Which Evaluations Uncover Sense Representations that Actually Make Sense?

Linguistic Resources and Evaluation Conference, 2020, 10 pages.

Contact Jordan Boyd-Graber (jbg@boydgraber.org) for questions about this paper.
Which Evaluations Uncover Sense Representations that Actually Make Sense?

Fenfei Guo, Jordan Boyd-Graber, Mohit Iyyer, Leah Findlater
University of Maryland, University of Maryland, University of Massachusetts, University of Washington
fenfeigo@cs.umd.edu, jbg@umiacs.umd.edu, miyery@cs.umass.edu, leahkt@uw.edu

1. Context, Sense, and Representation

Computers need to represent the meaning of words in context. BERT (Devlin et al., 2019) and ELMo (Peters et al., 2018) have dramatically changed how natural language processing represents text. Rather than one-size-fits-all word vectors that ignore the nuance of how words are used in context, these new representations have topped the leaderboards for question answering, inference, and classification. Contextual representations have supplanted multisense embeddings (Camacho-Collados and Pilehvar, 2018). While these methods learn a vector for each sense, they do not encode meanings in downstream tasks as well as contextual representations (Peters et al., 2018).

However, computers are not the only consumer of text representations. Humans also use word representations to understand diachronic drift, investigate a language’s sense inventory, or to cluster and explore documents. Thus, a primary role for multisense word embeddings is human understanding of word meanings. Unfortunately, multisense models have only been evaluated on computer-centric dimensions and have ignored the question of sense interpretability.

We first develop measures for how well models encode and select which sense is used in a sentence better than single-prototype word embeddings tied to each type. However, existing sense representations are not uniformly better: although they work well for computer-centric evaluations, they fail for human-centric tasks like inspecting a word’s sense inventory. To expose this discrepancy, we propose a new coherence evaluation for sense embeddings. We also describe a minimal model (Gumbel Attention for Sense Induction) optimized for discovering interpretable sense representations that are more coherent than existing sense embeddings.

Keywords: sense representations, interpretability evaluation, embeddings

2. Attentional Sense Induction

Before we explore human interpretability of sense induction, we first describe our simple models to disentangle word senses. Our two models are built on Word2Vec (Mikolov et al., 2013a, Mikolov et al., 2013b), which we review in Section 2.1. Both models use a straightforward attention mechanism to select which sense is used in a token’s context, which we contrast to alternatives for sense selection (Section 2.3). Building on these foundations, we introduce our model, GASI, and along the way introduce a soft-attention stepping-stone (SASI).

2.1. Foundations: Skip-Gram and Gumbel

Word2Vec jointly learns word embeddings \( \mathbf{W} \in \mathbb{R}^{V \times d} \) and context embeddings \( \mathbf{C} \in \mathbb{R}^{V \times d} \). More specifically, given a vocabulary \( V \) and embedding dimension \( d \), it maximizes the likelihood of the context words \( c_{ji} \) that surround a given center word \( w_i \) in a context window \( \tilde{c}_i \),

\[
J(\mathbf{W}, \mathbf{C}) \propto \sum_{w_i \in V} \sum_{c_{ji} \in \tilde{c}_i} \log P(c_{ji} | w_i; \mathbf{W}, \mathbf{C}),
\]

where \( P(c_{ji} | w_i) \) is over the vocabulary,

\[
P(c_{ji} | w_i; \mathbf{W}, \mathbf{C}) = \frac{\exp(c_{ji}^\top w_i)}{\sum_{a \in V} \exp(c_{ja}^\top w_i)}.
\]

In practice, \( \log P(c_{ji} | w_i) \) is approximated by negative sampling. We extend it to learn representations for individual word senses.

2.2. Gumbel Softmax

As we introduce word senses, our model will need to select which sense is relevant for a context. The Gumbel softmax (Jang et al., 2017, Maddison et al., 2017) approximates the sampling of discrete random variables; we use...
it to select the sense. Given a discrete random variable \( X \) with \( P(X = k) \propto \alpha_k, \alpha_k \in (0, \infty) \), the Gumbel-max (Gumbel and Lieblein, 1954) refactors the sampling of \( X \) into
\[
X = \arg \max_k (\log \alpha_k + g_k),
\]
where the Gumbel noise \( g_k = -\log(-\log(u_k)) \) and \( u_k \) are i.i.d. from Uniform(0, 1). The Gumbel softmax approximates sampling one-hot \( \arg \max_k (\log \alpha_k + g_k) \) by
\[
y_k = \text{softmax}(\log \alpha_k + g_k)/\tau.
\]
Unlike soft selection of senses, the Gumbel softmax can make harder selections, which will be more interpretable to humans.

2.3. Why Attention? Musing on Alternatives
For fine-grained sense inventories, it makes sense to have graded assignment of tokens to senses (Erk et al., 2009, Jurgens and Klapavikus, 2015). However, for coarse senses—except for humor (Miller et al., 2017)—words typically are associated with a single sense, often a single sense per discourse (Gale et al., 1992). A good model should respect this. Previous models either use non-differentiable objectives or—in the case of the current state of the art, MUSE (Lee and Chen, 2017)—reinforcement learning to select word senses. By using Gumbel softmax, our model both approximates discrete sense selection and is differentiable. As we argue in the next section, applications with a human in the loop are best assisted by discrete senses; the Gumbel softmax, which we develop for our task here, helps us discover these discrete senses.

2.4. Attentional Sense Induction
Embeddings We learn a context embedding matrix \( C \in \mathbb{R}^{V \times d} \) and a sense embedding tensor \( S \in \mathbb{R}^{V \times K \times d} \). Unlike previous work (Neelakantan et al., 2014, Lee and Chen, 2017), no extra embeddings are kept for sense induction.

Number of Senses For simplicity and consistency with previous work, our model has \( K \) fixed senses. Ideally, if we set a large number of \( K \), with a perfect pruning strategy, we can estimate the number of senses per type by removing duplicated senses.

However, this is challenging (McCarthy et al., 2016); instead we use a simple pruning strategy. We estimate a pruning threshold \( \lambda \) by averaging the estimated duplicate sense and true neighbor distances,
\[
\lambda = \frac{1}{2} (\text{mean}(D_{dup}) + \text{mean}(D_{nn})),
\]
where \( D_{dup} \) are the cosine distances for duplicated sense pairs and \( D_{nn} \) is that of true neighbors (different types). We sample 100 words and if two senses are top-5 nearest neighbors of each other, we consider them duplicates.

After pruning duplicated senses with \( \lambda \), we can retrain a new model with estimated number of senses for each type by masking the sense attentions.\(^1\) Results in Table 2 and 5 validate our pruning strategy.

\(^1\) More details and analysis about pruning are in Appendix C.1

Sense Attention in Objective Function Assuming a center word \( w_i \) has senses \( \{s_{i1}^k, s_{i2}^k, \ldots, s_{iK}^k\} \), the original Skip-Gram likelihood becomes a marginal distribution over all senses of \( w_i \) with sense induction probability \( P(s_{i1}^k \mid w_i) \); we focus on the disambiguation given local context \( c_i \) and estimate \( P(s_{i1}^k \mid w_i, c_i) \) and thus,
\[
P(c_i^j \mid w_i) \approx \sum_{k=1}^K P(c_i^j \mid s_{i1}^k) P(s_{i1}^k \mid w_i, c_i),
\]
Replacing \( P(c_i^j \mid w_i) \) in Equation 1 with Equation 6 gives our objective function \( J(S, C) \propto \)
\[
\sum_{w_i \in V} \sum_{c_i \in \tilde{c}_i} \log \sum_{k=1}^K P(c_i^j \mid s_{i1}^k) P(s_{i1}^k \mid w_i, c_i).
\]

Modeling Sense Attention We can model the contextual sense induction distribution with soft attention. We call the resulting model soft-attention sense induction (SASI). Although it is a stepping stone to our final model, we compare against it in our experiments as it isolates the contributions of hard attention. In SASI, the sense attention is conditioned on the entire local context \( \tilde{c}_i \) with softmax:
\[
P(s_{i1}^k \mid w_i, \tilde{c}_i) = \frac{\exp((\tilde{c}_i^\top s_{i1}^k))/\tau}{\sum_{k=1}^K \exp((\tilde{c}_i^\top s_{i1}^k))/\tau},
\]
where \( \tilde{c}_i \) is the mean of the context vectors in \( \tilde{c}_i \). A derivation of how this affects negative sampling is in Appendix A.

2.5. Scaled Gumbel Softmax for Sense Disambiguation
To learn distinguishable sense representations, we implement hard attention in our full model, Gumbel Attention for Sense Induction (GASI). While hard attention is conceptually attractive, it can increase computational difficulty: discrete choices are not differentiable and thus incompatible with modern deep learning frameworks. To preserve differentiability (and to avoid equally complex reinforcement learning), we apply the Gumbel softmax reparameterization trick to our sense attention function (Equation 8).

Vanilla Gumbel The discrete sense sampling from Equation 8 can be refactored
\[
\tilde{z}_i = \text{one-hot}(\arg \max_k (\tilde{c}_i^\top s_{i1}^k + g_k)),
\]
and the hard attention approximated
\[
y_k = \text{softmax}(\exp((\tilde{c}_i^\top s_{i1}^k + g_k))/\tau).
\]

Scaled Gumbel Gumbel softmax learns a flat distribution over senses even with low temperatures: the dot product \( \tilde{c}_i^\top s_{i1}^k \) is too small\(^2\) compared to the Gumbel noise \( g_k \). Thus we use a scaling factor \( \beta \) to encourage sparser distributions,\(^3\)
\[
\tilde{z}_i = \text{softmax}((\tilde{c}_i^\top s_{i1}^k + \beta g_k)/\tau),
\]
\(^2\) This is from float32 precision and saturation of \( \log(\sigma(z_i)) \); detailed further in Figure 3 in Appendix.

\(^3\) Normalizing \( \tilde{c}_i^\top s_{i1}^k \) or directly using \( \log P(s_{i1}^k \mid w_i, \tilde{c}_i) \) results in a similar outcome.
and tune it as a hyperparameter. We append GASI-$\beta$ to the name of models with a scaling factor. This is critical for learning distinguishable senses (Figure 2, Table 2, and Table 5). Our final objective function for GASI-$\beta$ is

$$J(S, C) \propto \sum_{w_i \in V} \sum_{w_c \in C_i} \sum_{k=1}^{K} \gamma_k \log P(w_c \mid s_k^i). \quad (12)$$

### 3. Data and Training

For fair comparisons, we try to remain consistent with previous work (Huang et al., 2012, Neelakantan et al., 2014, Lee and Chen, 2017) in all aspects of training. In particular, we train GASI on the same April 2010 Wikipedia snapshot (Shaoul and Westbury, 2010) with 1B tokens and the same vocabulary released by Neelakantan et al. (2014); set the number of senses $K = 3$ and dimension $d = 300$ for each word unless otherwise specified. More details are in the Appendix. Following Maddison et al. (2017), we fix the temperature $\tau = 0.5$, and tune the scaling factor $\beta = 0.4$ using grid search within $\{0.1 \ldots 0.9\}$ on AvgSimC for contextual word similarity (Section 5); this tuning preceded all interpretability experiments. If not reprinted, numbers for competing models are either computed with pre-trained embeddings released by authors or trained on released code.

### 4. Evaluating Interpretability

We turn to traditional evaluations of sense embeddings later (Section 5), but our focus is on human interpretability. If you show a human the senses, can they understand why a model would assign a sense to that context? This section evaluates whether the representations make sense to human consumers of multisense models.

In the age of BERT and ELMo, these are the dimensions that are most critical for multisense representations. While contextual word vectors are most useful for computer understanding of meaning, humans often want an overview of word meanings for other tasks.

Sense representations are useful for human-in-the-loop applications. They help understand semantic drift (Hamilton et al., 2016): how do the meanings of “gay” reflect social progress? They help people learn languages (Noraset et al., 2017): what does it mean when someone says that I “embarrassed” them? They help linguists understand the sense inventory of a language (Kawahara et al., 2014): what are the frames that can be used by the verb “participate”? These questions (and human understanding) are helped by discrete senses, which the Gumbel softmax uncovers.

More broadly, this is the goal of interpretable machine learning (Doshi-Velez and Kim, 2017). While downstream models do not always need an interpretable explanation of why a model uses a particular representation, interactive machine learning and explainable machine learning do. To date, multisense representations ignore this use case.

**Qualitative analysis** Previous papers use nearest neighbors of a few examples to qualitatively argue that their models have captured meaningful senses of words. We also give an example in Figure 2, which provides an intuitive view on how the learned senses are clustered by visualizing the nearest neighbors of word “bond” using t-SNE projection (van der Maaten and Hinton, 2008). Our model (right) disentangles the three sense of “bond” clearly.

However, examples can be cherry-picked. This problem bedeviled topic modeling until rigorous human evaluation was introduced (Chang et al., 2009). We adapt both aspects of their evaluations: word intrusion (Schnabel et al., 2015) to evaluate whether individual senses are coherent and topic intrusion—rather sense intrusions in this setting—to evaluate whether humans agree with models’ sense assignments in context. Using crowdsourced evaluations from Figure-Eight, we compare our models with two previous state-of-the-art sense embeddings models, i.e., MSSG (Neelakantan et al., 2014) and MUSE (Lee and Chen, 2017).  

### 4.1. Word Intrusion for Sense Coherence

Schnabel et al. (2015) suggest a “good” word embedding should have coherent neighbors and evaluate coherence by word intrusion. They present crowdworkers four words: three are close in embedding space but one is an “intruder”. If the embedding makes sense, contributors will easily spot the word that “does not belong”.

Similarly, we examine the coherence of ten nearest neighbors of senses in the contextual word sense selection task (Section 4.2) and replace one neighbor with an “intruder”. We generate three intruders for each sense and collect three judgments per intruder. To account for variation in users and intruders, we count an instance as “correct” if two or more crowdworkers correctly spot the intruder.

Like Chang et al. (2009), we want the “intruder” to be about

<table>
<thead>
<tr>
<th>Model</th>
<th>Sense Accuracy</th>
<th>Judgment Accuracy</th>
<th>Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUSE</td>
<td>67.33</td>
<td>62.89</td>
<td>0.73</td>
</tr>
<tr>
<td>MSSG-30K</td>
<td>69.33</td>
<td>66.67</td>
<td>0.76</td>
</tr>
<tr>
<td>GASI-β</td>
<td>71.33</td>
<td>67.33</td>
<td>0.77</td>
</tr>
</tbody>
</table>

**Table 1**: Word intrusion evaluations on top ten nearest neighbors of sense embeddings. Users find misfit words most easily with GASI-β, suggesting these representations are more interpretable.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>$P$</th>
<th>Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUSE</td>
<td>28.0</td>
<td>0.33</td>
<td>0.68</td>
</tr>
<tr>
<td>MSSG-30K</td>
<td>44.5</td>
<td>0.37</td>
<td>0.73</td>
</tr>
<tr>
<td>GASI (no β)</td>
<td>33.8</td>
<td>0.33</td>
<td>0.68</td>
</tr>
<tr>
<td>GASI-β</td>
<td>50.0</td>
<td>0.48</td>
<td>0.75</td>
</tr>
<tr>
<td>GASI-β-pruned</td>
<td>75.2</td>
<td>0.67</td>
<td>0.96</td>
</tr>
</tbody>
</table>

**Table 2**: Human-model consistency on contextual word sense selection; $P$ is the average probability assigned by the model to the human choices. GASI-β is most consistent with crowdworkers. Reducing sense duplication by retraining our model with pruning mask improves human-model agreement.
as frequent as the target but not too similar. For sense $s_i^m$ of word type $w_i$, we randomly select a word from the neighbors of another sense $s_i^n$ of $w_i$.

All models have comparable model accuracy. GASI-β learns senses that have the highest coherence while MUSE learns mixtures of senses (Table 1).

We use the aggregated confidence score provided by Figure-Eight to estimate the level of inter-rater agreement between multiple contributors (Figure Eight, 2018). The agreement is high for all models, and GASI-β has the highest agreement, suggesting that the senses learned by GASI-β are easier to interpret.

### 4.2. Contextual Word Sense Selection

The previous task measures whether individual senses are coherent. Now we evaluate models’ disambiguation of senses in context.

**Task Description**  Given a target word in context, we ask a crowdworker to select which sense group best fits the sentence. Each sense group is described by its top ten distinct nearest neighbors, and the sense group order is shuffled.

**Data Collection**  We select fifty nouns with five sentences from SemCor 3.0 (Miller et al., 1994). We first filter all word types with fewer than ten sentences and select the fifty most polysemous nouns from WordNet (Miller and Fellbaum, 1998) among the remaining senses. For each noun, we randomly select five sentences.

**Metrics**  For each model, we collect three judgments for each question. We consider a model correct if at least two crowdworkers select the same sense as the model.

**Sense Disambiguation and Interpretability**  If humans consistently pick the same sense as the model, they must first understand the choices, thus implying the nearest neighbor words were coherent. Moreover, they also agree that among those senses, that sense was the right choice for this token. GASI-β selections are most consistent with humans’; its selections have the highest accuracy and assigns the largest probability assigned to the human choices (Table 2). Thus, GASI-β produces sense embeddings that are both more interpretable and distinguishable. GASI without a scaling factor, however, has low consistency and flat sense distribution.

**Model Confidence**  However, some contexts are more ambiguous than others. For fine-grained senses, best practice is to use graded sense assignments (Erk et al., 2013). Thus, we also show the model’s probability of the top human choice; distributions close to $\frac{1}{K}$ ($0.33$) suggest the model learns a distribution that cannot disambiguate senses. We consider granularity of senses further in Section 6.
Inter-rater Agreement We use the confidence score computed by Figure-Eight to estimate the raters’ agreement for this task. GASI-β has the highest human-model agreement, while both MUSE and GASI without scaling have the lowest.

Error Analysis Next, we explore why crowdworkers disagree with the model even though the senses are interpretable (Table 1). Is it that the model has learned duplicate senses that both the users and model cannot distinguish (the senses are all bad or identical) or is it that crowdworkers agree with each other but disagree with the model (the model selects bad senses)?

Two trends suggest duplicate senses cause disagreement both for humans with models and humans with each other. For two measures of sense similarity—simple word overlap and GloVE similarity—similarity is lower when users and models agree (Table 3). Humans also agree with each other more. For GASI-β, pairs with perfect agreement have a word overlap of around 2.5, while the senses with lowest agreement have overlap around 5.5.

To reduce duplicated senses, we retrain the model with pruning (Section 2.4, Equation 5). We remove a little more than one sense per type on average. To maintain the original setting, for word types that have fewer than three senses left, we compute the nearest neighbors to dummy senses represented by random embeddings. Our model trained with pruning mask (GASI-β-pruned) reaches very high inter-rater agreement and higher human-model agreement than models with a fixed number of senses (Table 2, bottom).

5. Word Similarity Evaluation

GASI and GASI-β are interpretable, but how do they fare on standard word similarity tasks?

Contextual Word Similarity Tailored for sense embedding evaluation, Stanford Contextual Word Similarities (Huang et al., 2012, SCWS) has 2003 word pairs tied to context sentences. These tasks assign a pair of word types (e.g., “green” and “buck”) a similarity/relatedness score. Moreover, both words in the pair have an associated context. These contexts disambiguate homonymous and polysemous word types and thus captures sense-specific similarity. Thus, we use this dataset to tune our hyperparameters, comparing Spearman’s rank correlation \( \beta \) between embedding similarity and the gold similarity judgments: higher scores imply the model captures semantic similarities consistent with the trusted similarity scores.

To compute the word similarity scores with senses we use two metrics (Reisinger and Mooney, 2010) that take context and sense disambiguation into account: MaxSimC computes the cosine similarity \( \cos(s_1^i, s_2^j) \) between the two most probable senses \( s_1^i \) and \( s_2^j \) that maximizes \( P(s_1^i | w_1, c_1) \). AvgSimC weights average similarity over the combinations of all senses \( \sum_{i=1}^{K_1} \sum_{j=1}^{K_2} P(s_1^i | w_1, c_1) P(s_2^j | w_2, c_2) \cos(s_1^i s_2^j) \).

We compare variants of our model with existing sense embedding models (Table 5), including two previous SOTAs: the clustering-based Multi-Sense Skip-Gram model (Neelakantan et al., 2014, MSSG) on AvgSimC and the RL-based Modularizing Unsupervised Sense Embeddings (Lee and Chen, 2017, MUSE) on MaxSimC. GASI better captures similarity than SASI, corroborating that hard attention aids word sense selection. GASI without scaling has the best MaxSimC; however, it learns a flat sense distribution (Figure 2). GASI-β has the best AvgSimC and a competitive MaxSimC. While MUSE has a higher MaxSimC than GASI-β, it fails to distinguish senses as well (Figure 2, Section 4).

We also evaluate the retrained model with pruning mask on this dataset. GASI-β-pruned has the same AvgSimC as GASI-β and higher local similarity correlation (Table 5, bottom), validating our pruning strategy (Section 2.4).

Word Sense Selection in Context SCWS evaluates models’ sense selection indirectly. We further compare GASI-β with previous SOTA, MSSG-30K and MUSE, on the Word in Context dataset (Pilehvar and Camacho-Collados, 2019, WIC) which requires the model to identify whether a word has the same sense in two contexts. To reduce the variance in training and to focus on evaluating the sense selection module, we use an evaluation suited for unsupervised models: if the model selects different sense vectors given contexts, we mark that the word has different senses.5 For MUSE, MSSG and GASI-β, we use each model’s sense selection module; for DeConf (Pilehvar and Collier, 2016) and SW2V (Mancini et al., 2017), we follow Pilehvar and Camacho-Collados (2019) and Pelevena et al. (2016) by selecting the closest sense vectors to the context vector. DeConf results are comparable to supervised results (59.4±0.7), GASI-β has the best result (55.3) apart from DeConf itself (58.55)(full results in Table 8 in appendix), which uses the same sense inventory (Miller and Fellbaum, 1998, WordNet) as WIC.

Non-Contextual Word Similarity While contextual word similarity is best suited for our model and goals, other datasets without contexts (i.e., only word pairs and a rating) are both larger and ubiquitous for word vector evaluations. To evaluate the semantics captured by each sense-specific embeddings, we compare the models on non-contextual

<table>
<thead>
<tr>
<th>Model</th>
<th>MaxSimC</th>
<th>AvgSimC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huang et al. (2012)-50d</td>
<td>26.1</td>
<td>65.7</td>
</tr>
<tr>
<td>MSSG-6K</td>
<td>57.3</td>
<td>69.3</td>
</tr>
<tr>
<td>MSSG-30K</td>
<td>59.3</td>
<td>69.2</td>
</tr>
<tr>
<td>Tian et al. (2014)</td>
<td>63.6</td>
<td>65.4</td>
</tr>
<tr>
<td>Li and Jurafsky (2015)</td>
<td>66.6</td>
<td>66.8</td>
</tr>
<tr>
<td>Qiu et al. (2016)</td>
<td>64.9</td>
<td>66.1</td>
</tr>
<tr>
<td>Bartunov et al. (2016)</td>
<td>53.8</td>
<td>61.2</td>
</tr>
<tr>
<td>MUSE_Boltzmann</td>
<td>67.9</td>
<td>68.7</td>
</tr>
<tr>
<td>SASI</td>
<td>55.1</td>
<td>67.8</td>
</tr>
<tr>
<td>GASI (w/o scaling)</td>
<td>68.2</td>
<td>68.3</td>
</tr>
<tr>
<td>GASI-β</td>
<td>66.4</td>
<td>69.5</td>
</tr>
<tr>
<td>GASI-β-pruned</td>
<td>67.0</td>
<td>69.5</td>
</tr>
</tbody>
</table>

Table 5: Spearman’s correlation 100\(\beta\) on SCWS (trained on 1B token, 300d vectors except for Huang et al.). GASI and GASI-β both can disambiguate the sense and correlate with human ratings. Retraining the model with pruned senses further improves local similarity correlation.

5For monosemous or out of vocab words, we choose randomly.
Table 6: Spearman’s correlation on non-contextual word similarity (MaxSim). GASI-β has higher correlation on three datasets and is competitive on the others. PFT-GM is trained with two components/senses while other models learn three. A full version including MSSG is in appendix.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MUSE</th>
<th>SASI</th>
<th>GASI</th>
<th>GASI-β</th>
<th>PFT-GM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimLex-999</td>
<td>39.61</td>
<td>31.56</td>
<td>40.14</td>
<td>41.68</td>
<td>40.19</td>
</tr>
<tr>
<td>WS-353</td>
<td>68.41</td>
<td>58.31</td>
<td>68.49</td>
<td>69.36</td>
<td>68.6</td>
</tr>
<tr>
<td>MEN-3k</td>
<td>74.06</td>
<td>65.07</td>
<td>73.13</td>
<td>72.32</td>
<td>77.40</td>
</tr>
<tr>
<td>MC-30</td>
<td>81.80</td>
<td>70.81</td>
<td>82.47</td>
<td>85.27</td>
<td>74.63</td>
</tr>
<tr>
<td>RG-65</td>
<td>81.11</td>
<td>74.38</td>
<td>77.19</td>
<td>79.77</td>
<td>79.75</td>
</tr>
<tr>
<td>YP-130</td>
<td>43.56</td>
<td>48.28</td>
<td>49.82</td>
<td>56.34</td>
<td>59.39</td>
</tr>
<tr>
<td>MT-287</td>
<td>67.22</td>
<td>64.54</td>
<td>67.37</td>
<td>66.13</td>
<td>69.66</td>
</tr>
<tr>
<td>MT-771</td>
<td>64.00</td>
<td>55.00</td>
<td>66.65</td>
<td>66.70</td>
<td>68.91</td>
</tr>
<tr>
<td>RW-2k</td>
<td>48.46</td>
<td>45.03</td>
<td>47.22</td>
<td>47.69</td>
<td>45.69</td>
</tr>
</tbody>
</table>

word similarity datasets.\(^6\) Like Lee and Chen (2017) and Athiwaratkun et al. (2018), we compute the word similarity based on senses by MaxSim (Reisinger and Mooney, 2010), which maximizes the cosine similarity over the combination of all sense pairs and does not require local contexts,

\[
\text{MaxSim}(w_1, w_2) = \max_{0 \leq s_1, s_2 \leq K} \cos(s_1^1, s_2^2). \tag{13}
\]

GASI-β has better correlation on three datasets, is competitive on the rest (Table 6), and remains competitive without scaling. GASI is better than MUSE, the other hard-attention multi-prototype model, on six datasets and worse on three. Our model can reproduce word similarities as well or better than existing models through our sense selection.\(^7\)

5.1. Word Similarity vs. Interpretability

Word similarity tasks (Section 5) and human evaluations (Section 4) are inconsistent. GASI, GASI-β and MUSE are all competitive in word similarity (Table 5 and Table 6), but only GASI-β also does well in the human evaluations (Table 2). Both GASI without scaling and MUSE fail to learn distinguishable senses and cannot disambiguate senses. High word similarities do not necessarily indicate “good” sense embeddings quality; our human evaluation—contextual word sense selection—is complementary.

6. Related Work: Representation, Evaluation

Schütze (1998) introduces context-group discrimination for senses and uses the centroid of context vectors as a sense representation. Other work induces senses by context clustering (Parundare and Pedersen, 2004) or probabilistic mixture models (Brody and Lapata, 2009). Reisinger and Mooney (2010) first introduce multiple sense-specific vectors for each word, inspiring other multi-prototype sense embedding models. Generally, to address polysemy in word embeddings, previous work trains on annotated sense corpora (Iacobacci et al., 2015, Gómez-Pérez and Denaux, 2019) or external sense inventories (Labutov and Lipson, 2013, Chen et al., 2014, Jauhar et al., 2015, Chen et al., 2015, Wu and Giles, 2015, Pilehvar and Colliner, 2016, Mancini et al., 2017); Rothe and Schütze (2017) extend word embeddings to lexical resources without training; others induce senses via multilingual parallel corpora (Guo et al., 2014, Šuster et al., 2016, Ettinger et al., 2016).

We contrast our GASI to unsupervised monolingual multi-prototype models along two dimensions: sense induction methodology and differentiability. Our focus is unsupervised induction because for interpretability to be useful, we assume that sense inventories and disambiguations are either unavailable or imperfect.

On the dimension of sense induction methodology, Huang et al. (2012) and Neelakantan et al. (2014) induce senses by context clustering; Tian et al. (2014) model a corpus-level sense distribution; Li and Jurafsky (2015) model the sense assignment as a Chinese Restaurant Process; Qiu et al. (2016) induce senses by minimizing an energy function on a context-depend network; Bartunov et al. (2016) model the sense assignment as a steak-breaking process; Nguyen et al. (2017) model the sense embeddings as a weighted combination of topic vectors with pre-computed weights by topic models; Athiwaratkun et al. (2018) model word representations as Gaussian Mixture embeddings where each Gaussian component captures different senses; Lee and Chen (2017) compute sense distribution by a separate set of sense induction vectors. The proposed GASI marginalizes the likelihood of contexts over senses and induces senses by local context vectors; the most similar sense selection module is a bilingual model (Šuster et al., 2016) except that it does not introduce lower bound for negative sampling but uses weighted embeddings, which results in mixed senses.

On the dimension of differentiability, most sense selection models are non-differentiable and discretely select senses, with two exceptions: Šuster et al. (2016) use weighted vectors over senses; Lee and Chen (2017) implement hard attention with RL to mitigate the non-differentiability. In contrast, GASI keeps full differentiability by reparameterization and approximates discrete sense sampling with the scaled Gumbel softmax.

However, the elephants in the room are BERT and ELMo. While there are specific applications where humans might be better served by multisense embeddings, computers seem to be consistently better served by contextual representations. A natural extension is to use the aggregate representations of word senses from these models. Particularly for ELMo, one could cluster individual mentions (Chang, 2019), but this is unsatisfying at first blush: it creates clusters more specific than senses. BERT is even more difficult: the transformer is a dense, rich representation, but only a small subset describes the meaning of individual words. Probing techniques (Perone et al., 2018) could help focus on semantic aspects that help humans understand word usage.
6.1. Granularity

Despite the confluence of goals, there has been a disappointing lack of cross-fertilization between the traditional knowledge-based lexical semantics community and the representation-learning community. Following the trends of sense learning models, we—from the perspective of those used to VerbNet or WordNet—use far too few senses per word. While there is disagreement about sense inventory, “hard” and “line” (Leacock et al., 1998) definitely have more than three senses. Expanding to granular senses presents both challenges and opportunities for future work.

While moving to a richer sense inventory is valuable future work, it makes human annotation more difficult (Erk et al., 2013)—while we can expect humans to agree on which of three senses are used, we cannot for larger sense inventories. In topic models, Chang et al. (2009) develop topic log odds (in addition to the more widely used model precision) to account for graded assignment to topics. Richer user models would need to capture these more difficult decisions.

However, moving to more granular senses requires richer modeling. Bayesian nonparametrics (Orbanz and Teh, 2010) can determine the number of clusters that best explain the data. Combining online stick breaking distributions (Wang et al., 2011) with GASI’s objective function could remove unneeded complexity for word types with few senses and consider the richer sense inventory for other words.

7. Conclusion

The goal of multi-sense word embeddings is not just to win word sense evaluation leaderboards. Rather, they should also describe language: given millions of tokens of a language, what are the patterns in the language that can help a lexicographer or linguist in day-to-day tasks like building dictionaries or understanding semantic drift. Our differentiable Gumbel Attention Sense Induction (GASI) offers comparable word similarities with multisense representations while also learning more distinguishable, interpretable senses.

However, simply asking whether word senses look good is only a first step. A sense induction model designed for human use should be closely integrated into that task. While we use a Word2Vec-based objective function in Section 2, ideally we should use a human-driven, task-specific metric (Feng and Boyd-Graber, 2019) to guide the selection of senses that are distinguishable, interpretable, and useful.

Acknowledgement

We thank the anonymous reviewers for their insightful and constructive comments. We thank Martha Palmer, Kevin Seppi, and the rest of the “closing the loop” team for useful discussions during the development of this research. This work was supported by NSF Grant 118-1409287. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsor.

8. Bibliographical References


Pelevina, M., Arefiyev, N., Biemann, C., and Panchenko, A.
(2016). Making sense of word embeddings. In Workshop on Representation Learning for NLP.
Appendix

Equation and figure numbers continue from main submission.

A. Derivation Desiderata

Like the Skip-Gram objective (Equation 2), we model the
likelihood of a context word given the center sense $P(c_j \mid s_k)$
using softmax,

$$P(c_j \mid s_k) = \frac{\exp \left( c_j^\top s_k \right)}{\sum_{j=1}^{|V|} \exp \left( c_j^\top s_k \right)}, \quad (14)$$

where the bold symbol $s_k$ is the vector representation
of sense $s_k$ from $S$, and $c_j$ is the context embedding of word
$c_j$ from $C$.

Computing the softmax over the vocabulary is time-
consuming. We want to adopt negative sampling to ap-
proximate $\log P(c_j \mid s_k)$, which does not exist explicitly in
our objective function (Equation 7).\(^8\)

However, given the concavity of the logarithm function, we
can apply Jensen’s inequality,

$$\log \left[ \sum_{k=1}^K P(c_j \mid s_k) P(s_k \mid w, \bar{c}) \right] \geq \sum_{k=1}^K P(s_k \mid w, \bar{c}) \log P(c_j \mid s_k), \quad (15)$$

and create a lower bound of the objective. Maximizing this
lower bound gives us a tractable objective, $J(S, C) \propto$

$$\sum_{w_i \in V} \sum_{c_j \in \Sigma} \sum_{k=1}^K P(s_k \mid w_i, \bar{c}) \log P(c_j \mid s_k), \quad (16)$$

where $\log P(c_j \mid s_k)$ is estimated by negative sam-
ping (Mikolov et al., 2013b),

$$\log \sigma(c_j^\top s_k) + \sum_{j=1}^n E_{c_j \sim P_c(c)}[\log \sigma(-c_j^\top s_k)]$$

B. Training Details

During training, we fix the window size to five and the
dimensionality of the embedding space to 300 for compari-
son to previous work. We initialize both sense and context
embeddings randomly within $U(-0.5/\text{dim}, 0.5/\text{dim})$ as in
Word2Vec. We set the initial learning rate to 0.01; it is de-
creased linearly until training concludes after 5 epochs. The
batch size is 512, and we use five negative samples per cen-
ter word-context pair as suggested by Mikolov et al. (2013a).
The subsample threshold is 1e-4. We train our model on the
GeForce GTX 1080 Ti, and our implementation (using
PyTorch 3.0) takes ~ 6 hours to train one epoch on the April
2010 Wikipedia snapshot (Shaoul and Westbury, 2010) with
100k vocabulary. For comparison, our implementation of
Skip-Gram on the same framework takes ~ 2 hours each
epoch.

C. Number of Senses

For simplicity and consistency with most of previous work,
we present our model with a fixed number of senses $K$.

C.1. Post-training Pruning and Retraining

For words that do not have multiple senses or have most
senses appear very low-frequently in corpus, our model
(as well as many previous models) learns duplicate senses.
Ideally, if we set a large number of $K$, with a perfect pruning
strategy, we can estimate the number of senses per type by
removing duplicated senses and retrain a new model with

\(^8\) Deriving the negative sampling requires the logarithm of a
softmax (Goldberg and Levy, 2014).
Figure 5: As the scale factor $\beta$ increases, the sense selection distribution for “bond” given examples from SemCor 3.0 for synset “bond.n.02” becomes flatter, indicating less disambiguated sense vectors.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MSSG-30K</th>
<th>MSSG-6K</th>
<th>MUSE_Boltzmann</th>
<th>SASI</th>
<th>GASI</th>
<th>GASI-$\beta$</th>
<th>PFT-GM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimLex-999</td>
<td>31.80</td>
<td>28.65</td>
<td>39.61</td>
<td>31.56</td>
<td>40.14</td>
<td>41.68</td>
<td>40.19</td>
</tr>
<tr>
<td>WS-353</td>
<td>65.69</td>
<td>67.42</td>
<td>68.41</td>
<td>58.31</td>
<td>68.49</td>
<td>69.36</td>
<td>68.6</td>
</tr>
<tr>
<td>MEN-3k</td>
<td>65.99</td>
<td>67.10</td>
<td>74.06</td>
<td>65.07</td>
<td>73.13</td>
<td>72.32</td>
<td>77.40</td>
</tr>
<tr>
<td>MC-30</td>
<td>67.79</td>
<td>76.02</td>
<td>81.80</td>
<td>70.81</td>
<td>82.47</td>
<td>85.27</td>
<td>74.63</td>
</tr>
<tr>
<td>RG-65</td>
<td>73.90</td>
<td>64.97</td>
<td>81.11</td>
<td>74.38</td>
<td>77.19</td>
<td>79.77</td>
<td>79.75</td>
</tr>
<tr>
<td>YP-130</td>
<td>40.69</td>
<td>42.68</td>
<td>43.56</td>
<td>48.28</td>
<td>49.82</td>
<td>56.34</td>
<td>59.39</td>
</tr>
<tr>
<td>MT-287</td>
<td>65.47</td>
<td>64.04</td>
<td>67.22</td>
<td>64.54</td>
<td>67.37</td>
<td>66.13</td>
<td>69.66</td>
</tr>
<tr>
<td>MT-771</td>
<td>61.26</td>
<td>58.83</td>
<td>64.00</td>
<td>55.00</td>
<td>66.65</td>
<td>66.70</td>
<td>68.91</td>
</tr>
<tr>
<td>RW-2k</td>
<td>42.87</td>
<td>39.24</td>
<td>48.46</td>
<td>45.03</td>
<td>47.22</td>
<td>47.69</td>
<td>45.69</td>
</tr>
</tbody>
</table>

Table 7: Spearman’s correlation $100\rho$ on non-contextual word similarity (MaxSim). GASI-$\beta$ outperforms the other models on three datasets and is competitive on others. PFT-GM is trained with two components/senses while other models learn three.

Table 8: Sense selection on Word in Context (WiC) dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>unsupervised multi-prototype models</td>
<td></td>
</tr>
<tr>
<td>MESS-30K</td>
<td>54.00</td>
</tr>
<tr>
<td>MUSE_Boltzmann</td>
<td>52.14</td>
</tr>
<tr>
<td>GASI-$\beta$</td>
<td>55.27</td>
</tr>
<tr>
<td>semi-supervised with lexical resources</td>
<td></td>
</tr>
<tr>
<td>DeConf</td>
<td>58.55</td>
</tr>
<tr>
<td>sw2v</td>
<td>54.56</td>
</tr>
</tbody>
</table>

the estimated number of senses instead of a fixed number $K$.

However, this is challenging (McCarthy et al., 2016): instead we use a simple pruning strategy and remove duplicated senses with a threshold $\lambda$. Specifically, for each word $w_i$, if the cosine distance between any of its sense embeddings ($s_m^i$, $s_n^i$) is smaller than $\lambda$, we consider them to be duplicates. After discovering all duplicate pairs, we start pruning with the sense $s_n^i$ that has the most duplication and keep pruning with the same strategy until no more duplicates remain.

**Model-specific pruning** We estimate a model-specific threshold $\lambda$ from the learned embeddings instead of deciding it arbitrary. We first sample 100 words from the negative sampling distribution over the vocabulary. Then, we retrieve the top-5 nearest neighbors (from all senses of all words) to each sense of each sampled word. If one of a word’s own senses appears as a nearest neighbor, we append the distance between them to a sense duplication list $D_{dup}$. For other nearest neighbors, we append their distances to the word neighbor list $D_{nn}$. After populating the two lists, we want to choose a threshold that would prune away all of the sense duplicates while differentiating sense duplication with other distinct neighbor words. Thus, we compute

$$\lambda = \frac{1}{2}(\text{mean}(D_{dup}) + \text{mean}(D_{nn})). \quad (17)$$

C.2. Number of Senses vs. Word Frequency

A common heuristic is that more frequent words have more senses. Figure 4 shows a histogram of the number of senses left for words ranked by their frequency, and the results agree with the assumption. Generally, the model learns more sense for high frequent words, except for the most frequent ones. The most frequent words are usually considered stopwords, such as “the”, “a” and “our”, which have only one common meaning. Moreover, we compare our model initialized with three senses (GASI-0.4, $K = 3$) against the one that has five (GASI-0.4, $K = 5$). Initializing with a larger number of senses, the model is able to uncover more senses for most words.

C.3. Duplicated Senses and Human-Model Agreement

We measure distinctness both by counting shared nearest neighbors and the average cosine similarities of GloVe embeddings. Specifically, MUSE learns duplicate senses for most words, preventing users from choosing appropriate senses and preventing human-model agreement. GASI-$\beta$ learns some duplicated senses and some distinguishable senses.
senses. MSSL appears to learn the fewest duplicate senses, but they are not distinguishable enough for humans. Users disagree with each other (0.33 agreement) even when the number of overlaps is very small (Figure 6). Table 4 shows an intuitive example. If we use rater agreement to measure how distinguishable the learned senses are to humans, GASI-β learns the most distinguishable senses.

The model is more likely to agree with humans when humans agree with each other (Figure 7), i.e., human-model consistency correlates with rater agreement (Figure 7). MSSL disagrees with humans more even when raters agree with each other, indicating worse sense selection ability.