Clustering Analysis for Error Detection in Named Entity Recognition Datasets

Matthew Flynn and Timothy Obiso and Sam Newman* **and Constantine Lignos**
Michtom School of Computer Science, Brandeis University
{matthewflynn,timothyobiso,lignos}@brandeis.edu
snewman.aa@gmail.com

Abstract

This paper introduces a method for the automatic detection of annotation errors and corrections in named entity recognition datasets using a novel two-stage dimension reduction of dense sentence embeddings. We first find the top-n principal components of an embedding and then use UMAP for second-stage, non-linear dimension reduction and clustering using different distance metrics. We analyze these clusters using silhouette scores to flag outlier mentions for correction. Using the corrections in the CoNLL# dataset as a benchmark, all of the top-five outliers needed correction, as did 7 of the top-10. This approach also identified 32 of the top-50 outlier mentions that are corrections. This method offers a relatively low-effort way to leverage text embeddings and dimensionality reduction to identify likely annotation errors. We release related code and data at https://github.com/bltlab/clustering-for-ner.

1 Introduction

Evaluating the quality of named entity recognition (NER) datasets is a labor-intensive process that requires multiple types of expertise. Adjudicators must check whether annotations are correct, whether an entity type forms a coherent category, and even whether the ontology's categories form semantically coherent groupings. This requires language fluency and, in many cases, domain expertise. Evaluating quality for a single dataset is feasible for a single language, but no single reviewer can assess and standardize across large, multilingual datasets. Thorough manual review becomes impractical.

We propose using silhouette scores on dimension-reduced sentence embeddings to automate the review and evaluation of NER datasets and their labels during the adjudication process or as post-hoc correction. The core intuition is that mentions of the same type should cluster closer together in embedding space. In these cases, the silhouette scores will be high; outlier mentions will have low or negative silhouette scores. These scores provide a quantitative signal for reviewers to determine edge cases or annotation errors. Silhouette scores act as an analogue for semantic coherence and can speak to how well the defined labels capture the natural semantic groups present in the data. This process can also highlight difficult cases, as a correct location mention that clusters near a person mention may reveal oddities in the annotation guidelines that may confuse annotators or systems.

Our approach uses principal component analysis (PCA, Hotelling, 1936) to first identify the top-n principal components of a sentence embedding. We then apply uniform manifold approximation (UMAP, McInnes et al., 2020) to map the principal components into clusters using a variety of distance metrics, including Euclidean, Chebyshev (Han et al., 2012), and Canberra (Lance and Williams, 1966), among others, as well as the Bray-Curtis dissimilarity (Ricotta and Podani, 2017). We compute the silhouette scores on the clusters formed by this dimension reduction pipeline and evaluate the scores to find and analyze the top-k outlier mentions.

We validate this approach with the CoNLL-03 English test set (Tjong Kim Sang and De Meulder, 2003) and compare outliers against corrections made in CoNLL# (Rueda et al., 2024). We show that examining the top-10 outliers in the original test set reliably identifies mentions that were later corrected. We also analyze the clusters formed by the corrected test set and demonstrate that the top-50 outliers can also reliably identify mentions that had been corrected.

Our main contributions are as follows. We propose methods for analyzing clusters and outliers of dimension-reduced sentence embeddings

*Independent researcher. Work completed while at Brandeis University.

Proceedings of the 20th Linguistic Annotation Workshop (LAW XX), pages 229–240

for named entity mentions using silhouette scores. We apply these methods on the CoNLL-03 English benchmark dataset using a variety of embedding models. Finally, we introduce `Clusters`, a command-line utility to facilitate the application and extension of these methods to further datasets and tasks.

2 Related Work

There has long been interest in evaluating the quality of datasets in NLP, as some rate of annotation errors is expected and generally accepted. Various methods and approaches to identifying and correcting such mistakes have included manual, semi-automated, and automated approaches. Dickinson (2015) provides an earlier review of such methods.

With the named entity recognition task, much work has focused on the enduring and popular CoNLL-03 English benchmark dataset to identify errors in annotation. Such work has been applied to the test set specifically (Stanislawek et al., 2019; Wang et al., 2019; Rueda et al., 2024), or the entire dataset (Reiss et al., 2020; Muthuraman et al., 2021). These efforts have produced and released corrected versions of the CoNLL-03 English test set, such as ReCoNLL (Fu et al., 2020) and CoNLL# (Rueda et al., 2024).

As the CoNLL-03 data is from the 1996 Reuters Corpus (Lewis et al., 2004), there are other concerns about how this publicly available newswire data may impact the performance of large, modern models that are trained on similar, and potentially the same, data. To evaluate this, the CoNLL++ dataset (Liu and Ritter, 2023) uses the CoNLL-03 annotation guidelines to create a modern version of the dataset using newswire data from 2020. This enables evaluation of how generations of state-of-the-art CoNLL models perform and generalize to modern data of the same domain and format. These authors also release an updated test set of CoNLL-03 English that removes the tabular, ticker-style data, such as sports scores, from the original.

Another popular dataset that has been reviewed is OntoNotes 5.0 (Weischedel, Ralph et al., 2013), which contains 17 different entity types in comparison to the four of CoNLL-03. Bernier-Colborne and Vajjala (2024) review and correct close to 10% of this dataset and observe that these corrections improve performance of models by an average of 1.23% in overall F1-scores, and they note an even larger improvement of more than 10% for certain entity types.

Similar work in identifying annotation errors has also been conducted on non-English datasets, such as Uyghur (Abudukelimu et al., 2018), Japanese (Ichihara et al., 2015), and Hindi (Saha et al., 2009).

Entity Type	Count
LOC	1633
MISC	754
ORG	1701
PER	1594

Table 1: Counts of mentions for each entity type in the corrected CoNLL# English test set

3 Methodology

We implemented an end-to-end pipeline to generate and evaluate clusters for CoNLL-formatted NER datasets. This includes loading and validating the data, generating embeddings, performing dimension reduction, plotting, and reporting.

3.1 Dataset

As prior work shows, the dataset from the CoNLL-03 shared task has proven popular over generations of models as a benchmark, and it has been a focus for analysis of annotation errors. The CoNLL-03 ontology consists of the Person ("PER"), Location ("LOC"), Organization ("ORG"), and Miscellaneous ("MISC") entity types, uses BIO encoding (Ramshaw and Marcus, 1995), and derives its text from the newswire domain (Lewis et al., 2004). We analyze the outliers identified in both the original English test set and in the corrected CoNLL# test set, which was chosen as it follows the original annotation guidelines when making corrections. Entity type counts for the mentions included in the CoNLL# test set are given in Table 1.

As CoNLL# also corrects sentence boundaries that were incorrectly split in the original English test set, it was necessary to align the original version to the corrected test set to ensure the correct mapping of mentions across datasets for analysis. This includes correcting mentions that were split across the original sentence boundaries.

3.2 Loading and Validating Data

To ensure data integrity and consistency, all data was loaded and validated using SeqScore (Palen-Michel et al., 2021; Lignos et al., 2023), an evaluation and validation toolkit for NER. The list of tokens for each mention was joined into a single

string for embedding, and it was mapped to its entity type. For example, a mention sequence with labels `['B-PER', 'I-PER', 'I-PER']` was mapped to `PER` for analysis and plotting.

3.3 Embedding Mentions

We selected a series of freely-available and self-hostable embedding models using the best scores on the Massive Multilingual Text Embedding Benchmark (MMTEB, Enevoldsen et al., 2025).[1] We also selected models based on foundational encoder models, such as SBERT (Reimers and Gurevych, 2019) and XLM-RoBERTa (Conneau et al., 2020). These models vary in architectures, context size, and the dimensions of their output embeddings.

For embedding, we used Qwen's Qwen3 embedding model family (Zhang et al., 2025),[2] Tencent's Gemma 3-based KaLM embedding model (Hu et al., 2025; Zhao et al., 2025),[3] intfloat's XLM-RoBERTa-based Multilingual E5 Instruct (Wang et al., 2024),[4] and the SBERT-based all-MiniLM-L6-v2.[5]

Our implementation supports any embedding model that can be reached at an OpenAI API-compatible embedding endpoint. We downloaded all models from HuggingFace and self-hosted them with vLLM (Kwon et al., 2023) to efficiently generate the embeddings and maintain a consistent embedding interface across models.

3.4 Prompts

All of the embedding models, with the exception of all-MiniLM-L6-v2, are instruction-tuned and expect a prompt as part of the embedding input. To standardize this input parameter, we used the same prompt template for all models, including all-MiniLM-L6-v2. This template includes a prompt instruction to inform the model of the purpose of the task, the sentence containing the mention as context, and the mention itself separately. We provide our full prompt template in the Appendix A.1.

3.5 Dimension Reduction

We reduced the dimensions of all embeddings before clustering and analysis to avoid the curse of dimensionality (Peng et al., 2025). We first found the top-n principal components of each embedding with scikit-learn's (Pedregosa et al., 2011) implementation of PCA. We then used UMAP to generate clusters from the n-dimension principal components using a variety of distance metrics, including Canberra (Lance and Williams, 1966), Correlation (Székely et al., 2007), Chebyshev, Cosine, Euclidean, Manhattan, and Minkowski (Han et al., 2012), in addition to the Bray-Curtis dissimilarity (Ricotta and Podani, 2017).

3.6 Silhouette Scores

With the clusters generated from the dimension-reduced embeddings, the silhouette score for each mention is calculated. This score is a metric for evaluating clusters, where each point's silhouette score $s(i)$ is the difference between its average distance from points in the next-nearest cluster $b(i)$, and its average distance from points in its own cluster $a(i)$. This difference is divided by the max of either $a(i)$ or $b(i)$ to obtain the respective point's silhouette score. This normalizes silhouette scores in the range $[-1, 1]$. A higher score implies better clustering (Shahapure and Nicholas, 2020), and, in general, a score $s(i) > 0.7$ signals strong clustering, and a score $0.7 > s(i) > 0.5$ is reasonable. A score close to 0.0 represents overlapping clustering.

$$a(i) = \frac{1}{|C(i)| - 1} \sum_{\substack{j \in C(i) \\ j \neq i}} d(i,j), \quad b(i) = \min_{C \neq C(i)} \frac{1}{|C|} \sum_{j \in C} d(i,j),$$
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad S = \frac{1}{N} \sum_{i=1}^{N} s(i). \tag{1}$$

We used these silhouette scores to identify outlier mentions for analysis.

4 Experiments

We completed a comprehensive grid search across all models summarized in Table 2 for both of our experiments. Our first experiment identifies mentions in the original CoNLL-03 English test set that were later corrected in CoNLL#, and the second identifies mentions in CoNLL# that are corrections.

For each experiment, we evaluate the top-performing models and configurations, the relative performance of the three different-sized embedding models in the Qwen3 family, and the relative performance of all models and architectures.

[1] huggingface.co/spaces/mteb/leaderboard
[2] huggingface.co/collections/Qwen/qwen3-embedding
[3] huggingface.co/tencent/KaLM-Embedding-Gemma3-12B-2511
[4] huggingface.co/intfloat/multilingual-e5-large-instruct
[5] huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Model	Parameters	Dimensions
Qwen/Qwen3-Embedding-0.6B	0.6B	1024
Qwen/Qwen3-Embedding-4B	4B	2048
Qwen/Qwen3-Embedding-8B	8B	4096
tencent/KaLM-Embedding-Gemma3-12B-2511	11.76B	3840
intfloat/multilingual-e5-large-instruct	0.6B	512
sentence-transformers/all-MiniLM-L6-v2	22.7M	384

Table 2: Comparison of embedding model sizes (parameter count) and embedding dimensions

Hyperparameter	Values
PCA Components	50, 75, 100
Clustering Components	16, 32, 64
Distance Metric	Bray-Curtis, Canberra, Chebyshev, Correlation, Cosine, Euclidean, Manhattan, Minkowski

Table 3: Hyperparameters for the grid search to find the optimal configuration for each model to identify mentions that were corrected

Hyperparameter	Values
PCA Implementation	Original, Kernel, Truncated
PCA Components	50, 75, 100
Clustering Components	16, 32, 64
Cluster Labels	True, False
Distance Metric	Bray-Curtis, Canberra, Chebyshev, Correlation, Cosine, Euclidean, Manhattan, Minkowski

Table 4: Hyperparameters for the grid search to find the optimal configuration for each model to identify mentions that are corrections

4.1 Identifying Mentions that were Corrected

We evaluate the performance of models at identifying how many of the top-five and top-10 of their outlier mentions were later corrected by CoNLL#. For this experiment, we held the number of UMAP cluster neighbors constant at 100, and the UMAP clustering used labeled data when learning the clusters. The hyperparameters for this experiment are summarized in Table 3.

4.2 Identifying Mentions that are Corrections

We evaluate performance by comparing how many of the top-50 outlier mentions identified by the model are corrections. As before, we held the number of UMAP cluster neighbors constant at 100; however, we varied the PCA implementation across the original, truncated, and kernel versions of PCA. We also varied the cluster learning with and without labels. The hyperparameters for this experiment are summarized in Table 4.

5 Results

Our experiments provide insight into optimal dimension reduction ratios, model architectures, embedding sizes, hyperparameter configurations, and distance metrics.

5.1 Identifying Mentions that were Corrected

We now report model and configuration performance on identifying mentions that were corrected in CoNLL#.

5.1.1 Top-Performing Models

Among its top-five and top-10 outliers, our best-performing model correctly identified five and seven mentions, respectively, that were later corrected. The three mentions that were not correctly identified as later corrected are all MISC mentions. Three different models reported among the top-five scores, with four of the top-five scores using Manhattan distance when learning clusters with UMAP. For the fourth and fifth best runs, using the KaLM-Embedding-Gemma3-12B-2511 model, the hyperparameters only differed in their distance metric. Interestingly, Canberra can be interpreted as a weighted Manhattan distance, and for this model, it performed slightly worse than its unweighted Manhattan counterpart. These results are summarized in Table 5, and 2d and 3d t-SNE (Hinton and Roweis, 2002) and UMAP projections are available in Appendix A.2.[6]

[6]While the authors were not aware of this at the time of submission, Peter Mayhew blogged about using 2d t-SNE projections of mention embeddings to help interpret tags and mentions from the CoNLL-03 English dataset. We refer the reader to his blog for more discussion: mayhewsw.github.io/2022/01/30/conll-span-embeddings/

Model	Metric	PCA Comp	Cluster Comp	Top-5	Top-10
Qwen3-Embedding-4B	Manhattan	50	32	5	7
multilingual-e5-large-instruct	Manhattan	50	32	5	5
multilingual-e5-large-instruct	Manhattan	50	16	5	5
KaLM-Embedding-Gemma3-12B-2511	Manhattan	50	16	4	6
KaLM-Embedding-Gemma3-12B-2511	Canberra	50	16	4	4

Table 5: Best-performing model configurations at identifying mentions that were later corrected

Entity Type	Mention	Score	CoNLL#	Corrected
ORG	Portsmouth	-0.9270	LOC	✓
ORG	Oxford	-0.9269	LOC	✓
LOC	DURBAN	-0.9235	PER	✓
LOC	SANTIAGO	-0.9218	PER	✓
ORG	DENVER	-0.9139	MISC	✓
ORG	GREEN BAY	-0.9041	MISC	✓
MISC	Lombardi Award	-0.9022	MISC	
ORG	OHIO STATE	-0.9021	MISC	✓
MISC	LOMBARDI AWARD	-0.9018	MISC	
MISC	AMERICAN	-0.9015	MISC	

Table 6: Top-10 outliers as identified by the top-performing Qwen3-Embedding-4B model

5.1.2 Qwen3 Embedding Model Family

The three different Qwen3 embedding models provide an interesting opportunity to compare the relative performance of the different sizes of this model family on the same task and using the same grid search to identify the optimal model configurations. Table 7 summarizes the Qwen3 scores for the first experiment.

Interestingly, the 4B model with 2048 embedding dimensions performed best among the Qwen3 embedding family, and the best overall for this experiment.

This finding runs counter to the intuition that the largest model with the most embedding dimensions should be able to perform better than a smaller counterpart, as is often the case in MMTEB performance. However, with the dimension-reduced embeddings, the 4B model performed best. Additionally, the optimal configuration for each model used a different distance metric. These metrics are related, however, as the Manhattan distance used by the 4B model is a Minkowski distance where $p = 1$, with the 8B model performing best with the default UMAP Minkowski configuration where $p = 2$. As noted above, the Canberra distance used by the 0.6B model can be interpreted as a weighted Manhattan distance. These results are summarized in Table 7.

5.1.3 All Architectures

We also evaluated the relative performance of each model family and architecture on this task, and the top score for each model is summarized in Table 8.

Consistent with other models, the best configuration for all-MiniLM-L6-v2, the only remaining unreported model for this first experiment, also reported its best run using Manhattan distance. This result also follows the intuition that the smallest, non-instruction-tuned embedding model would perform worse relative to the other larger, instruction-tuned embedding models.

5.2 Identifying Mentions that are Corrections

We now report model and configuration performance on identifying mentions that are corrections in CoNLL#, which includes some corrections made in prior attempts to correct the CoNLL-03 English data.

5.2.1 Top-Performing Models

Unlike the results in the first experiment, the best-performing model for identifying mentions in CoNLL# that are corrections is the small, non-instruction-tuned all-MiniLM-L6-v2. It reports all five of the top-five runs, and its top-scoring run correctly identifies 32 corrected mentions among its top-50 outliers. Similar to experiment one, Canberra distance also performs well at this task, as well as Chebyshev. These results are summarized in Table 9, and 2d and 3d t-SNE and UMAP projections are available in Appendix A.3.

Table 10 summarizes the top-10 outliers identified by all-MiniLM-L6-v2. All are MISC mentions that were corrected from either LOC or O mentions

Model	Metric	PCA Comp	Cluster Comp	Top-5	Top-10
Qwen3-Embedding-0.6B	Canberra	50	32	4	4
Qwen3-Embedding-4B	Manhattan	50	32	**5**	**7**
Qwen3-Embedding-8B	Minkowski	50	16	4	6

Table 7: Performance of Qwen3 embedding family at identifying mentions that were later corrected

Model	Metric	PCA Comp	Cluster Comp	Top-5	Top-10
Qwen3-Embedding-4B	Manhattan	50	32	5	7
multilingual-e5-large-instruct	Manhattan	50	32	5	5
KaLM-Embedding-Gemma3-12B-2511	Manhattan	50	16	4	6
Qwen3-Embedding-8B	Minkowski	50	16	4	6
Qwen3-Embedding-0.6B	Canberra	50	32	4	4
all-MiniLM-L6-v2	Manhattan	50	16	3	3

Table 8: Performance of each embedding model family and architecture at identifying mentions that were later corrected

in CoNLL#. This pattern suggests that MISC is the noisiest of the labels in the CoNLL-03 ontology and that, while all of these mentions form a cluster given their similar or identical textual content, they are distant from the core MISC cluster. Notably, these MISC mentions have no context, as they are entire sentences in the CoNLL dataset and exist among the ticker-style sports scores present in the test set.

5.2.2 Qwen3 Embedding Model Family

As before, we compared the relative performance of the three sizes of Qwen3 embedding. Unlike the first experiment, the largest model with 8B parameters significantly outperformed the smaller models. It neared the performance of all-MiniLM-L6-v2, identifying 30 corrected mentions among its top-50 outliers. These results are summarized in Table 11.

All three Qwen models performed best with the Bray-Curtis dissimilarity metric, which differs from the classic or weighted Minkowski distance metrics that score best for other models. Bray-Curtis is a popular statistical metric in ecology and biology, and it quantifies the difference between two different samples.

5.2.3 All Architectures

These results show that the best-performing model overall is all-MiniLM-L6-v2, which is also the smallest of the models we evaluated. Conversely, KaLM-Embedding-Gemma3-12B-2511 is the largest of the models we evaluated and while it boasts the strongest performance on MMTEB, it performed significantly worse than all other models across any configuration for this second experiment. It identified just five corrected mentions among its top-50 outliers. These results are summarized in Table 12.

6 Discussion

This method provides insight into automated detection of annotation errors in NER, as well as the relative performance of model sizes and architectures in the clustering of mentions. It can also be useful in designing ontologies and creating datasets.

6.1 Identifying Annotation Errors

Our implementation lists the top-*k* outlier mentions for each entity type in a dataset and overall. These outliers are useful for surfacing potential annotation errors for closer review without needing to review each mention in a dataset individually, and for evaluating a corrected dataset to ensure corrected mentions are consistent with existing annotations.

Treating CoNLL# as the gold standard, our experiments show that our method can identify among its top-five and top-10 outliers, five and seven labels that were corrected. Likewise, the same pipeline when applied to the gold CoNLL# data can identify 32 mentions that are corrections among its top-50 outliers.

Our experiments also reveal that different model sizes and architectures perform differently on this task. In identifying mentions that were corrected, larger models showed superior performance using weighted and unweighted Minkowski distance metrics where $p = 1$ or $p = 2$. However, at identifying corrected mentions, the smallest model, all-MiniLM-L6-v2 with only 22.7M parameters and embeddings with 384 dimensions, performed best. This finding makes it possible to use this method

Model	Metric	Labels	PCA Implementation	PCA Comp	Cluster Comp	Corrected
all-MiniLM-L6-v2	Canberra	False	PCA	50	32	32
all-MiniLM-L6-v2	Chebyshev	False	Truncated	50	16	31
all-MiniLM-L6-v2	Canberra	False	Kernel	50	32	31
all-MiniLM-L6-v2	Chebyshev	False	PCA	50	32	31
all-MiniLM-L6-v2	Chebyshev	False	PCA	50	64	31

Table 9: Best-performing model configurations at identifying mentions that are corrections

Entity Type	Mention	Score	CoNLL#	Corrected
LOC	ATLANTIC DIVISION	-0.2328	MISC	✓
LOC	ATLANTIC DIVISION	-0.2317	MISC	✓
LOC	PACIFIC DIVISION	-0.2311	MISC	✓
LOC	PACIFIC DIVISION	-0.2310	MISC	✓
O	CENTRAL DIVISION	-0.2299	MISC	✓
O	CENTRAL DIVISION	-0.2288	MISC	✓
O	CENTRAL DIVISION	-0.2274	MISC	✓
O	EASTERN DIVISION	-0.2239	MISC	✓
O	EASTERN DIVISION	-0.2238	MISC	✓
O	WESTERN DIVISION	-0.2238	MISC	✓

Table 10: Top-10 outliers as identified by all-MiniLM-L6-v2

with limited computing resources, and even on just a CPU. Its top scores were also with Canberra and Chebyshev distance, while the large, instruction-tuned pooling models performed best with the Bray-Curtis dissimilarity metric at this task.

6.2 Dataset Creation

Dataset creation can be a time-consuming and burdensome process, especially in domain-specific, multilingual, or low-resourced settings, where finding domain experts or native speakers available for quality data annotation work can be difficult, and potentially expensive. Using the top-k outlier mentions for each entity type during the annotation workflow can reveal errors and tough mentions. These tough mentions can also provide insight and data points for any labels that may be unclear or borderline, which can help improve and clarify annotation guidelines and the ontology.

Strategies, such as the MATTER lifecycle (Pustejovsky and Stubbs, 2012) or some other workflow, exist for this process. These strategies include ontology creation and review, in addition to adjudication and annotation, as part of the overall dataset creation process. Data annotation tools, such as brat (Stenetorp et al., 2012) and Label Studio (Tkachenko et al., 2020-2025), among others, also exist to facilitate and streamline this process. Integrating our method as part of the workflow can reduce burden as it quickly surfaces the top-k outliers for each label at any step. With this quantitative information and supporting visual projections, dataset authors can improve the quality of their annotated datasets and provide consistent feedback to annotators during the annotation process as part of a continuous integration workflow and without creating additional burdens for reviewers.

6.3 MISC Mentions

Being a catchall entity type, it follows that MISC mentions are the noisiest and least semantically coherent. All of the missed mentions in our first experiment were MISC, and our second experiment shows that among the top-10 outlier mentions identified by the top-performing all-MiniLM-L6-v2 model, all were very similar, and sometimes identical, MISC mentions. All were corrected labels as well. This suggests that MISC mentions were indeed the noisiest and least coherent.

Our method can quickly surface all outlier MISC mentions, report silhouette scores for all their mentions for comparison, reveal their relative coherence given the MISC type's catchall definition, and quantitatively show the relative value of MISC in an ontology. Our method can help ontology designers decide if it is better to split MISC into other entity types, or to drop it altogether.

7 Future Work and Conclusion

We show that using silhouette scores and dimension-reduced embeddings to evaluate ontologies and datasets can automatically identify annotation errors in original datasets and corrected mentions in updated datasets. To facilitate such

Model	Metric	Labels	PCAI	PCAC	CC	Corrected
Qwen3-Embedding-0.6B	Bray-Curtis	False	PCA	75	16	12
Qwen3-Embedding-4B	Bray-Curtis	False	Kernel	75	16	24
Qwen3-Embedding-8B	Bray-Curtis	False	PCA	50	16	**30**

Table 11: Performance of Qwen3 embedding family at identifying mentions that are corrections

Model	Metric	Labels	PCA Impl	PCA Comp	Cluster Comp	Corr
all-MiniLM-L6-v2	Canberra	False	PCA	50	32	32
Qwen3-Embedding-8B	Bray-Curtis	False	Kernel	100	16	30
multilingual-e5-large-instruct	Chebyshev	False	Truncated	50	32	28
Qwen3-Embedding-4B	Bray-Curtis	False	Kernel	75	16	24
Qwen3-Embedding-0.6B	Bray-Curtis	False	PCA	75	16	12
KaLM-Embedding-Gemma3-12B-2511	Bray-Curtis	False	PCA	75	32	5

Table 12: Performance of all architectures at identifying mentions that are corrections

future work, we release the `Clusters` utility under a permissive open source license.

Future work can extend this approach and the `Clusters` utility to other NLP domains that use labeled data. It is also possible to extend the two-stage dimension reduction to unlabeled data, such as that used to train LLMs, in order to analyze the semantic distribution of the training data, and to help ensure it is balanced and not biased to any domain or topic. The approach can also be applied for analysis in any of the numerous domains that use embeddings, such as dense information retrieval.

Limitations

With dimension reduction using PCA and UMAP, there is inherently some amount of information loss. Depending on the task or embedding size, that may be undesirable and would limit the applicability of our method.

Due to the limited public availability of previously corrected NER datasets, we were forced to confine our experiments to English data.

Modern language models are often trained on large and diverse corpora, with popular encoding models trained on 100 languages (Conneau et al., 2020) and some modern LLMs trained on 140+ languages (Team et al., 2025). The issue of low-resourced languages, however, remains. This clustering analysis is ultimately reliant on the performance of embedding models to generate the embeddings that are used for the silhouette score analysis. If the encoding model does not perform well or understand a low-resourced language corpus, the performance of this approach may be less reliable.

Likewise, embedding models and LLMs alike inherit the biases of their training data, and that can also impact the usefulness of their generated embeddings at this task.

Acknowledgments

Constantine Lignos was partially supported by the grant *Improving Relevance and Recovery by Extracting Latent Query Structure* from eBay to Brandeis University.

References

Halidanmu Abudukelimu, Abudoukelimu Abulizi, Boliang Zhang, Xiaoman Pan, Di Lu, Heng Ji, and Yang Liu. 2018. Error analysis of Uyghur name tagging: Language-specific techniques and remaining challenges. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Gabriel Bernier-Colborne and Sowmya Vajjala. 2024. Annotation Errors and NER: A Study with OntoNotes 5.0. *Preprint*, arXiv:2406.19172.

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.

Markus Dickinson. 2015. Detection of Annotation Errors in Corpora. *Language and Linguistics Compass*, 9(3):119–138.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, and 67 others. 2025.

MMTEB: Massive Multilingual Text Embedding Benchmark. *Preprint*, arXiv:2502.13595.

Jinlan Fu, Pengfei Liu, and Qi Zhang. 2020. Rethinking Generalization of Neural Models: A Named Entity Recognition Case Study. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7732–7739.

Jiawei Han, Micheline Kamber, and Jian Pei. 2012. 2 - getting to know your data. In *Data Mining: Concepts and Techniques (Third Edition)*, third edition edition, The Morgan Kaufmann Series in Data Management Systems, pages 39–82. Morgan Kaufmann, Boston.

Geoffrey E Hinton and Sam Roweis. 2002. Stochastic Neighbor Embedding. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press.

Harold Hotelling. 1936. Relations Between Two Sets of Variates. *Biometrika*, 28(3/4):321–377.

Xinshuo Hu, Zifei Shan, Xinping Zhao, Zetian Sun, Zhenyu Liu, Dongfang Li, Shaolin Ye, Xinyuan Wei, Qian Chen, Baotian Hu, Haofen Wang, Jun Yu, and Min Zhang. 2025. KaLM-Embedding: Superior Training Data Brings A Stronger Embedding Model. *Preprint*, arXiv:2501.01028.

Masaaki Ichihara, Kanako Komiya, Tomoya Iwakura, and Maiko Yamazaki. 2015. Error analysis of named entity recognition in BCCWJ. *Recall*, 61:2641.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with Pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Godfrey N Lance and William T Williams. 1966. Computer programs for hierarchical polythetic classification ("similarity analyses"). *The Computer Journal*, 9(1):60–64.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *J. Mach. Learn. Res.*, 5:361–397.

Constantine Lignos, Maya Kruse, and Andrew Rueda. 2023. Improving NER research workflows with SeqScore. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 147–152, Singapore. Association for Computational Linguistics.

Shuheng Liu and Alan Ritter. 2023. Do CoNLL-2003 named entity taggers still work well in 2023? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8254–8271, Toronto, Canada. Association for Computational Linguistics.

Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *Preprint*, arXiv:1802.03426.

Karthik Muthuraman, Frederick Reiss, Hong Xu, Bryan Cutler, and Zachary Eichenberger. 2021. Data cleaning tools for token classification tasks. In *Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances*, pages 59–61, Online. Association for Computational Linguistics.

Chester Palen-Michel, Nolan Holley, and Constantine Lignos. 2021. SeqScore: Addressing barriers to reproducible named entity recognition evaluation. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 40–50, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Dehua Peng, Zhipeng Gui, and Huayi Wu. 2025. Interpreting the curse of dimensionality from distance concentration and manifold effect. *Preprint*, arXiv:2401.00422.

James Pustejovsky and Amber Stubbs. 2012. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. " O'Reilly Media, Inc.".

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Preprint*, arXiv:1908.10084.

Frederick Reiss, Hong Xu, Bryan Cutler, Karthik Muthuraman, and Zachary Eichenberger. 2020. Identifying incorrect labels in the CoNLL-2003 corpus. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 215–226, Online. Association for Computational Linguistics.

C. Ricotta and J. Podani. 2017. On some properties of the Bray-Curtis dissimilarity and their ecological meaning. *Ecological Complexity*, 31:201–205.

Andrew Rueda, Elena Alvarez-Mellado, and Constantine Lignos. 2024. CoNLL#: Fine-grained error analysis and a corrected test set for CoNLL-03 English. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3718–3728, Torino, Italia. ELRA and ICCL.

Sujan Kumar Saha, Sudeshna Sarkar, and Pabitra Mitra. 2009. Hindi named entity annotation error detection and correction. *Language Forum*, 35(2):73–93. Publisher: Bahri Publications.

Ketan Rajshekhar Shahapure and Charles Nicholas. 2020. Cluster Quality Analysis Using Silhouette Score. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 747–748.

Tomasz Stanislawek, Anna Wróblewska, Alicja Wójcicka, Daniel Ziembicki, and Przemyslaw Biecek. 2019. Named entity recognition - is there a glass ceiling? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 624–633, Hong Kong, China. Association for Computational Linguistics.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations Session at EACL 2012*, Avignon, France. Association for Computational Linguistics.

Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. 2007. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6).

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 Technical Report. *Preprint*, arXiv:2503.19786.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020-2025. Label Studio: Data labeling software. Open source software available from https://github.com/HumanSignal/label-studio.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 Text Embeddings: A Technical Report. *arXiv preprint arXiv:2402.05672*.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5154–5163, Hong Kong, China. Association for Computational Linguistics.

Weischedel, Ralph, Palmer, Martha, Marcus, Mitchell, Hovy, Eduard, Pradhan, Sameer, Ramshaw, Lance, Xue, Nianwen, Taylor, Ann, Kaufman, Jeff, Franchini, Michelle, El-Bachouti, Mohammed, Belvin, Robert, and Houston, Ann. 2013. OntoNotes Release 5.0.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *arXiv preprint arXiv:2506.05176*.

Xinping Zhao, Xinshuo Hu, Zifei Shan, Shouzheng Huang, Yao Zhou, Xin Zhang, Zetian Sun, Zhenyu Liu, Dongfang Li, Xinyuan Wei, Youcheng Pan, Yang Xiang, Meishan Zhang, Haofen Wang, Jun Yu, Baotian Hu, and Min Zhang. 2025. KaLM-Embedding-v2: Superior Training Techniques and Data Inspire A Versatile Embedding Model. *Preprint*, arXiv:2506.20923.

A Appendix

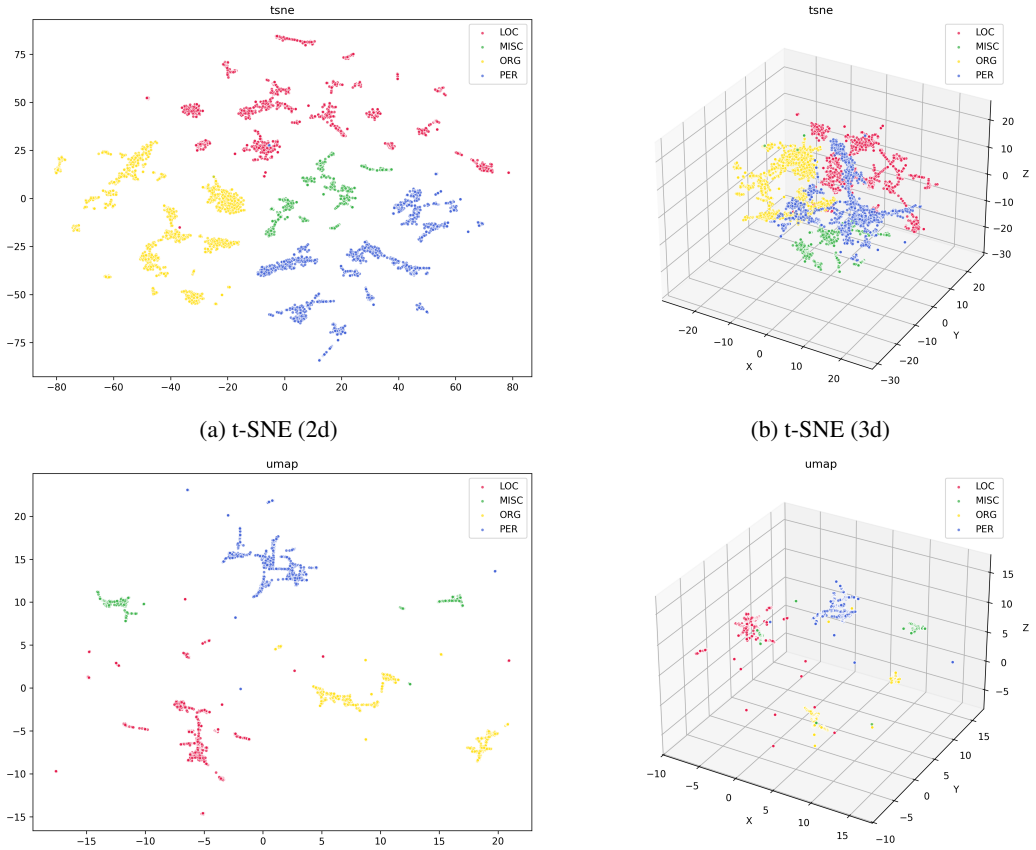
A.1 Prompts

```
PROMPT: Embed the following for use
in clustering analysis with dimension
reduction
### Context
{context}

### Mention
{mention}
```

In this template, "context" represents the entire sentence that contains the mention, and "mention" represents the mention itself.

A.2 Projections of Mentions that were Corrected



(a) t-SNE (2d)

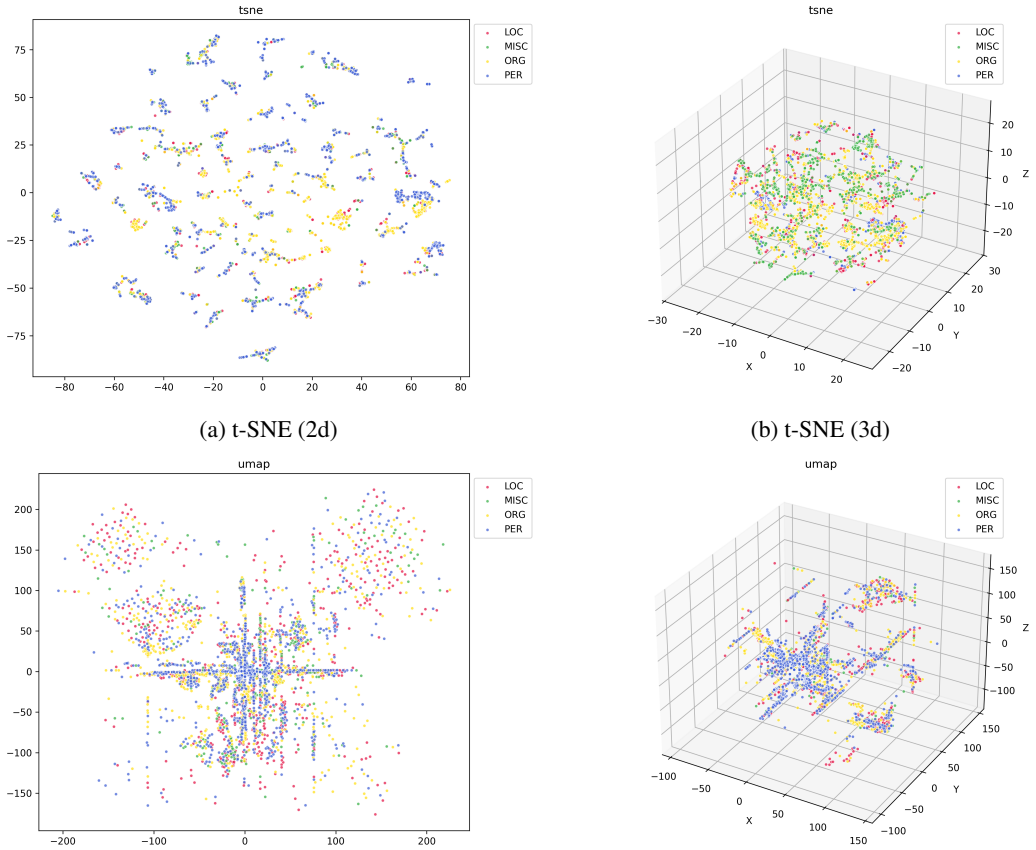
(b) t-SNE (3d)

(c) UMAP (2d)

(d) UMAP (3d)

Figure 1: t-SNE and UMAP projections in 2d and 3d for the best-performing run with Qwen3-Embedding-4B at identifying mentions that were corrected

A.3 Projections of Mentions that are Corrections



(a) t-SNE (2d)

(b) t-SNE (3d)

(c) UMAP (2d)

(d) UMAP (3d)

Figure 2: t-SNE and UMAP projections in 2d and 3d for the best-performing run with all-MiniLM-L6-v2 at identifying mentions that are corrections