

Ondřej Herman

SENSE CLASSIFICATION FOR WORD SKETCHES

Abstract The Word Sketch is a core tool for modern corpus-based lexicography, providing a concise summary of a word's grammatical and collocational behavior. A longstanding limitation of standard Word Sketches is the lack of polysemy awareness: when a lexicographer examines a highly polysemous word such as *bank*, the resulting collocate lists mix financial terms like *account* and *deposit* with geographical terms like *river* and *sand*. In this paper, we present and evaluate a fully automated method for enriching Word Sketches with word sense information. Using the Adaptive Skip-gram model, we perform unsupervised word sense induction and project the induced senses onto the type-level collocate lists of a Word Sketch, so that collocates with sufficiently confident predictions are associated with the sense they most strongly represent. On a manually annotated test set of 23 polysemous English words, our method achieves a Precision of 0.64 and Recall of 0.77 ($F1 \approx 0.70$). The method requires no manual training data and is already deployed in Sketch Engine for eight languages. We describe the user interface, which allows lexicographers to filter Word Sketches by sense, and discuss implications for dictionary-writing workflows.

Keywords Word sense induction; word sketch; polysemy; corpus lexicography; sketch engine

1 Introduction

The Word Sketch (Kilgarrieff et al., 2014) is a widely used tool in corpus-based lexicography. For any given headword, it summarizes the word's combinatorial behavior, organizing collocates into grammatical relations such as modifiers, objects, and subjects. This allows lexicographers to quickly understand the usage patterns of a word and select appropriate collocates, examples, and usage notes for a dictionary entry.

However, an important shortcoming of the standard Word Sketch is its lack of treatment of polysemy. The collocate lists are computed over lemmas, so grammatical relations conflate the different senses of the headword. In the modifier relation of the noun *bank*, for instance, *commercial*, *central*, and *investment* sit alongside *steep*, *grassy*, and *opposite*, with no indication that they describe two unrelated meanings. The lexicographer is left to mentally disambiguate the senses, scanning long collocate lists to identify which entries belong together. For words with many senses, this is a time-consuming and error-prone process.

The value of addressing this limitation became apparent during the development of a semi-automatic Tagalog–English–Korean dictionary (Baisa et al., 2019), in which an earlier, experimental word sense induction method was used to assist annotators in grouping collocates by sense. The experience suggested that even imperfect sense clustering can assist lexicographic work, motivating us to develop a more reliable, scalable method.

In this paper, we present a method for enriching Word Sketches with automatically induced sense information. The approach uses the Adaptive Skip-gram model (Bartunov et al., 2016), an unsupervised word sense induction method that requires no manually labeled training data and is therefore applicable to any language with a sufficiently large corpus. We project the model's token-level sense predictions onto the type-level collocate lists of a Word Sketch and color-code each collocate by its dominant sense. We evaluate the result on a manually annotated test set of 23 polysemous English words, where it outperforms an earlier clustering-based approach (Herman et al., 2019).

2 Background

2.1 Word Sketches

A Word Sketch (Kilgarriff et al., 2014) is an automatically generated summary of a word's grammatical and collocational behavior in a given corpus. It is computed by the Sketch Engine corpus management system (Jakubíček et al., 2013) using a set of pattern-matching rules called a sketch grammar, which identify collocations and classify them into grammatical relations. The output is a set of triples of the form (*headword, relation, collocate*), each accompanied by an association score that reflects the statistical salience of the collocation.

For example, the Word Sketch for the noun *bank* may contain triples such as (*bank*, modifier, *central*), (*bank*, object_of, *rob*), or (*bank*, modifier, *river*). The collocates within each grammatical relation are ranked by their association score, so that the most characteristic collocates appear at the top.

Word Sketches are well established in practical lexicography: they have been used in the creation of numerous dictionaries and form a standard component of the dictionary-writing workflow in Sketch Engine.

2.2 Word Sense Induction

Word Sense Induction (WSI) is the task of automatically discovering the different senses of a word from unlabeled text. It differs from Word Sense Disambiguation (WSD), which assigns occurrences of a word to senses drawn from a fixed inventory such as WordNet. WSI is, in principle, more flexible: it does not require a predefined sense inventory and can discover senses that are not listed in any existing dictionary.

WSI methods build on the distributional hypothesis (Harris, 1954): words that occur in similar contexts tend to have similar meanings. The corollary is that different meanings of the same word tend to occur in different contexts, which makes it possible to identify word senses by clustering contextual patterns.

In this paper, we use *sense* to mean a cluster of usages occurring in related contexts. The sense induced by the WSI algorithm does not map directly to an entry in a fixed dictionary inventory. This approach treats word senses as abstractions over specific usage patterns (Kilgarriff, 1997).

A relevant observation for our work is the “one sense per collocation” hypothesis (Yarowsky, 1993), which states that a given collocation (e.g., *bank account*) very often corresponds to a single sense of the ambiguous word. If a collocate strongly prefers one sense of the headword, we can use the sense distribution observed in concordance lines to assign the collocate to that sense.

The WSI task has been the subject of several shared tasks in the SemEval evaluation campaigns, including SemEval-2007 Task 02 (Agirre & Soroa, 2007), SemEval-2010 Task 14 (Manandhar et al., 2010), and SemEval-2013 Task 13 (Jurgens & Klapaftis, 2013). These campaigns established standard evaluation methodologies and demonstrated the difficulty of the task, with the best systems achieving moderate agreement with gold-standard sense inventories.

An earlier approach to WSI for Word Sketches (Herman et al., 2019) used spectral clustering and HDBSCAN applied to averaged fastText word embeddings. While that method produced

usable sense clusters for many words, results were sensitive to parameter settings and often left a large proportion of collocates unclassified. Because of these limitations, the present work instead adopts the Adaptive Skip-gram model described in this paper.

3 Methodology

3.1 *The Adaptive Skip-gram Model*

The Adaptive Skip-gram model (Bartunov et al., 2016) is an extension of the Skip-gram word embedding model. While standard Skip-gram assigns a single vector to each word in the vocabulary, Adaptive Skip-gram is capable of learning multiple vectors for a single word, where each vector approximates a recurring contextual usage pattern.

The key advantage of Adaptive Skip-gram over other multi-sense embedding methods is that the number of senses per word is not fixed in advance. Instead, the model uses a non-parametric Bayesian framework to automatically determine how many senses each word needs based on the training data. A granularity parameter, denoted α (alpha), controls the tendency of the model to split words into multiple senses. A low α produces fewer, coarser senses; a high α encourages finer distinctions.

To handle the large corpora used in Sketch Engine (often exceeding one billion tokens), we re-implemented the Adaptive Skip-gram training algorithm in the Rust programming language, which allows us to train models on corpora of this scale within practical time limits. A fuller technical account of the implementation and its use for sense-shift detection is given by Herman (2026).

3.2 *Projecting Senses onto Word Sketches*

The method we use for projecting token-level WSI predictions onto the type-level Word Sketch collocate lists follows these steps:

1. **Model Training.** An Adaptive Skip-gram model is trained on the target corpus. The model learns multiple usage prototypes for polysemous words and a single vector for monosemous words. A single model is trained per language and applied to all corpora in that language, whether preloaded or user-uploaded.
2. **Instance Retrieval.** For a specific Word Sketch triple (*headword*, *relation*, *collocate*), we retrieve all positions in which this particular collocation occurs.
3. **Sense Prediction.** For each retrieved concordance line, the Adaptive Skip-gram model predicts the probability distribution over the senses of the headword given the surrounding context, so each concordance line generates a vector of sense probabilities.
4. **Aggregation and Thresholding.** The sense probability distributions are averaged across all concordance lines for a given collocate. This provides a single sense distribution for the collocation. Based on the “one sense per collocation” hypothesis, we expect most stable collocations to exhibit a strong preference for a single sense. If the averaged probability for a specific sense exceeds a predefined threshold, the collocate is assigned to that sense. If no sense crosses the threshold, or if the collocation is used in divergent contexts, the collocate remains unassigned (“No sense” is shown in the user interface).

To illustrate, consider the collocate *commercial* in the modifier relation for the noun *bank*. The model retrieves all corpus positions where *commercial* acts as a modifier for *bank*, predicts the distribution of senses of *bank* for each context, and averages them. If the result assigns 85% probability to a “financial institution” sense and 15% to a “river bank” sense, the financial sense exceeds the chosen threshold, so the collocate *commercial* is assigned to the financial sense of *bank*.

3.3 Sense Labeling

Because the method is unsupervised, the induced senses do not come with human-readable labels such as “financial institution” or “river bank.” Instead, we represent each sense by its nearest semantic neighbors in the Adaptive Skip-gram vector space. For the noun *bank*, Sense 1 might be represented by the words *institution*, *fund*, *lending*, *lender*, while Sense 2 is represented by *edge*, *along*, *river*, *side*. These labels are generated automatically and serve as hints that help the user identify what each sense corresponds to. While not always perfectly transparent, they are often sufficient for a lexicographer to understand the grouping.

4 User Interface and Visualization

We designed a visualization within Sketch Engine that makes the sense information accessible. Figure 1 shows the Word Sketch for the noun *bank* with the sense classification enabled.



FIGURE 1 Word Sketch for the noun *bank* with sense classification; colored bars indicate the assigned sense for each collocate, and the legend at the top shows automatically generated sense labels

At the top of the Word Sketch view, a “Senses” legend is displayed. Each induced sense is represented by the nearest-neighbor labels described above and is assigned a distinct color. In the case of *bank*, the system identifies five senses; alongside the financial and geographic senses, these include a payment card sense (represented by *credit*, *debit*, *account*, *card*).

In the Word Sketch columns below, each collocate is marked with a colored vertical bar corresponding to its assigned sense, so that the financial and geographic collocates of *bank* can be told apart at a glance. The user can also filter the Word Sketch by sense, hiding all collocates that do not belong to a selected sense.

5 Evaluation

5.1 Test Set

To evaluate the quality of the sense classification, we created a manually annotated test set of 23 known-polysemous English words, including nouns such as *bank*, *band*, *chip*, and *crane*, as well as two verbs, *fire* and *lie*. For each word, a single annotator inspected the top 150 collocates across all grammatical relations and grouped them by sense based on their corpus contexts, without consulting a fixed dictionary sense inventory. Only collocates that could be assigned to a specific sense with high confidence were annotated; ambiguous cases were excluded from the evaluation. The number of gold senses per word ranges from 2 (for *bark*, *beam*, *club*, *file*, *lie*) to 7 (for *base*), with a mean of 3.5 senses.

5.2 Metrics

We report the following evaluation metrics, defined in terms of the relationship between the gold-standard senses and the clusters produced by the system. **Gold Senses** is the number of senses we annotated for a word. **Found Senses** is the number of those senses that at least one cluster can be associated with, even if that cluster also covers other senses. **Pure Senses** are the Found Senses whose associated clusters contain no entries mapping to a different annotated sense. From the cluster side, **Mixed Clusters** are clusters that map to more than one annotated sense, while **Unique Clusters** map to at most one annotated sense. Together they account for every cluster the system produces.

Precision is the number of Pure Senses divided by the total number of clusters (Mixed Clusters + Unique Clusters). High precision means that the clusters the system produces tend to be sense-coherent.

Recall is the number of Found Senses divided by the number of Gold Senses. High recall means that the system identifies most of the senses present in the test set.

We also report **Pure Recall**, the number of Pure Senses divided by the number of Gold Senses. We summarize the results using the harmonic mean (F1) of Precision and Recall.

5.3 Results

We evaluated the system using the best-performing configuration from a parameter sweep: a model with an embedding dimension of 256, a context window of 14 tokens, and a sense assignment threshold of 0.75. The per-word results are shown in Table 1.

The system achieves a mean Precision of 0.64 and a mean Recall of 0.77, yielding an F1 score of approximately 0.70. Three words — *bark*, *file*, and *fire* — achieve perfect scores ($P = 1.0$, $R = 1.0$), meaning that the system recovers all gold senses with perfectly pure clusters. Several other words also achieve high scores, including *tank* ($P = 1.0$, $R = 0.80$), *bat* ($P = 1.0$, $R = 0.67$), and *bow* ($P = 1.0$, $R = 0.60$).

The most challenging words are *base* ($P = 0.17$, with 7 gold senses and many mixed clusters), *jumper* ($P = 0.00$, all collocates assigned to a single mixed cluster), and *change* ($P = 1.0$ but $R = 0.33$, only one of three senses is found).

TABLE 1 Evaluation results for 23 polysemous English words. P = Precision, R = Recall, PR = Pure Recall, the Mean row averages across the evaluated words

Word	Gold Senses	Found Senses	Pure Senses	P	R	PR
band-n	6	5	5	0.62	0.83	0.83
bank-n	4	2	2	1.00	0.50	0.50
bark-n	2	2	2	1.00	1.00	1.00
base-n	7	6	1	0.17	0.86	0.14
bat-n	3	2	2	1.00	0.67	0.67
beam-n	2	2	2	0.67	1.00	1.00
board-n	5	3	3	0.50	0.60	0.60
bow-n	5	3	3	1.00	0.60	0.60
change-n	3	1	1	1.00	0.33	0.33
chip-n	4	2	2	0.40	0.50	0.50
club-n	2	1	1	0.33	0.50	0.50
crane-n	3	3	1	0.33	1.00	0.33
deck-n	3	2	2	1.00	0.67	0.67
file-n	2	2	2	1.00	1.00	1.00
fire-v	3	3	3	1.00	1.00	1.00
iron-n	3	3	3	0.43	1.00	1.00
jumper-n	3	2	0	0.00	0.67	0.00
lie-v	2	2	2	0.40	1.00	1.00
party-n	4	3	3	0.60	0.75	0.75
pole-n	3	2	2	0.67	0.67	0.67
spring-n	4	3	3	0.43	0.75	0.75
tank-n	5	4	4	1.00	0.80	0.80
tear-n	3	3	1	0.20	1.00	0.33
Mean	3.52	2.65	2.17	0.64	0.77	0.65

5.4 Discussion

An F1 score of approximately 0.70 may seem modest by the standards of many NLP tasks, but it is useful as a pre-filtering step in a manual workflow. Our approach is to give the lexicographer an initial view to refine, instead of an undifferentiated collocate list. Perfect scores are unlikely, because the system groups usage patterns drawn from a limited sample of a language, without any task-specific tuning. The practical question is therefore whether the grouping is coherent enough to reduce manual effort while still allowing the editor to inspect the underlying corpus evidence.

Compared to an earlier approach based on HDBSCAN clustering of averaged fastText embeddings (Herman et al., 2019), the method based on Adaptive Skip-gram performs substantially better. The earlier method achieved a mean Precision of 0.43 and Recall of 0.63 ($F1 \approx 0.51$) on the same test set. The new method improves Precision by 0.21 and Recall by 0.14, raising F1 by 0.19. Moreover, we observed that the Adaptive Skip-gram method is more stable over different training runs and produces fewer unclassified collocates.

Note that WSI is a difficult task even for human annotators. In the ShadowSense evaluation (Herman & Jakubíček, 2024), the average inter-annotator Rand Index obtained for English was 0.82, which is high, but far from perfect.

6 Conclusion and Future Work

We have presented a method for enriching Word Sketches with automatically induced word sense information. The method is based on the Adaptive Skip-gram model, which learns multiple usage prototypes for polysemous words without requiring any manually labeled training data. By projecting sense predictions from concordance lines onto the type-level collocate lists of a Word Sketch, we enable a visualization in which collocates are color-coded by their dominant sense. On a test set of 23 polysemous English words, the method achieves an F1 score of approximately 0.70, representing a substantial improvement over the earlier clustering-based approach.

The feature is already deployed in Sketch Engine for all corpora in eight supported languages: English, Spanish, Italian, French, Ukrainian, Czech, Slovak, and Estonian. It groups collocates based on distributional semantics.

We are currently working on expanding the language coverage to all languages supported by Sketch Engine. We also plan to improve automatic sense labeling to produce more transparent labels, to explore more sophisticated models for sense induction, and to extend the sense awareness to other features within Sketch Engine. The first target is the Trends functionality, which tracks changes in word frequency over time. Combining the per-sense information from Adaptive Skip-gram with diachronic corpus data makes it possible to detect newly appearing word senses. The combination has already been investigated (Herman & Rychlý, 2026). The work that remains is primarily its engineering integration into Sketch Engine's production infrastructure.

References

- Agirre, E. & Soroa, A. (2007). SemEval-2007 Task 02: Evaluating word sense induction and discrimination systems. In E. Agirre, L. Márquez i Villodre & R. Wicentowski (Eds.), *Proceedings of the 4th International Workshop on Semantic Evaluations* (pp. 7–12). Association for Computational Linguistics.
- Baisa, V., Blahuš, M., Cukr, M., Herman, O., Jakubíček, M., Kovář, V., Medveď, M., Měchura, M., Rychlý, P. & Suchomel, V. (2019). Automating dictionary production: A Tagalog–English–Korean dictionary from scratch. In I. Kosem, T. Zingano Kuhn, M. Correia, J.P. Ferreria, M. Jansen, I. Pereira, J. Kallas, M. Jakubíček, S. Krek & C. Tiberius (Eds.), *Electronic Lexicography in the 21st Century: Proceedings of the eLex 2019 Conference* (pp. 805–818). Lexical Computing CZ, s.r.o.
- Bartunov, S., Kondrashkin, D., Osokin, A. & Vetrov, D. (2016). Breaking sticks and ambiguities with adaptive skip-gram. In A. Gretton & C. C. Robert (Eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics* (pp. 130–138). PMLR.
- Harris, Z. S. (1954). Distributional structure. *Word*, 10(2–3), 146–162.
- Herman, O. (2026). *Automatic Detection of Word Sense Shift* [Doctoral dissertation, Masaryk University].
- Herman, O. & Jakubíček, M. (2024). ShadowSense: A multi-annotated dataset for evaluating word sense induction. In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti & N. Xue (Eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (pp. 14763–14769). ELRA.

- Herman, O., Jakubíček, M., Rychlý, P. & Kovář, V. (2019). Word sense induction using Word Sketches. In C. Martín-Vide, M. Purver & S. Pollak (Eds.), *Statistical Language and Speech Processing: 7th International Conference, SLSP 2019, Proceedings* (pp. 83–91). Springer.
- Herman, O. & Rychlý, P. (2026). Detecting subtle sense shift with polysemy-aware trends. In V. Demberg, K. Inui & L. Marquez (Eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 60–65). Association for Computational Linguistics.
- Jakubíček, M., Kilgariff, A., Kovář, V., Rychlý, P. & Suchomel, V. (2013). The TenTen corpus family. In *7th International Corpus Linguistics Conference* (pp. 125–127).
- Jurgens, D. & Klapaftis, I. (2013). SemEval-2013 Task 13: Word sense induction for graded and non-graded senses. In S. Manandhar & D. Yuret (Eds.), *Second Joint Conference on Lexical and Computational Semantics, Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (pp. 290–299). Association for Computational Linguistics.
- Kilgariff, A. (1997). I don't believe in word senses. *Computers and the Humanities*, 31(2), 91–113.
- Kilgariff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P. & Suchomel, V. (2014). The Sketch Engine: Ten years on. *Lexicography*, 1(1), 7–36.
- Manandhar, S., Klapaftis, I. P., Dligach, D. & Pradhan, S. S. (2010). SemEval-2010 Task 14: Word sense induction & disambiguation. In K. Erk & C. Strapparava (Eds.), *Proceedings of the 5th International Workshop on Semantic Evaluation* (pp. 63–68). Association for Computational Linguistics.
- Yarowsky, D. (1993). One sense per collocation. In M. Kaufmann (Ed.), *Human Language Technology: Proceedings of a Workshop Held at Plainsboro* (pp. 266–271).

Contact information

Ondřej Herman
 Lexical Computing, Brno, Czech Republic
ondrej.herman@sketchengine.eu