Modeling Word Learning

Su Wang

DEPARTMENT OF LINGUISTICS
UNIVERSITY OF TEXAS AT AUSTIN

September 13, 2016

1 Introduction

In this research report, I introduce two Bayesian models for word property learning, where Model I is count-based and Model II is topic-based. The learning objective for an unknown word (e.g. *alligator*) is to obtain a probability distribution over a set of properties (e.g. `an_animal`, `has_4_legs`, etc.) and evaluate the top k most probable properties for the unknown word against human-judged properties for it. Both models make use of the word norms in McRae et al. (2005), where properties are elicited from human participants for 541 nouns in English. The production frequencies for the properties[1] recorded are also exploited.

For the count-based Model I, I learn the property distributions of unknown words by first creating count vectors over properties for each of a selected set of verbs and update the unknown word's property count vector with the corresponding count vectors of the subset of the verbs that have been observed to have appeared as the word's predicates in a corpus. The updated count vector is then used as the parameter of a Dirichlet distribution to analytically compute a posterior predictive distribution over properties for the unknown word.

For the topic-based Model I, a set of *pseudo-documents* are generated for each of a set of verb-role[2] pairs. Standard topic modeling is then applied to the documents to obtain two distributions for each verb-role pair:

(i) A distribution over topics[3].

(ii) A distribution over properties for each topic.

A new distribution over properties is then computed for each verb-role pair using the two distributions. To learn the property distribution of an unknown

[1] i.e. The frequency of production for each properties from 30 participants.

[2] A role corresponds to one of a verb's *core argument* position, e.g. subject, direct object and indirect object (Van Valin & LaPolla (1997)).

[3] Note that the *topic*s here can be roughly interpreted as latent semantic dimensions.

word, I initialize a vector over properties for the word and update it using the property distributions computed for the set of its observed predicates, and finally use the updated vector as the parameter of a Dirichlet distribution to compute a posterior predictive distribution over properties.

The rest of the report is organized as follows: Section 2 introduces the notations used for the models; Section 3 describes the two models in detail; Section 4 describes the evaluation schemes for the models; and finally Section 5 details the algorithms for the implemetation of the models.

2 Notations

- $\mathcal{C}$: The corpus used for model training. In this case, it refers in particular to the British National Corpus (BNC).
- W: The set of word norms in McRae et al. (2005).
- Q: The set of properties (ibid.).
- V: The set of verbs which have evidenced to be predicates of W in $\mathcal{C}$.
- u: An "unknown" word, $u \in W$. On choosing an unknown word from W, W is then set as $W = W - \{u\}$.
- w: A single word norm, $w \in W$.
- v: A single verb, $v \in V$.
- r: A core argument position of a verb, $r \in \{\text{SUBJECT}, \text{OBJECT}\}$.
- (v, r): Verb-role pairs, $(v, r) \in VR$.
- $c_{(v,r)}$: A property count vector for $c_{(v,r)} \in C$, where $c_{(v,r)} \in \mathbb{R}^{|Q|}$.
- $d_{(v,r)}$: A pseudo-document corresponding to a verb-role pair (v, r), $d_{(v,r)} \in D$.
- z: A topic, $z \in T$.
- $\theta_{(v,r)}$: A probability distribution over topics for a verb-role pair (v, r).
- $\phi_{(v,r),z}$: A probability distribution over properties wrt. a verb-role pair (v, r) and a topic z.
- γ: A parameter vector for a Dirichlet distribution. It is initialize as $\gamma = \langle 1, ..., 1 \rangle, \gamma \in \mathbb{R}^{|Q|}$.
- ξ: A posterior predictive distribution over properties.

3 Model Description

3.1 Model I: Count-Based

To extract information from $\mathcal{C}$ for model training, I first parse the set of all sentences of $\mathcal{C}$[4], and then gather V from the parsed corpus using W, which is imported from McRae et al. dataset, from which I also collect Q. With W, V and Q, I then use the corpus and the McRae norm data to obtain two mappings:

(i) $f : VR \rightarrow W$, a mapping from each (v, r) pair to a list[5] of word norms $Set(\mathbf{w}_{list}) \subset W$.

(ii) $g : W \rightarrow Q$, a mapping from each w to a set of properties $\mathbf{q} \subset Q$.

The composite mapping $h = f \circ g : VR \rightarrow Q$ can then be obtained from f and g, a mapping from each (v, r) pair to a list of properties $\mathbf{q}_{list}$, where repeated ws lead to repetition in properties. From h, I am able to construct property count vectors $c_{(v,r)} \in C$ for all $(v, r) \in VR$.

On encounter an unknown word u, and a set of sentences where u appears in some core argument position, I first obtain the set of verb-role pairs $\mathbf{vr}$, and initialize a "zero-observation" parameter vector γ for u: $\gamma = \langle 1, ..., 1 \rangle$. I then update[6] γ by $\gamma := \gamma + c_{(v,r)}$ for all $(v, r) \in \mathbf{vr}$. Finally, the updated γ is used as the parameter in the distribution $Dir(\gamma)$, from which I analytically compute a posterior predictive distribution ξ over properties.

The "learing result" for u (with its $\mathbf{vr}$ obtained from $\mathcal{C}$), then, is the distribution ξ from which we are able to extract, e.g., top k most probable properties for u.

3.2 Model II: Topic-Based

The same procedure as in Model I is applied to obtain the mapping $f : VR \rightarrow W$. I then compute a different mapping $g : W \rightarrow p(Q)$, where $p(Q)$ is a probability distribution over properties computed from McRae et al. (2005)'s production frequency. Specifically, each q in the set of properties elicited for $w \in W$ is normalized to a probability $p(q)$ by dividing it by the sum of all the production frequency counts for w.

To make a set of pseudo-documents D, the following procedure is applied: For each verb-role pair $(v, r) \in VR$, I use f to return the set of corresponding word norms $\mathbf{w} \subset W$, and for each $w \in \mathbf{w}$ I sample a property using g and append it to $d_{(v,r)}$. Applying standard topic modeling method[7] to D, I obtain:

(i) $p_r(z \mid v) \sim \theta_{(v,r)}$, where $z \sim T$.

(ii) $p_r(q \mid z, v) \sim \phi_{(v,r),z}$.

[4] Either Stanford CoreNLP 3.6.0 or SpaCy 1.8.0

[5] Note this is a list where word norms are allowed to repeat.

[6] Pointwise addition of two vectors.

[7] e.g. Hoffman et al. (2010), implemented in python package Gensim.

The two distributions are then used to compute a probability distribution over properties for each (v, r) pair:

$$p_r(q \mid v) = \sum_{z \in T} p_r(q \mid z, v) p_r(z \mid v)$$

Now, on encountering an unknown word u, and its set of (v, r) pairs **vr** (as in Model I), I first initialize γ as before, and then update γ by $\gamma := \gamma + \vec{p}_{(v,r)}$, where $\vec{p}_{(v,r)}$ is a vector in $\mathbb{R}^{|Q|}$ obtained by computing $p_r(q \mid v)$, the cell corresponding to q, for all $q \in Q$. The update is applied for all $(v, r) \in \mathbf{vr}$. Finally, I compute the posterior predictive distribution ξ from $Dir(\gamma)$ as before.

4 Evaluation

I adopt two evaluation schemes for the models. First consider the evaluation for a single unknown word u:

(i) Scheme 1: From the posterior predictive distribution ξ of u, extract the top k most probably properties. Each property accounts for $\frac{1}{k}$ of the total accuracy for u.

(ii) Scheme 2: Extract the top k properties as in Scheme 1. If one of the top k properties is in $g(u)$ (cf. Section 3.1), the prediction for u is considered accuracy (i.e. accuracy score set as 1), otherwise inaccuracy (i.e. accuracy score set as 0).

For the entire set of word norms W, I use the *leave-one-out (LOO)* evaluation method: For all $u \in W$, train a model on $W - \{u\}$, and evaluate the model using either Scheme 1 or 2, and compute the average accuracy[8].

5 Algorithm

5.1 Model I

1. Using $\mathcal{C}$ and W, obtain V;
2. Using W, V and Q, obtain $f : VR \to W$ and $g : W \to Q$, and subsequently $h = f \circ g : VR \to Q$;
3. For each unknown word $u \in W$, for each verb-role pair $(v, r) \in \mathbf{vr}$,
   i Initialize γ;
   ii Obtain $c_{(v,r)}$ from h;
   iii Update $\gamma := \gamma + c_{(v,r)}$.
4. Evaluate as described in Section 4.

[8]For an extension, precision, recall and F1 score can also be computed.

5.2 Model II

1. Using $\mathcal{C}$ and W, obtain V;
2. Using W, V and Q, obtain $f : VR \rightarrow W$ and $g : W \rightarrow p(Q)$;
3. Make pseudo-documents $d_{(v,r)} \in D$ using f and g, and then run topic model;
4. For each verb-role pair $(v, r) \in VR$, using its corresponding $p_r(z \mid v) \sim \theta_{(v,r)}$ and $p_r(q \mid z, v) \sim \phi_{(v,r),z}$ to compute the vector $\vec{p}_{(v,r)}$, where each cell corresponding to p is $p_r(q \mid v) = \sum_{z \in T} p_r(q \mid z, v) p_r(z \mid v)$;
5. For each unknown word $u \in W$, for each verb-role pair $(v, r) \in \mathbf{vr}$,
   i Initialize γ;
   ii Obtain $\vec{p}_{(v,r)}$;
   iii Update $\gamma := \gamma + \vec{p}_{(v,r)}$.
6. Evaluate as described in Section 4.