Three-order normalized PMI and other lessons in tensor analysis of verbal selectional preferences

Márton Makrai[1,2]

[1] Institute of Cognitive Neuroscience and Psychology
Research Centre for Natural Sciences, Budapest
[2] Research Group for Language Technology
Hungarian Research Centre for Linguistics
makrai.marton@ttk.hu

Abstract. We investigate several questions in transitive verb structure representation by decomposing tensors populated with different subject-verb-object association measures, including a novel generalization of normalized pointwise mutual information to the higher-order (>2) case. Which association measure works the best in modeling verb structures? Should we include occurrences with unfilled arguments in our statistics? We also investigate qualitatively the latent dimensions, and the difference between each noun as a subject versus an object.

Keywords: selectional preferences, tensor decomposition, association measures

1 Introduction

Verbs have been characterized on the basis of how frequently various syntactic constituents occur in various grammatical relations to them, which is, not surprisingly, related to the meaning of the verb (Levin, 1993). These selectional preferences have been analyzed with machine learning tools (Van de Cruys, 2009). Verb structures include collocations, whose syntactic modifiability or semantic compositionality is reduced: their linguistic distribution may be idiosyncratic or the sense of the combination may be habitual or even fixed (Bouma, 2009).

Tensors (>2-dimensional arrays) generalize matrices; while matrices contain numbers aligned in two dimensions, rows and columns, tensors have more of these dimensions, also called *axes* or *modes*[1] Singular value decomposition (SVD) of a co-occurrence matrix is a natural tool to compute generalizations about the interactions between two modes, like words and documents (LSA, Landauer and Dumais (1997)), target and context words (words embeddings, Mikolov et al. (2013b); Levy and Goldberg (2014b); Pennington et al. (2014)), or words and dependency contexts (Levy and Goldberg, 2014a). Four ways of looking at SVD (in LSA) can be distinguished (Turney and Pantel, 2010): the goal can be the modeling of some latent meaning, noise reduction, indirect aka. high-order co-occurrences (when two words appear in similar contexts), or sparsity reduction. Intuitively, language features multi-mode interactions: *the turntable playing the piano* can be strange (Van de Cruys, 2009), while the two-mode relations

[1] The term *mode* is preferred when data from different modalities are fused.

⟨play, SUBJ, turntable⟩ and ⟨play, OBJ, piano⟩ are perfect. Tensor generalizations of matrix decomposition (Kolda and Bader, 2009), especially *low-rank factorizations*, open the way for the analysis of such interactions.

It seems that, after intensive early research (Van de Cruys, 2009; Van de Cruys et al., 2013; Polajnar et al., 2014; Fried et al., 2015; Hashimoto and Tsuruoka, 2015), results obtained with skip-gram and related word embedding methods outshone tensor methods for verb argument structure. Yet tensor decomposition has developed remarkably, and NLP test-beds in the domain of verb argument structure have been involved in cutting-edge scalable, noise-robust tensor works (Sharan and Valiant, 2017; Bailey et al., 2018; Frandsen and Ge, 2019). The data-driven linguistic understanding of word ambiguity and especially that of verb selection is still immature. Here we try to make progress in the linguistic direction by further research on tensorial analysis of verb argument structure.

Tensor decomposition provides embedding vectors for each mode (in our case, nouns as subjects, verb, and nouns as objects) analogous to word embeddings in (shallow or deep) neural networks. In this paper, we compute different association measures between subjects, verbs, and objects, populate tensors with these measures, decompose the tensors with different algorithms, and investigate the resulting word embeddings quantitatively and qualitatively to answer the following questions.

Our first four questions will be answered quantitatively in the modeling of English subject-verb-object triple similarity, while the last two questions are qualitative.

- Which *association measure* yields the best representations? We experiment with several measures, including our novel generalization of normalized pointwise mutual information to the higher-order (>2) case.
- Should we include *empty argument fillers* (subjects or objects) in our co-occurrence statistics? Ideally, including them may help generalization over the transitive and the intransitive uses of the same verb, while discarding them may help focusing on transitive structures cleanly as a separate phenomenon.
- The two tensor decomposition algorithms, CPD and Tucker, which we will introduce in Section 3, have very different time-complexity: Tucker is much faster. Tensor decomposition has hyper-parameters like the decomposition rank and the frequency cutoff. Both are related to memory limitation, especially the latter. It would be beneficial, *if the two algorithms reached the best results with the same hyper-parameters,* because then a fast parameter tuning with Tucker would also benefit CPD. Is this the case?
- How does the trade-off between the three hyper-parameters related to the *size of the decomposition* (i.e. the decomposition rank, the inclusion of empty fillers, and the frequency cutoff) look like?
- Do latent dimension of our word embeddings reflect lexical knowledge?
- Can the difference between each noun as a subject versus an object correspond to some intuitive difference between subjecthood and objecthood?

Section 2 describes the linguistically motivated association measures between subjects, verbs, and objects we apply. These measures include ones that are novel to the best of our knowledge. Section 3 offers an introduction to tensor decomposition. Section 4 describes our experiments. Our code is available online.[2]

2 Counts, weighting, and associations

Word co-occurrences form *sparse* arrays, as most words do not occur empirically with most words, and frequencies span many orders of magnitude (*Zipf* or power law distribution, Manin (2008); Gittens et al. (2017)). In order to scale to large data, linguistic tensor decomposition methods have to be based on sparse tensors populated with more sophisticated scores than frequency. Now we turn to these weighting functions and especially to linguistically motivated association scores.

The simplest choice is the logarithm of the co-occurrence frequency (Pennington et al., 2014; Sharan and Valiant, 2017). Jenatton et al. (2012) places the modeling of the $\langle\text{subject}, \text{verb}, \text{object}\rangle$ triples in the context of multi-relational learning, and apply a weighting function related to the log-bilinear model (Mnih and Hinton, 2007; Mikolov et al., 2013a).

Van de Cruys (2009, 2011); Van de Cruys et al. (2013), and Bailey et al. (2018) use three-mode generalizations of the information-theoretic association measure *(Positive) Pointwise Mutual Information* ((P)PMI). Positivity is related to sparse inputs: in order to attribute higher scores to actual co-occurrences than unattested ones, PMI and the lexicographic association scores introduced in the following paragraph, *positive* variants of the association measures have to be used, e.g. PPMI, which replaces negative PMI entries with zero. We discuss the two types of three-variable generalization of PPMI in Section 2.2: the more standard total correlation (that we still call PMI) and interaction information.

We also experiment with generalizing Log Dice (Rychlý, 2008) to three axes

$$\log \frac{3f(x,y,z)}{f(x)+f(y)+f(z)} + c,$$

where c is chosen so that the Log-Dice values are non-negative. (While 3 in the nominator is redundant, because it is subsumed under c, we keep it in the formula to make it more reminiscent of the established 2-variable case.) The use of Log Dice as well as salience introduced in the next paragraph has, to the best of our knowledge, mainly been limited so far to lexicography.

2.1 Salience and normalized PPMI

PPMI, despite of its nice information-theoretic interpretability, is biased towards rare events (Turney and Pantel, 2010; Levy et al., 2015; Zhuang et al., 2018). This motivates the Sketch Engine lexicographic software (Kilgarriff et al., 2004) to multiply vanilla PPMI by $\log f$ (in our case, by $\log(f(x,y,z))$, to get the measure of *salience*. We apply similar modifications to every score introduced in Section 2 so far. Denoting vanilla PPMI, interaction information and Log Dice by `pmi-vanl`, `iact-vanl`, and

[2] `https://github.com/anonymous`

`Dice-vanl`, respectively, we get `pmi-sali`, `iact-sali`, and `Dice-sali` by multiplying the vanilla score by $\log f(x, y, z)$.

There is a theoretically better motivated way of transforming PMI to some measure which is less biased towards rare combinations. In Bouma (2009)'s approach, *normalization* is related to boundedness. He looks for measures whose absolute value is pointwise larger than that of PMI. Entropy and negative log probability are two of those measures, and we follow the literature in choosing the latter. In our experiments, we apply this normalization to the two multi-mode generalizations of PMI which will be introduced in Section 2.2, interaction information and the one which we will still call PMI. While normalized interaction information does not excel in our experiments, tree-variable normalized PMI, which is to the best of our knowledge the novelty of the present paper, proves the best among the alternatives considered. Empirically, when divided by $-\log p(x, y, z)$, positive interaction information and the more standard 3-mode PPMI is upper-bounded by 1 and 2, respectively.

2.2 Higher-order PMI

One would think that it's obvious that the 3-variable generalization of Pointwise Mutual Information (PMI) is

$$\log \frac{p(x,y,z)}{p(x)p(y)p(z)}, \tag{1}$$

but it turns out that this is only one of the possible generalizations. Van de Cruys (2011) introduces two pointwise association measures, whose expected values are two different multivariate generalizations of mutual information (Shannon and Weaver, 1949): interaction information (McGill, 1954) and total correlation (Watanabe, 1960).

Pointwise *interaction information* is based on the notion of conditional mutual information.[3]

$$\log \frac{p(x,y)p(x,z)p(y,z)}{p(x,y,z)p(x)p(y)p(z)}$$

Total correlation on the other hand quantifies the amount of information that is shared among the variables, with a pointwise variant defined by the formula in Equation 1. Following the literature (Villada Moirón, 2005; Van de Cruys, 2009; Van de Cruys et al., 2013; Bailey et al., 2018), when we speak about *(multivariate Positive) Pointwise Mutual Information* in this paper, we will mean (pointwise) total correlation.

Van de Cruys (2011) reports that, in their Dutch experiments, both methods are able to extract salient subject verb object triples (prototypical SVO combinations like *poll represents opinion* and fixed expressions). Narrowing the scope to the word *play*, they find that interaction information picks up on prototypical SVO combos e.g. *orchestra plays symphony*, while the more established one (which he calls specific correlation) picks up on *play a role* and salient subjects that go with the expression.

[3] Mnemonically, the formula of the pointwise variant generalizes the 2-mode case along the inclusion and exclusion principle, except it has the numerator and the denominator swapped to ensure a proper set-theoretic measure.

3 Tensor decomposition

The main entry point to tensor computation is Kolda and Bader (2009), but Rabanser et al. (2017) is also worth consulting.

There is no single generalization of the SVD concept, the two most popular extensions, Canonical Polyadic Decomposition and the more general Tucker, feature different generalized properties. Sidiropoulos et al. (2017) discuss the interpretation of these two different ways of decomposition in signal processing and machine learning points of view.

3.1 Canonical Polyadic Decomposition

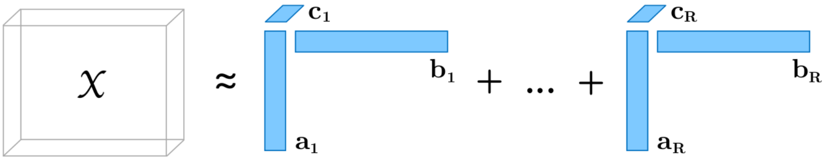

Fig. 1. Canonical Polyadic Decomposition, figure from Rabanser et al. (2017).

Canonical Polyadic Decomposition (CPD, aka. CanDecomp, `Para`llel `Fac`tor model, CanDecomp, rank decomposition, or Kruskal decomposition, (Carroll and Chang, 1970)) expresses a tensor as a minimum-length linear combination of rank-1 tensors. A rank-1 tensors is the tensor product of a collection of vectors, just as the dyadic product of two vectors is a 1-rank matrix, see Figure 1.

The alternating least squares algorithm (ALS, Carroll and Chang (1970); Harshman (1970)) is an iterative method for CPD. In each iteration, all but one of the modes are fixed and the remaining one is fitted. ALS does not guarantee convergence, and even if it converges, this cannot be detected in a trivial way. Orth-ALS (Sharan and Valiant, 2017) improves on ALS.

3.2 Tucker decomposition

While CPD seems more relevant for linguistics representation, we also discuss Tucker decomposition, because it can be computed much more efficiently. Tucker decomposition (aka. `H`igher `O`rder SVD, Tucker (1966)) factorizes a tensor into a core tensor $\mathcal{G}$ multiplied by a matrix along each mode, see Figure 2. In the case of

$$\text{subject} \times \text{verb} \times \text{object}$$

tensors, rows of the three matrices contain embedding vectors of entities (subjects or objects) and those of verbs (“relation”), and entries of the core tensor $\mathcal{G}$ determine the

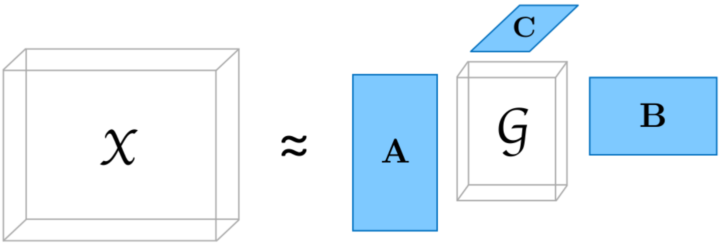

Fig. 2. Tucker Decomposition, figure from Rabanser et al. (2017).

levels of interactions between the former three. Tucker decomposition is not unique, because we can transform $\mathcal{G}$ without affecting the fit if we apply the inverse of that transformation to the factor matrices. Uniqueness can be improved (Kolda and Bader, 2009) by imposing e.g. sparsity, making the elements small, or making the core "all-orthogonal". Other priors and constraints in tensor learning involve non-negativity and independence (Lahat et al., 2015).

4 Experiments

In this section, we report our experiments. After introducing, in Section 4.1, the corpus that serves as the basis of our empirical investigations, Section 4.2 compares association measures, the two alternatives for treating missing arguments, the two decomposition algorithms, and some other hyper-parameters (the decomposition rank and the frequency cutoff) in the classical task of predicting the similarity of English subject-verb-object triples (Kartsaklis and Sadrzadeh, 2014). Then in Section 4.3, we investigate the latent dimensions qualitatively. Section 4.4 compares the embedding vector of each noun as a subject versus an object, to see how differently nouns behave in the two roles.

4.1 Experimental setting: the corpus and the similarity task

In our experiments, we took the occurrence counts of $\langle\text{subject}, \text{verb}^4, \text{direct object}\rangle$ triples from the automatically dependency-parsed (Nivre et al., 2016) English corpus DepCC (Panchenko et al., 2018), irrespectively of whether there were other arguments or adjuncts. Regarding empty fillers, we investigated two alternatives: including them (represented by a fixed string) or discarding them from our statistics. `tensorly` (Kossaifi et al., 2016) was used for CPD and (general and non-negative) Tucker decomposition of tensors. For tensor population in `COO`rdinate format, we use the `sparse` Python library.

Our quantitative tests are based on a classical similarity data-set for English transitive verb structures (SVO triples) by Kartsaklis and Sadrzadeh (2014, KS14). The data-set contains triples with gold (human) similarity scores. We represent SVO triples

[4] *Verb* means, in UD terms, that the `upos` starts with `VB`.

by concatenating the corresponding subject, verb, and object embedding vector (we experimented with normalizing the vectors, but we did not find it useful), and computed the Spearman correlation between the cosine similarities of the (long) vectors in each pair with the human scores.

cutoff	shape with unfilled	shape without unfilled
1	(324 196, 90 606, 287 967)	(206 488, 41 075, 188 619)
10	(160 629, 37 427, 129 694)	(109 432, 19 824, 92 635)
100	(92 999, 20 937, 69 536)	(71 768, 13 907, 57 420)
1000	(44 168, 10 444, 32 359)	(40 309, 8 838, 30 280)
10000	(13 765, 5 070, 12 313)	(13 610, 4 895, 12 115)
100000	(3 474, 2 313, 4 120)	(3 463, 2 308, 4 108)
1000000	(546, 814, 981)	(545, 813, 980)
10000000	(36, 194, 87)	(35, 194, 86)

Table 1. The length of each axis, i.e. the number of subjects, verbs, and objects, at different frequency cutoffs.

4.2 Quantitative results in transitive structure similarity

We populated tensors with the association measures introduced in Section 2. The statistics were based on either including empty argument fillers (i.e. treating all arguments "optional") or excluding these occurrences. We took different cutoffs and computed non-negative or general CPD or Tucker decompositions in different ranks. Table 1 shows the length of each axis, i.e. the number of subjects, verbs, and objects, at different frequency cutoffs.

Correlations we obtain in the subject-verb-object task are shown in Table 2. The properties of the original sparse tensor (the association measure, whether empty fillers are included, and the frequency cutoff) are show on the left of the vertical line, while those of the decompositions (non-negative or general CPD or Tucker decompositions to the specified rank) are shown on the right. The table shows, in addition to the the best setting, each setting obtained by changing one meta-parameter. The best result is obtained by non-negative CPD. The horizontal lines shows the place of our best general Tucker, general CPD, and non-negative Tucker decompositions, which we discuss later in this subsection. In Tucker decompositions, we use the same rank among all axes.

We obtained the best correlation, 0.7360, from the decomposition of a tensor populated with salience-weighted PMI values, including empty fillers, and setting the frequency cutoff to 1 million, i.e. restricting the axes of the tensor to the subjects, verbs, and objects that appear at least 1 million times. This best correlation was obtained with non-negative CPD in rank 64. This correlation value is in the same range as 0.76 obtained by Hashimoto et al. (2014) with a much more complex system that used to be the state-of-the-art, when this task was fashionable.

The table shows the correlation obtained by changing each (meta-)parameter. While the results seem to be relatively robust with respect to the decompositions *rank*, it may

assoc measure	unfilled	cutoff	non-negative	decomp algo	rank	corr
pmi-sali	included	1 000 000	non-neg	parafac	64	0.7359
pmi-sali	included	1 000 000	non-neg	parafac	128	0.7097
pmi	included	1 000 000	non-neg	parafac	64	0.6857
pmi-sali	included	1 000 000	non-neg	parafac	32	0.6773
pmi-sali	included	*300 000*	non-neg	parafac	64	0.6630
npmi	included	1 000 000	non-neg	parafac	64	0.6602
dice-sali	included	1 000 000	non-neg	parafac	64	0.4709
pmi-sali	excluded	1 000 000	non-neg	parafac	64	0.4578
pmi-sali	included	1 000 000	*general*	parafac	64	0.4560
ldice	included	1 000 000	non-neg	parafac	64	0.4409
log-freq	included	1 000 000	non-neg	parafac	64	0.4322
iact-sali	included	1 000 000	non-neg	parafac	64	0.4112
niact	included	1 000 000	non-neg	parafac	64	0.4068
pmi-sali	included	*3 000 000*	non-neg	parafac	64	0.3936
iact	included	1 000 000	non-neg	parafac	64	0.3248
pmi-sali	included	1 000 000	non-neg	*tucker*	64	0.2989

Table 2. Quantitative results: correlations in the subject-verb-object triple similarity task (Kartsaklis and Sadrzadeh, 2014) obtained with word embeddings of tensor decompositions.

be interesting that when we concatenate the subject, the verb, and the object embedding vectors, 64 dimensional each, we get a vector in the famous range of a couple of hundreds of dimensions, which proved to work well in many different scenarios like LSA and static word embeddings (see the introduction).

As for our *association measures,* different weighted variants (salience, vanilla, or normalization) of PMI work the best, followed by log-Dice and log frequency. Variants of interaction information performs the worst.

The inclusion of empty fillers, the frequency cutoff, and the decomposition rank are all related to the *size of the tensors*. While we have already seen that the decomposition rank does not have a great influence on the results, if we exclude empty fillers, a more generous frequency cutoff may theoretically lead to better results than if we change only one of these two parameters. It turns out, that we can indeed get relatively good result (0.694181) this way, but with general Tucker decomposition (instead of non-negative CPD) and log-Dice (instead of salience-weighted). The cutoff is 1 million.

Non-negative decomposition is advantageous from the interpretational point of view, because in our experiments, they resulted in embedding matrices which are *sparse* in the broad sense that most coordinates are low. Figure 3 shows a histogram of the matrix elements. Note that the vertical axis, which corresponds to the histogram count in each bin, is logarithmic. The figure suggests that frequency decreases faster than exponentially as larger weights are considered. The good performance of non-negative CPD suggests that non-negativity introduces meaningful structure. Sparsity raises the hope that coordinate are interpretable, i.e. they correspond to concepts or properties.

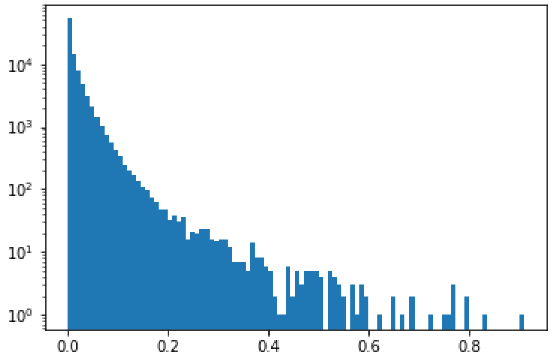

Fig. 3. The histogram of the verb embedding matrix elements. Note that the vertical axis, which corresponds to the histogram count in each bin, is logarithmic. The figure suggests that frequency decreases faster than exponentially as larger weights are considered.

CPD has the advantage that it maps the *modes in the same space*. In our case, this is the most interesting for subjects and objects: we can compare the same noun in the two roles. We return to this in Section 4.4.

While our best results have been obtained with non-negative CPD, we discuss general Tucker and CPD and non-negative Tucker as well. Results with general decompositions and non-negative Tucker are shown in Table 3 and Table 4, respectively. General Tucker and CPD and non-negative Tucker all prefer normalized PMI as the association measure, disfavor interaction information, and results with log frequency and log Dice vary. General and non-negative Tucker obtains the best results with the same rank as non-negative CPD, and the two non-negative decomposition algorithms also share the value for a best cutoff. It is inconclusive whether it is advantageous to include occurrences with unfilled arguments in our statistics.

4.3 Qualitative analysis of latent dimensions

Now we investigate the latent dimensions obtained by tensor decomposition. We experimented with non-negative and general CPD and Tucker decomposition with the hyper-parameters that reached the best result in the SVO-similarity task.

The latent dimensions are shown in Tables 5 to 7. (Dimensions with general Tucker are degenerate, and they are omitted to save space.) Each line corresponds to a latent dimension. Dimensions are visualized by the words with the greatest coordinates in the dimension. Blocks represent dimension triples. $\emptyset$ denotes that the corresponding grammatical function is unfilled. Some latent dimensions, like the first one in our non-negative CPD are dominated by (the empty filler and) pronouns. In these cases we *emphasize* the first contentful filler. `-rrb-` stands for right round brackets, and its appearance may be an artifact of the corpus.

assoc measure	unfilled	cutoff	rank	correlation
npmi	included	100 000	64	0.7191
pmi-sali	included	100 000	64	0.7049
log-freq	included	100 000	64	0.6883
pmi	included	100 000	64	0.6759
npmi	included	30 000	64	0.6729
ldice	included	100 000	64	0.6685
ldice-sali	included	100 000	64	0.6666
npmi	included	300 000	64	0.6598
npmi	included	100 000	128	0.6540
npmi	included	100 000	32	0.6042
npmi	excluded	100 000	64	0.5207
iact-sali	included	100 000	64	0.5059
niact	included	100 000	64	0.4632
iact	included	100 000	64	0.4316

assoc measure	unfilled	cutoff	rank	correlation
npmi	excluded	300 000	256	0.6383
pmi-sali	excluded	300 000	256	0.6166
pmi	excluded	300 000	256	0.5811
npmi	excluded	1 000 000	256	0.5754
npmi	excluded	100 000	256	0.5713
npmi	excluded	300 000	512	0.5677
npmi	excluded	300 000	128	0.5290
npmi	excluded	30 000	256	0.5239
npmi	included	300 000	256	0.5070
log-freq	excluded	300 000	256	0.2465
ldice	excluded	300 000	256	0.2093
iact-sali	excluded	300 000	256	0.1280
niact	excluded	300 000	256	0.0726
iact	excluded	300 000	256	0.0615

Table 3. Results with general Tucker (left) and general CPD (right).

assoc measure	unfilled	cutoff	rank	correlation
npmi	excluded	1 000 000	64	0.5186
npmi	excluded	1 000 000	128	0.5102
npmi	excluded	300 000	64	0.4814
pmi	excluded	1 000 000	64	0.4563
pmi-sali	excluded	1 000 000	64	0.4387
npmi	excluded	1 000 000	32	0.3753
npmi	excluded	3000 000	64	0.3366
npmi	optional	1 000 000	64	0.2889
iact	excluded	1 000 000	64	0.0989
log-freq	excluded	1 000 000	64	0.0763
ldice	excluded	1 000 000	64	0.0698
ldice-sali	excluded	1 000 000	64	0.0619
niact	excluded	1 000 000	64	0.0454
iact-sali	excluded	1 000 000	64	0.0064

Table 4. Results with non-negative Tucker.

dim	words
0	∅, that, which, it, *story*, he, they, who, what, one, she, work, event, -rrb-, this, you…
0	catch, attract, draw, pay, deserve, capture, gain, grab, get, receive, focus, require,…
0	attention, eye, crowd, interest, fire, visitor, audience, conclusion, breath, people, …
1	∅, who, we, he, I, you, she, they, -rrb-, *student*, member, people, group, Center, parti…
1	attend, host, hold, organize, schedule, enjoy, join, arrange, cancel, miss, watch, pla…
1	meeting, event, conference, session, party, show, school, class, dinner, church, tour,…
2	that, which, it, this, ∅, *change*, factor, they, choice, condition, decision, issue, -rr…
2	affect, impact, influence, improve, hurt, reflect, benefit, change, damage, enhance, a…
2	ability, performance, health, outcome, life, quality, result, business, development, e…
3	file, which, page, site, that, it, book, report, section, document, collection, websit…
3	contain, include, provide, have, list, feature, display, show, comprise, present, give…
3	information, link, material, number, list, datum, name, content, statement, reference,…

Table 5. Latent dimensions with Non-negative ParaFac

dim	words
5	court, Court, judge, panel, official, we, he, it, authority, government, -rrb-, Board,…
10	reject, dismiss, deny, grant, hear, consider, decide, accept, throw, resolve, sustain,…
7	motion, appeal, claim, request, argument, case, challenge, application, complaint, att…
4	revenue, sale, share, price, stock, production, cost, rate, order, volume, number, fut…
3	rise, fall, increase, jump, drop, decline, climb, decrease, grow, gain, slip, represen…
1	percent, %, $, increase, point, most, rate, level, average, less, matter, value, cost,…
11	hotel, property, room, restaurant, home, Center, house, location, facility, House, are…
8	offer, boast, feature, have, provide, include, enjoy, serve, accommodate, occupy, prep…
9	room, pool, accommodation, access, facility, restaurant, variety, service, view, range…
6	board, Council, Board, Commission, Committee, member, committee, Congress, Court, cour…
2	approve, adopt, reject, pass, consider, review, endorse, propose, award, recommend, ac…
2	resolution, request, budget, plan, proposal, contract, change, application, project, i…

Table 6. Latent dimensions with Non-negative Tucker

In the case of CPD, the dimensions are enumerated in the order as returned by the algorithm. With Tucker, the values g_{ijk} in the core tensor $\mathcal{G}$ represent the interaction between the ith latent dimension for subjects, the jth one for verbs, and the kth one for objects. We sorted the triples of SVO latent dimensions in our best non-negative and general Tucker decomposition by this interaction strength. The index of each dimension, as returned by the algorithm, is also shown in the table. E.g. the first block in non-negative Tucker shows that the strongest interaction is between the 5th latent dimension of subjects, the 10th one for verbs, and the 7th one for objects. Note that in the non-negative case, $g_{ijk} \geq 0$, so we do not have to take the absolute value. Dimensions obtained with the two *non-negative algorithms* seem semantically interpretable, while those from general decomposition are less convincing.

dim	words
0	Israel, group, government, Foundation, Association, company, -rrb-, military, army, Cl…
0	launch, wage, suspend, mount, begin, run, fund, organize, sponsor, administer, carry, …
0	campaign, attack, program, initiative, operation, strike, programme, website, effort, …
1	user, you, application, customer, developer, visitor, client, processor, device, User,…
1	access, select, specify, upload, view, enter, edit, browse, click, create, retrieve, m…
1	file, datum, content, document, page, parameter, site, folder, node, Internet, informa…
2	device, assembly, means, structure, system, element, plate, section, interface, unit, …
2	comprise, include, contain, have, utilize, employ, represent, say, mean, control, enab…
2	layer, element, device, tube, housing, spring, electrode, pump, plate, container, memb…
3	attorney, plaintiff, defendant, party, respondent, prosecutor, State, lawyer, governme…
3	file, receive, oppose, make, give, present, withdraw, handle, publish, drop, provide, …
3	motion, notice, petition, appeal, response, answer, objection, charge, request, submis…

Table 7. Latent dimensions with General ParaFac

4.4 Comparing subject and object vectors

Tensor decomposition can shed light on how differently nouns behave as subjects and as objects. This question is related to symmetric facotrization (Bailey et al., 2018), which imposes symmetry constraints between the embeddings of the same entities in different modes (in our case, between the embeddings of the same noun as a subject or an object). Our approach is complementary, based on that CPD maps nouns as subjects and objects in the same space.

In our experiments, we consider (non-negative) CPD decomposition with the hyper-parameters that proved best in English SVO-similarity. We computed the (unnormalized) dot product similarity between the subject and object vector of each noun, and sorted all the nouns by this similarity. The largest distance is found with *∅, he, she, they, I, device, system, that, you, it…*, while the most symmetric nouns are *doubt, reality, future, same, hope, feeling, mine, reason, consumer, plenty…* A possible explanation is that the former, especially personal pronouns, are much more frequent in agentive roles than other nouns, while they are infreqent in patient roles. Words in the second group can be framed in language both as animate and as inanimate. *Future* or *hope* are not alive in the biological sense, but they are often attributed agentive roles (what can be called a metaphorical use of language but being metaphorical dos not mean that the usage is preipheral, as it has been noted by linguists).

5 Conclusion and future work

Weighted variants of positive pointwise mutual information proved better than the considered alternatives in modelling subject-verb-object structure similarity. It does not matter, whether we include occurrences with unfilled arguments in our statistics. Our best results were obtained with non-negative CPD. The best frequency cutoff and the decomposition rank is the same for the two non-negative decomposition algorithms, which raises the hope that these hyper-parameters of non-negative CPD can be fine-tuned based on the much faster non-negative Tucker, but this needs to be tested in other setups. Our experiments provided lexically interpretable latent dimensions and verb clusters, and the difference between subject and object embeddings can be related to animacy, at least in the case of non-negative CPD.

Tensor decompositions offer a direction orthogonal to the mainstream (Rogers et al., 2020) in the data-driven understanding of linguistic structure. Our line of research can be extended cross-lingually (Vulić et al., 2017; Majewska et al., 2018; Sun et al., 2010). I thank Gábor Berend, András Kornai, Bálint Sass, and Tibor Szécsényi for useful comments.

Bibliography

Bailey, E., Meyer, C., Aeron, S.: Learning semantic word respresentations via tensor factorization (2018), `https://openreview.net/forum?id=B1kIr-WRb`, arXiv:1704.02686

Bouma, G.: Normalized (pointwise) mutual information in collocation extraction. In: GSCL 2009: International Conference of the German Society for Computational Linguistics and Language Technology (2009)

Carroll, J.D., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. Psychometrika 35, 283–319 (1970)

Van de Cruys, T.: A non-negative tensor factorization model for selectional preference induction. In: Proceedings of the Workshop on Geometrical Models of Natural Language Semantics. pp. 83–90. Association for Computational Linguistics, Athens, Greece (3 2009), `https://www.aclweb.org/anthology/W09-0211`

Van de Cruys, T.: Two multivariate generalizations of pointwise mutual information. In: Proceedings of the Workshop on Distributional Semantics and Compositionality. pp. 16–20. Association for Computational Linguistics, Portland, Oregon, USA (6 2011), `https://www.aclweb.org/anthology/W11-1303`

Van de Cruys, T., Poibeau, T., Korhonen, A.: A tensor-based factorization model of semantic compositionality. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1142–1151. Association for Computational Linguistics, Atlanta, Georgia (6 2013), `https://www.aclweb.org/anthology/N13-1134`

Frandsen, A., Ge, R.: Understanding composition of word embeddings via tensor decomposition. In: 7th International Conference on Learning Representations, ICLR 2019 (5 2019), `https://openreview.net/forum?id=H1eqjiCctX`, arXiv preprint arXiv:1902.00613

Fried, D., Polajnar, T., Clark, S.: Low-rank tensors for verbs in compositional distributional semantics. In: ACL (2015)

Gittens, A., Achlioptas, D., Mahoney, M.W.: Skip-gram – zipf + uniform = vector additivity. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 69–76. Association for Computational Linguistics (2017), `http://aclweb.org/anthology/P17-1007`

Grefenstette, E., Sadrzadeh, M.: Experimenting with transitive verbs in a DisCoCat. In: Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics. pp. 62–66. Association for Computational Linguistics, Edinburgh, UK (7 2011), `https://www.aclweb.org/anthology/W11-2507`

Harshman, R.A.: Foundations of the parafac procedure: Models and conditions for an "explanatory" multi-modal factor analysis. UCLA Working Papers in Phonetics 16, 1–84 (1970), `http://publish.uwo.ca/~harshman/wpppfac0.pdf`

Hashimoto, K., Stenetorp, P., Miwa, M., Tsuruoka, Y.: Jointly learning word representations and composition functions using predicate-argument structures. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. p. 1544–1555 (2014)

Hashimoto, K., Tsuruoka, Y.: Learning embeddings for transitive verb disambiguation by implicit tensor factorization. In: 3rd Workshop on Continuous Vector Space Models and their Compositionality (2015)

Jenatton, R., Roux, N.L., Bordes, A., Obozinski, G.: A latent factor model for highly multi-relational data. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2. pp. 3167–3175. NIPS'12, Curran Associates Inc. (2012), `http://dl.acm.org/citation.cfm?id=2999325.2999488`

Kalivoda, Á.: Igekötős szerkezetek a magyarban. Ph.D. thesis, Pázmány Péter Katolikus Egyetem, Bölcsészet- és Társadalomtudományi Kar, Nyelvtudományi Doktori Iskola, Budapest (2021)

Kartsaklis, D., Sadrzadeh, M.: A study of entanglement in a categorical framework of natural language. In: The 11th workshop on Quantum Physics and Logic (6 2014), arXiv:1412.8102

Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D.: Sketch engine. In: Williams, G., Vessier, S. (eds.) Proceedings of Euralex. pp. 105–116. Lorient, Université de Bretagne-Sud, Faculté des lettres et des sciences humaines (7 2004)

Kipper, K., Korhonen, A., Ryant, N., Palmer, M.: A large-scale classification of English verbs. Language Resources and Evaluation 42(1), 21–40 (2008)

Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM review 51(3), 455–500 (2009)

Kossaifi, J., Panagakis, Y., Anandkumar, A., Pantic, M.: Tensorly: Tensor learning in python. Journal of Machine Learning Research (JMLR) 20, 1–6 (2016), arXiv preprint arXiv:1610.09555

Lahat, D., Adali, T., Jutten, C.: Multimodal data fusion: an overview of methods, challenges, and prospects. Proceedings of the IEEE 103(9), 1449–1477 (2015)

Landauer, T.K., Dumais, S.T.: A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review 104(2), 211 (1997)

Levin, B.: English Verb Classes and Alternations: A Preliminary Investigation. University of Chicago Press (1993)

Levy, O., Goldberg, Y.: Dependency-based word embeddings. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 302–308. Association for Computational Linguistics, Baltimore, Maryland (06 2014a), `http://www.aclweb.org/anthology/P14-2050`

Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 27. pp. 2177–2185 (2014b)

Levy, O., Remus, S., Biemann, C., Dagan, I.: Do supervised distributional methods really learn lexical inference relations? In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 970–976. Association for Computational Linguistics, Denver, Colorado (2015), `https://www.aclweb.org/anthology/N15-1098`

Majewska, O., Vulić, I., McCarthy, D., Huang, Y., Murakami, A., Laippala, V., Korhonen, A.: Investigating the cross-lingual translatability of VerbNet-style classification. Language Resources and Evaluation 52(3), 771–799 (2018)

Manin, D.Y.: Zipf's law and avoidance of excessive synonymy. Cognitive Science 32, 1075–1098 (2008)

McGill, W.: Multivariate information transmission. Transactions of the IRE Professional Group on Information Theory 4(4), 93–111 (1954)

McInnes, L., Healy, J., Astels, S.: hdbscan: Hierarchical density based clustering. The Journal of Open Source Software 2(11) (3 2017), `https://doi.org/10.21105%2Fjoss.00205`

McInnes, L., Healy, J., Saul, N., Grossberger, L.: Umap: Uniform manifold approximation and projection. The Journal of Open Source Software 3(29), 861 (2018)

Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings (05 2013a), `http://arxiv.org/abs/1301.3781`

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013b), `https://bit.ly/39HikH8`

Mnih, A., Hinton, G.: Three new graphical models for statistical language modelling. In: Proceedings of the 24th international conference on Machine learning. pp. 641–648. ACM (2007)

Nivre, J., de Marneffe, M.C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D.: Universal dependencies v1: A multilingual treebank collection. In: Proc. LREC 2016. pp. 1659–1666 (5 2016)

Panchenko, A., Ruppert, E., Faralli, S., Ponzetto, S., Biemann, C.: Building a web-scale dependency-parsed corpus from common crawl. In: Proceedings of LREC 2018. ELRA (2018)

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)

Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543. Association for Computational Linguistics (2014), `http://www.aclweb.org/anthology/D14-1162`

Polajnar, T., Rimell, L., Clark, S.: Using sentence plausibility to learn the semantics of transitive verbs. In: NIPS Learning Semantics Workshop (2014), in arXiv, some minor errata fixed.

Rabanser, S., Shchur, O., Günnemann, S.: Introduction to tensor decompositions and their applications in machine learning (11 2017), `http://arxiv.org/abs/1711.10781v1`, arXiv:1711.10781 [stat.ML]

Rogers, A., Kovaleva, O., Rumshisky, A.: A primer in bertology: What we know about how bert works. arXiv preprint arXiv:2002.12327 (2020)

Rychlý, P.: A lexicographer-friendly association score. In: Proceedings of Recent Advances in Slavonic Natural Language Processing. pp. 6–9 (2008)

Sass, B.: 28 millió szintaktikailag elemzett mondat és 500000 igei szerkezet [28 million syntactically analyzed sentences and 500 000 verb constructions in Hungarian]. In: Attila, T., Viktor, V., Veronika, V. (eds.) XI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2015). pp. 303–308. Szegedi Tudományegyetem Informatikai Tanszékcsoport (2015)

Shannon, C.E., Weaver, W.W.: The Mathematical Theory of Communication. University of Illinois Press, Urbana (1949)

Sharan, V., Valiant, G.: Orthogonalized ALS: A theoretically principled tensor decomposition algorithm for practical use. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017. pp. 3095–3104 (8 2017), `http://proceedings.mlr.press/v70/sharan17a.html`

Sidiropoulos, N.D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E.E., Faloutsos, C.: Tensor decomposition for signal processing and machine learning. IEEE Transactions on signal processing 65(13), 3551–3582 (7 2017), `https://doi.org/10.1109/TSP.2017.2690524`

Sun, L., Korhonen, A., Poibeau, T., Messiant, C.: Investigating the cross-linguistic potential of VerbNet: style classification. In: Proceedings of the 23rd International Conference on Computational Linguistics. pp. 1056–1064. Association for Computational Linguistics (2010)

Szécsényi, T.: Argumentumszerkezet-variánsok korpusz alapú meghatározása [Corpus-based identification of Hungarian argument structure variants]. In: Berend, G., Gosztolya, G., Vincze, V. (eds.) XV. Magyar Számítógépes Nyelvészeti Konferencia. pp. 315–331. Szegedi Tudományegyetem TTIK, Informatikai Intézet (1 2019)

Tucker, L.R.: Some mathematical notes on three-mode factor analysis. Psychometrika 31(3), 279–311 (1966)

Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research 37, 141–188 (2010)

Villada Moirón, M.B.: Data-driven identification of fixed expressions and their modifiability. Ph.D. thesis, University of Groningen (2005)

Vulić, I., Mrkšić, N., Korhonen, A.: Cross-lingual induction and transfer of verb classes based on word vector space specialisation. arXiv preprint arXiv:1707.06945 pp. 2546–2558 (9 2017), `https://www.aclweb.org/anthology/D17-1270`

Watanabe, S.: Information theoretical analysis of multivariate correlation. IBM Journal of research and development 4(1), 66–82 (1960)

Zhuang, Y., Xie, J., Zheng, Y., Zhu, X.: Quantifying context overlap for training word embeddings. In: EMNLP (2018)

6 Follow-up

In this section, we report experiments, wich didn't appear in the official version of this paper.

6.1 Clustering verb vectors

Semantic *classes* of verbs like those in VerbNet (Kipper et al. (2008), which are refinements of Levin (1993)'s classes) may be induced by clustering verb embedding vectors. If clusters obtained in unsupervised fashion correspond to gold verb classes, ambiguous verbs like *play* mentioned in Section 1 may be detected as outliers from the clusters, as their uses are composed of occurrences corresponding to different clusters.

Our recipe for obtaining verb clusters consists of mapping verb embedding vectors to a lower dimensional space with UMAP (McInnes et al., 2018), and clustering them

# verbs	verbs
702	have, do, get, go, take, think, know, want, need, give, look, work, provide, try, …
131	live, talk, stand, die, walk, wait, sit, stay, wonder, care, arrive, fly, gon, sleep, …
86	kill, catch, trust, bear, email, marry, fuck, date, judge, bless, honor, forgive, beg,…
85	add, eat, produce, deliver, prepare, drink, spread, cook, burn, taste, wash, supply, …
80	use, develop, manage, perform, complete, replace, install, connect, test, conduct, …
80	let, reach, hit, cost, exceed, rate, approach, /, -lsb-_VBD, rank, -lsb-_VB, \, -lsb-_…
79	put, break, pull, throw, push, lay, stick, grab, touch, press, suck, kick, shake, …
77	identify, commit, defend, repeat, expose, separate, dig, heal, dress, distinguish, …
76	send, check, view, click, display, generate, update, access, search, store, delete, …
65	leave, enter, visit, fill, explore, ride, clean, cross, surround, locate, clear, rent,…
59	be, come, start, happen, seem, begin, continue, appear, lead, end, occur, prove, …
58	help, keep, bring, remind, hurt, strike, worry, blow, inspire, bother, surprise, suit,…
57	tell, ask, call, thank, please, join, contact, become, assist, hire, name, engage, …
51	pay, spend, save, raise, determine, compare, charge, measure, adjust, predict, invest,…
46	make, see, find, love, like, hear, enjoy, remember, miss, guess, recommend, notice, …
43	understand, discover, recognize, examine, evaluate, investigate, acknowledge, assess, …
43	face, experience, address, fix, handle, suffer, solve, celebrate, resolve, mark, …
39	receive, win, lose, earn, gain, extend, deserve, capture, retain, lack, exercise, …
37	plan, fail, focus, vote, act, deal, attempt, rely, struggle, participate, benefit, …

Table 8. Verb clusters obtained from our verb embedding vectors in an unsupervised fashion. The smallest cluster is omitted to save space.

with HDBScan (McInnes et al., 2017), which is a hierarchical, density based clustering algorithm. Dimensionality reduction is needed because density makes little sense in hundreds of dimensions. Our choices of UMAP meta parameters are the following: We map verb embedding vectors to 16 or 32 dimensions (fine-tuned in a comparison to VerbNet, see later). In HDBScan, we set the number of neighbors to 30 and the minimum distance to 0, following the recommendations at `readthedocs`[5]. The metric in the ambient space (i.e. the original, high-dimensional one) is cosine. Minimum cluster size is 15 or 5, and the related parameter of `min_samples` is 5.

We compare non-negative and general CPD and Tucker decompositions. The parameters of the tensor and its decompositions are set to the value with the best score in the SVO-similarity task. We set one hyper-parameters of UMAP and HDBScan each, namely the dimension we map to and minimum cluster size, based on comparison to VerbNet classes. In these computation we take VerbNet from the `nltk.corpus` package. In many cases, there are more class IDs associated to a verb. We take the first one, as returned by the corresponding function. Out-of-vocabulary verbs are treated as a separate class. We compare are clustering to VerbNet classes with adjusted rand score in scikit-learn (Pedregosa et al., 2011). We get the greatest score with non-negative Tucker (embeddings mapped to 16 dimensions, and minimum cluster size set to 15).

[5] `https://umap-learn.readthedocs.io/en/latest/clustering.html#umap-enhanced-clustering`

Table 8 shows the greatest clusters of English verbs. The greatest cluster, separated by a line in the table is the one called `-1` in HDBScan. It contains points that “fall out”’ in the hierarchy. The algorithm considers them outliers[6]. In our case, it seems that they are general verbs, especially those that we find in light verb constructions. The remaining clusters seem to be semantically coherent.

6.2 Hungarian data and preverbs

preverb	verb			args			gloss
∅	bíz(ik)	NOM		-bAn ‘in’			trust sth
(rá) ‘onto’	bíz	NOM	ACC		-rA ‘onto’		entrust sg to sy
meg Perfect	bíz(ik)	NOM		-bAn ‘in’			trust sy
meg Perfect	bíz	NOM	ACC			INS	entrust sy with sg
el ‘away’	bíz(za)	NOM	self-ACC				get conceited

Table 9. Argument structure variants of the Hungarian verb *bíz(ik)* based on Szécsényi (2019).

We propose hypotheses for future work. In Hungarian, there are two phenomena that interfere with verb agument structure and ambiguity. Table 9, based on Szécsényi (2019), illustrates these with the verb *bíz(ik)* ‘trust’. We can see that preverbs (verb particles, which can modify both the aspect and the meaning of a verb (Kalivoda, 2021)) interfere with verb meaning, and the apparently incidental appearance of the suffix *-ik* (which can be argued to be related to unaccusativity) increases data sparsity. In our preliminary experiments, we built a *subject* × *preverb* × *verb* × *object* tensor from verb constructions in the data-base of the Mazsola verb argument browser (Sass, 2015). In this earlier, unpublished phase of the project, we used CPD decomposition, solved by the Orth-ALS (Sharan and Valiant, 2017) algorithm. For the future, we suggest introducing a mode for *-ik*. The “vocabulary” of this axis would consists of only two choices: with or without *-ik*. The hypothesis is that this tensor would profit from denser data representation.

[6] See `https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html#`