PolyLM: Learning about Polysemy through Language Modeling

Alan Ansell[1,2] , Felipe Bravo-Marquez[3], Bernhard Pfahringer[2]
[1]Language Technology Lab, University of Cambridge
[2]Department of Computer Science, University of Waikato
[3]Department of Computer Science, University of Chile & IMFD
aja63@cam.ac.uk, fbravo@dcc.uchile.cl,
bernhard@waikato.ac.nz

Abstract

To avoid the "meaning conflation deficiency" of word embeddings, a number of models have aimed to embed individual word *senses*. These methods at one time performed well on tasks such as word sense induction (WSI), but they have since been overtaken by task-specific techniques which exploit contextualized embeddings. However, sense embeddings and contextualization need not be mutually exclusive. We introduce PolyLM, a method which formulates the task of learning sense embeddings as a language modeling problem, allowing contextualization techniques to be applied. PolyLM is based on two underlying assumptions about word senses: firstly, that the probability of a word occurring in a given context is equal to the sum of the probabilities of its individual senses occurring; and secondly, that for a given occurrence of a word, one of its senses tends to be much more plausible in the context than the others. We evaluate PolyLM on WSI, showing that it performs considerably better than previous sense embedding techniques, and matches the current state-of-the-art specialized WSI method despite having six times fewer parameters. Code and pre-trained models are available at https://github.com/AlanAnsell/PolyLM.

1 Introduction

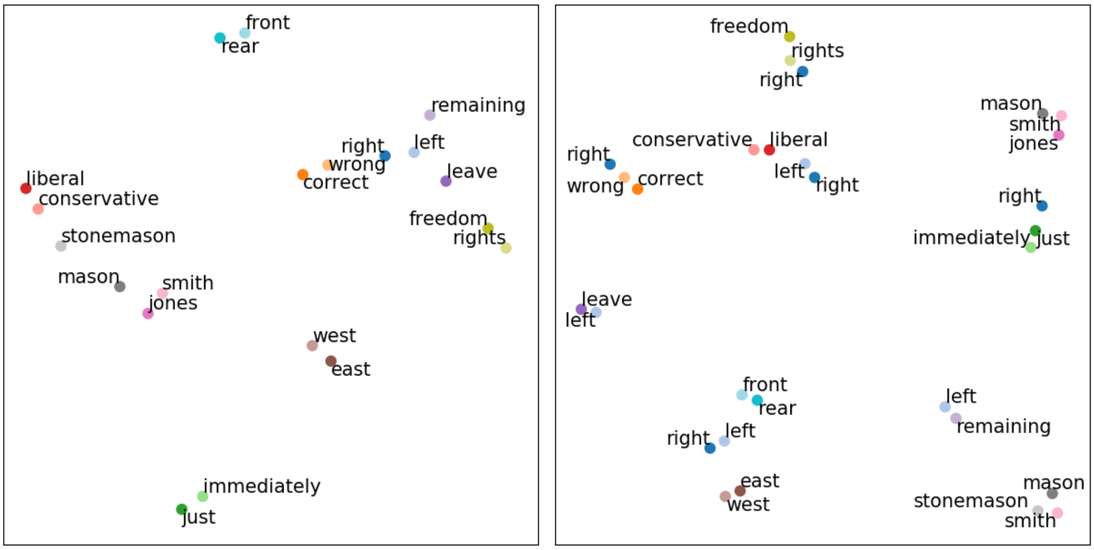
Much work in NLP has been dedicated to vector representations of words, but it has been recognized since as early as (Schütze, 1998) that such representations fail to capture the polysemous nature of many words, conflating their multiple senses into a single point in semantic space. There have been several attempts at embedding individual word senses to avoid this issue, termed the "meaning conflation deficiency" by Camacho-Collados and Pilehvar (2018) in their survey on the area.

We propose PolyLM, an unsupervised sense embedding model which is effective and easy to apply to downstream tasks. PolyLM can be thought of as both a (masked) language model and a *sense* model, as it calculates a probability distribution both over words and word senses at masked positions. The formulation is derived from two observations about word senses: firstly, that the probability of a word occurring in a given context is equal to the sum of the probabilities of its individual senses occurring; and secondly, that for a given occurrence of a word, one of its senses tends to be much more plausible in the context than the others.

There are several reasons for the interest in sense representations. The first is the downsides associated with the meaning conflation deficiency. Word embedding models can have difficulty distinguishing which sense of an ambiguous word applies in a given context (Yaghoobzadeh and Schütze, 2016). Additionally, homonymy and polysemy cause distortion in word embeddings: for instance, we would find the unrelated words *left* and *wrong* unreasonably close in the vector space due to their similarity to two different senses of the word *right*, an effect noted by Neelakantan et al. (2014) and illustrated in Figure 1. Intuitively we would expect that sense embedding models could gain superior semantic understanding by avoiding these problems.

In addition to well-established applications for sense representations such as word sense disambiguation (WSD) and induction (WSI), another interesting use case is the automatic construction of lexical resources (Neale, 2018). While there are existing human-curated word sense inventories for English such as such as WordNet (Miller, 1995), these are expensive to create and are unavailable for most languages. Panchenko (2016) showed that sense embeddings learned using the model of Bartunov et al. (2016) could be linked with word senses contained in BabelNet (Navigli and Ponzetto, 2012) with a reasonable degree of precision, although the mapping struggled with recall. PolyLM represents a significant advance over Bartunov et al.'s in terms of WSI performance, so

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 563–574
April 19 - 23, 2021. ©2021 Association for Computational Linguistics



(a) Word embeddings

(b) Sense embeddings

Figure 1: An illustration of the meaning conflation deficiency, showing selected word and sense embeddings learned by PolyLM visualized using t-SNE (Maaten and Hinton, 2008) and adjustText (Flyamer, 2017). Sense embeddings were learned by training PolyLM$_{\text{SMALL}}$ with the standard 8 senses per word; word embeddings were learned by training PolyLM$_{\text{SMALL}}$, but with a single sense per word. Note that both models were trained on unlemmatized data, unlike those used in the WSI experiments. The occurrence of closely related polysemous words nearby in the word embedding space (i.e. *left* and *right*) causes unrelated words to be closer together (e.g. *left* and *wrong*) and related words to be further apart (e.g. *right* and *east*) than they otherwise would be. The use of sense embeddings avoids such distortion. PolyLM is capable of detecting comparatively rare word senses, such as the political senses of *left* and *right*, and the use of *smith* and *mason* to refer to tradespeople.

it seems reasonable to imagine that this approach to lexical resource construction might now be more feasible.

The emergence of contextualized models such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) has had a tremendous impact on the area of semantic representation. Rather than representing words using a single embedding, or even a set of sense embeddings, these models allow words to be represented using an infinite set of possible embeddings depending on the context. This approach has been very effective across NLP and many state-of-the-art systems incorporate contextualized models, including systems for WSD and WSI. The success of contextualized models raises the question of whether there is still value in learning discrete sense representations.

However, contextualized models still rely on word embeddings, and are therefore subject to the meaning conflation deficiency. Furthermore it could be argued that it is inefficient to have the same representation size for all words regardless of how diverse their range of senses is. Another drawback is that before they can be applied to word sense-related tasks, an adaptation step such as clustering to induce discrete senses or fine-tuning is generally required, which is often expensive in terms of both research and compute time.

The contributions of this paper can be summarized as follows:

- We propose PolyLM, an end-to-end, unsupervised neural sense embedding model derived from two simple assumptions about word senses. We demonstrate that PolyLM learns senses which correspond well to human notions by showing that it performs well at WSI.

- PolyLM is flexible in that it can use any "contextualizer" (a useful term coined by Liu et al. (2019)), so it will remain relevant as contextualization techniques improve.

- We reduce the effect of the meaning conflation deficiency by disambiguating word senses at the input with a neural "disambiguation layer." We show that good performance on WSI can be achieved using the output of this layer alone, suggesting that it could be a useful component in many neural networks for language understanding.

2 Related Work

One of the first works in unsupervised learning of sense representations was by Schütze (1998), who proposed a two-step process, where vector representations are first derived for each context containing an ambiguous word, and these are then clustered into a pre-defined number of groups. Huang et al. (2012) added a third step, where after sense-labeling each word according to its context cluster, sense representations are learned through neural language modeling.

A number of later approaches employed a joint training approach, where sense labeling and sense representation learning happen in parallel. Neelakantan et al. (2014), Li and Jurafsky (2015) and Bartunov et al. (2016) each proposed multi-sense variants of the Skip-Gram model (Mikolov et al., 2013). Various approaches were tried for determining the number of senses per word: for instance, Li and Jurafsky and Bartunov et al. used Chinese Restaurant Processes and Dirichlet Processes respectively to automatically learn an appropriate number of senses for each word.

Many joint training approaches have the disadvantage that they create ambiguity in the context representation by representing context words with word embeddings in order to avoid considering the exponential number of possible sense labelings for the context. Qiu et al. (2016) and Lee and Chen (2017) propose purely sense-based approaches which can sense-label the input efficiently.

Arora et al. (2018) took a novel approach to the problem of learning word senses, demonstrating that the embedding learned by traditional techniques for an ambiguous word tends to be very close to a linear combination of the hypothetical vectors corresponding to its individual senses. They proposed a method for recovering the underlying sense vectors and coefficients, and evaluated their system on WSI.

Since the emergence of contextualized models, there have been a number of other systems which have exploited their powerful semantic representations for specific tasks such as word sense disambiguation (Huang et al., 2019; Vial et al., 2019) and induction (Amrami and Goldberg, 2018, 2019), however none of these methods creates explicit sense embeddings.

3 PolyLM

3.1 Overview

Consider a typical neural language model. Each word w in a vocabulary V is assigned a single embedding, resulting in an embedding matrix $M \in \mathbb{R}^{|V| \times d}$, where d is the embedding dimensionality. The probability of w occurring in a context c is estimated as

$$\mathbb{P}(w \mid c) = \big[\text{softmax}\big(M\boldsymbol{y}(c) + \boldsymbol{a}\big)\big]_w, \quad (1)$$

where $\boldsymbol{y}(c) \in \mathbb{R}^d$ is a vector representation of c and $\boldsymbol{a} \in \mathbb{R}^{|V|}$ is a trainable bias vector. In BERT (Devlin et al., 2019) for instance, $\boldsymbol{y}(c)$ corresponds to the final output of multiple Transformer encoder layers (Vaswani et al., 2017).

Now suppose that for each $w \in V$, there is a corresponding set S_w of *sememes*, or senses which w can have. For instance, intuitively we might have $S_{\text{rock}} = \{\text{rock:stone, rock:musical genre, rock:shake}\}$. We assume that the S_w are disjoint, i.e. $S_w \cap S_{w'} = \emptyset$ whenever $w \neq w'$, and we define the full sense inventory $S = \bigcup_{w \in V} S_w$.

Context induces specific senses for the words it contains. Thus a passage of text can be thought of as a sequence of sememes as well as a sequence of words. The first observation underlying PolyLM is that the probability of a word w occurring in a context c is equal to the sum of the probabilities of w's component sememes occurring in the context, i.e.

$$\mathbb{P}(w \mid c) = \sum_{s \in S_w} \mathbb{P}(s \mid c). \quad (2)$$

We wish to learn representations for individual senses, and so we assign an embedding to each sememe in our sense inventory, resulting in a matrix E with dimension $|S| \times d$ and bias vector $\boldsymbol{b}$ of dimension $|S|$. Note that this assumes that we know the number of senses of each word *a priori*, an assumption whose consequences we discuss later. Following Eq. 1, we define the vector $\boldsymbol{p}(c) \in \mathbb{R}^{|S|}$ of sememe probabilities in a context c as

$$\boldsymbol{p}(c) = \text{softmax}\big(E\boldsymbol{x}(c) + \boldsymbol{b}\big). \quad (3)$$

Considering Eq. 2, we have

$$\mathbb{P}(w \mid c) = \sum_{s \in S_w} p(c)_s, \quad (4)$$

allowing us to formulate the problem of learning sense representations with a language modeling objective.

PolyLM is constructed from three components: the input layer, which represents the input tokens as aggregates of their sense embeddings, the disambiguation layer, which attempts to determine the contextually appropriate sense embeddings for the input, and the prediction layer, which implements the language modeling objective.

We adopt the masked language modeling (MLM) task used for training BERT. When training, we select a subset $T \subset \{1, 2, ..., n\}$ of the tokens in the input sequence as targets for prediction, and produce a masked version $c' = w'_1, w'_2, ..., w'_n$ of the original sequence $c = w_1, w_2, ..., w_n$ as follows: 15% of tokens are chosen at random as targets, of which 80% are replaced with a special [MASK] token, 10% are replaced with a random token, and 10% are left unchanged.

3.2 Input Layer

We define a contextualizer to be a function which maps a sequence of input representations $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n \in \mathbb{R}^d$ to a corresponding sequence of output representations $\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n \in \mathbb{R}^d$. Recurrent Neural Networks and Transformer architectures are both commonly used as contextualizers for language modeling. Typically the input representations are drawn from an embedding matrix $I \in \mathbb{R}^{|V| \times d}$. It has become common (e.g. BERT) to set I equal to O, the embedding matrix used at the language modeling output, as recommended by Press and Wolf (2017), and thus have a single embedding matrix E.

The issue of input representation poses a problem for our model. Our output embeddings $E \in \mathbb{R}^{|S| \times d}$ correspond to sememes. We cannot straightforwardly tie our input and output embeddings as Press and Wolf suggest, because we receive words rather than sememes as input. We solve this problem by setting the input representation of a word to be a convex combination of the representations of its sememes, i.e.

$$\boldsymbol{x}(w) = \sum_{s \in S_w} \lambda_{ws} \boldsymbol{e}_s, \tag{5}$$

where $\boldsymbol{e}_s$ is the row of E corresponding to sememe s, and $\boldsymbol{\lambda}_w$ is a learnable weight vector with the properties that $\sum_{s \in S_w} \lambda_{ws} = 1$ and $\boldsymbol{\lambda}_w \geq \boldsymbol{0}$ (in practice, $\boldsymbol{\lambda}_w$ is the softmax of an underlying, unconstrained variable vector).

3.3 Disambiguation Layer

The disambiguation layer attempts to infer the contextually appropriate sememe embeddings for the input based on the conflated representations from the input layer.

Representations $\boldsymbol{x}(w'_1), \boldsymbol{x}(w'_2), ..., \boldsymbol{x}(w'_n)$ of c', calculated according to Eq. 5, are fed into a contextualizer instance C^D, which outputs representations $\boldsymbol{y}^D_1(c'), \boldsymbol{y}^D_2(c'), ..., \boldsymbol{y}^D_n(c')$. We use these representations to calculate a probability distribution over each sense of the tokens in the input:

$$\boldsymbol{q}^D_i(c') = \mathrm{softmax}\big(E^{(w'_i)} \boldsymbol{y}^D_i(c') + \boldsymbol{b}^{(w'_i)}\big), \tag{6}$$

where $E^{(w'_i)}$ is a submatrix of E containing only the rows corresponding to senses of token w'_i, and similarly $\boldsymbol{b}^{(w'_i)}$ is a subvector of a learnable bias vector $\boldsymbol{b} \in \mathbb{R}^{|S|}$. In other terms,

$$q^D_{is}(c') = \frac{e^{\boldsymbol{e}_s^\top \boldsymbol{y}^D_i(c') + b_s}}{\sum\limits_{s' \in S_{w'_i}} e^{\boldsymbol{e}_{s'}^\top \boldsymbol{y}^D_i(c') + b_{s'}}}, \tag{7}$$

where $s \in S_{w'_i}$. $q^D_{is}(c')$ corresponds to the probability that the ith token in sequence c' has sense s.

The disambiguated representation of a token could simply be its highest-probability sememe embedding in the context, but to allow gradients to flow through the disambiguation layer, we take the sum of the sememe embeddings weighted by their probabilities:

$$\boldsymbol{x}^P_i(c') = \sum_{s \in S_{w'_i}} q^D_{is}(c') \boldsymbol{e}_s. \tag{8}$$

3.4 Prediction Layer

The prediction layer maps a sequence of disambiguated input representations onto a corresponding set of output representations, and from each output representation estimates the probability of every sememe in the sense inventory occurring at the corresponding position of the sequence.

Disambiguated representations $\boldsymbol{x}^P_1(c'), \boldsymbol{x}^P_2(c'), ..., \boldsymbol{x}^P_n(c')$ are fed into another contextualizer instance C^P, which returns output representations $\boldsymbol{y}^P_1(c'), \boldsymbol{y}^P_2(c'), ..., \boldsymbol{y}^P_n(c')$. These are used to calculate a probability distribution over the entire sense inventory, as prescribed by Eq. 3:

$$\boldsymbol{p}_i(c') = \mathrm{softmax}(E \boldsymbol{y}^P_i(c') + \boldsymbol{b}). \tag{9}$$

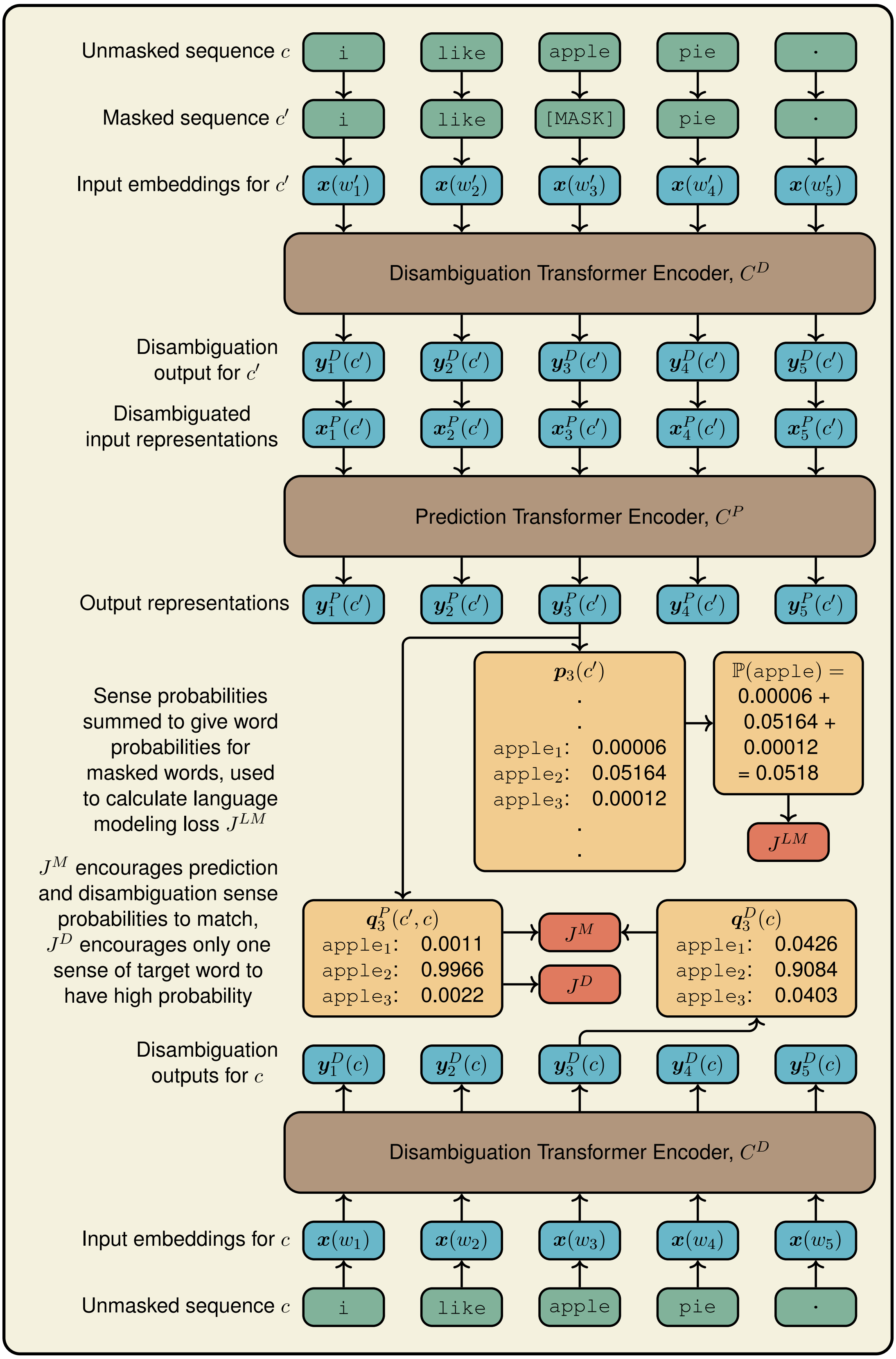


Figure 2: Architecture diagram for PolyLM when training, illustrated on the sentence "I like apple pie.", where the word "apple" is chosen as a target and masked (note that "apple" is ambiguous when tokens are lower-cased, as it may refer to a fruit or a technology company). At inference time, the bottom components (up to and including $\boldsymbol{q}^D(c)$) do not need to be evaluated, and the sequence may not be masked at the input.

We define an additional set of probabilities $\boldsymbol{q}^P$ analogous to $\boldsymbol{q}^D$ defined in Eq. 6:

$$\boldsymbol{q}_i^P(c', c) = \text{softmax}\big(E^{(w_i)}\boldsymbol{y}_i^P(c') + \boldsymbol{b}^{(w_i)}\big). \tag{10}$$

$\boldsymbol{q}_i^P$ takes both c' and the unmasked sequence c as arguments because we are interested in the sense probabilities of the words w_i that actually occurred. $\boldsymbol{q}_i^P$ will be used later for defining the loss function and is useful for downstream tasks.

3.5 Loss Function

We seek to minimize a loss function J with three components, each of which is explained below:

$$\begin{aligned} J(c, c', T) = J^{LM}(c, c', T) &+ \\ J^D(c, c', T) &+ \\ J^M(c, c', T) & \end{aligned} \tag{11}$$

3.5.1 Language Modeling Loss

The language modeling loss J^{LM} is defined as the mean negative log likelihood of the target tokens occurring:

$$\begin{aligned} &J^{LM}(c, c', T) \\ &= \frac{-1}{|T|}\sum_{i\in T}\log\hat{\mathbb{P}}(w_i \mid c') \qquad (12) \\ &= \frac{-1}{|T|}\sum_{i\in T}\log\sum_{s\in S_{w_i}}\hat{\mathbb{P}}(\text{sememe } i \text{ is } s \mid c') \qquad (13) \\ &= \frac{-1}{|T|}\sum_{i\in T}\log\sum_{s\in S_{w_i}} p_{is}(c'), \qquad (14) \end{aligned}$$

where $\boldsymbol{p}_i$ is as defined in Eq. 9.

3.5.2 Distinctness Loss

Recall that we assume in advance a number of senses for each word. In practice we guess a relatively high number to avoid missing senses. When we overestimate the number of senses, we find that two different sense embeddings for a word converge to essentially the same meaning. The aim of the distinctness loss is to ensure that each sense has a distinct meaning, and to "kill off" superfluous senses by causing them to have very low probability in all contexts.

The second key observation of PolyLM is that if the sememes corresponding to a word w are distinct, then in contexts where w occurs, we would expect one of these sememes to have a high estimated probability of occurring, and the rest to have a low probability. The distinctness loss, given by,

$$J^D(c, c', T) = \frac{-1}{r|T|}\sum_{i\in T}\log\sum_{s\in S_{w_i}}\big(q_{is}^P(c', c)\big)^r, \tag{15}$$

with hyperparameter $r > 1$, encourages this separation to occur. A full justification is given in Appendix A.

3.5.3 Match Loss

Without extra supervision, the disambiguation layer tends to very quickly allocate almost all of the probability mass for a word to a single one of its senses. This appears to be due to a "rich get richer" effect in Eq. 8, where the sense embedding with the highest weight has larger gradients associated with it.

A more reliable source of sense probabilities is the output of the prediction layer, as this is more closely associated with the ground truth. Therefore we encourage the disambiguation sense probabilities $\boldsymbol{q}^D$ to be similar to the prediction sense probabilities $\boldsymbol{q}^P$ by adding a sense probability "match loss," which is proportional to the cosine similarity between $\boldsymbol{q}^D$ and $\boldsymbol{q}^P$.

Because $\boldsymbol{q}_i^D(c')$ is meaningless when token i is replaced with [MASK], when calculating the match loss we evaluate the disambiguation layer on the unmasked sequence (shown with bottom-up arrows in Figure 2), obtaining $\boldsymbol{q}_i^D(c)$. The match loss is defined as

$$J^M(c, c', T) = \frac{-\lambda^M}{|T|}\sum_{i\in T}\frac{\boldsymbol{q}_i^D \cdot \boldsymbol{q}_i^P}{\|\boldsymbol{q}_i^D\|\|\boldsymbol{q}_i^P\|}, \tag{16}$$

where $\boldsymbol{q}_i^D$ and $\boldsymbol{q}_i^P$ are shorthand for $\boldsymbol{q}_i^D(c)$ and $\boldsymbol{q}_i^P(c', c)$ respectively, and λ^M is a hyperparameter.

As we wish the disambiguation layer to learn from the prediction layer rather than the other way around, we do not allow gradients from the match loss to propagate through $\boldsymbol{q}_i^P$.

3.6 Details and Parameters

3.6.1 Preprocessing

To avoid the issue of how to represent a word's sense when it is broken into sub-word level tokens, our vocabulary consists of whole-word tokens. However the WSI tasks on which we evaluate our model operate on the lemma level, so we lemmatize our training corpus as described in Appendix B. The vocabulary consists of the ~86K tokens appearing more than 500 times in our training corpus, which like BERT's consists of English Wikipedia + BookCorpus (Zhu et al., 2015). All tokens are lower-cased.

3.6.2 Contextualizers

One of the advantages of PolyLM is that it can be used with any type of contextualizer - note however that we must train our contextualizers together with the rest of the model rather than using pretrained contextualizer instances, because their word embedding matrix would not match our sense embedding matrix. In this paper we present results where the disambiguation and prediction contextualizers C^D and C^P use BERT's implementation of the Transformer encoder architecture.

3.6.3 Parameters

To keep the total number of embeddings reasonable, we allow only the ~10,000 tokens which occur more than 20,000 times in the training corpus, or appear as focuses in the evaluation datasets, to have multiple senses. Specifically, we assign these tokens a fixed number of $k = 8$ embeddings, and other tokens a single embedding. Since according to Zipf's law (Zipf, 1950), it is the most frequent words which tend to have the most senses, we expect not to miss too many senses by assuming that infrequent words are monosemous. We leave the investigation of more sophisticated methods for pre-allocating or dynamically updating the number of senses for each token for future work.

We train two PolyLM models of different sizes, PolyLM$_{\text{SMALL}}$ and PolyLM$_{\text{BASE}}$. Due to the prohibitive computational cost of training a model of BERT$_{\text{LARGE}}$'s size, we use significantly smaller dimensions, as shown in Table 1.

Models were trained over 6,000,000 batches consisting of 32 sequences of length 128 using the Adam optimizer (Kingma and Ba, 2014). The learning rate was increased linearly from 0 to 3e-5 over the first 10,000 batches, and then reduced linearly back to zero over the remaining batches. The hyperparameters λ^M and r specific to PolyLM's loss function were first increased linearly and then left constant, λ^M from 0 to 0.1 over the first 1,000,000 batches, and r from 1.0 to 1.5 over the first 2,000,000 batches.

It is important for r to be gradually increased in this manner because if r is large initially, then the effect of the distinctness loss reduces the diversity of the senses learned. On the other hand, increasing r too slowly seems to be detrimental to the senses' distinctness.

4 Experiments

Word sense induction (WSI) is the task of inferring the senses of a word in an unsupervised manner. This is precisely the aim of our method, and so is an ideal test task. We evaluate PolyLM on two WSI datasets, SemEval-2010 Task 14 (Manandhar et al., 2010) and SemEval-2013 Task 13 (Jurgens and Klapaftis, 2013). Both datasets consist of passages containing one of a set of polysemous focus words. The occurrences of the focus words in the test set have been sense-labeled by human annotators according to a reference sense inventory.

In the SemEval-2010 dataset, each instance is labeled with a single sense, whereas in the SemEval-2013 dataset an instance may be labeled with several relevant senses, each with a corresponding weight denoting its degree of applicability in the context.

Performance on SemEval-2010 is measured using paired F-Score (F-S) and V-Measure (V-M), and on SemEval-2013 using Fuzzy B-Cubed (FBC) and Fuzzy Normalized Mutual Information (FNMI). Overall performance on each task (**AVG**) is typically defined as the geometric mean of its two sub-metrics.

Currently, the best performing system on both datasets is that of Amrami and Goldberg (2019). Their system uses the idea of *substitute vectors*, first devised by Başkaya et al. (2013). For each instance, a set of most likely words that could have occurred instead of the focus word is obtained from the output of a language model. These sets are then clustered, and each cluster is taken to correspond to a different sense of the focus word. Amrami and Goldberg use BERT$_{\text{LARGE}}$ as their language model.

PolyLM can be used for WSI without any further training. For the SemEval-2010 dataset, each instance c is labeled with the sense of the focus word w_i which has the highest predicted probability, i.e. $\text{argmax}_{s \in S_{w_i}} \boldsymbol{q}^P_{is}(c', c)$, where c' is formed from c by replacing w_i with [MASK]. For SemEval-2013, we consider a sense applicable if it has a predicted probability $q^P_{is}(c', c) > p^{\text{thresh}}$, and the weight assigned to each applicable sense is its probability $q^P_{is}(c', c)$. We arbitrarily set p^{thresh} to 0.2.

Model	d	Filter size	No. attn. heads	No. layers	Seq. len.	Vocab size	No. embeddings	Total params
PolyLM$_{\text{SMALL}}$	128	512	8	4 (C^D), 8 (C^P)	128	86K	157K	24M
PolyLM$_{\text{BASE}}$	256	1024	8	4 (C^D), 12 (C^P)	128	86K	157K	54M
BERT$_{\text{LARGE}}$	1024	4096	16	24	512	30K	30K	340M

Table 1: Parameters of PolyLM and BERT$_{\text{LARGE}}$.

System	Version	SemEval-2010			SemEval-2013		
		F-S	V-M	AVG	FBC	FNMI	AVG
Amrami and Goldberg (2019)	BERT$_{\text{LARGE}}$	**71.3**	40.4	**53.6**	64.0	21.4	37.0
AutoSense (Amplayo et al., 2019)		62.9	10.1	25.2	61.7	8.0	22.2
PolyLM†	BASE	65.8	**40.5**	51.6	**64.8**	**23.0**	**38.6**
	SMALL	65.6	35.7	48.4	64.5	18.5	34.5
Qiu et al. (2016)†		-	-	-	56.9	6.7	19.5
SE-WSI-fix-cmp (Song et al., 2016)†		54.3	16.3	29.8	-	-	-
AdaGram (Bartunov et al., 2016)†		43.9	20.0	29.6	13.2	8.9	10.8
Arora et al. (2018)†	$k = 5$	46.4	11.5	23.1	-	-	-

Table 2: Comparison of sense embedding models and WSI-specific techniques on the SemEval 2010 and 2013 WSI tasks. SE-WSI-fix-cmp is based on Neelakantan et al. (2014)'s MSSG model. † - models which obtain explicit sense embeddings.

Description	SemEval-2010			SemEval-2013		
	F-S	V-M	AVG	FBC	FNMI	AVG
PolyLM$_{\text{SMALL}}$	**65.6**	**35.7**	**48.4**	**64.5**	**18.5**	**34.5**
No distinctness loss	53.5	33.4	42.3	57.4	16.3	30.5
No disambiguation layer	64.9	25.5	40.6	**64.5**	17.5	33.6
Disambiguation layer only	63.6	29.3	43.2	62.7	15.7	31.4

Table 3: PolyLM ablation study.

Results are shown in Table 2. Both PolyLM models comprehensively outperform previous sense embedding methods. PolyLM$_{\text{BASE}}$ and Amrami and Goldberg's system slightly outperform each other on one dataset each, suggesting similar overall proficiency at WSI. However it is worth noting that the BERT$_{\text{LARGE}}$ language model used by Amrami and Goldberg has more than six times as many parameters as PolyLM$_{\text{BASE}}$ and is much more computationally expensive to train and run.

PolyLM scales well for the sizes tested, with PolyLM$_{\text{BASE}}$ outperforming PolyLM$_{\text{SMALL}}$ by 3.2 and 4.1 points in **AVG** score on the two datasets with a 2.25x increase in the number of parameters. Even if further increases in model dimensions yielded much smaller improvements in performance, it seems likely that a PolyLM model of BERT$_{\text{LARGE}}$'s 340 million parameter size would achieve results significantly better than those of Amrami and Goldberg (2019).

4.1 Ablation Study

We test three alternative configurations against PolyLM$_{\text{SMALL}}$: one where the distinctness loss term is removed from the objective ("no distinctness loss"), one where the disambiguation layer is removed ("no disambiguation layer"), and one where the disambiguation sense probabilities $\boldsymbol{q}^D$ are used in place of $\boldsymbol{q}^P$ when performing WSI ("disambiguation layer only"). Note that the first two configurations require new models to be trained, whereas the last simply uses PolyLM$_{\text{SMALL}}$ in a different way. Results are shown in Table 3.

The use of the distinctness loss has a big impact on model performance, while the disambiguation layer is somewhat less important but still useful. The model still performs surprisingly well when the disambiguation rather than the prediction sense probabilities are used; these are the output of only four Transformer layers and hence are much cheaper to compute. This suggests that it might be practical to add the disambiguation layer at the input of various neural NLP models to improve their understanding of polysemy.

5 Conclusions

PolyLM is a novel model of polysemy based on two assumptions about word senses: firstly, that the probability of a word occurring in a context is equal to the sum of its individual senses occurring, as expressed by the language modeling loss; and secondly, that generally only one sense of a word ought to have a high probability of occurring in a given context, as expressed by the distinctness loss. PolyLM does indeed learn word senses which correspond well to human notions, as demonstrated by its performance on word sense induction, which matches that of the previous state-of-the-art system despite having 6 times fewer parameters. It can be easily applied to many word-sense related tasks, as it generates a probability distribution over the senses of each word in the input text. It is not specific to any one contextualizer and so can be improved as contextualizers improve.

Acknowledgements

Felipe Bravo-Marquez was funded by ANID FONDECYT grant 11200290, U-Inicia VID Project UI-004/20 and ANID - Millennium Science Initiative Program - Code ICN17_002.

References

Reinald Kim Amplayo, Seung-won Hwang, and Min Song. 2019. AutoSense model for word sense induction. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6212–6219.

Asaf Amrami and Yoav Goldberg. 2018. Word sense induction with neural biLM and symmetric patterns. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4860–4867, Brussels, Belgium. Association for Computational Linguistics.

Asaf Amrami and Yoav Goldberg. 2019. Towards better substitution-based word sense induction.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495.

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. 2016. Breaking sticks and ambiguities with adaptive skip-gram. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 130–138, Cadiz, Spain. PMLR.

Osman Başkaya, Enis Sert, Volkan Cirik, and Deniz Yuret. 2013. AI-KU: Using substitute vectors and co-occurrence modeling for word sense induction and disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 300–306, Atlanta, Georgia, USA. Association for Computational Linguistics.

Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63:743–788.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Ilya Flyamer. 2017. adjustText.

Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea. Association for Computational Linguistics.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3507–3512, Hong Kong, China. Association for Computational Linguistics.

David Jurgens and Ioannis Klapaftis. 2013. SemEval-2013 task 13: Word sense induction for graded and non-graded senses. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 290–299, Atlanta, Georgia, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Guang-He Lee and Yun-Nung Chen. 2017. MUSE: Modularizing unsupervised sense embeddings. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 327–337, Copenhagen, Denmark. Association for Computational Linguistics.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1722–1732, Lisbon, Portugal. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605.

Suresh Manandhar, Ioannis Klapaftis, Dmitriy Dligach, and Sameer Pradhan. 2010. SemEval-2010 task 14: Word sense induction &disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 63–68, Uppsala, Sweden. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

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

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Steven Neale. 2018. A survey on automatically-constructed WordNets and their evaluation: Lexical and word embedding-based approaches. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar. Association for Computational Linguistics.

Alexander Panchenko. 2016. Best of both worlds: Making word sense embeddings interpretable. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2649–2655, Portorož, Slovenia. European Language Resources Association (ELRA).

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Lin Qiu, Kewei Tu, and Yong Yu. 2016. Context-dependent sense embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 183–191, Austin, Texas. Association for Computational Linguistics.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Linfeng Song, Zhiguo Wang, Haitao Mi, and Daniel Gildea. 2016. Sense embedding learning for word sense induction. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 85–90, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic knowledge of WordNet for neural word sense disambiguation. In *Proceedings of the Tenth Global Wordnet Conference*, pages 108–117, Poland.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2016. Intrinsic subspace evaluation of word embedding representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 236–246, Berlin, Germany. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 19–27, Washington, DC, USA. IEEE Computer Society.

George K. Zipf. 1950. Human behavior and the principle of least effort. *Journal of Clinical Psychology*, 6(3):306–306.

A Justification of the Distinctness Loss

Consider the derivative of the language modeling loss for one particular target position $i \in T$ with respect to the pre-softmax scores $\boldsymbol{e}_k^\top \boldsymbol{y}_i^P + b_k$ of the target word w_i's sense embeddings $k \in S_{w_i}$. For brevity, we define $y_k = \boldsymbol{e}_k^\top \boldsymbol{y}_i^P + b_k$.

$$-\frac{\partial}{\partial y_k} J^{LM}(c, c', \{i\})$$
$$= \frac{\partial}{\partial y_k} \log \sum_{s \in S_{w_i}} \left[\text{softmax}(E\boldsymbol{y}_i^P + \boldsymbol{b})\right]_s$$
$$= \frac{\partial}{\partial y_k} \log \sum_{s \in S_{w_i}} \frac{e^{y_s}}{\sum_{s' \in S} e^{y_{s'}}}$$
$$= \frac{\partial}{\partial y_k} \log \frac{\sum_{s \in S_{w_i}} e^{y_s}}{\sum_{s \in S_{w_i}} e^{y_s} + \sum_{s \in S \setminus S_{w_i}} e^{y_s}}$$
$$= \frac{\partial}{\partial y_k} \log \frac{\sum_{s \in S_{w_i}} e^{y_s}}{\sum_{s \in S_{w_i}} e^{y_s} + C},$$

where $C = \sum_{s \in S \setminus S_{w_i}} e^{y_s}$,

$$= \frac{\partial}{\partial y_k} \left(\log \sum_{s \in S_{w_i}} e^{y_s} - \log \left(\sum_{s \in S_{w_i}} e^{y_s} + C \right) \right)$$
$$= \frac{\frac{\partial}{\partial y_k} \sum_{s \in S_{w_i}} e^{y_s}}{\sum_{s \in S_{w_i}} e^{y_s}} - \frac{\frac{\partial}{\partial y_k} (\sum_{s \in S_{w_i}} e^{y_s} + C)}{\sum_{s \in S_{w_i}} e^{y_s} + C}$$
$$= \frac{e^{y_k}}{\sum_{s \in S_{w_i}} e^{y_s}} - \frac{e^{y_k}}{\sum_{s \in S_{w_i}} e^{y_s} + C}$$
$$= \frac{e^{y_k}}{\sum_{s \in S_{w_i}} e^{y_s}} - \frac{e^{y_k}}{\sum_{s \in S} e^{y_s}}$$
$$= q_{ik}^P(c', c) - p_{ik}(c).$$

Since $q_{ik}^P > p_{ik}$, $\frac{\partial}{\partial y_k} J^{LM}(c, c', \{i\})$ will always be negative, meaning that every sense embedding for the target word will always move towards the contextualized representation $\boldsymbol{y}_i^P$. This is undesirable, because it means that even senses which are irrelevant in a context will receive a positive update.

Now consider the derivatives of the distinctness loss:

$$-\frac{\partial}{\partial x_y} J^D(c, c', \{i\})$$
$$= \frac{\partial}{\partial y_k} \frac{1}{r} \log \sum_{s \in S_{w_i}} \left(q_{is}^P(c', c) \right)^r$$
$$= \frac{1}{r} \frac{\partial}{\partial y_k} \log \sum_{s \in S_{w_i}} \left(\frac{e^{y_s}}{\sum_{s' \in S_{w_i}} e^{y_{s'}}} \right)^r$$
$$= \frac{1}{r} \frac{\partial}{\partial y_k} \log \sum_{s \in S_{w_i}} \frac{e^{r y_s}}{(\sum_{s' \in S_{w_i}} e^{y_{s'}})^r}$$
$$= \frac{1}{r} \frac{\partial}{\partial y_k} \log \frac{\sum_{s \in S_{w_i}} e^{r y_s}}{(\sum_{s \in S_{w_i}} e^{y_s})^r}$$
$$= \frac{1}{r} \frac{\partial}{\partial y_k} \left(\log \sum_{s \in S_{w_i}} e^{r y_s} - \log \left(\sum_{s \in S_{w_i}} e^{y_s} \right)^r \right)$$
$$= \frac{1}{r} \left(\frac{\frac{\partial}{\partial y_k} \sum_{s \in S_{w_i}} e^{r y_s}}{\sum_{s \in S_{w_i}} e^{r y_s}} - \frac{r \frac{\partial}{\partial y_k} \sum_{s \in S_{w_i}} e^{y_s}}{\sum_{s \in S_{w_i}} e^{y_s}} \right)$$
$$= \frac{1}{r} \left(\frac{r e^{r y_k}}{\sum_{s \in S_{w_i}} e^{r y_s}} - \frac{r e^{y_k}}{\sum_{s \in S_{w_i}} e^{y_s}} \right)$$
$$= \frac{e^{r y_k}}{\sum_{s \in S_{w_i}} e^{r y_s}} - q_{ik}^P(c', c).$$

When $r > 1$, $\frac{e^{r y_k}}{\sum_{s \in S_{w_i}} e^{r y_s}}$ is a "sharpened" version of $q_{ik}^P(c', c)$: it is larger than q_{ik}^P when q_{ik}^P is large, and smaller when q_{ik}^P is small.

Now we have

$$-\frac{\partial}{\partial y_k} \left(J^{LM}(c, c', \{i\}) + J^D(c, c', \{i\}) \right)$$
$$= -\frac{\partial}{\partial y_k} J^{LM}(c, c', \{i\}) - \frac{\partial}{\partial y_k} J^D(c, c', \{i\})$$
$$= q_{ik}^P(c', c) - p_{ik}(c) + \frac{e^{r y_k}}{\sum_{s \in S_{w_i}} e^{r y_s}} - q_{ik}^P(c', c)$$
$$= \frac{e^{r y_k}}{\sum_{s \in S_{w_i}} e^{r y_s}} - p_{ik}(c)$$
$$= q_{ik}^{\text{sharp}}(c', c) - p_{ik}(c).$$

Thus the addition of the distinctness loss results in even stronger reinforcement for senses which are highly applicable in the context, and even weaker (possibly negative) reinforcement for senses which are inapplicable. This encourages only one sense of a word to have high probability in a given context, as desired.

B Lemmatization

The training corpus and all text used for evaluation are lemmatized as follows: first, we perform part-of-speech (POS) tagging using Stanford CoreNLP's POS tagger (Manning et al., 2014). Any token with a tag associated with inflectional morphology in English (NNS, JJR, JJS, RBR, RBS, VBD, VBG, VBP, VBZ or VNB) is split into two separate tokens, its lemmatized form and a special token. There is a unique special token for each of the above tags except the pairs JJR and RBR (comparative adjectives and adverbs) and JJS and RBS (superlative adjective and adverbs), which share [COMP] and [SUP] tokens respectively.