Improving Word Sense Induction through Adversarial Forgetting of Morphosyntactic Information

Deniz Ekin Yavas[*,1], **Timothée Bernard**[2], **Laura Kallmeyer**[1], **Benoît Crabbé**[2]
Heinrich Heine University Düsseldorf[1], Université Paris Cité[2]
{deniz.yavas, laura.kallmeyer}@hhu.de[1]
{timothee.bernard, benoit.crabbe}@u-paris.fr[2]

Abstract

This paper addresses the problem of word sense induction (WSI) via clustering of word embeddings. It starts from the hypothesis that contextualized word representations obtained from pre-trained language models (LMs), while being a valuable source for WSI, encode more information than what is necessary for the identification of word senses and some of this information affect the performance negatively in unsupervised settings. We investigate whether using contextualized representations that are invariant to these 'nuisance features' can increase WSI performance. For this purpose, we propose an adaptation of the adversarial training framework proposed by Jaiswal et al. (2020) to erase specific information from the representations of LMs, thereby creating *feature-invariant representations*. We experiment with erasing (i) morphological and (ii) syntactic features. The results of subsequent clustering for WSI show that these features indeed act like noise: Using feature-invariant representations, compared to using the original representations, increases clustering-based WSI performance. Furthermore, we provide an in-depth analysis of how the information about the syntactic and morphological features of words relate to and affect WSI performance.

1 Introduction

Words in their different senses occur in different contexts. Contextualized word representations obtained from transformer based pre-trained language models (LMs) such as BERT (Devlin et al., 2019) are especially suitable for Word Sense Disambiguation (WSD) because they capture the sentential context of a word and thereby oftentimes allow to distinguish different senses of a word. They have indeed been successfully used for WSD in recent work (Hadiwinoto et al., 2019; Loureiro et al., 2021; Vandenbussche et al., 2021).

However, in both unsupervised WSD, where the goal is to identify the instances of a specific sense, and word sense induction (WSI), which allows the discovery of novel senses, using the LM representations alone does not yield satisfactory results (Pilehvar and Camacho-Collados, 2019; Samih and Kallmeyer, 2023). In both, similarity of the representations plays a crucial role, and this similarity is determined by many features. Indeed, the contextualized representation of a word usually encodes a wide range of linguistic information about the word in its context, such as its syntactic function, its morphological properties, its position, its casing, and the identity of its neighbouring words (Sajjad et al., 2022). However, most of the encoded information that determines the similarity of the representations is not relevant to word senses (Yavas, 2024). Note that this is not a problem for supervised WSD, as a supervised model can learn to ignore those features that are not discriminative for word senses.

Building on these insights, we focus on WSI and aim at investigating the relationship between specific types of information encoded in contextualized representations of LMs and WSI performance. Concretely, we examine whether erasing certain information from the representations of LMs can lead to an increase in performance for a simple clustering-based WSI system. Our investigation examines two types of information that have been observed to affect the word sense clustering performance negatively. Yavas (2024) have shown that in word sense clustering with BERT representations on SemCor (Miller et al., 1993), word instances are frequently clustered together based on the similarities between their morphological and syntactic features (more specifically, *syntactic role* of the word) rather than their semantic similarities. For example, past tense instances of a specific verb, or all instances of a specific noun occurring as direct objects, are clustered together.

We adapt the adversarial training framework of

[*]This work was conducted during the author's visit to Université Paris Cité.

*Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 238 - 251

Jaiswal et al. (2020) in order to train a forget-gate that erases information from the representations of LMs, resulting in *feature-invariant representations*. We experiment with the BERT model and create feature-invariant representations for both of the above-mentioned types of features (morphological and syntactic). Finally, we evaluate the performance of WSI on SemCor with different feature-invariant representations, comparing them to the original word representations obtained from BERT. Furthermore, we conduct an in-depth analysis of how the information about the syntactic and morphological features of words relate to and affect WSI performance.[1]

Our results show that words' morphological and syntactic features indeed act like noise that negatively affects clustering performance and syntax- and morphology-invariant representations are better suited to WSI than the original BERT representations. Furthermore, a more detailed analysis of the relation between these information types and WSI performance shows that even though syntactic features are more correlated to word senses than morphological features are, they still affect the WSI performance negatively overall.

This paper makes several contributions. First, we propose an adaptation of the framework proposed by Jaiswal et al. (2020) to erase unwanted information from the representations of LMs. Secondly, we use this method to generate syntax- and morphology-invariant representations from the word representations of the BERT model and achieve better performance in clustering-based WSI. Lastly, we provide an in-depth analysis of how the morphological and syntactic features of words affect WSI performance.

The paper is structured as follows: We review related work in Section 2, then we introduce our framework for creating feature-invariant representations in Section 3 and report the results of the creation process. Finally in Section 4, we report the experiments on WSI with an analysis of the relation between the information types and WSI performance.

[1]We also experimented with positional information. In our experiments, we successfully removed the positional information from the representations, however, these representations exhibited unexpected behaviour in clustering experiments. As a result, we decided to exclude this feature type. We intend to investigate the underlying reasons in the future.

2 Related Work

Word Sense and Information Encoded in Contextualized Representations. Contextualized representations of pre-trained LMs encode more contextual information than what is necessary for the identification of word senses and this information can affect the similarity of the representations in an unwanted way. Sajjad et al. (2022) have shown that semantic, morphological, and syntactic concepts are encoded in contextualized representations. These concepts include words' POS tags, CCG super-tags, ngrams, casings, WordNet concepts, and so on. Furthermore, clustering of contextualized word representations reveal these similarities between the words.

In their detailed qualitative analysis, Yavas (2024) have shown that word sense clustering with BERT's representations on SemCor is heavily and negatively affected by information encoded in the representations from the sentence context that is insignificant to WSD, such as some morphological features of the words, their syntactic role, the presence of some punctuation marks and function words in the sentence (e.g. 'not', 'then', etc.). In the present study, we aim to investigate whether the effects of some of these features can be controlled and whether doing so can increase performance in WSI on the same dataset.

Similar effects have been found in lexical semantic change detection. Laicher et al. (2021) have observed that BERT representations are influenced by the orthographic form of words. Consequently, this affects how the representations are clustered. They have shown that removal of this bias increases the clustering performance. In order to do so, they preprocess the input data by lemmatizing the target word in each sentence before feeding it to the model.

Adversarial Training for Invariant Representation Learning. Invariant representation learning aims to create representations that do not encode certain unwanted features of data, such as nuisances, biases, or domain-specific features. Nuisances are features in the data that have no or little relevance to the task but influence model performance, like facial expressions in face recognition (Bronstein et al., 2003) or orientation in image recognition (Khotanzad and Hong, 1990). The creation of representations invariant to nuisances aims to increase model performance and robustness.

In this study, we consider morphological and

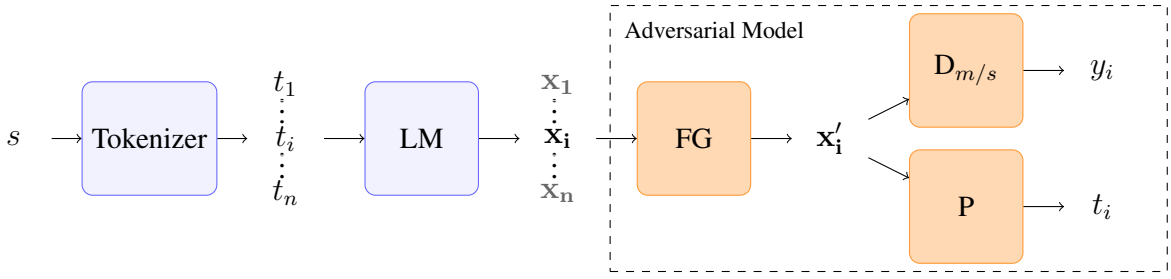


Figure 1: Framework for training a forget-gate (FG) to create feature-invariant representations from the representations of a pre-trained model. The FG is trained as a part of an adversarial model. The *LM* is only used as a feature extractor and its parameters are frozen.

syntactic features of words as nuisance features for WSI because they are not directly related to different senses of words but affect performance. We acknowledge that the syntactic features of words are, to some extent, relevant to WSI, but while these features may aid in identifying the senses of some words, we hypothesize that it introduces noise overall.

Invariant representation learning is widely used to create representations that are invariant to nuisance features in Computer Vision (Louizos et al., 2017; Xie et al., 2017; Jaiswal et al., 2018, 2020). However, in NLP, most applications of this technique center around learning domain-invariant representations (Louizos et al., 2017; Jaiswal et al., 2018; Peng and Zhang, 2020; Xin et al., 2022) . As for our knowledge, there has been no attempt to create contextualized representations invariant to any linguistic information.

Jaiswal et al. (2020) propose a framework for learning invariant representations through adversarial training in a Computer Vision context. They train an encoder (and a decoder) to generate representations for a set of inputs. At the same time, a *forget-gate* is trained to generate masks meant to be applied to the representations in order to create invariant representations. The forget-gate is trained as part of an adversarial model, in which a *discriminator* predicts the unwanted information from the masked representation while a *predictor* predicts some task-related information. Our framework for learning invariant representations is inspired by Jaiswal et al. (2020) while showing clear differences. We do not train an encoder-decoder model but, instead, we utilize LMs and create invariant representations from their representations. Furthermore, forgetting is not done by masking but by transforming the LM representations through a feed-forward network.

3 Creating Feature-Invariant Representations via Adversarial Training

In order to obtain contextualized representations that are invariant to certain features, we propose to add a forget-gate on top of a pre-trained LM. The forget-gate applies a nonlinear transformation and thereby selectively removes the unwanted information from the original contextualized representations. It is trained as a part of an adversarial model inspired by Jaiswal et al. (2020). Concretely, we train two forget-gates to create feature-invariant representations for syntactic and morphological features. We will refer to the respective resulting representations as *syntax-invariant*, and *morphology-invariant representations*.

3.1 Framework

Our framework is illustrated in Figure 1. We define a neural network named *forget-gate* (FG). This network is implemented as a feedforward neural network with two hidden layers with ReLU activation function. It applies a transformation to the representations obtained from a pre-trained LM to create representations that are invariant to specific information. The input of FG ($\mathbf{x}$) is the representation we aim to transform, namely the token embedding from the pre-trained LM. The output of FG ($\mathbf{x'}$) is the feature-invariant representation.

Given a sentence s, we first tokenize it with the LM tokenizer and then pass the tokenized sentence $(t_1, ..., t_i, ..., t_n)$ to the LM in order to extract the token embeddings $(\mathbf{x_1}, ..., \mathbf{x_i}, ..., \mathbf{x_n})$, obtained from the last layer of the LM (i.e., the LM acts as a feature extractor). Each of these embeddings constitutes one input data point to the FG. We use the BERT (*base-cased*) model.[2]

[2]The embeddings are extracted using the Transformers library (Wolf et al., 2020).

The forget-gate FG is trained as part of an adversarial model with two auxiliary modules; a discriminator (D) and a predictor (P). During training, the representation produced by the forget-gate ($\mathbf{x}'_\mathbf{i} = FG(\mathbf{x_i})$ for token t_i) is given to P and D. D is tasked with probing for unwanted information (some label y_i for token t_i) in the embedding $\mathbf{x}'_\mathbf{i}$, and P is tasked with recovering the identity of the token (t_i) from $\mathbf{x}'_\mathbf{i}$. The adversarial model is trained on the representations of both masked and unmasked tokens (i.e., we sometimes substitute the [MASK] token for t_i in the input of the LM).

The training of the adversarial model alternates between three types of batch, each batch containing training data for only one of the three subnetworks of the adversarial model. On the first type of batch, the parameters of D, on the second type of batch, the parameters of P, and on the third type of batch, the parameters of FG are updated. There are two batches of the first type (for D) for one batch of the second and one batch of the third type. So, D is trained more than the rest of the network. The parameters of FG are updated based on the combined loss L_{FG} of D and P as indicated in (1). The loss of D is given as negative since we want to increase it.[3]

$$L_{FG}(x, y, t) = -L_D(D(x'), y) + L_P(P(x'), t) \quad (1)$$

For each feature type (morphological and syntactic), we train an adversarial model with a unique discriminator to obtain a feature-specific forget-gate.[4] The discriminators, D_m and D_s, are trained as classification models and towards labelling tokens with POS tags (from the Penn Treebank tagset, Marcus et al., 1993 — these tags are fine-grained and provide morphological information such as number for nouns and tense for verbs) and (incoming) dependency labels respectively; the training labels are predicted, see Section 3.3. For each feature type, the corresponding discriminator aims to probe for this specific feature, while the forget-gate simultaneously aims to erase it. The details about the architecture of the different modules of the adversarial model and their loss functions can be found in Appendix A.

We train the adversarial models for 800 epochs creating checkpoints every 100th epoch and select the best checkpoint a posteriori based on the evaluation results (see next section). The details about the hyperparameters and the training of the adversarial model can be found in Appendix B. As a result of training the two adversarial models, we obtain two different forget gates, FG_m and FG_s. These forget gates, when applied to a BERT word embedding, yield the respective feature-invariant representations.

3.2 Evaluation

In order to evaluate this method, we create representations using the trained forget-gates for each type of information and use these representations to train two models from scratch: one for word (i.e., token) prediction and the other for unwanted information probing. The performance of these models on the test data tells us whether the feature-invariant representation creation was successful.

We compare the performances of these models to a lower and an upper bounds. The upper bound for a task is defined as the performance of a similar system trained using the original BERT representations. The lower bound is defined differently for the two types of information. For syntactic information, the probing model is trained using the non-contextualized word representations used by BERT as input to its first layer. For morphological information, the lower bound is given by the most frequent POS baseline. It is calculated for each grammatical category (i.e. noun, verb, and so on.) by predicting the most frequent POS tag for that category and averaged for all categories.

The probing models for morphological and syntactic features are similar to the respective discriminators in the adversarial models: They share the same architecture, training and test data. The word predictors are also similar to the respective predictors in the adversarial models, in regard to their architecture, training and test data.

We compare the performances of different models on the test data. We use perplexity as the metric to evaluate the word predictors and accuracy for the probing models. We select the best forget-gate for each feature type aiming at a low probing performance (close to the lower bound): We evaluate all checkpoints and pick the forget-gate with the lowest probing scores (if not lower than the lower

[3] In Jaiswal et al. (2020)'s framework, forgetting is not done by using the opposite of D's loss on the correct labels, but by using D's loss on random labels.

[4] We target morphological and syntactic information independently, even though theoretically, they are interrelated. However, this does not affect the relevance of our method, but only some linguistic interpretations of the results.

bound). Details about the hyperparameters and the training of the lower bound and upper models, evaluation models (probing and word prediction), and the selected forget-gates are given in Appendix B.[5]

3.3 Data

For the training and evaluation of the models, we use the Brown corpus (Kučera et al., 1967). We extract the token representations by BERT of each sentence. These token representations are then used as the input for the forget-gate. Since words are tokenized into WordPiece subwords by the BERT tokenizer, we work with these subwords rather than entire words.

In cases a word is split in multiple parts, we only take the first subword into account, i.e. we only erase information from the first subword a word and we only use this subword for WSI. We expect the first subwords to encode more relevant information for WSI because they are more likely to align with the stems of the words as opposed to suffixes (e.g. 'booklets' is tokenized into 'booklet' and 's' by the model tokenizer).

We assign two labels to each token; one for the discriminator (or probe) and another for the predictor. The predictor's label corresponds to the token ID assigned by the BERT's tokenizer to the token. The discriminator's label varies depending on the feature type: the label is either the label of the incoming syntactic dependency or the POS tag (of the word associated with the token). We get these labels automatically using spaCy.[6]

The dataset for morphological information only contains tokens of words belonging to grammatical categories that exhibit inflection in English: nouns, verbs, adjectives, adverbs, and pronouns. No such restrictions apply to syntactic information. This process yields datasets containing 2,341,954 tokens for syntactic information, and 1,315,988 tokens for morphological information. All datasets are split to train, development, and test data with the ratio 80:10:10.

3.4 Results

Both feature-invariant representations achieve good results in word prediction and probing tasks; the unwanted information is erased from the representations while their word prediction capabilities are intact. The lower bounds and upper bounds for all tasks and the evaluation results for feature-invariant representations can be seen in Table 1.

	Word Prediction	Probing
Syntactic Information		
Upper Bound	3.0	85.0
Lower Bound	-	70.2
Invariant Rep.	4.1	72.1
Morphological Information		
Upper Bound	3.0	89.1
Lower Bound	-	62.1
Invariant Rep.	7.1	75.9

Table 1: Evaluation results for the feature-invariant representations with comparison to the upper and lower bounds of the tasks. *Accuracy* is given for probing (in this context, lower is better) and *perplexity* is given for word prediction results (lower is better).

Erasing morphological information impacts the performance of word prediction more. This is expected because the morphological features of words (for instance grammatical number for nouns) are strongly correlated with their word forms.

4 Word Sense Induction Performance

Our aim is to investigate whether using feature-invariant contextualized representations can improve WSI performance. For this purpose, we compare the performance on WSI of three variants of the same system, respectively using three different representations; 1) the original contextualized representations of the BERT model, 2) syntax-invariant, and 3) morphology-invariant contextualized representations, where the latter two are obtained by applying our trained forget-gates FG_m and FG_s to the BERT representations. Furthermore, we provide a detailed analysis of the relation between the morphological and syntactic features of words and WSI and how the erasure of this information affects the WSI performance.

4.1 Data

We evaluate our WSI systems on SemCor. SemCor is based on a subset of the Brown Corpus and it provides sentences in which a word, the *target word*, is labelled with a WordNet sense (Fellbaum, 1998) as shown in (1). We focus on nouns and verbs and exclude other grammatical categories. We further exclude the words that have only one sense, and the senses that occur in less than 10 sentences.

[5]The code for this project is available at: https://github.com/yavasde/Adversarial-Forgetting-of-Morphosyntactic-Information.

[6]https://spacy.io/, model: *en_core_web_trf*.

(1) officer:

a. "An officer with a squad of men had been waiting on the bank."
(*officer.n.01*)

b. "And the policy officer has the hounds of time snapping at his heels."
(*officer.n.02*)

One of the advantages of using SemCor for WSI is that it is a subset of a bigger corpus (Brown Corpus), that we can use to train the forget-gates. The forget-gates are then trained on the same kind of texts that the ones used for WSI, which helps ensuring the quality of the invariant representations used during clustering. There is no methodological problem in doing so as the gold clusters are not used at any time during the training of the forget-gates. This approach can be applied to any dataset by training a forget-gate and performing WSI on the same data. Note that while the training of the forget-gates requires feature annotation, this does not limit the applicability of our approach as we perform it automatically.

4.2 Method

We cluster instances of words using their representations (BERT, syntax-invariant or morphology-invariant) in the sentences. We tokenize each sentence with the BERT tokenizer and give the tokenized sentence to the model to extract the representations of the target word from the last layer of the BERT model. In cases where the words are tokenized into subwords, we only use the first subword token. We create feature-invariant representations from the original representations of BERT for each information type using the information-specific forget-gate (FG_s or FG_m). We apply normalization to all embeddings before clustering.

We use the K-Means algorithm for clustering.[7] K-Means requires the cluster number as a hyperparameter. To determine the optimal number of clusters for each word, we run the algorithm with different cluster numbers between 2 and 6 and select the one with the highest silhouette score.[8]

We evaluate the clustering performance by comparing cluster assignments and the WordNet senses of word instances and average the result over all words. The evaluation metric used is the *Adjusted Rand Index* (ARI) (Hubert and Arabie, 1985). ARI measures the similarity between two clusterings by counting the pairs that are assigned to the same or different clusters in both the gold clusters and predicted clusters. It is adjusted to account for chance agreement and gives a score between -1 and 1 where 1 indicates perfect agreement between the two clusterings, while scores below 0 suggest that the match is worse than random chance. For the ARI formula and different clustering evaluation metrics see Appendix C.

We compare the WSI performance with 3 different types of representations. We run the clustering algorithm 5 times for each type of representation with different random states. We report the mean of 5 runs. We apply unpaired t-test to determine if the performance difference is statistically significant. We compare the overall performance and the performance based on grammatical category (verbs and nouns).

	Overall	nouns	verbs
BERT	0.210 ($8\text{x}10^{-4}$)	0.251 ($1\text{x}10^{-3}$)	0.174 ($1\text{x}10^{-3}$)
Syn-Inv	**0.221** ($1\text{x}10^{-3}$)	**0.263** ($4\text{x}10^{-4}$)	**0.185** ($2\text{x}10^{-3}$)
Morph-Inv	**0.232** ($1\text{x}10^{-3}$)	**0.267** ($1\text{x}10^{-3}$)	**0.201** ($2\text{x}10^{-3}$)

Table 2: WSI performance with different representation types. The performance is measured using *ARI*. Results are presented for all words in the dataset, as well as for verbs and nouns individually. The mean results over 5 runs are given with standard deviation in brackets. The scores that surpass the BERT representations are in bold.

4.3 Results

Results are shown in Table 2. WSI is performed better with feature-invariant representations than with the original BERT representations for both feature types, with statistically significant differences observed through unpaired t-tests (p-value: 0.0001). The best results are obtained with morphology-invariant representations overall. The largest gain in the performance happens for verbs with morphology-invariant representations. For a more detailed evaluation of the clustering performance using different metrics see Appendix C. Further analysis of the specific cases and reasons behind performance increases and decreases are addressed in the following section.

[7] The algorithms are implemented using the Scikit-learn library (Pedregosa et al., 2011).

[8] The silhouette score measures how similar a sample is to its cluster compared to other clusters. It's calculated for each sample and then averaged for the entire dataset.

	Syntax						Morphology					
	#	T_L	T_U	Sense	BERT	Invariant	#	T_L	T_U	Sense	BERT	Invariant
MI	-	9.4	27.8	14.7	26.4	23.1	-	4.7	40.5	9.4	41.8	25.4
Case 1	59	-	-	-	**0.34**	0.33	11	-	-	-	**0.66**	065
Case 2	34	-	-	-	0.07	**0.10**	80	-	-	-	0.04	**0.09**
Case 3	53	-	-	-	**0.19**	0.18	55	-	-	-	0.28	**0.29**

Table 3: Relation between the feature types and the WSI performance. MI scores for the association between linguistic features and sense or cluster assignments are given. T_L and T_U refers to the lower and upper threshold for MI. For each *Case*, ARI scores for BERT clusters and the clusters formed by the feature-invariant representations are given. Performance increases are in bold.

4.4 Analysis of the Relation Between the Information Types and WSI Performance

Our aim is to determine in which cases the erasure of these information types helps the WSI process. More specifically, we aim to investigate whether, for individual words, word senses are distinguishable by the word's morphological and syntactic features and therefore, whether the existence or the erasure of the related information helps the WSI process. Even though the overall WSI performance improves with feature-invariant representations, it is possible that for some words, the information erased is actually useful for sense identification. In these cases, the information erasure would negatively affect the WSI process.

In order to investigate this, we identify the three following cases and assess the performance with the original BERT representations and feature-invariant representations for each case:

- *Case 1:* The senses of a word are distinguishable by the word's morphological, or syntactic features. In this case, we expect the performance with invariant representations to be lower than with the original BERT representations.
- *Case 2:* The senses of a word are *not* distinguishable by the word's morphological, or syntactic features, but clusters of BERT representations are distinguishable by these features — which then can be assumed to be noise for clustering-based WSI. In this case, we expect the performance with invariant representations to be higher than with the original BERT representations.
- *Case 3:* The senses of a word are *not* distinguishable by the morphological, or syntactic features of the word, and clusters of BERT representations are also *not* distinguishable by these features. In this case, we expect the performance with invariant representations to be the same as with the original BERT representations.

4.4.1 Method

We measure the association between the features of the word instances and their sense or cluster assignments using *Mutual Information* (MI).[9] We use this information to automatically categorize words into the three cases outlined above.

In order to determine the features of the word instances, we use again the POS tags and dependency labels obtained from spaCy. [10] We refer to them as *linguistic labels*. For each word and for each type of feature we compute three MI scores. Firstly, we calculate the MI score between the linguistic labels of the instances and their gold WordNet sense labels (*Sense MI*). Secondly, we calculate the MI score between the linguistic labels of the instances and their cluster labels, considering the clusters formed by the BERT representations *(BERT MI)*. Lastly, we assess the MI score between the linguistic labels of the instances and their cluster labels, considering this time the clusters formed by the feature-invariant representations (*Invariant MI*).

We compare the MI scores to lower (T_L) and upper thresholds (T_U). The lower and upper threshold are calculated for each feature type as the first quartile and third quartile for all MI scores for this feature. We interpret scores below the lower threshold as indicating no association, and scores above

[9] The mutual information between two variables X and Y is defined as follows:

$$MI(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) \quad (2)$$

[10] For some instances of words, the target word is not found in the sentence due to a lemmatization error. We discard these words and experiment with 540 words in total (out of 567).

the upper threshold as indicating an association. We then automatically categorise word types. *Case 1* words have high Sense MI scores, *Case 2* words have low Sense MI and high BERT MI scores, and finally, *Case 3* words have low Sense MI and low BERT MI scores. We compute the average ARI score for each word within each case and compare their performances.

4.4.2 Results

The results of the analysis can be seen in Table 3. MI scores show that the linguistic labels are more strongly associated with BERT clusters than with the sense groups. This suggests that these features are dominant in the BERT clusters more than necessary. This and the fact that ARI performance is lowest for the *Case 2* words indicate that these features introduce noise that affects WSI performance negatively. With the feature-invariant representations, this effect is limited to some extent.

Regarding different *Cases*, the results mostly align with our expectations. Clustering performance with *Case 1* words is slightly higher with original BERT representations. Clustering performance with *Case 2* words is increased with feature-invariant representations. However, the increase for *Case 2* words is much higher than the decrease for *Case 1* words, showing that the erasure of syntactic and morphological information benefits the WSI performance overall. Finally, with *Case 3* words, there is a slight increase or decrease in performance depending on the different feature types.

Regarding different feature types, we observe that the morphological features of words introduce a lot of noise to WSI performance (Sense MI vs. BERT MI). Only for 11 words (out of 540), morphological features of words are found to be associated with different senses (*Case 1*). For 80 words, these features are found to be associated with different BERT clusters, even though they are not relevant to different senses (*Case 2*), therefore introducing noise. Similarly, the average BERT MI score for morphological features surpasses the upper threshold (T_U) of association, showing that there is a high level of association between morphological features and BERT clusters. Conversely, syntactic features of words have more relevance to word senses. For 59 words, these features show associations with different senses, and both the Sense MI score is higher, and the difference between Sense and BERT MI scores is lower for this feature type. These differences are also evident

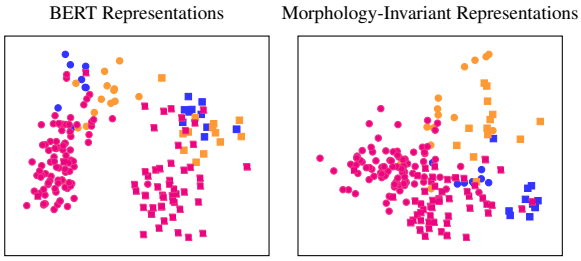


Figure 2: PCA visualizations of BERT representations (left) and morphology-invariant representations (right) of different sense instances of 'area'. Different data point colors refer to different senses. Different marker styles refer to instances with different morphological features, i.e. grammatical number; *circles* for *singular nouns*, *squares* for *plural nouns*.

in the WSI performance. Erasing morphological information benefits WSI performance more than erasing syntactic information (Table 2).

Let us illustrate these findings with a few examples. The noun 'area' has 3 senses in the data. WSI performs worse with BERT representations than with morphology-invariant representations on this word (BERT ARI: 0.03, INV ARI: 0.49). With BERT representations, we observe that two clusters are formed and that they are formed mostly based on the grammatical number of the instances, although there is no association between grammatical number and the senses of the word (Sense MI: 0.0, BERT MI: 57.7, Invariant MI: 4.2). With morphology-invariant representations, we observe that this pattern is broken. Grammatical number does not affect the similarity of the representations and the instances of each sense are closer to each other. Singular and plural instances of the third sense are successfully clustered together. Even though first and second sense instances form only one cluster, instances of each sense are closer to each other and the senses are more separable. See Figure 2 for the PCA visualization of the different representations of 'area' instances. For a more detailed plot see Figure 4 in Appendix D.

Let's consider a contrasting example. The noun 'field' has 4 senses in the data. WSI performs better with BERT representations than syntax-invariant representations (BERT ARI: 0.62, INV ARI: 0.26) and senses are associated with the syntactic features of the word (Sense MI: 40.3, BERT MI: 43.7, Invariant MI: 11.0). The 3rd sense of 'field' has the meaning 'somewhere (away from a studio or office or library or laboratory) where practical work is done or data is collected' and almost all of its

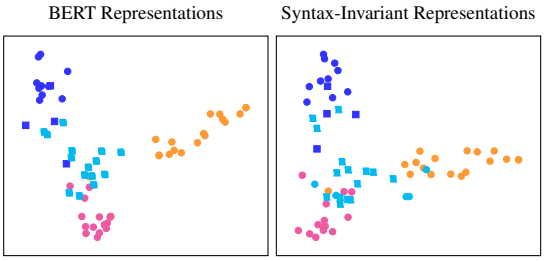


Figure 3: PCA visualizations of BERT representations (left) and syntax-invariant representations (right) of different sense instances of 'field'. Different data point colors refer to different senses. *Light blue* data points represent the 3rd sense instances of the noun. Different marker styles refer to instances with different syntactic roles; *squares* for *compounds heads*, *circles* for others.

instances are heads of a compound as in example (2).

(2) a. They will give suggestions that can be worked up into **field** procedures.
b. Actually, none of these papers says much directly about **field** techniques.

The 3rd sense instances are clustered together when BERT representations are used. After the erasure of syntactic information, their representations are closer to the representations of other sense instances. As a result, they are clustered with other sense instances when syntax-invariant representations are used, resulting in poor performance. See Figure 3 for the PCA visualization of the different representations of 'field' instances. For a more detailed plot see Figure 5 in Appendix D.

5 Conclusion

We adapt the framework proposed by Jaiswal et al. (2020) in order to erase specific information from the representations of LMs. With this method, we create two types of representations from BERT embeddings: invariant to either (i) morphological features or (ii) syntactic features. Our results show that the resulting feature-invariant representations are more suitable for the WSI task. Furthermore, we show that even though some syntactic features provide valuable information for WSI, both types of features introduce noise that, overall, negatively impacts the performance of clustering-based WSI.

6 Acknowledgement

This study is funded by Labex EFL and the DFG project "Coercion and Copredication as Flexible Frame Composition". We would like to thank the anonymous reviewers for their valuable comments.

References

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12:461–486.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. 2003. Expression-invariant 3d face recognition. In *international conference on Audio-and video-based biometric person authentication*, pages 62–70. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized word representations. 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 5297–5306, Hong Kong, China. Association for Computational Linguistics.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2:193–218.

Ayush Jaiswal, Daniel Moyer, Greg Ver Steeg, Wael AbdAlmageed, and Premkumar Natarajan. 2020. Invariant representations through adversarial forgetting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4272–4279.

Ayush Jaiswal, Rex Yue Wu, Wael Abd-Almageed, and Prem Natarajan. 2018. Unsupervised adversarial invariance. *Advances in neural information processing systems*, 31.

Alireza Khotanzad and Yaw Hua Hong. 1990. Rotation invariant image recognition using features selected via a systematic method. *Pattern recognition*, 23(10):1089–1101.

Henry Kučera, Winthrop Francis, William Freeman Twaddell, Mary Lois Marckworth, Laura M Bell, and John Bissell Carroll. 1967. Computational analysis of present-day american english. *Brown University Press, Providence, RI*.

Severin Laicher, Sinan Kurtyigit, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2021. Explaining and improving BERT performance on lexical semantic change detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 192–202, Online. Association for Computational Linguistics.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. 2017. The variational fair autoencoder.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. Analysis and evaluation of language models for word sense disambiguation. *Computational Linguistics*, 47(2):387–443.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Minlong Peng and Qi Zhang. 2020. Weighed domain-invariant representation learning for cross-domain sentiment analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 251–265, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Firoj Alam, Abdul Khan, and Jia Xu. 2022. Analyzing encoded concepts in transformer language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3082–3101, Seattle, United States. Association for Computational Linguistics.

Younes Samih and Laura Kallmeyer. 2023. Unsupervised semantic frame induction revisited. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 89–93, Nancy, France. Association for Computational Linguistics.

Pierre-Yves Vandenbussche, Tony Scerri, and Ron Daniel Jr. 2021. Word sense disambiguation with transformer models. In *Proceedings of the 6th Workshop on Semantic Deep Learning (SemDeep-6)*, pages 7–12, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. 2017. Controllable invariance through adversarial feature learning. *Advances in neural information processing systems*, 30.

Ji Xin, Chenyan Xiong, Ashwin Srinivasan, Ankita Sharma, Damien Jose, and Paul Bennett. 2022. Zero-shot dense retrieval with momentum adversarial domain invariant representations. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4008–4020, Dublin, Ireland. Association for Computational Linguistics.

Deniz Ekin Yavas. 2024. Assessing the significance of encoded information in contextualized representations to word sense disambiguation. In *Proceedings of the Third Workshop on Understanding Implicit and Underspecified Language*, pages 42–53, Malta. Association for Computational Linguistics.

Ying Zhao and George Karypis. 2001. Criterion functions for document clustering: Experiments and analysis.

A Model Architectures

- **Discriminator for Synactic Information:** 2-layer nonlinear neural network for classification. ReLU is used as the activation function. Given a token embedding, it predicts the dependency label of the token. Output dimension is the number of classes. Cross-entropy loss is used as the loss function and Adam is used as the optimizer.
- **Discriminator for Morphological Information:** 2-layer nonlinear neural network for classification. ReLU is used as the activation function. Given a token embedding, it predicts the fine-grained POS tag of the token. Output dimension is the number of classes found in the dataset. Cross-entropy loss is used as the loss function and Adam is used as the optimizer.
- **Predictor:** 3-layer linear neural network that maps the token embedding to the vocabulary of BERT with size 30522. 2nd layer is for normalization and drop-out (0.1) is applied before the output layer. Cross-entropy loss is used as the loss function and Adam is used as the optimizer.
- **Forget-Gate:** 3-layer nonliner neural network that transforms the input embedding. ReLU is used as the activation function. Cross-entropy loss is used as the loss function and Adam is used as the optimizer.

B Model Training Procedures

Adversarial Models. All adversarial models are trained with batch size 128 and learning rate 10^{-6} for the predictor, 10^{-5} for the discriminator, and 10^{-4} for the forget gate. All the models are trained for 800 epochs.

Upper Bounds.

- **Probing for syntactic information:** 74 epochs, batch size 128, learning rate 10^{-5}.
- **Probing for morphological information:** 66 epochs, batch size 128, learning rate 10^{-5}.
- **Word prediction:** 132 epochs, batch size 128, learning rate 10^{-6}.

Lower Bounds.

- **Probing for syntactic information:** 9 epochs, batch size 128, learning rate 10^{-4}.

Feature-Invariant Representations Evaluation.

- **Morphology-Invariant Representations:** The word predictor is trained for 30 epochs with batch size 128, learning rate 10^{-5}. The probing model is trained for 22 epochs with batch size 128, learning rate 10^{-4}. The best forget-gate is obtained from the 400th epoch of the adversarial model's training.
- **Syntax-Invariant Representations:** The word predictor is trained for 20 epochs with batch size 128, learning rate 10^{-5}. The probing model is trained for 37 epochs with batch size 128, learning rate 10^{-4}. The best forget-gate is obtained from the 500th epoch of the adversarial model's training.

C Clustering Performance Details

We evaluate the clustering performance using metrics from 4 different categories based on the categorization in Amigó et al. (2009) because different categories have different strengths in measuring clustering quality; metrics based on set matching (*Purity*, *Inverse Purity* (Zhao and Karypis, 2001) and their harmonic mean *PIF*), metrics based on entropy (*V-Measure* (Rosenberg and Hirschberg, 2007)), metrics based on counting pairs (*Adjusted Rand Index* (Hubert and Arabie, 1985)), and BCubed metrics (*BCubed Precision*, *Recall* and *F-score* (Bagga and Baldwin, 1998)).

C being the set of clusters and L being the true grouping, *Purity* and *Inverse Purity* are calculated as follows:

$$\text{Purity} = \sum_i \frac{|C_i|}{N} \max_j \text{Precision}(C_i, L_j) \quad (3)$$

$$\text{Inverse Purity} = \sum_i \frac{|L_i|}{N} \max_j \text{Precision}(L_i, C_j) \quad (4)$$

V-measure is calculated based on the homogeneity and completeness. Homogeneity measures how much each cluster contains only data points that are members of a single class. Completeness measures how much all data points that are members of a given class are assigned to the same cluster. It is calculated as follows:

	ARI	V-M	PU	IPU	PIF	P-Bcubed	R-Bcubed	F-Bcubed
BERT	0.210 (0.0008)	0.265 (0.0007)	0.732 (0.0007)	0.658 (0.001)	0.670 (0.001)	0.652 (0.0006)	0.575 (0.0008)	0.580 (0.0004)
Syn-Invariant	0.221 (0.001)	0.274 (0.001)	0.732 (0.0006)	0.682 (0.0009)	0.684 (0.0008)	0.653 (0.0007)	0.599 (0.001)	0.594 (0.0008)
Morph-Invariant	0.232 (0.001)	0.283 (0.0006)	0.736 (0.0004)	0.683 (0.001)	0.688 (0.0007)	0.657 (0.0004)	0.598 (0.0008)	0.597 (0.0005)

Table 4: Clustering evaluation results with different representations with different metrics for all words in the data. The mean results over 5 runs are given with standard deviation in brackets.

	ARI	V-M	PU	IPU	PIF	P-Bcubed	R-Bcubed	F-Bcubed
BERT	0.251 (0.001)	0.309 (0.001)	0.769 (0.0007)	0.689 (0.002)	0.707 (0.001)	0.699 (0.0004)	0.607 (0.001)	0.623 (0.0008)
Syn-Invariant	0.263 (0.0004)	0.320 (0.0005)	0.772 (0.0006)	0.682 (0.001)	0.706 (0.001)	0.703 (0.0003)	0.600 (0.001)	0.622 (0.0009)
Morph-Invariant	0.267 (0.001)	0.322 (0.001)	0.772 (0.001)	0.682 (0.001)	0.705 (0.001)	0.704 (0.001)	0.598 (0.001)	0.620 (0.001)

Table 5: Clustering evaluation results with different representations with different metrics for nouns. The mean results over 5 runs are given with standard deviation in brackets.

	ARI	V-M	PU	IPU	PIF	P-Bcubed	R-Bcubed	F-Bcubed
BERT	0.174 (0.001)	0.227 (0.001)	0.699 (0.001)	0.632 (0.001)	0.638 (0.0008)	0.611 (0.001)	0.547 (0.0006)	0.542 (0.0001)
Syn-Invariant	0.185 (0.002)	0.233 (0.002)	0.698 (0.001)	0.681 (0.002)	0.665 (0.001)	0.608 (0.001)	0.598 (0.001)	0.570 (0.001)
Morph-Invariant	0.201 (0.002)	0.248 (0.002)	0.705 (0.001)	0.684 (0.002)	0.672 (0.002)	0.616 (0.001)	0.599 (0.002)	0.576 (0.002)

Table 6: Clustering evaluation results with different representations with different metrics for verbs. The mean results over 5 runs are given with standard deviation in brackets.

$$\mathrm{V} = 2 \times \frac{\text{Homogeneity} \times \text{Completeness}}{\text{Homogeneity} + \text{Completeness}} \quad (5)$$

Adjusted Rand Index (ARI) adjusts the *Rand Index* (RI) to account for chance agreement. *RI* calculates the similarity between two clusterings by considering pairs of samples and determining whether they are assigned to the same cluster or different clusters in both clusterings. They are calculated as follows:

$$\mathrm{RI} = \frac{\text{correct similar pairs + correct dissimilar pairs}}{\text{total number of pairs}} \quad (6)$$

$$\mathrm{ARI} = \frac{\max(RI) - \text{Expected\_RI}}{RI - \text{Expected\_RI}} \quad (7)$$

Correctness is the relation between e and e' in the distribution, where $C(e)$ denotes the cluster and $L(e)$ true grouping of the item. Correctness means that both items have the same category and belong to the same cluster. The overall *Precision BCubed* and *Recall BCubed* are obtained by averaging the precision and recall scores of all items in the dataset as follows:

$$\text{Precision BCubed} = Avg_e[Avg'_{e \cdot C(e)=C(e')}[Correctness(e, e')]] \quad (8)$$

$$\text{Recall BCubed} = Avg_e[Avg'_{e \cdot L(e)=L(e')}[Correctness(e, e')]] \quad (9)$$

The detailed evaluation of the clustering performance with different metrics for all words can be seen in Table 4, for nouns in Table 5 and verbs in Table 6. The mean results over 5 runs are given.

D Clustering Visualizations

The PCA visualizations of the BERT representations and morphology-invariant representations of 'area' instances can be seen in Figure 4. Similarly, the PCA visualizations of the BERT representations and syntax-invariant representations of 'field' instances can be seen in Figure 5.

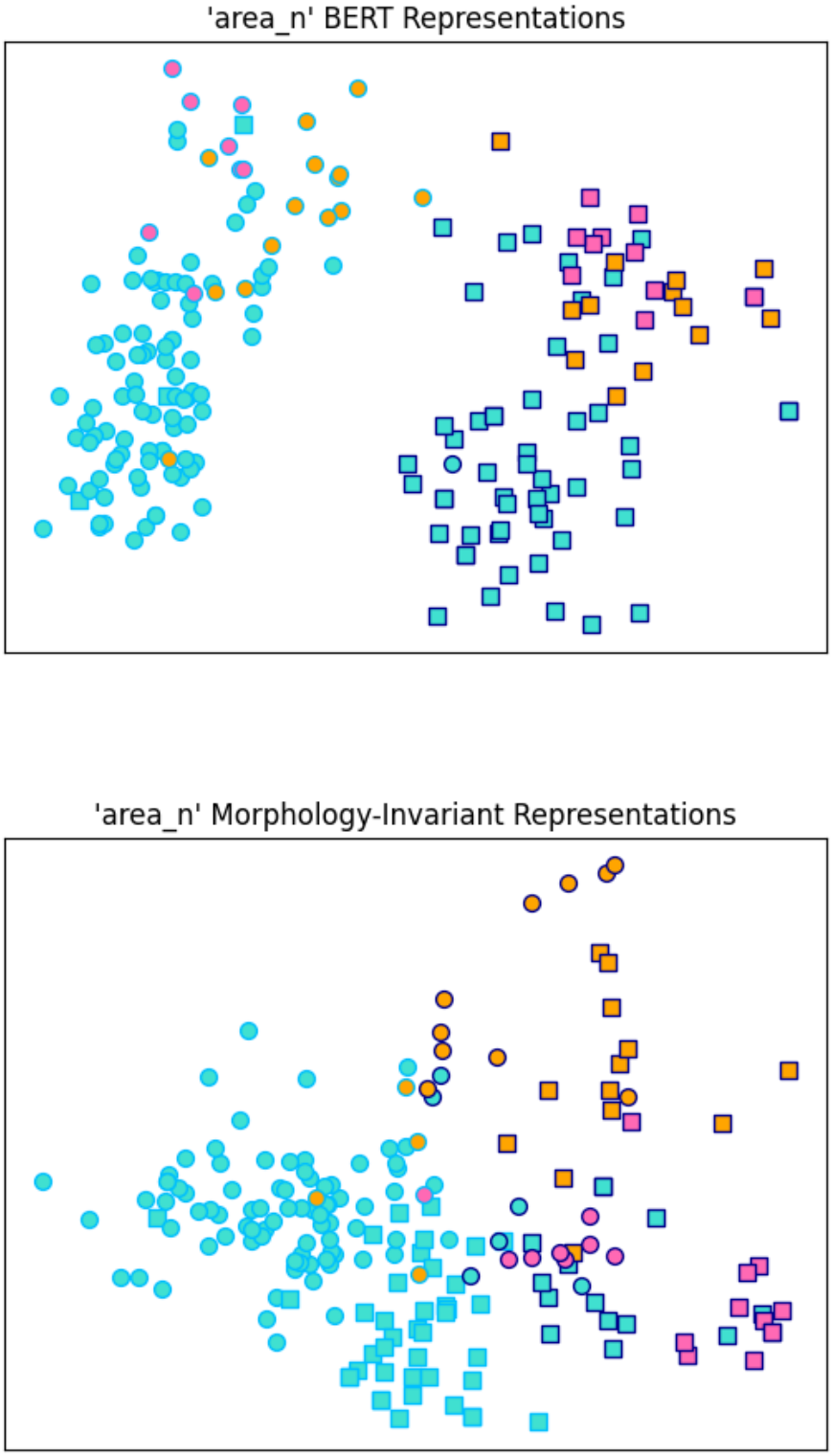
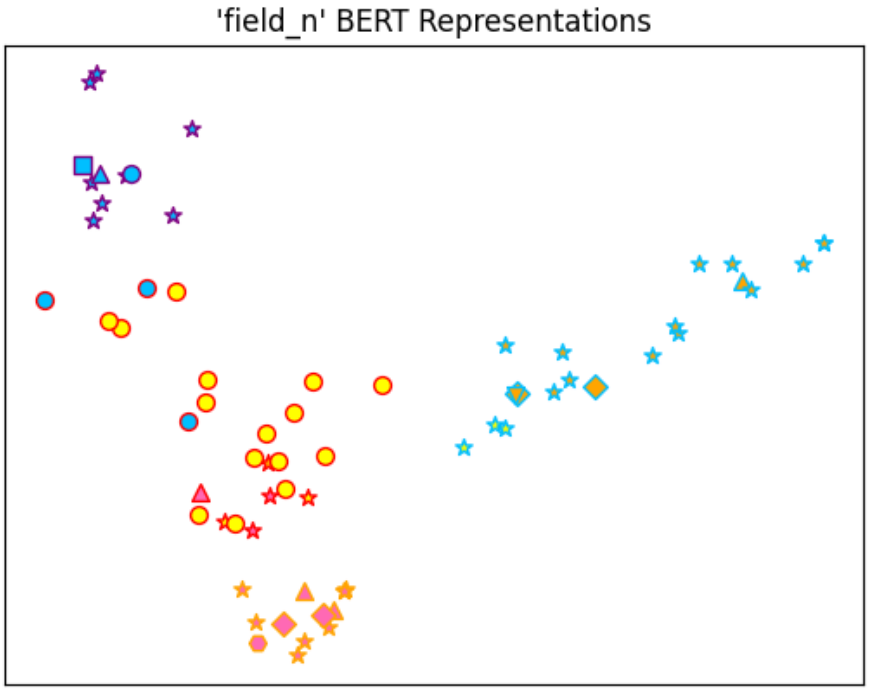


Figure 4: PCA visualizations of BERT representations (top) and morphology-invariant representations (bottom) of different sense instances of 'area'. Different data point colors refer to different senses, and different border colors refer to different clusters. Additionally, different marker styles refer to instances with different morphological features, i.e. grammatical number; *circles* for *singular nouns*, *squares* for *plural nouns*.



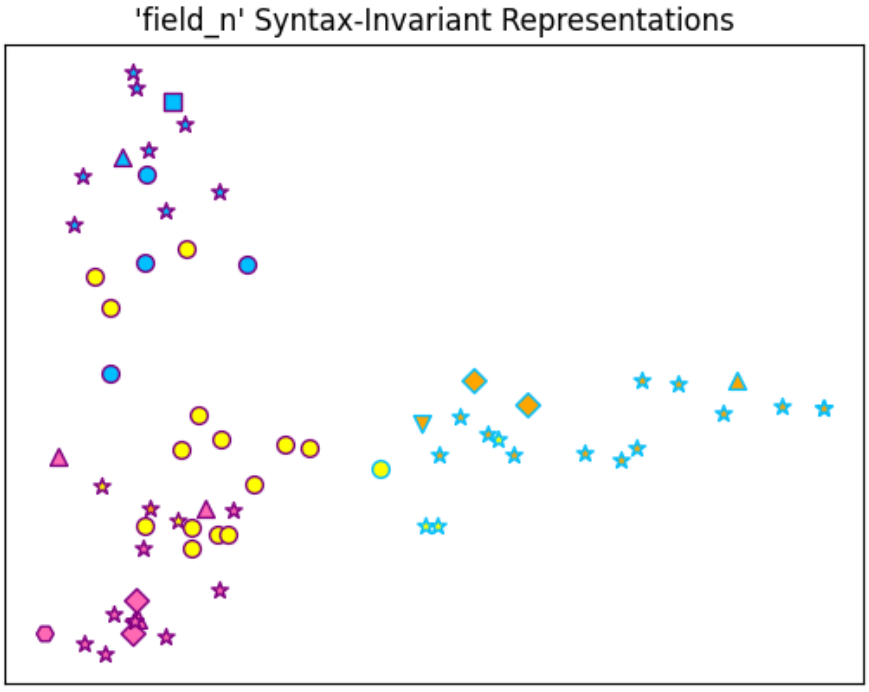


Figure 5: PCA visualizations of BERT representations (top) and syntax-invariant representations (bottom) of different sense instances of 'field'. Different data point colors refer to different senses, and different border colors refer to different clusters. Additionally, different marker styles refer to instances with different syntactic roles; *circles* for *compound heads*, *stars* for *prepositional objects*, *triangles* for *direct objects*, *diamonds* for *subjects*, *reversed triangles* for *passive subjects*, and *hexagons* for *attributes*.