GRASP at CoMeDi Shared Task: Multi-Strategy Modeling of Annotator Behavior in Multi-Lingual Semantic Judgments

David Alfter
GRIDH
University of Gothenburg
Sweden
david.alfter@gu.se

Mattias Appelgren
CLASP
University of Gothenburg
Sweden
mattias.appelgren@gu.se

Abstract

This paper presents the GRASP team's systems for the CoMeDi 2025 shared task on disagreement prediction in semantic annotation. The task comprises two subtasks: predicting median similarity scores and mean disagreement scores for word usage across multiple languages including Chinese, English, German, Norwegian, Russian, Spanish, and Swedish. For subtask 1, we implement three approaches: Prochain, a probabilistic chain model predicting sequential judgments; FARM, an ensemble of five fine-tuned XLM-RoBERTa models; and THAT, a task-specific model using XL-Lexeme with adaptive thresholds. For subtask 2, we develop three systems: LAMP, combining language-agnostic and monolingual models; BUMBLE, using optimal language combinations; and DRAMA, leveraging disagreement patterns from FARM's outputs. Our results show strong performance across both subtasks, ranking second overall among participating teams. The probabilistic Prochain model demonstrates surprisingly robust performance when given accurate initial judgments, while our task-specific approaches show varying effectiveness across languages.

1 Introduction

The growing importance of modeling annotator disagreement in NLP has emerged as a crucial challenge for developing more robust and nuanced language understanding systems. While traditional approaches often treat divergent annotations as noise to be filtered out, recent work suggests that systematic patterns in annotator disagreement can provide valuable insights into linguistic ambiguity, contextual interpretation, and the inherent complexity of language understanding tasks (Uma et al., 2021; Leonardelli et al., 2023).

The 2025 Workshop on Context and Meaning - Navigating Disagreements in NLP Annotations (CoMeDi)[1] addresses this challenge through a shared task focused on predicting patterns of annotator disagreement across multiple languages. The task encompasses seven languages (Chinese, English, German, Norwegian, Russian, Spanish, and Swedish), drawing from various semantic change datasets as shown in Table 1. This multilingual scope provides a unique opportunity to explore how annotator disagreement patterns manifest across different linguistic and cultural contexts.

In this paper, we present a range of approaches for modeling annotator behavior and predicting disagreement patterns. Our methods span from probabilistic modeling of sequential judgments to neural architectures specifically designed to capture the nuanced nature of semantic annotation tasks. Through these diverse approaches, we aim to contribute to the broader understanding of how to effectively model and utilize annotator disagreement in NLP systems.

2 Related Work

Prior work on modeling annotator disagreement falls into several key areas. Early approaches treated disagreement primarily as noise to be filtered out through measures like inter-annotator agreement (Artstein and Poesio, 2008) or adjudication (Passonneau, 2004). However, recent work has shown that systematic patterns in annotator disagreement can provide valuable linguistic insights (Plank et al., 2014; Pavlick and Kwiatkowski, 2019).

In the context of semantic annotation, several studies have specifically examined disagreement patterns in word sense annotation. Erk et al. (2013) introduced a graded approach to word sense, showing that annotators often perceive multiple valid interpretations rather than discrete senses. This finding was further supported by Jurgens (2014),

[1] https://comedinlp.github.io/

Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation, pages 78–89
January 19, 2025.

Language	Datasets (version) [citation]
Chinese	ChiWUG (1.0.0) [Chen et al. (2023)]
English	DWUG_EN (3.0.0), DWUG_EN_resampled (1.0.0) [Schlechtweg et al. (2024)]
German	DWUG_DE (3.0.0), DWUG_DE_resampled (1.0.0), DiscoWUG (2.0.0), RefWUG (1.1.0) [Schlechtweg et al. (2024), Kurtyigit et al. (2021)]
	DURel (3.0.0) [Schlechtweg et al. (2018)]
	SURel (3.0.0) [Hätty et al. (2019)]
Norwegian	NorDiaChange1, NorDiaChange2 [Kutuzov et al. (2022)]
Russian	RuSemShift_1, RuSemShift_2 [Rodina and Kutuzov (2020)]
	RuShiftEval1, RuShiftEval2, RuShiftEval3 [Kutuzov and Pivovarova (2021)]
	RuDSI [Aksenova et al. (2022)]
Spanish	DWUG_ES (4.0.1) [Schlechtweg et al. (2024)]
Swedish	DWUG_SV (3.0.0), DWUG_SV_resampled (1.0.0) [Schlechtweg et al. (2024)]

Table 1: Overview of Semantic Change Datasets by Language

who demonstrated that disagreements often reflect genuine semantic ambiguity rather than annotator error.

Cross-lingual aspects of semantic annotation have been explored in various contexts. Bender and Friedman (2018) highlighted how linguistic and cultural differences can lead to systematic variations in annotation patterns across languages. This work was extended by Chang et al. (2014), who showed that annotation disagreements often reflect genuine cross-linguistic differences in semantic categorization.

Recent work has increasingly focused on computational approaches to modeling annotator behavior. Uma et al. (2021) demonstrated the effectiveness of learning annotator-specific patterns for improving overall annotation quality. Similarly, Davani et al. (2022) showed how multi-task learning can help capture individual annotator preferences while maintaining consistent predictions.

3 Approaches

The shared task consists of two sub-tasks. For subtask 1, participants are asked to predict the *median* similarity score of a word in two sentences based on multiple human annotations (between 2 and 7). For sub-task 2, participants are asked to predict the mean disagreement of human annotators given a target word and two example contexts.

3.1 Sub-task 1

We approach sub-task 1 in two different ways: first, we use a simple method that relies on probabilities *between* human judgments. We predict the probability of each judgment given the previous judgment(s). Second, we model annotators using two different architectures: XLM-RoBERTa (Conneau et al., 2020) and XL-Lexeme (Cassotti et al., 2023). XL-Lexeme is a WordEncoder model that has been fine-tuned on Word-in-Context tasks and thus should be an apt choice to model the semantic closeness of target words given two sentences.

3.1.1 Prochain

Our first system, Prochain (probabilistic chain), is a non-parametric probabilistic model. While the human judgments are made independently, our model exploits potential underlying patterns in these independent assessments to create a probabilistic framework for prediction. This approach assumes that even though judges make decisions independently, there exist statistical relationships between different judgment aspects that can be leveraged for prediction.

For training, given a tuple of three judgments (j_1, j_2, j_3), we calculate the frequency distribution of j_2 given j_1, and of j_3 given (j_1, j_2). We then normalize these frequency distributions to obtain probability distributions, as shown in Equations 1 and 2.

$$P(j_2|j_1) = \frac{\text{count}(j_1, j_2)}{\text{count}(j_1)} \tag{1}$$

$$P(j_3|j_1, j_2) = \frac{\text{count}(j_1, j_2, j_3)}{\text{count}(j_1, j_2)} \tag{2}$$

For prediction, given a first judgment j_1, we predict j_2 as a probability distribution based on the normalized frequencies observed during training. Similarly, given (j_1, j_2), we predict j_3 as a probability distribution based on the observed frequencies of j_3 for each combination of (j_1, j_2) in the training data.

Since this method requires the first judgment to be calculated by other means, we use the training data and XL-Lexeme to calculate the cosine similarity between the target word in the two sentences for each item in the training data, then map this continuous value to a discrete value j_0 as shown in Equation 3, then learn mappings between the predicted value j_0 and j_1 in the training data, as we did for j_2 and j_3.

$$j_0 = \begin{cases} 1 & \text{if sim} < 0.4 \\ 2 & \text{if } 0.4 \leq \text{sim} < 0.6 \\ 3 & \text{if } 0.6 \leq \text{sim} < 0.8 \\ 4 & \text{if sim} \geq 0.8 \end{cases} \tag{3}$$

At prediction time, for the prediction of the first judgment, we calculate the cosine similarity between the target word in the two sentences, map this value to a discrete value, then use the Prochain method to predict 11 values, of which we take the most frequently predicted value as j_1.

3.1.2 FARM

Our second system, FARM (Five Adapted Roberta Models), is an XLM-Roberta-base model which is fine-tuned for sentence classification, in the standard way, i.e. a classification head is placed on the special first token, $\langle \text{s} \rangle$. To model the disagreement between judgments we create 5 separate "datasets" and train a model on each of these sets. The datasets vary simply in which label we select as the target. If J is the set of judgments for a particular pair of sentences then dataset d_i labels the pair with $j_{imod|J|}$.

For prediction we simply have each of the five models predict their output and then take the median of the 5 predictions.

Each of the Roberta models is trained for 3 epochs using learning rate $2e^{-5}$ and 200 warmup steps. A batch size of 8, a linear learning rate scheduler, the ADAM optimizer, optimized against cross-entropy loss. We use the hugging face trainer interface with any unmentioned arguments left as the default.

3.1.3 THAT

Our third system, THAT (Task-specific Human-like Adaptive Thresholds), is a fine-tuned XL-Lexeme model (Cassotti et al., 2023). The model is trained to embed two sentences such that the cosine similarity between the sentences is inverse proportional to to the label between them, i.e., sentences which are scored as 4 are closer together while sentences which are labeled 1 are further apart. The model is trained to minimize the contrastive loss (Hadsell et al., 2006) as described in in Cassotti et al. (2023).

At prediction time we calculate the cosine similarity between the sentences and we then set three thresholds, t_1, t_2, t_3. We label a sentence pair as 1 if $cosine(s_1, s_2) < t_1$, 2 if its less than t_2, 3 if its less than t_3 and 4 otherwise. We tune the thresholds on the dev-set using the following algorithm.

We begin the thresholds regularly spaced: $t_1 = 0.4, t_2 = 0.6, t_3 = 0.8$ we then vary the thresholds between -0.05, 0, or $+0.05$ from the base threshold. This creates 3^3 different possible threshold combinations. We evaluate each against the dev-set, selecting the threshold which gives the highest score. We then repeat the process until we converge on stable threshold values.

We found that the method converged such that $t_1 = t_2 = t_3$ which means in practice that the best results were gained when we simply predicted 1 or 4.

However, for the purpose of this task we wanted to actually model disagreement between annotators. One way in which annotators may be different is that they have different thresholds for what they think is for example a 3 vs 4. We model this by creating 5 different threshold functions. The thresholds are random perturbations around the optimal threshold. These are not validated directly on the dev set. We then select the median value as the actual label. Given that they all rely on the same underlying similarity function the main benefit of this method is to find examples which are close to the decision boundary and perhaps changing their label from for example 1 to 2.

3.2 Subtask 2

We approach sub-task 2 in two different ways: first, we use feature-engineering to extract features from the sentences and target words. The features were specifically developed for the shared task. We then train regression models on the features, with the target mean disagreement as label. Second, we use the output from our FARM model to calculate disagreement.

Systems 1 and 2 are feature-based systems using a common set of features described in the next section and XGBoost as regressor (Chen and Guestrin, 2016). We performed a hyper-parameter search to fix the best parameters using the dev set. For tagging, we use spacy (Honnibal et al., 2020), and for WordNet features, we use nltk (Bird et al., 2009). For preprocessing, we use pymorphy2 (Korobov, 2015) to lemmatize Russian and jieba[2] to tokenize Chinese. All other languages are transformed to lowercase.

3.2.1 Feature extraction

NLP features We use various NLP features to represent the example contexts. The features are: cosine similarities between context 1 and 2 based on XLM-RoBERTa embeddings of the target token, and between each context (CLS token) and the target word (target token embedding), as well as cosine similarities based on XL-Lexeme, the length of each context, and the length difference in characters between the two contexts, and between each context and the target word, as well as the ratio of lengths between the two contexts, word overlap between the two contexts, fuzzy ratios between the two contexts, and each context and the target word, NER overlap between the two contexts, n-gram overlap ($n = 2$ and $n = 3$) between the two contexts, the position of the target word in each context, whether the target word has the same (1) part-of-speech, (2) NER tag, (3) dependency relation in the two contexts, and WordNet features for supported languages (all except Russian and German): the number of lemmas in the first synset, the depth of the first synset, and the number of hypernyms and hyponyms.

Psycholinguistic features We use concreteness, imageability, familiarity and age-of-acquisition from the MRC database (Wilson, 1988). Since this database only contains data for English, we fine-tuned XLM-RoBERTa models on each of the features for 3 epochs, then use these models to predict the features for all languages.

Prototype features We calculate sense prototypes for each target word using a custom algorithm.[3] The algorithm is an iterative, non-parametric approach to inducing word sense prototypes from contextual representations using the XL-Lexeme transformer model. The core induction process performs multiple iterations (default: 51) where each iteration processes contexts in random order, maintaining a set of induced sense prototypes while comparing new contextualized embeddings with existing prototypes using cosine similarity, either merging similar senses or creating new prototypes based on a similarity threshold. The prototype merging strategy computes pairwise similarities between sense representations, identifies the most similar pairs across different iteration results, and creates aggregate prototypes by averaging the vector representations, using a similarity threshold to control the granularity of sense distinctions. The algorithm builds consensus across iterations by identifying the most frequent number of induced senses (mode), filtering iteration results to retain only those matching the modal number of senses, aligning similar senses across different iterations through similarity-based matching, and creating final sense prototypes by merging aligned sense representations. In the final stage, the algorithm assigns sense labels to the induced prototypes, maps each context back to its most similar prototype, and creates a mapping between context IDs and sense labels. This approach allows for dynamic sense discovery without pre-specifying the number of senses, while maintaining consistency through multiple iterations and consensus building.

After running the algorithm, we assign each word its closest prototype vector. For a target word t and two contexts, we then use the cosine similarity between the prototypes p_1 and p_2, and between each prototype and each target word embedding (t_1 to p_1, t_1 to p_2, t_2 to p_1 and t_2 to p_2).

On length differences Two of the less straightforward features might be differences in context length and between contexts and target words. Let us imagine two contexts for the target word *bark*:

- The bark was rough
- The bark was rough and dark brown, typical of old oak trees in this forest that had weathered many storms

A longer context provides more specific information and constraints about what the target word means (tree bark), while the shorter context leaves more room for ambiguity (it could be dog bark or tree bark). This difference in specificity could lead annotators to have more disagreement with shorter contexts due to lack of disambiguating information, and show more agreement with longer contexts that provide clear contextual clues.

For the difference in length between the words and the contexts, it can be said that a short target word in a long context usually has clear situational grounding, while a longer target phrase in a short

[2] https://github.com/fxsjy/jieba

[3] https://github.com/daalft/senseprototypeinduction

context might lack sufficient contextual support for judgment. This could lead to systematic patterns in annotator disagreement based on these length relationships.

Feature importance Given the large number of features, we use CorrelationAttributeEval from WEKA (Frank et al., 2016) with ten-fold cross-validation to calculate feature importance, and find that all features are important to the task, with the most predictive features being character overlap of trigrams, word overlap between sentences, and familiarity. See Appendix A for the full list of features. Table 9 in the Appendix lists the average merit and rank of each feature.

3.2.2 LAMP

Our first submission, LAMP (Language Agnostic, Monolingual, Prochain), uses a combination of models to produce a result. We train one language-agnostic model on all languages, as well as one model per language. We also include Prochain with an iteration count of 3, based on which we calculate disagreement. We then average all predictions to arrive at the final prediction.

3.2.3 BUMBLE

Our second submission, BUMBLE (Best Universal Model By Language Ensemble), uses a single model to predict the disagreement. We train different models on all possible combinations of languages (single-language models, two languages, . . . up to all languages), then select the best model for each language based on its score on the dev set. Results show that the best models are two- or three-language models, but that these models do not always include the language they are predicting.

3.2.4 DRAMA

Our third submission, Disagreement Rating Across Multiple Answers (DRAMA), uses the five judgments generated from FARM and calculates the difference scores from those judgments. I.e. FARM, being 5 different fine-tuned Roberta models output five different judgments J and while FARM calculates the median value over these five judgments, DRAMA calculates the mean difference score as described in the task:

$$D(J) = \frac{1}{|J|} \sum_{(j_1,j_2) \in J} (|j_1 - j_2|) \tag{4}$$

Due to the fact that we use 5 judgments while the actual data uses a varying number of judgments which is usually lower than 5 (e.g. 2 for Chinese) the scores are likely to be higher on average than the true data. However, if the models have successfully modeled the variation in the data, i.e., that more ambiguous utterances have more variance, then the correlation score would still reflect this.

4 Results and Analysis

Tables 2 and 3 summarize our results on the test set for Tasks 1 and 2 respectively. All of our submitted systems demonstrate strengths in specific languages and scenarios, suggesting that different approaches capture different aspects of annotator behavior.

Language	Prochain	FARM	THAT
Chinese	**0.332**	0.177	0.317
German	0.619	0.515	**0.656**
English	0.565	**0.608**	0.555
Norwegian	0.469	0.285	**0.589**
Russian	0.464	0.344	**0.487**
Spanish	0.593	0.582	**0.636**
Swedish	0.556	0.481	**0.648**
Overall	0.514	0.428	**0.555**

Table 2: Results for task 1 according to Krippendorff's α. The best results per language are indicated in bold.

Language	LAMP	DRAMA	BUMBLE
Chinese	0.265	0.498	**0.539**
German	**0.135**	0.123	0.108
English	0.062	**0.097**	0.041
Norwegian	0.269	**0.317**	0.272
Russian	0.110	0.159	**0.167**
Spanish	0.102	0.101	**0.115**
Swedish	0.204	0.233	**0.296**
Overall	0.164	0.218	**0.220**

Table 3: Results for task 2 calculated according to equation 4. The best results per language are indicated in bold.

4.1 Task 1

For predicting median similarity scores, our task-specific model THAT achieved the best overall performance (α = 0.555), followed by Prochain (α = 0.514) and FARM (α = 0.428). Several interesting patterns emerge from these results:

1. The systems consistently performed better on Germanic languages, with particularly strong results for German (THAT: 0.656), Swedish (THAT: 0.648), and English (FARM: 0.608). This pattern holds across all three systems, suggesting that either these languages share helpful structural similarities, or their annotators demonstrate more consistent judgment patterns.
2. Despite its simplicity, Prochain performed surprisingly well, even outperforming FARM overall. This suggests that sequential dependencies between judgments might be more important than previously thought. When provided with correct initial judgments on development data, Prochain achieves remarkably high performance (see Section 5.2), indicating strong predictability in how annotators influence each other's subsequent judgments.
3. All systems struggled most with Chinese data, with the best performance being Prochain's α = 0.332. This might be attributed to several factors:
    - The extremely skewed label distribution (83% label 4)
    - The fundamental differences in how word meanings are constructed in Chinese
    - The smaller number of annotators per item in the Chinese dataset

4.2 Task 2

The disagreement prediction task proved more challenging overall, with markedly different patterns from Task 1: BUMBLE (0.220) and DRAMA (0.218) performed similarly overall but showed distinct strengths across languages. Notably, the best performance was achieved on Chinese (0.539 with BUMBLE) - a striking contrast to Task 1 where Chinese was the most challenging language.

BUMBLE's language combination strategy revealed that optimal performance often came from models trained on two or three languages, but surprisingly, these optimal combinations didn't always include the target language. This suggests the existence of cross-linguistic patterns in annotator disagreement that transcend individual language boundaries.

4.3 Overall

In the context of other participating teams, our systems achieved competitive results: Second place overall in both tasks, *first* place for Chinese, English, and Norwegian in Task 2, and *second* place for Russian, Spanish, and Swedish in both tasks.

These results suggest that our multi-strategy approach, combining probabilistic modeling, neural architectures, and feature engineering, successfully captures different aspects of annotator behavior across languages.

5 Discussion

5.1 Label distribution

Overall, we notice that the data labels are strongly skewed towards label 4, as illustrated in Table 4. For all languages, most of the labels are 4, and on average, label 1 comes second. This might explain why THAT was gravitating towards a binary threshold, i.e., dividing the data into labels 1 and 4.

	1	2	3	4
Overall	0.150	0.094	0.130	0.630
Chinese	0.009	0.043	0.120	0.830
German	0.130	0.210	0.170	0.500
Russian	0.230	0.032	0.170	0.650
Norwegian	0.140	0.032	0.088	0.770
Spanish	0.210	0.088	0.210	0.500
English	0.230	0.170	0.140	0.460
Swedish	0.200	0.089	0.090	0.620

Table 4: Label distribution for median judgments task 1. Percentage of samples given a particular label in the train-set.

5.2 Prochain

The Prochain method is surprisingly strong in subtask 1, despite its simplicity, *if* the first judgment is given and correct. This is confirmed by the results on the development data for subtask 1: when taking the first judgment from the development data label file, and predicting a second and third judgment using PROCHAIN, then calculating the median value, we reach results of 0.938 on average,as illustrated in Table 5.

5.3 THAT – DRAMA

Table 6 shows the results of DRAMA and THAT2 (calculating disagreement on the output of THAT; we did not submit THAT2 run) on the dev data of the second task. The NaN numbers for Chinese come from the fact that in the dev-set there is no disagreement which means that the Spearman rank

Language	Krippendorff's α
Average	0.938
Chinese	1.000
English	0.950
German	0.902
Norwegian	0.948
Russian	0.862
Spanish	0.962
Swedish	0.939

Table 5: Results of Prochain if the first judgment is taken from the gold labels

cannot calculate a difference. A surprising fact is how poorly XL-Lexeme with 5 different thresholds performs. It was our best model in the first subtask, however, it seems that the thresholding technique does not align with the difference observed in the human judges. This would suggest that the judges do not have an equivalent difference space to XL-Lexeme in their heads but different thresholds for judging something a 2 or a 3. Our choice of thresholds may also have been suboptimal.

	DRAMA	THAT2
German	0.158	0.049
Russian	0.028	0.043
Swedish	0.125	0.083
Spanish	0.014	-0.051
English	0.084	0.051
Chinese	NAN	NAN
Norwegian	0.275	0.0551
All	0.114	0.0383

Table 6: Results for DRAMA and THAT2 on the dev set

5.4 The lack of disagreement in Chinese data

We noticed that in the Chinese data for task 1, no disagreement was found between annotators. In addition, only two annotations were present. While puzzling at first, this may well be due to differences in annotation procedure. It is conceivable that annotations were consolidated to resolve disagreements before the data was released. However, the data paper states that they follow the same guidelines as other data sets (Chen et al., 2023).

5.5 The case of English

An unexpected finding in our results is the particularly challenging nature of English data for disagreement prediction, despite the language's extensive resources and representation in training data. While English achieves moderate performance in median prediction (Task 1) with $\alpha = 0.565$, it shows strikingly low correlation scores in disagreement prediction (Task 2), with even our best system DRAMA achieving only 0.097. Several factors may contribute to this counterintuitive result. First, the English dataset demonstrates more balanced label distribution (46% label 4 compared to the overall average of 63%), suggesting annotators may be making more nuanced distinctions rather than defaulting to high similarity judgments. Second, English's rich polysemy and extensive metaphorical usage may lead to more genuine cases of ambiguity, making annotator disagreement patterns less systematic and therefore harder to predict. This hypothesis is supported by the fact that even our more sophisticated neural approaches failed to capture these patterns effectively.

6 Conclusion

The GRASP team's participation in the CoMeDi shared task has led to several important insights into modeling annotator disagreement across multiple languages. Our diverse approach, implementing both probabilistic and neural methods, proved effective across both subtasks, securing second place overall.

The strong performance of our simple Prochain model highlights the value of probabilistic approaches in capturing annotator behavior, while the varying success of our more complex models across languages suggests that language-specific factors play a crucial role in disagreement prediction.

The skewed label distribution toward label 4 significantly influenced model behavior, particularly affecting our threshold-based approaches. Future work could focus on better handling this class imbalance and developing more robust cross-lingual disagreement modeling techniques.

Acknowledgements

The authors would like to thank Pierluigi Cassotti for sharing the code to fine-tune XL-Lexeme. This work has in part been funded by the research program Change is Key! supported by Riksbankens Jubileumsfond (under reference number M21-0021).

References

Anna Aksenova, Ekaterina Gavrishina, Elisei Rykov, and Andrey Kutuzov. 2022. RuDSI: Graph-based word sense induction dataset for Russian. In *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, pages 77–88, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational linguistics*, 34(4):555–596.

Emily M Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

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

Pierluigi Cassotti, Lucia Siciliani, Marco DeGemmis, Giovanni Semeraro, and Pierpaolo Basile. 2023. XL-LEXEME: WiC pretrained model for cross-lingual LEXical sEMantic changE. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1577–1585, Toronto, Canada. Association for Computational Linguistics.

Angel Chang, Manolis Savva, and Christopher D Manning. 2014. Learning spatial knowledge for text to 3D scene generation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 2028–2038.

Jing Chen, Emmanuele Chersoni, Dominik Schlechtweg, Jelena Prokic, and Chu-Ren Huang. 2023. ChiWUG: A graph-based evaluation dataset for Chinese lexical semantic change detection. In *Proceedings of the 4th International Workshop on Computational Approaches to Historical Language Change*, Singapore. Association for Computational Linguistics.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. *Preprint*, arXiv:1911.02116.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2013. Measuring word meaning in context. *Computational Linguistics*, 39(3):511–554.

Eibe Frank, Mark A Hall, and Ian H Witten. 2016. *The WEKA workbench*. Morgan Kaufmann.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1735–1742.

Anna Hätty, Dominik Schlechtweg, and Sabine Schulte im Walde. 2019. SURel: A gold standard for incorporating meaning shifts into term extraction. In *Proceedings of the 8th Joint Conference on Lexical and Computational Semantics*, pages 1–8, Minneapolis, MN, USA.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

David Jurgens. 2014. An analysis of ambiguity in word sense annotations. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3006–3012.

Mikhail Korobov. 2015. Morphological analyzer and generator for russian and ukrainian languages. In Mikhail Yu. Khachay, Natalia Konstantinova, Alexander Panchenko, Dmitry I. Ignatov, and Valeri G. Labunets, editors, *Analysis of Images, Social Networks and Texts*, volume 542 of *Communications in Computer and Information Science*, pages 320–332. Springer International Publishing.

Sinan Kurtyigit, Maike Park, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2021. Lexical Semantic Change Discovery. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online. Association for Computational Linguistics.

Andrey Kutuzov and Lidia Pivovarova. 2021. Rushifteval: a shared task on semantic shift detection for russian. *Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Dialog conference*.

Andrey Kutuzov, Samia Touileb, Petter Mæhlum, Tita Enstad, and Alexandra Wittemann. 2022. NorDiaChange: Diachronic semantic change dataset for Norwegian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2563–2572, Marseille, France. European Language Resources Association.

Elisa Leonardelli, Gavin Abercrombie, Dina Almanea, Valerio Basile, Tommaso Fornaciari, Barbara Plank, Verena Rieser, Alexandra Uma, and Massimo Poesio. 2023. SemEval-2023 task 11: Learning with disagreements (LeWiDi). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2304–2318, Toronto, Canada. Association for Computational Linguistics.

Rebecca J Passonneau. 2004. Computing reliability for coreference annotation. In *4th International Conference on Language Resources and Evaluation, LREC 2004*, pages 1503–1506. European Language Resources Association (ELRA).

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Julia Rodina and Andrey Kutuzov. 2020. RuSemShift: a dataset of historical lexical semantic change in Russian. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1037–1047, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Dominik Schlechtweg, Pierluigi Cassotti, Bill Noble, David Alfter, Sabine Schulte im Walde, and Nina Tahmasebi. 2024. More DWUGs: Extending and evaluating word usage graph datasets in multiple languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida. Association for Computational Linguistics.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. Diachronic Usage Relatedness (DURel): A framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 169–174, New Orleans, Louisiana.

Alexandra N Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Michael Wilson. 1988. MRC psycholinguistic database: Machine-usable dictionary, version 2.00. *Behavior research methods, instruments, & computers*, 20(1):6–10.

A Feature list

Feature Name	Description
cosine_sim_sent1_sent2	BERT embedding cosine similarity between the two sentences
cosine_sim_sent1_target	BERT embedding cosine similarity between first sentence and target word
cosine_sim_sent2_target	BERT embedding cosine similarity between second sentence and target word
len_diff_sent1_sent2	Absolute character length difference between sentences
len_ratio_sent1_sent2	Ratio of first sentence length to second sentence length
len_diff_sent1_target	Character length difference between first sentence and target
len_diff_sent2_target	Character length difference between second sentence and target
word_overlap_sent1_sent2	Jaccard similarity of word sets between sentences
word_overlap_sent1_target	Jaccard similarity between first sentence words and target word
word_overlap_sent2_target	Jaccard similarity between second sentence words and target word
fuzz_ratio_sent1_sent2	Levenshtein ratio between the two sentences
fuzz_ratio_sent1_target	Levenshtein ratio between first sentence and target
fuzz_ratio_sent2_target	Levenshtein ratio between second sentence and target
ner_count_sent1	Number of named entities in first sentence
ner_count_sent2	Number of named entities in second sentence
ner_overlap	Number of shared named entities between sentences
char_ngram_overlap_2	Overlap of character bigrams between sentences
char_ngram_overlap_3	Overlap of character trigrams between sentences
target_position_sent1	Relative position of target word in first sentence
target_position_sent2	Relative position of target word in second sentence
sent1_length	Word count of first sentence
sent2_length	Word count of second sentence
length_diff	Absolute difference in sentence word counts
word_overlap	Jaccard similarity of lowercased words
same_pos	Binary indicator if target words have same POS tag
same_ner	Binary indicator if target words have same NER tag
same_dep	Binary indicator if target words have same dependency relation
corpus_frequency	Brown corpus frequency (English only)
avg_word_vec_similarity	Cosine similarity of averaged spaCy word vectors
num_synsets	Number of WordNet synsets for target word
num_lemmas	Number of lemmas in target word’s synsets
first_synset_depth	Depth of first synset in WordNet hierarchy
num_hypernyms	Number of hypernyms for first synset
num_hyponyms	Number of hyponyms for first synset
xl_similarity	XL-Lexeme embedding cosine similarity between the two sentences

Table 7: Features Extracted for Disagreement Prediction (NLP features)

Feature Name	Description
conc	Concreteness
imag	Imageability
fam	Familiarity
aoa	Age-of-acquisition
proto_sim	Cosine similarity between the prototypes (p1, p2) of the target word in both sentences
proto_sim_sent1	Cosine similarity between the target word embedding in sentence 1 (t1) and its prototype (p1)
proto_sim_sent2	Cosine similarity between the target word embedding in sentence 2 (t2) and its prototype (p2)
cross_proto_sim1	Cross-prototype similarity: target word embedding from sentence 1 (t1) to prototype from sentence 2 (p2)
cross_proto_sim2	Cross-prototype similarity: target word embedding from sentence 2 (t2) to prototype from sentence 1 (p1)

Table 8: Features Extracted for Disagreement Prediction (Psycholinguistic and prototype features)

Average Merit	Average Rank	Attribute
0.119 ± 0.003	1.0 ± 0.00	char_ngram_overlap_3
0.107 ± 0.003	2.0 ± 0.00	word_overlap_sent1_sent2
0.093 ± 0.003	3.3 ± 0.46	char_ngram_overlap_2
0.089 ± 0.003	3.8 ± 0.60	len_diff_sent1_target
0.085 ± 0.002	4.9 ± 0.30	len_diff_sent2_target
0.079 ± 0.002	6.0 ± 0.00	same_pos
0.067 ± 0.002	8.1 ± 0.94	num_lemmas
0.067 ± 0.002	8.1 ± 0.94	num_synsets
0.066 ± 0.002	8.5 ± 1.12	word_overlap
0.064 ± 0.002	9.3 ± 1.00	sent2_length
0.054 ± 0.003	11.0 ± 0.00	sent1_length
0.044 ± 0.002	12.0 ± 0.00	len_diff_sent1_sent2
0.035 ± 0.002	13.1 ± 0.30	same_ner
0.032 ± 0.004	14.1 ± 0.70	word_overlap_sent2_target
0.026 ± 0.002	15.7 ± 0.64	cosine_sim_sent1_sent2
0.027 ± 0.002	16.0 ± 1.48	conc
0.024 ± 0.003	16.8 ± 1.17	avg_word_vec_similarity
0.022 ± 0.003	17.7 ± 0.90	x1_p1
0.018 ± 0.004	19.7 ± 1.42	aoa
0.018 ± 0.001	19.9 ± 0.83	corpus_frequency
0.016 ± 0.003	21.1 ± 1.45	x1_p2
0.015 ± 0.002	21.9 ± 1.04	length_diff
0.014 ± 0.002	22.2 ± 1.40	word_overlap_sent1_target
0.010 ± 0.002	24.4 ± 1.20	same_dep
0.010 ± 0.002	25.0 ± 0.63	cosine_sim_sent2_target
0.007 ± 0.003	26.1 ± 0.94	imag
0.004 ± 0.004	27.6 ± 1.96	fuzz_ratio_sent1_sent2
0.003 ± 0.002	27.8 ± 1.08	len_ratio_sent1_sent2
0.001 ± 0.003	29.2 ± 1.25	x2_p2
-0.002 ± 0.002	30.9 ± 1.37	cosine_sim_sent1_target
-0.004 ± 0.003	31.6 ± 1.28	p1_p2
-0.008 ± 0.003	33.8 ± 0.98	x2_p1
-0.008 ± 0.002	34.1 ± 0.94	xl_sim
-0.010 ± 0.004	34.6 ± 1.56	fuzz_ratio_sent2_target
-0.012 ± 0.002	35.5 ± 1.02	fuzz_ratio_sent1_target
-0.017 ± 0.003	37.5 ± 1.12	target_position_sent1
-0.018 ± 0.003	38.0 ± 0.77	target_position_sent2
-0.019 ± 0.003	38.2 ± 0.98	first_synset_depth
-0.051 ± 0.005	40.4 ± 0.49	num_hyponyms
-0.051 ± 0.002	40.6 ± 0.49	num_hypernyms
-0.101 ± 0.003	42.0 ± 0.00	fam

Table 9: Feature ranking by average merit (correlation with target variable). Positive values indicate features positively correlated with annotator agreement, while negative values indicate features correlated with disagreement.