

Conceptual Shadows: Visualizing Concept-specific Dimensions of Meaning in Word Embeddings with Self Organizing Maps

Laura Spillner¹, Robert Porzel¹, Robin Nolte¹ and Rainer Malaka¹

¹University of Bremen, Digital Media Lab, Bibliothekstr. 5, 28359 Bremen, Germany

Abstract

Word embeddings (high-dimensional vectors) are common input representations in NLP. However, this kind of representation is not meaningful to humans; it presents a black box that makes it difficult to explain how the vectors influence downstream models. Visualizing word vectors usually requires dimensionality reduction. We explore the visualization of word vectors as 2D images (one image per word, one pixel per vector dimension) by organizing the dimensions in the image with a self-organizing map. This method reveals new insights into how and where semantic information is encoded in the vector and allows us to pinpoint the source of downstream classification errors in the input representation. In this paper, we present the first results of an investigation into word embeddings that visualizes individual word vectors as images and explores what information the individual dimensions of the vectors encode. As this encoded information is specific to the given target concepts of a symbolic downstream classification task, it can be regarded as a projection from the symbolic space to that of the deep neural network.

Keywords

Word-Embeddings, Ontologies, Language Processing,

1. Introduction

Undoubtedly, both symbolic and sub-symbolic approaches to artificial intelligence (AI) have their respective merits and individual shortcomings. In many applications, they are already joined at the hip, as the output of deep learning models often consists of classes that are symbolically described and used further on in some overall processing pipeline. One of the main areas of interest in the field of explainable artificial intelligence (XAI), and arguably one of the driving factors of recent interest in the field, is the explanation of *black-box* deep learning models. In natural language processing (NLP), deep neural networks (DNNs) are used in two ways: Firstly, to produce numeric input representations from natural language texts, and secondly, to solve

CAOS VII: *Cognition and Ontologies, 9th Joint Ontology Workshops (JOWO 2023), co-located with FOIS 2023, 19-20 July, 2023, Sherbrooke, Québec, Canada*

✉ laura.spillner@uni-bremen.de (L. Spillner); porzel@uni-bremen.de (R. Porzel); nolte@uni-bremen.de (R. Nolte); malaka@tzi.de (R. Malaka)

🌐 <https://www.uni-bremen.de/dmlab/team/laura-spillner> (L. Spillner);

<https://www.uni-bremen.de/dmlab/team/dr-ing-robert-porzel> (R. Porzel);

<https://www.uni-bremen.de/dmlab/team/robin-nolte> (R. Nolte);


<https://www.uni-bremen.de/dmlab/team/rainer-malaka> (R. Malaka)

🆔 0000-0001-8490-8961 (L. Spillner); 0000-0002-7686-2921 (R. Porzel); 0009-0004-2975-6378 (R. Nolte);

0000-0002-7686-2921 (R. Malaka)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

downstream tasks, e.g., classification, clustering, or language generation. In many of these downstream tasks, conceptual models, such as ontologies, of the task-specific domain constitute the target representations for the classification.

For example, when classifying the part of speech (POS) of the words used in a sentence, specific classes are used as the values of the POS attribute, e.g., NOUN, VERB, or ADJ. These values are often part of a conceptual model, e.g., an ontology of linguistic entities, such as the GOLD ontology [1], the OntoWordNet model [2], or the LingInfo model [3]. In many cases, therefore, sub-symbolic approaches are used to classify entities stemming from some ontological model. In these sub-symbolic approaches, representations in which one word constitutes one symbol, such as bag-of-words or n-gram models, have largely been replaced by distributed semantic representations – also called *word embeddings* or *word vectors* – to represent text. It is generally accepted that the embeddings encode semantic information about a word and that words close to each other in the vector space are similar in meaning [4, 5]. However, high-dimensional word vectors pose difficulty from the XAI perspective because they essentially add a second black box, the model learning the embeddings, around the model used for the task itself.

When it comes to fields such as computer vision, many techniques have been developed to explain DNNs, e.g., by generating example images of the classes they are trained to identify or by highlighting image areas of particular importance in the classification [6]. Even though the input is represented numerically, the representation (the digital image) is still meaningful to humans. In contrast, a word vector as a point in very high-dimensional vector space is rather difficult to imagine or to represent visually. Because of this, visual explanations in the field of NLP usually fall into one of two categories: One option is to use dimensionality reduction to represent word vectors as points in 2D space, thus making it possible to see which words are close together. The other option is to consider not individual words but rather texts and highlight words as salient features, e.g., when predicting the topic of a text [7].

In this paper, we present the first results of an investigation into word embeddings that takes a different approach: We visualize individual word vectors as images and, inspired by XAI methods from computer vision, explore what information the individual dimensions of the vectors encode. This encoded information is specific to the given target concepts of the downstream classification task at hand. It can be regarded as a projection from the symbolic conceptual space to that of the DNN. For each conceptual entity, e.g., NOUN or VERB, CAT or DOG, etc., we obtain its visual projection into the sub-symbolic space. We call this the *conceptual shadow* of that entity. One application for this approach is to improve understanding of the input representations we use for NLP tasks. We hope to utilize this method to understand the origin of mistakes in the downstream model, such as incorrect classifications where a given ontological model constitutes the target representation. In the long run, this work seeks to connect sub-symbolic and symbolic representations of the same conceptual entity.

2. Related Work

In this section, we provide short overviews of prior art with respect to ontological models of linguistic knowledge, word embeddings, and explainable natural language processing.

2.1. Modeling Linguistic Knowledge

Various approaches have been proposed to model linguistic knowledge, i.e., the entities and features that make up human language, in formal ontologies. These approaches differ in some respects, such as alignment to upper layers, their modeling intent, and their scope. One point of divergence lies in the alignment to a foundational layer. While, for example, the GOLD ontology [1] aligns with the SUMO upper ontology [8], the OntoWordNet model [2] aligns with the DOLCE foundational ontology [9]. The LingInfo model [3] can be used with any foundational framework as it relies on meta-classes to model information about the lexical entities. For also representing pragmatically relevant information, the SOMA-SAY [10] is based on Dolce Ultra Light and the Descriptions & Situations Module [11]. In contrast, the OntoWordNet aims at merging the linguistic information contained in WordNet with the respective classes employed in specific domain models, while both LingInfo and GOLD seek to incorporate more linguistic information, such as morphological and grammatical features of language. They all allow a direct connection of the respective linguistic information for terms with corresponding classes and properties in a domain ontology. Each model could be integrated into an NLP system as an additional module to allow reasoning about linguistic information or as a link between lexical and ontological resources.

2.2. Word Embeddings

Semantic embeddings have become standard input representations for many machine learning NLP tasks. Since the conception of word vectors in [4], improvements have been made with the introduction of character-based models and contextual representation [12, 13], which allow fine-tuning of pre-trained embeddings for downstream tasks [12, 14], as well as with the addition of transformer-based models [15] and attention mechanisms [16]. For this work, it is mainly important to differentiate between static representations, used in older models such as GloVe embeddings [5], and dynamic embeddings, which are part of Language Models like BERT [12]. With static embeddings, the same word is invariably represented by the same vector - it does not differ between different uses of the same word, e.g., homonyms or the exact spelling used as different POS. These static word vectors are then used as the input representation for downstream tasks. In contrast, when using dynamic embeddings, each use of the word in a text is represented by a different vector. Language models still represent each token in a text as a unique vector, but these are not generally intended to be accessible from outside the model.

2.3. Explainable NLP

The XAI literature differentiates between three types of explanations [7, 6]:

1. Explanations of network *processing*, including, e.g., Linear Proxy Models such as LIME [17]; salience mapping through occlusion [18]; etc.
2. Explanations of *representations* by probing the role of individual layers or individual neurons, for example, to generate images that maximize the activation of a given neuron and can be seen as prototypical examples of a given class [19, 20].
3. Systems that produce explanations.

Many works use explainable NLP in the third category to explain other models [21]. However, the focus of this work is different: Instead, we aim to explore on a deeper level where conceptual information is encoded in distributed semantic representations and which part of the information might be the cause for downstream symbolic predictions. Much of the work on explanations in NLP, especially when it comes to visual explanations, either utilizes dimensionality reduction or highlights salient features on the scale of words in a text [7]. However, it is not strictly necessary to reduce the dimensions of a word vector to visualize it. We tend to think of embeddings as vectors in high-dimensional space (e.g., 300 dimensions for GloVe embeddings) so that similar words are close to each other in this space. Yet a single word vector only consists of 300 numbers, while the numeric representation of an image might be made up of 6.000.000 numbers (a 1000px by 2000px RGB image). A word vector can easily be visualized as a kind of “barcode” of colors, with all 300 numbers arrayed in one dimension, the value of each number represented by the color. On this barcode, salient features (that is, the most critical dimensions in the vector) can easily be highlighted. This method has been used previously to produce visual explanations for NLP tasks by [22].

3. Visualizing Word Embedding

The method presented in this paper is based on this same idea: Even though the individual dimensions of high-dimensional word embeddings do not obviously correspond to meaningful features to human eyes, they arguably still represent different features of what context a word usually appears in. By visualizing and analyzing these individual dimensions, we hypothesize that we can discover some clues as to *which* information is encoded *where* in the word embedding. A word vector can be visualized as a kind of “barcode” of colors - but to make it easier for the human eye to differentiate the individual dimensions, it might be helpful to visualize the same vector as an image, e.g., 300 numbers as a 300 pixel (15px by 20px) image. The main problem with this method is that humans will intuitively attribute meaning to the distance or closeness of individual pixels (e.g., “This area over there...”). This meaning, however, does not exist in reality, as the order of dimensions in the vector is random. Thus, we want to find a more meaningful organization of the dimensions of a word vector in a 2-dimensional space to visualize concept-specific areas of word embeddings as 2D images, so-called *shadows*.

To organize where in the image the dimensions of the vector should be placed, that is, which pixel corresponds to which dimension, a self-organizing map (SOM) [23] presents an elegant solution. A SOM is trained on x examples, each represented by y features. The examples are then organized on a map. On these SOMs input vectors that are alike move closer together and ones that differ move away from each other by means of unsupervised clustering, i.e. learning vector quantization. When it comes to words represented by word embeddings, a naive approach would be to take for n words as examples and their k -dimensional word embeddings as their feature representation, which would result in a SOM that can place words on a map based on their embeddings (here, the SOM would be a method of dimensionality reduction). Our use case is different: We want to organize the dimensions in the embedding on a map. Thus, the k dimensions of the word vectors constitute the examples. The representations of these examples are the values of the dimension across the known words, meaning that each of the k examples

will be represented as a n -dimensional feature vector.

By training a SOM with as many neurons as the word embeddings have dimensions, it is possible to arrive at a model in which each dimension is recognized by exactly one of the neurons. Using this SOM, a word embedding (e.g., of the word 'the') can be visualized as an image with as many pixels as there are dimensions in the embedding. Each pixel is colored based on the value of the dimension that is associated with the corresponding neuron on the map.

3.1. Projecting Static Embeddings

We first used this method to investigate static word embeddings: We analyzed the 300-dimensional GloVe embeddings provided by the open-source natural language library spaCy [24]. The SOM is used to arrange the 300 dimensions of the word vectors in a (small) 2D image. Thus the SOM provides a map encoding which pixel in the image represents which of the dimensions of the word vector. The pixel is colored based on the value of the word vector in the associated dimension. This means that the SOM is not used later for predictions on new examples - it is only used once to construct this dimension-to-pixel map, and is not required to generalize at all, as there are no other possible examples beyond the known vector dimensions.

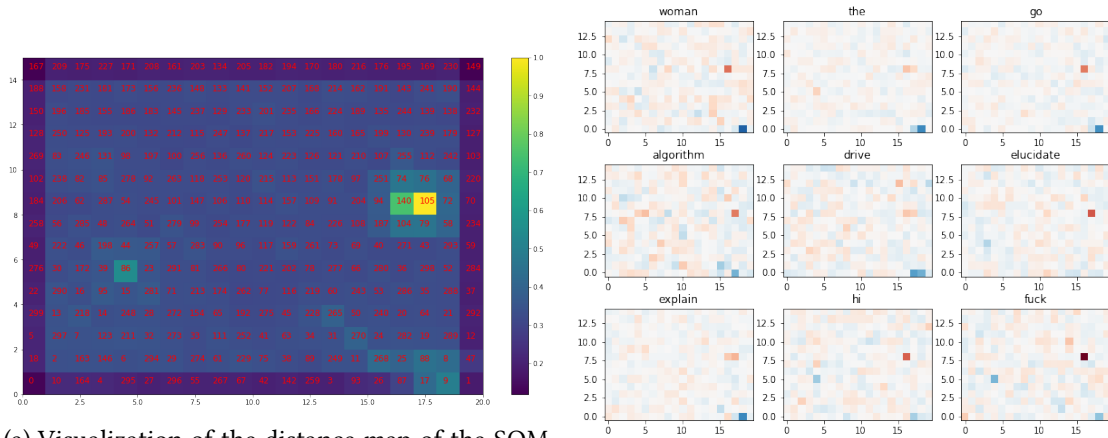
We trained a 15x20 SOM on the 300-dimensional GloVe embeddings of 10,000 unique English words to analyze static embeddings. There are around 170,000 words in the English language, but not all have pre-trained GloVe embeddings. We constructed a dataset of words by first collecting all lemmas included in WordNet [25] through NLTK [26]. From this set, we identified the 64,466 words included in spaCy as GloVe embeddings. Out of these, we took a random sample of 10,000 words on which to train the SOM, as we discovered through several trials that a corpus of 10,000 words is appropriate in terms of error and training time.

The training parameters of the SOM were adjusted empirically until the trained model arrived at a one-to-one matching of dimensions to neurons in the SOM (meaning that the SOM was able to correctly identify each dimension, as each neuron was trained to respond to exactly one of the dimensions). The trained SOM consistently achieved a quantization error of approx. 0.0005 over 2000 training iterations.

Figure 1 shows the layout of the trained SOM and a number of examples of words represented with the resulting layout. It stands to reason that those dimensions with values far from zero (positive or negative) contribute the most information, while those close to zero are less important. Therefore values at 0 are colored white, negative values are red and positive values are blue. The distance map of the SOM shows that there is overall very little variation, except for a few outliers located in three regions. These same regions can be found in the images showing a number of example words, where these pixels stand out in red or blue.

3.1.1. Analysis of Individual Words from Static Embeddings

Most interesting about the SOM shown in Figure 1a is that, while most of the neurons are relatively evenly spaced, there are several outliers - dimensions that are somehow more different from their neighbors than most. Most apparent are the pair of neurons corresponding to dimensions 140 and 105, the single neuron corresponding to dimension 86, and the cluster in the lower right corner. By comparing the map of the SOM in Figure 1a to the examples in



(a) Visualization of the distance map of the SOM trained on static word embeddings. The map is comprised of 300 neurons, organized in a 15x20 map. Each neuron represents exactly one of the 300 dimensions of the embedding; overlaid in red are the numbers of the dimensions as they are ordered in the word vectors.

(b) A number of example word vectors visualized as images based on the SOM organization. Values around 0 are white, negative numbers red, and positive numbers blue. It appears that the most yellow dimensions identified in the SOM also have among the highest absolute values.

Figure 1: Visualization of static word vectors.

Figure 1b, it becomes clear that the outlier neurons in the SOM correspond to those dimensions with greater absolute values than most.

Manual inspection of random words and their shadows (Figure 1b depicts a sample) revealed that in words of a comparatively high register (‘elucidate’), the right pixel (105) of the pair on the right stands out, and that in curse words, the left one (140) stands out strongly. We noticed that the two neurons 105 and 140, which appear as a pair in the SOM, never stand out together - it is always either one or the other that appears dark red. In some words (like ‘the’), neither stands out. Moreover, these two pixels often appear in dark red (negative value) but never in dark blue (positive value).

We inspected several synonyms of high-register words, such as ‘explain’ instead of ‘elucidate’, and found that neither pixel stood out for those. Furthermore, we also inspected several informal words such as ‘hi’, and found that in those, pixel 140 stood out almost as strongly as for curse words. We hypothesize that these dimensions capture the register of a word and act as opposites and calculated for each word x in the corpus an r so that:

$$r_x = \max(0, -x_{105}) - \max(0, -x_{140})$$

We then sorted all words by their r value. Those words with very low r -value are where pixel 140 is dark red while pixel 104 is neutral, and vice versa. Table 1 shows the ten words at either end of the list.

In the same way, we sorted the entire corpus of static embeddings based on the value at dimension 86 (a single pixel that stands out on the left of the map). This dimension apparently captures not the formality or register of words but instead seems to activate strongly if a word

word	140	105	r	word	140	105	r
'coercive'	0.89	-3.15	3.15	'fucking'	-4.21,	-0.04,	-4.17
'plurality'	0.55	-3.12	3.12	'ass'	-4.14,	-0.03,	-4.11
'minimise'	0.31	-3.11	3.11	'fuck'	-3.93,	0.08,	-3.93
'deleterious'	0.48	-3.10	3.10	'wanna'	-3.94,	-0.02,	-3.91
'lessening'	0.20	-3.07	3.07	'song'	-3.90,	0.63,	-3.90
'predetermined'	0.05	-3.05	3.05	'lol'	-3.87,	0.27,	-3.87
'concomitant'	0.90	-3.05	3.05	'bitch'	-3.85,	0.02,	-3.85
'societal'	0.04	-3.03	3.03	'cute'	-3.78,	0.38,	-3.78
'constraining'	0.62	-3.03	3.03	'ya'	-3.78,	0.29,	-3.78
'quantifiable'	0.80	-3.04	3.02	'movie'	-3.74,	0.40,	-3.74

(a) Highest r-values

(b) Lowest r-values

Table 1

Comparison of words at either end of the spectrum of r-values in the dataset. Duplicates (due to spelling differences and compounds like 'ass-kisser') are omitted. On the left (right) are the ten words in the dataset with the highest (lowest) r-value. Looking at this sample, it appears clear that those words where dimension 140 stands out are of a higher register, while the other group is decidedly informal.

is likely to appear in a pornographic context. We tried the same with the cluster of pixels in the lower right of the map, both with individual pixels and combinations of the group. While there were some similarities, these were not as clear or meaningful as observed before (for example, sorting by 17 & 9 produced many words related to Catholicism on the one end, including, e.g., 'antipope', 'tensured', 'archpriest'; and words which appeared related to customer service at the other, e.g., 'management', 'service', 'customer').

We also sorted the corpus of all words by their value in other random dimensions that do not stand out on the SOM, to assess whether these would also appear to indicate similar semantic explanations for their values. However, the lists of words produced from sorting by other dimensions had no seeming correlations or common characteristics.

We investigated the register-pair 140 & 105 further, testing what effect switching the value of the two dimensions might have on a word. When taking, for example, a word of high register such as 'elucidate', switching the respective values of dimensions 140 and 105 results in a new 300-dimensional vector that does not belong to any known word. However, searching through the corpus of all words for the most similar word to this switched vector (in terms of cosine similarity of the vectors) results in the word 'explain'. This connection holds for many words: applying the same technique to 'sufficient' results in 'enough', switching 'corrosion' leads to 'rust', 'covertly' to 'secret', 'occur' to 'happen', and so on - in a way, this can be used to find simpler synonyms.

3.2. Projecting Dynamic Embeddings

Analyzing the outlier dimensions in static word embeddings lead to some interesting insights. However, it seems that there are only very few dimensions that directly encode symbolic concepts such as register. For most other dimensions, the distance map of the SOM shows that there is very little variation, and their position in the resulting image is likely to be random. With dynamic embeddings, the word vector of a given word depends strongly on the context in which it appeared in the training text. Therefore, we examined whether visualizing these

vectors might make it possible to investigate the results of word classification tasks such as POS tagging. If the same word can be used either as a verb or as a noun, somewhere in the vector, some information should be encoded as to which concept is more likely at hand in the given context. Our aim was that by visualizing dynamic word vectors with the SOM mapping, we might be able to find regions - that is, groups of dimensions - that are of particular importance for specific POS concepts.

SpaCy also provides a trainable part-of-speech (POS) tagging model, which consists of two layers: one takes a text and predicts dynamic, 96-dimensional embeddings for each token in the text, and the second predicts POS tags for these tokens based on the embeddings. We used these 96-dimensional word vectors to investigate dynamic embeddings. To train a SOM on static embeddings, we collected the pre-trained GloVe embeddings of a list of words. Due to the nature of dynamic embeddings, however, this is not possible here; an actual text is required since the conceptual representation of a word differs depending on its current context. Therefore, we used the Brown corpus [27] and generated the dynamic embeddings from spaCy's pre-trained language model. As spaCy cannot process arbitrarily long texts, we only used the full sentences up to the 1.000.000th character. By removing punctuation and particle tokens, we obtained a dataset of 166.738 non-unique words with unique (dynamic) 96-dimensional word vectors.

First, we applied the same method as described above for static embeddings, training the SOM on the transposed matrix of word vectors. While there was some more variation in the SOM distance map, there did not appear to be any outliers as strong as in the static embedding map, and this method was not successful in differentiating between different POS concepts. Because of this, we took inspiration from two XAI approaches from computer vision research: the use of occlusion to analyze which features of the input representation are most important in the classification [18], and the generation of a prototypical image for a given class [20].

3.2.1. Masking the Shadows

By occluding parts of the word vectors, we hoped to find out which of the dimensions were actually necessary to recognize a word as a particular POS class, thus reducing the vector only to the essential areas. First, we tried this with words that the model had classified as a noun. For this, dimensions of the vector were occluded (set to zero) one by one, at each step choosing the dimension of which the removal had the least negative impact on the probability of the vector being a noun. This was repeated until the probability dropped below 99% and then until it dropped below 50%. The first few removals increase the confidence in the noun classification instead of decreasing it. Testing this with a large number of words revealed that confidence in the noun classification usually stayed above 50% until only a few dimensions were left, sometimes as little as two. However, the remaining dimensions (visualized as pixels in the image) are not always the same, although many of the dimensions reappear over repeated tests.

We repeated the same process for all POS classes that the spaCy model identifies, which are based on the Penn Treebank classes [28]. The results are that most of the dimensions in a word vector are irrelevant for it to be classified as the same POS with above 50% confidence. This does not change if all but a few dimensions are reduced, when almost the entire vector is occluded, the prediction changes to a different class. Interestingly, NN (singular noun) appears to be the default classification: a vector with only 0s is classified as a noun, albeit with low confidence.

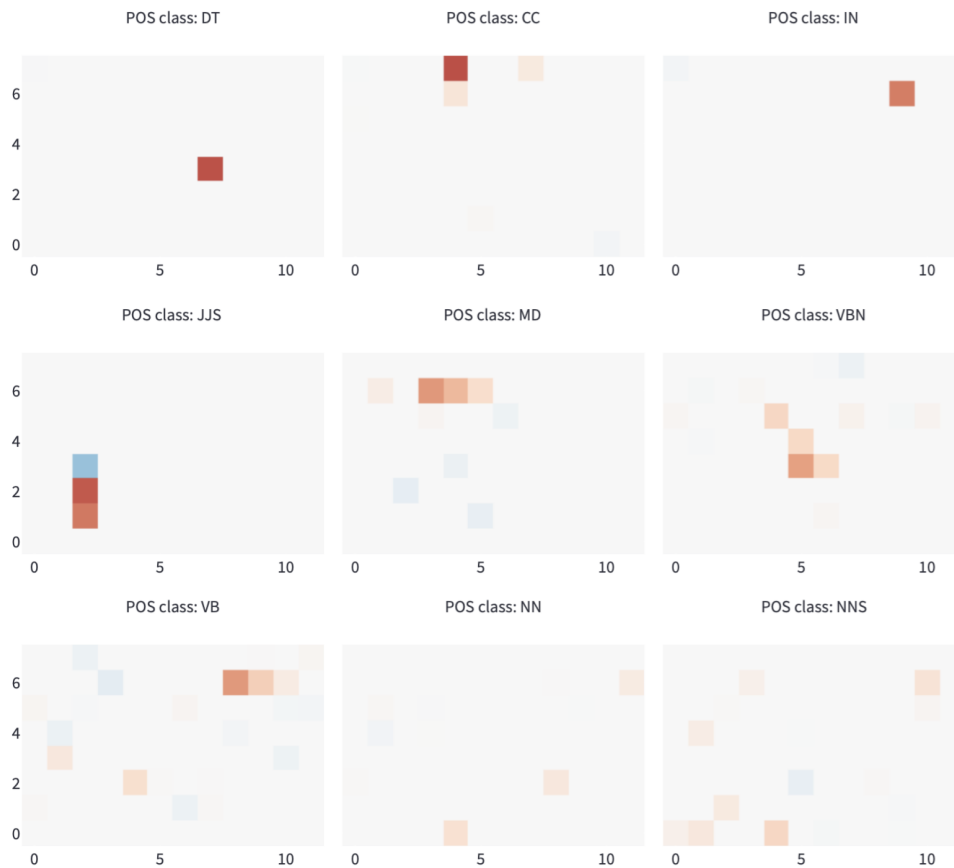
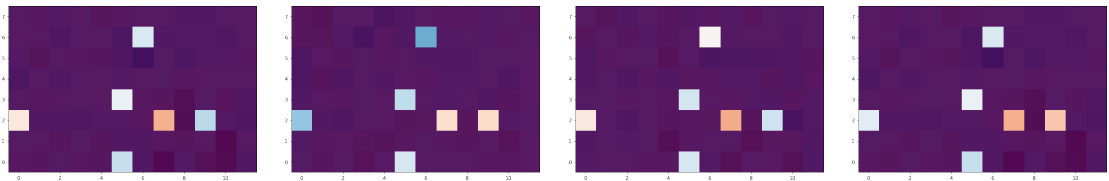


Figure 2: Average Shadows of POS Concepts

3.2.2. Projecting Prototypical Shadows

Next, we systematically tested the outcome of occlusion by reducing the vectors of all the tokens in the dynamic embedding corpus that the model had originally classified with high confidence, that is greater than 99%, until confidence dipped below 50%. For each POS class, we calculated the average of these reduced vectors. Figure 2 shows the resulting images for a selection of classes. In some cases, such as for DT (determiners), the result of the reduction almost always leaves the same dimension unoccluded, leading to an average image where only one or few dimensions appear very strongly. However, for others such as NN, there are many different possible results of the reduction. Thus, the average image is more translucent, and does not show a specific region. Classes such as MD, VB, or VBN (different verb forms) seem to be concentrated around different regions. It appears that for POS classes that can be considered conceptually more precise, there are only a few dimensions that are often or always very important for the classification. This is especially the case for those classes with a limited number of possible words, or which are marked by their form such as comparative or superlative adjectives. In contrast, concepts like nouns or verbs are more difficult to grasp.



(a) Used as a noun, incorrectly classified as a verb. (b) Used as a noun, correctly classified as a noun. (c) Used as a verb, correctly classified as a verb. (d) Same vector as in (a) with two inverted values.

Figure 3: Four different vector representations of the word ‘garlic’, overlaid to highlight which pixels are especially relevant for classification as nouns. Here, we use a purple overlay over those dimensions which did not appear in at least 1/4 of minimal nouns.

3.2.3. Analysis of POS-tagging from Dynamic Embeddings

It appears that for POS classification, it is possible to identify areas in the vector images that are most important for the model to identify different POS classes. Therefore, we used these visualizations to investigate a problem that we had come up against repeatedly in prior work: models that are fine-tuned from pre-trained embeddings tend to struggle with very domain-specific language that differs from more standard texts. In particular, we have often struggled with the problem that recipe texts employ a kind of language that makes it difficult to identify the main verb of a sentence. This can be due to, for example, words being used as both verbs and nouns (e.g. ‘juice’), other words being left out (e.g. “chop tomatoes” instead of “chop the tomatoes”), missing punctuation, etc.

First, we decided to investigate one particular word, which we had stumbled upon in a previous study as an example that spaCy’s POS tagging model misclassified. We looked at three sentences containing the word ‘garlic’:

- (1) Add the garlic to the pan.
- (2) Add cauliflower and garlic mixture to the pot, mixing carefully to combine.
- (3) You have to garlic and salt the food.

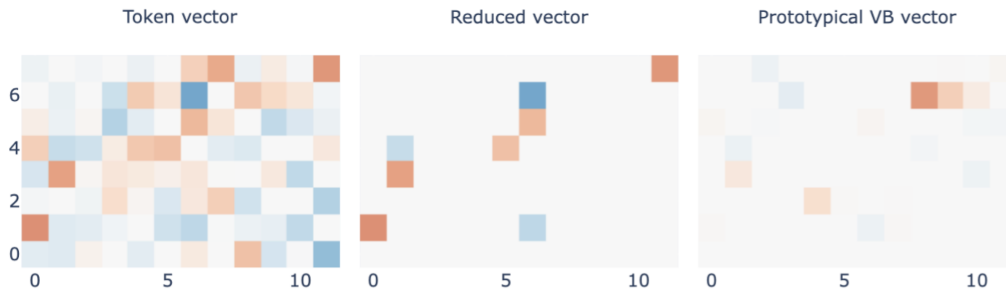
In sentence (1), ‘garlic’ is correctly classified as a noun. In sentence (2), however, it is incorrectly classified as a verb with a probability of 49%, while the noun tag only has a probability of 46%. Sentence (3) is an example in which ‘garlic’ is used as a verb and correctly classified as such.

Figure 3 shows the visualizations of the three different vectors representing the word garlic in these three different sentences. As the confidence for the second version is already quite low, and reducing the vectors would lead to different dimensions being left unoccluded, we did not reduce the vectors here. Instead, we masked most of the image, leaving those dimensions highlighted that were most often (across the whole corpus) unoccluded at 50% confidence.

It appears that the vector of the noun use of garlic, which was incorrectly classified as a verb (sentence (2), 4a), most strongly differs from the correctly classified noun (sentence (1), 3b) in the pixel on the far left at (0,2) and the one on the right at (9,2). Those two pixels are the same color in the vector representing garlic as a verb (sentence (3), 4b), opposite colors from the noun in 3b. Thus, we inverted these two pixels by multiplying their respective values with -1.



(a) Vector representation of 'Heat' in "Heat oil in a deep frying pan or wok until very hot."



(b) Vector representation of 'Heat' in "Heat some vegetable oil in the same frying pan you used before."

Figure 4: Two different vector representations of the word 'Heat'. The three images show the entire embedding, then the same reduced to 50% confidence, and then the prototypical concept shadow of the POS class it was classified as.

Figure 3d depicts the result of these inversions. We used this 'corrected' vector as input for spaCy's POS tagging model. As expected, the model now classifies this vector as a noun, with a confidence of 88%. This means that we were able to visually identify the exact dimension that was the reason for the incorrect classification of this token.

Next, we tried a slightly different approach with another problem, where a verb was incorrectly classified as an adjective, as seen in 4. As noun seems to be the default POS class, reducing the vectors of noun tokens leaves only very few dimensions unoccluded at 50%, and comparing them to the conceptual shadow shown in Figure 2 is not very helpful. However, this is not a problem for adjectives. We, therefore, considered two sentences:

- (4) Heat oil in a deep frying pan or wok until very hot.
- (5) Heat some vegetable oil in the same frying pan you used before.

The sentence (4), the first token 'heat' was incorrectly classified as an adjective instead of a verb. Therefore, we looked at the reduced vector of the token, as well as the conceptual shadows of the verb and adjective classes. Those were compared to a vector from sentence (5), where the same word, 'heat', in the same position in the sentence, was classified as a verb correctly. Figure 4 shows these images. The two vectors that both represent the word 'heat' clearly share some features. Interestingly, many of the dimensions that are left in the reduced vectors are similar in both versions - clearly, small changes are enough to switch the classification from verb to adjective.

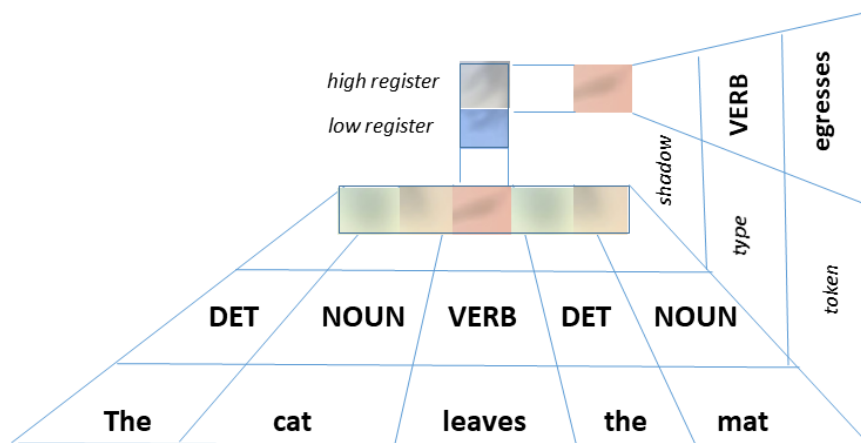


Figure 5: The general picture, showing a set of tokens in a given context, their conceptual types and the corresponding conceptual shadows (as illustrations).

4. Discussion

The nature of this work is rather exploratory. The results of our experiments shed some light on how the meaning of linguistic concepts is encoded in high-dimensional word embeddings, which until now have been a black box in NLP that was quite securely closed. Addressing the cognitive elephant in the room, it is clear that human cognition is based on combinations of statistical processes together with increasingly symbolic generalizations over the extracted patterns. Some well-known phenomena such as prototypicality effects or radial categories [29] are out of the scope of most symbolic approaches, yet become quite easy to see in the conceptual shadows shown herein - where we can compare the shadows of very prototypical nouns and verbs to ones that are less *nouny* or *verbly*. We are not aware of many other works which use visualizations of word vectors in their entirety, apart from the “bar code”-like images described in the beginning. By organizing the dimensions of these vectors on a map by training a SOM, we were able to identify areas of interest, as well as dimensions that appear to “belong together” as the pair of dimensions that seems to encode the register of a word. Together with the regions decoding POS, we can now form rudimentary ensembles of shadows that encode, for example, high-register nouns or vernacular verbs as depicted in Figure 5.

So far, this method has allowed us to identify small areas which appear to have recognizable tasks in the semantic representation, and to point out which of the dimensions of a word vector might be responsible e.g., for an incorrect classification. However, this in turn poses the question of why the dimension in question was “wrong” in the first place. To investigate this, we have to follow this lead one step deeper, and investigate what resulted in this particular weight when the vector was generated from the input text. One application for this work is to use it as a starting point from which to analyze downstream errors in NLP tasks and explain their origins.

It is important to point out that any conclusion drawn from these visualizations is only ever related to the specific set of vectors on which the SOM was trained. A different kind of static

embedding than GloVe might very well result in a very different map, with different outlier dimensions which might not appear to hold similar meaning to the ones we found here. This, however, is more feature than bug in our minds, as we visualize how a specific sub-symbolic system encodes conceptual dimension, which is – by its very nature – based on its training. In spite of the current limitations of the work presented above, we find that mapping the individual dimensions of word embeddings as a 2D image makes it possible to gather fascinating insights into the internal makeup of distributed semantic representations. We hope that this kind of low-level analysis of embeddings can serve as a starting point to gain deeper understanding of neural networks used in NLP and other symbolic classification tasks.

References

- [1] S. Farrar, T. Langendoen, A linguistic ontology for the semantic web, *GLOT International* 7 (2004) 97–100.
- [2] A. Gangemi, R. Navigli, P. Velardi, The ontowordnet project: Extension and axiomatization of conceptual relations in wordnet, in: R. Meersman, Z. Tari, D. C. Schmidt (Eds.), *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 820–838.
- [3] P. Cimiano, P. Buitelaar, A. Frank, S. Racioppa, M. Sintek, M. Kiesel, M. Romanelli, B. Loos, T. Declerck, R. Engel, D. Sonntag, V. Micelli, R. Porzel, *Linginfo: Design and applications of a model for the integration of linguistic information in ontologies*, in: *Proc. of OntoLex at LREC, ELRA*, 2006, pp. 28–32.
- [4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, *arXiv preprint arXiv:1310.4546* (2013).
- [5] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [6] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, L. Kagal, Explaining explanations: An overview of interpretability of machine learning, in: *Proc. of DSAA 2018, IEEE*, 2018, pp. 80–89.
- [7] M. Danilevsky, S. Dhanorkar, Y. Li, L. Popa, K. Qian, A. Xu, Explainability for natural language processing, in: *Proc. of KDD 2021*, 2021, pp. 4033–4034.
- [8] I. Niles, A. Pease, Towards a standard upper ontology, in: *Proc. of FOIS 2021*, Association for Computing Machinery, New York, NY, USA, 2001, p. 2–9.
- [9] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, *Wonderweb deliverable d18, ontology library (final)*, ICT project 33052 (2003) 31.
- [10] R. Porzel, V. S. Cangalovic, What say you: An ontological representation of imperative meaning for human-robot interaction, in: *Proc. of JOWO 2020*, volume 2708 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020.
- [11] A. Gangemi, P. Mika, Understanding the semantic web through descriptions and situations, in: *Proceedings of the ODBASE Conference*, Springer, 2003.
- [12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).

- [13] A. Akbik, D. Blythe, R. Vollgraf, Contextual string embeddings for sequence labeling, in: Proc. of COLING 2018, 2018, pp. 1638–1649.
- [14] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018).
- [15] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, arXiv preprint arXiv:1706.03762 (2017).
- [17] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?": Explaining the predictions of any classifier, in: Proc. of KDD 2016, ACM, New York, NY, USA, 2016, p. 1135–1144.
- [18] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 818–833.
- [19] A. M. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, J. Clune, Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, CoRR abs/1605.09304 (2016). URL: <http://arxiv.org/abs/1605.09304>. arXiv: 1605. 09304.
- [20] A. Nguyen, J. Yosinski, J. Clune, Understanding neural networks via feature visualization: A survey, in: Explainable AI: interpreting, explaining and visualizing deep learning, Springer, 2019, pp. 55–76.
- [21] D. Doran, S. Schulz, T. R. Besold, What does explainable ai really mean? a new conceptualization of perspectives, arXiv preprint arXiv:1710.00794 (2017).
- [22] J. Li, X. Chen, E. Hovy, D. Jurafsky, Visualizing and understanding neural models in nlp, in: 2016 north american chapter of the association for computational linguistics, Association for Computational Linguistics, 2016, pp. 681–691.
- [23] T. Kohonen, The self-organizing map, Proceedings of the IEEE 78 (1990) 1464–1480. doi:10.1109/5.58325.
- [24] M. Honnibal, I. Montani, spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing, 2017. To appear.
- [25] G. A. Miller, Wordnet: A lexical database for english, Commun. ACM 38 (1995) 39–41. URL: <https://doi.org/10.1145/219717.219748>. doi:10.1145/219717.219748.
- [26] S. Bird, E. Klein, E. Loper, Natural language processing with Python: analyzing text with the natural language toolkit, " O'Reilly Media, Inc.", 2009.
- [27] W. N. Francis, H. Kucera, Brown Corpus Manual, Technical Report, Department of Linguistics, Brown University, Providence, Rhode Island, US, 1979. URL: <http://icame.uib.no/brown/bcm.html>.
- [28] M. Marcus, B. Santorini, M. A. Marcinkiewicz, Building a large annotated corpus of english: The penn treebank (1993).
- [29] E. Rosch, Cognitive representations of semantic categories, Journal of Experimental Psychology: General 104 (1975) 192–233. doi:10.1037/0096-3445.104.3.192.