PIC a Different Word: A Simple Model for Lexical Substitution in Context

Stephen Roller
Department of Computer Science
The University of Texas at Austin
roller@cs.utexas.edu

Katrin Erk
Department of Linguistics
The University of Texas at Austin
katrin.erk@mail.utexas.edu

Abstract

The Lexical Substitution task involves selecting and ranking lexical paraphrases for a target word in a given sentential context. We present *PIC*, a simple measure for estimating the appropriateness of substitutes in a given context. PIC outperforms another simple, comparable model proposed in recent work, especially when selecting substitutes from the entire vocabulary. Analysis shows that PIC improves over baselines by incorporating frequency biases into predictions.

1 Introduction

Lexical substitution (McCarthy and Navigli, 2009) is a task in which word meaning in context is described not through dictionary senses but through substitutes (paraphrases) chosen by annotators. For example, consider the following usage of the adjective *bright*: “The *bright* girl was reading a book.” Valid lexical substitutions for *bright* include adjectives like *smart* and *intelligent*, but not words like *luminous* or *colorful*.

Originally introduced as a SemEval task in 2007, lexical substitution has often been used to evaluate the ability of distributional models to handle polysemy (Erk and Padó, 2008; Thater et al., 2010; Dinu and Lapata, 2010; Van de Cruys et al., 2011; Melamud et al., 2015b; Melamud et al., 2015a; Kawakami and Dyer, 2015). Recent models include a simple but high-performing method by Melamud et al. (2015b), which uses the Skip-gram model of Mikolov et al. (Mikolov et al., 2013) to compute the probability of a substitute given a sentence context, and integrates it with the probability of the substitute given the target. The current state of the art is held by another model of Melamud (Melamud et al., 2015a), which uses a more complex architecture.

In this paper we build on the simple model of Melamud et al. (2015b), as simpler methods are easier to recreate and integrate into larger pipelines.[1] We explore a weak form of supervision that recently has proved beneficial on many NLP tasks: using a language modeling task on unannotated data. We find a strong improvement over Melamud's simple measure, particularly on the all-words ranking task. Interestingly, analysis of *PIC* shows it improves over baselines by incorporating frequency biases into predictions.

2 Prior Work

In the lexical substitution task, an annotator is given a target word in context and generates one or more substitutes. As multiple annotators label a target, the result is a weighted list of substitutes, where weights indicate how many annotators chose a particular substitute (McCarthy and Navigli, 2009).

There have been numerous approaches on the lexical substitution task of varying complexity and using various lexical resources (McCarthy and Navigli, 2007). Some approaches focus on explicitly modeling an in-context vector (Erk and Padó, 2008; Dinu and Lapata, 2010; Thater et al., 2010; Van de Cruys et al., 2011; Kremer et al., 2014; Kawakami and Dyer, 2015), while others approach it using more sophisticated pipelines, in both super-

[1] Code and models available at https://github.com/stephenroller/naacl2016.

vised (Szarvas et al., 2013) and unsupervised (Melamud et al., 2015a) settings. The latter is the current state-of-art system, and is based around generating and pruning second-order word representations using language models.

In this work, we limit our comparisons to the model of Melamud et al. (2015b), a method which performs nearly state-of-art, is extremely easy to implement, and is a good testbed for focused hypotheses. They propose a novel measure which uses dependency-based word and context embeddings derived from Skip-gram Negative Sampling algorithm (SGNS) (Mikolov et al., 2013; Levy and Goldberg, 2014a). Their measure *addCos* for estimating the appropriateness of a substitute s as a substitute for t in the context $C = \{c_1, c_2, \ldots\}$ is defined as follows:[2]

$$\text{addCos}(s|t, C) = \cos(s, t) + \sum_{c \in C} \cos(s, c).$$

They also propose a similar measure *balAddCos*, which controls for the context size:

$$\text{balAddCos}(s|t, C) = |C|\cos(s, t) + \sum_{c \in C} \cos(s, c).$$

3 Proposed Measure

We propose a new measure, called Probability-in-Context (PIC), based on SGNS context vectors to estimate the appropriateness of a lexical substitute. Similar to *balAddCos*, the measure has two equally-weighted, independent components measuring the appropriateness of the substitute for both the target and the context, each taking the form of a softmax:[3]

$$\begin{aligned} \text{PIC}(s|t, C) &= P(s|t) \times P(s|C) \\ P(s|t) &= \frac{1}{Z_t} \exp\left\{s^\top t\right\} \\ P(s|C) &= \frac{1}{Z_C} \exp\left\{\sum_{c \in C} s^\top [Wc + b]\right\} \end{aligned}$$

The values Z_t and Z_C are normalizing constants to make sure each distribution sums to one. This measure has two free parameters, W and b, which act as a linear transformation over the context vectors. These parameters are estimated from the *original corpus*, and are trained to maximize the prediction of a *target* from only its syntactic contexts (c.f. Section 4.4). Given this formulation, a natural question is why not train the embeddings to optimize the softmax directly? We choose to parameterize the measure rather than the embeddings because (i) SGNS embeddings are already popular and readily available and (ii) it ensures the quality of embeddings remains constant across experimental settings.

To measure the importance of parameterization, we also compare to a non-parameterized PIC (*nPIC*), which only uses a softmax over the dot product:

$$\begin{aligned} \text{nPIC}(s|t, C) &= P(s|t) \times P_n(s|C) \\ P_n(s|C) &= \frac{1}{Z_n} \exp\left\{\sum_{c \in C} s^\top c\right\} \end{aligned}$$

4 Experimental Setup

We compare our proposed measures to three baselines: OOC, the Out-of-Context cosine similarity between the word and target ($\cos(s, t)$), and the *addCos* and *balAddCos* measures. It is important to note that existing papers on Lexical Substitution all contain subtle differences in experimental setup (vocabulary coverage, candidate pooling, etc.). We compare to our own re-implementation of the baselines, so our numbers differ slightly from those in the literature.

4.1 Data sets

We evaluate on three lexical substitution data sets.

SE07: The data set used in the original SemEval 2007 shared task (McCarthy and Navigli, 2007) consists of 201 words manually chosen to exhibit polysemy, with 10 sentences per target. For a given target in a particular context, five annotators were asked to propose up to 3 substitutes. As all our experiments are unsupervised, we always evaluate over the entire data set, rather than the original held-out test set.

Coinco: The Concepts-in-Context data set (Kremer et al., 2014) is a large lexical substitution corpus with proposed substitutes for nearly all content

[2] We abuse notation and allow s, t and c to refer to both the lexical items and their corresponding vectors.

[3] Note that $P(s|t)$ measures paradigmatic similarity of s and t, while $P(s|C)$ is syntagmatic fit to the context. For $P(s|t)$, Mikolov et al. (2013) show that cosine similarity of SGNS embeddings predicts paradigmatic similarity. $P(s|C)$ can be interpreted as the PMI of s and C (Levy and Goldberg, 2014b).

words in roughly 2,500 sentences from a mixture of genres (newswire, emails, and fiction). Crowdsourcing was used to obtain a minimum of 6 contextually-appropriate substitutes for over 15k tokens.

TSWI2: The Turk bootstrap Word Sense Inventory 2.0 (Biemann, 2012) is a crowdsourced lexical substitution corpus focused on about 1,000 common English nouns. The data set contains nearly 25,000 contextual uses of these nouns. Though the data set was originally constructed to induce a word-sense lexicon based on common substitution patterns, here we only use it as a lexical substitution data set.

4.2 Task Evaluation

We compare models on two variations of the lexical substitution task: candidate ranking and all-words ranking. In the *candidate ranking* task, the model is given a list of candidates and must select which are most appropriate for the given target. We follow prior work in pooling candidates from all substitutions for a given lemma and POS over all contexts, and measure performance using Generalized Average Precision (GAP). GAP is similar to Mean Average Precision, but weighted by the number of times a substitute was given by annotators. See Thater et al. (2010) for full details of the candidate ranking task.

The second task is the much more difficult task of *all-words ranking*. In this task, the model is not provided any gold list of candidates, but must select possible substitutes from the entire vocabulary.[4] We measure performance by (micro) mean Precision@1 and P@3: that is, of a system's top one/three guesses, the percentage also given by human annotators. These evaluation metrics are similar to the *best* and *oot* metrics reported in the literature, but we find P@1 and P@3 easier to interpret and analyze.

4.3 Word and Context Vectors

We use the word and context vectors released by Melamud et al. (2015b),[5] which were previously shown to perform strongly in lexical substitution tasks. These embeddings were computed from a corpus of (word, relation, context) tuples extracted from ukWaC and processed using the dependency-based word2vec model of Levy and Goldberg (2014a). These embeddings contain 600d vectors for 173k words and about 1M syntactic contexts.

4.4 Training Procedure

To train the W and b parameters, we extract tokens with syntactic contexts using the same corpus (ukWaC), parser (Chen and Manning, 2014), and extraction procedure used to generate the embeddings. See (Melamud et al., 2015b) for complete details. After extracting every token with its contexts, we randomly sample 10% of the data to reduce computation time, leaving us with 190M tokens for training W and b. We use sampled softmax to reduce training time (Jean et al., 2015), sampling 15 negative candidates uniformly from the vocabulary, optimizing cross-entropy over just these 16 words per sample. We optimize W and b in one epoch of stochastic gradient descent (SGD) with a learning rate of 0.01, momentum of 0.98, and a batch size of 2048. We found all of these hyperparameters worked well initially, and did not tune them.

5 Results

Table 1 contains results for all measures across all experimental settings.

The first observation we make is that the *PIC* measure performs best in all evaluations on all data sets by a significant margin.[6] In the GAP evaluation, all measures perform substantially better than the OOC baseline, and the *nPIC* measure performs comparably to *balAddCos*. We note that context-sensitive measures give the most improvement in SE07, reflecting its greater emphasis on polysemy.

As we turn to the all-words ranking evaluations, we observe that the absolute numbers are much lower, reflecting the increased difficulty of the task. We also see the that *nPIC* and *PIC* both improve greatly over all baselines: The *nPIC* measure is a relative 30% improvement over *balAddCos* in SE07 and Coinco, and the *PIC* measure is a relative 50% improvement over *balAddCos* in 5 evaluations.

Since both measures have a clear improvement over the baselines, especially in the more difficult

[4] All models are also hardcoded not to predict substitutes with the same stem as the target, e.g. for the *bright girl* example, models cannot predict *brighter* or *brightest*.

[5] `http://www.cs.biu.ac.il/nlp/resources/downloads/lexsub_embeddings`

[6] Wilcoxon signed-rank test, $p < 0.01$

Measure	SE07	Coinco	TWSI2
Candidate Ranking (GAP)			
OOC	44.2	44.5	57.9
addCos	51.2	46.3	62.2
balAddCos	49.6	46.5	61.3
nPIC	51.3	46.4	61.8
PIC	**52.4**	**48.3**	**62.8**
All-Words Ranking (Mean Precision@1)			
OOC	11.7	10.9	9.8
addCos	12.9	10.5	7.9
balAddCos	13.4	11.8	9.8
nPIC	17.3	16.3	11.1
PIC	**19.7**	**18.2**	**13.7**
All-Words Ranking (Mean Precision@3)			
OOC	9.7	8.6	7.0
addCos	9.0	7.9	6.1
balAddCos	9.8	9.1	7.4
nPIC	13.1	12.1	7.9
PIC	**14.8**	**13.8**	**10.1**

Table 1: Lexical Substitution results for candidate ranking (GAP) and all-words ranking tasks (P@1, P@3).

all-words task, we next strive to understand why.

5.1 Analysis

We first examine cherry and lemon-picked examples to give intuitions about why our model performs better. Table 2 contains the cherry example, where our model performs better than prior work. While OOC and *balAddCos* both suggest replacements with reasonable semantics, but are all misspelled. *nPIC* and *PIC* only pick words with the correct spellings, with the exception of "realy."

Table 3 shows the lemon example, where our model performs worse. We notice that the unusual "sea-change" item is prominent in the OOC and *balAddCos* models, but has dropped from the rankings in our models. From these and other examples, we hypothesize the model is simply guessing more frequent terms.

We consider a few experiments with this hypothesis that the measures do better because they capture better *unigram* statistics than the baselines. Recent literature found that the vector norm of SGNS embeddings correlates strongly with word frequency (Wilson and Schakel, 2015). We verified this for ourselves, computing the Spearman's rank correlation between the corpus unigram frequency and the vector length and found $rho = 0.90$, indicating the two correlate very strongly. Since the dot product is also the unnormalized cosine, it follows that *nPIC* and *PIC* should depend on unigram frequency.

To verify that the *nPIC* and *PIC* measures are indeed preferring more frequent substitutes, we compare the single best predictions (P@1) of the *balAddCos* and *nPIC* systems on all-words prediction on Coinco. Roughly 42% of the predictions made by the systems are identical, but of the remaining items, 74% of predictions made by *nPIC* have a higher corpus frequency than *balAddCos* (where chance is 50%). We find *balAddCos* and *PIC* make the same prediction 37% of the time, and *PIC* predicts a more frequent word in 83% of remaining items. The results for SE07 and TWSI2 are similar.

This indicates that the unigram bias is even higher for *PIC* than *nPIC*. To gain more insight, we manually inspect the learned parameters W and b. We find that the W matrix is nearly diagonal, with the values along the diagonal normally distributed around $\mu = 1.11$ ($\sigma = 0.02$) and the rest of the matrix normally distributed roughly around 0 (μ=2e-5, σ=0.02). This is to say, the *PIC* model is approximately learning to *exaggerate* the magnitude of the dot product, $s^\top c$. This suggests one could even replace our parameter W with a single scaling parameter, though we leave this for future work.

To inspect the bias b, we compute the inner product of the b vector with the word embedding matrix, to find each word's a priori bias, and correlate it with word frequencies. We find $rho = 0.25$, indicating that b is also capturing unigram statistics.

Is it helpful in lexical substitution to prefer more frequent substitutes? To test this, we pool all annotator responses for all contexts in Coinco, and find the number of times a substitute is given correlates strongly with frequency ($rho = 0.54$).

These results emphasize the importance of incorporating unigram frequencies when attempting the lexical substitution task (as with many other tasks in NLP). Compared to cosine, the dot product in *nPIC* stresses unigram frequency, and the parameters W and b strengthen this tendency.

OOC	*balAddCos*	*nPIC*	*PIC*
You can sort of challenge them well, did you **really** know the time when you said yes?			
trully	proably	realy	**actually**
actually	trully	**truly**	**truly**
actaully	acutally	**actually**	already
acutally	actaully	hardly	barely
proably	probaly	**definitely**	just

Table 2: Example where the *PIC* performs better in the All-Words Ranking task. The target word and correct answers are bolded.

OOC	*balAddCos*	*nPIC*	*PIC*
As a general rule, point of view should not **change** during a scene.			
sea-change	**alter**	reoccur	re-occur
alter	sea-change	re-occur	appear
shift	**shift**	prevail	overstate
downshift	downshift	deviate	differ
re-configure	increase/decrease	divulged	disappear

Table 3: Example where the *PIC* performs worse the All-Words Ranking task. The target word and correct answers are bolded.

6 Conclusion

We have presented *PIC*, a simple new measure for assessing the appropriateness of a substitute in a particular context for the Lexical Substitution task. The measure assesses the fit of the substitute both to the target word and the sentence context. It significantly outperforms comparable baselines from prior work, and does not require any additional lexical resources. An analysis indicates its performance improvements derive primarily from a tendency to lean more strongly on unigram statistics than baselines. In future work, our measure could be simplified by implementing the bias as a single scaling parameter.

Acknowledgments

We would like to thank Karl Pichotta, Martin Reidl, and the anonymous reviewers for their helpful comments and suggestions. This research was supported by the NSF grant IIS 1523637. We acknowledge the Texas Advanced Computing Center for providing grid resources that contributed to these results.

References

Chris Biemann. 2012. Turk bootstrap word sense inventory 2.0: A large-scale resource for lexical substitution. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 4038–4042, Istanbul, Turkey, May. European Language Resources Association.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 740–750, Doha, Qatar, October. Association for Computational Linguistics.

Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1162–1172, Cambridge, MA, October. Association for Computational Linguistics.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 897–906, Honolulu, Hawaii, October. Association for Computational Linguistics.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1–10, Beijing, China, July. Association for Computational Linguistics.

Kazuya Kawakami and Chris Dyer. 2015. Learning to

Represent Words in Context with Multilingual Supervision. *ArXiv e-prints*, abs/1511.04623, November.

Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. 2014. What substitutes tell us - analysis of an "all-words" lexical substitution corpus. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 540–549, Gothenburg, Sweden, April. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 302–308, Baltimore, Maryland, June. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pages 48–53, Prague, Czech Republic, June. Association for Computational Linguistics.

Diana McCarthy and Robert Navigli. 2009. The English lexical substitution task. *Language Resources and Evaluation*, 43(2):139–159. Special Issue on Computational Semantic Analysis of Language: SemEval-2007 and Beyond.

Oren Melamud, Ido Dagan, and Jacob Goldberger. 2015a. Modeling word meaning in context with substitute vectors. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 472–482, Denver, Colorado, May–June. Association for Computational Linguistics.

Oren Melamud, Omer Levy, and Ido Dagan. 2015b. A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, Denver, Colorado, June. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of International Conference on Learning Representations*.

György Szarvas, Chris Biemann, and Iryna Gurevych. 2013. Supervised all-words lexical substitution using delexicalized features. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1131–1141.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 948–957, Uppsala, Sweden, July. Association for Computational Linguistics.

Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. 2011. Latent vector weighting for word meaning in context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1012–1022, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Benjamin J. Wilson and Adriaan M. J. Schakel. 2015. Controlled experiments for word embeddings. *ArXiv e-prints*, abs/1510.02675, October.