Inferring Selectional Preferences from Part-Of-Speech N-grams

Hyeju Jang and Jack Mostow
Project LISTEN (www.cs.cmu.edu/~listen), School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
hyejuj@cs.cmu.edu, mostow@cs.cmu.edu

Abstract

We present the PONG method to compute selectional preferences using part-of-speech (POS) N-grams. From a corpus labeled with grammatical dependencies, PONG learns the distribution of word relations for each POS N-gram. From the much larger but unlabeled Google N-grams corpus, PONG learns the distribution of POS N-grams for a given pair of words. We derive the probability that one word has a given grammatical relation to the other. PONG estimates this probability by combining both distributions, whether or not either word occurs in the labeled corpus. PONG achieves higher average precision on 16 relations than a state-of-the-art baseline in a pseudo-disambiguation task, but lower coverage and recall.

1 Introduction

Selectional preferences specify plausible fillers for the arguments of a predicate, e.g., *celebrate*. Can you celebrate a birthday? Sure. Can you celebrate a pencil? Arguably yes: *Today the Acme Pencil Factory celebrated its one-billionth pencil.* However, such a contrived example is unnatural because unlike *birthday*, *pencil* lacks a strong association with *celebrate*. How can we compute the degree to which *birthday* or *pencil* is a plausible and typical object of *celebrate*?

Formally, we are interested in computing the probability Pr(*r* | *t*, *R*), where (as Table 1 specifies), *t* is a target word such as *celebrate*, *r* is a word possibly related to it, such as *birthday* or *pencil*, and *R* is a possible relation between them, whether a semantic role such as the agent of an action, or a grammatical dependency such as the object of a verb. We call *t* the "target" because originally it referred to a vocabulary word targeted for instruction, and *r* its "relative."

Notation	Description
R	a relation between words
t	a target word
r, *r'*	possible relatives of *t*
g	a word N-gram
g_i and g_j	i^{th} and j^{th} words of *g*
p	the POS N-gram of *g*

Table 1: Notation used throughout this paper

Previous work on selectional preferences has used them primarily for natural language analytic tasks such as word sense disambiguation (Resnik, 1997), dependency parsing (Zhou et al., 2011), and semantic role labeling (Gildea and Jurafsky, 2002). However, selectional preferences can also apply to natural language generation tasks such as sentence generation and question generation. For generation tasks, choosing the right word to express a specified argument of a relation requires knowing its connotations – that is, its selectional preferences. Therefore, it is useful to know selectional preferences for many different relations. Such knowledge could have many uses. In education, they could help teach word connotations. In machine learning they could help computers learn languages. In machine translation, they could help generate more natural wording.

This paper introduces a method named PONG (for Part-Of-Speech N-Grams) to compute selectional preferences for many different relations by combining part-of-speech information and Google N-grams. PONG achieves higher precision on a pseudo-

disambiguation task than the best previous model (Erk et al., 2010), but lower coverage.

The paper is organized as follows. Section 2 describes the relations for which we compute selectional preferences. Section 3 describes PONG. Section 4 evaluates PONG. Section 5 relates PONG to prior work. Section 6 concludes.

2 Relations Used

Selectional preferences characterize constraints on the arguments of predicates. Selectional preferences for semantic roles (such as agent and patient) are generally more informative than for grammatical dependencies (such as subject and object). For example, consider these semantically equivalent but grammatically distinct sentences:

Pat opened the door.
The door was opened by Pat.

In both sentences the agent of *opened*, namely *Pat*, must be capable of opening something – an informative constraint on *Pat*. In contrast, knowing that the grammatical subject of *opened* is *Pat* in the first sentence and *the door* in the second sentence tells us only that they are nouns.

Despite this limitation, selectional preferences for grammatical dependencies are still useful, for a number of reasons. First, in practice they approximate semantic role labels. For instance, typically the grammatical subject of *opened* is its agent. Second, grammatical dependencies can be extracted by parsers, which tend to be more accurate than current semantic role labelers. Third, the number of different grammatical dependencies is large enough to capture diverse relations, but not so large as to have sparse data for individual relations. Thus in this paper, we use grammatical dependencies as relations.

A parse tree determines the basic grammatical dependencies between the words in a sentence. For instance, in the parse of *Pat opened the door*, the verb *opened* has *Pat* as its subject and *door* as its object, and *door* has *the* as its determiner. Besides these basic dependencies, we use two additional types of dependencies.

Composing two basic dependencies yields a *collapsed dependency* (de Marneffe and Manning, 2008). For example, consider this sentence:

The airplane flies in the sky.

Here *sky* is the prepositional object of *in*, which is the head of a prepositional phrase attached to *flies*. Composing these two dependencies yields the collapsed dependency *prep_in* between *flies* and *sky*, which captures an important semantic relation between these two content words: *sky* is the location where *flies* occurs. Other function words yield different collapsed dependencies. For example, consider these two sentences:

The airplane flies over the ocean.
The airplane flies and lands.

Collapsed dependencies for the first sentence include *prep_over* between *flies* and *ocean*, which characterizes their relative vertical position, and *conj_and* between *flies* and *lands*, which links two actions that an airplane can perform. As these examples illustrate, collapsing dependencies involving prepositions and conjunctions can yield informative dependencies between content words.

Besides collapsed dependencies, PONG infers inverse dependencies. Inverse selectional preferences are selectional preferences of arguments for their predicates, such as a preference of a subject or object for its verb. They capture semantic regularities such as the set of verbs that an agent can perform, which tend to outnumber the possible agents for a verb (Erk et al., 2010).

3 Method

To compute selectional preferences, PONG combines information from a limited corpus labeled with the grammatical dependencies described in Section 2, and a much larger unlabeled corpus. The key idea is to abstract word sequences labeled with grammatical relations into POS N-grams, in order to learn a mapping from POS N-grams to those relations. For instance, PONG abstracts the parsed sentence *Pat opened the door* as NN VB DT NN, with the first and last NN as the subject and object of the VB. To estimate the distribution of POS N-grams containing particular target and relative words, PONG POS-tags Google N-grams (Franz and Brants, 2006).

Section 3.1 derives PONG's probabilistic model for combining information from labeled and unlabeled corpora. Section 3.2 and Section 3.3 describe how PONG estimates probabilities from each corpus. Section 3.4 discusses a sparseness problem revealed during probability estimation, and how we address it in PONG.

3.1 Probabilistic model

We quantify the selectional preference for a relative r to instantiate a relation R of a target t as the probability $\Pr(r \mid t, R)$, estimated as follows. By the definition of conditional probability:

$$\Pr(r \mid t, R) = \frac{\Pr(r, t, R)}{\Pr(t, R)}$$

We care only about the relative probability of different r for fixed t and R, so we rewrite it as:

$$\propto \Pr(r, t, R)$$

We use the chain rule:

$$= \Pr(R \mid r, t) \cdot \Pr(r \mid t) \cdot \Pr(t)$$

and notice that t is held constant:

$$\propto \Pr(R \mid r, t) \cdot \Pr(r \mid t)$$

We estimate the second factor as follows:

$$\Pr(r \mid t) = \frac{\Pr(t, r)}{\Pr(t)} = \frac{\text{freq}(t, r)}{\text{freq}(t)}$$

We calculate the denominator freq(t) as the number of N-grams in the Google N-gram corpus that contain t, and the numerator freq(t, r) as the number of N-grams containing both t and r.

To estimate the factor Pr(R | r, t) directly from a corpus of text labeled with grammatical relations, it would be trivial to count how often a word r bears relation R to target word t. However, the results would be limited to the words in the corpus, and many relation frequencies would be estimated sparsely or missing altogether; t or r might not even occur.

Instead, we abstract each word in the corpus as its part-of-speech (POS) label. Thus we abstract *The big boy ate meat* as DT JJ NN VB NN. We call this sequence of POS tags a POS N-gram. We use POS N-grams to predict word relations. For instance, we predict that in any word sequence with this POS N-gram, the JJ will modify (*amod*) the first NN, and the second NN will be the direct object (*dobj*) of the VB.

This prediction is not 100% reliable. For example, the initial 5-gram of *The big boy ate meat pie* has the same POS 5-gram as before. However, the *dobj* of its VB (*ate*) is not the second NN (*meat*), but the subsequent NN (*pie*). Thus POS N-grams predict word relations only in a probabilistic sense.

To transform Pr(R | r, t) into a form we can estimate, we first apply the definition of conditional probability:

$$\Pr(R \mid t, r) = \frac{\Pr(R, t, r)}{\Pr(t, r)}$$

To estimate the numerator Pr(R, t, r), we first marginalize over the POS N-gram p:

$$= \sum_p \frac{\Pr(R, t, r, p)}{\Pr(t, r)}$$

We expand the numerator using the chain rule:

$$= \sum_p \frac{\Pr(R \mid t, r, p) \cdot \Pr(p \mid t, r) \cdot \Pr(t, r)}{\Pr(t, r)}$$

Cancelling the common factor yields:

$$= \sum_p \Pr(R \mid p, t, r) \cdot \Pr(p \mid t, r)$$

We approximate the first term Pr(R | p, t, r) as Pr(R | p), based on the simplifying assumption that R is conditionally independent of t and r, given p. In other words, we assume that given a POS N-gram, the target and relative words t and r give no additional information about the probability of a relation. However, their respective positions i and j in the POS N-gram p matter, so we condition the probability on them:

$$\Pr(R \mid p, t, r) \approx \Pr(R \mid p, i, j)$$

Summing over their possible positions, we get

$$\Pr(R \mid r, t)$$

$$\approx \sum_p \sum_{i \neq j} \Pr(R \mid p, i, j) \cdot \Pr(p \mid t = g_i, r = g_j)$$

As Figure 1 shows, we estimate Pr(R | p, i, j) by abstracting the labeled corpus into POS N-grams. We estimate Pr(p | $t = g_i$, $r = g_j$) based on the frequency of partially lexicalized POS N-grams like DT JJ:*red* NN:*hat* VB NN among Google N-grams with t and r in the specified positions.

Sections 3.2 and 3.3 describe how we estimate Pr(R | p, i, j) and Pr(p | $t = g_i$, $r = g_j$), respectively. Note that PONG estimates relative rather than absolute probabilities. Therefore it cannot (and does not) compare them against a fixed threshold to make decisions about selectional preferences.

3.2 Mapping POS N-grams to relations

To estimate Pr(R | p, i, j), we use the Penn Treebank Wall Street Journal (WSJ) corpus, which is labeled with grammatical relations using the Stanford dependency parser (Klein and Manning, 2003).

To estimate the probability Pr(R | p, i, j) of a relation R between a target at position i and a relative at position j in a POS N-gram p, we compute what fraction of the word N-grams g with POS N-gram p have relation R between some target t and relative r at positions i and j:

$$\Pr(R \mid p, i, j) =$$

$$\frac{\text{freq}(g \ s.t. \text{POS}(g) = p \wedge \text{relation}(g_i, g_j) = R)}{\text{freq}(g \ s.t. \text{POS}(g) = p \wedge \text{relation}(g_i, g_j))}$$

3.3 Estimating POS N-gram distributions

Given a target and relative, we need to estimate their distribution of POS N-grams and positions.

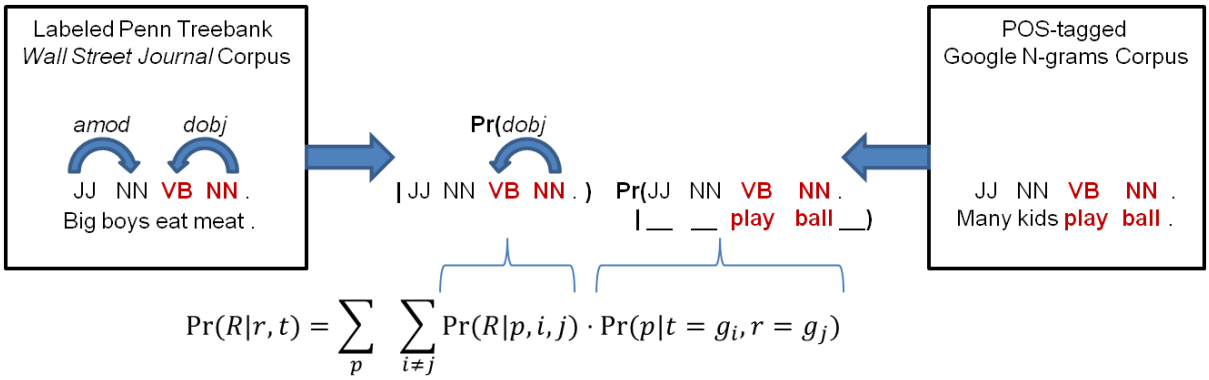

Figure 1: Overview of PONG.
From the labeled corpus, PONG extracts abstract mappings from POS N-grams to relations.
From the unlabeled corpus, PONG estimates POS N-gram probability given a target and relative.

A labeled corpus is too sparse for this purpose, so we use the much larger unlabeled Google N-grams corpus (Franz and Brants, 2006).

The probability that an N-gram with target *t* at position *i* and relative *r* at position *j* will have the POS N-gram *p* is:

$$\Pr(p \mid t = g_i, r = g_j) = \frac{\text{freq}(g \ s.t. \text{POS}(g) = p, g_i = t, g_j = r))}{\text{freq}(g \ s.t. g_i = t \wedge g_j = r)}$$

To compute this ratio, we first use a well-indexed table to efficiently retrieve all N-grams with words *t* and *r* at the specified positions. We then obtain their POS N-grams from the Stanford POS tagger (Toutanova et al., 2003), and count how many of them have the POS N-gram *p*.

3.4 Reducing POS N-gram sparseness

We abstract word N-grams into POS N-grams to address the sparseness of the labeled corpus, but even the POS N-grams can be sparse. For n=5, the rarer ones occur too sparsely (if at all) in our labeled corpus to estimate their frequency.

To address this issue, we use a coarser POS tag set than the Penn Treebank POS tag set. As Table 2 shows, we merge tags for adjectives, nouns, adverbs, and verbs into four coarser tags.

Coarse	Original
ADJ	JJ, JJR, JJS
ADVERB	RB, RBR, RBS
NOUN	NN, NNS, NNP, NNPS
VERB	VB, VBD, VBG, VBN, VBP, VBZ

Table 2: Coarser POS tag set used in PONG

To gauge the impact of the coarser POS tags, we calculated Pr(*r* | *t*, *R*) for 76 test instances used in an earlier unpublished study by Liu Liu, a former Project LISTEN graduate student. Each instance consists of two randomly chosen words in the WSJ corpus labeled with a grammatical relation. Coarse POS tags increased coverage of this pilot set – that is, the fraction of instances for which PONG computes a probability – from 69% to 92%.

Using the universal tag set (Petrov et al., 2011) as an even coarser tag set is an interesting future direction, especially for other languages. Its smaller size (12 tags vs. our 23) should reduce data sparseness, but increase the risk of over-generalization.

4 Evaluation

To evaluate PONG, we use a standard pseudo-disambiguation task, detailed in Section 4.1. Section 4.2 describes our test set. Section 4.3 lists the metrics we evaluate on this test set. Section 4.4 describes the baselines we compare PONG against on these metrics, and Section 4.5 describes the relations we compare them on. Section 4.6 reports our results. Section 4.7 analyzes sources of error.

4.1 Evaluation task

The pseudo-disambiguation task (Gale et al., 1992; Schutze, 1992) is as follows: given a target word *t*, a relation *R*, a relative *r*, and a random distracter *r'*, prefer either *r* or *r'*, whichever is likelier to have relation *R* to word *t*.

This evaluation does not use a threshold: just prefer whichever word is likelier according to the model being evaluated. If the model assigns only one of the words a probability, prefer it, based on the assumption that the unknown probability of the other word is lower. If the model assigns the same probability to both words, or no probability to either word, do not prefer either word.

4.2 Test set

As a source of evaluation data, we used the British National Corpus (BNC). As a common test corpus for all the methods we evaluated, we selected one half of BNC by sorting filenames alphabetically and using the odd-numbered files. We used the other half of BNC as a training corpus for the baseline methods we compared PONG to.

A test set for the pseudo-disambiguation task task consists of tuples of the form (*R, t, r, r'*). To construct a test set, we adapted the process used by Rooth et al. (1999) and Erk et al. (2010).

First, we chose 100 (*R, t*) pairs for each relation *R* at random from the test corpus. Rooth et al. (1999) and Erk et al. (2010) chose such pairs from a training corpus to ensure that it contained the target *t.* In contrast, choosing pairs from an unseen test corpus includes target words whether or not they occur in the training corpus.

To obtain a sample stratified by frequency, rather than skewed heavily toward high-frequency pairs, Erk et al. (2010) drew (*R, t*) pairs from each of five frequency bands in the entire British National Corpus (BNC): 50-100 occurrences; 101-200; 201-500; 500-1000; and more than 1000. However, we use only half of BNC as our test corpus, so to obtain a comparable test set, we drew 20 (*R, t*) pairs from each of the corresponding frequency bands in that half: 26-50 occurrences; 51-100; 101-250; 251-500; and more than 500.

For each chosen (*R, t*) pair, we drew a separate (*R, t, r*) triple from each of six frequency bands: 1-25 occurrences; 26-50; 51-100; 101-250; 251-500; and more than 500. We necessarily omitted frequency bands that contained no such triples. We filtered out triples where *r* did not have the most frequent part of speech for the relation *R*. For example, this filter would exclude the triple (*dobj, celebrate, the*) because a direct object is most frequently a noun, but *the* is a determiner.

Then, like Erk et al. (2010), we paired the relative *r* in each (*R, t, r*) triple with a distracter *r'* with the same (most frequent) part of speech as the relative *r*, yielding the test tuple (*R, t, r, r'*). Rooth et al. (1999) restricted distracter candidates to words with between 30 and 3,000 occurrences in BNC; accordingly, we chose only distracters with between 15 and 1,500 occurrences in our test corpus. We selected *r'* from these candidates randomly, with probability proportional to their frequency in the test corpus. Like Rooth et al. (1999), we excluded as distracters any actual relatives, i.e. candidates *r'* where the test corpus contained the triple (*R, t, r'*). Table 3 shows the resulting number of (*R, t, r, r'*) test tuples for each relation.

Relation R	# tuples for R	# tuples for R^T
advmod	121	131
amod	162	128
conj_and	155	151
dobj	145	167
nn	173	158
nsubj	97	124
prep_of	144	153
xcomp	139	140

Table 3: Test set size for each relation

4.3 Metrics

We report four evaluation metrics: precision, coverage, recall, and F-score. Precision (called "accuracy" in some papers on selectional preferences) is the percentage of all covered tuples where the original relative *r* is preferred. Coverage is the percentage of tuples for which the model prefers *r* to *r'* or vice versa. Recall is the percentage of all tuples where the original relative is preferred, i.e., precision times coverage. F-score is the harmonic mean of precision and recall.

4.4 Baselines

We compare PONG to two baseline methods.

EPP is a state-of-the-art model for which Erk et al. (2010) reported better performance than both Resnik's (1996) WordNet model and Rooth's (1999) EM clustering model. EPP computes selectional preferences using distributional similarity, based on the assumption that relatives are likely to appear in the same contexts as relatives seen in the training corpus. EPP computes the similarity of a potential relative's vector space representation to relatives in the training corpus.

EPP has various options for its vector space representation, similarity measure, weighting scheme, generalization space, and whether to use PCA. In re-implementing EPP, we chose the options that performed best according to Erk et al. (2010), with one exception. To save work, we chose not to use PCA, which Erk et al. (2010) described as performing only slightly better in the dependency-based space.

Relation	Target	Relative	Description
advmod	verb	adverb	Adverbial modifier
amod	noun	adjective	Adjective modifier
conj_and	noun	noun	Conjunction with "*and*"
dobj	verb	noun	Direct object
nn	noun	noun	Noun compound modifier
nsubj	verb	noun	Nominal subject
prep_of	noun	noun	Prepositional modifier
xcomp	verb	verb	Open clausal complement

Table 4: Relations tested in the pseudo-disambiguation experiment.
Relation names and descriptions are from de Marneffe and Manning (2008) except for *prep_of*.
Target and relative POS are the most frequent POS pairs for the relations in our labeled WSJ corpus.

Relation	Precision (%)			Coverage (%)			Recall (%)			F-score (%)		
	PONG	EPP	DEP	PONG	EPP	DEP	PONG	EPP	DEP	PONG	EPP	DEP
advmod	78.7	-	98.6	72.1	-	69.2	56.7	-	68.3	65.9	-	80.7
advmod^{T}	89.0	71.0	97.4	69.5	100	59.5	61.8	71.0	58.0	73.0	71.0	72.7
amod	78.8	-	99.0	90.1	-	61.1	71.0	-	60.5	74.7	-	75.1
amod^{T}	84.1	74.0	97.3	83.6	99.2	57.0	70.3	73.4	55.5	76.6	73.7	70.6
conj_and	77.2	74.2	100	73.6	100	52.3	56.8	74.2	52.3	65.4	74.2	68.6
conj_and^{T}	80.5	70.2	97.3	74.8	100	49.7	60.3	70.2	48.3	68.9	70.2	64.6
dobj	87.2	80.0	97.7	80.7	100	60.0	70.3	80.0	58.6	77.9	80.0	73.3
dobj^{T}	89.6	80.2	98.1	92.2	100	64.1	82.6	80.2	62.9	86.0	80.2	76.6
nn	86.7	73.8	97.2	95.3	99.4	63.0	82.7	73.4	61.3	84.6	73.6	75.2
nn^{T}	83.8	79.7	99.0	93.7	100	60.8	78.5	79.7	60.1	81.0	79.7	74.8
nsubj	76.1	77.3	100	69.1	100	42.3	52.6	77.3	42.3	62.2	77.3	59.4
nsubj^{T}	78.5	66.9	95.0	86.3	100	48.4	67.7	66.9	46.0	72.7	66.9	62.0
prep_of	88.4	77.8	98.4	84.0	100	44.4	74.3	77.8	43.8	80.3	77.8	60.6
prep_of^{T}	79.2	76.5	97.4	81.7	100	50.3	64.7	76.5	49.0	71.2	76.5	65.2
xcomp	84.0	61.9	95.3	85.6	100	61.2	71.9	61.9	58.3	77.5	61.9	72.3
xcomp^{T}	86.4	78.6	98.9	89.3	100	63.6	77.1	78.6	62.9	81.5	78.6	76.9
average	**83.0**	**74.4**	**97.9**	**82.6**	**99.9**	**56.7**	**68.7**	**74.4**	**55.5**	**75.0**	**74.4**	**70.5**

Table 5: Coverage, Precision, Recall, and F-score for various relations; R^{T} is the inverse of relation *R*.
PONG uses POS N-grams, EPP uses distributional similarity, and DEP uses dependency parses.

To score a potential relative r_0, EPP uses this formula:

$$Selpref_{R,t}(r_0) = \sum_{r \in Seen\arg s(R,t)} \frac{wt_{R,t}(r)}{Z_{R,t}} \cdot sim(r_0, r)$$

Here $sim(r_0, r)$ is the nGCM similarity defined below between vector space representations of r_0 and a relative r seen in the training data:

$$sim_{nGCM}(a, a') = \exp(-\sqrt{\sum_{i=1}^{n} (\frac{a_{b_i}}{\left\|\vec{a}\right\|} - \frac{a'_{b_i}}{\left\|\vec{a}'\right\|})^2})$$

$$where \left\|\vec{a}\right\| = \sqrt{\sum_{i=1}^{n} a_{b_i}^2}$$

The weight function $wt_{r,t}(a)$ is analogous to inverse document frequency in Information Retrieval.

DEP, our second baseline method, runs the Stanford dependency parser to label the training corpus with grammatical relations, and uses their frequencies to predict selectional preferences. To do the pseudo-disambiguation task, DEP compares the frequencies of (*R*, *t*, *r*) and (*R*, *t*, *r'*).

4.5 Relations tested

To test PONG, EPP, and DEP, we chose the most frequent eight relations between content words in the WSJ corpus, which occur over 10,000 times and are described in Table 4. We also tested their inverse relations. However, EPP does not compute selectional preferences for adjective and adverb as relatives. For this reason, we did not test EPP on *advmod* and *amod* relations with adverbs and adjectives as relatives.

4.6 Experimental results

Table 5 displays results for all 16 relations. To compute statistical significance conservatively in comparing methods, we used paired t-tests with N = 16 relations.

PONG's precision was significantly better than EPP ($p<0.001$) but worse than DEP ($p<0.0001$). Still, PONG's high precision validates its underlying assumption that POS N-grams strongly predict grammatical dependencies.

On coverage and recall, EPP beat PONG, which beat DEP ($p<0.0001$). PONG's F-score was higher, but not significantly, than EPP's ($p>0.5$) or DEP's ($p>0.02$).

4.7 Error analysis

In the pseudo-disambiguation task of choosing which of two words is related to a target, PONG makes errors of coverage (preferring neither word) and precision (preferring the wrong word).

Coverage errors, which occurred 17.4% of the time on average, arose only when PONG failed to estimate a probability for either word. PONG fails to score a potential relative *r* of a target *t* with a specified relation *R* if the labeled corpus has no POS N-grams that (a) map to *R*, (b) contain the POS of *t* and *r*, and (c) match Google word N-grams with *t* and *r* at those positions. Every relation has at least one POS N-gram that maps to it, so condition (a) never fails. PONG uses the most frequent POS of *t* and *r*, and we believe that condition (b) never fails. However, condition (c) can and does fail when *t* and *r* do not co-occur in any Google N-grams, at least that match a POS N-gram that can map to relation *R*. For example, *oversee* and *diet* do not co-occur in any Google N-grams, so PONG cannot score *diet* as a potential *dobj* of *oversee*.

Precision errors, which occur 17% of the time on average, arose when (a) PONG scored the distracter but failed to score the true relative, or (b) scored them both but preferred the distracter. Case (a) accounted for 44.62% of the errors on the covered test tuples.

One likely cause of errors in case (b) is over-generalization when PONG abstracts a word N-gram labeled with a relation by mapping its POS N-gram to that relation. In particular, the coarse POS tag set may discard too much information. Another likely cause of errors is probabilities estimated poorly due to sparse data. The probability of a relation for a POS N-gram rare in the training corpus is likely to be inaccurate. So is the probability of a POS N-gram for rare co-occurrences of a target and relative in Google word N-grams. Using a smaller tag set may reduce the sparse data problem but increase the risk of over-generalization.

5 Relation to Prior Work

In predicting selectional preferences, a key issue is generalization. Our DEP baseline simply counts co-occurrences of target and relative words in a corpus to predict selectional preferences, but only for words seen in the corpus. Prior work, summarized in
Table 6, has therefore tried to infer the similarity of unseen relatives to seen relatives. To illustrate, consider the problem of inducing that the direct objects of *celebrate* tend to be days or events.

Resnik (1996) combined WordNet with a labeled corpus to model the probability that relatives of a predicate belong to a particular conceptual class. This method could notice, for example, that the direct objects of *celebrate* tend to belong to the conceptual class *event*. Thus it could prefer *anniversary* or *occasion* as the object of *celebrate* even if unseen in its training corpus. However, this method depends strongly on the WordNet taxonomy.

Rather than use linguistic resources such as WordNet, Rooth et al. (1999) and Wald et al. (2008) induced semantically annotated subcategorization frames from unlabeled corpora. They modeled semantic classes as hidden variables, which they estimated using EM-based clustering. Ritter (2010) computed selectional preferences by using unsupervised topic models such as LinkLDA, which infers semantic classes of words automatically instead of requiring a pre-defined set of classes as input.

The contexts in which a linguistic unit occurs provide information about its meaning. Erk (2007) and Erk et al. (2010) modeled the contexts of a word as the distribution of words that co-occur with it. They calculated the semantic similarity of two words as the similarity of their context distributions according to various measures. Erk et al. (2010) reported the state-of-the-art method we used as our EPP baseline.

In contrast to prior work that explored various solutions to the generalization problem, we don't so much solve this problem as circumvent it. Instead of generalizing from a training corpus directly to unseen words, PONG abstracts a word N-gram to a POS N-gram and maps it to the relations that the word N-gram is labeled with.

Reference	Relation to target	Lexical resource	Primary corpus (labeled) & information used	Generalization corpus (unlabeled) & information used	Method
Resnik, 1996	Verb-object Verb-subject Adjective-noun Modifier-head Head-modifier	Senses in WordNet noun taxonomy	Target, relative, and relation in a parsed, partially sense-tagged corpus (Brown corpus)	none	Information theoretic model
Rooth et al., 1999	Verb-object Verb-subject	none	Target, relative, and relation in a parsed corpus (parsed BNC)	none	EM-based clustering
Ritter, 2010	Verb-subject Verb-object Subject-verb-object	none	Subject-verb-object tuples from 500 million web-pages	none	LDA model
Erk, 2007	Predicate and Semantic roles	none	Target, relative, and relation in a semantic role labeled corpus (FrameNet)	Words and their relations in a parsed corpus (BNC)	Similarity model based on word co-occurrence
Erk et al., 2010	SYN option: Verb-subject Verb-object, and their inverse relations SEM option: verb and semantic roles that have nouns as their headword in a primary corpus, and their inverse relations	none	Target, relative, and relation in SYN option: a parsed corpus (parsed BNC) SEM option: a semantic role labeled corpus (FrameNet)	Two options: WORDSPACE: an unlabeled corpus (BNC) DEPSPACE: Words and their subject and object relations in a parsed corpus (parsed BNC)	Similarity model using vector space representation of words
Zhou et al., 2011	Any (relations not distinguished)	none	Counts of words in Web or Google N-gram	none	PMI (Pointwise Mutual Information)
This paper	All grammatical dependencies in a parsed corpus, and their inverse relations	none	POS N-gram distribution for relations in parsed WSJ corpus	POS N-gram distribution for target and relative in Google N-gram	Combine both POS N-gram distributions

Table 6: Comparison with prior methods to compute selectional preferences

To compute selectional preferences, whether the words are in the training corpus or not, PONG applies these abstract mappings to word N-grams in the much larger Google N-grams corpus.

Some prior work on selectional preferences has used POS N-grams and a large unlabeled corpus. The most closely related work we found was by Gormley et al. (2011). They used patterns in POS N-grams to generate test data for their selectional preferences model, but not to infer preferences. Zhou et al. (2011) identified selectional preferences of one word for another

by using Pointwise Mutual Information (PMI) (Fano, 1961) to check whether they co-occur more frequently in a large corpus than predicted by their unigram frequencies. However, their method did not distinguish among different relations.

6 Conclusion

This paper describes, derives, and evaluates PONG, a novel probabilistic model of selectional preferences. PONG uses a labeled corpus to map POS N-grams to grammatical relations. It combines this mapping with probabilities estimated from a much larger POS-tagged but unlabeled Google N-grams corpus.

We tested PONG on the eight most common relations in the WSJ corpus, and their inverses – more relations than evaluated in prior work. Compared to the state-of-the-art EPP baseline (Erk et al., 2010), PONG averaged higher precision but lower coverage and recall. Compared to the DEP baseline, PONG averaged lower precision but higher coverage and recall. All these differences were substantial ($p < 0.001$). Compared to both baselines, PONG's average F-score was higher, though not significantly.

Some directions for future work include: First, improve PONG by incorporating models of lexical similarity explored in prior work. Second, use the universal tag set to extend PONG to other languages, or to perform better in English. Third, in place of grammatical relations, use rich, diverse semantic roles, while avoiding sparsity. Finally, use selectional preferences to teach word connotations by using various relations to generate example sentences or useful questions.

Acknowledgments

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080157. The opinions expressed are those of the authors and do not necessarily represent the views of the Institute or the U.S. Department of Education. We thank the helpful reviewers and Katrin Erk for her generous assistance.

References

de Marneffe, M.-C. and Manning, C.D. 2008. Stanford Typed Dependencies Manual. http://nlp.stanford.edu/software/dependencies_manual.pdf, Stanford University, Stanford, CA.

Erk, K. 2007. A Simple, Similarity-Based Model for Selectional Preferences. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, June, 2007, 216-223.

Erk, K., Padó, S. and Padó, U. 2010. A Flexible, Corpus-Driven Model of Regular and Inverse Selectional Preferences. Computational Linguistics 36(4), 723-763.

Fano, R. 1961. Transmission O F Information: A Statistical Theory of Communications. MIT Press, Cambridge, MA.

Franz, A. and Brants, T. 2006. All Our N-Gram Are Belong to You.

Gale, W.A., Church, K.W. and Yarowsky, D. 1992. Work on Statistical Methods for Word Sense Disambiguation. In Proceedings of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language, Cambridge, MA, October 23–25, 1992, 54-60.

Gildea, D. and Jurafsky, D. 2002. Automatic Labeling of Semantic Roles. Computational Linguistics 28(3), 245-288.

Gormley, M.R., Dredze, M., Durme, B.V. and Eisner, J. 2011. Shared Components Topic Models with Application to Selectional Preference, NIPS Workshop on Learning Semantics Sierra Nevada, Spain.

im Walde, S.S., Hying, C., Scheible, C. and Schmid, H. 2008. Combining Em Training and the Mdl Principle for an Automatic Verb Classification Incorporating Selectional Preferences. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, Columbus, OH, 2008, 496-504.

Klein, D. and Manning, C.D. 2003. Accurate Unlexicalized Parsing. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, Japan, July 7-12, 2003, E.W. HINRICHS and D. ROTH, Eds.

Petrov, S., Das, D. and McDonald, R.T. 2011. A Universal Part-of-Speech Tagset. ArXiv 1104.2086.

Resnik, P. 1996. Selectional Constraints: An Information-Theoretic Model and Its Computational Realization. Cognition 61, 127-159.

Resnik, P. 1997. Selectional Preference and Sense Disambiguation. In ACL SIGLEX Workshop on

Tagging Text with Lexical Semantics: Why, What, and How, Washington, DC, April 4-5, 1997, 52-57.

Ritter, A., Mausam and Etzioni, O. 2010. A Latent Dirichlet Allocation Method for Selectional Preferences. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 2010, 424-434.

Rooth, M., Riezler, S., Prescher, D., Carroll, G. and Beil, F. 1999. Inducing a Semantically Annotated Lexicon Via Em-Based Clustering. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, College Park, MD, 1999, Association for Computational Linguistics, 104-111.

Schutze, H. 1992. Context Space. In Proceedings of the AAAI Fall Symposium on Intelligent Probabilistic Approaches to Natural Language, Cambridge, MA, 1992, 113-120.

Toutanova, K., Klein, D., Manning, C. and Singer, Y. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of the Human Language Technology Conference and Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), Edmonton, Canada, 2003, 252–259.

Zhou, G., Zhao, J., Liu, K. and Cai, L. 2011. Exploiting Web-Derived Selectional Preference to Improve Statistical Dependency Parsing. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, OR, 2011, 1556–1565.