Conceptualized and Contextualized Gaussian Embedding

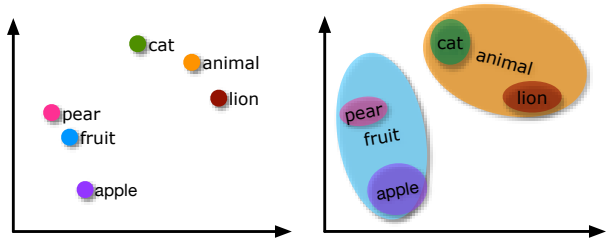
Chen Qian[1], **Fuli Feng**[2*], **Lijie Wen**[1*], **Tat-Seng Chua**[2]
[1] School of Software, Tsinghua University, Beijing, China
[2] Scholar of Computing, National University of Singapore, Singapore
qc16@mails.tsinghua.edu.cn, fulifeng93@gmail.com, wenlj@tsinghua.edu.cn, chuats@comp.nus.edu.sg

Abstract

Word embedding can represent a word as a *point vector* or a *Gaussian distribution* in high-dimensional spaces. Gaussian distribution is innately more expressive than point vector owing to the ability to additionally capture semantic uncertainties of words, and thus can express asymmetric relations among words more naturally (*e.g.,* `animal` entails `cat` but not the reverse). However, previous Gaussian embedders neglect inner-word conceptual knowledge and lack tailored Gaussian contextualizer, leading to inferior performance on both intrinsic (context-agnostic) and extrinsic (context-sensitive) tasks. In this paper, we first propose a novel Gaussian embedder which explicitly accounts for inner-word conceptual units (sememes) to represent word semantics more precisely; during learning, we propose *Gaussian Distribution Attention* over Gaussian representations to adaptively aggregate multiple sememe distributions into a word distribution, which guarantees the *Gaussian linear combination property*. Additionally, we propose a Gaussian contextualizer to utilize outer-word contexts in a sentence, producing contextualized Gaussian representations for context-sensitive tasks. Extensive experiments on intrinsic and extrinsic tasks demonstrate the effectiveness of the proposed approach, achieving state-of-the-art performance with near 5.00% relative improvement.

Introduction

Word embedding aims to learn low-dimensional word representations that encode semantic and syntactic information (Mikolov et al. 2013a). According to the form of word representations, word embedding evolves in two main directions: *point embedding* and *Gaussian embedding*. Point embedding (Figure 1(a)) represents each word as a deterministic *point vector* (Mikolov et al. 2013a) in a semantic space where the semantic similarity and other symmetric word relations can be effectively captured by the relative positions of points. However, it struggles to naturally model entailments among words (*e.g.,* `animal` entails `cat` but not the reverse) or other asymmetric relations. Asymmetries can reveal hierarchical structures among words (Athiwaratkun and Wilson 2018) and are crucial in knowledge representation and reasoning (Roller, Erk, and Boleda 2014). By contrast,



(a) Point embedding (b) Gaussian embedding

Figure 1: Point embedding vs. Gaussian embedding (a multi-dimensional Gaussian distribution is exemplified as a soft-region ellipse for better visualization). As for the latter, more specific words (*e.g.,* `cat` and `lion`) have smaller uncertainties, while those denoting broader concepts (*e.g.,* `animal`) have larger uncertainties.

Gaussian embedding (Figure 1(b)), on the other hand, represents each word as a "probabilistic" *Gaussian distribution*, which is innately more expressive owing to the ability to additionally capture semantic uncertainties of words (as their "geometric shapes") to represent words more naturally and more accurately than point vectors (Vilnis and McCallum 2015). For example, as Figure 1(b) shows, a word with larger uncertainty (e.g. `animal`) can semantically entail some words with smaller uncertainties (e.g. `cat`).

Nevertheless, recent advances are primarily focused on point embedding, making the line of Gaussian embedding lag very far behind. Concretely, sharing inner-word and outer-word linguistic information between words has shown remarkable success in point-embedding-based techniques (Bojanowski et al. 2017; Peters et al. 2018). Many point embedders consider inner-word information (*e.g.,* FASTTEXT) by explicitly encouraging words sharing similar inner-word structures (*e.g.,* surface subwords or conceptual units) to have "close" points (Bojanowski et al. 2017; Niu et al. 2017). Meanwhile, point-based contextualizers consider outer-word contexts (*e.g.,* ELMO) to yield dynamic representations for a word in different sentences, leading to dynamic representations and better word sense disambiguation (Peters et al. 2018; Devlin et al. 2019). By contrast, the line of Gaussian embedding: 1) represents word

*Fuli Feng and Lijie Wen are the co-corresponding authors.

semantics by utilizing surface subwords merely without considering the fruitful inner-word conceptual knowledge; and 2) produces merely context-agnostic Gaussian representations[1] due to the lack of tailored contextualizer. The two issues inevitably lead to inferior performance on both intrinsic (context-agnostic) and extrinsic (context-sensitive) tasks.

In this paper, we upgrade Gaussian embedding with considering both inner-word conceptual units (sememes[2]) and outer-word contexts to benefit both intrinsic context-agnostic and context-sensitive scenarios. Nevertheless, a key challenge is the lack of neural techniques specifically tailored for Gaussian embedding, such as *attention* and *contextualization* over Gaussian distributions, which limits the use of fruitful inner-word and outer-word information.

Aiming to tackle the challenge, we first propose *Gaussian Distribution Attention* (GDA) to dynamically aggregate inner-word sememe representations into word representations, which operates on Gaussian distributions directly and guarantees the *Gaussian linear combination property* (*i.e.,* the linear combination of mutually independent Gaussian variables is still a Gaussian distribution). Moreover, we adopt a training objective that incorporates the symmetric measure between a word and its synonym(s) and the asymmetric measure between the word and its hypernym(s) to explicitly capture the proper "positions" (*i.e.,* semantics) and "shapes" (*i.e.,* uncertainties) of words in a high-dimensional space. We will show that the two key considerations produce satisfying conceptualized Gaussian representations. Furthermore, after obtaining "static" conceptualized Gaussian representations, for a word in different sentences, we utilize outer-word contexts and introduce a dual contextualizer specially designed for Gaussian distributions which consists of two ELMO-style (Peters et al. 2018) context encoders and is supervised by the labeled signal of a specific downstream task (*e.g.,* text classification), aiming to produce contextualized Gaussian representations of words in varying sentences. To the best of our knowledge, we are the first attempt to explore Gaussian-embedding contextualization.

We conduct extensive experiments on two intrinsic tasks (*word similarity* and *word entailment*) and three types of extrinsic tasks (*single sentence tagging*, *single sentence classification* and *sentence pair classification*). The results show that our approach consistently outperforms state-of-the-art methods, which validates the effectiveness of the learned conceptualized and contextualized Gaussian representations. Moreover, integrating our Gaussian representations with advanced point-based contextualizer - BERT (Devlin et al. 2019) - achieves further improvement on these tasks, which shows the complementary information encoded by our Gaussian representations and proves that Gaussian embedding can also serve as an effective auxiliary for current point-based methods.

[1]In this paper, we use *Gaussian embedding* to denote a word embedding task, and *Gaussian representations* to denote the word representations produced by Gaussian embedding.

[2]The intuition of considering sememes rather than subwords is that morphologically similar words do not always relate with similar concepts (*e.g.,* `march` and `match`).

Related Work

Point embedding has been an active research area, including non-neural (Brown et al. 1992; Blitzer, McDonald, and Pereira 2006) and neural (Mikolov et al. 2013a,b; Pennington, Socher, and Manning 2014) methods. Recently, incorporating morphological subwords (Bojanowski et al. 2017; Chaudhary et al. 2018; Xu et al. 2018), syntactic structures (Vashishth et al. 2019; Kulmizev et al. 2019; Levy and Goldberg 2014; Li et al. 2017) or external knowledge (Wang et al. 2014; Yu and Dredze 2014; Liu et al. 2015, 2018; Niu et al. 2017; Alsuhaibani et al. 2018; Zhang et al. 2019) into point embedding shows significant improvements. Point-based contextualizers set out to produce context-sensitive word representations by integrating contextual information (Peters et al. 2018; Radford et al. 2018; Devlin et al. 2019).

Recognizing that the point-based world struggles to naturally model entailments among words (*e.g.,* `animal` entails `cat` but not the reverse) or other asymmetric relations, *Gaussian embedding* emerges to additionally capture uncertainties of words, which can better capture word semantics and express asymmetries more naturally (than dot product or cosine similarity in the point-based world). Vilnis and McCallum (2015) represented the semantic and uncertainty of each word with the mean and covariance of a Gaussian distribution. Inspired by multi-prototype embedding that learns multiple word representations for a word to better capture the semantics of words (Reisinger and Mooney 2010; Huang et al. 2012; Tian et al. 2014; Neelakantan et al. 2014), Gaussian mixture embedding (Chen et al. 2015; Athiwaratkun and Wilson 2017) was proposed to capture the meanings of polysemous words via multiple Gaussian distributions. In addition, Athiwaratkun and Wilson (2018) utilized Gaussian embedding to learn hierarchical encapsulation of words. One drawback of the above approaches was their inability to represent rare words. To remedy this, Athiwaratkun, Wilson, and Anandkumar (2018) (the most relevant to our work) represented a word by the sum of its surface subwords.

Nevertheless, the line of Gaussian embedding: 1) represents word semantics by utilizing surface subwords without considering the inner-word conceptual knowledge; and 2) produces merely context-agnostic Gaussian representations due to the lack of tailored contextualizer. The two issues inevitably lead to inferior performance on both intrinsic (context-agnostic) and extrinsic (context-sensitive) tasks. In this paper, we first propose a Gaussian embedder that represents each word by aggregating its conceptual units (sememes), which is more credible to capture intrinsic word semantics than aggregating surface subwords. We then propose a Gaussian contextualizer to produce contextualized Gaussian representations for a word in varying sentences for more potentially-benefited extrinsic tasks.

Gaussian Embedder

Gaussian embedder represents each word w in a pre-defined vocabulary $\mathcal{V}$ as a standard D-dimensional Gaussian distribution G_w:

$$G_w \sim \mathcal{N}(\mu_w, \Sigma_w) = \frac{e^{-\frac{1}{2}(x-\mu_w)^\top \Sigma_w^{-1}(x-\mu_w)}}{\sqrt{(2\pi)^D |\Sigma_w|}} \quad (1)$$

where the mean vector μ_w represents the semantics ("position") of w and the covariance matrix Σ_w represents the uncertainty ("geometric shape") of w (see Figure 1(b)). Gaussian embedder aims to learn the model parameters $\{(\mu_w, \Sigma_w)\}_{w \in \mathcal{V}}$ of words from a large-scale corpus.

Note that the most recent Gaussian embedder (Athiwaratkun, Wilson, and Anandkumar 2018) represents words merely based on morphological subwords and distributional hypothesis (Harris 1954), encouraging co-occurring word-pairs to have closer representations than randomly selected (negative) ones, which is too implicit to capture uncertainties of words. Instead, our training objective is to model both the symmetric relation and the asymmetric relation between words, *i.e.,* explicitly learning which words should be close to each other and which words should be "fatter" or "thinner". Meanwhile, the Gaussian representations are expected to be conceptualized, *i.e.,* perceiving inner-word conceptual sememes.

Model Training

Obviously, the pairwise relation between a word and its synonym(s)[3] is symmetric (*e.g.,* `apple` resembles `peach`, and vice versa) while asymmetric between the word and its hypernym(s) (*e.g.,* `fruit` entails `apple`, but not the reverse). Thus, the objective (loss function) of our Gaussian embedder is to jointly model symmetric word relations (*i.e.,* close positions and similar shapes) and asymmetric word relations (*i.e.,* close positions and "inclusive" shapes) via words' synonyms and hypernyms, respectively. Formally, we devise the following training objective:

$$\begin{aligned} &\mathcal{L}(w_c, w_s, w_h, w'_s, w'_h) = \max\big(0, m + \\ &\alpha \mathcal{L}_S(w_c, w_s, w'_s) + (1-\alpha)\mathcal{L}_A(w_c, w_h, w'_h)\big) \end{aligned} \quad (2)$$

where m serves as a margin; $\alpha \in (0, 1)$ is a trade-off parameter; $(w_c, w_s, w_h, w'_s, w'_h)$ is a training example where w_s and w_h are a positive synonym and a positive hypernym of w_c respectively; w'_s and w'_h are a negative synonym and a negative hypernym of w_c respectively; $\mathcal{L}_S(\cdot)$ aims to model the symmetric word relations with word-synonym triples (w_c-w_s-w'_s); $\mathcal{L}_A(\cdot)$ models asymmetric word relations with word-hypernym triples (w_c-w_h-w'_h); The training examples are generated by a word-quintuple sampler, which will be detailed in the following. The objective would push the likelihood of the positive example over the negative one by a margin m, *i.e.,* pushing the "positions" of a word and its positive synonym "closer" than its negative synonym, and meanwhile pushing the "shapes" of the word and its hypernym "more inclusive" than its negative counterpart.

[3]Based on distributional hypothesis (Harris 1954; Sahlgren 2008) that words occurring in nearby contexts tend to be semantically related. Here, we use "synonym" to denote a semantically-related word for brevity.

Following Vilnis and McCallum (2015), we employ a standard inner product to measure the symmetries between two Gaussian distributions:

$$\begin{aligned} &\mathcal{L}_S(w_c, w_s, w'_s) = \log \mathcal{S}(w_c, w'_s) - \log \mathcal{S}(w_c, w_s) \\ &\mathcal{S}(u, v) = \int \mathcal{N}(\mu_u, \Sigma_u)\mathcal{N}(\mu_v, \Sigma_v) dx \\ &= -\frac{1}{2}\log|\Sigma_u + \Sigma_v| - \frac{D}{2}\log(2\pi) \\ &\quad - \frac{1}{2}(\mu_u - \mu_v)^\top (\Sigma_u + \Sigma_v)^{-1}(\mu_u - \mu_v) \end{aligned} \quad (3)$$

Besides, we employ Kullback-Leibler (KL) divergence, which is widely used for representing asymmetries (Athiwaratkun and Wilson 2018):

$$\begin{aligned} &\mathcal{L}_A(w_c, w_h, w'_h) = \log \mathcal{A}(w_c, w'_h) - \log \mathcal{A}(w_c, w_h) \\ &-\mathcal{A}(u, v) = \int \mathcal{N}(\mu_u, \Sigma_u) \log \frac{\mathcal{N}(\mu_u, \Sigma_u)}{\mathcal{N}(\mu_v, \Sigma_v)} dx \\ &= \frac{1}{2}(\log \frac{|\Sigma_u|}{|\Sigma_v|} - D + tr(\Sigma_u^{-1}\Sigma_v) \\ &\quad + (\mu_u - \mu_v)^\top \Sigma_u^{-1}(\mu_u - \mu_v)) \end{aligned} \quad (4)$$

where $tr(M)$ denotes the trace of a matrix M. Note the leading negative sign since KL is a distance function and not a similarity. Equation 4 can effectively push a word to be encompassed by its hyponym (Athiwaratkun and Wilson 2018).

Sememe

Inspired by the success of considering conceptual knowledge in point embedding, our Gaussian embedder incorporates *sememe* which is the minimum conceptual (semantic) unit in linguistics (Bloomfield 1926). Linguistic experts constructed commonsense knowledge bases where words are composed of sememes. For instance, HowNet (Dong and Dong 2003) annotates a word with three-layer concept hierarchy (word-sense-sememe) and utilizes sememes to differentiate diverse senses of each word (Qi et al. 2018, 2019). As shown in Figure 2, the word `bank` is annotated with three main senses (`institution`, `land` and `facility`) and the sense `land` is annotated with two main sememes (`near`, and `waters`). In the remainder, we use the notation $\preceq$ to denote the (conceptually) subordinating relations. If a word w contains a sense s and s contains a sememe m, m is subordinated by w, denoted as $m \preceq w$.

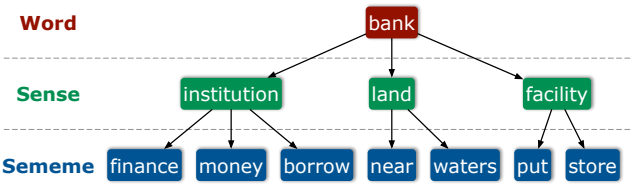


Figure 2: The concept hierarchy of the word `bank`.

Gaussian Distribution Attention

To better represent the semantics of a word by utilizing its concept hierarchy, we aggregate Gaussian representations of its sememes into its word representation. The intuition is to "pull" sememe-sharing words closer to each other, similar to feature-sharing mechanisms (Liu et al. 2019). To achieve that, inspired by the success of attention mechanism in NLP tasks, we propose *Gaussian Distribution Attention*[4] (GDA) to perform dynamic aggregation over sememe representations. The key difference between GDA and previous attention mechanisms designed for point vectors is that GDA operates over Gaussian distributions and guarantees the linear combination property of Gaussian distributions (Dwyer 1958) (*i.e.,* the linear combination of mutually independent Gaussian variables is still a Gaussian distribution):

$$\sum \beta_m \mathcal{N}(\mu_m, \Sigma_m) \sim \mathcal{N}(\sum \beta_m \mu_m, \sum \beta_m^2 \Sigma_m) \quad (5)$$

Note that all sememe Gaussian representations are independently initialized, aggregated and updated; any of them doesn't numerically influence another, *i.e.,* without numerical conditional dependence, which makes it possible to aggregate several sememe Gaussian representations into a word representation by utilizing the Gaussian property.

In the following, to differentiate words and sememes, we use $\mathcal{N}(\hat{\mu}_m, \hat{\Sigma}_m)$ to denote the Gaussian representations of a sememe m. Specifically, the formulation of GDA is:

$$GDA(\{G_m | m \preceq w\}) \sim \mathcal{N}(\beta_m \hat{\mu}_m, \beta_m^2 \hat{\Sigma}_m) \quad (6)$$

where β_m is the coefficient of sememe m indicating its importance towards word w, which is calculated as:

$$\begin{aligned} \beta_m &= \frac{\exp\big(\text{LeakyReLU}(e_m)\big)}{\sum_{m' \preceq w} \exp\big(\text{LeakyReLU}(e_{m'})\big)} \\ e_m &= \max_{l \preceq w_i \in S} \cos(W\hat{\mu}_m, W\hat{\mu}_l) \end{aligned} \quad (7)$$

where e_m is calculated according to the contextual sememes which are subordinated by contextual words of w in a sentence $S = \langle w_1, \cdots, w_i, \cdots, w_n \rangle$; $W \in \mathbb{R}^{D \times D}$ is the learnable parameter of GDA; LeakyReLU is a non-linear activation unit. Intuitively, the importance of sememe m to w will be high if similar sememe occurs in the context of w_c, which is a more reliable method than knowledge-free (*i.e.,* subword-aggregated) methods.

Word Sampling

We now describe how to sample the word quintuple $\{(w_c, w_s, w_h, w'_s, w'_h)\}$ used in the training objective. Most of the distributional hypothesis (Harris 1954; Sahlgren 2008) based approaches only select window-based contexts as "synonyms" (Mikolov et al. 2013a,b; Niu et al. 2017; Xu et al. 2018; Athiwaratkun, Wilson, and Anandkumar 2018). That is to say, these approaches select a word within and outside a fixed window centered by w_c as the positive (w_s) and negative (w'_s) synonyms respectively. However, as Vashishth et al. (2019) indicated, these approaches inevitably neglect some semantically relevant words lying beyond the window. To overcome this issue, based on the concept hierarchies of words, we sample additional positive synonyms which have common sememes with w_c although they lie beyond the window. Specifically, in a sentence, for a central word w_c, we sample a word co-occurring with w_c in a fixed window or a word w_s sharing common sememes with w_c ($\exists m, m \preceq w_c \wedge m \preceq w_s$) as a positive synonym of w_c.

Meanwhile, we observe that a sense of a word in HowNet usually refers to one of its hypernyms. As Figure 2 shows, a bank could refer to an institution, a land or a facility. Benefiting from this, for a central word w_c, we sample a sense w_h in its concept hierarchy as a positive hypernym of w_c.

Negative synonym (w'_s) and hypernym (w'_h) are sampled according to a distribution $P_n(w') \propto U(w')^{\frac{3}{4}}$, which is a distorted version of the unigram distribution $U(w')$ that also serves to diminish the relative importance of frequent words (Mikolov et al. 2013b).

Gaussian Contextualizer

Gaussian embedding produces merely context-agnostic Gaussian representations due to the lack of tailored contextualizer (for downstream tasks) (Vilnis and McCallum 2015; Athiwaratkun and Wilson 2017, 2018; Athiwaratkun, Wilson, and Anandkumar 2018). Hence, the representation of words would not be adapted according to their changing contexts. As such, applying these Gaussian representations typically leads to inferior performance especially on context-sensitive tasks such as named entity recognition and text classification. To this end, we make the first attempt to propose to produce contextualized Gaussian representations given a sentence by utilizing the outer-word contexts in varying sentences.

Considering that 1) Gaussian distributions could be parameterized by mean vectors (μ) and covariance matrices (Σ), 2) μ and Σ represent two different aspects: semantics and uncertainties, and 3) μ and Σ are on different scales and ranges where $\mu \in (-\infty, +\infty)$ but $\Sigma \in [0, +\infty)$, we thus equip the Gaussian contextualizer ($\mathcal{C}$) with two ELMO-style contextual encoders (Peters et al. 2018) (multi-layer BiLSTMs) to contextualize semantics and uncertainties respectively. As shown in Figure 3, given a sequence of words $S = \langle w_1, w_2, \cdots, w_n \rangle$, for each pretrained (conceptualized) Gaussian representation of w_i, we extract its mean vector and the covariance matrix: μ_i and ξ_i (the flatten Σ_i). We then pass the two sequential vectors ($\langle \mu_1, \mu_2, \cdots, \mu_n \rangle$ and $\langle \xi_1, \xi_2, \cdots, \xi_n \rangle$) through two ELMO encoders (position encoder $\mathcal{C}^\mu$ and shape encoder $\mathcal{C}^\xi$), and "assemble" the context-aggregated outputs as the mean and covariance of the contextualized Gaussian distribution of w_i:

$$\mathcal{C}(\mathcal{N}(\mu_i, \Sigma_i)) \sim \mathcal{N}\big(\mathcal{C}^\mu(\mu_i), \mathcal{C}^\xi(\xi_i)\big) \quad (8)$$

Moreover, inspired by Devlin et al. (2019), we can utilize a starting symbol `[CLS]` and a separating symbol

[4] Although similar names, the previous Gauss-style attention mechanisms (Guo, Zhang, and Liu 2019; Sah et al. 2017; Zhang, Winn, and Tomioka 2017) use Gaussian-distribution-normalized weights to score point vectors, while our proposed mechanism here learns weights to score Gaussian distributions.

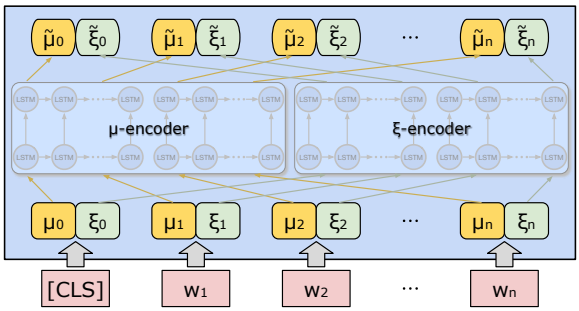


Figure 3: The architecture of our ELMO-style Gaussian contextualizer tailored for Gaussian embedding.

[SEP] to unambiguously represent a single sentence or a sentence pair. It it also worth mentioning that although the two parts are structurally independent, the parameters of the two ELMO architectures are jointly updated with the supervised signals of a downstream task, so that the contextualized Gaussian representations are dynamically and interactively adapted to the task.

Evaluation

In this section, we refer to the conceptualized Gaussian embedder as TIGER and the Gaussian contextualizer as GIANT for brevity. We validate whether our conceptualized Gaussian representations learned by TIGER can capture symmetric and asymmetric word relations well on intrinsic tasks, including *Word Similarity* and *Word Entailment*. We also validate whether the contextualized Gaussian representations generated by GIANT effectively capture the semantic and syntactic information on three types of extrinsic tasks: *single sentence tagging*, *single sentence classification* and *sentence pair classification*. Furthermore, we concatenate contextualized Gaussian representations with point vectors produced by BERT (Devlin et al. 2019) to explore whether our Gaussian embedding can augment point-based methods.

Baselines

We compare the proposed approach against three groups of representative word embedders:

- The first group of point embedders precisely captures semantic and syntactic relations of words based on local or global co-occurrence statistics in large-scale linguistic corpora. It includes a) WORD2VEC (SKIPGRAM) (Mikolov et al. 2013a) that captures the semantic similarity of co-occurring word-pairs in a local window; b) GLOVE (Pennington, Socher, and Manning 2014) that captures global linguistic information through the factorization of global word co-occurrence matrix; and c) WORD2SENSE (Panigrahi, Simhadri, and Bhattacharyya 2019) that represents words as sparse points where the magnitude of each coordinate represents the importance of the corresponding sense to the word.
- The second group of point embedders utilizes knowledge bases or syntactic dependencies to improve the quality of point vectors. It includes a) SEWRL (Niu et al. 2017) that utilizes inner-word conceptual sememes from HowNet to capture the semantics of words accurately; b) JOINTREPS (Alsuhaibani et al. 2018) that utilizes WordNet (Miller 1995) to augment semantic similarity prediction and entailment recognition; and c) SYNGCN (Vashishth et al. 2019) that utilizes dependency structures and graph convolutional networks to propagate syntactic information among words.
- The third group of Gaussian embedders includes a) W2G (Vilnis and McCallum 2015) that represents each word with a high-dimensional Gaussian distribution; b) W2GM (Athiwaratkun and Wilson 2017) that represents each word with multiple (*a.k.a.* multi-prototype) Gaussian representations to capture word meanings; and c) PFTGM (Athiwaratkun, Wilson, and Anandkumar 2018) that represents each word as a Gaussian mixture distribution where the mean vector of a mixture component is given by the sum of inner-word subwords.

Implementation Details

We train on a concatenation of two English datasets: UKWAC and Wackypedia (Baroni et al. 2009), which consists of 3.3 billion tokens. We discard words that occur fewer than 100 times in the corpora, which results in a vocabulary with 216,249 words. The fixed hyperparameters include an embedding dimension D=300, a margin m=1, the layer of BiLSTM in GIANT L=2 and a batch size of 128. We also experiment with a linearly decreasing weight α from 1.0 to 0.9 and Adagrad optimizer with a dynamic learning rate from 0.05 to 0.00001. Additionally, following Athiwaratkun and Wilson (2017), we use the diagonal covariances to reduce computation complexity from $O(D^3)$ to $O(D)$.

Evaluation on Intrinsic Tasks

Does our approach capture symmetric and asymmetric word relations better? We evaluate the learned symmetric word relations via *word similarity* (evaluating the closeness between two words) and the asymmetric word relations via *word entailment* (inferring whether a word is semantically inclusive in another).

For word similarity (SIM), we evaluate on multiple standard word similarity datasets: MC (Miller and Charles 1991), MEN (Bruni, Tran, and Baroni 2014), RG (Rubenstein and Goodenough 1965), RW (Luong, Socher, and Manning 2013), SL (Hill, Reichart, and Korhonen 2015), YP (Yang and Powers 2006) and SCWS Huang et al. (2012). Each dataset contains a list of word pairs with a real-valued score of their gold-standard similarity. Following Athiwaratkun and Wilson (2017), we use *cosine similarity* (COS) to calculate word similarity between point vectors or the mean vectors of Gaussian representations. We report the Spearman correlation (ρ) (Spearman 1904) between gold-standard scores and evaluated ones. For word entailment (ENT), we evaluate on the standard word entailment dataset (SED) (Baroni et al. 2012) which contains hyponym-hypernym pairs and gold-standard binary labels. Following Athiwaratkun and Wilson (2017), we use COS and *KL divergence* (KL) for entailment scoring, produce binary labels

Method	Dim.	Word Similarity (ρ)								Word Entailment (F_1)		
		MC	MEN	RG	RW	SL	YP	SCWS	AVG.	SED_{COS}	SED_{KL}	AVG.
WORD2VEC	300	63.96	70.27	70.01	25.43	29.39	39.34	59.57	51.14	63.79	54.62	59.21
GLOVE	300	70.20	73.75	76.95	33.55	37.05	56.13	55.42	57.58	70.47	67.65	69.06
WORD2SENSE	2,250	80.61	77.25	79.00	37.48	38.83	45.10	72.26	61.50	68.26	70.64	69.45
SEWRL	300	82.69	76.58	73.56	43.15	48.99	59.78	70.80	65.08	75.46	69.39	72.43
JOINTREPS	300	76.61	73.03	78.55	26.91	35.15	54.07	62.28	58.09	76.10	69.61	72.86
SYNGCN	300	82.40	65.65	63.07	40.51	44.75	57.68	63.02	59.58	75.02	66.14	70.58
W2G	300	82.42	78.40	80.34	35.49	38.84	46.40	66.20	61.16	76.49	76.02	76.26
W2GM	300	84.58	78.76	80.95	42.73	39.62	47.12	66.50	62.89	75.31	77.90	76.61
PFTGM	300	80.93	79.65	79.81	49.36	39.60	54.93	67.20	64.50	76.82	78.29	77.56
TIGER	300	**85.18**	**79.85**	**85.05**	**50.54**	**51.50**	**66.60**	**72.55**	**70.20**	**78.48**	**82.33**	**80.40**

Table 1: Comparison on the standard word similarity and entailment datasets. For each dataset, we boldface the score with the best performance and underline the score with the second-best performance across all methods.

(under a best score threshold) and measure the classification performance via macro-F_1 score. Note that COS only considers mean vectors while KL incorporates both means and covariances; thus, for KL, we associate point vectors with covariance matrices filled with tiny constant 10^{-6} to make them as "tiny balls" (*i.e.,* points).

The results are shown in Table 1, from which we have several key observations. 1) Our method TIGER outperforms all baselines on standard word similarity and entailment datasets, obtaining 5.70% and 2.84% relative performance improvement compared to the prior best-performing baseline (PFTGM) respectively. The results demonstrate that the learned conceptualized Gaussian representations are capable of capturing symmetric similarity and asymmetric entailment between words effectively. 2) Specifically, the improvement of TIGER over SEWRL which also considers the sememe HowNet-based hierarchies validates the effectiveness of employing Gaussian distributions to represent words. Moreover, TIGER outperforms PFTGM, a Gaussian embedder considering subwords instead of sememes, validating the effectiveness of utilizing concept hierarchies. 3) We can see that considering inner-word subwords (PFTGM) or sememes (SEWRL and TIGER), in comparison to the outer-word relation incorporated method (JOINTREPS), can effectively improve the ability to express word similarity; while JOINTREPS is relatively more beneficial on word entailment recognition. 4) Interestingly, as for the performance difference between two metrics in entailment recognition, we observe that considering uncertainties (KL) boosts the performance for the third group of Gaussian embedders. This shows that uncertainties of Gaussian embedding can effectively capture the "geometric shapes" of words to better express their asymmetric relation such as entailment, which is consistent with previous evidence of Athiwaratkun and Wilson (2017). This finding thus provides a promising pattern that considering both semantics and uncertainties to capture entailment relations in word-semantic studies.

Evaluation on Extrinsic Tasks

Can our Gaussian representations facilitate diverse downstream tasks with the help of the Gaussian contextualizer? We evaluate the contextualized Gaussian representations on 1) single-sentence-tagging tasks - *Part-of-Speech tagging* (POS) and *Named Entity Recognition* (NER) - with the CoNLL-2003 dataset (Sang and Meulder 2003); 2) single text classification task (STC) with the WeBis (Chen et al. 2019) dataset; and 3) sentence pair classification task - *Recognition Textual Entailment* (RTE) - with the RTE-5 (Bentivogli et al. 2009) dataset. For fair comparisons, all the baseline methods are also contextualized by the proposed GIANT. Moreover, we perform *ablation studies* by removing the contextualization operation ($\backslash$GIANT), removing the HowNet ($\backslash$HowNet; *i.e.,* without the sememe aggregation mechanism), removing the Gaussian distribution attention mechanism ($\backslash$GDA), removing sampling sememe-sharing words ($\backslash$SAM; *i.e.,* adopting traditional window-based sampling), and removing uncertainty information while retaining contextualized mean vectors of Gaussian representations ($\backslash\xi$) in inference. We follow the standard procedure to train a task-specific prediction layer (*i.e.,* MLP) on top of GIANT.

The results are summarized in Table 2, from which we have several key observations. 1) Overall, when provided with the same contextualizer (*i.e.,* GIANT), TIGER achieves the best performance on the four downstream tasks, which validates that conceptual knowledge and Gaussian embedding are also helpful for extrinsic tasks. 2) The phenomenon that considering outer-word context information (TIGER+GIANT) consistently boosts the performance on extrinsic (context-sensitive) tasks shows that incorporating context information produces dynamic word representations and is thus beneficial for sentence-level tasks. 3) We can also see that the variant, $\backslash$HowNet, hurts the performance much, indicating that conceptual knowledge incorporation is indeed effective for capturing word semantics via explicitly "exposing" sememes. The employed mechanisms, $\backslash$GDA and $\backslash$SAM, are also helpful. 4) Interestingly, we find that removing the uncertainty information ($\backslash\xi$) does not hurt the performance much on the first three tasks. We conjecture

Method	POS	NER	STC	RTE
WORD2VEC + GIANT	66.89	83.67	66.00	55.00
GLOVE + GIANT	77.49	88.15	75.50	55.50
WORD2SENSE + GIANT	73.14	83.16	74.00	55.00
SEWRL + GIANT	79.38	87.29	76.50	56.00
JOINTREPS + GIANT	70.07	83.67	76.50	57.00
SYNGCN + GIANT	79.66	83.57	76.50	57.50
W2G + GIANT	73.02	84.06	67.00	55.00
W2GM + GIANT	73.76	84.38	68.00	56.00
PFTGM + GIANT	74.03	85.74	69.50	56.50
TIGER + GIANT	**81.46**	**89.68**	**77.50**	**58.50**
\GIANT	67.20	80.85	56.85	45.50
\HowNet	76.88	85.93	71.84	56.98
\GDA	79.66	86.30	76.48	56.48
\SAM	79.89	88.81	75.88	56.77
$\backslash\xi$	81.40	88.36	**77.50**	56.50

Table 2: F_1-score (%) on four downstream tasks: POS, NER, STC and RTE. For each dataset, we boldface the score with the best performance and underline the score with the best performance across all baselines.

that the reason is that the performance on these tasks mainly depends on the semantics of words and their uncertainties are relatively unimportant.

Further Investigations

Does our contextualized Gaussian representations encode complementary information to advanced point-based contextualizers? Recent work has shown that the strong point-based contextualizer BERT (Devlin et al. 2019) performs well on diverse NLP tasks (Wang et al. 2019; Lin, Tan, and Frank 2019). Following Vashishth et al. (2019), we perform evaluation by concatenating the outputs of pretrained uncased $\text{BERT}_{\text{BASE}}$ and our contextualized Gaussian representations on the aforementioned six tasks.

The results are reported in Table 3. We can see that BERT behaves worse on two intrinsic tasks. This is somewhat surprising but consistent with previous findings of Meng et al. (2019). It is probably because BERT aims to learn context-aware representations, but the word similarity and entailment evaluation are conducted in a context-free manner. Thus, BERT is more like a contextualizer rather than a pure word embedder. Moreover, the results further show that our conceptualized and contextualized Gaussian representations indeed encode complementary information which is not captured by BERT, *i.e.,* the consideration of extra knowledge can improve the performance of BERT on both intrinsic and extrinsic tasks consistently. Hence, our Gaussian representations could serve as an effective combination with other point vectors by integrating Gaussian representations into current point-based systems.

Conclusion and Future Work

We proposed a conceptual-knowledge-based Gaussian embedder (TIGER) and a dual-ELMO Gaussian contextualizer (GIANT) to produce conceptualized and contextualized Gaussian representations. The extensive experiments evaluated on multiple intrinsic and extrinsic datasets validate the effectiveness of the learned conceptualized and contextualized Gaussian representations, consistently outperforming state-of-the-art methods by a margin.

Method + GIANT	SIM	ENT	POS	NER	STC	RTE
BERT	32.22	57.55	87.67	96.40	82.50	62.23
⊕WORD2SENSE	56.73	65.85	87.70	96.40	82.50	62.23
⊕SEWRL	57.04	67.87	88.86	96.50	82.55	62.25
⊕PFTGM	60.23	73.55	89.51	**96.95**	**83.00**	62.38
⊕TIGER	**63.36**	**76.96**	**90.20**	**96.95**	**83.00**	**62.46**

Table 3: The average results of concatenating BERT representations with GIANT-contextualized representations on the two intrinsic tasks and four extrinsic tasks.

Here, we list main conclusions/findings as follows: 1) Equipped with the ability to capture both "positions" and "shapes", TIGER can capture the word semantics more precisely, including symmetric word similarity and asymmetric word entailment. 2) GIANT can effectively produce contextualized Gaussian representations (first attempt in Gaussian-embedding studies) to facilitate different types of context-sensitive tasks. and 3) Our approach provides complementary information to BERT and thus can also serve as an effective auxiliary by integrating into current point-based models.

In the future, we are interested to explore more advanced encoders. For instance, contextualizing Gaussian representations with two BERT-style encoders may further improve performance. Besides, according to Dubossarsky, Grossman, and Weinshall (2018), random assignment of words to senses is shown to improve performance in the same task, we would assign a word to its specific sememe/sense. We hope this methodology can shed light on the scenarios where the uncertainty information and asymmetric relation are crucial, by embedding fruitful semantic information to reduce the stress of designing downstream algorithms.

Acknowledgments

We thank the three anonymous reviewers for their valuable suggestions. The work was supported by the National Key Research and Development Program of China (No. 2019YFB1704003), the National Nature Science Foundation of China (No.71690231), Tsinghua BNRist and NExT++ research supported by the National Research Foundation, Prime Minister's Office, Singapore under its IRC@SG Funding Initiative.

References

Alsuhaibani, M.; Bollegala, D.; Maehara, T.; Kenichi; and Kawarabayashi. 2018. Jointly Learning Word Embeddings using A Corpus and A Knowledge Base. In *PLoS ONE*.

Athiwaratkun, B.; and Wilson, A. G. 2017. Multimodal Word Distributions. In *ACL*, 1645–1656.

Athiwaratkun, B.; and Wilson, A. G. 2018. Hierarchical Density Order Embeddings. In *ICLR*.

Athiwaratkun, B.; Wilson, A. G.; and Anandkumar, A. 2018. Probabilistic FastText for Multi-Sense Word Embeddings. In *ACL*, 1–11.

Baroni, M.; Bernardi, R.; Do, N.-Q.; and chieh Shan, C. 2012. Entailment above the Word Level in Distributional Semantics. In *EACL*, 23–32.

Baroni, M.; Bernardini, S.; Ferraresi, A.; and Zanchetta, E. 2009. The Wacky Wide Web: A Collection of Very Large Linguistically Processed Web-crawled Corpora. In *Language Resources and Evaluation*, 209– 226.

Bentivogli, L.; Clark, P.; Dagan, I.; and Giampiccolo, D. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *TAC*.

Blitzer, J.; McDonald, R.; and Pereira, F. 2006. Domain Adaptation with Structural Correspondence Learning. In *EMNLP*, 120–128.

Bloomfield, L. 1926. A Set of Postulates for the Science of Language. In *Language*, 153–164.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching Word Vectors with Subword Information. In *TACL*, 135–146.

Brown, P. F.; Pietra, V. J. D.; deSouza, P. V.; Lai, J. C.; and Mercer, R. L. 1992. Class-Based n-gram Models of Natural Language. In *Computational Linguistics*, 467–480.

Bruni, E.; Tran, N. K.; and Baroni, M. 2014. Multimodal Distributional Semantics. In *Journal of Artificial Intelligence Research*, 1–47.

Chaudhary, A.; Zhou, C.; Levin, L.; Neubig, G.; Mortensen, D. R.; and Carbonell, J. G. 2018. Adapting Word Embeddings to New Languages with Morphological and Phonological Subword Representations. In *EMNLP*, 3285–3295.

Chen, X.; Qiu, X.; Jiang, J.; and Huang, X. 2015. Gaussian Mixture Embeddings for Multiple Word Prototypes. In *arXiv preprint arXiv:1511.06246*.

Chen, Z.; Shen, S.; Hu, Z.; Lu, X.; Mei, Q.; and Liu, X. 2019. Emoji-Powered Representation Learning for Cross-Lingual Sentiment Classification. In *WWW*, 251–262.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 4171–4186.

Dong, Z.; and Dong, Q. 2003. HowNet - A Hybrid Language and Knowledge Resource. In *NLP-KE*, 820–824.

Dubossarsky, H.; Grossman, E.; and Weinshall, D. 2018. Coming to Your Senses: on Controls and Evaluation Sets in Polysemy Research. In *EMNLP*, 1732–1740.

Dwyer, P. S. 1958. Generalizations of a Gaussian Theorem. In *the Annals of Mathematical Statistics*, 106–117.

Guo, M.; Zhang, Y.; and Liu, T. 2019. Gaussian Transformer: A Lightweight Approach for Natural Language Inference. In *AAAI*, 6489–6496.

Harris, Z. S. 1954. Distributional Structure. In *WORD*, 146–162.

Hill, F.; Reichart, R.; and Korhonen, A. 2015. SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. In *Computational Linguistics*, 665–695.

Huang, E.; Socher, R.; Manning, C.; and Ng, A. 2012. Improving Word Representations via Global Context and Multiple Word Prototypes. In *ACL*, 873–882.

Kulmizev, A.; de Lhoneux, M.; Gontrum, J.; Fano, E.; and Nivre, J. 2019. Deep Contextualized Word Embeddings in Transition-Based and Graph-Based Dependency Parsing - A Tale of Two Parsers Revisited. In *EMNLP*, 2755–2768.

Levy, O.; and Goldberg, Y. 2014. Dependency-Based Word Embeddings. In *ACL*, 302–308.

Li, B.; Liu, T.; Zhao, Z.; Tang, B.; Drozd, A.; Rogers, A.; and Du, X. 2017. Investigating Different Syntactic Context Types and Context Representations for Learning Word Embeddings. In *EMNLP*, 2421–2431.

Lin, Y.; Tan, Y. C.; and Frank, R. 2019. Open Sesame: Getting Inside BERT's Linguistic Knowledge. In *ACL*, 241–253.

Liu, Q.; Huang, H.; Zhang, G.; Gao, Y.; Xuan, J.; and Lu, J. 2018. Semantic Structure-Based Word Embedding by Incorporating Concept Convergence and Word Divergence. In *AAAI*.

Liu, Q.; Jiang, H.; Wei, S.; Ling, Z.-H.; and Hu, Y. 2015. Learning Semantic Word Embeddings based on Ordinal Knowledge Constraints. In *ACL*, 1501–1511.

Liu, X.; Wong, D. F.; Liu, Y.; Chao, L. S.; Xiao, T.; and Zhu, J. 2019. Shared-Private Bilingual Word Embeddings for Neural Machine Translation. In *ACL*, 3613–3622.

Luong, T.; Socher, R.; and Manning, C. 2013. Better Word Representations with Recursive Neural Networks for Morphology. In *CoNLL*, 104–113.

Meng, Y.; Huang, J.; Wang, G.; Zhang, C.; Zhuang, H.; Kaplan, L.; and Han, J. 2019. Spherical Text Embedding. In *NeurIPS*, 8208–8217.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient Estimation of Word Representations in Vector Space. In *arXiv preprint arXiv:1301.3781*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013b. Distributed Representations of Words and Phrases and their Compositionality. In *NeurIPS*, 3111–3119.

Miller, G. A. 1995. WordNet: A Lexical Database for English. In *Communications of the ACM*, 39–41.

Miller, G. A.; and Charles, W. G. 1991. Contextual Correlates of Semantic Similarity. In *Language and Cognitive Processes*, 1–28.

Neelakantan, A.; Shankar, J.; Passos, A.; and McCallum, A. 2014. Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space. In *EMNLP*, 1059–1069.

Niu, Y.; Xie, R.; Liu, Z.; and Sun, M. 2017. Improved Word Representation Learning with Sememes. In *ACL*, 2049–2058.

Panigrahi, A.; Simhadri, H. V.; and Bhattacharyya, C. 2019. Word2Sense: Sparse Interpretable Word Embeddings. In *ACL*, 5692–5705.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*, 1532–1543.

Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *NAACL*, 2227–2237.

Qi, F.; Lin, Y.; Sun, M.; Zhu, H.; Xie, R.; and Liu, Z. 2018. Cross-lingual Lexical Sememe Prediction. In *EMNLP*, 358–368.

Qi, F.; Yang, C.; Liu, Z.; Dong, Q.; Sun, M.; and Dong, Z. 2019. OpenHowNet: An Open Sememe-based Lexical Knowledge Base. In *arXiv preprint arXiv:1901.09957*.

Radford, A.; Narasimhan, K.; Salimans, T.; et al. 2018. Improving Language Understanding with Unsupervised Learning. In *Technical report, OpenAI*.

Reisinger, J.; and Mooney, R. J. 2010. Multi-Prototype Vector-Space Models of Word Meaning. In *NAACL*, 109–117.

Roller, S.; Erk, K.; and Boleda, G. 2014. Inclusive yet Selective: Supervised Distributional Hypernymy Detection. In *COLING*, 1025–1036.

Rubenstein, H.; and Goodenough, J. B. 1965. Contextual Correlates of Synonymy. In *Communications of the ACM*, 627–633.

Sah, S.; Nguyen, T.; Dominguez, M.; Such, F. P.; and Ptucha, R. 2017. Temporally Steered Gaussian Attention for Video Understanding. In *CVPR*, 33–41.

Sahlgren, M. 2008. The Distributional Hypothesis. In *Journal of Disability Studies*, 33–53.

Sang, E. F. T. K.; and Meulder, F. D. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *HLT-NAACL*, 142–147.

Spearman, C. 1904. The Proof and Measurement of Association between Two Things. In *the American Journal of Psychology*, 88–103.

Tian, F.; Dai, H.; Bian, J.; Gao, B.; Zhang, R.; Chen, E.; and Liu, T.-Y. 2014. A Probabilistic Model for Learning Multi-Prototype Word Embeddings. In *COLING*, 151–160.

Vashishth, S.; Bhandari, M.; Yadav, P.; Rai, P.; Bhattacharyya, C.; and Talukdar, P. 2019. Incorporating Syntactic and Semantic Information in Word Embeddings using Graph Convolutional Networks. In *ACL*, 3308–3318.

Vilnis, L.; and McCallum, A. 2015. Word Representations via Gaussian Embedding. In *ICLR*.

Wang, A.; Hula, J.; Xia, P.; Raghavendra Pappagari, R. T. M.; Patel, R.; Kim, N.; Tenney, I.; Yinghui Huang, K. Y.; Jin, S.; Durme, B. C. B. V.; Edouard Grave, E. P.; and Bowman, S. R. 2019. Can You Tell Me How to Get Past Sesame Street? Sentence-Level Pretraining Beyond Language Modeling. In *ACL*, 4465–4476.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge Graph and Text Jointly Embedding. In *EMNLP*, 1591–1601.

Xu, Y.; Liu, J.; Yang, W.; and Huang, L. 2018. Incorporating Latent Meanings of Morphological Compositions to Enhance Word Embeddings. In *ACL*, 1232–1242.

Yang, D.; and Powers, D. M. 2006. Verb Similarity on the Taxonomy of WordNet. In *the International WordNet Conference*, 121–128.

Yu, M.; and Dredze, M. 2014. Improving Lexical Embeddings with Semantic Knowledge. In *ACL*, 545–550.

Zhang, L.; Winn, J.; and Tomioka, R. 2017. Gaussian Attention Model and Its Application to Knowledge Base Embedding and Question Answering. In *ICLR*.

Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; and Liu, Q. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *ACL*, 1441–1451.