SENTECON: Leveraging Lexicons to Learn Human-Interpretable Language Representations

Victoria Lin
Carnegie Mellon University
vlin2@andrew.cmu.edu

Louis-Philippe Morency
Carnegie Mellon University
morency@cs.cmu.edu

Abstract

Although deep language representations have become the dominant form of language featurization in recent years, in many settings it is important to understand a model's decision-making process. This necessitates not only an interpretable model but also interpretable features. In particular, language must be featurized in a way that is interpretable while still characterizing the original text well. We present SENTECON, a method for introducing human interpretability in deep language representations. Given a passage of text, SENTECON encodes the text as a layer of interpretable categories in which each dimension corresponds to the relevance of a specific category. Our empirical evaluations indicate that encoding language with SENTECON provides high-level interpretability at little to no cost to predictive performance on downstream tasks. Moreover, we find that SENTECON outperforms existing interpretable language representations with respect to both its downstream performance and its agreement with human characterizations of the text.

1 Introduction

Deep language representations have become the dominant form of language featurization in recent years. These black-box representations perform excellently on a diverse array of tasks and are widely used in state-of-the-art machine learning pipelines. In many settings, however, it is important to understand a model's decision-making process, which necessitates not only an interpretable model but also interpretable features. To be useful, language must be featurized in a way that is interpretable while still characterizing the original text well. The fields of affective computing, computational social science, and computational psychology often use models to elucidate the relationships between patterns of language use and specific outcomes (Lin et al., 2020; Wörtwein et al., 2021). Moreover, interpretability is necessary to enforce desirable criteria like fairness (Du et al., 2021), robustness (Doshi-Velez and Kim, 2017), and causality (Veitch et al., 2020; Feder et al., 2022).

Despite advances in deep language representations, they are not considered human-interpretable due to their high dimensionality and the fact that their dimensions do not correspond to human-understandable concepts. Instead, researchers in need of interpretable language representations often turn to lexicons (Morales et al., 2017; Saha et al., 2019; Relia et al., 2019), which map words to meaningful categories or concepts. While useful in their simplicity, lexicons capture much less information about the text than do deep language representations. Most notably, because they parse text on the level of individual words, lexicons are unable to represent how those words are used within the broader context of the text, which can lead to misrepresentation of the text's meaning or intent. Consequently, lexicon-based language representations may not necessarily correspond well with how a human, who is able to comprehend the entire passage context, would perceive the text; and they may not perform well when used in downstream tasks.

With an eye toward addressing these concerns, we present SENTECON,[1] a method for introducing human interpretability in deep language representations. Given a sentence,[2] SENTECON encodes the text as a layer of interpretable categories in which each dimension corresponds to the relevance of a specific category (Figure 1). The output of SENTECON can itself therefore be viewed as an interpretable language representation. As language use can vary across text domains, we also present an extension, SENTECON+, that can adapt to specific domains via a *reference corpus*, a collection of unlabeled text passages from a target domain.

[1]Our code and data are publicly available at https://github.com/torylin/sentecon/.

[2]We use the term "sentence" for clarity, but our approach is also applicable to longer passages of text like paragraphs and documents, as our experiments show.

Findings of the Association for Computational Linguistics: ACL 2023, pages 4312–4331
July 9-14, 2023 ©2023 Association for Computational Linguistics

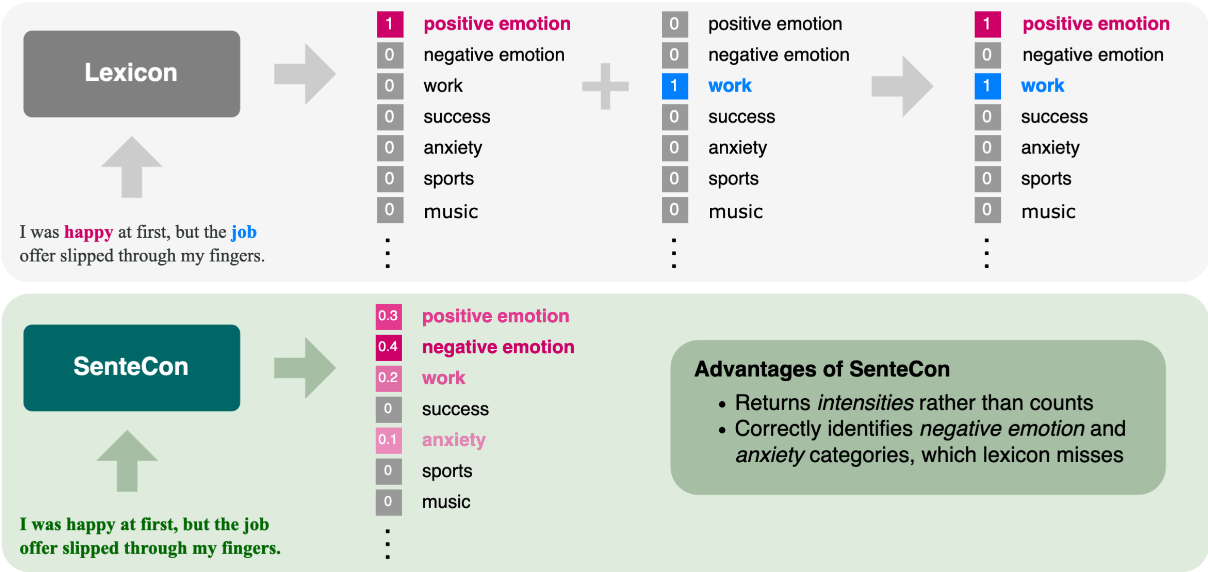

Figure 1: A comparison of lexicon-based language representations and SENTECON. While lexicons encode word-level category counts, SENTECON parses whole sentences and encodes sentence-level category intensities.

We evaluate SENTECON and SENTECON+ (jointly denoted hereafter as SENTECON(+)) with respect to both human interpretability and empirical performance. We first conduct an extensive human study that measures how well SENTECON(+) characterizes text compared to traditional lexicons. We complement this study with experiments using SENTECON(+) interpretable representations in downstream tasks, which allow us to compare its performance with that of existing interpretable and non-interpretable language representations. Finally, we analyze SENTECON(+) representations to determine whether they indeed are influenced by sentence context in a meaningful way.

2 Related Work

Lexicons. One of the primary existing interpretable language representations is the *lexicon*. A lexicon is a mapping of words to one or more categories (often linguistic or topical) that can be used to compute a score or weight for those categories from a passage of text. Popular lexicons include Linguistic Inquiry and Word Count (LIWC), a human-constructed lexicon for psychology and social interaction (Pennebaker et al., 2015); Empath, a general-purpose lexicon in which categories are generated automatically from a small set of seed words (Fast et al., 2016); and SentiWordNet, an automatically-generated lexicon for sentiment analysis and opinion mining (Baccianella et al., 2010).

Contextual lexicons. Contextual lexicons attempt to incorporate sentence context while retaining a lexicon structure. In one class of methods, adjustments are made to the lexicon via human-defined rules that depend on the context of the word being parsed. However, the reliance of these rule-based approaches on human intervention limits their wider use. For example, Muhammad et al. (2016) modify the sentiment score output of their lexicon based on the proximity of negation words and valence shifters, and Vargas et al. (2021) construct a lexicon that explicitly defines words that are context-independent (i.e., will retain their meaning regardless of context) and context-dependent.

Interpretable deep language models. A number of works provide some degree of interpretability to black-box language models via post-hoc analyses. Clark et al. (2019) analyze BERT's (Devlin et al., 2019a) attention heads and link them to attributes like delimiter tokens or positional offsets, while Bolukbasi et al. (2021) examine individual neurons within the BERT architecture that spuriously appear to encode a single interpretable concept. Górski et al. (2021) adapt the Grad-CAM visual explanation method (Selvaraju et al., 2017) for a CNN-based text processing pipeline. Although these analyses lend some insight, interpretability is limited to the low-level concepts associated with individual attention heads or neurons, and substantial manual probing is required for each network.

Methods. Several previous works contain elements of methodological similarity to SENTE-

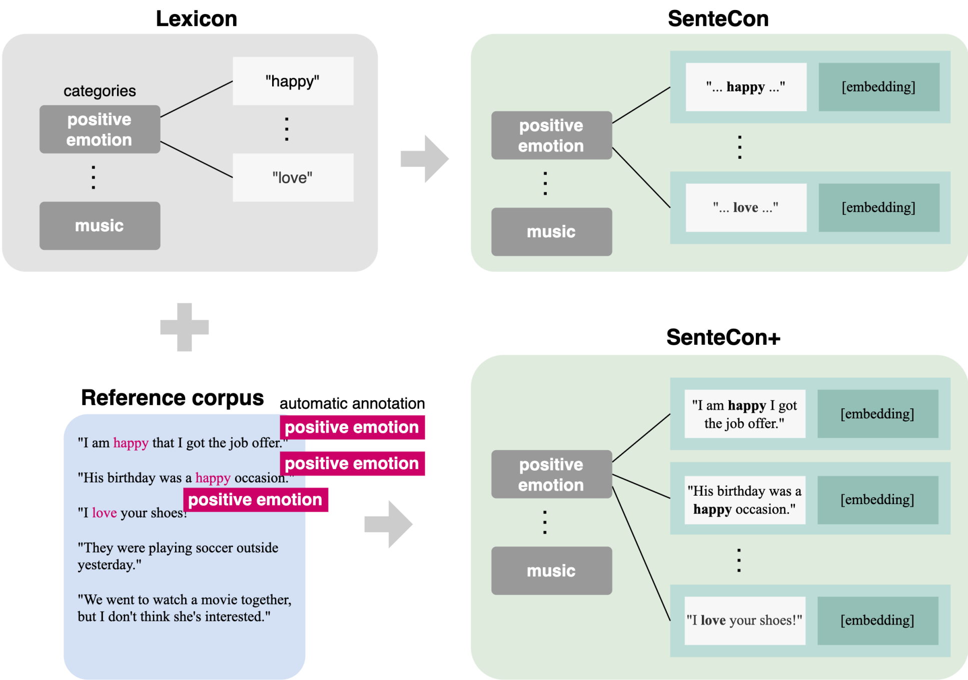

Figure 2: An illustration of the SENTECON and SENTECON+ methods. Starting with a traditional lexicon, it is possible to obtain either SENTECON (top row) or—using a reference corpus—SENTECON+ (bottom row).

CON(+) but differ in their aims. To address gaps in the LIWC lexicon vocabulary, Gong et al. (2018) implement a soft matching scheme based on non-contextual WordNet (Miller, 1995) and word2vec (Mikolov et al., 2013) embeddings. Given a new word, their method increases a category's weight if the embedding similarity between the new word and any word associated with the category is greater than some threshold. Onoe and Durrett (2020) propose a method for interpretable entity representations as a probability vector of entity types. They train text classifiers for each entity type, which is computationally expensive and requires large quantities of training data and labels. Modifying either the predicted entity types or the data domain involves retraining the classifiers.

3 SENTECON(+)

SENTECON(+) draws upon the notion of the lexicon; however, rather than mapping words to categories, SENTECON(+) maps the categories to the *deep embeddings of sentences that contain those words*. This, in effect, automatically generates dictionaries of sentence embeddings. To encode the categories of a new sentence, SENTECON(+) uses the similarity between the embedding of the new sentence and the embeddings of the sentences associated with each category.

Generally, SENTECON(+) can be thought of as two parts: (1) building a sentence embedding dictionary and (2) using that dictionary to generate an interpretable representation for a new sentence. We describe the details of the procedure in Sections 3.1 and 3.2. The full SENTECON(+) method is formally outlined in Algorithm 1.

3.1 Building a sentence embedding lexicon

We present two variants of our approach, SENTECON and SENTECON+, both of which are possible ways to build a sentence embedding dictionary. An illustration of the two variants can be found Figure 2. To begin, suppose we have a traditional lexicon L that maps words to categories.

SENTECON efficiently approximates sentences for each category using the deep embeddings of the words L associates with that category. Loosely speaking, a word embedding from a language model contains information from all sentences in the training corpus that use that word. As the state-of-the-art pre-trained language models are trained on vast corpora, a word embedding from a pre-trained (or pre-trained, then fine-tuned) language model will capture in some sense the "typical" sentence context for that word. The word embedding can thus be treated as representative of all sentences that use that word. Therefore, the embeddings for

Algorithm 1 SENTECON(+)

Initialize deep language model M_θ
Obtain all categories $C = \{c_i\}_{i=1}^{d}$ in chosen lexicon L
if SENTECON **then**
    Obtain all words $S_{c_i} = \{s_{j,c_i}\}_{j=1}^{m}$ that L maps to category c_i
else if SENTECON+ **then**
    Obtain all sentences $S_{c_i} = \{s_{j,c_i}\}_{j=1}^{m}$ containing words that L maps to category c_i
end if
$r_{s_{new}} = M_\theta(s_{new})$ ▷ Get embedding for new sentence s_{new}
for $i \in [d]$ **do**
    $\mathcal{R}_{c_i} = \{M_\theta(s_{j,c_i})\}_{j=1}^{m}$, where $\mathcal{R}_{c_i} = (r_{jk})_{1\leq j\leq m, 1\leq k\leq n}$ ▷ Get deep embeddings
    for $k \in [n]$ **do**
        $\mathbf{centroid}(c_i)_k = \frac{1}{m}\sum_{j=1}^{m} r_{jk}$ ▷ Get centroid of embeddings
    end for
    $\mathbf{h}(s_{new})_i = g(r_{s_{new}}, \mathbf{centroid}(c_i))$ ▷ Compute similarity g of new sentence and centroid
end for
return $\mathbf{h}(s_{new})$ ▷ Return representation of s_{new}

all words in a category form a compact representation of all sentences in the training corpus containing any words associated with that category.

SENTECON+ allows our interpretable language representation to further adapt to a particular data domain using only unlabeled text from that domain. Language patterns are not necessarily the same across different domains. Consequently, we can improve how well SENTECON representations characterize the text in different settings by altering the method by which we construct the sentence embedding dictionary. Specifically, we tailor SENTECON to the data using a *reference corpus* of unlabeled sentences from the domain of interest. Sentences from the reference corpus are mapped to a category if the sentence contains at least one word that the lexicon L associates with a category.

We use a deep language model M_θ to produce the embeddings for the words (for SENTECON) or sentences (for SENTECON+) $S_{c_i} = \{s_{j,c_i}\}_{j=1}^{m}$ associated with each category $c_i \in C$, where $C = \{c_i\}_{i=1}^{d}$. Sentence embeddings are computed via average pooling of token embeddings. This yields a $m \times n$ matrix of embeddings, $\mathcal{R}_{c_i} = \{M_\theta(s_{j,c_i})\}_{j=1}^{m}$, where m is the number of words or sentences associated with the category and n is the hidden size of M_θ.

3.2 Generating a SENTECON(+) representation

After obtaining deep embeddings for all SENTECON words or SENTECON+ sentences, we find the centroid of the embeddings for each category to obtain a compact and efficient representation of the category.[3] For a category c_i, the centroid is found by taking the column-wise mean of $\mathcal{R}_{c_i}$, resulting in a $1 \times n$ vector. That is, letting r_{jk} denote the element of $\mathcal{R}_{c_i}$ in row j, column k, we find the k-th element of the centroid as

$$\mathbf{centroid}(c_i)_k = \frac{1}{m}\sum_{j=1}^{m} r_{jk}$$

Given a new sentence s_{new}, generating a SENTECON(+) representation requires us to compute the similarity between the new sentence and each of the categories. This is done by first embedding the new sentence as $r_{s_{new}} = M_\theta(s_{new})$, then using a similarity function g to obtain a distance between $r_{s_{new}}$ and each category centroid $\mathbf{centroid}(c_i)$. Specifically, for each category c_i, $i \in [d]$, we compute the similarity as $g(r_{s_{new}}, \mathbf{centroid}(c_i))$ and assign this value as the weight for category c_i. That is, letting $\mathbf{h}(s_{new})$ be the SENTECON(+) representation of s_{new}, we have for all $i \in [d]$,

$$\mathbf{h}(s_{new})_i = g(r_{s_{new}}, \mathbf{centroid}(c_i))$$

4 Experimental Setup

To assess the utility of SENTECON and SENTECON+, we evaluate both methods to determine how well they characterize text in comparison to both

[3] If there are thematic or topical groupings of words or sentences within a single category, multiple centroids per category may be used. Therefore, the number of centroids per category can be viewed as a tunable hyperparameter. We elaborate further on this topic in the appendix (Section A.1).

existing lexicon-based methods and deep language models. When computing SENTECON(+) representations, we use MPNet (Song et al., 2020) as our deep language model M_θ and cosine similarity as our similarity metric g. Our experiments consist of both human evaluations of SENTECON(+) language representations and tests of performance when using them in downstream predictive tasks.

4.1 Lexicons

Linguistic Inquiry and Word Count (**LIWC**) is a human expert-constructed lexicon generally viewed as a gold standard for lexicons (Pennebaker et al., 2015). Its 2015 version has a vocabulary of 6,548 words that belong to one or more of its 85 categories, most of which are related to psychology and social interaction. We choose to exclude the 33 grammatical categories and retain the remaining 52 topical categories (list in appendix Section B.1).

Empath is a semi-automatically generated lexicon with a default vocabulary of 16,159 words that belong to one or more of its 194 categories (Fast et al., 2016). Empath defines a category using a small number of human-selected seed words, which are used to automatically discover related words that are then also associated with the category. Empath relates words using the cosine similarity of contextualized word embeddings from a deep skip-gram network trained for word prediction, and its categories are chosen from common dependency relationships in the ConceptNet (Liu and Singh, 2004) knowledge base.

4.2 Datasets for downstream tasks

In our performance experiments, we evaluate across several benchmark datasets: Stanford Sentiment Treebank (**SST**), a collection of polarized sentences from movie reviews (Socher et al., 2013); Multimodal EmotionLines Dataset (**MELD**), a multimodal dialogue dataset from the TV show *Friends* (Poria et al., 2019); Large Movie Review Dataset (**IMDb**), which comprises complete movie reviews from the website IMDb (Maas et al., 2011); and Multimodal Opinion-level Sentiment Intensity Corpus (**MOSI**), a set of opinion video clips from YouTube (Zadeh et al., 2016). These datasets were chosen to represent a range of data domains and scenarios in which lexicons like LIWC and Empath would typically be used, such as sentiment analysis, social interaction, and dialogue. Additional details are provided in the appendix (Section B.4).

For each of these datasets, we reserve a held-out set (without labels) to use as the SENTECON+ reference corpus. This allows us to adapt our SENTECON+ representation for the task domain.

4.3 Baseline representations and models

Our first evaluation is to compare interpretable representations of sentences with human judgements of those sentences (see Section 4.4). We have two primary baselines: **Lexicon** and **Lexicon+word2vec**. The **Lexicon** representation uses a bag-of-categories approach to encode the text using a traditional lexicon; in our experiments, we use LIWC and Empath, giving us the lexicon-specific baselines **Lexicon (L)** and **Lexicon (E)**, respectively. Bag-of-categories uses a lexicon to label each word in a text with one or more categories. From these categorized words, a vector of category counts can be constructed for a sentence.

The **Lexicon+word2vec** language representation implements the previously mentioned soft matching approach proposed by Gong et al. (2018). Although the authors describe the method for LIWC only, we generalize the method to Empath also, from which we obtain the baselines **LIWC+word2vec** and **Empath+word2vec**. We include this baseline to separate the effects of adding sentence context from the effects of soft matching. In our human evaluation, we focus on LIWC given its broad use in many research areas and use Lexicon (L) and LIWC+word2vec as baselines.

In our downstream prediction experiments, we include an additional baseline model based on recent transformer self-attention architectures, **MPNet** (Song et al., 2020), to show performance for a non-interpretable language representation. We chose MPNet over other transformer architectures due to its better performance; we report results using other language models in the appendix (Section A.2). Pre-trained and fine-tuned MPNet are also used as M_θ, the deep language model used to generate sentence embeddings for SENTECON(+).

Taking both LIWC and Empath as our traditional lexicons, we evaluate SENTECON and SENTECON+ against Lexicon, Lexicon+word2vec, and MPNet. For all language representations, we add a linear layer over the representation and train the linear layer on the downstream task to obtain our predictions. Details about the training procedures are provided in the appendix (Section B.5).

We note that we do not expect SENTECON(+)

to outperform non-interpretable transformer-based language models on predictive tasks. We instead view MPNet as a reasonable upper bound for the performance of interpretable approaches.

4.4 Methodology for human evaluation

As a fair and reliable way to compare SENTECON(+) to other lexicon-based language representations, we collected an extensive set of human sentence-level annotations for all 52 non-grammatical categories of LIWC. In total, 100 sentences randomly sampled from MELD were each annotated across 52 categories by 6 human raters, for a total of 31,200 annotations. These annotations are available as a public dataset on our GitHub repository.

The human annotation study was conducted on the online research platform Prolific.[4] To avoid annotator fatigue, the 52 categories were randomly split into 5 sets of roughly equal size, and each set was given its own annotation task. Sentences were annotated in batches of 20, and each annotation task had 6 independent annotators. During the study, each annotator was shown one sentence at a time, alongside one set of 8 to 10 LIWC categories. Annotators were then asked to rate on a scale from 0 to 2 the extent to which each of the categories is expressed. This yielded a human score (averaged over the 6 annotators) of the relevance of each category for each annotated sentence.

We assessed the reliability of our annotations using intraclass correlation coefficients (ICC). Generally speaking, ICC values above 0.50, 0.75, and 0.90 indicate moderate, good, and excellent inter-rater reliability, respectively (Koo and Li, 2016). We obtained an average ICC estimate of 0.686 with a 95% confidence interval of [0.606, 0.746], demonstrating moderate to good reliability.

Further details about this study and its results are provided in the appendix (Section B.3).

5 Results and Discussion

5.1 Human evaluation

Using the human annotations described in Section 4.4, we examine how well the different interpretable language representations reflect human perceptions of the text. Across all annotated sentences, we computed Pearson correlations between the human-annotator category scores and the category weights from each sentence representation

[4] https://www.prolific.co/

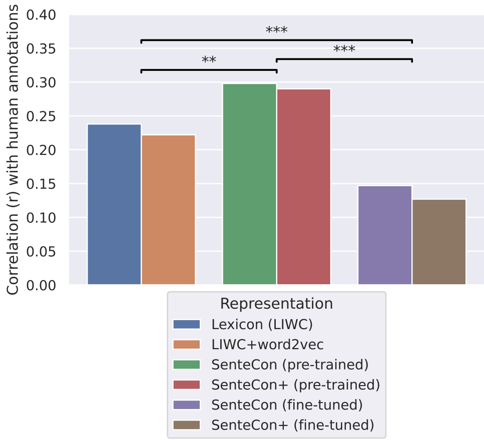

Figure 3: Average Pearson correlations (r) between human category annotations and interpretable language representations. ** denotes a difference with $p < 0.005$, and *** denotes a difference with $p < 0.0005$.

(Lexicon (L), LIWC+word2vec, SENTECON with pre-trained M_θ, SENTECON+ with pre-trained M_θ, SENTECON with M_θ fine-tuned on MELD, and SENTECON+ with M_θ fine-tuned on MELD). These results are shown in Figure 3. For illustrative purposes, we include correlations for 10 randomly selected sentences in Table 10 in the appendix.

We observe that when M_θ is pre-trained, SENTECON(+) correlates much more strongly with human category ratings than do either of the existing lexicon methods, Lexicon (L) and LIWC-word2vec. Using a paired two-sided t-test, we find that this difference is statistically significant. Importantly, these results suggest that **when used with a pre-trained M_θ, SENTECON and SENTECON+ better characterize the text than existing interpretable methods do**, since they are more consistent with human perceptions of the text.

Interestingly, when M_θ is fine-tuned on the target domain, SENTECON(+) correlates much *less* strongly with human category ratings than the existing lexicon methods do. This difference is also statistically significant. These results suggest that downstream performance gains from fine-tuning M_θ may come at a cost to interpretability.

We find no statistically significant difference between SENTECON and SENTECON+ given the same M_θ. That is, SENTECON (pre-trained) and SENTECON+ (pre-trained) have no statistically significant difference, nor do SENTECON (fine-tuned) and SENTECON+ (fine-tuned). We also find no statistically significant difference between Lexicon (L) and LIWC-word2vec.

Representation	Interpretable?	M_θ	MELD (e)	MELD (s)	SST	IMDb	MOSI
Majority / mean	-	-	48.1	48.1	49.9	50.0	-0.001
Lexicon (L)	Yes	-	46.5	49.5	67.8	76.7	0.202
LIWC+word2vec	Yes	-	47.5	49.4	78.7	81.4	0.270
SENTECON (L)	Yes	Pre-trained	47.7	57.6	86.5	84.2	**0.505**
SENTECON+ (L)	Yes	Pre-trained	**54.6**	**61.6**	**88.0**	**86.3**	0.487
Lexicon (E)	Yes	-	39.7	44.4	63.4	74.9	$\ll 0$
Empath+word2vec	Yes	-	46.0	50.8	81.4	85.1	0.222
SENTECON (E)	Yes	Pre-trained	51.5	59.2	88.7	87.0	0.450
SENTECON+ (E)	Yes	Pre-trained	**52.4**	**60.4**	**88.9**	**88.3**	**0.468**
Pre-trained MPNet	No	-	58.9	65.0	89.5	89.2	0.482

Table 1: Performance comparisons of SENTECON(+) and traditional lexicon-based methods when used in downstream prediction tasks. (L) indicates that LIWC was used as the base lexicon, while (E) indicates that Empath was used. The best result for each base lexicon choice is bolded. We report test accuracy for MELD (on both emotion and sentiment tasks), SST, and IMDb and test R^2 for MOSI.

Representation	Interpretable?	M_θ	MELD (e)	MELD (s)	SST	IMDb	MOSI
SENTECON (L)	Yes	Fine-tuned	57.2	68.1	**93.4**	**95.1**	0.672
SENTECON+ (L)	Yes	Fine-tuned	**59.9**	**68.1**	93.2	95.0	**0.673**
SENTECON (E)	Yes	Fine-tuned	56.3	67.3	93.2	94.9	**0.709**
SENTECON+ (E)	Yes	Fine-tuned	**59.3**	**68.5**	**93.3**	**95.0**	0.702
Fine-tuned MPNet	No	-	59.8	67.8	93.4	95.1	0.694

Table 2: Performance comparisons of SENTECON(+) and deep language representations when used in downstream prediction tasks. (L) indicates that LIWC was used as the base lexicon, while (E) indicates that Empath was used. The best result for each base lexicon choice is bolded. We report test evaluation metrics.

5.2 Performance on downstream tasks

We evaluate the implications of SENTECON(+) on downstream predictive performance. Our results, including comparisons with baseline models, are shown in Tables 1 and 2. Importantly, we find that:

(1) Both SENTECON and SENTECON+ perform better than the Lexicon and Lexicon+word2vec approaches do on downstream tasks (Table 1). This finding suggests that by modeling sentence-level context, SENTECON and SENTECON+ improve text characterization with respect to not only human evaluation but also downstream prediction. Across all classification tasks (MELD, SST, and IMDb), SENTECON and SENTECON+ achieve substantially higher accuracy than Lexicon and Lexicon+word2vec do, regardless of whether LIWC or Empath is used as the base lexicon. Likewise, SENTECON and SENTECON+ achieve substantially higher R^2 on the MOSI regression task than Lexicon and Lexicon+word2vec do.

(2) When used with a fine-tuned M_θ, SENTECON and SENTECON+ provide interpretability to deep language models at no cost to performance (Table 2). Across all downstream tasks, SENTECON(+) representations—particularly SENTECON+ representations—with fine-tuned M_θ achieve virtually equal performance compared to fine-tuned MPNet, the deep language model over which they are constructed. This observation holds for both choices of base lexicon L. We must emphasize the significance of this result: we are able to construct a layer of high-level interpretable concepts, pass it into a single linear layer (itself an interpretable model), and predict a target with equal performance as if we had used a non-interpretable deep language model fine-tuned on the task. In other words, *we can clearly understand the relationship between these interpretable concepts and the target without compromising performance*. This type of interpretability is far beyond that achieved by existing analyses of deep language models, and this type of performance is far beyond that achieved by existing lexicon-based methods.

(3) SENTECON+ offers performance improvements over SENTECON without negatively impacting interpretability (Tables 1 and 2), supporting the utility of using a reference corpus from the task data domain to refine SENTECON representa-

Word	Meaning 1	Meaning 2	Matching-sense similarity	Opposing-sense similarity	Individual similarity ratio
bright	*shining*	*intelligent*	**0.692**	0.608	1.139**
hard	*forceful*	*difficult*	**0.677**	0.539	1.256***
dull	*boring*	*unintelligent*	**0.686**	0.591	1.161***
dark	*dim*	*sinister*	**0.614**	0.488	1.258***
cool	*calm*	*impressive*	**0.419**	0.292	1.433***

Table 3: Similarities between contextualized SENTECON representations of homonyms and their matching- and opposing-sense meanings. ** denotes a difference with $p < 0.005$, and *** denotes a difference with $p < 0.0005$.

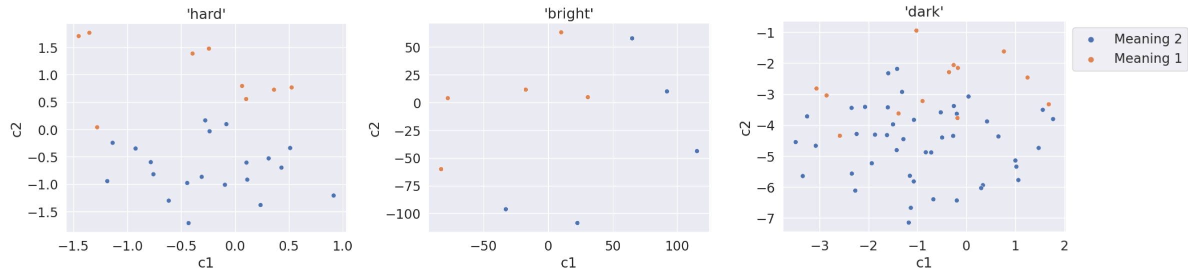

Figure 4: t-SNE plots of contextualized SENTECON representations of homonyms show separation by word sense.

tions. While fine-tuning M_θ allows SENTECON(+) to achieve the best performance, it does so at some cost to how well the representation agrees with human evaluations (Figure 3). When human agreement is a priority—e.g., in applications like healthcare and psychology—it may be more desirable to use SENTECON+ with a pre-trained M_θ instead. This configuration confers performance gains over SENTECON without compromising human agreement. Furthermore, even when M_θ is fine-tuned, SENTECON+ still often outperforms SENTECON, particularly when Empath is the base lexicon L.

5.3 Model analysis: Word sense

Given these results, we would like to gain some understanding of how SENTECON(+) is able to improve on existing lexicon-based interpretable language representations.

Prior work on BERT has demonstrated that its strength as a language representation lies partially in its ability to distinguish different word senses based on sentence context (Reif et al., 2019; Wiedemann et al., 2019; Schmidt and Hofmann, 2020). We postulate that sentence context similarly enables SENTECON(+) to distinguish different word senses, yielding the observed empirical gains in interpretability and performance. To explore this hypothesis, we conduct an experiment to verify whether a word's sentence context changes its SENTECON representation to be more similar to its true meaning in the sentence.

5.3.1 Method

Collecting homonyms. We first selected words with multiple common meanings (homonyms)—for example, the word *bright*. We began with a list of homonyms compiled from online sources.[5,6] For each homonym on the list, we collected all sentences in MELD and SST containing the word. We chose the dataset with more sentences containing the word, and we retained all homonyms for which there were 10 or more associated sentences. We annotated each sentence with the word's corresponding meaning (e.g., we labeled the sentences as using *bright* either to mean *shining* or to mean *intelligent*). For every sentence, this yields a "matching-sense" meaning and an "opposing-sense" meaning. We retained all homonyms for which each meaning of the word had 5 or more associated sentences.

Distinguishing word sense. With this set of homonyms, we verified whether SENTECON is capable of distinguishing word sense using a procedure similar to one in Reif et al. (2019) for BERT representations. For each sentence, we obtained the contextualized SENTECON representation for the selected homonym. We also obtained the non-contextualized SENTECON representations of three keywords for each meaning of the word (e.g., for

[5] https://7esl.com/homonyms/

[6] https://examples.yourdictionary.com/examples-of-homonyms.html

bright, these keywords are (1) *shining*, *vivid*, *beaming* and (2) *intelligent*, *smart*, *clever*). These keywords were randomly selected from the Oxford English Dictionary synonyms for each meaning of the word. Then—again for each sentence—we computed the cosine similarity between the SENTECON representations of the homonym and its matching-sense keywords, then the similarity between the SENTECON representations of the homonym and its opposing-sense keywords.

5.3.2 Results

The results of this experiment, which we report in Table 3, indicate that SENTECON representations are indeed able to distinguish different word senses. When used in a particular sentence context, words with multiple meanings show significantly more similarity to their matching-sense definition than they do to their opposing-sense definition. We formalize this with the *individual similarity ratio* metric defined by Reif et al. (2019), which is the ratio of matching-sense similarity to opposing-sense similarity. If a representation is able to correctly distinguish word sense, this ratio should be greater than 1, which we observe to be the case across all selected homonyms. Additionally, t-tests indicate that the difference in similarity is statistically significant across all homonyms.

We further visualize the separation of word senses via t-SNE plots of our SENTECON representations, similar to experiments by Wiedemann et al. (2019) on BERT embeddings. These plots show that SENTECON representations of the same word separate clearly in embedding space according to their meanings (Figure 4).

These results support our claim that SENTECON(+) uses sentence context to improve interpretability and performance on downstream tasks. The ability to distinguish word senses helps SENTECON(+) to correctly identify relevant categories where traditional lexicons may be not be able to do so, thereby allowing SENTECON(+) to better characterize the text.

6 Conclusion

In this paper we introduced SENTECON, a human-interpretable language representation that captures sentence context while retaining the benefits of interpretable lexicons. We conducted human evaluations to determine the agreement between SENTECON representations and the actual content of the text, and we ran a series of experiments using SENTECON in downstream predictive tasks. In doing so, we demonstrated that SENTECON and its extension, SENTECON+, better represent the character and content of the text than traditional lexicons do. Furthermore, we showed that when used in conjunction with language models fine-tuned on the downstream task, SENTECON and SENTECON+ provide interpretability to deep language models without any loss of performance. These findings render SENTECON and SENTECON+ compelling candidates for problems in fields like medicine, social science, and psychology, where understanding language use is an important part of the scientific process and where insight into a model's decision-making process can be paramount.

7 Acknowledgements

This material is based upon work partially supported by National Science Foundation awards 1722822 and 1750439, and National Institutes of Health awards R01MH125740, R01MH132225, R01MH096951 and R21MH130767. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors, and no official endorsement should be inferred.

8 Limitations

We recognize that several limitations remain with SENTECON and SENTECON+.

(1) Despite the gains in performance obtained by using a fine-tuned M_θ with SENTECON, we note that this version of SENTECON has significantly worse agreement with human evaluation than when a pre-trained M_θ is used. It is not immediately obvious why this should be the case. Although it is always possible to use SENTECON+ with a pre-trained M_θ in cases where agreement with human evaluation is particularly important, future work should examine why this degradation occurs and explore whether it is possible to maintain human agreement while also seeing those same performance gains (possibly through a secondary loss term that prioritizes human agreement).

(2) When building a sentence embedding dictionary, the base lexicon of SENTECON(+) may map lexically similar sentences to the same categories, regardless of attributes like negation. Despite this, SENTECON produces meaningful repre-

sentations for sentences that require compositional understanding, which we attribute to the large number of sentences mapped to each category (recall that each contextualized word embedding mapped to a category can be viewed as a summary of all sentences in the language model pre-training corpus containing that word). For example, the number of negated sentences in the sentence embedding dictionary is far smaller than the number of non-negated sentences—and likewise for other attributes requiring compositional parsing. Consequently, each category's centroid is still approximately an average of the non-negated sentences.

The same principle applies to SENTECON+ if a reasonably-sized reference corpus is used. If, however, only a very small reference corpus is available and the task dataset is known to require strong compositional understanding, SENTECON should be used instead of SENTECON+.

9 Ethics Statement

Broader impact. As deep language models gain greater prominence in both research and real-world use cases, concerns have arisen regarding their opaque nature (Rudin, 2019; Barredo Arrieta et al., 2020), their tendency to perpetuate and even amplify social biases in the data on which they are trained (Bolukbasi et al., 2016; Swinger et al., 2019; Caliskan et al., 2017), and their encoding of spurious relationships between the target and irrelevant parts of the input (Veitch et al., 2021). Particularly given their increasing deployment in healthcare, psychology, and social science, as we mention earlier in this paper, it is crucial that these black-box models be rendered more transparent to ensure that decisions are being made in a principled way. In other words, interpretability is not only an intellectual goal but also an ethical one.

In service of this goal, our proposed language representation, SENTECON, provides clear insight into the relationship between human-interpretable concepts and outcomes of interest in machine learning tasks. It is able to do so without negatively impacting predictive performance—an important factor, since a primary motivator for using non-interpretable language representations is their excellent performance on machine learning tasks. We hope that this will motivate others to use SENTECON, and we also hope that using SENTECON will allow users to better understand how their machine learning pipelines make decisions, evaluate their models for bias, and enforce correct and robust relationships between inputs and outputs.

Ethical considerations. This work involves the collection of new data to assess the consistency of SENTECON(+) representations with human annotations of the content of text passages. No information was collected about the annotators, and the data is not sensitive in nature. In the course of data collection, we took measures to ensure fair compensation and treatment of annotators. Annotators were provided a description of the study and given the option to decline the study after learning its details, and all annotators were paid at a rate above the local minimum wage.

SENTECON(+) relies on pre-trained deep language models to compute language representations. Our use of these pre-trained models is limited to research purposes only and is compliant with their intended use. We acknowledge that the use of pre-trained models introduces the possibility that SENTECON(+) may encode some biases contained in those models. As a consequence, interpretations of the relationships between SENTECON(+) categories and targets (when using SENTECON(+) in modeling) may also contain elements of bias.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Viégas, and Martin Wattenberg. 2021. An interpretability illusion for bert.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Franck Dernoncourt and Ji Young Lee. 2017. PubMed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 308–313, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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

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

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning.

Mengnan Du, Fan Yang, Na Zou, and Xia Hu. 2021. Fairness in deep learning: A computational perspective. *IEEE Intelligent Systems*, 36(4):25–34.

Ethan Fast, Binbin Chen, and Michael S. Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 4647–4657, New York, NY, USA. Association for Computing Machinery.

Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. 2022. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *Transactions of the Association for Computational Linguistics*, 10:1138–1158.

Jeffrey M. Girard. 2020. agreement: An R package for the tidy analysis of agreement and reliability.

Yuan Gong, Kevin Shin, and Christian Poellabauer. 2018. Improving liwc using soft word matching. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '18, page 523, New York, NY, USA. Association for Computing Machinery.

Łukasz Górski, Shashishekar Ramakrishna, and Jedrzej M. Nowosielski. 2021. Towards grad-cam based explainability in a legal text processing pipeline. extended version. In *AI Approaches to the Complexity of Legal Systems XI-XII*, pages 154–168, Cham. Springer International Publishing.

E. Holliman, J. Godfrey, and J. McDaniel. 1992. Switchboard: telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, volume 1, pages 517–520, Los Alamitos, CA, USA. IEEE Computer Society.

Terry K. Koo and Mae Y. Li. 2016. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of Chiropractic Medicine*, 15(2):155–163.

Victoria Lin, Jeffrey M. Girard, Michael A. Sayette, and Louis-Philippe Morency. 2020. Toward multimodal modeling of emotional expressiveness. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, page 548–557, New York, NY, USA. Association for Computing Machinery.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Michelle Morales, Stefan Scherer, and Rivka Levitan. 2017. A cross-modal review of indicators for depression detection systems. In *Proceedings of the Fourth Workshop on Computational Linguistics and Clinical Psychology — From Linguistic Signal to Clinical Reality*, pages 1–12, Vancouver, BC. Association for Computational Linguistics.

Aminu Muhammad, Nirmalie Wiratunga, and Robert Lothian. 2016. Contextual sentiment analysis for social media genres. *Knowledge-Based Systems*, 108:92–101. New Avenues in Knowledge Bases for Natural Language Processing.

Yasumasa Onoe and Greg Durrett. 2020. Interpretable entity representations through large-scale typing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 612–624, Online. Association for Computational Linguistics.

James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. The development and psychometric properties of liwc2015. Technical report.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy. Association for Computational Linguistics.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of bert. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Kunal Relia, Zhengyi Li, Stephanie H. Cook, and Rumi Chunara. 2019. Race, ethnicity and national origin-based discrimination in social media and hate crimes across 100 u.s. cities. *Proceedings of the International AAAI Conference on Web and Social Media*, 13(01):417–427.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

Koustuv Saha, Benjamin Sugar, John Torous, Bruno Abrahao, Emre Kıcıman, and Munmun De Choudhury. 2019. A social media study on the effects of psychiatric medication use. *Proceedings of the International AAAI Conference on Web and Social Media*, 13(01):440–451.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Florian Schmidt and Thomas Hofmann. 2020. Bert as a teacher: Contextual embeddings for sequence-level reward.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

Nathaniel Swinger, Maria De-Arteaga, Neil Thomas Heffernan IV, Mark DM Leiserson, and Adam Tauman Kalai. 2019. What are the biases in my word embedding? In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, page 305–311, New York, NY, USA. Association for Computing Machinery.

Francielle Vargas, Fabiana Rodrigues de Góes, Isabelle Carvalho, Fabrício Benevenuto, and Thiago Pardo. 2021. Contextual-lexicon approach for abusive language detection. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1438–1447, Held Online. INCOMA Ltd.

Victor Veitch, Alexander D' Amour, Steve Yadlowsky, and Jacob Eisenstein. 2021. Counterfactual invariance to spurious correlations in text classification. In *Advances in Neural Information Processing Systems*, volume 34, pages 16196–16208. Curran Associates, Inc.

Victor Veitch, Dhanya Sridhar, and David Blei. 2020. Adapting text embeddings for causal inference. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 919–928. PMLR.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788. Curran Associates, Inc.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, pages 161–170, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Torsten Wörtwein, Lisa B. Sheeber, Nicholas Allen, Jeffrey F. Cohn, and Louis-Philippe Morency. 2021. Human-guided modality informativeness for affective states. In *Proceedings of the 2021 International*

Conference on Multimodal Interaction, ICMI '21, page 728–734, New York, NY, USA. Association for Computing Machinery.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. 2016. Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages. *IEEE Intelligent Systems*, 31(6):82–88.

A Effects of SENTECON(+) parameter choices

To ensure that our findings in Section 5.2 are robust to different parameter choices in SENTECON(+), we conduct analyses over the number of centroids per category, choice of deep language model M_θ, and choice of reference corpus. We take LIWC as our base lexicon for all experiments.

A.1 Number of centroids per category

If our lexicon categories are very broad, we may have reason to believe that it would be useful to have multiple centroids per category, rather than summarizing the category as a single centroid. Here, we report the effects of different numbers of centroids per category on SENTECON(+) performance on downstream tasks.

To define multiple centroids for a given category, we use an unsupervised clustering method to create P clusters of word or sentence embeddings for each category. For each of the P clusters, we compute the centroid as before, so we now have P centroids for every category.

Now, given a new sentence s_{new}, we compute the similarity between the new sentence and *each centroid of each category*. Then when computing our SENTECON(+) representation, the weight for category c_i is taken to be the largest similarity between s_{new} and any one of the centroids for c_i. That is, letting $\mathbf{centroid}(c_i)_p$ be the p-th cluster centroid for category c_i and $\mathbf{h}(s_{new})_i$ again be the SENTECON(+) weight for c_i,

$$\mathbf{h}(s_{new})_i = \max_{p \in P}(g(r_{s_{new}}, \mathbf{centroid}(c_i)_p)$$

Across our evaluation tasks, we do not find additional centroids to produce substantial performance gains (Table 4), though small improvements are observed for SENTECON on SST and MOSI. We encourage users of SENTECON(+) to treat the number of centroids as a tunable hyperparameter—but in many cases, including the ones we explore in our experiments, a single centroid per category should be sufficient.

A.2 Choice of language model

Here, we report the effects of different choices of M_θ model architectures on SENTECON performance on downstream tasks. All language models are pre-trained.

To determine the impact of selecting a well-performing language model as our M_θ, we construct additional SENTECON representations using pre-trained DistilRoBERTa (Sanh et al., 2020), MiniLM (Wang et al., 2020), BERT (Devlin et al., 2019b), and RoBERTa (Liu et al., 2019), all of which are transformer-based language models like MPNet. Comparing the performance of SENTECON representations to Lexicon and Lexicon-word2vec, *we observe that* SENTECON *continues to outperform both baselines across all choices of* M_θ (Table 5), even for the smaller MiniLM model.

SENTECON performance seems to scale generally—though not perfectly—with M_θ performance. For example, MPNet and RoBERTa are the best-performing pre-trained language models, and SENTECON with MPNet and RoBERTa as M_θ are the best-performing variants of SENTECON (aside from the MELD sentiment task, where SENTECON with BERT achieves the best performance).

A.3 Choice of reference corpus

In Section 4.2, we describe our approach for creating a reference corpus: using a held-out portion of the task dataset. However, it is useful to know whether the reference corpus must be from the *same* domain as the task or whether a reference corpus from a *similar* domain may suffice to improve performance over SENTECON. With MELD as our downstream task dataset, we select as our reference corpora one dataset that is similar to MELD (**Switchboard**, a series of utterances from dyadic phone conversations); one that is moderately different (**NYT**[7], a dataset of New York Times article summaries from 2020); and one that is extremely different (**PubMed**, a collection of abstracts from academic papers published in medical journals) (Holliman et al., 1992; Dernoncourt and Lee, 2017). More details about these datasets are provided in Section B.4. To reduce computational load, we use the smaller transformer-based language model MiniLM (Wang et al., 2020) as our M_θ.

We evaluate SENTECON+ representations on the MELD emotion and sentiment classification

[7] https://www.kaggle.com/datasets/benjaminawd/new-york-times-articles-comments-2020

Representation	# centroids	MELD (e)	MELD (s)	SST	IMDb	MOSI
SENTECON (L)	1	**57.2**	**68.1**	93.4	**95.1**	67.2
SENTECON (L)	2	55.8	67.7	93.1	94.9	67.8
SENTECON (L)	3	55.6	67.4	93.5	94.9	**69.3**
SENTECON (L)	4	55.5	67.2	**93.5**	**95.1**	68.4
SENTECON+ (L)	1	**59.9**	**68.1**	**93.2**	**95.0**	**67.3**
SENTECON+ (L)	2	59.0	67.4	92.8	94.7	66.9
SENTECON+ (L)	3	57.6	68.0	93.2	93.6	-
SENTECON+ (L)	4	55.3	66.9	93.1	93.8	-

Table 4: Performance comparisons of SENTECON(+) across different numbers of centroids per category. We use LIWC as the base lexicon and fine-tuned MPNet as M_θ. We report test accuracy for MELD, SST, and IMDb and test R^2 for MOSI.

Representation	M_θ	MELD (e)	MELD (s)	SST	IMDb	MOSI
Lexicon (L)	-	46.5	49.5	67.8	76.7	0.202
LIWC+word2vec	-	47.5	49.4	78.7	81.4	0.270
SENTECON (L)	MPNet	47.7	57.6	**86.5**	**84.2**	**0.505**
SENTECON (L)	MiniLM	50.7	56.4	77.9	75.7	0.411
SENTECON (L)	DistilRoBERTa	48.6	54.7	85.2	82.4	0.289
SENTECON (L)	BERT	58.7	**65.4**	81.3	84.4	0.364
SENTECON (L)	RoBERTa	**56.5**	60.7	79.4	83.7	0.118
Pre-trained embedding	MPNet	58.9	65.0	89.5	89.2	**0.482**
Pre-trained embedding	MiniLM	59.9	64.7	81.3	81.1	0.150
Pre-trained embedding	DistilRoBERTa	58.5	64.9	88.3	87.6	0.264
Pre-trained embedding	BERT	56.8	63.2	86.1	89.1	0.259
Pre-trained embedding	RoBERTa	**60.5**	**65.0**	**90.3**	**92.0**	0.177

Table 5: Performance comparisons of SENTECON when used with different pre-trained language models as M_θ in downstream prediction tasks. We report test accuracy for MELD, SST, and IMDb and test R^2 for MOSI.

Reference corpus	MELD (e)	MELD (s)
None	50.7	56.4
MELD	**55.5**	**61.3**
Switchboard	49.7	55.6
NYT	49.9	53.9
PubMed	50.6	55.7

Table 6: Performance comparisons of SENTECON+ on MELD when used with different reference corpora. We use LIWC as the base lexicon and pre-trained MiniLM as M_θ, and we report test accuracies.

tasks using the three new reference corpora (Table 6). We find that using any of the three new reference corpora yields worse performance than using a held-out set from MELD (and in fact, worse performance than not using a reference corpus at all). These results support the conclusion that the reference corpus should be from the *same* domain as the task. Only SENTECON+ with a reference corpus consisting of a portion of the task dataset itself provides performance improvements over SENTECON with no reference corpus.

B Experimental Details

B.1 LIWC categories

The full list of non-grammatical LIWC categories used in our experiments is as follows: *affect*, *posemo*, *negemo*, *anx*, *anger*, *sad*, *social*, *family*, *friend*, *female*, *male*, *cogproc*, *insight*, *cause*, *discrep*, *tentat*, *certain*, *differ*, *percept*, *see*, *hear*, *feel*, *bio*, *body*, *health*, *sexual*, *ingest*, *drives*, *affiliation*, *achiev*, *power*, *reward*, *risk*, *focuspast*, *focuspresent*, *focusfuture*, *relativ*, *motion*, *space*, *time*, *work*, *leisure*, *home*, *money*, *relig*, *death*, *informal*, *swear*, *netspeak*, *assent*, *nonflu*, *filler*.

The list of excluded grammatical LIWC categories is as follows: *function*, *pronoun*, *ppron*, *i*, *we*, *you*, *shehe*, *they*, *ipron*, *article*, *prep*, *auxverb*, *adverb*, *conj*, *negate*, *verb*, *adj*, *compare*, *interrog*, *number*, *quant*.

B.2 Empath categories

The full list of Empath categories used in our experiments is as follows: *help, office, dance, money, wedding, domestic_work, sleep, medical_emergency, cold, hate, cheerfulness, aggression, occupation, envy, anticipation, family, vacation, crime, attractive, masculine, prison, health, pride, dispute, nervousness, government, weakness, horror, swearing_terms, leisure, suffering, royalty, wealthy, tourism, furniture, school, magic, beach, journalism, morning, banking, social_media, exercise, night, kill, blue_collar_job, art, ridicule, play, computer, college, optimism, stealing, real_estate, home, divine, sexual, fear, irritability, superhero, business, driving, pet, childish, cooking, exasperation, religion, hipster, internet, surprise, reading, worship, leader, independence, movement, body, noise, eating, medieval, zest, confusion, water, sports, death, healing, legend, heroic, celebration, restaurant, violence, programming, dominant_heirarchical, military, neglect, swimming, exotic, love, hiking, communication, hearing, order, sympathy, hygiene, weather, anonymity, trust, ancient, deception, fabric, air_travel, fight, dominant_personality, music, vehicle, politeness, toy, farming, meeting, war, speaking, listen, urban, shopping, disgust, fire, tool, phone, gain, sound, injury, sailing, rage, science, work, appearance, valuable, warmth, youth, sadness, fun, emotional, joy, affection, traveling, fashion, ugliness, lust, shame, torment, economics, anger, politics, ship, clothing, car, strength, technology, breaking, shape_and_size, power, white_collar_job, animal, party, terrorism, smell, disappointment, poor, plant, pain, beauty, timidity, philosophy, negotiate, negative_emotion, cleaning, messaging, competing, law, friends, payment, achievement, alcohol, liquid, feminine, weapon, children, monster, ocean, giving, contentment, writing, rural, positive_emotion, musical*.

B.3 Human evaluation study details

Question. In the human evaluation study, annotators were asked the following question:

For each of the following topics or categories, please rate to what extent the topic is expressed in the language, content, and meaning of the sentence. It is possible that none of the topics may be expressed; it is also possible that the topic you feel is most strongly expressed is not present.

If a topic is marked with an asterisk, please hover your cursor over each topic for a more detailed description of the topic.

They were asked to rate according to the following scale and were provided with the accompanying descriptions.

- *Not expressed*: Out of all possible interpretations of the sentence above, you cannot imagine a scenario in which the speaker of the sentence was expressing the topic.
- *Potentially expressed*: You can imagine at least one scenario in which the speaker of the sentence was expressing the topic.
- *Most likely expressed*: The most natural interpretation of the sentence clearly expresses the topic.

Category batches. As mentioned in the main paper, the 52 LIWC categories were randomly split into 5 sets of roughly equal size to avoid annotator fatigue. The splits were as follows:

- Batch 1: *netspeak*, *differ*, *cause*, *nonflu*, *discrep*, *drivers*, *relig*, *swear*, *feel*, *home*, *family*
- Batch 2: *leisure*, *sexual*, *see*, *bio*, *certain*, *money*, *percept*, *female*, *death*, *anger*, *cogproc*
- Batch 3: *filler*, *sad*, *posemo*, *friend*, *relativ*, *ingest*, *body*, *work*, *time*, *social*, *informal*
- Batch 4: *focusfuture*, *anx*, *affiliation*, *motion*, *power*, *reward*, *space*, *tentat*, *risk*, *focuspresent*, *affect*
- Batch 5: *negemo*, *hear*, *male*, *health*, *insight*, *achiev*, *focuspast*, *assent*

Inter-rater reliability. To assess the reliability of our annotations, we calculated intraclass correlation coefficients (ICCs) using the *agreement* software package (Girard, 2020). For each batch of sentences, we computed the ICC and its 95% confidence interval, then averaged these across category batches (Table 7). We averaged ICCs over all batches to obtain the overall ICC.

Annotators. Annotators were required to be fluent in English and to be nationals of one of the following countries: the United States, the United Kingdom, Ireland, Australia, or Canada.

Annotators were further required to have a prior approval rating of $\geq 95\%$, and an attention check

Category batch	ICC
1	**0.580** [0.467, 0.662]
2	**0.688** [0.603, 0.749]
3	**0.730** [0.669, 0.777]
4	**0.715** [0.654, 0.763]
5	**0.718** [0.635, 0.777]
Average	**0.686** [0.606, 0.746]

Table 7: ICCs of human annotations of sentence categories across category batches, with 95% confidence intervals.

question was included in every sentence batch. All annotators passed the attention check.

We took care to compensate annotators at a rate above the local minimum wage. Annotators received an average hourly wage of 8.00 USD.

B.4 Data

Details of train, test, and reference corpus splits are provided in Table 8, including dataset composition and licensing information. For datasets released with existing train and test splits, we split the existing test set into a reference corpus and new test set. As mentioned in the main paper, all datasets are already publicly available, and the additional splits created for the reference corpora are available on our GitHub repository. All datasets are in English.

B.5 Training details

Our language models were built on the HuggingFace[10] `transformers` library (version 4.16.2), with pre-trained models taken from the HuggingFace model hub. When fine-tuning these models on the task datasets, we used an Adam optimizer and learning rates $[10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}]$, and we found 10^{-5} to be the best learning rate across all models. We trained for 15 epochs and selected the model with the best 5-fold cross-validation loss. All other hyperparameters were set to Trainer class defaults from the `transformers` library.

The number of parameters for each of the deep language models used is reported in Table 9. The license names for the models are also provided.

[6] https://github.com/A2Zadeh/CMU-MultimodalSDK/blob/master/LICENSE.txt

[7] https://catalog.ldc.upenn.edu/license/ldc-non-members-agreement.pdf

[10] https://huggingface.co/

B.6 Computing resources

SENTECON(+) requires only using an existing deep language model to generate embeddings and consequently is not particularly computationally demanding. Fine-tuning deep language models is more resource-intensive, but we use these only to a limited extent in our experiments, and only on small datasets. We estimate the number of GPU hours used in these experiments to be around 20. All experiments were conducted on machines with consumer-level NVIDIA graphics cards.

Dataset	n_{train}	n_{test}	$n_{reference}$	n_{total}	License
MELD	9,989	2,610	1,109	13,708	GPL-3.0
SST	6,920	1,821	872	9,613	Unknown
IMDb	25,000	15,000	10,000	50,000	Unknown
MOSI	1,034	500	665	2,199	Other[8]
Switchboard	-	-	15,000	-	Other[9]
NYT	-	-	16,784	-	CC BY-NC-SA 4.0
PubMed	-	-	15,000	-	Unknown

Table 8: Composition of dataset splits. The number of train, test, and reference corpus samples is given, along with total samples for each dataset. Licensing information is also given.

Language model	# dimensions	# parameters	License
MPNet	768	109,486,464	MIT
RoBERTa	768	124,645,632	MIT
BERT	768	109,482,240	Apache-2.0
MiniLM	384	22,713,216	Apache-2.0
DistilRoBERTa	768	82,118,400	Apache-2.0

Table 9: Number of dimensions, parameters, and license for each deep language model.

Sentence	Lexicon (L)	LIWC+ word2vec	SENTECON (pre-trained)	SENTECON+ (pre-trained)	SENTECON (fine-tuned)	SENTECON+ (fine-tuned)
What?	0.112	0.218	-0.054	**0.251**	0.223	-0.129
Really?	**0.211**	-0.153	0.175	-0.011	0.147	0.089
It's really sweet and— and tender.	0.001	0.284	0.273	**0.325**	0.260	0.003
Tell her to wear her own earrings.	0.222	0.239	0.307	**0.445**	0.260	0.003
This is totally your fault!	0.453	0.358	0.663	**0.672**	0.465	0.409
My first time with Carol was...	0.166	0.234	0.456	**0.487**	-0.041	0.126
No! Ah-ah-ah-ah-ah! You can have this back when the five pages are done! Ahh!	-0.064	**0.300**	0.192	0.138	-0.163	-0.176
Yeah, and to save you from any embarrassment umm, I think maybe I should talk first.	0.245	0.100	0.311	**0.381**	-0.026	0.126
Hey. Call me when you get there. Okay?	0.143	0.206	0.158	**0.365**	-0.049	0.314
What?! I didn't touch a guitar!	0.407	0.293	**0.646**	0.529	0.284	0.320

Table 10: Pearson correlations (r) between human category annotations and category encodings produced by traditional lexicon-based methods, SENTECON, and SENTECON+. We use SENTECON(+) with both pre-trained and fine-tuned MPNet as M_θ.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
8

☑ A2. Did you discuss any potential risks of your work?
9

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3, 4, 5

☑ B1. Did you cite the creators of artifacts you used?
4, 5, B.4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
B.3 (Table 7), B.4 (Table 8)

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
9

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Although we used crowdworkers, we did not collect any information about the workers themselves. We note in Section 9 that this is the case and that the nature of our data is not sensitive. All other datasets used are public and do not contain identifying information.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
4, B

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
B.4 (Table 7)

C ☑ Did you run computational experiments?

4, 5, A

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
B.4 (Table 8), B.5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4, B

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5, B.3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
B.5

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
4.4, 5.1

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
B.2

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
4.4, B.3

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
9

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
As no information was collected about the crowdworkers themselves, nor did we engage with or intervene upon the crowdworkers in any way, the data collection protocol does not fall under the definition of research that requires IRB review.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
We report geographic restrictions on annotator location in Section B.3. Although these restrictions were implemented in our crowdsourcing platform, we did not collect any information about the geographic location of the crowdworkers. We also did not collect any demographic information from the annotators.