous words present in the dataset, we selected 124 words that have the maximum frequency in the dataset for further processing. Not only frequency we also checked whether the word takes different senses in the sufficient number of instances in the dataset so as to make it good for training the WSD model for predicting different senses that the word can take in different situations. For example, the word آمُت (aamut) although is ambiguous and has a good frequency in the dataset still it was not chosen as the target word as it takes single sense in most of the sentences out of the three senses available in the Kashmiri WordNet for it. Based on the same reason, we dropped many other words from the target word list, even though they have good frequency in the dataset. Some of these words include زیادٕ (zyad), آگُر (aagur), دِل (dil), اہم (aham), أمِل (shamil), اِستعمال (istimall), and عمل (amal). Table 1 gives important information about the ambiguous words selected as target words.

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

Table 1. Target-words with total senses in Kashmiri WordNet and instances in annotated Dataset.

S. No.	Ambiguous Word	Transliteration	Total Instances	No. of Senses
1	دور	Doer	162	9
2	لۆت	Loet	120	6
3	موج	mauj	164	3
4	راوُن	ravun	105	6
5	خٲلی	kaali	120	5
6	مانُن	manun	105	9
7	سیۆد	seud	150	13
8	پَن	pun	105	3
9	مِٹاوُن	mitavun	105	6
10	پَردٕ	parde	135	7
11	جاے	jai	174	9
12	آب	aab	192	6
13	دٔلیل	daleel	120	5
14	کار	kar	120	5
15	حِصہٕ	hiss	187	18
16	تھۆد	thud	156	5
17	پرٛون	proen	163	7
18	کَم	kum	186	9
19	کۄر	koer	158	5
20	راے	rai	105	7
21	صاف	saaf	242	23
22	کال	call	120	3
23	رُکاوُن	rukavun	135	7
24	وَتھ	weth	156	6
25	سخت	saketh	134	12
26	واریاه	vareyeh	1011	11
27	ؤلِتھ	mauj	127	6
28	نۆو	nove	111	11
29	پھِرُن	pherun	140	9
30	ہیۆر	heur	120	4
31	کٲم	kaem	175	5

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

Table 1. Continued.

S. No.	Ambiguous Word	Transliteration	Total Instances	No. of Senses
32	ؤسِتھ	weseth	105	3
33	پُھٹُن	phetun	105	4
34	گوٟڈ	gued	120	4
35	زمانہٕ	zamaane	141	4
36	سیر	saer	105	3
37	راتھ	rath	140	4
38	لَگاوُن	lagavun	120	5
39	پیۆن	peun	135	9
40	تَھپ	theph	105	3
41	جان	jaan	174	16
42	بَنٛد	band	150	8
43	پالُن	palun	105	3
44	َبَرُن	barun	150	8
45	بۆڈ	boed	161	7
46	بُتھ	buth	162	7
47	نظر	nazar	160	4
48	ژَلُن	cxelun	250	39
49	پھیرٕ	faer	105	2
50	پَھسُن	phasun	140	4
51	آرام	aram	154	5
52	بار	bar	105	3
53	تیز	taez	195	24
54	لاگُن	lagun	136	22
55	سٕژ	cxech	105	3
56	کِرایہٕ	kerai	105	2
57	ووٕتُھن	wuthun	135	9
58	وایُن	wayun	105	3
59	رَلُن	ralun	150	10
60	شال	shal	105	5
61	سیٖٹ	seat	120	4

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

Table 1. Continued.

S. No.	Ambiguous Word	Transliteration	Total Instances	No. of Senses
62	ہٳر	haar	105	4
63	دۄر	door	220	7
64	ترٛاوُن	travun	575	29
65	پھرِتھ	ferith	105	3
66	لَگُن	lagun	156	14
67	ِبِتھ	behit	132	8
68	مُشکِل	mushkil	141	20
69	زامُت	zamut	120	2
70	تَکلیٖف	takleef	120	4
71	کۆچ	koech	165	8
72	پَکہٕ	paek	139	3
73	اَصِل	asal	330	12
74	واٹھ	wath	180	6
75	پٳس	pass	160	5
76	خَبَر	khabar	154	4
77	تُلُن	tulun	150	26
78	یَلہٕ	yel	135	11
79	کَھسُن	khasun	171	11
80	نیرُن	neeran	155	21
81	رَٹُن	ratun	170	15
82	کھیۆن	kheun	130	13
83	دَزُن	dazun	159	13
84	کَڑُن	kadun	145	23
85	کھالُن	khalun	134	11
86	کھولُن	kholun	162	11
87	وَسُن	wasun	135	13
88	تراوُن	travun	150	33
89	بَناوُن	banavun	160	16
90	مۆزۄری	mozoori	165	15
91	واتُن	watun	159	12
92	قأبِل	kael	210	6

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

Table 1. Continued.

S. No.	Ambiguous Word	Transliteration	Total Instances	No. of Senses
93	مُخٲلِف	mukhalif	176	8
94	زور	zoer	203	6
95	لال	laal	145	6
96	رَس	rus	160	7
97	داو	daev	115	6
98	گِنٛدُن	gindun	165	9
99	نَرٕم	narum	179	9
100	جال	jaal	156	6
101	اِشارٕ	ishare	147	7
102	کوٚنٛڑ	kund	163	8
103	بَچہٕ	bacche	135	8
104	تنٛگ	tang	159	6
105	ژوٚک	cxook	152	8
106	آواز	awaz	174	9
107	توٚن	tun	147	7
108	ٹاس	tass	142	7
109	داغ	daag	145	5
110	خَراب	kharab	154	5
111	ہاوُن	haavun	138	11
112	لایُن	layun	177	11
113	پَکُن	pakun	153	12
114	نِشانہٕ	nishaan	174	12
115	ٹھیک	theek	157	12
116	آزاد	azad	150	13
117	دورٕ	dore	105	2
118	گَنٛڈ	gand	120	5
119	زَنٛگ	zeng	105	3
120	دَباوُن	dabavun	150	10
121	جوڈ	joed	120	3
122	گوٚل	gul	135	5
123	حِساب	hisaab	120	4
124	موٚکُر	mukur	150	10

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

Table 2. Senses for word تھوٚد (*thud*) in Kashmiri WordNet

S.No.	Meaning	Transliteration
1	يُس تَھزرَس آسہِ	yus thazeres aasi
2	يُس زاتہِ ،اُہدَس ،خوٚبی ، بیترِ منٛز ہیوٚر آسہِ	yus zate auhds khubi baiter manz heur aasi
3	يُس عام سطح یا جایہِ کھوتہٕ تھزرس آسہِ	yus aam satah ya jaaye khote thazres aasi
4	يُس زیوٚٹھ قَدُک آسہِ	yus zeuth kaduk aasi
5	لاگنَس مَنٛز ژھوٚٹ یا تھوٚد	lagnes manz cxoot ya thud

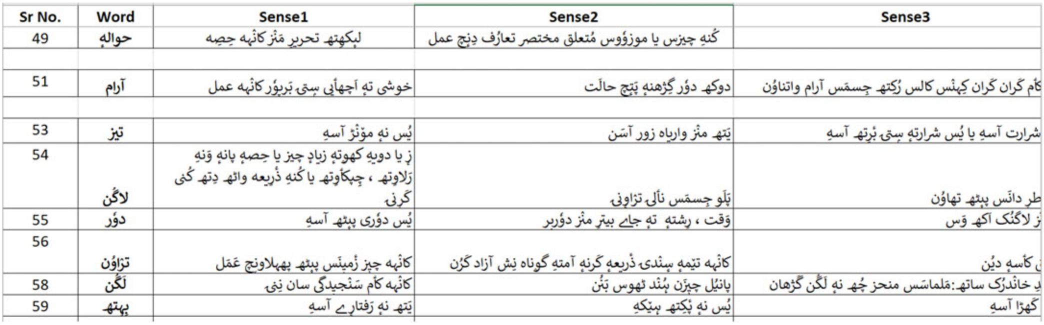
3.4 Sense inventory

After finalizing the target word list, we need possible senses that these words can take in different contexts. To get the different senses resources like machine readable dictionaries, Wikipedia, and WordNet have been used by researchers. In this research work, we had two resources available to us; Kashmiri WordNet and trilingual (English-Hindi-Kashmiri) E-Dictionary. We carefully analyzed the senses obtained from these resources to select the best option for sense inventory creation. It was observed that Kashmiri WordNet has a greater number of senses for many target words than are available in trilingual E-Dictionary. From Kashmiri WordNet, the range of senses for target words spans from 2 to 33 which is higher than the range of senses for these words available in trilingual E-Dictionary. For example, for the word تھوٚد*(thud)* having NOUN as a lexical category, Kashmiri WordNet gives 5 senses as shown in Table 2 below.

In trilingual (English-Hindi-Kashmiri) trilingual E-Dictionary the word دور *(Door)* has 11 meanings. For the word تراوُن *(travun),* Kashmiri WordNet has 33 meanings, whereas trilingual E-Dictionary returns 20 meanings for the word. There are 9 senses for the word مانُن *(manun)* in Kashmiri WordNet, whereas trilingual E-Dictionary contains only 7 senses for this word. Kashmiri WordNet gives 13 senses for the word سیوٚد *(seud)* for which trilingual E-Dictionary has only 7 senses. Likewise, there are 12 senses for the word پَکُن *(pakun)* in Kashmiri WordNet, whereas trilingual E-Dictionary has only 2 senses for the same. There are many other cases where Kashmiri WordNet gives more senses than trilingual E-Dictionary; hence, we choose it for sense inventory creation. After choosing Kashmiri WordNet for sense inventory creation, we extracted senses for the target words from Kashmiri WordNet. Kashmiri WordNet actually divides the words in terms of concepts. The fundamental unit in Kashmiri WordNet is synset in the same manner WordNet for English and other languages. Each word is associated with a synset and has a synset ID, lexical category, concept, and example sentence. Kashmiri WordNet can be accessed from the Indo WordNet[g] interface. Two important characteristics that we found in the senses available in the Kashmiri WordNet for the target words are a) the senses are very much fine-grained and b) it gives senses that are not commonly used. As far as our task is concerned, it is very much difficult to get instances for fine-grained senses. So, to overcome this problem, we adopted the same procedure as is followed by other researchers, i.e., convert these fine-grained senses into coarse-grained senses. This transformation involves grouping the fine-grained senses that are similar in meaning to a more general sense. This objective is achieved by following the sense compression technique proposed in Vial *et al.* (2019). The compression technique utilizes the different semantic relationships (hypernymy/hyponymy/synonymy/antonymy/meronymy/holonymy. . .) available in Kashmiri WordNet. Synsets are divided into clusters C(c_1, c_2, . . ., c_n) based on these relationships. These clusters are formed iteratively. Initially, each synset from the synset list S(s_1, s_2,. . ., s_n) is placed in a separate cluster, i.e., C= ((s_1), (c_2), . . ., (c_n)). Then at each iteration, we

[g] https://www.cfilt.iitb.ac.in/indowordnet/

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press



Sr No.	Word	Sense1	Sense2	Sense3
49	حوالہٕ	لبکھتھ تحریرِ مَنٛز کانٛہہ حِصہ	کُنہِ چیزس یا موزوٗوس مُتعلق مختصر تعارُف دِنٛچ عمل	
51	آرام	خوشی تہٕ آچھأیِ سِتی بَرپوٗر کانٛہہ عمل	دوکھ دوٗر گژھنہٕ پَتچ حالَت	کأم گران گران کِہنٛس کالس رُکتھ جِسمَس آرام واتناوُن
53	تیز	یُس نہٕ موٗنٛژ آسہِ	یَتھ منٛز واریاہ زور آسَن	شرارت آسہِ یا یُس شرارتہٕ سِتی بُرتھ آسہِ
54	لاگُن	زٕ یا دویہِ کھوتہٕ زیادٕ چیز یا حِصہٕ پانہٕ وَنہِ زَلاوِتھ ، چِپکأوِتھ یا کُنہِ ذٔریعہ واٹھ دِتھ کُنی گرنی	پَلو جِسمَس نألی تڑاوِنی	طرِ دانَس پیٹھ تھاوُن
55	دوٗر	یُس دوٗری پیٹھ آسہِ	وَقت ، رِشتہٕ تہٕ جاے بیترِ منٛز دوٗرِیر	زٕ لاگنُک اکھ وَس
56	تڑاوُن	کانٛہہ چیز زٔمینَس پیٹھ پھہلاونچ عَمَل	کانٛہہ تیٚمہٕ سِنٛدی ذٔریعہٕ کَرنہٕ آمتہِ گوناہ نِش آزاد کَرُن	ن کأسہٕ دِیُن
58	لَگُن	کانٛہہ کأم سَنٛجیدگی سان نِنی	پانیٛل چیزَن ہُنٛد تھوس بَنُن	دِ خانٛدرُک ساتھ:مَلماسَس منحز چُھ نہٕ لَگُن گژھان
59	پہتھ	یَتھ نہٕ رَفتارٕے آسہِ	یُس نہٕ پُکِتھ ہیٚکہِ	گھڑا آسہِ

Figure 3. Sense inventory snapshot.

arrange C according to the sizes of the clusters, and then we analyze whether the smallest cluster c_i is related to another cluster to c_j. If the two clusters are related by having some common semantic relation, then we merge the two clusters c_i,c_j. If there is no common semantic link between the two clusters, then the merge operation is canceled and then we take the next smallest cluster and repeat the same process. The algorithm stops when there is no further link possible. After converting fine-grained senses into coarse-grained senses, we created a sense inventory. The sense inventory generated contains the target word and the set of senses finalized to be used in the annotation process for the target words. Figure 3 shows a snapshot of the sense inventory.

3.5 Target word instances

The instances for the target words are extracted from the PoS-tagged dataset that we created after the preprocessing and PoS tagging discussed in Subsection 3.2. We obtained at least 75 + 15n (n is the number of senses for target word) number of sentences for each target word as per the guidelines set in the SenseEval-2 task for Lexical-Sample WSD task (Edmonds and Cotton 2001; Saeed *et al.* 2019). For example, the target word تیز *(taiz)* has 8 senses used in annotation and has (75 + 15 × 8) 195 instances in the final sense-tagged corpus.

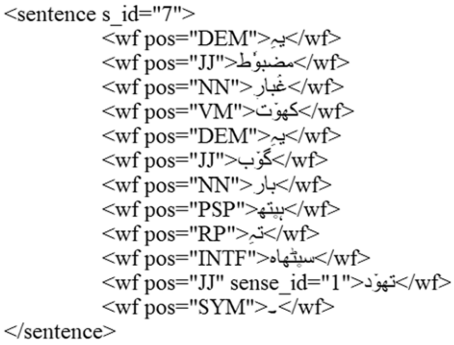
3.6 Sense annotation

The next phase in the dataset preparation process was to carry out the annotation. We employed the manual sense annotation technique for this purpose. Three annotators all naïve speakers of Kashmiri were given this task after making them familiar with the annotation task. To achieve a high-quality annotated corpus, we followed the steps used in Saeed *et al.* (2019). We divided the overall annotation process into two phases. Two annotators were given the task to annotate the target word with the appropriate sense as per the context. They were provided with files containing sentences separately for each target word and the senses to be used for annotation. Once the annotators finished the annotation process, they discussed the annotation task and resolved the cases where they found difficulty or confusion in the annotation. We then calculated the inter-annotator agreement and found it 92% and 0.83 weighted Kappa score (McHugh 2012) that indicated a high level of agreement. In the second phase of the annotation, the linguist expert was involved in reviewing the annotated dataset so as to make corrections wherever necessary.

3.7 XML encoding

The output of the preceding task is sense tagged corpus where the target words are tagged with the appropriate sense. This corpus contains simple text files one for each target word. In the next

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press



```
<sentence s_id="7">
                <wf pos="DEM">یہِ</wf>
                <wf pos="JJ">مضبوٗط</wf>
                <wf pos="NN">غبار</wf>
                <wf pos="VM">کھوٚت</wf>
                <wf pos="DEM">یہِ</wf>
                <wf pos="JJ">گوٚب</wf>
                <wf pos="NN">بار</wf>
                <wf pos="PSP">ہِیتھ</wf>
                <wf pos="RP">تہِ</wf>
                <wf pos="INTF">سیٹھاہ</wf>
                <wf pos="JJ" sense_id="1">تھوٚد</wf>
                <wf pos="SYM">۔</wf>
</sentence>
```

Figure 4. Example sentence from sense annotated corpus.

step, we transformed the sense-tagged corpus into XML format which is the commonly used format for storing sense-tagged datasets. This task is carried out using xml.etree. ElementTree module available in Python. The XML-encoded corpus has 124 total files one for each target word. Each file has <contextfile fileno='filenumber' filename='kmr_word'>as the root element in which the *fileno* attribute specifies the file number (1,2, . . . 124) and the *filename* attribute which specifies the target word for which the file contains instances. Each sentence starts with <sentence s_id="sentence_number">element which indicates the start of a new sentence and the *s_id* attribute gives the sentence number assigned to the sentence in the file. Each word in the sentence is enclosed within the <wf>element pair which has *pos* as an attribute specifying the PoS category of the word. <wf>has *sense_id* as an additional attribute for target words which specifies the unique number assigned to the sense that fits the target word in the given sentence. Figure 4 below shows an instance of the target word تھوٚد *(thud)* in the sense annotated corpus.

The sense-tagged dataset so created is finally divided into two partitions, training set and test sets for each target word. Splitting the dataset into training and test suites is governed by the following:

- Total instances for a particular word are divided into 80:20 ratio for training and test purposes.
- In order to make sure there is a proper mix of instances for each sense that a target word has in both training and test suites, 20% of instances for each sense are placed in the test suite and 80% instances in the training suite.
- Training and test splits for each target word are stored in separate XML file.

4. WSD experiments

After preparing the sense-tagged dataset, we used it to carry out experiments. We used it on different supervised machine learning methods to check its effectiveness for analyzing and assessing WSD models. To train the machine learning models, we used three distinct feature extraction techniques; most frequent sense (MFS) (Gale *et al.* 1992), traditional Bag of Words (BoW) (Goldberg and Hirst 2017) and Word2Vec[h] which are briefly discussed below. Different machine learning algorithms trained on these features are evaluated for their performance for handling Kashmiri WSD.

[h] https://code.google.com/archive/p/word2vec/

4.1 Most frequent sense

In natural languages, it is common that a polysemous word takes one sense more frequently than other senses it can have. This most frequently occurring sense of polysemous word is termed as MFS. In WSD systems, MFS-based approach is commonly used as a baseline to assess the performance of other machine learning techniques on a given dataset (Saeed *et al.* 2019; Kumar 2020). In this research work, we compared the results produced by other algorithms with MFS-based method.

4.2 Bag of words (Count vectorizing) model

BoW, also known as count vectorizing, is the basic model used to transform the raw data into a vector representation. In this approach, we calculate the prevalence of content words in the given data. This model converts the given input into numeric form in which different dimensions represent the words existing in the given input without taking into account the syntax or semantics. The dimensions may simply contain 0s or 1s representing the presence or absence of words in the given corpus or it may also represent frequency for that word in the given corpus. By default, this approach carries out some preprocessing steps like conversion of input to lowercase, utf-8 encoding, ignoring single characters, and punctuations. It also provides the user with the facility to customize tokenization, preprocessing, etc. so that it can be made helpful for languages other than English. In this study, we have implemented this approach for the Kashmiri language.

4.3 Word2Vec model

It is a deep-learning approach used to obtain dense vector representation for a given input. The vector representations computed using Word2Vec have proven to be fruitful in different downstream NLP tasks. Word2Vec requires a large corpus as input and generates the vocabulary of all the words in the corpus. Finally, dense embeddings are obtained for each word in the vocabulary. As vector representations produced by Word2Vec capture semantic and contextual information from the given input very well, many researchers have used this approach to solve the WSD problem (Kumari and Lobiyal 2022; Saeed *et al.* 2019). There are two variants present in Word2Vec to produce vector representations; a) continuous bag-of-words (CBOW) and b) Skip-Gram. The CBOW variant uses the context words in a specific window size to forecast the middle word (target word). It works on the assumption that the order of words in the context widow is irrelevant to the prediction hence is named so. The objective of the Skip-Gram variant of Word2Vec is to forecast the context words within a specific window size for a target word. In this study, we employed the Skip-Gram architecture to obtain word embeddings (WE) for all the target words. The model is trained on the whole corpus with the objective of maximizing the probability of forecasting the context in which the target word exists. Considering the word sequence ($w_1, w_2, \ldots, w_n$), the objective function can be written as (Mikolov *et al.* 2013):

$$\frac{1}{T}\Sigma_{t=1}^{T}\Sigma_{-n\leq i\leq n, i\neq 0}\log p(x_t+i|x_t) \tag{1}$$

where n is training context size. The basic formulation of Skip-Gram defines $p(x_t+i|x_t)$ using softmax function :

$$p(w_O|x_I) \equiv \frac{exp(v'w_O \top v_wI)}{\Sigma_{w=1}^{W} exp(v'w \top v_wI)} \tag{2}$$

vw and v’w depict input and output vector representations of w. w gives word count in vocabulary. We used the Gensim library to create the Word2Vec model. The model so created is used to obtain the nearest word embeddings for all target words keeping dimensions set to different values (100,

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

200, 300). Using these embeddings as features, we trained different machine learning algorithms and then tested these algorithms.

4.4 Classification algorithms used

Once the feature extraction process is over in the next step, we use these features to train machine learning algorithms. In this research work, we employed five classification algorithms which are discussed briefly here:

4.4.1 Naive Bayes

Naïve Bayes WSD refers to a technique that uses the Naive Bayes algorithm to perform WSD. It calculates the probability of a word belonging to a particular sense given the context in which it appears. This is done by estimating the conditional probabilities of the features (e.g., surrounding words and syntactic patterns) given each sense and then applying Bayes' theorem to obtain the posterior probability of each sense. In Naive Bayes WSD, the sense with the highest posterior probability is chosen as the disambiguated sense for the word in question.

4.4.2 Dl4jMlpClassifier

Dl4jMlpClassifier (Lang *et al.* 2019) is a deep learning library for Java that provides tools and functionality for building and training deep neural networks. The Dl4jMlpClassifier algorithm is based on a multilayer perceptron (MLP) architecture, which is a type of feedforward neural network. MLPs consist of multiple layers of interconnected artificial neurons, where each neuron is a mathematical function that takes input from the previous layer and produces an output.

4.4.3 IBk

IBk algorithm (Reynolds *et al.* 2011), also known as the k-nearest neighbors (k-NN) algorithm, is a simple and intuitive machine learning algorithm that classifies new instances based on their proximity to labeled instances in a feature space. The algorithm is trained on a set of features that capture the context in which the word appears. Given a new word instance to disambiguate, the algorithm identifies the k nearest neighbors in the feature space having the most similar feature representations to the new instance. The algorithm assigns the most frequent or highest-weighted sense from the neighbors as the disambiguated sense for the new word instance.

4.4.4 J48

J48 is a classification algorithm based on the C4.5 decision tree algorithm Quinlan (1993). The J48 algorithm constructs a decision tree based on the training data. The decision tree is a hierarchical structure of nodes and branches, where each node represents a feature test, and each branch corresponds to a specific outcome or value of the feature. Given a new word instance to disambiguate, the algorithm traverses the decision tree by evaluating the feature tests at each node. It follows the appropriate branch based on the feature values of the instance until reaching a leaf node. The leaf node reached by the instance determines the disambiguated sense. Each leaf node is associated with a specific word sense, and the majority or most frequently occurring sense among the instances falling into that leaf node is chosen as the disambiguated sense for the new word instance.

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

4.4.5 Support vector machines (SVM)

SVMs are binary classifiers that can be extended for multiclass classification, which is suitable for WSD where multiple word senses need to be distinguished Navigli (2009). The SVM algorithm learns a hyperplane that separates the feature vectors of different word senses in the training data. The goal is to find a hyperplane that maximizes the margin, i.e., the distance between the hyperplane and the nearest instances of each word sense. This hyperplane acts as the decision boundary for classification. Given a new word instance, the algorithm converts its features into a feature vector and applies the learned SVM model to classify the instance. The instance is assigned the word sense corresponding to the side of the decision boundary it falls into.

4.5 Evaluation

Experiments are carried out on all target words separately using classification algorithms discussed in the above section. To get a better overview of the performance reported by system and avoid complexity, we calculated the average of all the results produced by the system. To evaluate the performance of the WSD classifiers developed based on the created sense annotated corpus we used standard evaluation measures commonly used in machine learning, i.e., accuracy; precision, recall, and F1-measure.

Accuracy (A): Accuracy(A) is the accurate sense assignments made by the system over the total number of instances.

$$Accuracy(A) \equiv \frac{accurateassignmentsmade}{totalinstances} \tag{3}$$

Precision (P): Precision (P) is obtained by dividing the correct sense assignments total assignments made by the system.

$$Precision(P) \equiv \frac{correctassignments}{totaltestsetinstances} \tag{4}$$

Recall (R): Recall (R) is obtained by dividing the correct assignments by the total expected assignments by the system.

$$Recall(R) \equiv \frac{accurateassignmentsmade}{totalassignment} \tag{5}$$

F1-measure: Obtained by using equation (6) and is the harmonic mean of P and R.

$$F1 - measure \equiv \frac{2 * P * R}{P + R} \tag{6}$$

5. Results and discussions

The machine learning algorithms used produced different results based on the feature sets used. Machine learning algorithms were trained using the training suites and then evaluated using the test suites. Results produced by various algorithms used on the basis of features are compared and are shown in Table 3.

From Table 3, it is clear that all the algorithms used showed significant improvement in performance as compared to the baseline approach (MFS). Based on the feature set used to train the different machine learning WSD classifiers, there is a variance in their performance. SVM-based WSD model performed better in comparison to other WSD models with accuracy = 71.84%, precision = 0.71, recall = 0.70, and F1-measure = 0.70 when trained on BoW features. IBk-based WSD model trained on BoW features also performed well giving results close to SVM model with accuracy = 71.01%, precision = 0.70, recall = 0.69, and F1-measure = 0.69.

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

Table 3. Results Produced by Different Machine Learning Algorithms Using Different Features

Features	Machine Learning Model	Accuracy (A)	Precision (P)	Recall (R)	F1-measure
MFS	Baseline	52.47			
BoW	Dl4jMlpClassifier	70.43	0.68	0.69	0.68
	IBk	71.01	0.70	0.69	0.69
	Naive Bayes	67.08	0.66	0.64	0.65
	SVM	71.84	0.71	0.70	0.70
	J48	64.23	0.63	0.64	0.63
WE-100	Dl4jMlpClassifier	68.97	0.67	0.68	0.67
	IBk	67.74	0.67	0.66	0.66
	Naive Bayes	64.79	0.63	0.64	0.63
	SVM	67.97	0.67	0.67	0.67
	J48	62.98	0.62	0.61	0.61
WE-200	Dl4jMlpClassifier	73.74	0.72	0.72	0.72
	IBk	71.72	0.71	0.70	0.70
	Naive Bayes	68.88	0.67	0.68	0.67
	SVM	72.57	0.72	0.71	0.71
	J48	66.76	0.65	0.65	0.65
WE-300	Dl4jMlpClassifier	71.99	0.71	0.70	0.70
	IBk	71.04	0.71	0.69	0.70
	Naive Bayes	68.43	0.67	0.65	0.66
	SVM	71.90	0.71	0.70	0.70
	J48	65.35	0.65	0.64	0.64

WE-100 specifies that the feature vector for the target word is obtained using 100 words surrounding it, WE-200 specifies that 200 surrounding words are utilized and WE-300 specifies that 300 neighboring words are utilized to obtain the feature vector for the target word.

Dl4jMlpClassifier-based WSD model follows IBk-based WSD model in performance giving accuracy = 70.43%, precision = 0.68, recall = 69, and F1-measure = 0.68. Performance of the Naïve Bayes WSD model is lower than Dl4jMlpClassifier with accuracy = 67.08%, precision = 0.66, recall = 0.64, and F1-measure = 0.65. The lowest results with accuracy = 64.23%, precision = 0.63, recall = 0.64, and F1-measure = 0.63 are produced by the WSD model based on decision-tree when used with BoW features.

On using Word2Vec-based features with dimensions 100, 200, and 300, it is observed that all WSD classifiers showed lower performance than the BoW-based approach when using dimension size 100 (WE-100). However, using dimension values as WE-200 and WE-300 improved the performance of WSD classifiers and outperformed the BoW-based model.

As given in Table 3 among the different WSD algorithms used, the Dl4jMlpClassifer-based WSD model produced the best results with accuracy = 73.74%, precision = 0.72, recall = 0.72, and F1-measure = 0.72, when trained with WE obtained by using the dimensional value 200 and decision tree-based WSD model produced lowest results with accuracy = 66.76%, precision = 0.65,

Table 4. Senses predictions for the word کٲم (*kaem*) by Dl4jMlpClassifier

S.No.	Sentence	Correct Sense	Predicted Sense
1	یُشری مۆزوٗرَن اَتھہِ کٲم کَرناوِنی چُھ گوٗناہ	Sense 1	Sense 3
2	تسوٗنچِتھ سمجھِتھٕ کٕر مےٚ یتھ فرٛنس پیٹھ کٲم	Sense 2	Sense1
3	بےٚ ہِمتی سٟتی ہیٚچ نہٕ تٔمی مکانس کٲم کٔرِتھٕے	Sense 3	Sense 1
4	تٔمی سٕندس فرنَس چھُٚے نہٕ اَصِل کٲم گٔمٕژ	Sense 2	Sense 1
5	سُہ چُھ یَننہِ آرامہٕ مُطٲبِق مکانس کٲم کَران	Sense 3	Sense 1
6	سُہ چُھ کینٛہہ ضروٗری کٲم کَرنہٕ خٲطرٕ شہرٕ نیٚبَر	Sense 1	Sense 2
7	یٔمی یَلوِچ کٲم چھُٚے وِنہِ تہِ اَڑلیٚچِے	Sense 2	Sense 1

WSD Model Behaviour

recall = 0.65, and F1-measure = 0.65 using the same settings. The results shown in Table 3 clearly show that WSD models trained with word embedding perform better than WSD models trained using the BoW model in most cases. This led us to the conclusion that word embeddings are better at performing WSD in comparison to the BoW model.

From the overall results presented in Table 3, it may be observed irrespective of the feature extraction technique used WSD model created using the decision-tree algorithm does not perform to the level other WSD models perform. The main reason behind this may be that the decision tree does not get enough information from the features to make the sense prediction. On the other hand, the SVM-based WSD model makes better predictions in comparison to other WSD models when used with the BoW approach. The main reason behind the better performance of SVM may be attributed to its ability to handle data sparsity and being less sensitive to noise (Batuwita and Palade 2013).

Table 4 shows the wrong sense predictions made by the Dl4jMlpClassifier for the target word کٲم (*kaem*). “Correct” specifies the sense assigned to the word کٲم (*kaem*) in the sense annotated dataset, whereas the “Predicted” specifies the sense assigned by the Dl4jMlpClassifier-based WSD classifier to the word کٲم (*kaem*) in the concerned instance. There are three senses present in the sense inventory for the word کٲم (*kaem*): پیٚہتہِ یہِ کَرنہٕ (*te ye karne eie*), پَلوَس وغٲرَس پیٹھ یَننہٕ سٟتیٖ پوش یاتھَرِ وغٲرُک بناونہٕ آمُت نموٗنہٕ (*palves wagaeres peth pane saet posh ya ther wagaeruk banavne aamut namoen*), and بَناوِنچ کٲم (*banavnech kaem*). From Table 4, it is clear that mistakes are between Sense 2 and Sense 1 showing the behavior of the WSD algorithm.

5.1 Proposed WSD model error analysis

When the erroneous results produced by different WSD models based on different machine learning algorithms are analyzed, it is observed that the main reason behind the errors in sense predictions is the higher number of exhibiting senses for ambiguous words. The distribution of instances for different senses of an ambiguous word in the training dataset is nonuniform. The dataset has more sentences for MFS of target words making it skewed toward MFS. The results produced by the proposed WSD system highlight that the ambiguous words with a smaller number of senses are disambiguated better than the ambiguous words with a larger number of senses, i.e., with the increase in the number of senses the performance of the system diminishes. Figure 5 shows the degree to which the accuracy of the WSD model gets impacted by the increase in the

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

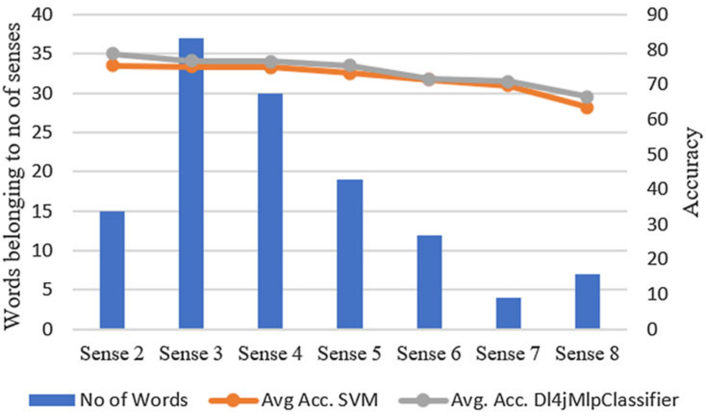


Figure 5. Average accuracies of SVM and Dl4jMlpClassifier with respect to number of senses.

number of senses. The figure shows the average accuracy reported by the two best-performing WSD models one is the SVM model which performed best when the BoW model was used for feature extraction and the second is Dl4jMlpClassifier which outperformed other WSD models when word embeddings were used as features.

In Figure 5, the X-axis gives the number of senses for the target words which is constrained in this study between 2 and 8. The Y-axis on the left side gives the number of target words having a specific number of senses, whereas the Y-axis present on the right side gives the average accuracy returned by the classifier (SVM and Dl4jMlpClassifier). From the graph, the performance degradation with the increase in the number of senses of words is visible. This is in line with the research efforts presented in Saeed *et al.* (2019) and Singh and Kumar (2019).

It is also observed that in the dataset there are sentences where a particular ambiguous word exists with similar words in the context but gives different interpretations. For example, sentences نٔوس شہرَس منٛز رأو تَس وَتھ تہٕ ووت سِٹیشَن (*naves shahres manz raev tus weth te voet station*) and شہرَس منٛز رأو تَس وَتھ تہٕ لأجٕنی ژۆر کَرٕنی (*shahres manz raev tus weth te laejen cxoor karin*) contain many similar content words, but they give different interpretations for the ambiguous word وَتھ (*weth*). In such cases, it is difficult for the WSD model to make the right sense prediction.

Another important factor that has an influence on sense prediction is the presence of more than one ambiguous word in the same sentence. For example, in the sentence وَتھ چھےٚ سیوٚد مقامس تام واتناوان یہٕ (*ye weth che seud makans taam watan*), words وَتھ (*weth*) and سیوٚد (*seud*) are ambiguous, and it is difficult to predict the correct sense of both the words in such cases.

In order to collect the instances for the target words, we have used Kashmiri WordNet in addition to other resources, the sentences extracted from the Kashmiri WordNet are very short in many cases. These short sentences do not provide enough contextual information for the WSD model to make correct sense prediction. For example, the sentence کم کال کھہٕ (*call khe cum*) has only three words in it, and for a WSD model, it is difficult to make the right sense prediction for the word کال (*call*).

6. Conclusion

The main objective of this research work was to develop a standard Lexical Sample WSD dataset for Kashmiri. The developed Lexical Sample WSD dataset contains 124 frequently used ambiguous Kashmiri words and 19854 instances. The instances for the selected words were extracted

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

from different resources, and Kashmiri Wordnet was used to develop the sense inventory. In addition to developing the sense-tagged dataset, the dataset was used to build WSD models to carry out disambiguation based on different feature sets. Five machine learning algorithms (SVM, J48, IBk, Naïve Bayes, and Dl4jMlpClassifier) were used to build WSD classifiers. The classifiers were trained using BoW model and word embeddings created using Word2Vec architecture. On the analysis of the results produced by different machine learning algorithms, it was observed word embeddings approach shows better performance than traditional BoW-based models. Among the different machine learning models, SVM showed the best performance when trained with BoW features. On the other hand, Dl4jMlpClassifier outperformed its counterparts when used with word embeddings. Out of the research work different observations were made like the performance of a WSD model is negatively impacted by the level of ambiguity a word has, the difficulty faced by the WSD model to predict the sense correctly when present in sentences with similar content words, shortage of contextual information provided by smaller sentences to make sense prediction. In the future, the dataset would be enhanced by incorporating sense-tagged instances for more ambiguous terms so as to increase language coverage. Also, deep learning techniques would be used to carry out experiments.

References

Abderrahim M. A. and Abderrahim M. E. A. (2022). Arabic word sense disambiguation for information retrieval. *Transactions On Asian and Low-Resource Language Information Processing* **21**(4), 1–19.

Agirre E. and Edmonds P. (eds) (2007). *Word Sense Disambiguation: Algorithms and Applications*, vol., **33**.Springer Science & Business Media.

Baker P., Hardie A., McEnery T., Cunningham H. and Gaizauskas R. J. (2002). EMILLE, a 67-million word corpus of Indic languages: Data Collection, Mark-up and Harmonisation. In *LREC'02*, Las Palmas, Canary Islands - Spain, pp. 819–825.

Banday T. A., Panzoo O., Lone F. A., Nazir S., Malik K. A., Rasoool S. and Maqsood S. (2009). Developing a trilingual (English-Hindikashmiri) E-dictionary: issues and solutions. *Interdisciplinary Journal of Linguistics* **2**(1), 295–304.

Batuwita R. and Palade V. (2013). Class imbalance learning methods for support vector machines. In *Imbalanced Learning: Foundations, Algorithms, and Applications*, pp. 83–99.

Board U. D. (2008). *Urdu Lughat.*Karachi: Urdu Lughat Board.

Das Dawn D., Khan A., Shaikh S. H. and Pal R. K. (2023). A dataset for evaluating Bengali word sense disambiguation technique. *Journal of Ambient Intelligence and Humanized Computing*, **14**, 4057–4086.

de Lacalle O. L. and Agirre E. (2015). A methodology for word sense disambiguation at 90% based on large-scale crowdsourcing. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, Denver, Colorado, pp. 61–70.

Edmonds P. and Cotton S. (2001). Senseval-2: overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pp. 1–5.

Gale W. A., Church K. and Yarowsky D. (1992). Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *30th Annual Meeting of the Association for Computational Linguistics*, Newark, Delaware, USA, pp. 249–256.

Goldberg Y.Hirst G. (2017). *Neural network methods in natural language processing*. San Rafael: Morgan & Claypool Publishers.

Hochreiter S. and Schmidhuber J. (1997, *Long Short-Term memoryNeural Computation* **9**(8), 1735–1780.

Haouassi H., Bekhouche A., Rahab H., Mahdaoui R. and Chouhal O. (2024). Discrete student psychology optimization algorithm for the word sense disambiguation problem. *Arabian Journal for Science and Engineering* **49**, 3487–3502.

Ide N., Baker C. F., Fellbaum C. and Passonneau R. J. (2010). The manually annotated sub-corpus: A community resource for and by the people. In *Proceedings of the ACL. 2010 Conference Short Papers*, Uppsala, Sweden, pp. 68–73.

İlgen B., Adali E. and Tantuğ A. C. (2012). Building up lexical sample dataset for Turkish word sense disambiguation. In *2012 International Symposium on Innovations in Intelligent Systems and Applications,* IEEE, Trabzon, Turkey, pp. 1–5.

Itankar P. Y. and Raza N. (2020). Ambiguity resolution: An analytical study. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* **6**(2), 471–479.

Jawaid B., Kamran A. and Bojar O. (2014). A tagged corpus and a tagger for urdu. In *LREC'14.* Reykjavik, Iceland, pp. 2938–2943.

Kak A. A., Ahmad F., Mehdi N., Farooq M. and Hakim M. (2017). Challenges, problems, and issues faced in language-specific synset creation and linkage in the Kashmiri WordNet. In *The WordNet in Indian Languages*. Springer Singapore, pp. 209–220.

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

Kak A. A., **Mehdi N. and Lawaye A. A.** (2009). What should be and what should not be? Developing a POS tagset for Kashmiri. *Interdisciplinary Journal of Linguistics (IJL)* **2**, 185–196.

Kaur R., **Sharma R. K.**, **Preet S. and Bhatia P.** (2010). Punjabi WordNet relations and categorization of synsets. In *3rd national workshop on IndoWordNet under the Aegis of the 8th International Conference on Natural Language Processing (ICON 2010)*, Kharagpur, India.

Kharate N. G. and Patil V. H. (2021). Word sense disambiguation for Marathi language using WordNet and the Lesk approach. In *Proceeding of First Doctoral Symposium on Natural Computing Research: DSNCR 2020*, Springer Singapore, pp. 45–54.

Kirillovich A., **Loukachevitch N.**, **Kulaev M.**, **Bolshina A. and Ilvovsky D.** (2022). Sense-annotated corpus for Russian. In *Proceedings of the 5th International Conference on Computational Linguistics in Bulgaria (CLIB 2022)*, Sofia, Bulgaria, pp. 130–136.

Kokane C. D. and Babar S. D. (2019). Supervised word sense disambiguation with recurrent neural network model. *International Journal of Engineering and Advanced Technology* **9**(2), 1447–1453.

Kokane C., **Babar S. and Mahalle P.** (2023). An adaptive algorithm for polysemous words in natural language processing. In *Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems: ICACECS 2022*, Singapore: Springer Nature Singapore, pp. 163–172.

Koul O. N. and Wali K. (2015). *Kashmiri Language Linguistics and Culture*. Central Institute of Indian Langugaes Manasagangotri Mysore.

Kumar P. (2020). Word sense disambiguation for Punjabi language using deep learning techniques. *Neural Computing and Applications* **32**(8), 2963–2973.

Kumari A. and Lobiyal D. K. (2022). Efficient estimation of Hindi WSD with distributed word representation in vector space. *Journal of King Saud University-Computer and Information Sciences* **34**(8), 6092–6103.

Laba Y., **Mudryi V.**, **Chaplynskyi D.**, **Romanyshyn M. and Dobosevych O.** (2023).Contextual embeddings for Ukrainian: A large language model approach to word sense disambiguation. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP),* Dubrovnik, Croatia, pp. 11–19.

Lang S., **Bravo-Marquez F.**, **Beckham C.**, **Hall M. and Frank E.** (2019). Wekadeeplearning4j: A deep learning package for Weka based on deeplearning4j. *Knowledge-Based Systems* **178**, 48–50.

Lawaye. A. A. and Purkayastha S. B. (2014). Kashmir part of speech tagger using CRF. *PARIPEX - Indian Journal of Research* **3**(3), 37–38.

Le M., **Postma M.**, **Urbani J. and Vossen P.** (2018). A deep dive into word sense disambiguation with LSTM. In *Proceedings of the 27th international conference on computational linguistics*, New Mexico, USA, pp. 354–365.

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 annual international conference on Systems documentation*, Toronto Ontario, Canada, pp. 24–26.

Luan Y., **Hauer B.**, **Mou L. and Kondrak G.** (2020). Improving word sense disambiguation with translations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4055–4065.

McHugh M. L. (2012). Interrater reliability: The Kappa statistic. *Biochemia medica*, **22**(3), 276–282.

Mehdi N. and Kak A. A. (2018). Developing a morphological analyser/Generator for Kashmiri. *Interdisciplinary Journal of Linguistics* **11**, 54–66.

Mehdi N. and Lawaye. A. A. (2011). Development of unicode complaint Kashmiri font: Issues and resolution. *Interdisciplinary Journal of Linguistics (IJL)* **4**, 195–200.

Mihalcea R. (2007). Using wikipedia for automatic word sense disambiguation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pp. 196–203.

Mikolov T., **Sutskever I.**, **Chen K.**, **Corrado G. S. and Dean J.** (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems.* NIPS 2013, pp. 1–9.

Miller G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM* **38**(11), 39–41.

Miller G. A., **Chodorow M.**, **Landes S.**, **Leacock C. and Thomas R. G.** (1994). Using a semantic concordance for sense identification. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro*, New Jersey, pp. 240–243.

Mir T. A., **Lawaye A. A.**, **Rana P. and Ahmed G.** (2023). Building kashmiri sense annotated corpus and its Usage in supervised word sense disambiguation. *Indian Journal of Science and Technology* **16**(13), 1021–1029.

Navigli R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys* **41**(2), 1–69.

Ng H. T. and Lee H. B. (1996). Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. *arXiv preprint cmp-lg/9606032.*

Nodehi A. K. and Charkari N. M. (2022). A metaheuristic with a neural surrogate function for word sense disambiguation. *Machine Learning with Applications* **9**, 9–11.

Park J. Y., **Shin H. J. and Lee J. S.** (2022). Word sense disambiguation using clustered sense labels. *Applied Sciences* **12**(4), 1–11.

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

Pasini T., Raganato A. and Navigli R. (2021.XL-WSD:). An extra-large and cross-lingual evaluation framework for word sense disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence* **35**(15), 13648–13656.

Patankar S. N. and Devane S. R. (2017). Issues in resolving word sense disambiguation for multilingual translation framework. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India: IEEE, pp. 230–235.

Patil S. S., Bhavsar R. P. and Pawar B. V. (2023). Bert and indowordnet collaborative embedding for enhanced marathi word sense disambiguation. *ICTACT Journal on Soft Computing* **13**(2), 2842–2849.

Quinlan J. R. (1986). Induction of decision trees. *Machine Learning* **1**(1), 81–106.

Quinlan J. R. (1993). *Programs for Machine Learning*. San Francisco, CA: Morgan Kaufmann.

Rahman N. and Borah B. (2022). An unsupervised method for word sense disambiguation. *Journal of King Saud University-Computer and Information Sciences* **34**(9), 6643–6651.

Rakho M., Laporte E. and Constant M. (2012). A new semantically annotated corpus with syntactic-semantic and cross-lingual senses. In *Language Resources and Evaluation (LREC'12)*, pp. 597–600.

Ramamoorthy L., Choudhary N. and Bhat S. M. (2019). *A Gold Standard Kashmiri Raw Text Corpus*. Central Institute of Indian Languages, Mysore.

Reynolds K., Kontostathis A. and Edwards L. (2011). Using machine learning to detect cyberbullying. In *2011 10th International Conference on Machine learning and applications and workshops,* IEEE, pp. 241–251.

Rouhizadeh H., Shamsfard M. and Tajalli V. (2022). SBU-WSD-corpus: a sense annotated corpus for persian all-words word sense disambiguation. *International Journal of Web Research* **5**(2), 77–85.

Saeed A., Nawab R. M. A., Stevenson M. and Rayson P. (2019). A word sense disambiguation corpus for Urdu. *Language Resources and Evaluation* **53**(3), 397–418.

Sarmah J. and Sarma S. K. (2016). Word sense disambiguation for Assamese. In *2016 IEEE 6th international conference on advanced computing (IACC)*, Bhimavaram: IEEE, pp. 146–151.

Sarma S. K., Medhi R., Gogoi M. and Saikia U. (2010). Foundation and structure of developing an Assamese wordnet, *Proceedings of 5th international conference of the global WordNet Association*, Mumbai, India.

Sarma S. K., Bharali H., Gogoi A., Deka R. and Barman A. (2012). A structured approach for building Assamese corpus: insights, applications and challenges. In *Proceedings of the 10th Workshop on Asian Language Resources*, Mumbai, India, pp. 21–28.

Scarlini B., Pasini T. and Navigli R. (2020). Sense-annotated corpora for word sense disambiguation in multiple languages and domains. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Marseille, France, pp. 5905–5911.

Sert B.Ş., Elma E. and Altinel A. B. (2023). Enhancing the Performance of WSD Task Using Regularized GNNs With Semantic Diffusion. *IEEE Access* **11**, 40565–40578.

Singh S., Siddiqui T. J. and Sharma S. K. (2014). Naïve Bayes classifier for Hindi word sense disambiguation. In *Proceedings of the 7th ACM India Computing Conference*, Nagpur, India, pp. 1–8.

Singh S. and Siddiqui T. J. (2015a). Role of Karaka relations in Hindi word sense disambiguation. *Journal of Information Technology Research* **8**(3), 21–42.

Singh S. and Siddiqui T. J. (2015b). Utilizing corpus statistics for Hindi word sense disambiguation. *International Arab Journal of Information Technology* **12**(6), 755–763.

Singh S. and Siddiqui T. J. (2016). *Sense annotated Hindi corpus*. In *2016 International Conference on Asian Language Processing (IALP),* IEEE, pp. 22–25.

Singh V. P. and Kumar P. (2019). Sense disambiguation for Punjabi language using supervised machine learning techniques. *Sādhanā* **44**(226), 1–15.

Singh V. P. and Kumar P. (2020). Word sense disambiguation for Punjabi language using deep learning techniques. *Neural Computing and Applications* **32**(8), 2963–2973.

Song Y., Ong X. C., Ng H. T. and Lin Q. (2021). Improved word sense disambiguation with enhanced sense representations. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4311–4320.

Sruthi S., Kannan B. and Paul B. (2022). Improved word sense determination in malayalam using latent dirichlet allocation and semantic features. *ACM Transactions on Asian And Low-Resource Language Information Processing* **21**(2), 1–11.

Tenney I., Das D. and Pavlick E. (2019). BERT rediscovers the classical NLP pipeline.arXiv preprint arXiv: 1905.

Taghipour K. and Ng H. T. (2015). One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, Beijing, China, pp. 338–344.

Vial L., Lecouteux B. and Schwab D. (2019). Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation, arXiv preprint arXiv: 1905.

Walia H., Rana A. and Kansal V. (2018). Word sense disambiguation: Supervised program interpretation methodology for Punjabi language. In *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India: IEEE, pp. 762–767.

Walia H., Rana A. and Kansal V. (2020). Comparative analysis of different classifiers for case based model in Punjabi word sense disambiguation. *Investigación Operacional* **41**(2), 273–289.

https://doi.org/10.1017/nlp.2024.31 Published online by Cambridge University Press

Wang J., **Li Y.**, **Huang X.**, **Chen L.**, **Zhang X. and Zhou Y.** (2023). Back deduction based testing for word sense disambiguation ability of machine translation systems. In *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, Seattle WA, USA, pp. 601–613.

Wijeratne S., **Balasuriya L.**, **Sheth A. and Doran D.** (2017). Emojinet: An open service and api for emoji sense discovery. In *Proceedings of the International AAAI Conference on Web and Social Media* **11**(1), 437–441.

Zhang C. X., **Zhang Y. L. and Gao X. Y.** (2023). Multi-head self-attention gated-dilated convolutional neural network for word sense disambiguation. *IEEE Access* **11**, 14202–14210.

Zheng H., **Li L.**, **Dai D.**, **Chen D.**, **Liu T.**, **Sun X. and Liu Y.** (2021). Leveraging word-formation knowledge for Chinese word sense disambiguation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic, pp. 918–923.

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*, Uppsala, Sweden, pp. 78–83.

Cite this article: Mir TA. and Lawaye AA. Word sense disambiguation corpus for Kashmiri. *Natural Language Processing* https://doi.org/10.1017/nlp.2024.31