Probing for Semantic Classes: Diagnosing the Meaning Content of Word Embeddings

Yadollah Yaghoobzadeh[1] **Katharina Kann**[2] **Timothy J. Hazen**[1] **Eneko Agirre**[3] **Hinrich Schütze**[4]

[1]Microsoft Research Montréal
[2]Center for Data Science, New York University
[3]IXA NLP Group, University of the Basque Country
[4]CIS, LMU Munich
yayaghoo@microsoft.com

Abstract

Word embeddings typically represent different meanings of a word in a single conflated vector. Empirical analysis of embeddings of ambiguous words is currently limited by the small size of manually annotated resources and by the fact that word senses are treated as unrelated individual concepts. We present a large dataset based on manual Wikipedia annotations and word senses, where word senses from different words are related by semantic classes. This is the basis for novel diagnostic tests for an embedding's content: we *probe word embeddings for semantic classes* and analyze the embedding space by classifying embeddings into semantic classes. Our main findings are: (i) Information about a sense is generally represented well in a single-vector embedding – if the sense is frequent. (ii) A classifier can accurately predict whether a word is single-sense or multi-sense, based only on its embedding. (iii) Although rare senses are not well represented in single-vector embeddings, this does not have negative impact on an NLP application whose performance depends on frequent senses.

1 Introduction

Word embeddings learned by methods like Word2vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014) have had a big impact on natural language processing (NLP) and information retrieval (IR). They are effective and efficient for many tasks. More recently, contextualized embeddings like ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) have further improved performance. To understand both word and contextualized embeddings, which still rely on word/subword embeddings at their lowest layer, we must peek inside the blackbox embeddings.

Given the importance of word embeddings, attempts have been made to construct diagnostic tools to analyze them. However, the main tool for analyzing their semantic content is still looking at nearest neighbors of embeddings. Nearest neighbors are based on full-space similarity neglecting the multifacetedness property of words (Gladkova and Drozd, 2016) and making them unstable (Wendlandt et al., 2018).

As an alternative, *we propose diagnostic classification of embeddings into semantic classes as a probing task to reveal their meaning content.* We will refer to semantic classes as *S-classes*. We use S-classes such as `food`, `drug` and `living-thing` to define word senses. S-classes are frequently used for semantic analysis, e.g., by Kohomban and Lee (2005), Ciaramita and Altun (2006) and Izquierdo et al. (2009) for word sense disambiguation, but have not been used for analyzing embeddings.

Analysis based on S-classes is only promising if we have high-quality S-class annotations. Existing datasets are either too small to train embeddings, e.g., SemCor (Miller et al., 1993), or artificially generated (Yaghoobzadeh and Schütze, 2016). Therefore, *we build WIKI-PSE, a WIKIpedia-based resource for Probing Semantics in word Embeddings*. We focus on common and proper nouns, and use their S-classes as proxies for senses. For example, "lamb" has the senses `food` and `living-thing`.

Embeddings do not explicitly address ambiguity; multiple senses of a word are crammed into a single vector. This is not a problem in some applications (Li and Jurafsky, 2015); one possible explanation is that this is an effect of sparse coding that supports the recovery of individual meanings from a single vector (Arora et al., 2018). But ambiguity has an adverse effect in other scenarios, e.g., Xiao and Guo (2014) see the need of filtering out embeddings of ambiguous words in dependency parsing.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5740–5753
Florence, Italy, July 28 - August 2, 2019.

We present the first comprehensive empirical analysis of ambiguity in word embeddings. Our resource, WIKI-PSE, enables novel diagnostic tests that help explain how (and how well) embeddings represent multiple meanings.[1]

Our diagnostic tests show: (i) Single-vector embeddings can represent many non-rare senses well. (ii) A classifier can accurately predict whether a word is single-sense or multi-sense, based only on its embedding. (iii) In experiments with five common datasets for mention, sentence and sentence-pair classification tasks, the lack of representation of rare senses in single-vector embeddings has little negative impact – this indicates that for many common NLP benchmarks only frequent senses are needed.

2 Related Work

S-classes (semantic classes) are a central concept in semantics and in the analysis of semantic phenomena (Yarowsky, 1992; Ciaramita and Johnson, 2003; Senel et al., 2018). They have been used for analyzing ambiguity by Kohomban and Lee (2005), Ciaramita and Altun (2006), and Izquierdo et al. (2009), *inter alia*. There are some datasets designed for interpreting word embedding dimensions using S-classes, e.g., SEMCAT (Senel et al., 2018) and HyperLex (Vulic et al., 2017). The main differentiator of our work is our probing approach using supervised classification of word embeddings. Also, we do not use WordNet senses but Wikipedia entity annotations since WordNet-tagged corpora are small.

In this paper, we probe word embeddings with supervised classification. Probing the layers of neural networks has become very popular. Conneau et al. (2018) probe sentence embeddings on how well they predict linguistically motivated classes. Hupkes et al. (2018) apply diagnostic classifiers to test hypotheses about the hidden states of RNNs. Focusing on embeddings, Kann et al. (2019) investigate how well sentence and word representations encode information necessary for inferring the idiosyncratic frame-selectional properties of verbs. Similar to our work, they employ supervised classification. Tenney et al. (2019) probe syntactic and semantic information learned by contextual embeddings (Melamud et al., 2016; McCann et al., 2017; Peters et al., 2018; Devlin et al., 2018) compared to non-contextualized embeddings. They do not, however, address ambiguity, a key phenomenon of language. While the terms "probing" and "diagnosing" come from this literature, similar probing experiments were used in earlier work, e.g., Yaghoobzadeh and Schütze (2016) probe for linguistic properties in word embeddings using synthetic data and also the task of corpus-level fine-grained entity typing (Yaghoobzadeh and Schütze, 2015).

We use our new resource WIKI-PSE for analyzing ambiguity in the word embedding space. Word sense disambiguation (WSD) (Agirre and Edmonds, 2007; Navigli, 2009) and entity linking (EL) (Bagga and Baldwin, 1998; Mihalcea and Csomai, 2007) are related to ambiguity in that they predict the context-dependent sense of an ambiguous word or entity. In our complementary approach, we analyze directly how multiple senses are represented in embeddings. While WSD and EL are important, they conflate (a) the evaluation of the information content of an embedding with (b) a model's ability to extract that information based on contextual clues. We mostly focus on (a) here. Also, in contrast to WSD datasets, WIKI-PSE is not based on inferred sense tags and not based on artificial ambiguity, i.e., pseudowords (Gale et al., 1992; Schütze, 1992), but on real senses marked by Wikipedia hyperlinks. There has been work in generating dictionary definitions from word embeddings (Noraset et al., 2017; Bosc and Vincent, 2018; Gadetsky et al., 2018). Gadetsky et al. (2018) explicitly adress ambiguity and generate definitions for words conditioned on their embeddings and selected contexts. This also conflates (a) and (b).

Some prior work also looks at how ambiguity affects word embeddings. Arora et al. (2018) posit that a word embedding is a linear combination of its sense embeddings and that senses can be extracted via sparse coding. Mu et al. (2017) argue that sense and word vectors are linearly related and show that word embeddings are intersections of sense subspaces. Working with synthetic data, Yaghoobzadeh and Schütze (2016) evaluate embedding models on how robustly they represent two senses for low vs. high skewedness of senses. Our analysis framework is novel and complementary, with several new findings.

Some believe that ambiguity should be elimi-

[1] WIKI-PSE is available publicly at https://github.com/yyaghoobzadeh/WIKI-PSE.

m1: due to its tartness, it is often combined it with sweeter juices, such as @apple@ or grape.

m2: @apple@ is rumored to be working on a smartwatch which may be called an "iwatch".

m3: a clic app was released for @apple@ 's iOS devices in August.

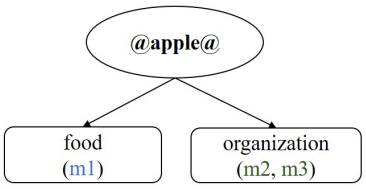


Figure 1: Example of how we build WIKI-PSE. There are three sentences linking "apple" to different entities. There are two mentions (m_2,m_3) with the `organization` sense (S-class) and one mention (m_1) with the `food` sense (S-class).

nated from embeddings, i.e., that a separate embedding is needed for each sense (Schütze, 1998; Huang et al., 2012; Neelakantan et al., 2014; Li and Jurafsky, 2015; Camacho-Collados and Pilehvar, 2018). This can improve performance on contextual word similarity, but a recent study (Dubossarsky et al., 2018) questions this finding. WIKI-PSE allows us to compute sense embeddings; we will analyze their effect on word embeddings in our diagnostic classifications.

3 WIKI-PSE Resource

We want to create a resource that allows us to probe embeddings for S-classes. Specifically, we have the following desiderata:
(i) We need a corpus that is S-class-annotated at the token level, so that we can train sense embeddings as well as conventional word embeddings.
(ii) We need a dictionary of the corpus vocabulary that is S-class-annotated at the type level. This gives us a gold standard for probing embeddings for S-classes.
(iii) The resource must be large so that we have a training set of sufficient size that lets us compare different embedding learners and train complex models for probing.

We now describe WIKI-PSE, a Wikipedia-driven resource for Probing Semantics in Embeddings, that satisfies our desiderata.

WIKI-PSE consists of a corpus and a corpus-based dataset of word/S-class pairs: an S-class is assigned to a word if the word occurs with that S-class in the corpus. There exist sense annotated corpora like SemCor (Miller et al., 1993), but due to the cost of annotation, those corpora are usually limited in size, which can hurt the quality of the trained word embeddings – an important factor for our analysis.

location, person, organization, art, event, broadcast_program, title, product, living_thing, people-ethnicity, language, broadcast_network, time, religion-religion, award, internet-website, god, education-educational_degree, food, computer-programming_language, metropolitan_transit-transit_line, transit, finance-currency, disease, chemistry, body_part, finance-stock_exchange, law, medicine-medical_treatment, medicine-drug, broadcast-tv_channel, medicine-symptom, biology, visual_art-color

Table 1: S-classes in WIKI-PSE sorted by frequency.

In this work, we propose a novel and scalable approach to building a corpus without depending on manual annotation except in the form of Wikipedia anchor links.

WIKI-PSE is based on the English Wikipedia (2014-07-07). Wikipedia is suitable for our purposes since it contains nouns – proper and common nouns – disambiguated and linked to Wikipedia pages via anchor links. To find more abstract meanings than Wikipedia pages, we annotate the nouns with S-classes. We make use of the 113 FIGER types[2] (Ling and Weld, 2012), e.g., `person` and `person/author`.

Since we use distant supervision from knowledge base entities to their mentions in Wikipedia, the annotation contains noise. For example, "Karl Marx" is annotated with `person/author`, `person/politician` and `person` and so is every mention of him based on distant supervision which is unlikely to be true. To reduce noise, we sacrifice some granularity in the S-classes. We only use the 34 *parent* S-classes in the FIGER hierarchy that have instances in WIKI-PSE; see Table 1. For example, we leave out `person/author` and `person/politician` and just use `person`. By doing so, mentions of nouns are rarely ambiguous with respect to S-class and we still have a reasonable number of S-classes (i.e., 34).

The next step is to aggregate all S-classes a surface form is annotated with. Many surface forms

[2]We follow the mappings in https://github.com/xiaoling/figer to first find the corresponding Freebase topic of a Wikipedia page and then map it to FIGER types.

are used for referring to more than one Wikipedia page and, therefore, possibly to more than one S-class. So, by using these surface forms of nouns[3], and their aggregated derived S-classes, we build our dataset of *words* and *S-classes*. See Figure 1 for "apple" as an example.

We differentiate linked mentions by enclosing them with "@", e.g., "apple" → "@apple@". If the mention of a noun is not linked to a Wikipedia page, then it is not changed, e.g., its surface form remains "apple". This prevents conflation of S-class-annotated mentions with unlinked mentions.

For the corpus, we include only sentences with at least one annotated mention resulting in 550 million tokens – an appropriate size for embedding learning. By lowercasing the corpus and setting the minimum frequency to 20, the vocabulary size is ≈500,000. There are ≈276,000 annotated words in the vocabulary, each with >= 1 S-classes. In total, there are ≈343,000 word/S-class pairs, i.e., words have 1.24 S-classes on average.

For efficiency, we select a subset of words for WIKI-PSE. We first add all multiclass words (those with more than one S-class) to the dataset, divided randomly into train and test (same size). Then, we add a random set with the same size from single-class words, divided randomly into train and test (same size). The resulting train and test sets have the size of 44,250 each, with an equal number of single and multiclass words. The average number of S-classes per word is 1.75.

4 Probing for Semantic Classes in Word Embeddings

We investigate embeddings by probing: Is the information we care about available in a word w's embedding? Specifically, we probe for S-classes: Can the information whether w belongs to a specific S-class be obtained from its embedding? The probing method we use should be: (i) simple with only the word embedding as input, so that we do not conflate the quality of embeddings with other confounding factors like quality of context representation (as in WSD); (ii) supervised with enough training data so that we can learn strong and non-linear classifiers to extract meanings from embeddings; (iii) agnostic to the model architecture that the word embeddings are trained with.

WIKI-PSE, introduced in §3, provides a text corpus and annotations for setting up probing methods satisfying (i) – (iii). We now describe the other elements of our experimental setup: word and sense representations, probing tasks and classification models.

4.1 Representations of Words and Senses

We run word embedding models like WORD2VEC on WIKI-PSE to get embeddings for all words in the corpus, including special common and proper nouns like "@apple@".

We also learn an embedding for each S-class of a word, e.g., one embedding for "@apple@-food" and one for "@apple@-organization". To do this, each annotated mention of a noun (e.g., "@apple@") is replaced with a word/S-class token corresponding to its annotation (e.g., with "@apple@-food" or "@apple@-organization"). These word/S-class embeddings correspond to sense embeddings in other work.

Finally, we create an alternative word embedding for an ambiguous word like "@apple@" by aggregrating its word/S-class embeddings by summing them: $\vec{w} = \sum_i \alpha_i \vec{w_{c_i}}$ where $\vec{w}$ is the aggregated word embedding and the $\vec{w_{c_i}}$ are the word/S-class embeddings. We consider two aggregations:

- For **uniform** sum, written as **unif**Σ, we set α_i = 1. So a word is represented as the sum of its sense (or S-class) embeddings; e.g., the representation of "apple" is the sum of its organization and food S-class vectors.

- For **weighted** sum, written as **wght**Σ, we set $\alpha_i = \text{freq}(w_{c_i}) / \sum_j \text{freq}(w_{c_j})$, i.e., the relative frequency of word/S-class w_{c_i} in mentions of the word w. So a word is represented as the *weighted* sum of its sense (or S-class) embeddings; e.g., the representation of "apple" is the weighted sum of its organization and food S-class vectors where the organization vector receives a higher weight since it is more frequent in our corpus.

unifΣ is common in multi-prototype embeddings, cf. (Rothe and Schütze, 2017). wghtΣ is also motivated by prior work (Arora et al., 2018). Aggregation allows us to investigate the reason for poor performance of single-vector embeddings. Is it a problem that a single-vector representation is used as the multi-prototype literature claims? Or are single-vectors in principle sufficient, but the way sense embeddings are aggregated in a single-

[3]Linked multiwords are treated as single tokens.

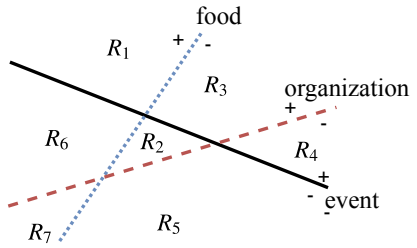


Figure 2: A 2D embedding space with three S-classes (food, organization and event). A line divides positive and negative regions of each S-class. Each of the seven R_i regions corresponds to a subset of S-classes.

vector representation (through an embedding algorithm, through unifΣ or through wghtΣ) is critical.

4.2 Probing Tasks

The first task is to probe for S-classes. We train, for each S-class, a binary classifier that takes an embedding as input and predicts membership in the S-class. An ambiguous word like "@apple@" belongs to multiple S-classes, so each of several different binary classifiers should diagnose it as being in its S-class. How well this type of probing for S-classes works in practice is one of our key questions: can S-classes be correctly encoded in embedding space?

Figure 2 shows a 2D embedding space: each point is assigned to a subset of the three S-classes, e.g., "@apple@" is in the region "+food ∩ +organization ∩ -event" and "@google@" in the region "-food ∩ +organization ∩ -event".

The second probing task predicts whether an embedding represents an unambiguous (i.e., one S-class) or an ambiguous (i.e., multiple S-classes) word. Here, we do not look for any specific meaning in an embedding, but assess whether it is an encoding of multiple different meanings or not. High accuracy of this classifier would imply that ambiguous and unambiguous words are distinguishable in the embedding space.

4.3 Classification Models

Ideally, we would like to have linearly separable spaces with respect to S-classes – presumably embeddings from which information can be effectively extracted by such a simple mechanism are better. However, this might not be the case considering the complexity of the space: non-linear models may detect S-classes more accurately. Nearest neighbors computed by cosine similarity are frequently used to classify and analyze embeddings, so we consider them as well. Accordingly, we experiment with three classifiers: (i) logistic regression (LR); (ii) multi-layer perceptron (MLP) with one hidden and a final ReLU layer; and (iii) KNN: K-nearest neighbors.

5 Experiments

Learning embeddings. Our method is agnostic to the word embedding model. Therefore, we experiment with two popular similar embedding models: (i) SkipGram (henceforth **SKIP**) (Mikolov et al., 2013), and (ii) Structured SkipGram (henceforth **SSKIP**) (Ling et al., 2015). SSKIP models word order while SKIP is a bag-of-words model. We use WANG2VEC (Ling et al., 2015) with negative sampling for training both models on WIKI-PSE. For each model, we try four embedding sizes: {100, 200, 300, 400} using identical hyperparameters: negatives=10, iterations=5, window=5.

emb	size	ln	LR	KNN	MLP
SKIP word	100	1	.723	.738	.773
	200	2	.740	.734	.786
	300	3	.745	.730	.787
	400	4	.747	.727	.786
SKIP wghtΣ	100	5	.681	.727	.752
	200	6	.695	.721	.756
	300	7	.699	.728	.752
	400	8	.702	.711	.753
SKIP unifΣ	100	9	.787	**.783**	.830
	200	10	.797	.773	.833
	300	11	.800	.765	.832
	400	12	**.801**	.758	**.834**
SSKIP word	100	13	.737	.749	.785
	200	14	.754	.745	.793
	300	15	.760	.741	.797
	400	16	.762	.737	.790
SSKIP wghtΣ	100	17	.699	.733	.762
	200	18	.710	.726	.764
	300	19	.714	.718	.767
	400	20	.717	.712	.763
SSKIP unifΣ	100	21	.801	**.783**	.834
	200	22	.809	.767	.840
	300	23	.812	.755	.842
	400	24	**.814**	.747	**.844**
random	–	–	.273	–	–

Table 2: F_1 for S-class prediction. emb: embedding, unifΣ (resp. wghtΣ): uniform (resp. weighted) sum of word/S-classes. ln: line number. Bold: best F_1 result per column and embedding model (SKIP and SSKIP).

5.1 S-class Prediction

Table 2 shows results on S-class prediction for word, unifΣ and wghtΣ embeddings trained using SKIP and SSKIP. Random is a simple baseline that randomly assigns to a test example each S-class

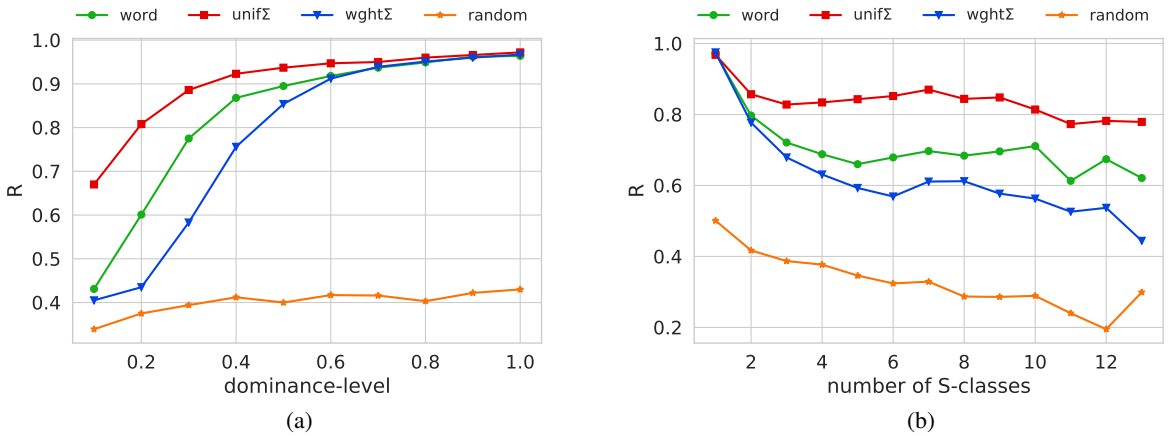


Figure 3: Results of S-class prediction as a function of two important factors: dominance-level and number of S-classes

according to its prior probability (i.e., proportion in train).

We train classifiers with Scikit-learn (Pedregosa et al., 2011). Each classifier is an independent binary predictor for one S-class. We use the global metric of micro F_1 over all test examples and over all S-class predictions. We see the following trends in our results.

MLP is consistently better than LR or KNN. Comparing MLP and LR reveals that the space is not linearly separable with respect to the S-classes. This means that linear classifiers are insufficient for semantic probing: *we should use models for probing that are more powerful than linear.*

Higher dimensional embeddings perform better for MLP and LR, but worse for KNN. We do further analysis by counting the number k of unique S-classes in the top 5 nearest neighbors for word embeddings; k is 1.42 times larger for embeddings of dimensionality 400 than 200. Thus, *more dimensions results in more diverse neighborhoods and more randomness*. We explain this by the increased degrees of freedom in a higher dimensional space: idiosyncratic properties of words can also be represented given higher capacity and so similarity in the space is more influenced by idiosyncracies, not by general properties like semantic classes. Similarity datasets tend to only test the majority sense of words (Gladkova and Drozd, 2016), and that is perhaps why similarity results usually do not follow the same trend (i.e., higher dimensions improve results). See Table 6 in Appendix for results on selected similarity datasets.

SSKIP performs better than SKIP. The difference between the two is that SSKIP models word order. Thus, we conclude that *modeling word order is important for a robust representation*. This is in line with the more recent FASTTEXT model with word order that outperforms prior work (Mikolov et al., 2017).

We now compare word embeddings, unifΣ, and wghtΣ. Recall that the sense vectors of a word have equal weight in unifΣ and are weighted according to their frequency in wghtΣ. The results for word embeddings (e.g., line 1) are between those of unifΣ (e.g., line 9) and wghtΣ (e.g., line 5). This indicates that their weighting of sense vectors is somewhere between the two extremes of unifΣ and wghtΣ. Of course, word embeddings are not computed as an explicit weighted sum of sense vectors, but there is evidence that they are implicit frequency-based weighted sums of meanings or concepts (Arora et al., 2018).

The ranking unifΣ $>$ word embeddings $>$ wghtΣ indicates how well individual sense vectors are represented in the aggregate word vectors and how well they can be "extracted" by a classifier in these three representations. Our prediction task is designed to find *all* meanings of a word, including rare senses. unifΣ is designed to give relatively high weight to rare senses, so it does well on the prediction task. wghtΣ and word embeddings give low weights to rare senses and very high weights to frequent senses, so the rare senses can be "swamped" and difficult to extract by classifiers from the embeddings.

Public embeddings. To give a sense on how well public embeddings, trained on much larger data, do on S-class prediction in WIKI-PSE, we use 300d GLOVE embeddings trained on 6B to-

emb	LR	KNN	MLP
word	.711	.605	.715
wghtΣ	.652	.640	.667
unifΣ	**.766**	**.709**	**.767**
GLOVE(6B)	.667	.638	.685
FASTTEXT(Wiki)	.699	.599	.697

Table 3: F_1 for S-class prediction on the subset of WIKI-PSE whose vocabulary is shared with GLOVE and FASTTEXT. Apart from using a subset of WIKI-PSE, this is the same setup as in Table 2, but here we compare word, wghtΣ, and unifΣ with public GLOVE and FASTTEXT.

kens[4] from Wikipedia and Gigaword and FASTTEXT Wikipedia word embeddings.[5] We create a subset of the WIKI-PSE dataset by keeping only single-token words that exist in the two embedding vocabularies. The size of the resulting dataset is 13,000 for train and test each; the average number of S-classes per word is 2.67.

Table 3 shows results and compares with our different SSKIP 300d embeddings. There is a clear performance gap between the two off-the-shelf embedding models and unifΣ, indicating that training on larger text does not necessarily help for prediction of rare meanings. This table also confirms Table 2 results with respect to comparison of learning model (MLP, LR, KNN) and embedding model (word, wghtΣ, unifΣ). Overall, the performance drops compared to the results in Table 2. Compared to the WIKI-PSE dataset, this subset has fewer (13,000 vs. 44,250) training examples, and a larger number of labels per example (2.67 vs. 1.75). Therefore, it is a harder task.

5.1.1 Analysis of Important Factors

We analyze the performance with respect to multiple factors that can influence the quality of the representation of S-class s in the embedding of word w: dominance, number of S-classes, frequency and typicality. We discuss the first two here and the latter two in the Appendix §A. These factors are similar to those affecting WSD systems (Pilehvar and Navigli, 2014). We perform this analysis for MLP classifier on SSKIP 400d embeddings. We compute the recall for various conditions.[6]

Dominance of the S-class s for word w is defined as the percentage of the occurrences of w where its labeled S-class is s. Figure 3a shows for each dominance level what percentage of S-classes of that level were correctly recognized by their binary classifier. For example, 0.9 or 90% of S-classes of words with dominance level 0.3 were correctly recognized by the corresponding S-class's binary classifier for unifΣ ((a), red curve). Not surprisingly, more dominant meanings are represented and recognized better.

We also see that word embeddings represent non-dominant meanings better than wghtΣ, but worse than unifΣ. For word embeddings, the performance drops sharply for dominance <0.3. For wghtΣ, the sharp drops happens earlier, at dominance <0.4. Even for unifΣ, there is a (less sharp) drop – this is due to other factors like frequency and not due to poor representation of less dominant S-classes (which all receive equal weight for unifΣ).

The **number of S-classes** of a word can influence the quality of meaning extraction from its embedding. Figure 3b confirms our expectation: It is easier to extract a meaning from a word embedding that encodes fewer meanings. For words with only one S-class, the result is best. For ambiguous words, performance drops but this is less of an issue for unifΣ. For word embeddings (word), performance remains in the range 0.6-0.7 for more than 3 S-classes which is lower than unifΣ but higher than wghtΣ by around 0.1.

5.2 Ambiguity Prediction

We now investigate if a classifier can predict whether a word is ambiguous or not, based on the word's embedding. We divide the WIKI-PSE dataset into two groups: unambiguous (i.e., one S-class) and ambiguous (i.e., multiple S-classes). LR, KNN and MLP are trained on the training set and applied to the words in test. The only input to a classifier is the embedding; the output is binary: one S-class or multiple S-classes. We use SSKIP word embeddings (dimensionality 400) and L2-normalize all vectors before classification. As a baseline, we use the word frequency as single feature (FREQUENCY) for LR classifier.

model	LR	KNN	MLP
FREQUENCY	64.8	-	-
word	77.9	72.1	81.2
wghtΣ	76.9	69.2	81.1
unifΣ	96.2	72.2	97.1

Table 4: Accuracy for predicting ambiguity

[4]https://nlp.stanford.edu/projects/glove/

[5]https://fasttext.cc/docs/en/pretrained-vectors.html

[6]Precision for these cases is not defined. This is similarly applied in WSD (Pilehvar and Navigli, 2014).

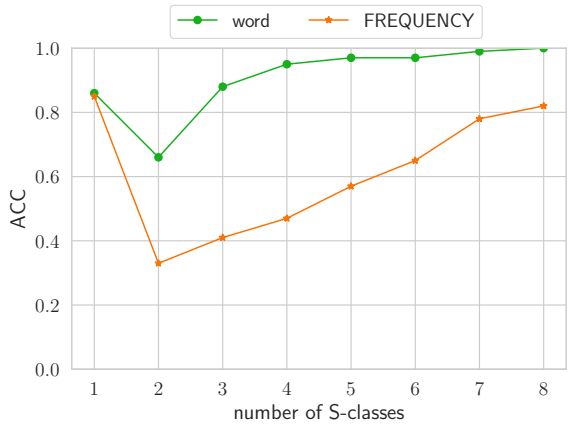


Figure 4: Accuracy of word embedding and FREQUENCY for predicting ambiguity as a function of number of S-classes, using MLP classifier.

Table 4 shows overall accuracy and Figure 4 accuracy as a function of number of S-classes. Accuracy of standard word embeddings is clearly above the baselines, e.g., 81.2% for MLP and 77.9% for LR compared to 64.8% for FREQUENCY. The figure shows that the decision becomes easier with increased ambiguity (e.g., ≈100% for 6 or more S-classes). It makes sense that a highly ambiguous word is more easily identifiable than a two-way ambiguous word. MLP accuracy for unifΣ is close to 100%. We can again attribute this to the fact that rare senses are better represented in unifΣ than in regular word embeddings, so the ambiguity classification is easier.

KNN results are worse than LR and MLP. This indicates that similarity is not a good indicator of degree of ambiguity: words with similar degrees of ambiguity do not seem to be neighbors of each other. This observation also points to an explanation for why the classifiers achieve such high accuracy. We saw before that S-classes can be identified with high accuracy. Imagine a multi-layer architecture that performs binary classification for each S-class in the first layer and, based on that, makes the ambiguity decision based on the number of S-classes found. LR and MLP seem to approximate this architecture. Note that this can only work if the individual S-classes are recognizable, which is not the case for rare senses in regular word embeddings.

In Appendix §C, we show top predictions for ambiguous and unambiguous words.

5.3 NLP Application Experiments

Our primary goal is to probe meanings in word embeddings without confounding factors like contextual usage. However, to give insights on how our probing results relate to NLP tasks, we evaluate our embeddings when used to represent word tokens.[7] Note that our objective here is not to improve over other baselines, but to perform analysis.

We select mention, sentence and sentence-pair classification datasets. For mention classification, we adapt Shimaoka et al. (2017)'s setup:[8] training, evaluation (FIGER dataset) and implementation. The task is to predict the contextual fine-grained types of entity mentions. We lowercase the dataset to match the vocabularies of GLOVE(6B), FASTTEXT(Wiki) and our embeddings. For sentence and sentence-pair classifications, we use the SentEval[9] (Conneau and Kiela, 2018) setup for four datasets: MR (Pang and Lee, 2005) (positive/negative sentiment prediction for movie reviews) , CR (Hu and Liu, 2004) (positive/negative sentiment prediction for product reviews), SUBJ (Pang and Lee, 2004) (subjectivity/objectivity prediction) and MRPC (Dolan et al., 2004) (paraphrase detection). We average embeddings to encode a sentence.

emb	MC	CR	MR	SUBJ	MRPC
word	64.6	70.4	71.4	89.2	71.3
wghtΣ	65.4	72.3	72.0	89.4	71.5
unifΣ	61.6	69.1	68.8	87.9	71.3
GLOVE(6B)	58.1	75.7	75.2	91.3	72.5
FASTTEXT(Wiki)	55.5	76.7	75.2	91.2	71.6

Table 5: Performance of the embedding models on five NLP tasks

Table 5 shows results. For MC, performance of embeddings is ordered: wghtΣ > word > unifΣ. This is the opposite of the ordering in Table 2 where unifΣ was the best and wghtΣ the worst. The models with more weight on frequent meanings perform better in this task, likely because the dominant S-class is mostly what is needed. In an error analysis, we found many cases where mentions have one major sense and some minor senses; e.g., unifΣ predicts "Friday" to be "location" in the context "the U.S. Attorney's Of-

[7] For the embeddings used in this experiment, if there are versions with and without "@"s, then we average the two; e.g., "apple" is the average of "apple" and "@apple@".

[8] https://github.com/shimaokasonse/NFGEC

[9] https://github.com/facebookresearch/SentEval

fice announced Friday". Apart from the major S-class "time", "Friday" is also a mountain ("Friday Mountain"). unifΣ puts the same weight on "location" and "time". wghtΣ puts almost no weight on "location" and correctly predicts "time". Results for the four other datasets are consistent: the ordering is the same as for MC.

6 Discussion and Conclusion

We quantified how well multiple meanings are represented in word embeddings. We did so by designing two probing tasks, S-class prediction and ambiguity prediction. We applied these probing tasks on WIKI-PSE, a large new resource for analysis of ambiguity and word embeddings. We used S-classes of Wikipedia anchors to build our dataset of word/S-class pairs. We view S-classes as corresponding to senses.

A summary of our findings is as follows. (i) We can build a classifier that, with high accuracy, correctly predicts whether an embedding represents an ambiguous or an unambiguous word. (ii) We show that semantic classes are recognizable in embedding space – a novel result as far as we know for a real-world dataset – and much better with a nonlinear classifier than a linear one. (iii) The standard word embedding models learn embeddings that capture multiple meanings in a single vector well – if the meanings are frequent enough. (iv) Difficult cases of ambiguity – rare word senses or words with numerous senses – are better captured when the dimensionality of the embedding space is increased. But this comes at a cost – specifically, cosine similarity of embeddings (as, e.g., used by KNN, §5.2) becomes less predictive of S-class. (v) Our diagnostic tests show that a uniform-weighted sum of the senses of a word w (i.e., unifΣ) is a high-quality representation of all senses of w – even if the word embedding of w is not. This suggests again that the main problem is not ambiguity per se, but rare senses. (vi) Rare senses are badly represented if we use explicit frequency-based weighting of meanings (i.e., wghtΣ) compared to word embedding learning models like SkipGram.

To relate these findings to sentence-based applications, we experimented with a number of public classification datasets. Results suggest that embeddings with frequency-based weighting of meanings work better for these tasks. Weighting all meanings equally means that a highly dominant sense (like "time" for "Friday") is severely downweighted. This indicates that currently used tasks rarely need rare senses – they do fine if they have only access to frequent senses. However, to achieve high-performance natural language understanding at the human level, our models also need to be able to have access to rare senses – just like humans do. We conclude that we need harder NLP tasks for which performance depends on rare as well as frequent senses. Only then will we be able to show the benefit of word representations that represent rare senses accurately.

Acknowledgments

We are grateful for the support of the European Research Council (ERC #740516) and UPV/EHU (excellence research group) for this work. Next, we thank all the anonymous reviewers their detailed assessment and helpful comments. We also appreciate the insightful discussion with Geoffrey J. Gordon, Tong Wang, and other members of Microsoft Research Montréal.

References

Eneko Agirre and Philip Edmonds. 2007. *Word Sense Disambiguation: Algorithms and Applications*, 1st edition. Springer Publishing Company, Incorporated.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *TACL*, 6:483–495.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566.

Tom Bosc and Pascal Vincent. 2018. Auto-encoding dictionary definitions into consistent word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1522–1532.

José Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artifical Intelligence*, 63:743–788.

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *EMNLP*, pages 594–602.

Massimiliano Ciaramita and Mark Johnson. 2003. Supersense tagging of unknown nouns in wordnet. In *EMNLP*, pages 168–175.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *ACL 2018*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*.

Haim Dubossarsky, Eitan Grossman, and Daphna Weinshall. 2018. Coming to your senses: on controls and evaluation sets in polysemy research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1732–1740.

Lucie Flekova and Iryna Gurevych. 2016. Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2029–2041.

Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. Conditional generators of words definitions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 266–271.

William A Gale, Kenneth W Church, and David Yarowsky. 1992. Work on statistical methods for word sense disambiguation. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, volume 54, page 60.

Anna Gladkova and Aleksandr Drozd. 2016. Intrinsic evaluations of word embeddings: What can we do better? In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP, RepEval@ACL 2016*, pages 36–42.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Rubén Izquierdo, Armando Suárez, and German Rigau. 2009. An empirical study on class-based word sense disambiguation. In *EACL*, pages 389–397.

Stanislaw Jastrzebski, Damian Lesniak, and Wojciech Marian Czarnecki. 2017. How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks. *CoRR*, abs/1702.02170.

Katharina Kann, Alex Warstadt, Adina Williams, and Samuel R Bowman. 2019. Verb argument structure alternations in word and sentence embeddings. In *Proceedings of the Society for Computation in Linguistics*.

Upali S. Kohomban and Wee Sun Lee. 2005. Learning semantic classes for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 34–41.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *EMNLP*, pages 1722–1732.

Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the 16th AAAI Conference on Artificial Intelligence*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, pages 233–242.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Proceedings of the Workshop on Human Language Technology*, pages 303–308.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2017. Geometry of polysemy. In *Proceedings of the 5th International Conference on Learning Representations*.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *EMNLP*, pages 1059–1069.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 271–278.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 115–124.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Mohammad Taher Pilehvar and Roberto Navigli. 2014. A large-scale pseudoword-based evaluation framework for state-of-the-art word sense disambiguation. *Computational Linguistics*, 40(4):837–881.

Sascha Rothe and Hinrich Schütze. 2017. Autoextend: Combining word embeddings with semantic resources. *Computational Linguistics*, 43(3):593–617.

Hinrich Schütze. 1992. Dimensions of meaning. In *Proceedings Supercomputing '92, Minneapolis, MN, USA, November 16-20, 1992*, pages 787–796.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Lutfi Kerem Senel, Ihsan Utlu, Veysel Yücesoy, Aykut Koc, and Tolga Çukur. 2018. Semantic structure and interpretability of word embeddings. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 26(10):1769–1779.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1271–1280.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *ICLR*.

Ivan Vulic, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4).

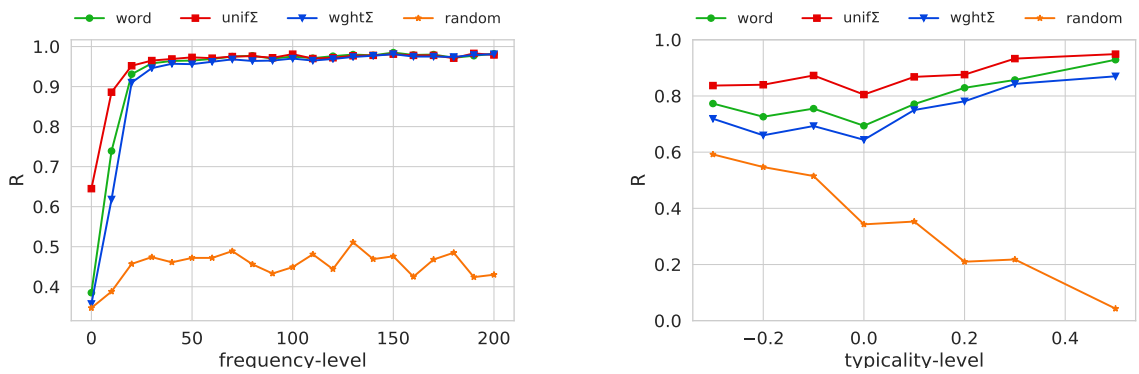
Laura Wendlandt, Jonathan K. Kummerfeld, and Rada Mihalcea. 2018. Factors influencing the surprising instability of word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2092–2102.

Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 119–129.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2015. Corpus-level fine-grained entity typing using contextual information. In *EMNLP*, pages 715–725.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2016. Intrinsic subspace evaluation of word embedding representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 236–246.

David Yarowsky. 1992. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *14th International Conference on Computational Linguistics*, pages 454–460.



(a) (b)

Figure 5: Results of word, uniform and weighted word/S-class embeddings for two other important factors: frequency and typicality of S-class.

A Analysis of important factor: more analysis

Frequency is defined as the absolute frequency of s in occurrences of w. Frequency is important to get good representations and the assumption is that more frequency means better results. In Figure 5a, prediction performance is shown for a varying frequency-level. Due to rounding, each level in x includes frequencies $[x-5, x+5]$. As expected higher frequency means better results. All embeddings have high performance when frequency is more than 20, emphasizing that embeddings can indeed represent a meaning well if it is not too rare. For low frequency word/S-class es, the uniform sum performs clearly better than the other models. This shows that word and weighted word/S-class embeddings are not good encodings for rare meanings.

Typicality of a meaning for a word is important. We define the typicality of S-class s for word w as its average compatibility level with other classes of w. We use Pearson correlation between S-classes in the training words and assign the compatibility level of S-classes based on that. In Figure 5b, we see that more positive typicality leads to better results in general. Each level in x axis represents $[x-0.05, x+0.05]$. The S-classes that have negative typicality are often the frequent ones like "person" and "location" and that is why the performance is relatively good for them.

B What does happen when classes of a word become balanced?

Here, we analyze the space of word embeddings with multiple semantic classes as the class distribution gets more balanced. In Figure 6, we show that for two-class words, the average number of unique classes in the top five nearest neighbors increases as the dominance level increases. The dominance-level of 0.4 is basically where the two classes are almost equally frequent. As the two classes move towards equal importance, their word embeddings move towards a space with more diversity.

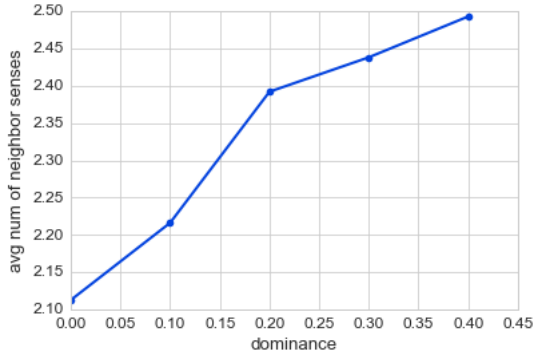


Figure 6: The average number of unique semantic classes in the nearest neighbors of words with two classes, in different dominance level.

C Ambiguity prediction examples

In Table 7, we show some example predicted ambiguous and unambiguous words based on the word embeddings.

D Supersense experiment

To confirm our results in another dataset, we try supersense annotated Wikipedia of UKP (Flekova and Gurevych, 2016). We use their published 200-dimensional word embeddings. A similar process

model	size	MEN	MTurk	RW	SimLex999	WS353	Google	MSR
SKIP	100	0.633	0.589	0.283	0.276	0.585	0.386	0.317
SKIP	200	0.675	0.613	0.286	0.306	0.595	0.473	0.382
SKIP	300	0.695	0.624	0.279	0.325	0.626	0.495	0.405
SKIP	400	0.708	0.630	0.268	0.334	0.633	0.506	0.416
SSKIP	100	0.598	0.555	0.313	0.272	0.559	0.375	0.349
SSKIP	200	0.629	0.574	0.310	0.306	0.592	0.464	0.413
SSKIP	300	0.645	0.588	0.300	0.324	0.606	0.486	0.430
SSKIP	400	0.655	0.576	0.291	0.340	0.616	0.491	0.431

Table 6: Similarity and analogy results of our word embeddings on a set of datasets (Jastrzebski et al., 2017). The table shows the Spearmans correlation between the models similarities and human judgments. Size is the dimensionality of the embeddings. Except for RW dataset, results improve by increasing embeddings size.

word	frequency	senses	likelihood
@liberty@	554	event, organization, location, product, art, person	1.0
@aurora@	879	organization, location, product, god, art, person, broadcast_program	1.0
@arcadia@	331	event, organization, location, product, art, person, living_thing	1.0
@brown@	590	food, event, title, organization, visual_art-color, person, art, location, people-ethnicity, living_thing	1.0
@marshall@	1070	art, location, title, organization, person	1.0
@green@	783	food, art, organization, visual_art-color, location, internet-website, metropolitan_transit-transit_line, religion-religion, person, living_thing	1.0
@howard@	351	person, title, organization, location	1.0
@lucas@	216	art, person, organization, location	1.0
@smith@	355	title, organization, person, product, art, location, broadcast_program	1.0
@taylor@	367	art, location, product, organization, person	1.0
...			
...			
@tom_cibulec@	47	person	0.0
@judd_winick@	113	person	0.0
@roger_reijners@	26	person	0.0
@patrick_rafter@	175	person	0.0
@nasser_hussain@	82	person	0.0
@sam_wyche@	76	person, event	0.0
@lovie_smith@	116	person	0.0
@calliostomatidae@	431	living_thing	0.0
@joe_girardi@	147	person	0.0
@old_world@	91	location, living_thing	0.0

Table 7: The top ten ambiguous words followed by the top unambiguous words based on our model prediction in Section 5.3. Each line is a word followed by its frequency in the corpus, its dataset senses and finally our ambiguity prediction likelihood to be ambiguous.

model	norm?	LR	KNN	MLP
MAJORITY	-	50.0	-	-
FREQUENCY	-	67.3	-	-
word embedding	yes	70.1	65.4	**72.4**
word embedding	no	72.3	65.4	**73.0**

Table 8: Ambiguity prediction accuracy for the supersense dataset. Norm: L2-normalizing the vectors.

as our WIKI-PSE is applied on the annotated corpus to build word/S-class dataset. Here, the S-classes are the supersenses. We consider NOUN categories of words and build datasets for our analysis by aggregating the supersenses a word annotated with in the corpus. Number of supersenses is 26 and train and test size: 27874. In Table 8, we show the results of ambiguity prediction. As we see, we can predict ambiguity using word embeddings with accuracy of 73%.