Dissecting the Practical Lexical Function Model for Compositional Distributional Semantics

Abhijeet Gupta, Jason Utt and **Sebastian Padó**
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
`[guptaat|uttjn|pado]@ims.uni-stuttgart.de`

Abstract

The Practical Lexical Function model (PLF) is a recently proposed compositional distributional semantic model which provides an elegant account of composition, striking a balance between expressiveness and robustness and performing at the state-of-the-art. In this paper, we identify an inconsistency in PLF between the objective function at training and the prediction at testing which leads to an over-counting of the predicate's contribution to the meaning of the phrase. We investigate two possible solutions of which one (the exclusion of simple lexical vector at test time) improves performance significantly on two out of the three composition datasets.

1 Introduction

Compositional distributional semantic models (CDSMs) make an important theoretical contribution, explaining the meaning of a phrase by the meanings of its parts. They have also found application in psycholinguistics (Lenci, 2011), in sentiment analysis (Socher et al., 2012), and in machine translation (Kalchbrenner and Blunsom, 2013).

A first generation of CDSMs represented all words as vectors and combined them by component-wise operations (Mitchell and Lapata, 2010). Given the conceptual limitations of this simple approach, numerous models were subsequently proposed which represent the meaning of predicates as higher-order algebraic objects such as matrices and tensors (Baroni and Zamparelli, 2010; Guevara, 2010; Coecke et al., 2010). For example, one-place predicates such as adjectives or intransitive verbs can be modeled as matrices (order-2 tensors), and two-place predicates, e.g., transitive verbs, as order-3 tensors, and so forth. While such tensors enable mathematically elegant accounts of composition, their large degrees of freedom lead to severe sparsity issues when they are learned from corpora.

The recently proposed Practical Lexical Function model (PLF; Paperno et al., 2014) represents a compromise between these two extremes by restricting itself to vectors and matrices, effectively reducing sparsity while retaining state-of-the-art performance across multiple datasets. It does away with tensors by ignoring interactions among the arguments of predicates p. Instead, each argument position arg is modeled as a matrix $\overset{\square_{arg}}{p}$ that is applied to a vector for the argument's meaning, $\overrightarrow{a}$. The meaning of the phrase is then defined as the sum of the lexical meaning of the predicate, $\overrightarrow{p}$, and the contributions of each argument (see Fig. 1). The matrices can be learned in a supervised manner with regression from pairs of corpus-extracted vectors for arguments and phrases.

In this paper, we identify an inconsistency between the training and testing phases of the PLF. More specifically, we show that its composition procedure leads to over-counting of the contribution of the predicate. We propose two remedies to harmonize the training and prediction phases – by excluding the predicate meaning from either training or testing. In an evaluation of the standard PLF and our variants on three datasets, we find that modifying the training phase fails, but that modifying testing phase improves performance on two out of three datasets. We analyze this effect in terms of a bias-variance tradeoff.

*Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (*SEM 2015)*, pages 153–158,
Denver, Colorado, June 4–5, 2015.

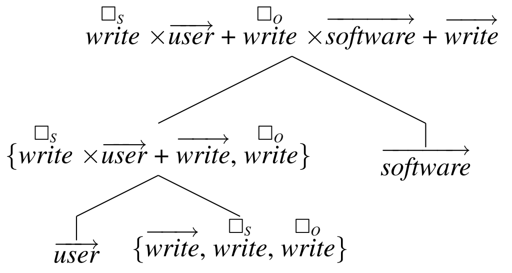
$\overset{\square_S}{write} \times \overrightarrow{user} + \overset{\square_O}{write} \times \overrightarrow{software} + \overrightarrow{write}$
$\{\overset{\square_S}{write} \times \overrightarrow{user} + \overrightarrow{write}, \overset{\square_O}{write}\}$
$\overrightarrow{software}$
$\overrightarrow{user}$
$\{\overrightarrow{write}, \overset{\square_S}{write}, \overset{\square_O}{write}\}$

Figure 1: Practical Lexical Function model derivation for the noun-verb-noun phrase *"user writes software"*.

2 Debugging the PLF model

2.1 An Inconsistency

We have identified an inconsistency in the PLF model as a result of which the predicted vector for a phrase systematically differs from the corpus-observed vector of the phrase. We will illustrate it on a minimal example, the phrase "*dogs sleep*".

Training Phase. The training of PLF creates three representations: (1), a lexical vector for the noun ($\vec{n}$); (2), the lexical vector for the verb ($\vec{v}$); and (3), a matrix for the subject argument position of the verb ($\overset{\square_S}{v}$). While (1) and (2) can be acquired directly from the corpus, (3) involves optimization, since the matrix (3) is supposed to account for the verb's disambiguating effect on all its subjects. PLF proposes to learn matrices via regression problems such as the following (Guevara, 2010), where $subj(v)$ comprises the subjects seen with the verb *v*:[1]

$$\overset{\square_S}{v} := \operatorname*{argmin}_{M} \sum_{n \in subj(v)} \|M \times \vec{n} - \overrightarrow{n\,v}\|^2 \qquad (1)$$

That is, the verb's subject matrix is learned as the matrix which, multiplied with a subject noun vector, best predicts the noun-verb phrase vector. If we assume that the verb of our example (*sleep*) is only seen with a single noun in the corpus, namely its subject *dog*, Eq. (1) has a particularly simple solution where the matrix can perfectly predict the phrase vector:

$$\overset{\square_S}{sleep} \times \overrightarrow{dog} = \overrightarrow{dog\ sleep} \qquad (2)$$

Testing Phase. PLF predicts the phrase meaning $\mathcal{P}$ for our example as predicate plus argument meaning:

$$\mathcal{P}(dog\ sleeps) = \overrightarrow{sleep} + \overset{\square_S}{sleep} \times \overrightarrow{dog} \qquad (3)$$

Intuitively, what we would expect as the result of this computation to be $\overrightarrow{dog\ sleeps}$ — the empirically observed vector for the noun-verb phrase. However, substituting Eq. (2) into Eq. (3), we instead obtain:

$$\mathcal{P}(dog\ sleeps) = \overrightarrow{sleep} + \overrightarrow{dog\ sleeps} \qquad (4)$$

The predicted phrase meaning does not correspond to the empirical phrase vector because in PLF, the verb contributes twice to the phrase meaning.

Discussion. This issue remains pertinent beyond the minimal example presented above. The reason is a discrepancy between the training and test setups: The argument matrices in PLF are learned so as to predict the *complete* phrase vector when multiplied with an argument (compare Eq. (1)).[2] This objective is inconsistent with the way phrase vectors are predicted at test time. The addition of the predicate's lexical vector thus amounts to a **systematic over-counting** of the predicate's lexical contribution.

2.2 Two Ways to Remedy the Inconsistency

The above description gives direct rise to two simple strategies to harmonize training and test procedures.

Adapting the Training Phase. One strategy is to adapt the training objective from Eq. (1). Recognizing that the predicate vector is added in by Eq. (3) at test time, we can attempt to learn a matrix that predicts not the phrase vector, but the *difference* between the phrase vector and the predicate vector. That means, the matrices capture only the disambiguating contribution of argument positions such as subject:

$$\overset{\square_S}{v} = \operatorname*{argmin}_{M} \sum_{n \in subj(v)} \|M \times \vec{n} - (\overrightarrow{n\,v} - \vec{v})\|^2 \qquad (5)$$

Adapting the Testing Phase. Another strategy is to adapt the phrase meaning prediction at test time by simply leaving out the predicate vector. For subject-verb combinations, we predict $\mathcal{P}(n\ v) = \overset{\square_S}{v} \times \vec{n}$.

[1] All matrices are learned using least-squares regression and, for the sake of simplicity, we ignore regularization. Adjective matrices are obtained in the same fashion.

[2] A formal, more general argument can be made based on the error term $\vec{\epsilon} = \overset{\square_{arg}}{v} \times \vec{n} - \overrightarrow{n\,v}$ which is minimized in training.

verb in context	landmark in context	similarity
private landlord ***charge*** *annual rent*	*private landlord* ***accuse*** *annual rent*	low
private landlord ***charge*** *annual rent*	*private landlord* ***bill*** *annual rent*	high
armed police ***charge*** *unemployed person*	*armed police* ***accuse*** *unemployed person*	high
armed police ***charge*** *unemployed person*	*armed police* ***bill*** *unemployed person*	low

Table 1: Example of experimental items in the ANVAN data sets (target verb: *charge*).

For transitive sentences (cf. Figure 1), we predict $\mathcal{P}(n\ v\ n) = \overset{\square_S}{\vec{v}} \times \vec{n} + \overset{\square_O}{\vec{v}} \times \vec{n}$ (the sum of the subject and the object contributions), and analogously for other constructions.

3 Experimental Setup

Evaluation Datasets. We evaluate the modifications from the last section on three standard benchmarks for CDSMs: ANVAN-1 (Kartsaklis et al., 2013), ANVAN-2 (Grefenstette, 2013) (Paperno et al.'s term) and NVN (Grefenstette and Sadrzadeh, 2011) (our term).

As the abbreviations indicate, the two ANVAN datasets contain transitive verbs whose NP arguments are modified by arguments; the NVN dataset contains only bare noun arguments. All three datasets are built around ambiguous target verbs that are combined with two disambiguating contexts (subjects plus objects) and two landmark verbs in a balanced design (cf. Table 1). Each context matches one of the landmark verbs, but not the other. Annotators were asked to rate the similarity between the target verb in context and the landmark on a Likert scale.

Corpus and Co-Occurrences. We followed the specifications by Paperno et al. (2014) as closely as possible to replicate the original PLF results. As corpora, we used ukWAC, English Wikipedia, and the BNC. We extracted a square co-occurrence matrix for the 30K most frequent content words using a 3-word window and applied the PPMI transformation. Subsequently, the matrix was reduced to 300 dimensions with SVD. In the same manner, we built a co-occurrence matrix for all corpus bigrams for relevant adjectives and verbs from the experimental materials, applying a frequency threshold of 5.

Composition Models and Evaluation. We build matrix representations for adjectives and subject and object positions of verbs using the DISSECT toolkit (Dinu et al., 2013). In addition to the standard PLF model, which we see as a baseline, we implement both proposals from Section 2.2. On the NVN dataset, both training and test modification can apply only to the verb (cf. Figure 1), which gives us two conditions. On the ANVAN datasets (cf. Figure 2), the changes can be applied to the verb, to the adjectives, or to both, for a total of six conditions.

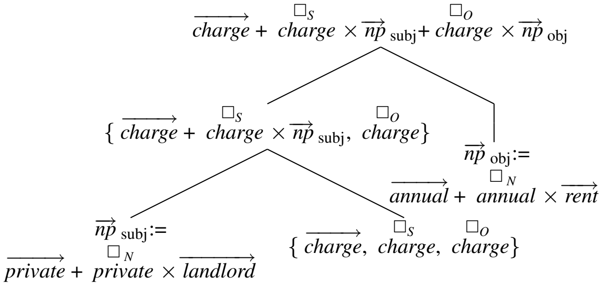
$\overrightarrow{charge} + \overset{\square_S}{charge} \times \overrightarrow{np}_{\text{subj}} + \overset{\square_O}{charge} \times \overrightarrow{np}_{\text{obj}}$
$\{\overrightarrow{charge} + \overset{\square_S}{charge} \times \overrightarrow{np}_{\text{subj}}, \overset{\square_O}{charge}\}$
$\overrightarrow{np}_{\text{obj}} :=$
$\overrightarrow{annual} + \overset{\square_N}{annual} \times \overrightarrow{rent}$
$\overrightarrow{np}_{\text{subj}} :=$
$\overrightarrow{private} + \overset{\square_N}{private} \times \overrightarrow{landlord}$
$\{\overrightarrow{charge}, \overset{\square_S}{charge}, \overset{\square_O}{charge}\}$

Figure 2: PLF Derivation for ANVAN phrase *"private landlord charge yearly rent"*.

Our evaluation measure is the nonparametric Spearman correlations between each annotator's similarity rating and the cosine between the predicted sentence vectors containing the ambiguous and landmark verb, respectively.

4 Evaluation

Main Results. The main results are shown in Table 2. Our PLF re-implementation in the first column almost replicates the results reported by Paperno et al. (2014) for ANVAN1 and ANVAN2 (20 and 36, respectively). On NVN, no results for the PLF were previously reported. Our result (35.4) is substantially above the result of 21.0 reported by Greffenstette and Sadrzadeh (2011) for their categorial model. This supports our general focus on the PLF as an interesting target for analysis.

		Training phase modifications			Test phase modifications		
Dataset	PLF	Sub Adj	Sub Verb	Sub Both	No Adj	No Verb	No Both
ANVAN1	20.6	18.7	-0.3	3.8	19.2	20.7	**22.1***
ANVAN2	35.2	32.8	13.8	17.0	33.8	**35.7**	35.4
NVN	35.4	–	25.5	–	–	**40.6****	–

Table 2: Experimental results (Spearman's ρ) on three dataset. Significant improvements over the PLF results are indicated with stars (*: p<0.05, **: p<0.01), – denotes non-applicability of parameter.

The results for the training phase modification are overwhelmingly negative. There is a minor degradation when the adjective is subtracted at training time, and major degradation when the verb is subtracted. We will come back to this result below.

In contrast, we obtain improvements when we modify the test phase, when we either leave out the verb or both the verb and the adjective in the composition. For two out of the three datasets, the respective best models perform statistically significantly better than the PLF as determined by a bootstrap resampling test (Efron and Tibshirani, 1993): ANVAN1 (+1.5%, p<0.05) and NVN (+5.2%, p<0.01). The improvement for ANVAN2 (+0.5%) is not large enough to reach significance.

	ANVAN1	ANVAN2	NVN
logf	-359***	-182 n.s.	-96***
ambig	118***	8 n.s.	6***
ModTest	438***	-2606***	-1413***
ModTest:logf	-53**	165***	94***
ModTest:ambig	20*	32***	8***

Table 3: Coefficients of Linear Mixed Effects Model.
*: p<0.05; **: p<0.01; ***: p<0.001. See text for details.

Discussion. These results leave us with two main questions: (a), why does the modification at training time fail so completely; and (b), can we develop a better understanding of the kind of improvement that the modification at test time introduces?

Regarding question (a), we believe that the difference between the phrase vector and the predicate vector that we are training the matrix to predict in Eq. (5) is, in practice, a very brittle representation. The reason is that typically the phrase nv is much less frequent than v, and therefore $\overrightarrow{n\,v} - \overrightarrow{v} \approx -\overrightarrow{v}$ (cf. Figure 3). Consequently, the matrix attempts to predict the verb vector from the noun – not only a very hard problem, but one that does not help solve the task at hand.

To answer question (b), we perform a mixed effects linear regression analysis (Hedeker, 2005) on the three datasets, concentrating on a comparison of the standard PLF and the best respective test phase modification. We follow the intuition that the frequency and ambiguity of the target verbs should influence the quality of the prediction both in the PLF and in the modified model, and that it might be informative to look at differences in these effects. To this effect, we construct a mixed-effects model which predicts, for each experimental item (cf. Table 1), the *absolute rank difference* between the item's rank in the gold standard ratings and the item's rank in the model prediction. Thus, high values of the output variable denote items which are difficult to predict, while low values of the output variable denote items which are easy to predict. As fixed effects, we include the target verbs' logarithmized corpus frequencies (*logf*), their ambiguities, measured as the number of WordNet top nodes subsuming their synsets (*ambig*), the presence of the test phase modification (NoVerb for ANVAN2 and NVN, NoBoth for ANVAN1; *ModTest*) as well as interaction terms between ModTest and the two other predictors. We also include the identity of the target verb as random effect.

The results are shown in Table 3. There are considerable differences between the datasets, but the overall patterns are nevertheless comparable. Notably, frequency has a negative effect on rank difference. In other words, more frequent verbs are easier to predict. Conversely, the ambiguity of the target verb has a positive effect on rank difference, that is, higher ambiguity makes predictions more difficult. Both of these effects are very strong on ANVAN1 and NVN

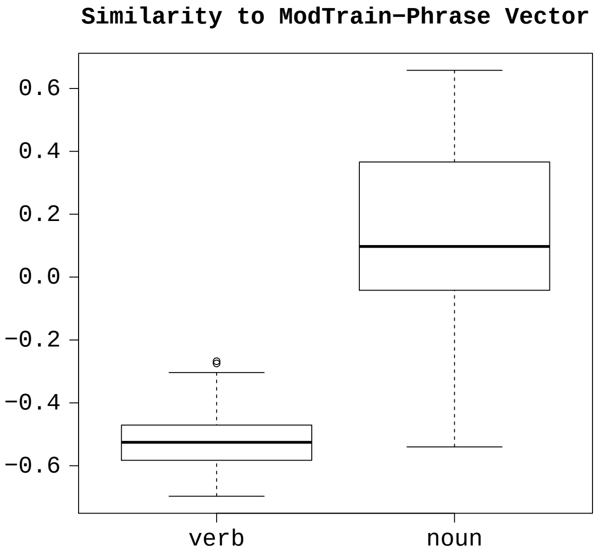

Figure 3: Similarities between the training-time modified phrase vector (subject-verb & verb-object) and the respective word vectors in the NVN dataset. The low values and smaller variance in verb similarities shows the information encoded by the modified phrase vector aligns better with the verb's (or predicate's) information than that of the noun (argument).

and not significant on ANVAN2, which appears to be a more controlled dataset. Taken together, the models still seem to struggle with ambiguous and infrequent target verbs.

The coefficients that we obtain for ModTest look puzzling at first glance: we obtain a negative coefficient (i.e., an overall improvement) only for ANVAN2 and NVN while the coefficient is positive for ANVAN1. For ANVAN1, the improvement is brought about by the interaction with the frequency variable: when the test phase is modified, the (beneficial) effect of frequency becomes much stronger, that is, the predictions for high-frequency verbs improve. In contrast, the effect of frequency becomes weaker for the test phase modification on ANVAN2 and NVN. What is true for all three datasets is that the effect of ambiguity gets stronger when the test phase is modified: ambiguous verbs become significantly more difficult to model.

On the basis of this analysis, we believe that this difference between the standard PLF and our test phase modification can be understood as a classical bias-variance tradeoff: the addition of the predicate meaning in the standard PLF reduces variance, ensuring that the phrase meaning stays close to the predicate meaning prior even for matrices that are difficult to learn, e.g., due to sparse data or high ambiguity. At the same time, this dilutes the disambiguating effect of composition. In our modified scheme, the situation is reversed: the composed representations vary more freely, which benefits well-learned matrices but leads to worse predictions for poorly learned ones.

5 Conclusion

In this paper, we have presented an analysis of the recent Practical Lexical Function (PLF) model in compositional distributional semantics. We have shown that the PLF contains an inconsistency between the objective function at training time and the definition of compositional phase construction at testing time. We have argued that either training or testing needs to be modified to harmonize the two. Our empirical evaluation found that testing phase modification is indeed effective (by reducing bias in the predictions), while the training phase modification is not (by relying on brittle representations). In the spirit of the bias-variance analysis, future work is to experiment with weighting schemes to optimize the relative contributions of predicate and arguments.

Acknowledgments

We gratefully acknowledge funding of our research by the DFG (SFB 732, Project D10).

References

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36:345–386.

Georgiana Dinu, Nghia The Pham, and Marco Baroni. 2013. DISSECT - DIStributional SEmantics Composition Toolkit. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 31–36, Sofia, Bulgaria.

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall, New York.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404, Edinburgh, Scotland, UK.

Edward Grefenstette. 2013. *Category-Theoretic Quantitative Compositional Distributional Models of Natural Language Semantics*. Ph.D. thesis.

Emiliano Guevara. 2010. A regression model of adjective-noun compositionality in distributional semantics. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, pages 33–37, Uppsala, Sweden, July. Association for Computational Linguistics.

Donald Hedeker. 2005. Generalized linear mixed models. In *Encyclopedia of Statistics in Behavioral Science*. Wiley, New York.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Melbourne, Australia.

Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen Pulman. 2013. Separating disambiguation from composition in distributional semantics. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 114–123, Sofia, Bulgaria.

Alessandro Lenci. 2011. Composing and updating verb argument expectations: A distributional semantic model. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 58–66, Portland, OR.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Denis Paperno, Nghia The Pham, and Marco Baroni. 2014. A practical and linguistically-motivated approach to compositional distributional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 90–99, Baltimore, Maryland.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1201–1211, Stroudsburg, PA, USA. Association for Computational Linguistics.