university of groningen

faculty of science and engineering

Nested State Clouds: Distilling Knowledge Graphs from Contextual Embeddings

Bachelor's Project Thesis

Paul Bricman, s3908194, p.a.bricman@student.rug.nl,
Supervisors: Prof. Dr. Herbert Jaeger, Dr. Jacolien van Rij-Tange

Abstract: Interpretability techniques help ensure the safe deployment of deep learning (DL) models into production by providing practitioners with diverse debugging tools, yet the inner workings of large models remain elusive. In this work, we propose a novel interpretability technique which can be used to distill sparse knowledge graphs from a model's high-dimensional embeddings using conceptors. This technique, termed Nested State Clouds (NSC), takes advantage of the way state clouds of contextual embeddings are positioned relative to each other in latent space. For instance, ”fruit” contextual embeddings appear to engulf ”apple” ones, as the former includes not only the senses of the latter, but some additional ones as well. We successfully apply NSC to a pretrained masked language model, and recover an ontology of concepts grounded in the model's latent space.

1 Introduction

1.1 Background

In the past years, DL models have been claimed to reach human parity on a range of tasks which were deemed challenging only a few years ago. For instance, machine translation is deemed on par with human translators on popular language pairs [Toral et al., 2018], scientific DL models are more accurate than explicit hand-crafted ones in a growing range of fields [Ravuri et al., 2021], while reinforcement learning agents based on DL models have outperformed professionals in multiple video games [Schrittwieser et al., 2020] and industrial control applications [Degrave et al., 2022].

The resounding success of recent DL work has in large part been attributed to the newly-gained ability of DL models to automatically extract relevant features from input data [LeCun et al., 2015], rather than making use of hand-crafted features. This has proven effective in advancing the state-of-the-art on numerous DL tasks [Radford et al., 2021] [Radford et al., 2018]. Concretely, the automatically-derived features are represented through the specific activation patterns of hidden layers as information propagates through the DL model [LeCun et al., 2015]. Besides early layers being able to already extract surface input features, a recurring finding has been the fact that input representations become increasingly abstract with each sequential layer, before collapsing again to the concrete particularities of the output towards the final layers [Tenney et al., 2019]. When such representations are continuous and dense, they are referred to as embeddings – high-dimensional vectors subjected to sequential transformations specified through the model's parameters.

In the context of our increased reliance on DL models on a societal level, the field of explainable AI (XAI) investigates methods for interpreting the inner workings of such models, which otherwise lack clear internal rules due to being largely trained on raw data [Arrieta et al., 2019]. Techniques of this kind help researchers debug such DL systems, ensuring their safe use in practice (e.g. avoids toxic cultural biases, is aligned to human values). A pervasive trade-off in XAI, however, is the conflict between formulating explanations which (1) accurately reflect the actual processing performed by DL models (i.e. functionally-grounded), yet (2) are highly comprehensible and intelligible for humans (i.e. human-grounded) [Madsen et al., 2021]. This resembles a constant conflict faced in machine translation (MT), where models should (1) adequately preserve input meaning (i.e. adequacy), while (2) producing a coherent translation in itself (i.e. fluency) [Koehn, 2017]. Framing explainable AI as a translation task from machine to human representations has proven a useful lens for understanding the obstacles faced by the current work, as the adequacy-fluency trade-off in MT is intuitive.

Given the central role of input representations in recent DL models, a large body of advances in XAI has focused on distilling high-dimensional embeddings into a form which is interpretable by humans [Madsen et al., 2021]. As the embeddings themselves arguably lack meaning when not grounded in input or output data, the vast majority of such work has specifically attempted to highlight how a certain DL model relates inputs and outputs by means of embeddings.

For instance, prior work has highlighted toxic gender biases in word embeddings (e.g. "woman" being represented as closer in meaning to "nurse" than "programmer" by a DL model), which promptly led to debiasing techniques being developed by the NLP community [Bolukbasi et al., 2016]. Moreover, methods have been developed to construct ontologies from embeddings by means of hierarchical clustering, in order to better understand how the underlying DL model groups concepts together [Liu et al., 2018]. As another line of research, behavioral and structural probes have been employed to locate the "site" of various computations (e.g. part-of-speech tagging in NLP), by means of relating embeddings from different layers to cruder external representations (e.g. part-of-speech) [Tenney et al., 2019]. Additionally, methods have been suggested to explicitly represent abstraction relations using embedding "columns", hinting at future DL models which are partially explainable themselves, even before making use of post-hoc XAI tools [Hinton, 2021].

However, even when interpretability techniques focus on directly relating inputs to outputs, embeddings are often involved as mediators [Danilevsky et al., 2020]. As an example of tracing specific input influences on the output, feature explanations highlight which particular aspects of the input data have been most influential in yielding the output [Jain and Wallace, 2019]. Alternatively, techniques based on counterfactuals and adversarial examples aim to find marginally different inputs which cause massive changes in output [Madsen et al., 2021]. As a particularly ergonomic family of explanations, methods have also been developed to incentivize models to directly explain themselves in natural language [Madsen et al., 2021]. Finally, another effective technique is based on extracting knowledge graphs by inferring entity-relationship-entity triples directly via (masked) language modeling [Wang et al., 2020].

In this context, we extend the existing toolkit of interpretability techniques by introducing a novel approach termed Nested State Clouds (NSC). This technique can be used to distill highly-comprehensible knowledge graphs directly from sets of high-dimensional embeddings, with no modality constraints. In other words, given a DL model and an auxiliary dataset, NSC appears to be a promising candidate technique for automatically organizing concepts in an abstraction hierarchy which reflects the model's internalized knowledge. An important benefit of NSC is its modality-agnostic design (i.e. appears applicable to DL models operating with various non-text modalities), in stark contrast to methods which only focus on distilling knowledge graphs from text [Wang et al., 2020]. By automatically analyzing the way state clouds of contextual embeddings are positioned relative to each other in latent space, NSC appears capable of distilling high-dimensional representations which have been abstracted away from particular modalities. Through the present work, we investigate the question of whether the modality-agnostic spatial layout of high-dimensional embeddings can be meaningfully interpreted so as to yield a relevant knowledge graph.

However, in the context of this paper, we limit ourselves to an initial investigation of a pretrained masked *language* model. Given this, we leave the applicability of NSC to arbitrary classification models (e.g. ViT [Dosovitskiy et al., 2021]) for future work. We speculate on the tractability of generalizing NSC in the discussion by hinting at the possibility of making use of state clouds obtained from individual class member embeddings, rather than token embeddings.

Attempting to place NSC in the existing landscape of interpretability techniques, we note that our approach can generate global explanations (i.e. which attempt to describe the model's processing across multiple inferences) which are provided post-hoc (i.e. after training the model) [Danilevsky et al., 2020]. This is in contrast to those XAI techniques which yield local explanations (i.e. describing the way a particular inference unfolds across layers) and those which are provided during the actual training of inherently interpretable models.

1.2 NSC Overview

Before introducing the core ideas behind NSC, we specify in more depth several terms which will be made heavy use of. By *symbol*, we refer to the explicit way in which a concept is represented in text (e.g. the string ”fruit”). Interestingly enough, our discussion illustrates plainly how the same symbol can refer to multiple concepts. By *exemplar*, we refer to a specific instance of a symbol in a certain context (e.g. ”fruit” in ”It's healthy to eat **fruit**s regularly.”). Importantly, each exemplar is a case of a symbol assuming specific semantics which are unique to its context (e.g. ”fruit” could refer to an apple or a banana, depending on context). In practice, those specific semantics are represented numerically by means of contextual embeddings [Devlin et al., 2019]. For the purposes of this work, we generally assume an exemplar-based view of concepts (e.g. defining the concept of fruit by means of the finite totality of its exemplars), as opposed to a prototype-based one (e.g. defining the concept of fruit based on one general idealized prototype).

NSC works by first generating a state cloud of contextual embeddings for each given symbol, representing the semantics of its exemplars (subfigure B of Figure 2.1). A state cloud is simply a set of such high-dimensional embeddings, which inevitably comes to exhibit a certain shape and directionality when regarded as a whole. This generation process is based on simply using the investigated model in inference mode together with the auxiliary corpus. For instance, the symbol ”fruit” can refer to a host of different objects, depending on the context of use – variation which is captured by distinct contextual embeddings. Following this initial step, NSC will have formed a collection of one state cloud of contextual embeddings per symbol. Each such state cloud will represent the distribution of semantics assumed by the various exemplars in their respective contexts.

Next, NSC compactly represents the overarching shape of each resulting state cloud as a *conceptor* identified with a high-dimensional ellipsoid, instead of a large set of contextual embeddings [Jaeger, 2017] (subfigure C of Figure 2.1). We find that this representation often results in orders-of-magnitude lower memory footprint compared to the naive approach of storing all individual embeddings. While the ellipsoids are conceptually similar to the ones obtained using principal component analysis (PCA), we opted for using conceptors as compact high-dimensional objects due to existing literature investigating meaningful ways of relating them to each other [Jaeger, 2017]. Specifically, an inherent *abstraction ordering* has been previously defined for pairs of conceptors (subfigure B of Figure 2.2). This refers to a means of comparing two such objects in terms of their level of abstraction, a process which we are exploiting in the present paper in an attempt to generate a knowledge graph. Loosely speaking, a conceptor which spatially engulfs another can be said to be more abstract, as the former encompasses a broader region of semantic space than the latter. We refer the reader to the relevant methods subsections for details on conceptors and their abstraction ordering.

After generating one state cloud per symbol, representing them as conceptor objects, and conducting pairwise comparisons of abstraction, an optimization algorithm is employed to generate the final output of NSC: a knowledge graph (subfigure C of Figure 2.2). Concretely, the algorithm based on simulated annealing is iteratively refining a directed graph which aims to accurately represent the estimated relations of abstraction. In contrast to simply constructing a directed graph by adding a new arc for each positive abstraction relation, an optimization algorithm allows us to better deal with noise. Additionally, the optimization framing enables us to specify additional ”nice-to-have” properties of the desired output graph [Madsen et al., 2021]. For instance, we penalize high numbers of arcs, parents per node, and children per node, in an attempt to keep the output explanations sparse and highly legible. In this, the objective function of the optimization algorithm essentially provides a ”slider” between functionally-grounded and human-grounded explanations.

In the first half of Section 2, we describe the model we are later applying NSC *to*. In the second half, we describe in more depth the individual stages of the NSC algorithm. In Section 3, we investigate the preliminary results of the novel technique. In Section 4, we explore potential issues of the technique and highlight opportunities for future work.

2 Methods

2.1 Model

To investigate the feasibility of NSC, we attempted to apply it to a pretrained BERT model, short for Bidirectional Encoder Representations from Transformers [Devlin et al., 2019]. For completeness, BERT is a transformer model which maps a set of subword tokens to another set of such tokens. BERT has been originally trained on two different natural language processing objectives using a mixed corpus comprised of public-domain books and an English wikipedia dump. First, it has been tasked with a masked language modeling (MLM) objective. This refers to the task of reconstructing a short input text which has been intentionally corrupted. The corruption typically consists in eliminating (i.e. masking) a random proportion of the tokens contained in the input text (e.g. "BERT is a transformer model." might be corrupted as "BERT is a [MASK] model."). Given this, the MLM task consists in reconstructing the pre-corruption text from the corrupted version.

The second objective employed in training BERT is a next sentence prediction (NSP) task. Given a pair of two sentences, BERT is tasked with predicting whether they are consecutive in the original text. The combination of those two conceptually simple objectives has been shown to help BERT learn rich semantic representations of the text being processed, as an instrumental goal in solving the two tasks. For instance, mean-pooling token embeddings across texts has been shown to be highly effective in downstream information retrieval tasks based on vector similarity [Reimers and Gurevych, 2019]. Moreover, mean-pooling token embeddings of a text and comparing the result with the mean-pooled embeddings of a set of labels (e.g. science, politics, economics), has been shown to be a competitive baseline in text classification [Yin et al., 2019]. Alternatively, BERT models fine-tuned on limited data from other tasks (e.g. natural language inference) had yielded state-of-the-art performance in multiple tasks [Jiang and de Marneffe, 2019].

Internally, BERT represents each input token (e.g. "fruit") as an embedding of dimensionality 768. The means of obtaining those representations are gradually learned during the training phase, as the model implicitly selects suitable information to store in this high-dimensional vector. As the model consists of a repetitive sequence of layers, the set of embeddings which represents tokens is adjusted from one layer to the next using multi-head attention mechanisms – means of routing information effectively across layers [Bahdanau et al., 2015]. It is precisely those token embeddings which we are trying to distill knowledge graphs from using the NSC approach. In the current study, we are attempting to interpret the set of embeddings which are generated in the *last* BERT layer. This has been hypothesized to contain high-level features extracted the input tokens which are based in large part on the tokens' contexts, after extensive processing in the earlier layers of the model [Tenney et al., 2019].

We opted for BERT due to its widespread use in industry applications and its large number of derivative models (e.g. RoBERTa [Liu et al., 2019], ALBERT [Lan et al., 2020], distilBERT [Sanh et al., 2020], DeBERTa [He et al., 2021], etc.). As mentioned before, BERT takes in a sequence of subword tokens as input and reconstructs it as output, while generating a unique contextual embedding for each token. Crucially, the same token can be attributed different embeddings in different contexts (e.g. "she" referring to different people). In practice, the contextual embeddings of individual tokens are mean-pooled together to yield an overarching document embedding. However, here we focus only on the contextual embeddings of *individual* tokens or at most short sequences of them which form a noun phrase (e.g. "orange juice"), so as to be able to investigate relations between specific concepts.

2.2 Data

As NSC requires an auxiliary dataset for generating state clouds of contextual embeddings, we employ one of the datasets which have been used for training the BERT model, namely BookCorpus [Devlin et al., 2019]. This corpus consists of 11,038 public-domain books across 16 different genres (985M words, 74M sentences), and provides many different contexts for symbols to appear in as exemplars [Zhu et al., 2015].

We focus our investigation on relating a set of 100 hand-picked concepts to each other. For each concept, we extract all contexts in which their associated symbols (e.g. the string "fruit") appear verbatim in the dataset as exemplars (e.g. the various instances of "fruit" which assume different semantics based on context). This is illustrated in subfigure A of Figure 2.1. A context is concretely defined as the span of text starting 300 characters before and ending 300 characters after the concept occurence. Additionally, we trim the incomplete beginning and ending sentences (i.e. trailing) from each context, leaving in only complete sentences surrounding the concept occurence.

For each context, we extract the contextual embedding of the exemplar, thus obtaining a set of such embeddings for each concept. This is illustrated in subfigure B of Figure 2.1. The cardinality of each set depends on the frequency of occurence of the respective symbol in the dataset. We further filter our set of symbols being considered based on the cardinality of their state clouds, eliminating concepts which had fewer exemplars than the number of BERT embedding dimensions (i.e. 768). We cover difficulties in handling sparser state clouds in the discussion.

2.3 Conceptors

From each remaining state cloud representing the set of exemplars contained in the dataset, we obtain a conceptor [Jaeger, 2017]. This is illustrated in subfigure C of Figure 2.1. For completeness, a conceptor is a mathematical object which models the distribution of state cloud in the space it populates. However, conceptors do *not* represent the *density* of embeddings in space using a probability density function. Rather, a conceptor represents the orthogonal *directions* across which the state cloud spreads most across space, together with the spread associated with each direction. This information can be compactly represented in a square matrix whose dimensionality matches the one of the space populated by the state cloud.

Both conceptors and Principal Component Analysis (PCA) specify high-dimensional ellipsoids by means of the correlation matrix of the state cloud. However, an additional parameter appears in the case of conceptors. Specifically, a conceptor also requires an aperture to be defined. The aperture α is a parameter which dictates the extent to which the shape of the state cloud is reflected in the associated conceptor object. For increasing aperture values, the conceptor matrix approaches the identity matrix. For decreasing aperture values, the conceptor matrix approaches the zero matrix.

Obtaining a conceptor from a state cloud is straight-forward and computationally cheap. Given the correlation matrix of the state cloud R and a real value α specified for the aperture parameter, the conceptor matrix can be obtained through the following closed-form equation:

$$C(R, \alpha) = R(R + \alpha^{-2} I)^{-1}.$$

In the present work, we only obtain conceptors from state clouds composed of contextual embeddings generated by the pretrained BERT model. Each state cloud contains contextual embeddings associated with one symbol composed of one or several tokens (e.g. "orange juice"), while each individual contextual embedding is associated with a specific exemplar. Given that we obtain conceptors from state clouds of BERT contextual embeddings, the dimensionality of the state cloud matches that of the embeddings, namely

$$d_{BERT} = 768.$$

We note that the interpretability technique we introduce does not require this specific dimensionality. Rather, the specific value of d_{BERT} is only an artifact of the investigated model's architecture. It is likely that NSC could be applied with minimal modifications to state clouds of lower or higher dimensionality. In fact, some of the experiments used to introduce conceptors as mathematical objects make use of state clouds of only several dimensions.

Additionally, it is useful to note the large difference in terms of memory footprint observed between a state cloud of BERT embeddings and a conceptor matrix derived from it. A state cloud containing $n_{embs} = 10^6$ BERT embeddings of dimensionality $d_{BERT} = 768$ naively requires $n_{embs} \cdot d_{BERT} = 7.68 * 10^8$ floating point values to fully represent. In contrast, the conceptor matrix obtained from the same state cloud, given a certain aperture, is a square matrix with d_{BERT} rows and columns. Hence, it only needs $d_{BERT}^2 \approx 5.89 * 10^5$ floating point values to be represented. In this specific case, the conceptor represents the state cloud

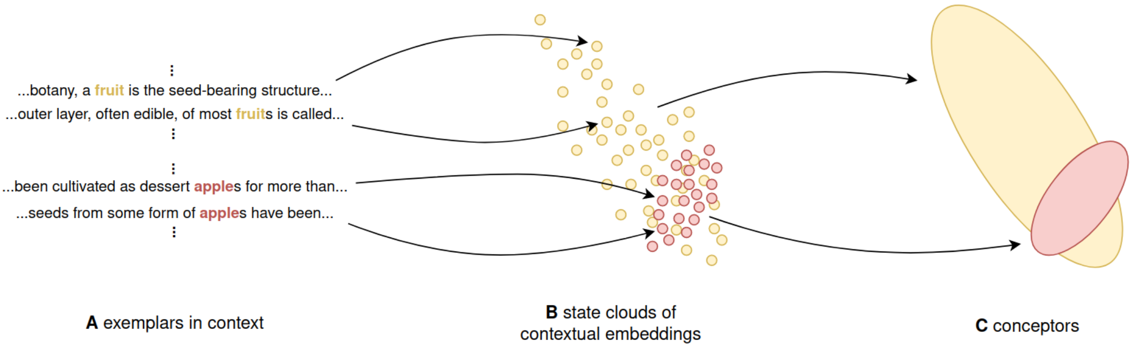
...botany, a fruit is the seed-bearing structure...
...outer layer, often edible, of most fruits is called...
...been cultivated as dessert apples for more than...
...seeds from some form of apples have been...
A exemplars in context
B state clouds of contextual embeddings
C conceptors

Figure 2.1: A: Exemplars are located and extracted together with their surrounding contexts. B: Each exemplar is encoded into a contextual embedding. Notably, the same symbol is encoded into different such embeddings based on context, resulting in a state cloud per symbol. C: Conceptors are derived from each state cloud as more compact representations.

with an approximately 10^3 times smaller memory footprint. Moreover, the memory footprint of the conceptor matrix is constant with respect to the cardinality of the state cloud. It is only the accuracy of representing the state cloud which increases with more samples in the form of new contextual embeddings, as the correlation matrix converges. The implication of this is that state clouds associated with symbols which have relatively high frequency in the auxiliary dataset get represented through conceptor matrices of the same size as those associated with symbols with relatively low frequency. That said, storing large numbers of conceptor matrices can still become expensive. When this is the case, a trade-off between numerical precision and memory footprint becomes relevant.

2.4 Abstraction Ordering

For each pair of conceptors (C_1, C_2) obtained from different state clouds of contextual embeddings, we attempt to estimate how they relate to each other in terms of abstraction. This is illustrated in subfigure A of Figure 2.2. We aim to determine whether (1) C_1 represents a concept which is more abstract than the one represented by C_2 (e.g. "fruit" $>$ "apple"), whether (2) it is the other way around (e.g. "apple" $<$ "fruit"), or whether (3) the two concepts represented lack a meaningful abstraction ordering (e.g. "fruit" $<>$ "galaxy"). Ideally, we would only want the final knowledge graph generated by NSC to represent valid relations of decreasing abstraction by means of arcs (e.g. "fruit" $>$ "apple", NOT "fruit" $<$ "apple"). This would conceivably lead to a knowledge graph which approaches a hierarchical structure.

In its original formulation, abstraction ordering of conceptors has two important characteristics [Jaeger, 2017]. First, this prior art only describes the three mutually-exclusive cases above, with hard limits. C_1 is described to be more abstract than C_2 if and only if the difference matrix $C_1 - C_2$ is positive definite. Due to the symmetry of abstraction ordering, $C_2 > C_1$ if and only if the difference matrix $C_2 - C_1$ is positive definite. In the third case, the conceptor matrices are equal.

Unfortunately, real data is noisy, making it extremely unlikely that the unambigious criterion of positive definiteness ever holds for conceptors obtained from non-synthetic data (e.g. BERT contextual embeddings). Besides the inevitable aleatoric noise associated with non-synthetic data, abstraction ordering in its original formulation is also hindered by the cumulative error introduced by limited machine precision when dealing with floating point values. Approaches from numerical methods, however, might help mitigate the impact of this second source of noise.

Besides requiring standards of precision and signal-to-noise ratio which are non-trivial to attain in practice, abstraction ordering in its original for-

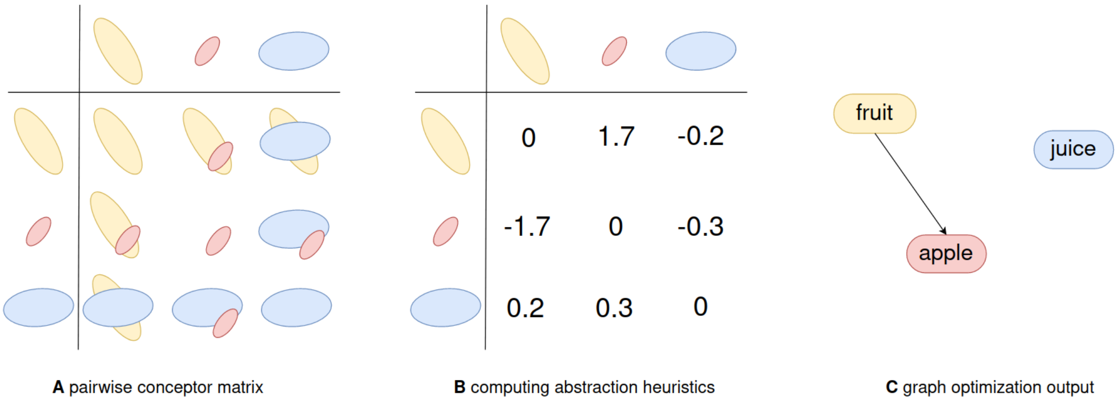
0
1.7
-0.2
-1.7
0
-0.3
0.2
0.3
0
fruit
juice
apple
A pairwise conceptor matrix
B computing abstraction heuristics
C graph optimization output

Figure 2.2: A: The matrix D represents the pairwise relations of abstraction between conceptors. B: Matrix D is populated using abstraction heuristics, where D_{ij} approximates how much more abstract conceptor C_i is with respect to conceptor C_j. C: The graph optimization process attempts to generate an ontology which is (1) faithful to the relations of abstraction identified previously, and (2) legible and sparse.

mulation also happens to provide hard cut-offs. C_1 can be determined to be more abstract than C_2, less abstract, or equally abstract. In practice, a continuous signal representing *how much* more abstract C_1 is compared to C_2 appears to be quite useful relative to the original ternary signal. In attempting to make use of an abstraction ordering signal which (1) is decently robust against the noise of real-world data, and (2) can be used to gauge the precise *magnitude* of the abstraction difference, we introduce a heuristic. The design of this heuristic has been informed by the fact that symmetric positive definite matrices only have positive eigenvalues. This property has led to the idea of mean-pooling eigenvalues and using both the polarity and magnitude of the result as a proxy for abstraction ordering of two given conceptors.

Concretely, we estimate the abstraction ordering of (C_1, C_2) by means of the following heuristic:

$$f(C_1, C_2) = \frac{1}{d_{BERT}} \sum_{i=1}^{d_{BERT}} \lambda_i(C_1 - C_2).$$

To unpack, we first substract one conceptor matrix from the other. Second, we compute the mean of the eigenvalues of the difference matrix, where λ_i denotes the *i'th* largest eigenvalue of said matrix. Intuitively, all eigenvalues of the difference matrix are positive if the first conceptor spatially engulfs the other, having higher spread than the second across all dimensions. This is illustrated in subfigure B of Figure 2.2. In the context of this project, across all $d_{BERT} = 768$ dimensions. Conversely, all such eigenvalues are negative if the first conceptor is completely contained by the second across all dimensions. Inevitably, however, the two conceptors will exhibit one such relation across *some* dimensions, while simultaneously exhibiting the opposite in other dimensions. Hence, we average the eigenvalues in an attempt to reach a ”consensus opinion” as to how the two conceptors are related to each other in terms of abstraction.

2.5 Graph Optimization

Given the pairwise estimates of abstraction ordering computed before, we conduct a graph optimization process. This is illustrated in subfigure C of Figure 2.2. All candidate graphs considered are directed ones, while nodes are identified with concepts, and arcs indicate relations of abstraction. We explore other types of relations in Section 4.

We attempt to solve the graph optimization task through the local search algorithm of simulated annealing (see Algorithm 2.1). Each candidate graph is represented through a Boolean adjacency matrix A. Specifically, A^k denotes the adjacency matrix of the candidate graph considered at step k of the graph optimization process. A^k_{ij} is a Boolean value indicating whether concept i links to concept j in the associated candidate graph of step k. As an initial candidate graph, the optimization process starts with a fully-disconnected graph, where no concepts are related to each other. This is represented through an adjacency matrix full of null values, $A^0 = 0$. Then, we randomly sample a new graph proposal by randomly mutating the current graph in one location – removing a previous arc or adding a new one. The acceptance probability is informed by a temperature schedule which linearly decreases from one to zero over the course of the optimization process, encouraging heavy exploration in the first epochs while using an increasingly conservative strategy towards the end.

The objective function which the search algorithm attempts to maximize is a linear combination of four different terms. Each term is a function of either or both (1) the adjacency matrix A^k which is identified with the state of the graph optimization process in step k, and (2) the matrix D containing pairwise estimates of abstract ordering. D_{ij} denotes the numerical estimate of abstraction between conceptor i and j. In other words,

$$D_{ij} = f(C_i, C_j).$$

We note that the particular way D_{ij} is related to D_{ji} is determined by the choice numerical heuristic employed for abstraction ordering. In our case (i.e. mean of eigenvalues of difference matrix), $D_{ij} = -D_{ji}$, yet this is not necessarily the case when opting for other heuristics, as explored in the discussion (e.g. positive-negative eigenvalues ratio). Besides the two matrices just described which influence the objective function through the four terms whose description follows, the objective function is also influenced by the four coefficients which are used to weigh the four terms.

The first term is a function of both A^k and D. It is equal to the mean of the abstraction ordering estimates represented in the candidate graph by means of arcs. From now on, we refer to this term as *expressed abstraction* (EA). In case of a fully-disconnected graph (i.e. one in which no arc exists in the graph at all) represented by $A^k = 0$, $EA(A^k, D) = 0$. In contrast, in case of a fully-connected graph (i.e. one in which there exists an arc between any two nodes) represented by $A^k = 1$, $EA(A^k, D) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n D_{ij}$. More generally,

$$EA(A^k, D) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} A^k_{ij} D_{ij}.$$

In such case, $min(D) \leq EA(A^k, D) \leq max(D)$. This is the only among the four terms of the linear combination which indicates functional-groundedness, as it reflects the proportion of the abstraction identified in high-dimensional space which gets represented in the output knowledge graph.

The second term is only a function of A^k. It is equal to the proportion of arcs contained by the candidate graph represented by adjacency matrix A^k, relative to the maximum number of possible arcs n^2:

$$AD(A^k) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} A^k_{ij}.$$

We refer to this term as *arc density* (AD). In case of a fully-disconnected graph with $A^k = 0$, $AD(A^k) = 0$. In case of a fully-connected graph with $A^k = 1$, $AD(A^k) = 1$. In all other cases, $0 < AD(A^k) < 1$.

The third term is also only a function of A^k. It is equal to the mean difference between a node's children count (i.e. number of nodes connected via outbound arcs) and a target children count set in advance:

$$CE(A^k) = \frac{1}{n} \sum_{i=1}^{n} \left| (\sum_{j=1}^{n} A^k_{ij}) - target\_children \right|.$$

We refer to this term as *children error* (CE). In case of a graph with adjacency matrix A^k in which every node only has children equal in count to $target\_children$, $CE(A^k) = 0$. For all other graphs, $CE(A^k) > 0$.

The fourth and final term of the objective function is extremely similar to the previous one, yet it addresses the distribution of parent counts, rather than children counts. Concretely, it is equal to the mean difference between a node's parent count (i.e. number of nodes connected via inbound arcs) and a target parent count set in advance:

$$PE(A^k) = \frac{1}{n} \sum_{j=1}^{n} \left| (\sum_{i=1}^{n} A^k_{ij}) - target\_parents \right|.$$

We refer to this term as *parent error* (PE). In case of a graph with adjacency matrix A^k in which every node only has parents equal in count to $target\_parents$, $PE(A^k) = 0$. For all other graphs, $PE(A^k) > 0$.

We occasionally refer to arc density, children error, and parent error collectively as *legibility terms*, because their sole role in the objective function of the graph optimization process is to directly influence the sparsity of the NSC output graph, as opposed to incentivizing the optimization process to adequately represent the relations of abstraction. The broader role of the legibility terms, as hinted at in the introduction, is to help generate sparse explanations which are cognitively ergonomic, or human-grounded, as phrased in the XAI literature [Madsen et al., 2021]. In contrast, expressed abstraction is the only term of the objective function which incentivizes the adequate representation of abstraction ordering.

Besides the four terms which are functions of either or both A^k and D, the objective function also contain four coefficients meant to influence the relative weight of each term. We denote these as α, β, γ, and δ in order for the four terms. Given the four terms, the two additional targets for child and parent count, and the four weighing coefficients included in the linear combination, the objective function can finally be defined as:

$$\begin{aligned}
&score(A^k, D, target\_children, target\_parents) = \\
&\alpha EA(A^k, D) \\
&- \beta AD(A^k) \\
&- \gamma CE(A^k, target\_children) \\
&- \delta PE(A^k, target\_parents).
\end{aligned}$$

The result of the graph search is the final output of NSC: a graph which indicates how the underlying DL model relates concepts by means of contextual embeddings. The final graph optimization step of NSC, which builds heavily on simulated annealing, has been summarized in pseudocode in Algorithm 2.1. Additionally, the whole NSC pipeline has been condensed in Algorithm 2.2 in order to offer an overview of the entire technique.

In the context of the present work, we have manually specified values for the four coefficients of the linear combination which comprises the graph opti-

Algorithm 2.1 Graph optimization in NSC

```
  Input: $D$
  Output: $A^{opt}$
$A^0 \Leftarrow 0$ (fully-disconnected graph)
for $k = 0$ to $epochs$ do
  $T \Leftarrow 1 - \frac{k}{epochs}$
  $A^{k+1} \Leftarrow neighbor(A^k)$ (one-arc change)
  if $P_{acceptance}(score(D, A^k), score(D, A^{k+1}), T) \leq random(0, 1)$ then
    $A^{k+1} \Leftarrow A^k$
  end if
end for
$A^{opt} \Leftarrow A^{epochs}$
```

Algorithm 2.2 Nested State Clouds

```
  Input: $symbols$
  Output: $A^{opt}$
$conceptors \Leftarrow \{\}$
for $s$ in $symbols$ do
  $a \Leftarrow contexts(s)$ (list of contexts for exemplars)
  $b \Leftarrow cloud(a)$ (list of contextual embeddings of exemplars)
  $c \Leftarrow conceptor(b)$ (conceptor matrix obtained from state cloud)
  $conceptors \Leftarrow conceptors \cup \{c\}$
end for
for $i, c_i$ in $conceptors$ do
  for $j, c_j$ in $conceptors$ do
    $D_{ij} \Leftarrow f(c_i, c_j)$
  end for
end for
$A^{opt} \Leftarrow graph\_optimization(D)$
```

mization objective: $\alpha = 1, \beta = 10^{-1}, \gamma = 10^{-1}$, and $\delta = 10^{-2}$. Besides those, we have also manually specified values for the two hyperparameters related to local graph structure, $target\_children = 3$ and $target\_parents = 1$, in order to nudge the optimization process towards hierarchical solutions. However, a more robust approach to specifying the values of those six hyperparameters would be to conduct a hyperparameter search. Concretely, one might resort to searching for appropriate values for those six hyperparameters – in addition to the number of graph optimization epochs, the temperature schedule, and the conceptor aperture – which successfully recover some presupposed relations of abstraction between the concepts being related by means of the resulting knowledge graph. We leave that for future work and expand on the possibility in the discussion.

3 Results

In this paper, we designed a new interpretability technique which can be used to extract knowledge graphs from state clouds of contextual embeddings. We have noticed that NSC is able to successfully recover commonsense relations of abstraction from raw text data (e.g. "apple" < "fruit", "orange juice" < "juice", see Figure 3.1). We explore challenges of scalability to WordNet-scale benchmarks in Section 4. Additionally, we have found that for a limited number of concepts to relate, the graph search is robust with respect to the starting state. Moreover, the legibility terms included in the linear combination which comprise the search objective (e.g. arc count) successfully nudge the search towards relatively sparse outputs.

Additionally, the graph search history profile exhibits proper foraging behavior, with fast increases in solution quality in the beginning, followed by a more conservative strategy which ends in marginal improvements towards the move to heavy exploitation (see Figure 3.2).

A graph optimization run with the same hyperparameter configuration and different targeted concepts resulted in a similar graph output (see Figure 3.3). Interestingly enough, while every single arc present in the output graph depicts a valid relation of abstraction (e.g. "food" > "vegetable", "vegetable" > "onion", etc.), the graph still has several shortcomings. The implicit structure of the concepts analysed would reflect the three vegetables mentioned (i.e. "carrot", "potato", and "onion") as children of the "vegetable" node. However, two of them (i.e. "carrot" and "potato") have been linked directly to the more abstract "food".

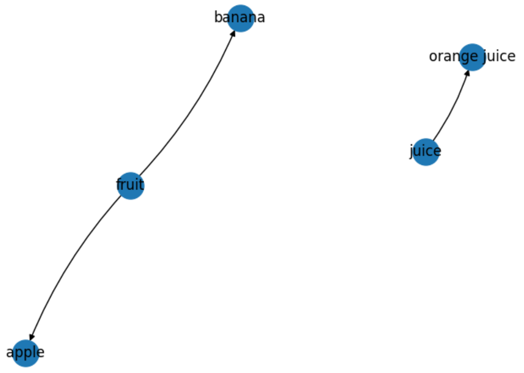

Figure 3.1: NSC output graph when applied to BERT using the shown symbols.

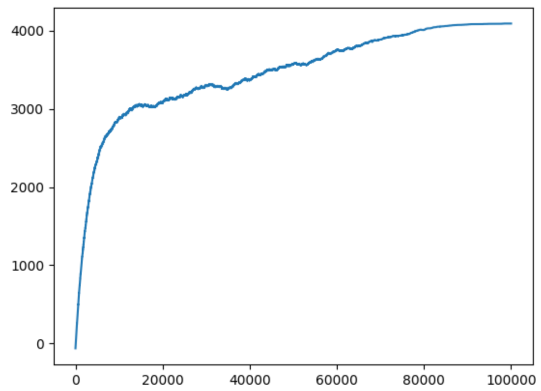

Figure 3.2: Candidate score by epoch during the graph optimization process.

One possible explanation of this outcome is a limitation of the objective function we employed. Namely, the four terms of the linear combination (i.e. expressed abstraction, arc density, children error, and parent error) have collectively proven insufficient for capturing and incentivizing this implicit structure. It appears more valuable for the graph optimization algorithm in terms of the objective function to populate the graph with arcs which denote particularly

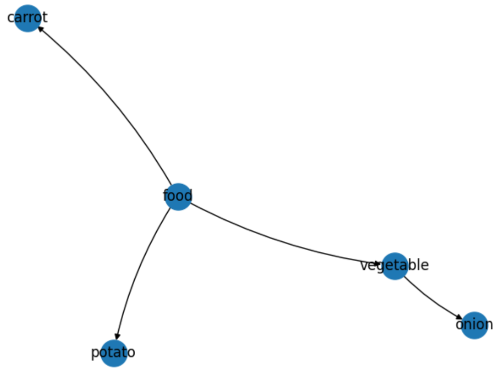

Figure 3.3: NSC output graph when applied to BERT using the shown symbols.

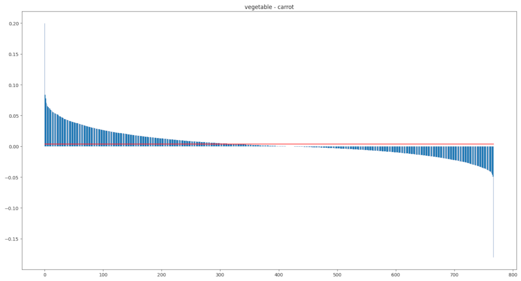

Figure 3.4: NSC output graph when applied to BERT using the shown symbols.

abrupt relations of abstraction. In this last graph output (see Figure 3.3), this is reflected in the favoring of the $(''food'', ''potato'')$ arc over the $\{(''food'', ''vegetable''), (''vegetable'', ''potato'')\}$ pair of arcs. Alternatively, arc density could be penalized less in order to render this arc pair more appealing in contrast to the single more abrupt arc.

Besides the final graph output generated by NSC, we can also investigate the intermediate step of estimating the relation of abstraction between two concepts. If we zoom in on the pairwise estimate of abstraction between "vegetable" and "carrot", we can observe a promising pattern. We consider the difference matrix $D = C_{vegetable} - C_{carrot}$, where the two C matrices correspond to the matrices of the conceptors obtained for the two concepts based on their associated exemplars. If we derive and plot the eigenvalues $\{\lambda_i(D) | i \in \{1, 2, ...d_{BERT}\}\}$ of the previous difference matrix, we notice that their mean (denoted by the horizontal red line in the Figure 3.4) is positive. However, we also notice that there is an important number of negative eigenvalues as well, hinting at the value of improving the noise robustness of NSC.

Additionally, we took note of the fact that the graph optimization process employed to output the final knowledge graph is highly sensitive to the precise choice of the objective function. Striking a balance between the legibility terms and the expressed abstraction is difficult to achieve manually. Often, one component of the linear combination tends to dominate the others. For instance, undervaluing the impact of the children error (i.e. the term which nudges the optimization process towards graphs whose nodes have a specific number of children) fails to penalize graphs which contain nodes with either numerous children or none at all (see Figure 3.5).

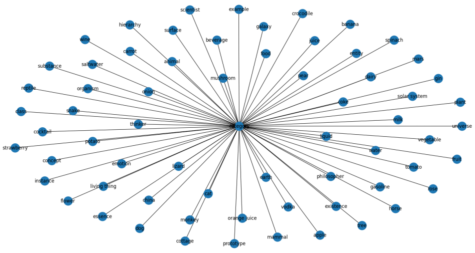

Figure 3.5: NSC output graph after undervaluing children count per node constraints.

A similar failure mode manifested itself when undervaluing the impact of arc density on graph optimization (i.e. the term which nudges the optimization process towards graphs which are sparse, in that they contain a relatively low number of arcs connecting nodes). In this other case, the optimization process favored graphs which contain many arcs (Figure 3.6). We speculate that this happened because considering candidate graphs which contain many arcs translates to an increase in expressed abstraction. In other words, simply including arcs which connect concepts which seemingly have a positive relation of abstraction (i.e. the arc's origin node is the slightest more abstract than the arc's target node) yield increases in terms of the objective function. Without a comparable arc density penalty, the optimization process approaches graphs overpopulated by arcs.

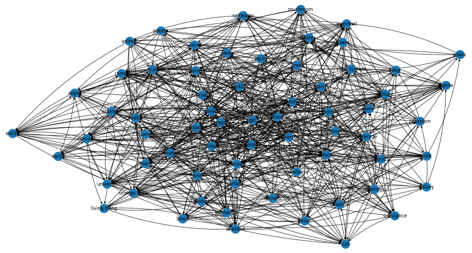

Figure 3.6: NSC output graph after undervaluing arc pruning.

This sensitive balance between the competing terms which make up the objective function echoes a broader concern which permeates XAI. There is a constant trade-off between functional-groundedness and human-groundedness which resembles the adequacy-fluency trade-off frequent in machine translation [Madsen et al., 2021]. In our case, the expressed abstraction term incentivizes functional-groundedness, the adequate communication of high-dimensional relations among concepts. In contrast, the three legibility terms employed in the objective function incentivize human-groundedness, the communication of DL model representations in a manner which is cognitively ergonomic for humans.

In our experiments, we find that striking this balance is especially difficult when trying to distill larger knowledge graphs containing more concepts from high-dimensional embeddings, bringing the scalability of NSC into question. That said, normalizing the objective's components relative to the total number of concepts being analyzed (i.e. all legibility terms assume values in $[0, 1]$ before weighting, regardless of node count) appears to improve scalability.

4 Discussion

4.1 Potential issues

4.1.1 The same symbols can represent different concepts.

Upon inspecting low-dimensional state cloud projections, we observed the presence of distinctive clusters across latent space (see Figure 4.1). For instance, the state cloud of the symbol ”plant” appears to be populated by at least three clusters. To investigate this, we ran a K-means clustering ($K = 3$) procedure on the ”plant” state cloud and surfaced the contexts which yielded contextual embeddings closest to the cluster centroids. Upon inspection of those contexts, we noticed that the context sets contained distinctive word senses, roughly corresponding to (1) vegetation, (2) the action of planting, and (3) factories (see Table 4.1).

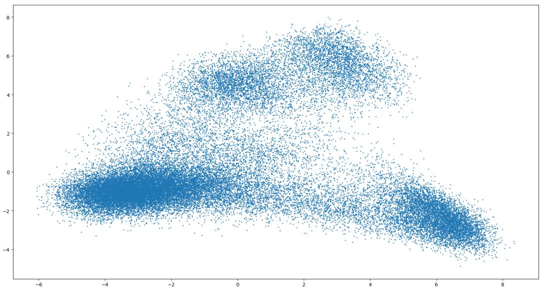

Figure 4.1: 2D PCA projection of the state cloud associated with the symbol ”plant”

This diversity of meanings assumed by the same symbol across the text corpus casts doubt on our assumption of there being a one-to-one correspondence between symbols and graph nodes. An intermediate clustering step might be effective in decoupling different word senses and producing different state clouds, though the issue of how many senses are there per symbol is non-trivial. Simi-

Table 4.1: Context samples by K-means cluster of ”plant” state cloud.

Cluster	Sample context
1	absently, i raised the blinds so that the plant was able to soak in the impromptu sunshine.
	i've brought you over a few macramé plant hangers to decorate your room.
2	i wanted to plant them myself.
	she'll just plant new ones and start all over again.
3	the computers running the plant were all infected, of course.
	it was plant shutdown for two weeks.

lar to how words themselves appear to discretely quantize the otherwise continuous semantic space, finite word senses as ”subsymbols” run into similar trade-offs between sparsity and accuracy. While we did try solving this polysemy challenge by K-means clustering with arbitrary values for K, the output graph quality degraded significantly, suggesting the difficulty in disentangling the different senses with this naive approach.

4.1.2 State clouds are non-linear.

While we employ conceptors as compact elliptical objects which approximate high-dimensional state clouds of contextual embeddings, their limited expressivity might fail to capture the intricate non-linear layout of real-world embeddings. Low-dimensional PCA projections of several state clouds of BERT embeddings radically diverge from Gaussian distributions, bringing into question the suitability of elliptical conceptors to represent them (see Figure 4.2). However, we note that state clouds of less ambiguous terms (i.e. limited number of word senses) appear more well-formed. Non-linearities might arise mainly from diverse word senses being assumed by the same symbols.

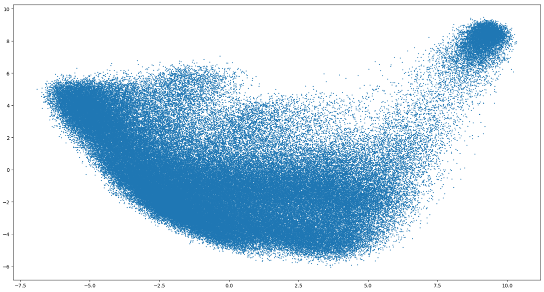

Figure 4.2: 2D PCA projection of the state cloud associated with the symbol ”earth”

4.1.3 NSC requires many exemplars.

The central role of state clouds in NSC means that the technique is highly dependent on a large number of occurences and contexts for each concept analyzed. This makes it difficult to interpret the model's internal representations with respect to obscure tokens, as those are extremely rare in natural datasets. However, synthetic datasets might address this issue, provided the ability to synthesize a wide range of unique contexts for an arbitrary concept [West et al., 2021].

4.1.4 NSC is influenced by the choice of auxiliary data.

NSC does not only require the model under investigation in order to work. There is also the requirement of an auxiliary dataset which contains a wide range of exemplars in different contexts. NSC then uses those exemplars in order to approximate the overarching state cloud with a conceptor object. However, the particular choice of dataset in which the concept instances are to be found has a high influence on the output knowledge graph.

For instance, consider two senses of the symbol ”property.” It might refer to (1) a characteristic, or (2) a real estate asset, among other senses. Different datasets might contain quantitatively distinct distributions of the meaning of ”property” across latent space. One predominantly containing text about real estate matters might be skewed towards considering sense (2) above as the most pervasive one. In contrast, a dataset containing text about the physical properties of certain materials might be skewed in the other direction, towards considering sense (1) above as the most typical one.

One might be tempted to simply aim for large datasets in order to alleviate this concern. Unfortunately, the issue as hand is not that certain senses are not represented enough through exemplars in order for their meaning to be captured. Rather, it is specifically an issue of representativity in the resulting population of meanings.

This has two implications, one concerning industry applications, and one concerning epistemics. The first is that, in practice, one would have to identify an auxiliary dataset which is representative enough of the concepts desired to be placed in the resulting knowledge graph. For instance, if one aims to investigate an DL model's internalized representations related to real estate matters, one concerning material science would be a poor choice of auxiliary dataset. The second implication hints at the fact that transparency tools like the ones used broadly in XAI can only meaningfully relate latent representations to the world models of people who authored the datasets. If one was to apply NSC on a book corpus, as we did, it would be unlikely for terms like ”entity,” ”object”, or ”thing” to be organized in an ontology in the same way a logician might organize them.

4.2 Future work

4.2.1 Improved noise robustness

Conceptors obtained from state clouds of contextual embeddings are challenged by several sources of noise. For one, the contextual embeddings themselves are noisy, as they have been generated by the pretrained BERT model, which in turn has been trained on a noisy real-world corpus comprised of books. In addition, the number of exemplars available for a given concept in the auxiliary corpus is implicitly limited, only depicting a partial representation of the underlying sampled distribution of meanings. Moreover, the closed-form step of obtaining conceptors from given state clouds is itself vulnerable to limited machine precision, questioning the suitability of the resulting conceptor object in representing the state cloud. The inversions of matrices containing low values and the eigenvalue extraction are particularly problematic stages in the conceptor learning process.

To this end, we argue that improved noise robustness would be a worthwhile future step to consider in improving the utility of NSC as an interpretability technique in practical applications, when dealing with other non-synthetic data sources. Noise robustness could be improved at many levels. For instance, generative methods for increasing the density of the state cloud by creating new exemplars might improve the accuracy with which the underlying sampled distribution is represented. Alternatively, methods for improving the numerical stability of obtaining conceptors from a given state cloud could be devised. Such techniques appear particularly relevant when employing relatively low aperture values for the learned conceptors, as they typically contain extremely low values as the conceptor matrix approaches the zero matrix.

4.2.2 Improved graph optimization scalability

The graph optimization process is the final step of NSC, and is responsible for actually generating the output knowledge graph given a host of constraints, including the *expressed abstraction* and *arc density*, among others. There are several hyperparameters which need to be specified in order to properly define the objective function used by the graph optimization procedure. For instance, there are the four coefficients to be found in the linear combination which defines the objective: α, β, γ, and δ. Besides, there are the two hyperparameters which help specify the constraints on local graph structure: $target\_children$ and $target\_parents$. Additionally, there are the few hyperparameters which help specify the underlying simulated annealing algorithm employed: the number of epochs and the temperature schedule used. Finally, the conceptor aperture needs to be specified in order to be able to obtain conceptors from state clouds.

In the context of the present work, all those values have been specified manually. Those have proven quite brittle, especially with larger numbers of concepts to relate to each other in the resulting knowledge graph. Beyond toy examples only containing a handful of concepts, it is extremely difficult to reach sensible outputs based on default hyperparameter values. Given this state of affairs, it would be useful to investigate systematic searches over hyperparameter space, so that suitable values for the graph optimization procedure can be automatically identified. The objective function of this meta-level search for hyperparameters of the graph optimization process can be informed by whether or not the process can successfully recover a few assumed relations of abstraction deemed true (e.g. "fruit" > "apple").

4.2.3 Beyond abstraction ordering

While this work focuses solely on the meronymous relationship of abstraction between to concepts, it is conceivable that the difference matrix computed in NSC as an intermediate step contains a rich representation of other relationships between concepts. This is reminiscent of the seemingly algebraic properties of prototype word embeddings (e.g. "king" + "woman" - "man" ≈ "queen"), which appear to encode diverse analogies and conceptual relationships. The relative spatial layout of entire state clouds might provide similar information.

However, mathematical objects depicting the aggregate shape of high-dimensional space clouds are likely to capture more information about the semantics of the symbols involved, compared to more rudimentary single prototype embeddings. Assuming the benefit of employing more expressive representation which stays faithful to large sets of exemplars at once, the difference matrices obtained from pairs of conceptor objects might also represent more nuance compared to the less expressive vector of displacement between prototype embeddings.

4.2.4 Beyond token embeddings

The present work focuses solely on extracting a knowledge graph from the state clouds comprised of contextual embeddings of text tokens (e.g. "juice") or short sequences of such tokens (e.g. "orange juice"). In section 1, however, we mention that NSC is modality-agnostic, in that it can theoretically be employed to distill knowledge graphs from embeddings of other modalities. One possible way of accomplishing that is by pooling together the data points associated with a certain class in an image classification setting. The class would then be identified with a concept based on its label (e.g. "fruit"), while its associated state cloud would not consist in a set of contextual embeddings, as in the case of text, but would be composed of a set of image embeddings generated for entire images (e.g. thousands of images of fruits). By learning conceptors from those state clouds, relating them using abstraction ordering heuristics, and searching for an appropriate knowledge graph structure, it is likely that meaningful meronymous structures would arise. Similar approaches could be employed for yet other modalities. For instance, audio classification datasets would give way to ontologies of audio concepts. Alternatively, datasets comprising video or 3D point clouds, together with labels unifying them in meaningful classes, might also lead to XAI tools in those other modalities.

4.2.5 Alternative abstraction heuristics

In the present work, we employ a rather specific heuristic for quantifying the relation of abstraction between two conceptors. For completeness, we mean-pool the eigenvalues of their difference matrix. Large positive values indicate the first conceptor is deemed to be more abstract than the second, while large negative values indicate the reverse. Values close to zero indicate a limited relation of abstraction between the two.

However, there might be other – better mathematically-motivated – ways of quantifying this relation. For instance, one might opt for different pooling mechanisms (e.g. median), which are more robust against outliers. Alternatively, one might eliminate the eigenvalue pooling mechanism entirely, and focus instead on the *ratio* between positive and negative eigenvalues.

4.3 Conclusion

In sum, the present work has been centered around the following contributions:

- We have provided qualitative evidence highlighting the connection between the spatial layout of nested state clouds and the abstraction relation of the concepts they represent.
- We have formulated an novel algorithm for flexibly distilling a set of high-dimensional state clouds into a compact directed graph which depicts relations of abstraction.
- We have drawn evidence-based observations on the way individual symbols relate to concepts in contextual embeddings.

References

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI, December 2019. URL `http://arxiv.org/abs/1910.10045`. Number: arXiv:1910.10045 arXiv:1910.10045 [cs].

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings, July 2016. URL http://arxiv.org/abs/1607.06520. Number: arXiv:1607.06520 arXiv:1607.06520 [cs, stat].

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. A Survey of the State of Explainable AI for Natural Language Processing. page 13, 2020.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de las Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, Seb Noury, Federico Pesamosca, David Pfau, Olivier Sauter, Cristian Sommariva, Stefano Coda, Basil Duval, Ambrogio Fasoli, Pushmeet Kohli, Koray Kavukcuoglu, Demis Hassabis, and Martin Riedmiller. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, February 2022. ISSN 1476-4687. 10.1038/s41586-021-04301-9. URL https://www.nature.com/articles/s41586-021-04301-9. Number: 7897 Publisher: Nature Publishing Group.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021. URL http://arxiv.org/abs/2010.11929. Number: arXiv:2010.11929 arXiv:2010.11929 [cs].

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention, October 2021. URL http://arxiv.org/abs/2006.03654. Number: arXiv:2006.03654 arXiv:2006.03654 [cs].

Geoffrey Hinton. How to represent part-whole hierarchies in a neural network, February 2021. URL http://arxiv.org/abs/2102.12627. Number: arXiv:2102.12627 arXiv:2102.12627 [cs].

Herbert Jaeger. Controlling Recurrent Neural Networks by Conceptors, April 2017. URL http://arxiv.org/abs/1403.3369. Number: arXiv:1403.3369 arXiv:1403.3369 [cs].

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *NAACL*, 2019.

Nanjiang Jiang and Marie-Catherine de Marneffe. Evaluating BERT for natural language inference: A case study on the CommitmentBank. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, January 2019. 10.18653/v1/D19-1630. URL https://par.nsf.gov/biblio/10158557-evaluating-bert-natural-language-inference-case-study-commitmentbank.

Philipp Koehn. Neural Machine Translation, September 2017. URL http://arxiv.org/abs/1709.07809. Number: arXiv:1709.07809 arXiv:1709.07809 [cs].

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, February 2020. URL http://arxiv.org/abs/1909.11942. Number: arXiv:1909.11942 arXiv:1909.11942 [cs].

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 1476-4687. 10.1038/nature14539. URL https://www.nature.com/articles/nature14539. Number: 7553 Publisher: Nature Publishing Group.

Ninghao Liu, Xiao Huang, Jundong Li, and Xia Hu. On Interpretation of Network Embedding via Taxonomy Induction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 1812–1820, New York, NY, USA, July 2018. Association for Computing Machinery. ISBN 978-1-4503-5552-0. 10.1145/

3219819.3220001. URL https://doi.org/10.1145/3219819.3220001.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, July 2019. URL http://arxiv.org/abs/1907.11692. Number: arXiv:1907.11692 arXiv:1907.11692 [cs].

Andreas Madsen, Siva Reddy, and Sarath Chandar. Post-hoc Interpretability for Neural NLP: A Survey. *arXiv:2108.04840 [cs]*, August 2021. URL http://arxiv.org/abs/2108.04840. arXiv: 2108.04840.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. page 12, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. Technical Report arXiv:2103.00020, arXiv, February 2021. URL http://arxiv.org/abs/2103.00020. arXiv:2103.00020 [cs] type: article.

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, Rachel Prudden, Amol Mandhane, Aidan Clark, Andrew Brock, Karen Simonyan, Raia Hadsell, Niall Robinson, Ellen Clancy, Alberto Arribas, and Shakir Mohamed. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, September 2021. ISSN 1476-4687. 10.1038/s41586-021-03854-z. URL https://www.nature.com/articles/s41586-021-03854-z. Number: 7878 Publisher: Nature Publishing Group.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Technical Report arXiv:1908.10084, arXiv, August 2019. URL http://arxiv.org/abs/1908.10084. arXiv:1908.10084 [cs] type: article.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, February 2020. URL http://arxiv.org/abs/1910.01108. Number: arXiv:1910.01108 arXiv:1910.01108 [cs].

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, December 2020. ISSN 1476-4687. 10.1038/s41586-020-03051-4. URL https://www.nature.com/articles/s41586-020-03051-4. Number: 7839 Publisher: Nature Publishing Group.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, 2019. Association for Computational Linguistics. 10.18653/v1/P19-1452. URL https://www.aclweb.org/anthology/P19-1452.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 113–123, Belgium, Brussels, 2018. Association for Computational Linguistics. 10.18653/v1/W18-6312. URL http://aclweb.org/anthology/W18-6312.

Chenguang Wang, Xiao Liu, and Dawn Song. Language Models are Open Knowledge Graphs. *arXiv:2010.11967 [cs]*, October 2020. URL http://arxiv.org/abs/2010.11967. arXiv: 2010.11967.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena D. Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. Symbolic Knowledge Distillation: from General Language Models to Commonsense Models, October 2021. URL http://arxiv.org/

abs/2110.07178. Number: arXiv:2110.07178 arXiv:2110.07178 [cs].

Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. Technical Report arXiv:1909.00161, arXiv, August 2019. URL http://arxiv.org/abs/1909.00161. arXiv:1909.00161 [cs] type: article.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, 2015.