

# VRep at SemEval-2016 Task 1 and Task 2: A System for Interpretable Semantic Similarity

**Sam Henry** and **Allison Sands**  
Computer Science Department  
Virginia Commonwealth University  
Richmond, VA 23284, USA  
henryst@vcu.edu  
allisonsands1224@gmail.com

## Abstract

VRep is a system designed for SemEval 2016 Task 1 - Semantic Textual Similarity (STS) and Task 2 - Interpretable Semantic Textual Similarity (iSTS). STS quantifies the semantic equivalence between two snippets of text, and iSTS provides a reason why those snippets of text are similar. VRep makes extensive use of WordNet for both STS, where the Vector relatedness measure is used, and for iSTS, where features are extracted to create a learned rule-based classifier. This paper outlines the VRep algorithm, provides results from the 2016 SemEval competition, and analyzes the performance contributions of the system components.

## 1 Introduction

VRep competed in SemEval 2016 Task 1 - Semantic Textual Similarity (STS) and Task 2 - Semantic Interpretable Textual Similarity (iSTS). Both of these tasks compute STS between two fragments of text. Task 2 expands upon Task 1 by requiring a reason for their similarity. VRep uses an STS measure based on the Vector relatedness measure (Pedersen et al., 2004), and a reasoning system based on JRIP (Cohen, 1995), an implementation of the iREP algorithm.

For Task 1, we are provided with paired sentences, and for each pair of sentences VRep assigns a number indicating their STS. The number ranges from 0 to 5, 0 indicating no similarity and 5 indicating equivalence.

For Task 2, we are provided with paired sentences and align the chunks of one sentence to the most

similar chunks in the other sentence. Next, a reason and score are computed for that alignment. A chunk is a fragment of text that conveys a single meaning such as in the following example for which chunks are bracketed.

[ Black Cow ]                    [ walking ]    [ in a pasture ]  
[ Black and white Cow ]    [ sitting ]    [ in the grass ]

Alignment reasons are selected from a small list of possible labels created by the event organizers (Agirre et al., 2015):

1. Equivalent (EQUI) - the two chunks convey an equivalent meaning (“hot water”, “scalding water”)
2. Opposite (OPPO) - the two chunks convey an opposite meaning (“hot water”, “cold water”)
3. More General (SPE1) - this chunk conveys a more general meaning than the other chunk (“hot water”, “water”)
4. More Specific (SPE2) - this chunk conveys a more specific meaning than the other chunk “water”, “hot water”)
5. Similar (SIMI) - the two chunks convey a similar meaning (“sip water”, “gulp water”)
6. Related (REL) - the two chunks are somehow related (“boil water”, “ocean water”)
7. No Alignment (NOALI) - there are no chunks in the other sentence that are semantically similar to this chunk

As in Task 1 the scores range from 0 to 5, 0 indicating no similarity and 5 indicating equivalence. VRep makes extensive use of WordNet (Fellbaum, 2005) to compute STS and assign a label in iSTS. Vrep is written in Perl and is freely available for download<sup>1</sup>.

## 2 Algorithm Description

The same measure of STS is used for both Task 1 and Task 2; however, the algorithm for Task 1 is simpler and consists of only the first two steps: Preprocessing and Semantic Textual Similarity. The steps are outlined below and are expanded on in subsequent subsections.

1. Preprocessing - text is standardized
2. Semantic Textual Similarity - the STS between two chunks or two sentences is computed. This is the final step for Task 1.
3. Chunk Alignment - align each chunk of one sentence to a chunk in another sentence. If no chunks are similar then no alignment (NOALI) is assigned.
4. Alignment Reasoning - assign a label to each aligned chunk pair
5. Alignment Scoring - assign an alignment score on a 0-5 scale

### 2.1 Preprocessing

In the first step, data is prepared for processing as outlined below:

1. Tokenization - spaces are used as a delimiter
2. Lowercase All Characters - standardizes string equivalence testing and prevents incorrect part of speech (POS) tagging. The POS tagger tends to tag most words that have a capital letter as a proper noun which is often incorrect. This is particularly problematic with the headlines data set.
3. Punctuation Removal - standardizes string equivalence testing

<sup>1</sup><http://www.people.vcu.edu/~henryst/>

4. POS tagging - Lingua::EN::Tagger<sup>2</sup> is used. POS tags are used for stop word removal and for alignment reasoning.
5. Stop Word Removal - remove any words that are not tagged as a noun, verb, adjective, or adverb. This reduces chunks and sentences to content words.

### 2.2 Semantic Textual Similarity (STS)

STS is computed in the same way for both tasks; however it is computed between two sentences for Task 1 and between two chunks for Task 2. While describing the computation of STS we refer to chunks; for Task 1 a sentence can be conceptualized as a chunk. VRep's STS computation is shown in Equation (1) and is similar to the method described by NeRoSim (Banjade et al., 2015) and Stefanescu (Ștefănescu et al., 2014). *chunkSim* takes two chunks ( $c_1, c_2$ ) as input and computes the weighted sum of maximum word to word similarities,  $sim(w_i, w_j)$ . To do this, the  $sim(w_i, w_j)$  is found for each word in  $c_2$  against  $c_1$ , and the maximum is added to a running sum.

$$chunkSim(c_1, c_2) = \frac{\sum_{i=1}^n \max_{j=1}^m sim(w_i, w_j)}{\min(n, m)} \quad (1)$$

where  $c_1$  and  $c_2$  are two chunks,  $n$  and  $m$  are the number of words in  $c_1$  and  $c_2$ ,  $w_i$  is word  $i$  of  $c_1$ ,  $w_j$  is word  $j$  of  $c_2$

$sim(w_i, w_j)$  is defined differently for words in WordNet and words not in WordNet. For words in WordNet,  $sim(w_i, w_j)$  is the Vector relatedness measure<sup>3</sup> (Pedersen et al., 2004) with a threshold applied. The Vector measure was chosen for several reasons. Firstly it returns values scaled between 0 and 1 which is beneficial for applying thresholds in both chunk alignment and alignment reasoning. A known scale also allows for a direct mapping from the weighted sum to the answer space of Task 1 (scaled 0-5). Secondly the Vector measure works well when  $w_i$  and  $w_j$  are different parts of speech because it does not rely on WordNet hierarchies. When calculating  $sim(w_i, w_j)$  all possible senses of both  $w_i$  and  $w_j$  are used, and  $sim(w_i, w_j)$  is chosen as

<sup>2</sup><http://search.cpan.org/~acoburn/Lingua-EN-Tagger/>

<sup>3</sup>WordNet::Similarity::Vector

the maximum value. This eliminates the need for word sense disambiguation (WSD). After computing the measure, a threshold is applied that reduces any value less than 0.9 to 0.0. This value was tuned separately using the training data for both tasks via a grid search and 0.9 was found to be optimum for both. The threshold prevents dissimilar terms from impacting the STS which improves the accuracy and prevents noisy chunk alignments.

For words not in WordNet,  $chunkSim(w_i, w_j)$  is a binary value: 1 if all the characters in both words match, 0 otherwise. Words not in WordNet tend to be proper nouns, abbreviations, or short words such as “he” or “she”, “is” or “in”, all of which are generally spelled identically making this a suitable measure.

$chunkSim$  is defined as the sum of maximum word to word similarities normalized by the number of words in the shorter of the chunk pair. Normalization prevents similarity scores from increasing as chunk length increases. It also scales  $chunkSim$  within a predictable range of about 0.0 – 1.0.

$chunkSim$  is used directly in Task 1 where it is linearly scaled by 5 to produce final output. We experimented with multiple regression fits (linear, exponential, logarithmic, power, and polynomial) between our  $chunkSim$  output and the provided gold standard values with little to no improvement, so the linear scaling of 5 was chosen for simplicity.

### 2.3 Chunk Alignment

$chunkSim$  is computed between each chunk of two aligned sentences and the chunk with the highest  $chunkSim$  is selected for alignment. Multiple alignments are allowed for a single chunk. If all chunks have a similarity of 0, no alignment (NOALI) is assigned. Due to the high  $sim(w_i, w_j)$  threshold, no threshold is required for  $chunkSim$  as with NeRoSim (Banjade et al., 2015).

### 2.4 Alignment Reasoning

Alignment Reasoning takes as input a chunk pair and provides a reason why that chunk pair is aligned. VRep’s alignment reasoning is inspired by NeRoSim (Banjade et al., 2015), and SVCTSTS (Karumuri et al., 2015). Both these systems classify a chunk pair using features extracted from the chunk pair itself. NeRoSim’s features tend to focus

more on the semantic relationship between chunk pairs, such as whether or not the two chunks contain antonyms, synonyms, etc. The features of SVCSTS focus more on the syntactic form of the chunks, such as the number of words or counts of parts of speech in a chunk pair. VRep combines the two approaches and extracts a total of 72 syntactic and semantic features for each chunk pair.

Gold Standard chunk pairs of the SemEval 2015 Task 2 Test Data<sup>4</sup> were used to train our classifier, WEKA’s (Hall et al., 2009) JRIP algorithm (Cohen, 1995) which creates a decision list for classification. The classifier uses only 24 of original 72 features and a series of 10 rules.

JRIP was chosen as a classifier due to its performance (see Table 5), and its concision. The rules generated are human readable which provides insight into how the classification occurs and the types of features that are discriminative. Classifiers were trained with chunk pairs from every data set (student answers, headlines, and images), both individually and combined. The best performing classifier for each topic was generated from the combined data. The set of features used and classification rules are shown below.  $\alpha$  and  $\beta$  designate the individual chunks in the chunk pair being classified, and  $\vec{x}_i$  indicates a feature vector created from a chunk pair.  $i$  indicates the feature number in the feature list below.

#### Features used in Classification:

- 1 - unmatched content word percentage of  $\alpha$
- 2 - unmatched content word percentage of  $\beta$
- 3 - 1 if  $\alpha$  contains a location
- 4 - 1 if  $\beta$  contains a location
- 5 - 1 if  $\alpha$  has a verb
- 6 - 1 if  $\alpha$  has an adjective
- 7 - 1 if  $\alpha$  has an adverb
- 8 - 1 if  $\alpha$  and  $\beta$  contain antonyms
- 9 - 1 if  $\alpha$  and  $\beta$  have a equivalent nouns
- 10 - 1 if  $\alpha$  and  $\beta$  contain numeric quantities
- 11 - number of words in  $\alpha$  (before stop word removal)
- 12 - number of words in  $\beta$  (before stop word removal)
- 13 - ratio of the number content words to all words in  $\alpha$  (before stop word removal)

<sup>4</sup>[http://alt.qcri.org/semeval2016/task2/data/uploads/train\\_2015\\_10\\_22.utf-8.tar.gz](http://alt.qcri.org/semeval2016/task2/data/uploads/train_2015_10_22.utf-8.tar.gz)

- 14 - difference in number of content words in  $\alpha$  and  $\beta$
- 15 - difference in number of words in  $\alpha$  and  $\beta$  (before stop word removal)
- 16 - absolute value of the difference in number of content words in  $\alpha$  and  $\beta$
- 17 - absolute value of the difference in number of words in  $\alpha$  and  $\beta$  (before stop word removal)
- 18 - ratio of the number of content words in  $\alpha$  to all words in  $\alpha$  (before stop word removal) over the ratio of the number of content words in  $\beta$  to all words in  $\beta$  (before stop word removal)
- 19 - number of nouns in  $\beta$
- 20 - ratio of the number of nouns in  $\alpha$  to the number of content words in  $\alpha$
- 21 - ratio of the number of verbs in  $\alpha$  to the number of content words in  $\alpha$
- 22 - ratio of the number of adjectives in  $\alpha$  to the number of content words in  $\alpha$
- 23 - ratio of the number of conjunctions in  $\alpha$  to the number of content words in  $\alpha$
- 24 - difference in the number of verbs in  $\alpha$  and  $\beta$

---

**Algorithm 1** Alignment Reasoning Rules
 

---

- 1: **if**  $\vec{x}_8 \geq 1$  &  $\vec{x}_{22} \geq 1$
  - 2: **or**  $\vec{x}_8 \geq 1$  &  $\vec{x}_4 \geq 1$  &  $\vec{x}_5 \geq 1$  &  $\vec{x}_{16} \geq 1$
  - 3: **or**  $\vec{x}_8 \geq 1$  &  $\vec{x}_{23} \geq \frac{1}{2}$  &  $\vec{x}_1 \geq \frac{1}{2}$
  - 4: **or**  $\vec{x}_{21} \geq \frac{1}{2}$  &  $\vec{x}_7 \geq 1$  &  $\vec{x}_{12} \leq 1$   
**return** OPPO
  - 5: **if**  $\vec{x}_2 \geq \frac{1}{3}$  &  $\vec{x}_{14} \leq -1$  &  $\vec{x}_9 \geq 1$  &  $\vec{x}_{19} \leq 0$   
 &  $\vec{x}_{13} \geq \frac{4}{7}$   
**return** SPE2
  - 6: **if**  $\vec{x}_{14} \geq 1$  &  $\vec{x}_9 \geq 1$  &  $\vec{x}_2 \leq 0$  &  $\vec{x}_6 \geq 1$   
 &  $\vec{x}_{11} \leq 4$  &  $\vec{x}_{24} \leq 0$
  - 7: **or**  $\vec{x}_{14} \geq 1$  &  $\vec{x}_{17} \geq 2$  &  $\vec{x}_{13} \geq \frac{5}{7}$
  - 8: **or**  $\vec{x}_{15} \geq 1$  &  $\vec{x}_{18} \geq \frac{8}{9}$  &  $\vec{x}_2 \leq 0$  &  $\vec{x}_3 \geq 1$
  - 9: **or**  $\frac{1}{4} \leq \vec{x}_1 \leq \frac{2}{3}$  &  $\vec{x}_{14} \geq 1$  &  $\vec{x}_5 \leq 0$   
 &  $\vec{x}_{20} \leq \frac{1}{5}$  &  $\vec{x}_{18} \leq 1$  &  $\vec{x}_{13} \geq \frac{2}{3}$   
**return** SPE1
  - 10: **if**  $\vec{x}_2 \geq \frac{1}{2}$  &  $\vec{x}_1 \geq \frac{1}{3}$  &  $\vec{x}_{10} \geq 1$   
**return** SIMI
  - 11: **return** EQUI
- 

It is interesting to note that there is no classifier for the REL class. The data set was heavily skewed towards the EQUI class which consisted of 60% of the total data, leaving a small percentage to be divided among the remaining 5 classes, with just around 5% being REL. With a larger training set we would ex-

pect a classifier for REL to be generated.

## 2.5 Alignment Scoring

Alignment scores are assigned as either the required scores, 0 for NOALI and 5 or EQUI, or the average alignment score for each class as in (Karumuri et al., 2015). The average alignment score for classes were computed both for each topic alone and for all topics combined. The best performing set of scores came for all topics, came from the images data set alone. Scores used for each class are as follows: EQUI = 5.00, OPPO = 4.00, SPE1 = 3.24, SPE2 = 3.69, SIMI = 2.975, REL = 3.00, NOALI = 0.00.

## 3 Results

The performance of VRep is shown below for SemEval 2016 Task 1 and Task 2 test data sets. The baseline described by the task organizers (Agirre et al., 2015) is shown for comparison for Task 2. Baseline results were not made available for Task 1.

### 3.1 Task 1 - Semantic Similarity

For Task 1 the Pearson Correlation Coefficient between VRep’s results and Gold Standard results are reported for the 2016 Task 1 Test Data<sup>5</sup>. A value of 1.0 indicates perfect correlation, 0.0 indicates no correlation. We ran VRep on five data sets with the results of each data set shown in Table 1. More details on the data sets and evaluation metrics are described in the competition summary<sup>6</sup>.

Data set	VRep
answers-answers	0.29487
headlines	0.68185
plagiarism	0.69730
post editing	0.72966
question-question	0.49029
<b>Mean</b>	0.578794

**Table 1:** Results of VRep on SemEval 2016 Task 1 Test Data

### 3.2 Task 2 - Interpretable Semantic Similarity

For Task 2, we report results for the Gold Chunks scenario (data is pre-chunked). Each data set is evaluated using the F1 score in four categories:

<sup>5</sup><http://alt.qcri.org/semeval2016/task1/data/uploads/sts2016-english-with-gs-v1.0.zip>

<sup>6</sup><http://alt.qcri.org/semeval2016/task1/>

**(Ali)** - Alignment - F1 score of the chunk alignment

**(Type)** - Alignment Type - F1 score of the alignment reasoning

**(Score)** - Alignment Scoring - F1 score of alignment scoring

**(Typ+Score)** - Alignment Type and Score - a combined F1 score of alignment reasoning and scoring

F1 scores range from 0.0 to 1.0 with 1.0 being the best score. Data sets are available online<sup>7</sup> and evaluation metrics are described in more detail in the competition summary (Agirre et al., 2015).

Data set	Baseline	VRep
<b>Answers-Students</b>		
F1 Ali	0.8203	0.7723
F1 Type	0.5566	0.5249
F1 Score	0.7464	0.7014
F1 Typ+Scor	0.5566	0.5226
<b>Headlines</b>		
F1 Ali	0.8462	0.8908
F1 Type	0.5462	0.6015
F1 Score	0.7610	0.8027
F1 Typ+Scor	0.5461	0.5964
<b>Images</b>		
F1 Ali	0.8556	0.8539
F1 Type	0.4799	0.5516
F1 Score	0.7456	0.7651
F1 Typ+Scor	0.4799	0.5478

**Table 2:** Results of VRep on SemEval 2016 Task 2 Test Data

## 4 Component Analysis

In this section the contributions of system components and possible additions are evaluated. VRep can be split into two major components, STS and Alignment Reasoning, both of which have different evaluation criteria. Data used for this section comes from the SemEval 2015 Task 1<sup>8</sup> training data and SemEval 2016 Task 2<sup>9</sup> training data. Two-Tailed p-values are shown in the Tables 3 and 4.

<sup>7</sup>[http://alt.qcri.org/semeval2016/task2/data/uploads/test\\_goldstandard.tar.gz](http://alt.qcri.org/semeval2016/task2/data/uploads/test_goldstandard.tar.gz)

<sup>8</sup><http://ixa2.si.edu.es/stswiki/images/2/21/STS2015-en-rawdata-scripts.zip>

<sup>9</sup>[http://alt.qcri.org/semeval2016/task2/data/uploads/train\\_2015\\_10\\_22.utf-8.tar.gz](http://alt.qcri.org/semeval2016/task2/data/uploads/train_2015_10_22.utf-8.tar.gz)

$$LevenshteinMeasure = \begin{cases} \frac{\beta - \delta}{\beta} & \delta < \beta \\ 0 & \delta \geq \beta \end{cases} \quad (2)$$

where  $\delta$  is the Levenshtein distance between the two words, and  $\beta$  is the threshold used

### 4.1 STS Component Analysis

Pearson Correlation Coefficients of STS scores of Task 1 and the F1 Ali Task 2 are used as evaluation metrics for the STS portion of VRep. Tables 3 and 4 show the effects of adding a component to the *Basic* system. Each component and the *Basic* system are described below:

1. As a baseline a *Basic* system which only applies Equation (1) is used. For Task 1 the result is scaled by 5. For Task 2 each chunk is aligned with the chunk with the highest *chunkSim*. No thresholding, or preprocessing is performed.
2. *Threshold* adds a threshold to  $sim(w_i, w_j)$  in Equation (1). A modest threshold of 0.4 was used. The optimum threshold of 0.9 used in the final system was found with the system as a whole. We did not perform a grid search to optimize the threshold for all component tests.
3. *Stop Removal* adds stop word removal as described in subsection 2.1.
4. *Levenshtein* modifies  $sim(w_i, w_j)$  for words not in WordNet. Rather than using a binary value for exact string matching the Levenshtein measure shown in Equation (2) is used. This allows for slight differences in spelling, plurality, tenses, etc. The measure requires a threshold parameter,  $\beta$  which limits the maximum Levenshtein distance ( $\delta$ ) and scales the Levenshtein Measure between 0.0 and 1.0.  $\beta = 2.0$  was found via a grid search to perform best. The Levenshtein Measure is unnecessary for these tasks most likely because, as stated in section 1, words not in WordNet tend to be proper nouns or abbreviations for which the spelling is the same, and for short words such as “he” or “she”, “is” or “in”, even small edit distances can transform the word into a completely unrelated word.

	F1 Ali	p-value	significant
<b>Basic</b>	65.40	-	-
<b>Threshold</b>	0.7263	<0.0001	yes
<b>Stop Removal</b>	0.7349	<0.0001	yes
<b>Levenshtein</b>	0.6586	0.1445	no

**Table 3:** The effects of additional components to the core VRep system on Task 1

	F1 Ali	p-value	significant
<b>Basic</b>	0.7508	-	-
<b>Threshold</b>	0.8812	<0.0001	yes
<b>Stop Removal</b>	0.7511	0.9920	no
<b>Levenshtein</b>	0.7524	0.7795	no
<b>WSD</b>	0.7799	<0.0001	yes
<b>Threshold + WSD</b>	0.8216	<0.0001	yes

**Table 4:** The effects of components to the core VRep system on Task 2

- Word Sense Disambiguation (*WSD*) should help to reduce noisy alignments by using the correct synset when computing the Vector relatedness measure. We used the entire sentence (all chunks) as input to SenseRelate::AllWords (Patwardhan et al., 2003). *WSD* improves results when used as a single component, but when used in combination with a threshold (*Threshold + WSD*) results are worse than a threshold alone. This is likely due to the fact that both *WSD* and thresholding aim to reduce noisy STS and chunk alignments. When used singularly they both achieve this task, but in combination *WSD* errors reduce performance.

Analysis of the test data indicated that the addition of these extra components was unnecessary, however to further analyze their contributions three runs were submitted for the both tasks 1 and 2. Run 1 used the *basic* system, run 2 eliminated the stop removal preprocessing step, and run 3 used the *basic* system with the *Levenshtein* measure described above. Test results were mixed and data set dependant, see the respective competition summaries (Agirre et al., 2016) for complete results.

#### 4.2 Alignment Reasoning Component Analysis

For alignment reasoning, only the assignment of a label (Type) to a chunk pair is evaluated. We used the gold standard alignments provided for each data set, converted each gold standard chunk pair to the

	Answers	Headlines	Images
<b>Baseline</b>	67.9	61.4	54.2
<b>Naive Bayes</b>	23.9	41.5	35.1
<b>Bayes Net</b>	49.3	58.9	51.3
<b>SMO</b>	69.6	65.7	54.9
<b>Decision Table</b>	67.5	65.7	55.8
<b>J48</b>	66.1	64.9	52.7
<b>Random Forest</b>	68.4	65.8	53.6
<b>JRip</b>	68.9	65.4	56.2

**Table 5:** The performance of different classifiers on alignment reasoning

entire set of 72 features and tested multiple classification algorithms. All classifiers are WEKA (Hall et al., 2009) implementations; results are shown in Table 5. The baseline score is calculated as simply assigning the most common class, EQUI.

## 5 Conclusions and Future Work

In future iterations, more analysis should be done to refine the features used in classification. Using JRIP and other analysis criteria we can see why certain features are discriminative, and develop more informative features.

Rather than relying solely on the *Levenshtein* measure for words outside of WordNet, additional metrics, such as word2vec (Mikolov et al., 2013) could be incorporated.

Additional data should be added for training classifiers. The top performing classifier was generated from all data combined indicating that additional samples are necessary. It is likely that given more data, topic specific classifiers will outperform the general classifier we evaluated. Additional data will also help to reduce the class imbalance and will likely result in a set of rules for the REL class.

Since VRep already makes use of WordNet, it could be easily expanded to compete in the polarity subtask by implementing a polarity classifier using SentiWordNet (Baccianella et al., 2010).

## 6 Acknowledgments

We would like to thank Dr. Bridget McInnes for her advice during the development of VRep, and Uzair Abbasi for his contributions to the initial development of the system.

## References

- Eneko Agirre, Carmen Banea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, June.
- Eneko Agirre, Aitor Gonzalez-Agirre, Inigo Lopez-Gazpio, Montse Maritxalar, German Rigau, and Larraitz Uribe. 2016. Semeval-2016 task 2: Interpretable semantic textual similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June.
- Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.
- Rajendra Banjade, Nobal B Niraula, Nabin Maharjan, Vasile Rus, Dan Stefanescu, Mihai Lintean, and Dipesh Gautam. 2015. Nerosim: A system for measuring and interpreting semantic textual similarity.
- William W. Cohen. 1995. Fast effective rule induction. In *Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann.
- Christiane Fellbaum. 2005. Wordnet and wordnets.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.
- Sakethram Karumuri, Viswanadh Kumar Reddy Vuggumudi, and Sai Charan Raj Chitirala. 2015. Umduluthblueteam: Svcsts-a multilingual and chunk level semantic similarity system. *SemEval-2015*, page 107.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Computational linguistics and intelligent text processing*, pages 241–257. Springer.
- Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity: measuring the relatedness of concepts. In *Demonstration papers at hlt-naacl 2004*, pages 38–41. Association for Computational Linguistics.
- Dan Ștefănescu, Rajendra Banjade, and Vasile Rus. 2014. A sentence similarity method based on chunking and information content. In *Computational Linguistics and Intelligent Text Processing*, pages 442–453. Springer.