Merging Verb Senses of Hindi WordNet using Word Embeddings

Sudha Bhingardive   Ratish Puduppully   Dhirendra Singh   Pushpak Bhattacharyya
Department of Computer Science and Engineering,
IIT Bombay, Powai, Mumbai, 400076.
{sudha,ratishp,dhirendra,pb}@cse.iitb.ac.in

Abstract

In this paper, we present an approach for merging fine-grained verb senses of Hindi WordNet. Senses are merged based on gloss similarity score. We explore the use of word embeddings for gloss similarity computation and compare with various WordNet based gloss similarity measures. Our results indicate that word embeddings show significant improvement over Word-Net based measures. Consequently, we observe an increase in accuracy on merging fine-grained senses. Gold standard data constructed for our experiments is made available.

1 Introduction

Hindi WordNet[1] (HWN) is the first Indian language WordNet. It was created manually from Princeton WordNet[2] (Christiane Fellbaum, 1998) using expansion approach and similarly other 16 Indian language WordNets were created from Hindi. This linked structure of Indian language WordNets is known as IndoWordNet[3] (Bhattacharya P., 2010) . It is as shown in Figure 1.

The structure of HWN is similar to the Princeton WordNet. It is composed of synsets and semantic relations. Synset is a set of synonyms representing the same concept. Synsets are linked with basic semantic relations *viz., hypernymy, hyponymy, meronymy, holonymy, troponymy etc*. In comparison with Princeton WordNet, HWN provides extra relations *e.g., gradation, causative,*

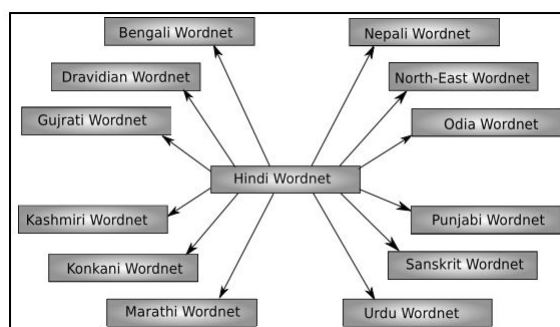compounds, conjunction etc. HWN is widely used

Figure 1: IndoWordNet

in Natural Language Applications (NLP) *viz.*, Machine Translation (Ananthakrishnan et al., 2008; Kunchukuttan et al., 2012), Word Sense Disambiguation (Khapra et al., 2010; Bhingardive et al., 2013), Sentiment Analysis (Balamurali et al., 2012; Popat et al., 2013) *etc*. Over-specified sense distinctions in HWN may not be useful for certain applications. Hence, generating a coarse-grained version of HWN is a crucial task in order to get better results for such applications. In this paper, we present a method for merging the fine-grained senses of HWN using gloss similarity. Word embeddings are used for computing this similarity. The presented method performs better as compared to baselines.

The paper is organised as follows. Section 2 describes the sense granularity that exists in HWN. Section 3 presents the related work. Section 4 gives details about Word Embeddings. Sense merging approach is given in section 5. Experiments and results are presented in section 6. Error analysis is given in section 7. Section 8 concludes the paper and points to the future work.

2 Hindi WordNet Sense Granularity

Different applications need different types of sense granularity. Fine-grained sense distinctions

---

to blow	1	मुँह बहुत थोडा खुला रखकर हवा बाहर निकालना (mumha bahuta thodA khulA rakhakar havA bAhar nikAlanA) **(blow air through a very small opening of mouth)**
	2	मुख से बजाए जाने वाले बाजों को फूँककर बजाना (mukha se bajAye jAne wAle bAjom ko phumkakara bajAnA) **(blowing the instruments that are played by mouth)**
to ignite	3	फूँक मार कर दहकाना या प्रज्वलित करना (phUmka mAra kara dahakAnA yA prajjvalita karanA) **(ignite by blowing)**
	4	आग के संयोग से किसी वस्तु को जलने में प्रवृत्त करना (Aga ke sanyoga se kisI vastu ko jalane mem pravarutt karanA) **(to burn something with fire)**
	5	आग लगाना (Aga lagAnA) **(to burn)**
to smoke	6	तम्बाकू गाँजे आदि का धुआँ मुँह से खींचकर बाहर निकालना (tambAkU, gAnje Adi kA dhumA mumha se khINcakara bAhara nikAlanA) **(to exhale the smoke of tobacco etc after inhaling)**

Table 1: Fine-grained senses of the verb फूँकना (*phUmkanA*); six senses can be merged into three sense groups

are suitable for language learners and applications like Document Categorization, Information Retrieval, Information Extraction *etc.* However, coarse-grained senses are sufficient for applications like Machine Translation and Word Sense Disambiguation. The main difficulty arises in finding the consistent criteria for making accurate sense distinctions.

HWN has many fine-grained senses. For example, there are six senses of word फूँकना (*phumkanA*), which can be merged into three sense groups as shown in Table 1. Hindi senses are distinguished depending on different types of linguistic properties like *properties of subject, object, time variations, compulsion, mode of communication, visibility, acts, existence etc.* Some of them are listed in Table 2 and explained below:

- **Subject property :** Senses can be distinguished depending on the properties of the subject. Consider the word काटना (*kAtanA*) which has two senses S_1 (to cut) and S_2 (insect bite) as shown in Table 2. In S_1, subject will always be an animate entity (a human being) while in S_2, it will always be an insect.

- **Object property :** Object property can also help in making sense distinction. For example, the word रखना (*rakhanA*) has two senses S_1 (to put) and S_2 (to present) as shown in Table 2, in which S_1 can take either animate or inanimate object while S_2 can take only abstract object.

- **Compulsion :** In some cases, senses are distinguished depending on the force of action.

For example, the word निकालना (*nikAlanA*) has two senses S_1 (to remove from a post) and S_2 (forcefully remove from a post) are distinguished by the force of action. Word Sense Disambiguation algorithms often fail in making such fine distinction.

- **Time period :** Consider the senses of word दिन (*dina*). There are total nine senses out of which three senses (ref Table 2) differ only in time period.

Fine grained sense distinctions are very difficult to capture programmatically. Sometimes even humans fail in making such distinctions. Hence, for applications which do not need fine-grained senses, a coarse-grained version of HWN is essential.

3 Related Work

Recently, a large number of sense clustering techniques have been proposed. These techniques rely on various information resources like ontological structure, external corpora, translation similarities, supervision *etc.*

WordNet ontology structure is very helpful for merging fine-grained word senses. Various synset similarity measures have been proposed *viz.,* Path Based Similarity (Wu and Palmer, 1994), (Leacock and Chodorow, 1998), Information Content Based Measures (Resnik, 1995) (Lin, 1998) (Jiang and Conrath, 1997), Gloss Based Heuristics (Lesk, 1986) (Banerjee and Pedersen, 2003) *etc.* These measures were used for creating coarse-grained

Linguistic Properties	Target word	Senses	Gloss/Definition in Hindi WordNet
Subject property (animate/inanimate)	काटना (kAtnA)	S_1	धारदार शस्त्र आदि से किसी वस्तु आदि के दो या कई खंड करना या कोई भाग अलग करना (dhArdAr shastra Adi se kisI vastu Adi ke do yA kaI khanda karnA ya koI bhAg alag karnA) **[cutting into two or more pieces using a sharp instrument]**
		S_2	विषैले कीडो जन्तुओं आदि का दाँत से काटना (vishaile kido, jantu aadi ka dat se kAtnA **[biting with teeth by poisonous insects and creatures]**
Object property (Knowledge/Experience)	रखना (rakhnA)	S_1	स्थित करना (sthit karnA) **[to put]**
		S_2	प्रस्तुत करना (prastuta karnA) **[to present]**
Compulsion	निकालना (nikAlnA)	S_1	स्थान स्वामित्व अधिकार पद आदि से अलग करना (sthAn, swAmitva, adhikAr, pad Adi se alag karnA) **[to remove from a place, ownership, rights, position etc.]**
		S_2	स्थान छोडने पर विवश करना (sthAn chodane ke liye vivash karnA) **[to force to leave a place]**
Time period	दिन (dina)	S_1	सूर्य निकलने से उसके अस्त होने तक का समय (surya nikalne se uske asta hone tak kA samay) **[the time after sunrise and before sunset]**
		S_2	एक सूर्योदय से लेकर दूसरे सूर्योदय तक का समय जो चौबीस घंटे का माना जाता है (ek suryoday se lekar dusre suryoday tak kA samay jo choubis ghante kA maana jAtA hai) **[time between two sunrise which considered as of 24 hours]**
		S_3	चौबीस घंटे में से वह समय जो सोने के बाद काम करने में गुजरता है (choubis ghante me se vaha samay jo sone ke bad kaam karane se gujaratA hai) **[within 24 hours, the time apart from sleeping that is spent working]**

Table 2: Hindi WordNet Sense Distinction

senses. Dolan (1994) first used ontological information for sense clustering. He presented a heuristic based algorithm for clustering senses of Longman's Dictionary of Contemporary English (LDOCE). Peters (1998) addressed different ways for reducing the fine-grainedness of EuroWordNet. In his approach, senses were grouped depending on the semantic relations like *sisters, twins, cousins, autohyponymy etc.*

Mihalcea and Moldovan (2001) derived a set of semantic and probabilistic rules for reducing average polysemy. This was the first attempt of grouping synsets rather than word senses. The resulting version of WordNet leads to reduction of polysemy by around 26% with an error rate of 2.1%. Tomuro (2001) used a similar approach but introduced more principled algorithms.

Agirre and Lacalle (2003) presented a clustering technique which uses confusion matrices, translation similarities, hand-tagged examples of the target word senses and other web information. McCarthy (2006) used combination of word-to-word distributional similarity along with WordNet based similarity measures for sense clustering.

Bhagwani et. al., (2013) proposed a semi-supervised approach which learns synset similarity by using graph based recursive similarity. Resulting coarse-grained sense inventory boosts performance of noun sense disambiguation.

Chugur et. al., (2002) used translational equivalences of word senses for sense merging. Two word senses are expected to be similar, if they lead to the same translation in other languages.

Several sense clustering attempts were made by mapping WordNet to other sense inventories either manually or automatically. Navigli (2006) proposed a sense clustering method by mapping WordNet senses to Oxford English Dictionary (OED). Martha Palmer (2007) suggested a semi-automatic technique for verb sense grouping by using Levin class theory.

Snow et. al., (2007) proposed a supervised approach using Support Vector Machine in which features were derived from WordNet and other

lexical resources.

Due to shallow hierarchy of verbs in WordNet, the knowledge based measures which exploit ontology structure are ineffective for sense merging. We therefore make use of gloss to infer fine-grained senses. We investigate usage of word embeddings for gloss similarity computation.

4 Word Embeddings

Word Embeddings are increasingly being used in variety of NLP tasks. Word Embeddings represent each word with low-dimensional real valued vector. Such models work under the assumption that similar words occur in similar context (Harris, 1968). (Collobert et al., 2011) used word embeddings for POS tagging, Named Entity Recognition and Semantic Role Labeling. Such embeddings have also been used in Sentiment Analysis (Tang et al., 2014), Word Sense Induction (Huang et al., 2012), Dependency Parsing (Bansal et al., 2014) and Constituency Parsing (Socher et al., 2013).

Word embeddings have been used for textual similarity computation (Mihalcea et al., 2006). We are using word embeddings for finding gloss similarity between synsets. The fine-grained senses can be merged based on the similarity values. Word embeddings have been trained using *word2vec*[4] tool (Mikolov et al., 2013). *word2vec* provides two broad techniques for word vectors generation: Continuous SkipGram and Continuous Bag of Words (CBOW). CBOW predicts current word based on surrounding context, whereas Continuous SkipGram model tries to maximize classification of word based on another word in the same sentence (Mikolov et al., 2013). The approach followed here is using SkipGram model by varying context window size (w). Like (Bansal et al., 2014) we find that lower window size results in syntactically similar words. As the window size increases, more semantically similar words are listed. For the experiments we performed, we fixed window size as $w = 7$ as we are interested in more semantically similar words. The word vectors have been trained on 44M sentence corpus (Bojar et al., 2014). The time taken to create word embeddings on the corpus was few minutes on a 2X2 GHz machine.

5 Gloss-based Semantic Similarity Measure used for Sense Merging

Let us consider the following example:

Example: Target Word: डरना (darnA)

- Sense 1: "किसी चीज का डर होना (kisI cheez kA dar honA)" **[to fear of something]**

- Sense 2: "अनिष्ट या हानि की आशंका से आकुल होना (anishta yA hAni kI aAshankA se Akul honA)" **[nervousness due to feeling of loss or premonition]**

Above two senses of word डरना (daranA) are too fine-grained. Lesk similarity (Lesk, 1986) and Extended Lesk similarity (Banerjee and Pedersen, 2003) comes out to be zero as there is no gloss overlap and no relation between these two senses in HWN. Therefore, instead of finding the gloss overlap, the approach followed here is to find whether words from two glosses are semantically related or not.

5.1 Mihalcea Text Similarity using Word Embeddings

We used word embeddings generated using *word2vec* (ref Section 4) for finding the semantic similarity between words from two glosses. We leverage the text similarity measure proposed by (Mihalcea et al., 2006) for gloss similarity computation. It considers both word-to-word similarity and word specificity. Word specificity indicates whether the word is specific or generic. Specificity of a word is measured using Inverse document frequency (idf) (Sparck-Jones et al., 1972). idf is defined as the total number of documents in the corpus divided by the total number of documents including that word. We used hindi wikipedia dump[5] for obtaining idf. Each wikipedia page is treated as single document.

The text similarity measure given in Equation 1 compares two text segments T_1 and T_2 for semantic similarity. For each word w in T_1, it finds the respective word in T_2 with which it has maximum similarity $maxSim(w, T_2)$.

$$sim(T_1, T_2) = \frac{1}{2} * \left(\frac{\sum_{w \in T_1}(maxSim(w, T_2) * idf(w))}{\sum_{w \in T_1} idf(w)} \right.$$
$$\left. + \frac{\sum_{w \in T_2}(maxSim(w, T_1) * idf(w))}{\sum_{w \in T_2} idf(w)} \right)$$
$$(1)$$

---

[4]https://code.google.com/p/word2vec/

[5]http://dumps.wikimedia.org/hiwiki/20140814/

Sense Merging Techniques	Precision	Recall	F-measure
Mihalcea Text Similarity Using Word Embeddings	**0.97**	0.39	0.56
Compositional Text Similarity using Word Embeddings	0.89	**0.43**	**0.58**
Lesk with idf	**0.97**	0.27	0.42
Lesk without idf	0.91	0.14	0.25
Path Similarity	0.89	0.22	0.36
WUP	0.86	0.087	0.16
LCH	0.53	0.24	0.33

Table 3: Sense Merging Results with similarity threshold ≥ 0.7

Sense Merging Techniques	Precision	Recall	F-measure
Mihalcea Text Similarity Using Word Embeddings	0.95	**0.54**	**0.69**
Compositional Text Similarity Using Word Embeddings	0.75	**0.54**	0.63
Lesk with idf	**0.97**	0.29	0.45
Lesk without idf	0.86	0.29	0.44
Path Similarity	0.90	0.24	0.38
WUP	0.82	0.21	0.33
LCH	0.43	0.28	0.34

Table 4: Sense Merging Results with similarity threshold ≥ 0.6

Sense Merging Techniques	Precision	Recall	F-measure
Mihalcea Text Similarity Using Word Embeddings	0.74	0.58	0.65
Compositional Text Similarity Using Word Embeddings	0.67	**0.69**	**0.68**
Lesk with idf	**0.96**	0.36	0.52
Lesk without idf	0.76	0.38	0.51
Path Similarity	0.82	0.27	0.41
WUP	0.61	0.24	0.35
LCH	0.39	0.34	0.36

Table 5: Sense Merging Results with similarity threshold ≥ 0.5

where, $maxSim(w, T_i)$ is computed on word embeddings by finding the maximum cosine similarity between w and words in T_i. The process is repeated for each word in T_2 w.r.t T_1. The similarities are weighted by idf values, summed up and normalized w.r.t to the length of the text segment. Similarity scores obtained are values between 0 and 1, where 0 indicates least similarity and 1 indicates maximum similarity.

5.2 Compositional Text Semantic Similarity Using Word Embeddings

In this approach, we consider the word embedding of the text segment T as compositionally obtained from that of its words. The principle behind the same is that the meaning of the sentence is derived from its constituent words. This is the Weighted

Addition model in (Mitchell and Lapata, 2008). For this system, we construct word embeddings for each text segment as in Equation 2:

$$vec(T_1) = \sum_{w \in T_1} (vec(w) * idf(w)) \quad (2)$$

$$sim(T_1, T_2) = cosine(vec(T_1), vec(T_2)) \quad (3)$$

where $vec(T)$ is the word embedding for text segment T.

6 Experiments and Results

For the purpose of experiments, we created gold standard data. It consists of 250 verbs each with two senses. The test set verbs were tagged as mergeable or not. Five annotators worked independently and created this data with 0.8 inter annotator agrement. This data is released for further

experimentation [6].

We compare our approach with WordNet based gloss similarity measures listed below:

- Lesk with idf: Senses are merged based on the word overlap between glosses (Lesk, 1986) with idf weighting applied on them. For this, we use the Equation 1 with $maxSim$ defined as follows:

$$maxSim(w, T_i) = 1 \quad if\, w \in T_i$$
$$= 0 \quad if\, w \notin T_i$$

- Lesk without idf: In this method, senses are merged based on the word overlap between glosses (Lesk, 1986) without applying idf weighting on them. The following equation is used which is derived from the Equation 1.

$$sim(T_1, T_2) = \frac{1}{2} * \left(\frac{\sum_{w \in T_1}(maxSim(w, T_2))}{count(T_1)} + \frac{\sum_{w \in T_2}(maxSim(w, T_1))}{count(T_2)} \right) \tag{4}$$

where $maxSim$ is as defined in Lesk with idf.

- Path Length Measure: It measures the similarity between two synsets depending on the number of links existing in the is-a hierarchy of WordNet.

This measure is defined as follows:

$$sim_{path} = \frac{1}{shortest\_path\_length(S1, S2)} \tag{5}$$

where $S1$, $S2$ are synsets.

The shorter the length of the path between them, the more related they are considered. Thus there is an inverse relation between the length of the path between the synsets and the similarity between them. This sim_{path} value is substituted into Equation 1.

- The Leacock Chodorow (Leacock and Chodorow, 1998) similarity is determined as:

$$sim_{LCH} = -log\frac{shortest\_path\_length(S1, S2)}{2 * D} \tag{6}$$

where D is the maximum depth of the taxonomy. This sim_{LCH} value is substituted into Equation 1.

---

[6] https://github.com/sudhabh/SenseMerging

- (Wu and Palmer, 1994) similarity metric measures the depth of two given synsets in the WordNet taxonomy, and the depth of the least common subsumer (LCS), and combines these figures into a similarity score:

$$sim_{WUP} = \frac{2 * depth(LCS)}{depth(S_1) + depth(S_2)} \tag{7}$$

This sim_{WUP} value is substituted into Equation 1.

Table 3, Table 4 and Table 5 present Precision, Recall and F-measure for sense merging techniques with similarity threshold as 0.7, 0.6 and 0.5. Here threshold is value above which the two candidate verb senses are considered similar. The similarity values range from 0 to 1. From the results, we observe that decreasing the value of similarity threshold leads to increase in recall with corresponding decrease in precision. Figure 2 and Figure 3 show the variation in F-measure across range of similarity thresholds. From the figures, again we observe that techiques based on Word Embeddings performs much better than techniques based on WordNet similarity measures with regard to F-measure.
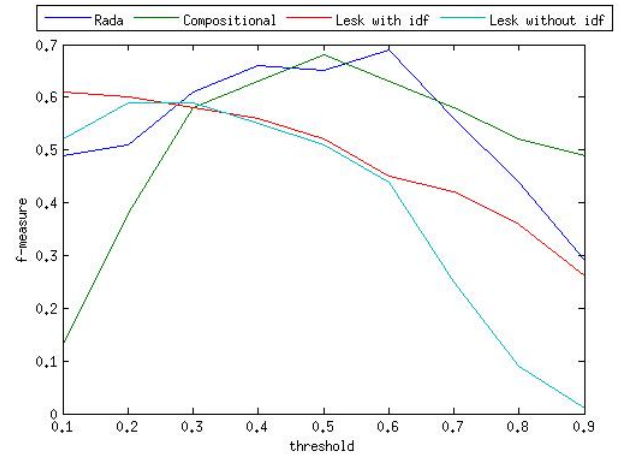

Figure 2: Plot of F-measure of Word embedding based measures (Rada and Compositional) and WordNet similarity based (Lesk with idf and Lesk without idf) figures against various threshold values

7 Error Analysis

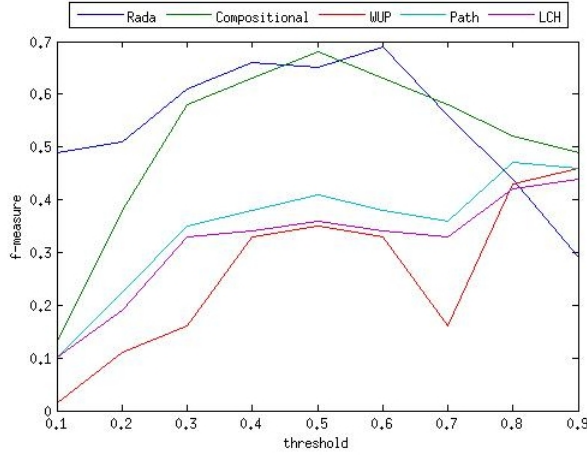Our approach suffers from some limitations listed below.

Figure 3: Plot of f-measure of Word embedding based measures (Rada and Compositional) and Wordnet based measures (WUP, Path and LCH) against various threshold values

1. Sometimes gloss semantic similarity score is very high, even though the word senses are not similar. This leads to an incorrect sense merging. Consider the two senses of the word पहुँचना (pahuchanA) listed below.

   - S_1: {पहुँचना, पहुंचना , फैलना } (pahumchanA, pahunchanA, failenA) - किसी स्थान तक फैलना (kisI sthAn tak failenA) [**to extend upto a place**]

   - S_2: {पहुँचना, पहुंचना} (pahumchanA, pahunchanA) - किसी पद स्थान आदि तक पहुँचना (kisI pad, sthAn aadI tak pahunchanA) [**to reach a position or a place**]

   Senses S_1 and S_2 are not similar, but they have high semantic similarity score resulting in an incorrect sense merging. This might have happened because स्थान ($sthAn$) is common between the two gloss and पहुंचना ($pahunchanA$) is semantically similar to फैलना ($failenA$) in the corpus.

2. Another source of error is disparity in idf values due to multiple ways of expressing Hindi word forms. For example, as seen in S_1, S_2 above, पहुँचना (pahumchanA), पहुंचना (pahunchanA) are two ways of saying the same word. This results in split in their counts and consequent change in idf value.

8 Conclusion and Future Work

We conclude that word embeddings are indeed effective in computing gloss similarity and can be used to merge fine-grained senses of Hindi WordNet. We report significant performance improvement with word embeddings over WordNet based similarity measures. The resulting coarse-grained verb senses of Hindi WordNet are important resources in applications which do not prefer the fine-grained sense distinctions.

In future, we will perform evaluation on verbs having more than two senses. Also, the same technique can be applied for merging senses of other Indian language WordNets. We plan to use the coarse grained senses in both Rule Based Machine Translation and Statistical Machine Translation systems and conduct experiments to verify increase in accuracy of translation. The Weighted Addition model for compositional meaning is agnostic to syntax of the sentence. We plan to explore additional models of representing phrases and sentences such as lexical function model in (Paperno et al., 2014).

References

Agirre E. and Lacalle. 2003. *Clustering wordnet word senses*, In RANLP, volume 260, pages 121-130.

Ananthakrishnan R., Hegde J., Bhattacharyya P. and Sasikumar M. 2008. *Simple Syntactic and Morphological Processing Can Help English-Hindi Statistical Machine Translation*, International Joint Conference on NLP (IJCNLP08), Hyderabad, India.

Balamurali A.R., Joshi A. and Bhattacharyya P. 2012. *Cross-Lingual Sentiment Analysis for Indian Languages using WordNet Synsets*, COLING, Mumbai.

Banerjee S. and Pedersen T. 2003. *Extended Gloss Overlaps as a Measure of Semantic Relatedness*, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico.

Bansal, Mohit and Gimpel, Kevin and Livescu, Karen 2014. *Tailoring Continuous Word Representations for Dependency Parsing*, Proceedings of ACL 2014.

Bhagwani S., Satapathy S. and Karnick H. 2013. *Merging Word Sense*, Association for Computational Linguistics, Proceedings of the TextGraphs-8 Workshop, USA.

Bhattacharya P. 2010. *IndoWordNet*, Lexical Resources Engineering Conference (LREC 2010), Malta.

Bhingardive S., Shaikh S. and Bhattacharyya P. 2013. *Neighbor's Help: Bilingual Unsupervised WSD Using Context*, ACL 2013, Sofia, Bulgaria.

Bojar Ondřej and Vojtěch Diatka and Pavel Rychlý and Pavel Straňák and Vít Suchomel and Aleš Tamchyna and Daniel Zeman 2014. *HindEnCorp - Hindi-English and Hindi-only Corpus for Machine Translation*, Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*, MIT Press.

Chugur I., Gonzalo J. and Verdejo F. 2002. *Polysemy and sense proximity in the senseval-2 test suite*, In Proceedings of the ACL 2002 WSD workshop.

Collobert Ronan , Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa 2011. *Natural language processing (almost) from scratch*, Journal of Machine Learning Research, 12:2493–2537.

Dolan W. 1994. *Word sense ambiguation: clustering related senses*, In Proceedings of the 15th conference on Computational linguistics - Volume 2, COLING,, pages 712-716, Stroudsburg, PA, USA. Association for Computational Linguistics.

Harris. Z. 1968. *Mathematical Structures of Language.*, Wiley, New York.

Huang Eric H. , Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. *Improving Word Representations via Global Context and Multiple Word Prototypes*, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1 Pages 873-882

Jiang J. and Conrath D. 1997. *Semantic similarity based on corpus statistics and lexical taxonomy*, In Proceedings on International Conference on Research in Computational Linguistics.

Khapra M., Shah S., Kedia P. and Bhattacharyya P. 2010. *Domain-Specific Word Sense Disambiguation Combining Corpus Based and WordNet Based Parameters*, 5th International Conference on Global WordNet (GWC 2010), Mumbai.

Kunchukuttan A., Roy S., Patel P., Ladha K., Gupta S., Khapra M. and Bhattacharyya P. 2012. *Experiences in Resource Generation for Machine Translation through Crowdsourcing*, Lexical Resources Engineering Conference (LREC 2012), Istanbul, Turkey.

Leacock C., and Chodorow M. 1998. *Combining local context and WordNet similarity for word sense identification*, In Fellbaum, C., ed., WordNet: An electronic lexical database. MIT Press, pages 265–283.

Lesk M. 1986. *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from and ice cream cone*, In Proceedings of the ACM SIGDOC Conference, pages 24–26, Toronto, Canada.

Lin D. 1998. *An information-theoretic definition of similarity*, In Proceedings of the International Conference on Machine Learning.

McCarthy D. 2006. *Relating WordNet Senses for Word Sense Disambiguation*, In proceedings of ACL Workshop on Making Sense of Sense.

Mihalcea Rada and Moldovan Dan 2001. *Ez.wordnet: principles for automatic generation of a coarse grained wordnet*, In Proceedings of Flairs 2001, pages 454-459.

Mihalcea Rada, Courtney Corley, Carlo Strapparava 2006. *Corpus-based and Knowledge-based Measures of Text Semantic Similarity*, In Proceedings of the American Association for Artificial Intelligence (AAAI 2006), Boston, July 2006.

Mikolov Tomas , Kai Chen, Greg Corrado, and Jeffrey Dean 2013. *Efficient Estimation of Word Representations in Vector Space*, In Proceedings of Workshop at ICLR, 2013.

Mitchell Jeff and Mirella Lapata 2008. *Vector-based models of semantic composition*, In Proceedings of ACL, pages 236–244, Columbus, OH.

Navigli R. 2006. *Meaningful clustering of senses helps boost word sense disambiguation performance*, In Proceedings of COLING-ACL, pages 105-112.

Palmer M., Dang H. and Fellbaum C. 2007. *Making Fine-grained and coarse-grained sense distinctions, both manually and automatically*, Natural Language Engineering.

Paperno Denis and Nghia The Pham and Marco Baroni 2014. *A practical and linguistically-motivated approach to compositional distributional semantics*, Proceedings of ACL 2014.

Pedersen T., Patwardhan S. and Michelizzi J. 2004. *WordNet::Similarity - Measuring the Relatedness of Concepts*, Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04), pages 38-41, Boston, MA.

Peters W., Peters I. and Vossen P. 1998. *Automatic sense clustering in Eurowordnet*, Proceedings of first international conference on language resource and evaluation : Granada, Spain, pages 409-416.

Popat K., Balamurali A., Bhattacharyya P. and Haffari G. 2013. *The Haves and the Have-Nots: Leveraging Unlabelled Corpora for Sentiment Analysis*, ACL 2013, Sofia, Bulgaria.

Resnik P. 1995. *Using information content to evaluate semantic similarity in a taxonomy*, In Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1, IJCAI'95, pages 448-453, San Francisco, CA, USA.

Snow R., Prakash S., Jurafsky D. and Andrew Ng. 2007. *Learning to Merge Word Senses*, In Proceedings of the Joint Meeting of the Conference on Empirical Methods on Natural Language Processing and the Conference on Natural Language Learning.

Socher Richard, John Bauer, Christopher D. Manning and Andrew Y. Ng. 2013. *Parsing With Compositional Vector Grammars*, In Proceedings of the ACL conference. 2013

Sparck-Jones, K. 1972. *A statistical interpretation of term specificity and its application in retrieval.*, Journal of Documentation 28(1):11–21

Tang, Duyu and Wei, Furu and Yang, Nan and Zhou, Ming and Liu, Ting and Qin, Bing 2014. *Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification*, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)

Tomuro M. 2001. *Tree-cut and a lexical based on systematic polysemy*, proc. Of the second meeting of the North America Chapter of the Association for Computational Linguistics, (NAACL).

Wu Z. and Palmer M. 1994. *Verb semantics and lexical selection*, In 32nd Annual Meeting of the Association for Computational Linguistics, pages 133–138.