The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

# Bayesian Verb Sense Clustering

**Daniel W. Peterson, Martha Palmer**
University of Colorado
{daniel.w.peterson,martha.palmer}@colorado.edu

**Abstract**

This work performs verb sense induction and clustering based on observed syntactic distributions in a large corpus. VerbNet is a hierarchical clustering of verbs and a useful semantic resource. We address the main drawbacks of VerbNet, by proposing a Bayesian model to build VerbNet-like clusters automatically and with full coverage. Relative to the prior state of the art, we improve accuracy on verb sense induction by over 20% absolute F1. We then propose a new model, inspired by the positive pointwise mutual information (PPMI). Our PPMI-based mixture model permits an extremely efficient sampler, while improving performance. Our best model shows a 4.5% absolute F1 improvement over the best non-PPMI model, with over an order of magnitude less computation time. Though this model is inspired by clustering verb senses, it may be applicable in other situations where multiple items are being sampled as a group.

## 1 Introduction

The distributional hypothesis is most eloquantly stated as, "You shall know a word by the company it keeps" (Firth 1957). Semantic vectors built on this principle (Deerwester et al. 1990; Mikolov, Yih, and Zweig 2013; Pennington, Socher, and Manning 2014) have proven to be highly effective representations of words for many natural language processing tasks. The vectors capture a view of a word's context, typically representing the words appearing within a reasonably small, sliding window.

This work builds on a modified distributional hypothesis for analyzing verb semantics, which states that the distribution of syntactic expressions can play an equally important role in reflecting a word's semantics (Levin 1993). VerbNet (Kipper-Schuler 2005) is a lexical resource relying on this syntactic version of the distributional hypothesis. It is a hierarchical clustering of verbs, manually built by linguists, on this theoretical footing. Polysemy, where a word may have more than one meaning, is important to verb semantics - "enter a room" is fundamentally different than "enter a race", and we should account for this fact. VerbNet handles polysemy by allowing distinct senses of verbs to participate in different clusters. VerbNet additionally lists the allowed syntactic patterns for each class, and labels the semantic roles of arguments. VerbNet's accuracy, and rich annotation, has made it useful for semantic tasks like semantic role labeling (Palmer, Gildea, and Xue 2010; Giuglea and Moschitti 2006).

In practice, it is difficult to get the full value from VerbNet. It suffers from lack of coverage, despite manual work that spans over a decade. Verb use and meaning varies widely, and specialized domains such as legal and medical documents are poorly covered. It has no coverage for languages other than English. Expanding coverage to these areas through manual effort alone is infeasible.

Automated approaches to this problem are promising in general, and address the main concerns of extending coverage and domain specificity. There has been a limited amount of work in this domain. Mainly it has focused on probabilistic graphical models, which can give a measure of uncertainty along with their estimates of cluster membership. Given a corpus, such a model can output a verb clustering that has full coverage, tailored specifically to the domain of interest. For any sentence, a full description of the possible cluster memberships can be trivially computed, and semantic processing of documents in the target domain can become much easier. The state-of-the-art model (Kawahara, Peterson, and Palmer 2014) breaks the problem into two steps: sense induction and verb clustering. This work includes contributions to each step.

Our first contribution is improved sense induction. Our new model improves clustering accuracy of verbs into senses by an absolute F1 of over 20%. These improvements carry over to direct, across-the-board improvements in the second step, which is clustering those senses into a VerbNet-like structure.

Our second contribution is a new model, based on the positive pointwise mutual information (PPMI). This measure has been shown to be extremely effective at modeling distributional similarity, and has wide applicability on semantic tasks (Levy and Goldberg 2014b). The main insight behind this model is that Dirichlet multinomial mixtures, like latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003), compute the sufficient statistics of word distributions in each topic. The PPMI can be computed with the same sufficient statistics, and is a sparse representation whose vector is semantically meaningful.

With this new model, we are able to achieve better clus-

ters in dramatically less time. Our best-performing model outperforms the prior state-of-the-art by 15% absolute F1, and outperforms the best non-PPMI model by almost 4.5% absolute F1. The speedup for the PPMI model is over 20x in the worst case.

The PPMI-based model is novel, but may be useful in non-textual domains. Word2Vec is implicitly tied to the pointwise mutual information, and has been adapted for use in areas like network analysis (Grover and Leskovec 2016) and genetics (Ng 2017). Our model is also likely to be useful for clustering items with rich contextual information, regardless of domain.

The main contributions of our paper are:

- improved sense induction, that dramatically outperforms the best previous approach;
- a novel PPMI-based exponential mixture model that is more accurate than prior models, is over an order of magnitude faster, and is potentially applicable in other domains; and
- a finer-grained feature set for sense clustering that boosts performance for the novel model.

### 1.1 Prior work

The state of the art model for generating VerbNet-like clusters (Kawahara, Peterson, and Palmer 2014) proposed a step-wise approach. First, induce verb senses using a Dirichlet process prior over Dirichlet-multinomial topics, with a model that is similar to latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003). Second, group senses into verb clusters using a second Dirichlet process mixture. These two steps are computed separately, instead of using the hierarchical Dirichlet process (HDP) (Teh et al. 2006), because the two steps benefit from different feature granularity. In the preliminary sense induction stage, lexicalized syntactic features (`slot:token` pairs) provide the best verb senses. However, when clustering known senses into verbs, simple `slot` features (aggregating over tokens sharing the same dependency relation) provide the most accurate clustering of verb senses.

Others have approached the task of capturing syntactic similarity by building semantic vectors using syntactic dependency relationships (Levy and Goldberg 2014a), but we still require a sense induction mechanism. We also require a map from dense semantic vectors to clusters, in order to compare to VerbNet. Both steps pose unsolved research challenges, and so we prefer clustering with Bayesian mixture models.

## 2 Verb Sense Induction using Latent Dirchlet Mixtures

The first stage in the step-wise verb clustering (Kawahara, Peterson, and Palmer 2014) is the induction of verb senses. In this step, we take the corpus as a set of "instances": each instance is extracted from a sentence in the corpus, and is a verb together with its labeled dependencies. For the sentence, "The dog chased the cat around the house," the extracted instance would look like (`verb:chase, subj:dog, dobj:cat, prep_around:house`). Sense induction is treated as a clustering problem, forming groups of instances that share a sense. Polysemy is only ambiguous within a verb, so we first partition all instances with the same verb into distinct documents, and treat the instances as atoms.

Similar to LDA, the prior work (Kawahara, Peterson, and Palmer 2014) defines a "topic" for each sense, which is a multinomial distribution over a vocabulary of `slot:token` pairs (e.g., `subj:dog` or `dobj:cat`). Topic distributions are drawn from a Dirichlet with a constant parameter $\beta$, which controls the sparseness of the multinomials. When $\beta < 1$, the topics tend to be sparse, which increases the likelihood that the induced senses are coherent, and makes sentences with similar arguments tend to group together. These topics are drawn from a Dirichlet process (DP) prior, which uses a Chinese restaurant process (CRP) (Ferguson 1973) to encourage a small, but unbounded and unspecified, number of clusters.

In particular, the probability of choosing a cluster $k$ based on the CRP is

$$P(k|\alpha, C_k(*)) \propto \begin{cases} C_k(*), & \text{if } C_k(*) > 0 \\ \alpha, & \text{if } k = k_{new}, \end{cases} \quad (1)$$

where $C_k(*)$ is the count of clustered items already in cluster $k$. The probability of choosing an instance, $I$, with slots $s_1, s_2, \ldots$, given a particular topic $\theta_k$, is given by

$$P(I|\theta_k, \beta) \propto \prod_{s_i \in I} P(s_i|\theta_k, \beta), \quad (2)$$

where $P(s_i|\theta_k, \beta) \propto C_{ik} + \beta$, and $C_{ik}$ being the count of observed instances of $s_i$ assigned to topic $k$. The final probability is given by the product of these components, namely,

$$P(k|I, \alpha, C_k(*), \theta_k, \beta) \propto P(k|\alpha, C_k(*)) \cdot P(I|\theta_k, \beta). \quad (3)$$

The sampling procedure is stated in Algorithm 1. The statement in line

**Algorithm 1** Sampling verb senses in the Dirichlet-Multinomial mixture

```
1:  for verb v in corpus C do
2:      Assign instances to random clusters and compute counts matrices
3:      for iteration in range(num_iters) do
4:          for instance I with current assignment k in v do
5:              Update counts matrices C_k and C_*k to remove instance I
6:              Sample topic k_new according to Equation (3)
7:              Update counts matrices C_k_new and C_*k to add instance I
8:          end for
9:      end for
10: end for
```

Given a large corpus and many examples of a particular verb, inducing verb senses can be accomplished using this Gibbs sampler, but the prior work also includes some refinements that speed convergence. The first is a reduction in vocabulary size by relabeling named entities (e.g. "John", "Microsoft") with a generic <name> token, and clausal complements (e.g. "John thought that <*some other sentence*>")

with a generic <CCOMP> token. We adopt the same conventions here[1]. The second is the introduction of "initial frames," groups of instances that share the same tokens in the same syntactic slots. The use of initial frames reduces the number of elements that need to be assigned to clusters, and helps initialize the topics with sensible values. However, the initial frames remain fixed throughout sampling, which means the model cannot recover from any mistakes in that initialization. Finally, treating each verb separately is both a benefit and a drawback. It is much more distributable, since there is no need to synchronize across verbs, and it allows the topics of syntactic arguments to be tailored to the specific usage of each verb. However, not all verbs are frequent, and less-frequent verbs will not necessarily have enough instances to produce coherent topics.

## 2.1 Simultaneous Sense Induction by Sharing Topics

In this work, rather than clustering instances of each verb separately, we treat the collection of instances for a verb as a document, and replace the DP with a fixed-size Dirichlet multinomial - that is to say, we sample using LDA, with a very small modification, to ensure that an instance with multiple syntactic slots will be sampled as a unit. With this modification, an instance (`subj:he`, `dobj:it`, `prep_to:me`) will have to draw all three syntactic slots from same topic, independently, as in Equation (2), and we also update the count matrices for each slot. This is detailed in Algorithm 2.

**Algorithm 2** Sampling verb senses with common topics

```
Assign instances to random clusters and compute counts matrices
for iteration in range(num_iters) do
    for verb v in corpus C do
        for instance I, with current assignment k, in v do
            Update counts matrices C_k and C_*k to remove instance I
            Sample topic k_new according to Equation (3)
            Update counts matrices C_k_new and C_*k to add instance I
        end for
    end for
end for
```

This approach is reminiscent of LDA-Frames (Materna 2012), but is much simpler. Sampling each instance as a unit encourages the topics to represent the entirety of their constituent units, and the `slot:token` vocabulary eliminates the need to sample unique topics for each syntactic slot. This has a huge advantage in ease of implementation because distributed and optimized samplers for LDA are freely available (Liu et al. 2011; Řehůřek and Sojka 2010). It is similar to the LDA-based selectional preference model of (Wu and Palmer 2015), which uses LDA on the bag of labeled-dependencies for each verb, and then uses the resulting topics for Semantic Role Labeling. Our contribution, compared to that work, is to analyze the topics in the context of sense induction; this requirement is the reason we insist that each instance should be assigned to exactly one cluster, even though the prior work did not. The Hierarchical Dirichlet Process(Teh et al. 2006) could also be used here, allowing us to leave the number of topics unspecified, but the algorithm is less practical on large datasets.

We ran our sense induction on two datasets. The first, in order to permit direct comparison with prior work, was the Gigaword corpus (Parker et al. 2011). The second is the freely-available Google Books syntactic n-grams corpus (Goldberg and Orwant 2013). To our knowledge, this is the largest dependency-parsed corpus in the English language, and the "verbargs" section neatly aggregates the information we need. Each line of the verbargs section represents a single pattern of verb and linked dependencies. Because the corpus is so large, these patterns also contain frequencies, and patterns occurring fewer than 10 times are not included. The paper associated with the release of the corpus has much more detail, but the verbargs include the direct syntactic dependencies of each verb pattern, which is exactly the information relevant to our model.

## 2.2 Clustering Evaluation

VerbNet is both the motivation for this work, and our gold standard clustering. In order to evaluate the quality of the models, we use instances from the SemLink corpus (Palmer 2009), which has VerbNet class annotation. Our test set only contains verbs that occur at least 100 times, because we want to ensure there are enough instances to compute meaningful metrics. Also, many of these verbs are polysemous, in the sense of having multiple VerbNet classes, so we have a test set that encompasses the behavior we aim to capture. Because the model is generative, we can compute probabilities of sense assignments, even for previously unseen instances. For each instance in the test set, we assign the maximum-probability (a posteriori) sense, given the learned topics. We compare this sense assignment against the known VerbNet classes using standard clustering metrics, in keeping with prior work on this task: (modified) purity and inverse purity (Korhonen, Krymolowski, and Marx 2003).

Purity is the proportion of "correct" assignments made by a clustering algorithm, and is analogous to precision in a supervised learning setting. Each found cluster is labeled with the most-frequent true class and any elements of other classes are counted as errors. Precisely, the purity for an induced clustering $K$ of $n$ items, given gold standard classes $G$ is

$$PU(K, G) = \frac{1}{n} \sum_i |K_i| \max_j \frac{|K_i \cap G_j|}{|K_i|}.$$

A perfect purity score may be trivially achieved by assigning each item to its own cluster. Indeed, the purity will always increase with the number of singleton clusters, so the purity may give misleading assessments about cluster quality. As in prior work (Korhonen, Krymolowski, and Marx 2003; Kawahara, Peterson, and Palmer 2014), we prefer the modified purity (mPU), where we count all clusters of size one as

[1] Although distinguishing named entity types is possible, exploring whether it is useful is left for future investigation.

errors,

$$mPU(K,G) = \frac{1}{n}\sum_i I(|K_i|)|K_i| \max_j \frac{|K_i \cap G_j|}{|K_i|},$$

where $I(x)$ is a step indicator function that is one iff $x > 1$, 0 otherwise.

The inverse purity (iPU) is analogous to recall, and is computed as

$$iPU(K,G) = \frac{1}{n}\sum_i |G_i| \max_j \frac{|K_j \cap G_i|}{|G_i|}.$$

This metric rewards grouping all items of the same true class together. Singleton clusters in the gold standard $G$ do always count toward increased inverse purity, but we do not generate $G$ and so have no reason to penalize its structure. There is no need for a modified inverse purity.

Table 1 shows the clustering mPU, iPU, and F1 score (simple harmonic mean of mPU and iPU) for senses induced from various models (trained on Gigaword or Google Books syntactic n-grams corpora, with 100 and 200 topics). Since we observe a large difference in relative verb frequencies, we compute micro-average and macro-average of mPU and iPU across verbs. The micro-average weights all instances equally, which gives more weight to accuracy on frequent verbs. The macro-average is the mean of the mPU (or iPU), taken on a verb-by-verb basis. We use the published model from the prior work (Kawahara, Peterson, and Palmer 2014) as a baseline. Again, we assign the instances to their maximum a posteriori sense given the published topics and topic sizes.

An odd result from this table is that the automatically-induced senses on the Google Books corpus do not generalize as well to SemLink data, despite being based on a much larger corpus. This result is discussed in more detail in Section 5.3.

## 3 Verb Clustering with Specific Features

Once we have each verb's instances grouped into senses, we can tackle the verb clustering problem. By sampling topics for all verbs together, we actually have a natural clustering of verbs induced by the topic assignments. However, in the prior state-of-the-art model, there were two parts. After the sense induction step, the induced verb senses were clustered using a mixture model with a modified vocabulary. Because Levin's distributional hypothesis is based primarily on syntax, `slot` features, which have summed across all tokens sharing the same syntactic slot, gave the best alignment to VerbNet classes.

We further refine the syntactic clustering model by introducing *pattern* features, such as `subj/dobj/prep_with`. The `slot` features worked best in the prior work, but the aggregated count of `subj` and `dobj` counts doesn't give a clear estimate of the number of transitive constructions. It is in general intractable to decipher which arguments occurred together in particular instances. Because we marginalize out the token information, the number of distinct syntactic patterns is reasonable, and there is a large amount of overlap across senses.

In Tables 2 and 3, we compare the effectiveness of these features using the model described in the prior work. For clarity, we show only the results from sense induction using 100 topics, because those induced senses perform better, and they have identical patterns of performance. As a baseline for each model, we include the performance of using the topics from the sense induction step. For the Gigaword corpus, we also show the accuracy of the prior state-of-the-art model.

## 4 A Novel Clustering Algorithm for Senses

In the second step of the clustering process we are in a situation that is not modeled well by a Dirichlet-Multinomial mixture. Each sense is an aggregate of hundreds or thousands of sentences, but should belong only to a single cluster. Slight differences in the distributions across topics get multiplied thousands of times, creating enormous differences in relative probability.

We propose a novel clustering algorithm, that better represents the sense clustering task, and is orders of magnitude faster. This model is one of the major contributions of this work, and it may find application in other domains.

It is based on Positive Pointwise Mutual Information (PPMI), which was shown to be implicitly related to popular semantic vector models (Levy and Goldberg 2014b). Indeed, the PPMI vectors of word context even perform well on word analogy tasks, so they seem to be extremely useful semantic representations. We generate a PPMI vector for each verb sense using the (fixed) counts of syntactic patterns. During sampling, we compute the count matrices of syntactic patterns assigned to each cluster, so we can easily generate a PPMI vector for each cluster, as well.

Then, given cluster PPMI vectors $\Theta_i, i \in [1,\ldots,K]$, and sense vector $x_s$

$$P(x_s|\Theta_k) \propto \exp\Big(\frac{x_s \cdot \Theta_k}{\tau}\Big), \tag{4}$$

where $\tau > 0$ is a parameter dictating "temperature", and the probability of assigning sense $s$ to cluster $k$, with cluster sizes $C_k(*)$, is

$$P(k|s,\Theta,C) \propto P(k|C_k(*))P(x_s|\Theta_k). \tag{5}$$

Note that computation of the left factor is given by Equation (1).

We maintain exactly the same count matrices maintained in LDA and the Dirichlet-Multinomial mixture, so the only new overhead is the computation of PPMI vectors from these matrices. This computation is more effort than the simple smoothing and normalization of the Dirichlet-Multinomial, but by batching samples before updates, this operation is infrequent. During our testing, we observed that quality does not suffer as the batch size increases, so we only update the vectors once at the end of each sampling iteration.

Once the cluster vectors are computed, Equation (4) can be computed at once for an entire batch, and for all topics, with a simple matrix computation. There's no need to add smoothing to the PPMI vectors, and the matrices are sparse. In the end, the runtime is dramatically lower than the Dirichlet-Multinomial model, even with the added overhead of computing PPMI.

| Corpus | Algorithm | Verb Sense (Micro) | | | Verb Sense (Macro) | | |
|---|---|---|---|---|---|---|---|
| | | mPU | iPU | F1 | mPU | iPU | F1 |
| Gigaword | Baseline | **85.08** | 20.44 | 32.96 | **71.92** | 38.72 | 50.34 |
| | mLDA-100 | 81.59 | **43.06** | **56.37** | 71.35 | **54.07** | **61.52** |
| | mLDA-200 | 80.29 | 40.62 | 53.95 | 68.16 | 50.07 | 57.73 |
| Google | mLDA-100 | **78.84** | **27.67** | **40.97** | **61.40** | **47.85** | **53.79** |
| | mLDA-200 | 75.21 | 26.04 | 38.68 | 57.01 | 44.21 | 49.80 |

Table 1: Sense induction accuracy, on the Gigaword (**Gigaword**) and Google Books syntactic n-gram **Google** corpora. **mLDA-100** refers to the modified LDA algorithm run with 100 topics, **mLDA-200** uses 200 topics. We include maximum a posteriori assignment from the published verb-specific models as a baseline (Kawahara, Peterson, and Palmer 2014), but this baseline is only available on the Gigaword corpus. The highest scores achieved by any model, on each corpus, are highlighted.

| Algorithm | `slot` features | | | `pattern` features | | |
|---|---|---|---|---|---|---|
| | mPU | iPU | F1 | mPU | iPU | F1 |
| Prior | 57.0 | 28.1 | 37.6 | 57.0 | 28.1 | 37.6 |
| None | 46.2 | **48.1** | 47.1 | 46.2 | 48.1 | 47.1 |
| D-M | $48.6 \pm 0.7$ | $47.7 \pm 0.9$ | $48.2 \pm 1.5$ | $36.3 \pm 0.7$ | **48.2** $\pm 6.0$ | $41.4 \pm 1.8$ |
| PPMI | **52.5** $\pm 0.8$ | $47.34 \pm 4.8$ | **49.8** $\pm 1.6$ | **60.6** $\pm 0.8$ | $46.60 \pm .04$ | **52.7** $\pm 0.2$ |

Table 2: Verb clustering accuracy, for both algorithms, on verb senses from the **Gigaword** dataset. **D-M** is the Dirichlet Multinomial model, and **PPMI** is the novel model proposed here. We include the published model of prior work (Kawahara, Peterson, and Palmer 2014) as a baseline, as well as **none**, which is the baseline where we skip the second-step clustering and simply use the shared topics from the sense-induction step as clusters. The highest scores achieved by any model are in bold face. Baseline scores are duplicated in both columns.

Pseudo-code for the PPMI-vector clustering algorithm is given in Algorithm 3

**Algorithm 3** Clustering with Exponential Mixture of PPMI Vectors

1: Compute $X$, the PPMI vectors for input matrix of sense-syntax counts
2: Assign senses to random clusters and compute counts matrices
3: Compute and normalize $Y$, the PPMI vectors for assigned clusters
4: **for** iteration in range($num\_iters$) **do**
5: Compute probabilities by Equation (5), using $\langle X \cdot Y \rangle$
6: Assign new topics to senses and compute counts matrices
7: Re-compute and normalize $Y$
8: **end for**

In Table 4, we report runtimes for comparison. Runtimes are measured in seconds, processed on the same single machine with roughly equivalent optimization. A few patterns in the table are worth mentioning. First, the accuracy of the clusters induced by the modified LDA is surprisingly high. On the Gigaword corpus, the mLDA clusters outperform the predictions of the baseline, prior state-of-the-art. The only model that dramatically outperforms this one-shot clustering is the PPMI model using pattern features. Second, the PPMI model always performs better with pattern features over slot features on the same data. Third, the PPMI model is more than an order of magnitude faster.

## 5 Implementational Notes and Observations

Prior work (Kawahara, Peterson, and Palmer 2014) proposed a model based on a Dirichlet process, but we used a fixed-size Dirichlet distribution for both sense induction and sense clustering. By using a Dirichlet with capacity $C$ that far exceeded the final number of clusters used, and fixing the concentration parameters at $\alpha/C$, we end up with a truncated approximation to the Dirichlet Process (Kurihara, Welling, and Teh 2007; Ishwaran and Zarepour 2002). This difference should have a minimal impact on performance. Further, in Equation (2), we posit that the probability of choosing a cluster in the Dirichlet-Multinomial mixture is proportional to the product of all independent observations. In practice, verb senses may have thousands of separate observations, and small differences in the topic probability across clusters tended to grow into extreme differences when amplified to such high powers. Our performance was much better, and much closer to the performance of prior work, when we used the geometric mean of probabilities, rather than the raw product. This also helps ensure that the Dirichlet prior over cluster sizes behaves on the same scale as it does in a traditional topic model. For individual instances, with only a small number of observed slots, this doesn't seem to make such a large difference.

### 5.1 Parameter choices in the PPMI mixture

The parameter $\tau$ is important to the convergence properties of the PPMI-based mixture model. It is the same form as the well-known softmax, and as the temperature goes from zero toward infinity, the resulting distribution switches from assigning all probability to the maximum of observed dot products, toward a uniform distribution. Essentially, it governs the

| Algorithm | slot features | | | pattern features | | |
|---|---|---|---|---|---|---|
| | mPU | iPU | F1 | mPU | iPU | F1 |
| None | 35.04 | 30.22 | 32.45 | 35.04 | 30.22 | 32.45 |
| D-M | **44.82** | 30.15 | **36.05** | **45.96** | 30.25 | **36.49** |
| PPMI | 14.05 | **83.71** | 24.07 | 19.50 | **57.99** | 29.18 |

Table 3: Verb clustering accuracy, for both algorithms, on verb senses from the **Google** Books syntactic n-grams dataset. **D-M** is the Dirichlet Multinomial model, and **PPMI** is the novel model proposed here. As a baseline, we include **none**, which is the result if we were to skip the second-step clustering and simply use the shared topics from the sense-induction step as clusters. The highest scores achieved for each feature set are in bold face. Baseline scores are duplicated in both columns.

| Dataset | Features | D-M runtime | PPMI runtime |
|---|---|---|---|
| Gigaword-100 | slots | 7400 | 160 |
| Gigaword-100 | patterns | 6600 | 280 |
| Gigaword-200 | slots | 5900 | 270 |
| Gigaword-200 | patterns | 9400 | 320 |
| Google-100 | slots | 6100 | 110 |
| Google-100 | patterns | 4000 | 150 |
| Google-200 | slots | 7800 | 110 |
| Google-200 | patterns | 4900 | 140 |

Table 4: Verb clustering runtime (in seconds) on automatically induced senses. The dataset names indicate the corpus and the number of topics used in the sense induction step.

extent to which small differences in dot product produce large differences in probability. We found that $\tau \in [0.01, 1]$ produced reasonable results. Lower temperature values caused the model to make dramatic reassignments frequently, converging quickly, but exhibiting occasional, dramatic shifts of the cluster vectors even after many iterations. Larger temperature settings made smaller, more stable steps, and used fewer clusters.

In order to determine the final cluster for a particular instance or verb sense, we looked at their probability under the produced models and made maximum-likelihood assignments.

The parameter $\alpha$ behaves exactly as it does in any Dirichlet mixture. We tuned it for distinct datasets, but the same setting worked well for both algorithms.

### 5.2 When not to use the PPMI mixture

In Tables 2 and 3, we see different stories about the effectiveness of the PPMI mixture. On the Gigaword corpus, it outperforms all other models. But on the Google Books syntactic n-grams, it does not perform well. In this section, we hypothesize why this happens.

The mPU/iPU tradeoff is governed by the number of clusters the model preferred, and depends more on $\tau$ than on $\alpha$. On the Google Books syntactic n-grams corpus, when setting $\tau > 0.1$ the model tended to use only one or two clusters, extremely favoring iPU. As $\tau$ was lowered progressively, the model used more clusters, and purity increased, until $\tau \approx 0.01$. Lowering $\tau$ further is impractical. It requires a more complex implementation to avoid numerical overflow after the exponential, but also, the signal from Equation (4) overwhelms the Dirichlet prior entirely. Functionally, the knob that governs the mPU/iPU tradeoff was at its extreme setting, and it still hadn't reached a region of acceptable peformance.

This issue seems to come up only on the Google Books syntactic n-grams corpus, and is worse using `slot` features. Together, these results give clues as to the suitability of the PPMI mixture model.

### 5.3 Performance on Google Books corpus

The Google Books syntactic n-grams makes poorer automatic sense distinctions, along with poorer verb clusters. This runs counter to the intuition that a larger corpus should prove more accurate for distributional models.

The Google Books corpus is nearly two orders of magnitude larger than the English Gigaword corpus. However, it is functionally smaller because it is so heavily trimmed. The number of distinct lexicalized syntactic structures is only roughly $1.5\times$ the number of distinct structures in Gigaword. Pruning of low-frequency items may remove noise, but it also has a dramatic smoothing effect. It seems that, at least for this purpose, some important signal was lost along with the noise.

This pruning removes many of the sentences that contribute to finer-grained description of automatic senses, as well. Since the `slot` features are broad and highly shared across senses, there doesn't seem to be enough discriminative power for the PPMI mixture to tell senses apart. The Dirichlet-Multinomial performs better at making these distinctions, or at least is able to reach a more sensible tradeoff between purity and inverse purity, on this dataset, but even so the `pattern` features seem to help somewhat.

## 6 Qualitative Cluster Analysis

VerbNet is built on a combination of linguistic theory and evidence. The linguistic theory points to the distribution of syntactic patterns and selectional preferences as being highly relevant to semantic clustering. This paper makes those explicit features, and runs an unsupervised clustering.

However, a qualitative analysis of the clusters suggests that semantic generalization is still a difficult task. We use SemLink instances for evaluation, but these are dominated by a small number of highly-frequent verbs, and typically only two or three automatically-induced senses are selected for the SemLink instances. The majority class, and purity, of most learned clusters are dominated by a single automatic sense assignment. It is difficult to find any VerbNet class well-represented by the induced clusters that also contains a

significant portion of SemLink instances from more than one verb.

For example, one learned cluster contains only the verbs "rise" and "find", both of which are well-represented in SemLink. However, the SemLink instances do not belong together, and there are many more "rise" instances. The instances of "rise" in this cluster belong to VerbNet class 45.6, and it has a relatively high purity. But all instances of "find" belong to entirely different VerbNet classes, 13.5.1 or 18.4. Other high-scoring clusters tend to follow this pattern.

Another cluster, one that has a lower purity, does show generalization: it correctly clusters instances of "correct", "accumulate", and "grab" from VerbNet class 13.5.2; it correctly joins distinct senses of "introduce" from VerbNet class 22.2 that the sense induction step separated incorrectly; and correctly clusters instances of "get", "switch", and "turn" that belong to VerbNet class 26.6.2. However, these three correct generalizations should not be part of the same cluster.

SemLink provides a starting point for evaluation, but leaves many questions unanswered when seeking to understand the behavior of these models. It may be necessary to design a task to measure coherence and generalization more directly. Use of the verb clusters in a semantic task (semantic role labeling, translation, or even sentiment analysis) would also provide a more compelling argument. These investigations are left to future work.

## 7 Future Work

The novel PPMI-vector mixture takes advantage of a few properties of the verb-clustering dataset, but is applicable in other domains, as long as those properties hold. The most important properties, as we see it, are: the items to be clustered are not singletons, but are collections of multiple, discrete units; the distributions for the collections are distinctive; and we believe they should be grouped into a small number of clusters. An example dataset with these properties is word sense disambiguation. Word2Vec and PPMI provided an inspiration for this mixture, but Word2Vec does not explicitly account for polysemy at all. There are multiple approaches to this problem in the literature (Chen, Liu, and Sun 2014; Trask, Michalak, and Liu 2015; Neelakantan et al. 2015), so we can test the effectiveness of the PPMI mixture against strong baselines.

The novel model should be more closely investigated - parameter settings, such as varying temperature over time to balance speed of convergence with stability, could play a key role in performance. Also, by analogy to Word2Vec, the Bayesian PPMI vectors should be useful semantic representations in their own right. We are looking forward to exploring this connection in more detail.

The models employed here rely on a dependency-based syntactic parse of a large corpus, but no corpus- or language-specific features, so application of this framework to resource-poor languages is also a promising direction for future work.

## 8 Conclusion

We have proposed a new framework for verb sense induction and clustering, that captures the distributional hypothesis for verbs, and achieved state-of-the-art results on replication of VerbNet, using a Bayesian model. Our framework includes a novel exponential mixture model, which takes advantage of recent advances in vector representations of words and senses.

Our recommendations for verb clustering are as follows. First, share topics across verbs, following the lines of LDA. This will automatically improve accuracy, and benefit representations for infrequent verbs. Second, use the entire corpus. Pruning of infrequent constructions may harm, rather than help, with sense induction and verb clustering. Third, run the verb clustering using the PPMI mixture with `pattern` features. Tune $\tau$ to give an appropriate number of clusters, if possible, and otherwise commence to run the Dirichlet-multinomial model.

Our work proposes a PPMI-based exponential mixture model whose sampler uses the same sufficient statistics as LDA. It performs well, and is much faster, when each atom in the mixture has a rich and unique context.

## Acknowledgements

Susan W. Brown was extremely helpful in the qualitative analysis, and we would like to thank her for her time and expertise. This work was completed while the lead author was employed by TrustYou, GmbH., and the authors are grateful for both the financial support and freedom to pursue independent studies. Additional thanks are due to the members of the Machine Learning Research Group at Oracle Labs, for helpful feedback on initial drafts.

## References

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet allocation. *the Journal of Machine Learning Research* 3:993–1022.

Chen, X.; Liu, Z.; and Sun, M. 2014. A unified model for word sense representation and disambiguation. In *EMNLP*, 1025–1035.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41(6):391.

Ferguson, T. S. 1973. A bayesian analysis of some nonparametric problems. *The annals of statistics* 209–230.

Firth, J. R. 1957. A synopsis of linguistic theory 1930-55. 1952-59:1–32.

Giuglea, A.-M., and Moschitti, A. 2006. Semantic role labeling via framenet, verbnet and propbank. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 929–936. Association for Computational Linguistics.

Goldberg, Y., and Orwant, J. 2013. A dataset of syntactic-ngrams over time from a very large corpus of english books.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864. ACM.

Ishwaran, H., and Zarepour, M. 2002. Exact and approximate sum representations for the dirichlet process. *Canadian Journal of Statistics* 30(2):269–283.

Kawahara, D.; Peterson, D. W.; and Palmer, M. 2014. A step-wise usage-based method for inducing polysemy-aware verb classes. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL2014)*.

Kipper-Schuler, K. 2005. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. Dissertation, University of Pennsylvania.

Korhonen, A.; Krymolowski, Y.; and Marx, Z. 2003. Clustering polysemic subcategorization frame distributions semantically. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL2003)*, 64–71.

Kurihara, K.; Welling, M.; and Teh, Y. W. 2007. Collapsed variational dirichlet process mixture models. In *IJCAI*, volume 7, 2796–2801.

Levin, B. 1993. *English verb classes and alternations: A preliminary investigation*. The University of Chicago Press.

Levy, O., and Goldberg, Y. 2014a. Dependency-based word embeddings. In *ACL (2)*, 302–308.

Levy, O., and Goldberg, Y. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, 2177–2185.

Liu, Z.; Zhang, Y.; Chang, E. Y.; and Sun, M. 2011. Plda+: Parallel latent dirichlet allocation with data placement and pipeline processing. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2(3):26.

Materna, J. 2012. Lda-frames: An unsupervised approach to generating semantic frames. In *International Conference on Intelligent Text Processing and Computational Linguistics*, 376–387. Springer.

Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, 746–751.

Neelakantan, A.; Shankar, J.; Passos, A.; and McCallum, A. 2015. Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654*.

Ng, P. 2017. dna2vec: Consistent vector representations of variable-length k-mers. *arXiv preprint arXiv:1701.06279*.

Palmer, M.; Gildea, D.; and Xue, N. 2010. Semantic role labeling. *Synthesis Lectures on Human Language Technologies* 3(1):1–103.

Palmer, M. 2009. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the generative lexicon conference*, 9–15. Pisa Italy.

Parker, R.; Graff, D.; Kong, J.; Chen, K.; and Maeda, K. 2011. English gigaword fifth edition ldc2011t07.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, 1532–1543.

Řehůřek, R., and Sojka, P. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA. http://is.muni.cz/publication/884893/en.

Teh, Y. W.; Jordan, M. I.; Beal, M. J.; and Blei, D. M. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association* 101(476).

Trask, A.; Michalak, P.; and Liu, J. 2015. sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings. *arXiv preprint arXiv:1511.06388*.

Wu, S., and Palmer, M. 2015. Can selectional preferences help automatic semantic role labeling? In ** SEM@ NAACL-HLT*, 222–227.