Disambiguated skip-gram model

Karol Grzegorczyk[1,2] and **Marcin Kurdziel**[1]
[1]Department of Computer Science,
AGH University of Science and Technology, Kraków, Poland
[2]Allegro, Poznań, Poland
{kgr,kurdziel}@agh.edu.pl

Abstract

We present *disambiguated skip-gram*: a neural-probabilistic model for learning multi-sense distributed representations of words. Disambiguated skip-gram jointly estimates a skip-gram-like context word prediction model and a word sense disambiguation model. Unlike previous probabilistic models for learning multi-sense word embeddings, disambiguated skip-gram is end-to-end differentiable and can be interpreted as a simple feed-forward neural network. We also introduce an effective pruning strategy for the embeddings learned by disambiguated skip-gram. This allows us to control the granularity of representations learned by our model. In experimental evaluation disambiguated skip-gram improves state-of-the are results in several word sense induction benchmarks.

1 Introduction

Distributed representations of words find applications in a broad range of tasks, from natural language parsing (Socher et al., 2013) to image captioning (Karpathy and Fei-Fei, 2015). Their usefulness led to a renewed interest in word embedding algorithms. The most popular algorithms of this kind learn word vectors in an unsupervised manner, e.g., from word contexts (Mikolov et al., 2013a) or from statistics of word co-occurrence (Pennington et al., 2014). Unsupervised learning of word embeddings has a clear advantage: both general and domain-specific text corpora are available for a number of languages, which greatly reduces the cost of training. That said, unsupervised learning of word embeddings comes with its own challenges. One of the most important is word ambiguity: words in a natural language often have more than one meaning. The word *mouse*, for example, may mean a pointing device or an animal. Word embedding algorithms often do not recognize this language feature and estimate only one vector representation per word. This may lead to suboptimal word representations.

The main contribution of this work is *disambiguated skip-gram*: a neural-probabilistic model for learning distributed representations of words that capture word ambiguity. Disambiguated skip-gram builds upon the *skip-gram* model introduced by Mikolov et al. (2013a,b). Skip-gram constructs word embeddings via an auxiliary prediction task: given a word in a sentence, skip-gram attempts to predict the surrounding words. To this end, skip-gram defines a simple softmax model for the conditional probability of observing a context word c given the center word w:

$$p\left(c \mid w\right) = \frac{\mathrm{e}^{\mathbf{v}_w^{\mathrm{T}}\mathbf{u}_c}}{\sum_{c' \in D} \mathrm{e}^{\mathbf{v}_w^{\mathrm{T}}\mathbf{u}_{c'}}}, \tag{1}$$

where D is the vocabulary. This log-bilinear model assigns two embedding vectors to every word $w \in D$: an *input embedding vector* $\mathbf{v}_w$ and an *output embedding vector* $\mathbf{u}_w$. Skip-gram defines the training objective for a single example as the log-probability: $\log p\left(c \mid w\right)$. By maximizing this objective, skip-gram estimates input and output vectors that reflect semantic relations between words that occur in similar contexts. The input vectors are then used as word embeddings.

The main idea behind disambiguated skip-gram is to jointly learn to disambiguate words and predict their contexts. We therefore extend skip-gram with a parametric word sense disambiguation model. This allows us to discover word senses in an unsupervised manner, while preserving the simplicity of the skip-gram approach. In particular, unlike previous probabilistic models for multi-sense word embeddings, disambiguated skip-gram can be seen as a simple feed-forward neural network amendable to end-to-end training with back-

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1445–1454
Brussels, Belgium, October 31 - November 4, 2018.

propagation. Furthermore, disambiguated skip-gram admits an effective pruning strategy for the learned word sense embeddings. In particular, we control the granularity of the learned representations by penalizing the entropy of the probability distributions learned by the disambiguation model. We then marginalize word sense probabilities over the training examples and prune embeddings with low marginal probability.

We have carried out an extensive experimental evaluation of disambiguated skip-gram. Our results demonstrate that the multi-sense word embeddings learned by disambiguated skip-gram improve state-of-the-art results in the word sense induction task.

2 Disambiguated skip-gram model

Let $X = [w_1, \ldots, w_n]$, where each $w_i \in D$, be a sequence of words from a vocabulary D. By $\mathcal{C}_{w_i}$ we denote the context of the word w_i in X. For example, $\mathcal{C}_{w_i}$ can be a set of words that are no further than l positions from w_i and are in the same sentence as w_i. To simplify notation we will usually omit the sequence index i and write $w \in X$ for an element of the input sequence X and $\mathcal{C}_w$ for its context. We will also use notation $\mathbf{y}_w$ for a vector $\mathbf{y}$ (from some set of vectors indexed by the vocabulary words) corresponding to the word w. Note that in this case we disregard the position of w in X and use just the word as the index. In particular, if some word w occurs multiple times in X, all occurrences share a single vector $\mathbf{y}_w$.

Similarly to skip-gram, the disambiguated skip-gram model constructs word embeddings by learning to predict context words $c \in \mathcal{C}_w$ given the center word $w \in X$. However, disambiguated skip-gram explicitly accounts for word ambiguity. To this end, we represent each word $d \in D$ by a set of k *sense embedding vectors* $\mathbf{v}_{dz}$, indexed by $z \in \{1, \ldots k\}$, and an *output embedding vector* $\mathbf{u}_d$. We then parametrize the conditional probability $p(c \mid w, z = j)$ of observing a word c in the context of the word w in its j-th sense with a softmax model similar to the original skip-gram parametrization:

$$p(c \mid w, z = j) = \frac{\mathrm{e}^{\mathbf{v}_{wj}^{\mathrm{T}} \mathbf{u}_c}}{\sum_{c' \in D} \mathrm{e}^{\mathbf{v}_{wj}^{\mathrm{T}} \mathbf{u}_{c'}}}. \quad (2)$$

Furthermore, in this work we assume that a sense of the word $w \in X$ can be guessed from its context $\mathcal{C}_w$[1], i.e. that:

$$z_w \sim p(z = j \mid w, \mathcal{C}_w), \quad j = 1, \ldots, k, \quad (3)$$

where z_w is an index of the sense of the word $w \in X$. Given this assumption, we parametrize the probability distribution for z_w using a softmax model similar to Eq. 2. That is, for each word $d \in D$ we introduce k *sense disambiguation vectors* $\mathbf{q}_{ds}$, $s \in \{1, \ldots, k\}$, and a *context embedding vector* $\mathbf{r}_d$. The conditional probability that the word $w \in X$ occurs in its j-th sense is then modelled as:

$$p(z = j \mid w, \mathcal{C}_w) = \frac{\mathrm{e}^{\mathbf{q}_{wj}^{\mathrm{T}} \bar{\mathbf{r}}_w}}{\sum_{s=1,\ldots,k} \mathrm{e}^{\mathbf{q}_{ws}^{\mathrm{T}} \bar{\mathbf{r}}_w}}, \quad (4)$$

where $\bar{\mathbf{r}}_w$ is a vector representation of $\mathcal{C}_w$. We represent $\mathcal{C}_w$ by an average of context embedding vectors:

$$\bar{\mathbf{r}}_w = \frac{1}{\#\mathcal{C}_w} \sum_{c \in \mathcal{C}_w} \mathbf{r}_c. \quad (5)$$

We can now define the training objective for a single word $w \in X$ as the expected negative log-likelihood of observing the context $\mathcal{C}_w$ under the distribution of senses of the center word w:

$$\mathcal{L}(\boldsymbol{\phi}, w) = \mathbb{E}_{z_w \sim p(z=j|w,\mathcal{C}_w)} \left[-\log \prod_{c \in \mathcal{C}_w} p(c \mid w, z = z_w) \right], \quad (6)$$

with the parameters: $\boldsymbol{\phi} = \{\mathbf{v}_{dz}, \mathbf{u}_d, \mathbf{q}_{dz}, \mathbf{r}_d | d \in D, z = 1, \ldots, k\}$.

The objective in Eq. 6 is inconvenient for gradient-based optimization, because the expectation is taken with respect to a probability distribution that is a function of model parameters. In principle, we can estimate the gradient of this objective with the score function estimator (Glynn, 1990; Williams, 1992). Unfortunately, the score function estimator suffers from a high variance, even when used with a control variate. One can also derive a low-variance unbiased gradient estimator for certain probability distributions, by expressing the samples as a differentiable function of model parameters and a random variable from some independent fixed distribution (Kingma and Welling, 2013). This approach is not directly applicable to our case, because categorical distribution do not admit reparametrization with a differentiable function. That said, a simple and effective

[1] This assumption follows from an observation that two different meanings of a given word will often have vastly different contexts.

biased gradient estimator for an expectation with respect to a categorical distribution was recently proposed by Jang et al. (2016) and Maddison et al. (2016). The basic idea is to reparametrize the samples from the categorical distribution with the *Gumbel-Max trick* (Gumbel, 1954) and then approximate the non-differentiable $\max$ operator with a softmax function with temperature hyper-parameter. This can be seen as a reparametrization trick for a continuous relaxation to discrete samples from the categorical distribution.

In our case, the samples from $p\left(z = j \mid w, \mathcal{C}_w\right)$ take the form:

$$z_{wj} = \begin{cases} 1 & \text{if } j = \arg\max_s (\xi_s + \\ & \qquad \log p\left(z = s \mid w, \mathcal{C}_w\right)), \\ 0 & \text{otherwise,} \end{cases}$$

where ξ_s are i.i.d. samples from the standard Gumbel distribution $f\left(0, 1\right)$. Note that the samples $\mathbf{z}_w = [z_{w1}, \ldots, z_{wk}]$ are now one-hot encoded. The continuous relaxation to $\mathbf{z}_w$ is:

$$\begin{aligned} &\tilde{\mathbf{z}}_w = \left[\tilde{z}_{wj}\left(\xi_1, \ldots, \xi_k, \mathcal{C}_w\right) \mid j = 1 \ldots k\right], \\ &\tilde{z}_{wj}\left(\xi_1, \ldots, \xi_k, \mathcal{C}_w\right) = \\ &\qquad \frac{\mathrm{e}^{[\xi_j + \log p(z=j|w,\mathcal{C}_w)]/\tau}}{\sum_{s=1,\ldots,k} \mathrm{e}^{[\xi_s + \log p(z=s|w,\mathcal{C}_w)]/\tau}}, \end{aligned} \tag{7}$$

where τ is the temperature hyper-parameter. When $\tau \to 0$, we recover the samples from $p\left(z = j \mid w, \mathcal{C}_w\right)$. However, for $\tau > 0$ the samples $\tilde{\mathbf{z}}_w$ are no longer discrete. In this case we consider a relaxed sense embedding vector:

$$\tilde{\mathbf{v}}_w = \sum_{j=1,\ldots,k} \tilde{z}_{wj}\left(\xi_1, \ldots, \xi_k, \mathcal{C}_w\right) \mathbf{v}_{wj} \tag{8}$$

and model the conditional probability of observing a word c in the context of the word w as:

$$p\left(c \mid w, \tilde{\mathbf{v}}_w\right) = \frac{\mathrm{e}^{\tilde{\mathbf{v}}_w^{\mathrm{T}} \mathbf{u}_c}}{\sum_{c' \in D} \mathrm{e}^{\tilde{\mathbf{v}}_w^{\mathrm{T}} \mathbf{u}_{c'}}}. \tag{9}$$

Our training objective for a single word $w \in X$ then takes the form:

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{\phi}, w\right) = &\mathbb{E}_{\xi_1, \ldots, \xi_k \sim f(0,1)} \\ &\left[-\log \prod_{c \in \mathcal{C}_w} p\left(c \mid w, \tilde{\mathbf{v}}_w\right)\right]. \end{aligned} \tag{10}$$

The relaxed objective in Eq. 10 is tractable and differentiable with respect to the parameters $\boldsymbol{\phi}$.

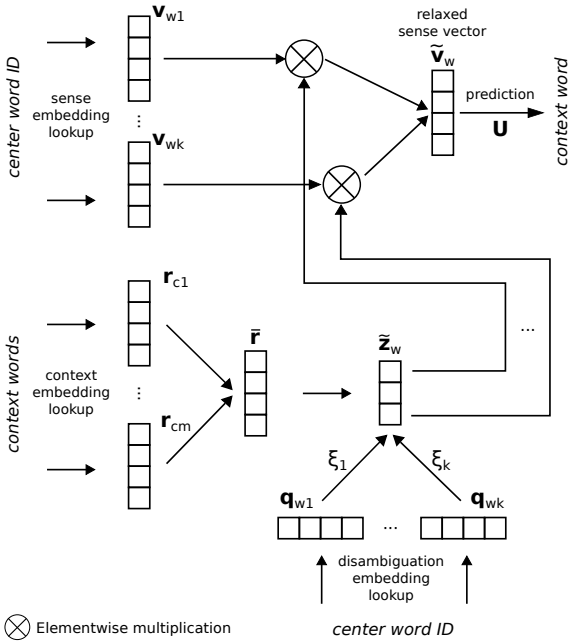

Figure 1: Disambiguated skip-gram model.

When $\tau \to 0$, it becomes equivalent to the objective in Eq. 6. In practice, we approximate the expectation in Eq. 10 with a one-sample Monte Carlo estimator. In these settings disambiguated skip-gram can be seen as a simple feed-forward neural network pictured in Fig. 1. During training, the network jointly estimates a sense disambiguation model (Eq. 4) and a context word prediction model (Eq. 2), which we use to construct multi-sense word embeddings.

2.1 Pruning word senses

The disambiguated skip-gram model is parametric, i.e. it allocates a fixed number of sense embedding vectors to each word, even though different words have different number of discernible senses. That said, we can prune the sense embedding vectors by considering their probabilities according to the learned disambiguation model. In particular, after training we estimate the marginal probability:

$$p\left(d, j\right) = \frac{1}{m_d} \sum_{\substack{w_i \in X, \\ w_i = d}} p\left(z = j \mid w_i, \mathcal{C}_{w_i}\right) \tag{11}$$

for each word $d \in D$ and each sense index $j \in \{1, \ldots, k\}$. We then prune sense embedding vectors with low marginal probability, e.g., $p\left(d, j\right) < 0.05$. The normalizing factor m_d in Eq. 11 is the number of occurrences of the word d in X.

The above pruning technique can be extended to allow for an explicit control over the granularity of the learned sense representations. To this end, we use an entropy regularization term similar to the one studied by Pereyra et al. (2017) in classification networks. In disambiguated skip-gram the granularity of the learned representations is controlled by the disambiguation model (Eq. 4). Therefore, we extend the objective of our model (Eq. 10) by adding to it an entropy S of the probability distribution $p\left(z = j \mid w, \mathcal{C}_w\right)$:

$$\mathcal{L}_r\left(\boldsymbol{\phi}, w\right) = \mathcal{L}\left(\boldsymbol{\phi}, w\right) + \gamma S\left(\boldsymbol{\phi}, w\right), \qquad (12)$$

where:

$$S\left(\boldsymbol{\phi}, w\right) = -\sum_{j=1}^{k} p\left(z = j \mid w, \mathcal{C}_w\right) \log p\left(z = j \mid w, \mathcal{C}_w\right).$$

The hyper-parameter γ, which we further call *entropy cost*, controls the strength of the regularization and, in turn, the granularity of the learned sense representations. In particular, $\gamma > 0$ encourages the model to learn more coarse-grained sense representations, whereas $\gamma < 0$ increases the granularity of the learned senses.

3 Related work

Algorithms for learning distributed multi-sense representations of words have been a focus of several recent works. Initial approaches to this task relied on clustering word contexts. One of the first algorithms of this kind was proposed by Huang et al. (2012). They learn multi-sense word representations in three steps. First, they estimate vector representations of words using a feed-forward neural language model. Next, they calculate average word vector for each context in the training corpus, cluster these context representations and relabel each word in the corpus to a word sense represented by the nearest cluster. Finally, they train the language model on the relabelled corpus and obtain vector representations for word senses.

Neelakantan et al. (2014) proposed the Multi-Sense Skip-gram (MSSG) model, that jointly learns context cluster prototypes and word sense embeddings. Their model extends skip-gram by maintaining context clusters for every word in the vocabulary. Given a training example with a center word w and its context representation c, they infer the word sense for w by a hard assignment of the context representation c to the cluster with the nearest centroid. Afterwards, they perform a skip-gram-like update on the vector representation of the selected word sense and the output vectors of the context words. Neelakantan et al. also proposed a non-parametric version of MSSG (NP-MSSG), in which the number of clusters, and in turn the number of word senses, increases during training. They use a simple heuristic to determine the number of word senses: NP-MSSG allocates a new sense for the center word w when the similarity between the context representation c and the nearest cluster centroid falls below some predefined threshold. Neelakantan et al. demonstrated that MSSG and NP-MSSG outperform the Huang et al. algorithm on a contextual word similarity task.

A disadvantage of the Huang et al. and Neelakantan et al. algorithms is that they do not follow a principled statistical approach, but instead rely on hard clustering of context vectors. This has been addressed in more recent algorithms, which learn multi-sense word representations in a probabilistic framework. Concurrent to Neelakantan et al. work, Tian et al. (2014) proposed a probabilistic Multi-Prototype Skip-Gram (MPSG) model. MPSG extends the skip-gram model by adding to each position in the input text a latent variable that encodes the index of the word sense at that position. Furthermore, for each word in the vocabulary MPSG maintains a fixed number of sense embedding vectors and a single output vector. These parameters define a softmax model for the conditional probability of observing a context word given the center word and the latent sense index. Finally, MPSG models the conditional probability of observing a context word given the center word with a mixture model whose components correspond to the senses of the center word. Tian et al. derived an expectation maximization algorithm for estimation of softmax parameters and prior sense probabilities in their model. MPSG was evaluated in a contextual word similarity task, where its performance was similar to that of the Huang et al. algorithm.

Bartunov et al. (2016) proposed the AdaGram model, which can be seen as a non-parametric version of MPSG. Similarly to MPSG, AdaGram introduces latent variables for word sense indexes in the input text. However, unlike MPSG, AdaGram does not assume a fixed number of word senses.

Instead, it defines the prior over word senses via a Dirichlet process. As a result, AdaGram automatically learns the number of senses for all words in the vocabulary. Unfortunately, defining the prior over word senses via a Dirichlet process gives an intractable model likelihood. Bartunov et al. therefore optimize variational lower bound of the AdaGram model likelihood using a stochastic variational inference algorithm. Bartunov et al. evaluated AdaGram performance on several word sense induction benchmarks. They demonstrated that AdaGram consistently outperforms MSSG, NP-MSSG and MPSG models in these benchmarks. AdaGram has been recently extended to handle parallel multilingual text corpora (Upadhyay et al., 2017).

For the prediction of context words (Eq. 2) disambiguated skip-gram adopts the softmax model used in MPSG. However, in contrast to the previous works, disambiguated skip-gram learns a parametric model for the conditional probability distribution over senses of the center word given the context words (Eq. 4). This allows us to define the training objective for disambiguated skip-gram as the expected negative log-likelihood of observing the context words under the distribution of senses of the center word. We use a biased low-variance gradient estimator for this objective, which enables stable end-to-end training with backpropagation.

The main goal of the AdaGram model is to automatically discover the number of word senses for the vocabulary words. This does not mean, however, that the number of senses learned by AdaGram is independent of model hyper-parameters. On the contrary, the number of senses learned by AdaGram is directly controlled by the hyper parameter α in the Dirichlet process used to define the prior over word meanings (Bartunov et al., 2016). Disambiguated skip-gram controls the number of learned senses by penalizing the entropy of the conditional probability distribution in the sense disambiguation model (Eq. 12). The entropy cost γ in this approach performs a function similar to the hyper-parameter α in AdaGram.

In addition to the works discussed above, word ambiguity was also modelled using topic models (Liu et al., 2015), large bi-directional language models (Peters et al., 2018) or subword information (Athiwaratkun et al., 2018). Also, Li and Jurafsky (2015) evaluated multi-sense embeddings in several downstream tasks. They found that multi-sense embeddings improve performance in tasks such as POS tagging or identification of semantic relations. They also identify downstream tasks which do not benefit from sense disambiguation. In sentiment analysis, for example, word sentiment usually does not depend on the inferred sense.

4 Experiments

We conducted a number of experiments to evaluate the quality of multi-sense word embeddings learned by disambiguated skip-gram. This section reports results from our evaluation. First, we report qualitative results from our model for several polysemous words. We then compare the performance of disambiguated skip-gram with several competing algorithms on a set of word sense induction tasks. Finally, we evaluate the effect of the entropy cost on the learned representations.

It is worth noting that the quality of multi-sense word embeddings was formerly assessed in contextual word similarity experiments. However, Bartunov et al. (2016) demonstrated that contextual word similarity experiments do not reflect the quality of multi-sense representations. In particular, the best performance in contextual word similarity task is often achieved by the baseline skip-gram model, which does not recognize word senses. This can be attributed to the fact that skip-gram objective directly optimizes similarity of vector representations of words that appear in similar contexts. Multi-sense models, on the other hand, solve a harder task: they disambiguate words in contexts and then model the similarities between the discovered senses. Bartunov et al. focus, therefore, on the performance of multi-sense embeddings in the word sense induction task. We adopt their evaluation methodology in this work.

We trained our disambiguated skip-gram models on the Westbury Lab Wikipedia corpus (Shaoul and Westbury, 2010). We optimized the models using mini-batch stochastic gradient descend with momentum.

4.1 Qualitative results

We begin our evaluation by presenting senses discovered by disambiguated skip-gram for several ambiguous words. For the demonstration we trained four 300-dimensional models with three sense embedding vectors allocated to each word

Word	$\gamma = 0.0$		$\gamma = 0.25$		$\gamma = 0.5$	
	p	Nearest neighbors	p	Nearest neighbors	p	Nearest neighbors
fox	0.52	nbc cbs network syndication espn	0.60	nbc cbs abc syndication network	0.68	cbs nbc abc cable colmes
	0.25	miller allen plummer crowe buck	0.22	miller allen terry russell soper	0.18	allen russell miller turner berry
	0.24	badger wolf coyote weasel marten	0.18	badger squirrel weasel raccoon marten	0.14	badger marten raccoon beaver mink
net	0.41	ebitda earnings annualized taxable depreciation	0.43	ebitda annualized jpy deadweight gni	0.77	ebitda deadweight isk annualized deducting
	0.32	trawl streamline maximising minimises counteracts	0.33	trawl minimises maximising streamlines stickiness	0.23	crossbar puck lob header dribbled
	0.26	crossbar puck lob sliothar offside	0.24	crossbar puck lob offside dribbled	0	
rock	0.41	band indie punk alternative supergroup	0.7	alternative glam progressive indie psychedelic	0.72	alternative punk indie glam progressive
	0.34	punk rockabilly pop psychedelia funk	0.17	boulder basalt outcrop quartzite cliffs	0.17	basalt boulders quartzite cliff outcropping
	0.26	boulder quartzite granite sandstone basalt	0.13	granite bluff pine pigeon ledge	0.11	pine bluff eagle pigeon turtle
plant	0.45	flowering perennial shrub grass fungus	0.46	flowering grasses shrub fungus herbaceous	0.48	flowering shrub grasses herbaceous fungus
	0.38	refinery smelter petrochemical processing factory	0.45	refinery factory megawatt smelter cogeneration	0.48	refinery factory smelter megawatt sellafield
	0.18	factory botanical laboratory farm nurseryman	0.08	weed planted shed grinder laboratory	0.04	
mouse	0.47	mickey rabbit goofy cat porky	0.5	rat mice rodent mus elegans	0.51	mice rodent rat mus elegans
	0.35	cursor joystick trackball touchpad touchscreen	0.49	rabbit goofy cat porky tigger	0.49	rabbit goofy porky tigger tweety
	0.19	rodent vole shrew pygmy rat	0.01		0	
apple	0.46	macintosh imac iigs iie iic	0.64	macintosh iigs imac iie iic	0.95	macintosh blackberry iigs imac apricot
	0.28	wozniak macworld macintosh ipod sculley	0.25	wozniak blackberry tomato potato popcorn	0.04	
	0.26	strawberry peach raspberry blueberry plum	0.11	peach pecan persimmon prune blueberry	0.01	
table	0.40	sortable column lookup hashed tray	0.48	sortable column lookup tray hashed	0.66	sortable tray column chairs buckets
	0.38	foosball carom languishing pool slipping	0.36	foosball ept languishing pool leaderboard	0.31	standings ept foosball leaderboard ittf
	0.22	sortable list alphabetical descending brackets	0.16	sortable descending list alphabetically please	0.03	

Table 1: Nearest neighbors and marginal probabilities p of word sense embedding vectors discovered by the disambiguated skip-gram model for several ambiguous words. Sense embedding vectors with a marginal probability $p < 0.05$ are pruned from the learned model.

and the entropy cost γ ranging from 0.0 to 0.5. For each of the evaluated words sense embeddings we calculated the cosine similarity to the remaining words and selected 5 nearest neighbours. The results are reported in Tab. 1.

Disambiguated skip-gram discovered main meanings of our test words. For example, the meanings discovered for the word *fox* correspond to a broadcasting company, an animal and a family name. The meanings discovered for the word *mouse* correspond to a cartoon character, a computer mouse and an animal.

Results in Tab. 1 also demonstrate that disambiguated skip-gram will often expose an internal structure in a word meaning, if that meaning appears in different contexts. For example, disambiguated skip-gram learned two embeddings for the word *plant* corresponding to its *factory* mean-

ing: one related to heavy industry and one related to a farm or a plant nursery. This is a consequence of the fact that disambiguated skip-gram discovers word senses using only the information about the contexts in which these words occur. In particular, it does not employ any supervision from an external knowledge base. Bartunov et al. (2016) refer to a related phenomenon in the AdaGram embeddings as the semantic resolution of the model.

4.2 Word-sense induction experiments

To compare disambiguated skip-gram with state-of-the-art competing algorithms we assessed its performance in a set of word sense induction tasks. In this evaluation we follow the experimental setup from (Bartunov et al., 2016), allowing for direct comparison with the results reported therein. In particular, we evaluated disambiguated skip-gram on the datasets from the SemEval-2007 Task 2 competition (SE-2007), SemEval-2010 Task 14 competition (SE-2010), SemEval-2013 Task 13 competition (SE-2013) and the Wikipedia Word-sense Induction (WWSI) dataset introduced by Bartunov et al. The test datasets consist of between 4664 and 36354 examples. Each example provides a ground truth sense of a center word and the context in which this sense appeared. The goal is to recognize the sense of the center word given the context. We use the preprocessed versions of SE-2007, SE-2010 and WWSI datasets made available by Bartunov et al. For the SemEval-2013 Task 13 we use the original competition dataset (Jurgens and Klapaftis, 2013) and follow the preprocessing steps reported in (Bartunov et al., 2016).

We use a simple procedure for resolving word senses. That is, we average all sense embedding vectors of all context words and select the sense z_w of the center word w whose embedding vector is most similar to the average vector:

$$z_w = \arg\max_j \cos\left(\mathbf{v}_{wj}, \bar{\mathbf{c}}_w\right), \qquad (13)$$

where:

$$\bar{\mathbf{c}}_w = \left(k \cdot \#\mathcal{C}_w\right)^{-1} \sum_{c \in \mathcal{C}_w} \sum_{s=1}^{k} \mathbf{v}_{cs}. \qquad (14)$$

The intuition behind this procedure is that we expect the average to preserve a shared component in the embedding vectors, namely embeddings for senses related to the sense of the center word, and cancel out embeddings of unrelated senses. In addition to averaging sense embedding vectors we also experimented with averaging output vectors of context words. However, this approach usually gave slightly worse results.

Following Bartunov et al. (2016), we use adjusted rand index (Hubert and Arabie, 1985) to compare ground truth senses for a given word with the senses inferred from disambiguated skip-gram embeddings. The final performance on a benchmark task is the average of adjusted rand index values over all test words in the task.

For this evaluation we trained a 300-dimensional disambiguated skip-gram model with 5 sense embedding vectors allocated to each word and no entropy cost ($\gamma = 0.0$). The comparison between our model and MSSG, NP-MSSG, MPSG and AdaGram is reported in Tab. 2. The results for MSSG, NP-MSSG, MPSG and AdaGram are taken from Bartunov et al. (2016). Multi-sense embedding vectors learned by disambiguated skip-gram outperform baseline methods on the SE-2007, SE-2010 and WWSI benchmarks, and achieve the second best result on the SE-2013 benchmark.

	SE-2007	SE-2010	SE-2013	WWSI
MSSG	0.048	0.085	0.033	0.194
NP-MSSG	0.033	0.044	0.033	0.110
MPSG	0.044	0.077	0.014	0.160
AdaGram	0.069	0.097	**0.061**	0.286
Disamb. skip-gram	**0.077**	**0.117**	0.045	**0.292**

Table 2: Performance of multi-sense word embedding methods in word sense induction tasks. The reported performance metric is the adjusted rand index averaged over all test words in the benchmark task. Results for all models except the disambiguated skip-gram (Disamb. skip-gram) are taken from (Bartunov et al., 2016).

4.3 Effect of the entropy cost

Results reported in Tab. 1 demonstrate that the entropy cost γ (Eq. 12) indeed allows for pruning senses learned by disambiguated skip-gram. In particular, when the entropy cost increases, disambiguated skip-gram allocates more of the marginal probability mass (Eq. 11) to the frequent meanings of the modelled words and, in effect, learns coarser representations.

For a quantitative evaluation of the effect of

entropy cost on the learned representations we trained 50-dimensional disambiguated skip-gram models with γ ranging from 0.0 to 1.0. All models allocate 5 sense embedding vectors to every word in the vocabulary. In Tab. 3 we report an average number of senses per word with marginal probability $p \geq 0.05$, depending on the value of the entropy cost. In Fig. 2 we also report histograms of marginal probabilities for selected entropy cost values.

Entropy cost	0.0	0.1	0.25	0.5	0.75	1.0
Avg. sense num.	4.7	4.3	3.7	3.2	2.8	2.5

Table 3: Average number of senses per word with marginal probability $p \geq 0.05$, learned by disambiguated skip-gram models with different values of the entropy cost.

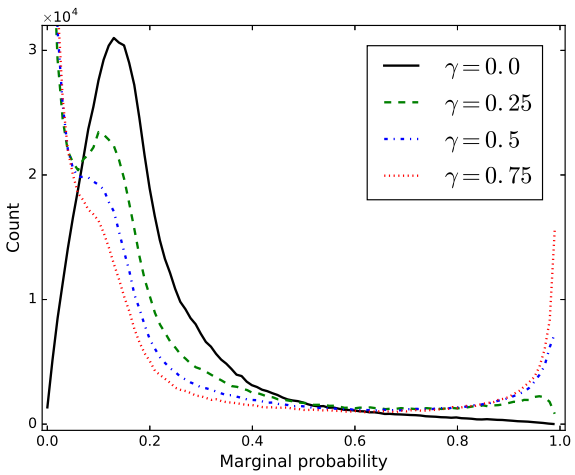

Figure 2: Histograms of marginal probabilities of word senses learned by disambiguated skip-gram models with different values of the entropy cost.

Histograms in Fig. 2 confirm our observation from the qualitative evaluation: when the entropy cost increases, disambiguated skip-gram learns more peaked distributions for the conditional sense probability $p(z = j \mid w, \mathcal{C}_w)$. This translates to coarser sense representations. In particular, the model with no entropy cost learned an average of 4.7 senses per word with marginal probability $p \geq 0.05$ (Tab. 3). This number decreases with an increasing entropy cost, reaching an average of 2.5 senses per word for $\gamma = 1.0$.

We also evaluated 50- and 300-dimensional models with different entropy costs in the word sense induction tasks. In each case we pruned senses with marginal probability $p < 0.05$. Results from this evaluation (Tab. 4) indicate that the desired granularity of the learned sense representations depends on the underlying task. In the WWSI benchmark the best performing models had no entropy cost, while in the SemEval tasks small entropy cost usually improved results. The results agree for both model dimensionalities.

Dim.	Entropy Cost	SE 2007	SE 2010	SE 2013	WWSI
50	No	0.064	0.107	0.040	**0.304**
	0.25	**0.083**	**0.116**	0.043	0.244
	0.5	0.064	0.091	**0.045**	0.182
300	No	0.077	**0.117**	0.045	**0.292**
	0.25	**0.079**	0.113	0.045	0.259
	0.5	0.065	0.091	**0.049**	0.183

Table 4: Performance of disambiguated skip-gram models with different entropy costs in the word sense induction tasks. The reported performance metric is the adjusted rand index averaged over all test words in the benchmark task.

5 Conclusions

In this work we developed disambiguated skip-gram: a novel neural-probabilistic model for learning multi-sense distributed representations of words. Unlike previous probabilistic models for multi-sense word embeddings, disambiguated skip-gram is a simple feed-forward neural network and can be trained end-to-end with backpropagation. In experimental evaluation disambiguated skip-gram improved over the state-of-the-art results in three out of four benchmark datasets and ranked second the fourth.

Disambiguated skip-gram optimizes expected log-likelihood of the context prediction model under the distribution of word senses parametrized by the disambiguation model. We choose to optimize this objective with a biased but low-variance gradient estimator. However, parallel to this work there has been a significant progress in gradient-based training of models with discrete latent variables. Specifically, Tucker et al. (2017) proposed an unbiased low-variance gradient estimator, called REBAR, that is applicable to models with categorical latent variables. REBAR may allow to efficiently optimize the original disambiguated skip-gram objective (Eq. 6), instead of the relaxed objective (Eq. 10). This may further improve the quality of embeddings learned with our approach.

Acknowledgments

This research was supported by National Science Centre, Poland grant no. 2013/09/B/ST6/01549 "Interactive Visual Text Analytics (IVTA): Development of novel, user-driven text mining and visualization methods for large text corpora exploration". The paper was partially financed by AGH University of Science and Technology Statutory Fund. The paper was also partially financed by Allegro. Experiments for this work were supported by the PL-Grid infrastructure and by the "HPC Infrastructure for Grand Challenges of Science and Engineering" project co-financed by the European Regional Development Fund under the Innovative Economy Operational Programme. Last but not least, we would like to thank Professor Witold Dzwinel for overall guidance and support.

References

Ben Athiwaratkun, Andrew Wilson, and Anima Anandkumar. 2018. Probabilistic FastText for multi-sense word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11. Association for Computational Linguistics.

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*, pages 130–138.

Peter W Glynn. 1990. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84.

Emil Julius Gumbel. 1954. *Statistical theory of extreme values and some practical applications: a series of lectures*. 33. US Govt. Print. Office.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y 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: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with Gumbel-Softmax. *arXiv preprint arXiv:1611.01144*.

David Jurgens and Ioannis P Klapaftis. 2013. Semeval-2013 task 13: Word sense induction for graded and non-graded senses. In *Proceedings of the 7th international workshop on semantic evaluation*, pages 290–299. Association for Computational Linguistics.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3128–3137.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

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 (EMNLP)*, pages 1722–1732. Association for Computational Linguistics.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2418–2424.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 14, pages 1532–1543. Association for Computational Linguistics.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*.

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. Association for Computational Linguistics.

Cyrus Shaoul and Chris Westbury. 2010. The Westbury lab Wikipedia corpus.

Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 455–465.

Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. 2014. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 151–160. Dublin City University and Association for Computational Linguistics.

George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. 2017. REBAR: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems 30*, pages 2624–2633.

Shyam Upadhyay, Kai-Wei Chang, Matt Taddy, Adam Kalai, and James Zou. 2017. Beyond bilingual: Multi-sense word embeddings using multilingual context. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 101–110. Association for Computational Linguistics.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.