UWB-NLP at SemEval-2026 Task 5: Synsets and their contexts

Stephen Taylor
University of West Bohemia at Pilsen
taylor@ntis.zcu.cz

Abstract

SemEval 2026 task 5 asks us to provide a program to try to match the human ratings of sense-appropriateness of a particular word in a series of very structured, very short stories.

Our system[1] associates a fixed list of 50 words with each WordNet synset, and computes several scores for each of the phrases in the story, to determine how closely the phrase matches the wordlist.

We received near-chance results, in spite of several different approaches to building and employing sets of word-lists. The stories in this dataset are designed to be ambiguous, and every story contains words associated with at least two senses of the target word. We now believe that our system's approach is inappropriate for this dataset.

1 Introduction

It seems to be a feature of human languages that many of the most common words can be used in very different ways. The logo on the CodaBench site for this task is a picture of upright (that is, head up) bat (that is, one of those flying mammals) holding a scaled-down replica of a baseball bat (that is, the stout wooden club used in the game of baseball). *Bat* is an example of an English word with several distinct definitions. We call each of these ways of using the word a *sense.* Sometimes we can tease out ways in which the senses might be historically related; for example, the sense of *bat* in the phrase 'she batted her eyelashes' [*flutter* has a sense which is a synonym] might be related to the way bats fly. This conjecture fits into the idea that primordial sense of the word *bat* might be a verb, meaning 'to strike'.

In spite of these tentative relationships, we find it useful to place the different definitions as separate entries in dictionaries. One computer resource for the senses of English words is WordNet (Miller, 1995; Fellbaum, 1998), which covers many nouns, verbs, adjectives, and adverbs. Wordnet has an easy Python API(Bird et al., 2009). In WordNet we refer to a word as a lemma, (one of the senses of *lemma* is 'a dictionary headword'), and we refer to each sense as a *synset*, which is a word made up by the WordNet developers, based on the idea that each synset has a set of synonyms, which are the lemmas which have this word as one of their senses, and that humans can easily compute this intersection. Each synset has a name, part of speech and a definition, and may participate in relations with other synsets. For example, `bat.n.01` is the flying mammal; `bat.v.02` is the verb for batting eyelashes, `bat.n.05` is a baseball bat. So the `.lemmas()` of `bat.v.02` are *bat* and *flutter*. The `hypernym` for `bat.n.05` is `club.n.03`.

The current version of English WordNet is 3.0. There are similar resources for other languages as well. A nice thing about WordNet is that a great deal of thought has gone into building it. A problem about WordNet is that it is descriptive, but not statistical. That is, none of the features it describes have measurements associated with them.

Obtaining such measurements has been a continuing project over the last thirty years. We report below on (Agirre et al., 2014), (Goikoetxea et al., 2015), and (Pilehvar and Collier, 2016) who use the Personalized PageRank algorithm (Haveliwala, 2002) on graphs built upon the relations between synsets given by WordNet to induce concrete probability estimates for the occurence of particular synsets in the context of English texts.

2 Data

2.1 Training and Test Files

SemEval 2026 task 5 asks us to provide a program to try to match the human ratings of sense-

[1] Our current software can be found at git@github.com:StephenETaylor/SE26-5.git

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 748–754

appropriateness of a particular word in a series of short stories. The development of the data is described in Gehring and Roth (2025). The task is described in Gehring et al. (2026)

Each story is accompanied by a homonym and an intended sense. The story consists of three sentences of precontext, which is supposed to establish a likely meaning, but does not use the test homonym; the usage sentence which always contains the homonym but is intended to be neutral, and an optional trailing sentence which may reinforce the precontext, reinforce a different sense, or be neutral.

"homonym":	"potential",
"judged_meaning":	"the difference in electrical charge between two points in a circuit expressed in volts",
"precontext":	"The old machine hummed in the corner of the workshop. Clara examined its dusty dials with a furrowed brow. She wondered if it could be brought back to life.",
"sentence":	"The potential couldn't be measured.",
"ending":	"She collected a battery reader and looked on earnestly, willing some life back into the old machine.",
"choices":	[4, 5, 2, 3, 1],
"average":	3.0,
"stdev":	1.5811388300841898,
"nonsensical":	[false, false, false, false, false],
"sample_id":	"1843",
"example_sentence":	"The circuit has a high potential difference."

Figure 1: A sample data record.

It seems logical that the ending sentence, if present, should be the fresher news for the reader, and might change the score given by human evaluators. It seems also possible that the human evaluator might be put off or delighted by the incongruity of the trailing sentence.

The data files are in `.json` format, (see Figure 1, which omits the record key, and the enclosing curly brackets for a python dictionary.) The `train.json` file has 2280 records; the `dev.json` has 588 records, and the `test.json` file contains 930.

Most stories occur in 6 records. Half of these use one definition, and half of them another; the three `precontext` sentences and the `sentence`-sentence, which contains the test word or `homonym` are common to all versions, and the ending is either blank, or one of two story-appropriate phrases. The `judged_meaning` is (except for punctuation) the definition of some WordNet synset. An example sentence (provided by WordNet) illustrates the homonym might be used in the indicated sense.

In the story in Figure 1, we see some other features of the training records. The integer scores provided by 5 human evaluators are provided, as well as an average and standard deviation for those scores. In addition we can see some logical problems with the story, partly from the fragmentation: *If the machine isn't running, why is it humming? A 'battery reader' (whatever that is) seems to relate slightly to voltage differences, and the presence of dials also suggests that the machine is electrical, but the phrase "the potential couldn't be measured" is a common saying, which uses the non-electrical meaning for 'potential'.* It's a bit of a mystery how the average rating for this meaning came out as high as '3.0' – neutral.

3 Related work

Rubenstein and Goodenough (1965) is an empirical study of similarity of context for synonyms; 65 pairs of words were sorted for similarity by paid students and the results were highly correlated. Based on the these results, sentences containing both words of a pair were produced by independent subjects, and the contexts compared. Contexts for highly related synonyms tended to include words related to the synonyms, but the reliably related context fell off quickly; words adjacent to the synonyms were judged related about 25% of the time.

In Patwardhan and Pedersen (2006) the authors develop 'gloss vectors' for comparing 'concepts' – by which they mean WordNet synsets in this case, but *could* mean something else, like Wikipedia articles. They call the gloss vectors 'secondary context vectors'. The first-order context is the gloss

of the synset, plus the glosses of all the words with Wordnet connections to it, ie hypernym, etc. The second-order vector is the built in vocabulary space of a particular corpus, and the chosen corpus for this paper is the concatenated definitions of all WordNet synsets. The vocabulary space (for WN 2.0?) was about 20K words, so the 'context vector' is the vector of counts of the number of times each vocabulary word in the corpus occurs in the context of a 1st-order context word.

Since these vectors are rather ungainly, they are compared with vector cosine.

The references are largely related to the specific problem of comparing synsets.

Agirre et al. (2014) summarizes work on two earlier papers in which Agirre is co-author. Random Walks refers to the idea that the Page Rank algorithm provides a summary of probabilities which could also be obtained by a large number of random walks.

The idea is to

1) build a complete network of the connection of all the words, senses, and synsets in Wordnet. (This process is described on page 63) Their original work used the added relations in the Extended Wordnet (Harabagiu and Moldovan, 1998; Mihalcea and Moldovan, 2001) and they added the WordNet3.0 gloss relation when it became available.

2) [modify the graph so that each context word in a particular instance is providing probability mass to the rest of the graph]

3) run the Personalized Page Rank algorithm to decide where the probability ends up

4) Choose among the synset alternatives for the particular instance the one which receives the most probability.

And it works pretty well. They note that skipping step (3), that is, providing no context, should approximate always choosing the most frequent sense, and show in table 8 that this scheme gives an 81.6 correlation with MFS, that is annotating each word with its most frequent sense, without considering the context.

Goikoetxea et al. (2015) builds on Agirre et al. (2014), building a non-directed graph with synsets as nodes, for edges, WordNet relations between synsets, and the WordNet gloss relation, which has a link between each synset and the synsets of words used in its definition.

For each synset, their algorithm uses a series of random walks to traverse the graph, beginning at the node corresponding to the synset, and at each step either terminating with a probability of .15, or following a randomly chosen outgoing edge on to the next synset node. When the walk terminates as some node, one of the lemmas of that node's synset is output at random, with probability matching that observed in a (relatively small) labelled corpus. After several such random walks, the sequence of emitted words is used as a context-window for the running the word2vec algorithms for a single step in building an embedding for the initial synset.

That is, this is a scheme for generating related/context words for the initial synset; each random walk generates an average of 6.66 words, some of which will be repeated on subsequent walks.

The authors say that they generated 70M contexts, or about 700/ synset. Their paper reports on measuring word similarity using the generated synset embeddings; their suggestion is that using the synset embeddings effectively condenses, or crystallizes out the similarity semantics of the synset, and lets words and synsets be compared with a simple vector cosine.

Pilehvar and Collier (2016) also makes an effort to develop a unique embedding for each synset, but they want to retrofit them into an existing word embedding space. They begin by using the Page-Rank algorithm on the graph of which the nodes are WordNet synsets and the edges are synset relations. (They also used the 'gloss relation', which we thought (not having yet read Goikoetxea et al. (2015) at the time we were first looking at this work) might mean either Extended Wordnet(Harabagiu and Moldovan, 1998; Mihalcea and Moldovan, 2001) or possibly a Lesk-like algorithm(Lesk, 1986).) The goal of the Page-Rank in this situation is to find the other synsets which are most related to some particular one. From a list of synsets, they obtain a list of lemmas, which they call the *bias words* and from these lemmas, which each have vectors in the word-embedding, they estimate approximately where in the vector space the particular synset might be, since they expect it to be near the centrum of the vectors of the words most closely related to it.

4 The design of our system

Our system is based on the idea that the topic of the story, and thus the words used in it, should relate to the correct sense of the target homonym, and not to the incorrect senses. A cursory glance at the

sample stories should make it clear that is not the case in the Ambi-story dataset.

However, as will be seen, we had other problems, and continued to press forward in hopes of resolving them, and perhaps learning something about the relationship between synsets and their contexts.

We had been attempting to duplicate the results of (Pilehvar and Collier, 2016), and had developed a list of fifty words for most WordNet 3.0 synsets of what we supposed were related words. We began by building features using our word-lists and the training data. The list of features grew during the pre-test period to include all those in section 4.3, each feature added because the then-existing set did not appear to provide a useful-enough signal. A description of individual features appears below, in subsection 4.4.

4.1 Top-level structure

Our system uses linear regression(Pedregosa et al., 2011) to estimate the score of a single human assessor, and averages five models, each of which has the same input X data, but a different Y, as each model is trained to a different one of the 5 choices of the human assessments. (See the `choices` field in the sample data record in Figure 1.)

In addition, we attempt to divide the cases into 9 groups, and provide separate linear regression models for each group. The groups are based on our estimate of *synset polarity* for the context and and ending phrases, each of which can have the polarity +1, 0, or -1. In conjunction with the 5 choice models, this gives us as many as 45 models, some of which are clearly starved for training data.

This complicated hierarchy was intended to help duplicate the 'human evalation score'. The range of human assessments on a single story is quite broad, for example, in the record shown in Figure 1, all possible scores 1-5 were assigned by the human evaluators. A program score is judged correct (for the accuracy part of the program assessment) if it is within one standard deviation from the human average, and fractional values are allowed, so that correct program scores for this record are any of those between 1.4188612 and 4.5811388.

We experimented with removing the synset polarity feature and ran the modified system on the development data, actually getting a slight, but not significant, improvement in the accuracy score. This particular feature is apparently not the main cause of our poor standing.

4.2 Low-level structure

The models use a number of tests, all based on a single idea: For every synset, there is a set of related words. If we have the right set of related words, we can measure how close a phrase is to a synset by how close in the semantic space the words in the phrase are to the words in the related set.

For each record (that is, story+homonym+judged_meaning) we determine the WordNet synset. This is easy, because the `judged_meaning` field in the record is taken from the WordNet synset definition.

Once we know the synset, we can chose the correct set of related words, which we call the *bias words*, following (Pilehvar and Collier, 2016), which introduced us to using the Page-Rank algorithm on the WordNet graph. We had encountered their work last year, and had been attempting to duplicate it.

It is clear from the detailed trace they provide in their paper for their intermediate results that we did not duplicate it. However, our list of bias-words looks at least plausibly like a set of related words, and we had been experimenting with it when SemEval task 5 was announced.

We had already precomputed the set of bias words for most WordNet synsets, so we prepared a file, sorted by synset name, `biases.txt`. In principle, the words in the set are ordered, so storing them in a string, where the order is obvious, seems like a reasonable idea. The file is 62.5 MB. We store it as a single bytes object, named File, but also maintain an in-memory table of the locations of 1024 synsets. Finding a synset and its string requires searching the index to bracket the location within about 100 synsets; then a slightly clumsier python-coded binary search finishes off finding the synset (or not. There are about 8500 old-maid synsets, which do not participate in any WordNet relations, and thus the Page-Rank algorithm is not useful. Of these 8500, only `flawlessly.r.01` appears in the task data. We use a hand-built table for this synset.)

The synsets are grouped in the task data; we use a synset every other record, and then possibly never again; so a fairly simple cache can avoid doing extra searches.

4.3 An alternate set of bias-words

While writing up our system description paper, we noticed a footnote we had previously missed, in Goikoetxea et al. (2015), providing a dead link to the Princeton WordNet definition annotation project, and a still-live one to a 'gloss relation' built on it, connecting each Wordnet synset to the synsets of the words used in its definition.[2]

The testing phase was over, but we redid our extraction of bias words, using the new relation to augment the built-in Wordnet semantic relationships. The resulting synset network is better connected, with the positive result that the 8500 former 'old maid' relations now produce sets of related words. The sets of bias of bias words change, but those given as examples in Pilehvar and Collier (2016) are still different than ours. And although the new sets of bias words again seem somewhat related to the synset, the accuracy scores did not improve, although the scores of many of the features did, as discussed in the next subsection. So the quality of the bias words is still rather weak.

4.4 The test suite

Twenty features in all are computed.

1. Six of the features are Jacquard set distances, i.e. $|B \cap P|/|B \cup P|$ where B is a set of either: the first 10 bias words (that is, the most highly ranked 10); or all 50 bias words; and P is a set of the lemmatized words of one of the phrases: `precontext`, `ending`, or both of them.

   It turns out that these Jacquard set distances are mostly zeros. Table 1. shows values for all six of these feature values, using the revised, post-testing, bias words, which have fewer zeros than the original set. (The difference ranges from 284 - 693, or 10% to 24%, but the lowest percentage of zero intersections is 55% for the revised bias words, and the highest percentage is 92%.) The mean values suggest that there was typically one item in the intersection, when it was non-zero; the lower denominators correspond to the difference of 40 between the full set of 50 bias words and the abbreviated set of 10.

2. Three of the features are an estimate of the ranking for a set or union of sets as a possibly fractional number between 1 and 5. It corresponds to the rank of the synset for this record, among all the synsets used for this story,

   in a sorted list of Jacquard set distances between the set union and the synset bias words.

   One of the features, comprising a comparison with the union of the precontext and the ending, has 956 zero values. Since these are not between 1 and 5, they clearly display a bug in the code, probably linked to the fact that not every record has an `ending` entry. Otherwise, there are slightly more values of 5 than 2.5, which would indicate a clear preference for more one synset for more than half the returned values of this feature. This, too, seems surprising, given the large fractions of zero Jacquard value in tests 0-5. Around 300 of the 2838 record for each of the three features are neither 2.5 nor 5, which seems in line with the low expectation that the Jacquard distances for two or more synsets would be non-zero.

3. Eight of the features are average cosine distances between either the 50 or 10 word bias set, and the words of the `precontext`, `ending`, `sentence` or the `precontext` combined with the `ending` For purposes of this and other word vector calculations, we use the English word embedding #20 from the NLPL collection(Fares et al., 2017).

   The revised bias words give mean cosine distances for all eight tests of 0.8258 for the bias-10 set, with a standard deviation of 0.0174; for the bias-50 set, 0.8268, with a standard deviation of 0.0117. Presumably if the means indicate anything, the 'okay' value and the 'not okay' values would have to be quite close together. It makes sense that the first 10 bias values, which are ranked higher than the next 40, should have a slightly lower cosine distance.

4. Three of the features are *synset polarity*, applied to the `precontext`, `sentence`, and

[2]The Princeton work is now at https://wordnetcode.princeton.edu/glosstag.shtml. The University of the Basque country work is at https://ixa2.si.ehu.eus/ukb. Both use a more basic form of synset identifier based on an 8-digit offset into the a,n,r,v database. This offset can be obtained in the python interface with the Synset.offset() method. There was no contemporaneous publication from Princeton about the tagging project, which they apparently considered incomplete, but we cite two later publications which do refer to it.

	PreContext		2cend		both	
	bias-50	**bias-10**	**bias-50**	**bias-10**	**bias-50**	**bias-10**
empty union	1896	2493	2342	2635	1574	2316
mean non-empty	1/61	1/32	1/51	1/18	1/62	1/38

Table 1: Jacquard differences (features 0-5

[For revised bias word set]

`ending`. This measure is adjusted so that only the values $-1, 0, +1$ can occur.

- A value of +1 indicates that the phrase is judged supportive of the synset.
- A value of -1 indicates that the phrase is judged supportive of the counter-synset, the synset associated with this story in other records, but not in this one.
- A value of 0 indicates that the phrase seems to be neutral.

One nice thing about this not-very parametric measure, is that it is relatively easy to extract gold values from the training and development data. If the story with the `ending` scores better or worse than the story without it, the `ending` phrase is respectively +1 or -1 with respect to the synset. If the `precontext` standing alone scores higher for one synset than the other, it is positively polarized for that synset, and so on.

We computed it by averaging cosine distances between all words in the phrase and all words in bias set, discarding phrase words whose distance was above a threshold. Since we could easily assess whether we were improving the gold polarity scores by adjusting the algorithm and parameters, it seems like we should be able to tune for better approximation to the known gold scores. In fact, our best accuracy measure for polarity is 0.412, noticeably better than blind chance, but there is plenty of room for improvement. (See Figure 2.)

precontext outcomes

goal	**feature value**		
	-1	0	+1
-1	459	45	315
0	321	51	270
+1	375	42	402

ending outcomes

goal	**feature value**		
	-1	0	+1
-1	313	36	202
0	216	33	177
+1	234	25	284

Figure 2: Synset-polarity scores

5 Results

Our results, which appear under user `stepheneugene` on the task leaderboard, are unimpressive. The system achieves 0.55 accuracy, that is, its fractional estimates are within 1 standard deviation of the average of five human assessments 55% of the time. The Spearman correlation of our scores with the gold scores is 0.13. This gives us a combined score of 0.34, and a placing of of 74th out of 78 competing entrants (not counting the organizer's place-holder system.)

6 Conclusions

It appears that the technique we used did not succeed with the bias word sets which we provided. There are a variety of possible reasons.

One certainty is that the dataset is designed to lead the imagination astray, and provides much deliberate ambiguity.

However, our features were quite ineffective in exploiting the possible relationships between our synset-related words and the story context. This suggests a difficulty with the basic idea. Given the relative success of more modern models on this task, it may not be worthwhile to use the 'related words technique.'

7 Limitations

Since this is a negative result, and we report inconsistencies which suggest programming errors, it is possible that another approach to basically the same technique could be more effective.

8 Acknowledgments

The work has been supported by the project R&D of Technologies for Advanced Digitalization in the Pilsen Metropolitan Area (DigiTech) No. CZ.02.01.01/00/23_021/0008436.

The use of computing facilities maintained by CESNET, including the MetaCentrum Cloud, is gratefully acknowledged.

References

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Murhaf Fares, Andrei Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 271–276, Gothenburg, Sweden. Linköping University Electronic Press.

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

Janosch Gehring, Selina Meyer, and Michael Roth. 2026. SemEval-2026 task 5: Rating plausibility of word senses in ambiguous stories through narrative understanding. In *Proceedings of the 20th International Workshop on Semantic Evaluation*, San Diego, California. Association for Computational Linguistics.

Janosch Gehring and Michael Roth. 2025. Ambistory: A challenging dataset of lexically ambiguous short stories. In *Proceedings of the 14th Joint Conference on Lexical and Computational Semantics*, pages 152–171. Association for Computational Linguistics.

Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. 2015. Random walks and neural network language models on knowledge bases. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1434–1439, Denver, Colorado. Association for Computational Linguistics.

Sanda Harabagiu and Dan Moldovan. 1998. Knowledge processing on an extended wordnet. *WordNet: An electronic lexical database*, 305:381–405.

Taher H. Haveliwala. 2002. Topic-sensitive pagerank. In *WWW02: Hypermedia Track of the Elevnth International Word-Wide Web Conference*.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26.

Rada Mihalcea and Dan I Moldovan. 2001. extended wordnet: Progress report.

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

Siddharth Patwardhan and Ted Pedersen. 2006. Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the Workshop on Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Mohammad Taher Pilehvar and Nigel Collier. 2016. Deconflated semantic representations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1680–1690, Austin, Texas. Association for Computational Linguistics.

Alexandre Rademaker, Abhishek Basu, and Rajkiran Veluri. 2023. Semantic parsing and sense tagging the Princeton WordNet gloss corpus. In *Proceedings of the 12th Global Wordnet Conference*, pages 243–253, University of the Basque Country, Donostia - San Sebastian, Basque Country. Global Wordnet Association.

Alexandre Rademaker, Bruno Cuconato, Alessandra Cid, Alexandre Tessarollo, and Henrique Andrade. 2019. Completing the Princeton annotated gloss corpus project. In *Proceedings of the 10th Global Wordnet Conference*, pages 378–386, Wroclaw, Poland. Global Wordnet Association.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633.