Training Distributional Matrices for Dutch Transitive Verbs with an Application in Ambiguous Relative Clauses

Luka van der Plas

4119142

Bachelorscriptie Taalwetenschap

Finale versie

Begeleider: Michael Moortgat

Table of contents

Summary

Distributional semantic models represent word meaning as vectors which reflect word distribution in corpora. The field of compositional distributional semantics investigates how these vectors can be composed to represent constituent or sentence meaning. Within this field, the categorial approach trains words that are assigned function types in typelogical grammars, including verbs, as higher-order tensors. This paper implements this approach by training decoupled verb matrices for Dutch transitive verbs and analysing their performance in derivationally ambiguous Dutch relative clauses. In the training of verb matrices, distributional data were partially imported from Tulkens, Emmery & Daelemans (2016) and partially extracted from the Lassy Groot corpus (Van Noord, 2006). Verb matrices were trained using Ridge regression. Analysing the performance of these matrices in relative clauses, it is found that trained matrices are generally sound, but show very little differentiation between subjects and objects. Possible causes and implications of this surprising result are discussed.

Introduction

The field of distributional semantics is based on the notion that the meanings of words can be inferred from the linguistic context in which they appear (Schütze, 1998). In practice, this notion is implemented by mapping words to vectors that represent their distribution in large corpora, where the distribution is normally assessed in an *n*-word window around instances of the target word. This framework has achieved promising results in tasks like word sense disambiguation (Schütze, 1998; McCarthy, Koeling, Weeds & Carroll, 2004), and is found to reflect human similarity judgements (McDonald & Ramscar, 2001). These results illustrate the validity of the distributional hypothesis.

When moving beyond the semantics of individual words to that of constituents and clauses, however, this field seems inherently limited in its attempt to represent natural language with flat distribution statistics. Indeed, when word vectors are used to analyse the meaning of larger constituents, commutative models of combining individual word vectors like elementwise multiplication and vector addition quickly show a decay in quality (Mitchell & Lapata, 2008; Clark, Rimell, Polajnar & Maillard, 2016).

Integrating this powerful method of representing word semantics with the structured nature of language is the primary concern of compositional distributional semantics, which investigates how to combine distributional representations of words into representations of larger constituents. The categorial framework represents one approach within this field, for which the foundations are laid out in Coecke, Sadrzadeh & Clark (2010). At its core, the framework is based on a type-driven grammar, with the insight that while fundamental types can be represented by vectors, function types can be represented as higher-order tensors, thus allowing an elegant translation from operations in typelogical grammars to operations in multilinear algebra.

In concrete terms, the categorial approach proposes a basic format in which function type lemmas are represented as linear transformations, and a general guideline for how syntactic types are translated to tensor spaces (Baroni, Bernardi & Zamparelli, 2014; Clark et al., 2016). The basic notion of the framework is as follows. Words that are assigned atomic types in typelogical grammars, like nouns, are represented by distribution vectors. The categorial approach then proposes that for a noun like *car*, the semantics of the phrase *red car* should be represented within the same vector space as *car*, denoted the N space. The word *red* can then be represented by a linear transformation on *car*, i.e. as a matrix in N ⊗ N, as in (1), taken from Clark et al. (2016, p. 11).

(1)

$$\begin{array}{cccc} \overline{red} & \overrightarrow{car} & & \overrightarrow{red\ car} \\ \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} & \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} & = & \begin{pmatrix} rc_1 \\ rc_2 \\ rc_3 \end{pmatrix} \end{array}$$

The holistic vector $\overrightarrow{red\ car}$ is again distributional, so its value can be extracted from corpus data in the same way as that of $\overrightarrow{car}$, based on the distribution of the phrase *red car*. For enough pairs of *X* and *red X*, it is possible to train a general $\overline{red}$ matrix using regression. The matrix $\overline{red}$ can then be used to calculate a distribution vector for *red unicycle* as the dot product $\overline{red} \cdot \overrightarrow{unicycle}$, even if the phrase *red unicycle* was never observed in the corpus.

Moving beyond the relatively simple semantic type of adjectives, the categorial approach provides a guideline of how to convert syntactic categories in a combinatorial grammar like

Combinatory Categorial Grammar to tensor spaces (Maillaird, Clark & Grefenstette, 2014). First, two fundamental vector spaces are defined. There is the noun space, N, as used above, and a sentence space S, which represents the semantics of sentences. These correspond to the atomic types of noun phrases (*NP*) and sentences (*S*) in CCG. Second, function types in CCG can be easily mapped to (multi)linear transformations by defining a homomorphism between semantic function types and tensor spaces. CCG features two forms of function application, namely forward and backward application, shown in the examples in (2).

(2) a. red car

NP/*NP* *NP* ⇒ *NP*

b. Alice walks

NP *S\NP* ⇒ *S*

The tensor spaces corresponding to function types in CCG are then derived by converting the atomic types to the basic vector spaces and replacing the slash operators in CCG representations with tensor product operators (idem), as in (3).

(3) a. red car

NP/*NP* *NP*

N ⊗ N N

b. Alice walks

NP *S\NP*

N S ⊗ N

The above examples contain function types that take only a single argument, resulting in matrix representations. For higher order functions, tensor representations are also of higher orders. For instance, a transitive verb corresponds to a third-order tensor in $S \otimes N \otimes N$. Applying forward and backward application in CCG grammar then corresponds to performing tensor contraction on the resulting tensors.

For a more complete theoretical account of the mapping from CCG to compositional distributional semantics, see Baroni, Bernardi & Zamparelli (2014), Maillaird et al. (2014), Clark et al. (2016). The current study will treat the syntactic representation of the sentence and its mapping to compositional distributional semantics as given, focusing on issues in the implementation of compositional distributional models. In this area, many fundamental questions remain open. One such issue is the realisation of the sentence space S. The grammar of the categorial approach makes no demands on its dimensions or semantics. Consequently, the implementation of the S space is a topic of debate. Possible implementations include a one- or two-dimensional vector space representing plausibility (Clark, 2013; Polajnar, Făgărăşan & Clark, 2014) or a high-dimensional space representing distribution like the N space. In the case of a distributional S space, there is the option to make the N and S spaces identical (Kartsaklis, Sadrzadeh & Pulman, 2012; Rimell, Maillard, Polajnar & Clark, 2016a). The issue of the sentence space will not be the primary focus of the current study, which will assume a separate distributional S space.

Another central issue in the categorial approach is determining effective models of composition. While using a grammar like CCG as a basis can provide a clear architecture for composition, speculation about composition models is worthwhile, since it has proven challenging to find any model that can compete with the structure-blind model of vector addition

(Clark et al., 2016). Additionally, the implementation of a model directly based on CCG is not tractable due to the large number of dimensions that more complex types require.

There are two aspects of compositional distributional models where dimensionality plays a role. The first is the number of dimensions in distribution vectors. A distributional vector for nouns is based on the noun's co-occurrence with other words, which is mapped to a vocabulary V. To allow for representations of more complex meanings, $|V|$ is often chosen fairly large, e.g. 10.000. Such a high dimensionality is typically impractical, including for the training of tensors (see below). As such, it is common to apply some method of dimensionality reduction to obtained word vectors like Single Value Decomposition (Rimell et al., 2016a). Another often-used solution is skip-gram, which trains a lower-dimension vector as it works through the corpus (Mikolov, Sutskever, Chen, Corrado & Dean, 2013).

Another issue in dimensionality is the order of tensors. In type-based grammars, more syntactically complex elements such as quantifiers and adverbs quickly increase in type complexity, resulting in increasingly higher-order tensors: "[i]n practice, syntactic categories such as *((N/N)/(N/N))/((N/N)/(N/N))* are not uncommon […]; such a category would require an *8th-order* tensor" (Clark et al., 2016). The increase in parameters involved in the training of high-order tensors, however, complicates concrete implementation. Not only does this greatly increase the complexity of the tensor representation, but higher-order tensors also require increasingly large training corpora. Besides the increased number of parameters to be trained, there is an increase in the required specificity of training data. To return to the example in (1), the matrix $\overline{red}$ requires a number of holistic vectors $\overrightarrow{red\ X}$ to be derived from corpus data, yet the frequency of each individual combination *red X* will be relatively low, resulting in an inaccurate holistic vector. A transitive verb like *chase* requires pairs of words $\langle X, Y \rangle$ for which a holistic vector

$\overrightarrow{X\,chases\,Y}$ can be derived. In general, the combinatorial increase in arguments for function types of higher orders creates a decrease in holistic vector training data that hinders implementation.

Addressing this issue, some studies have looked into ways to simplify representations of complex types. Polajnar, Făgărăşan & Clark (2014) discuss several methods to reduce the complexity of transitive verb representations, which produce results comparable to a full tensor representation. Similar to the *2Mat* model proposed in this study, Paperno, Pham & Beroni (2014) propose the *PLF* model. To represent transitive verbs, both *2Mat* and *PLF* represent a decoupled version of the verb. Instead of representing the verb as a third-order tensor in S ⊗ N ⊗ N, a transitive verb's interaction with its two arguments is modelled as two separate matrices $\mathbf{V^o}$ and $\mathbf{V^s}$. For a sentence like *dog chases cat*, the *2Mat* model defines the output vector $\vec{\mathrm{h}}$ as the concatenation of the verb's interaction with its arguments:

$$\vec{\mathrm{h}} = \left(\overrightarrow{dog} \cdot \overline{chase}^{\mathrm{s}} \parallel \overrightarrow{cat} \cdot \overline{chase}^{\mathrm{o}}\right)$$

where ǁ represents concatenation (adapted from Polajnar et al., 2014, p. 1039). A softmax function is then applied to convert the concatenated products to a vector in the S dimension. The *PLF* model defines the output as the sum of the two dot products and a vector representing the verb:

$$\vec{\mathrm{h}} = \overrightarrow{chase} + \overrightarrow{dog} \cdot \overline{chase}^{\mathrm{s}} + \overrightarrow{cat} \cdot \overline{chase}^{\mathrm{o}}$$

(adapted from Paperno et al., 2014). Gupta, Utt & Padó (2015) point out that the addition of the verb vector creates an inaccuracy in the output, and show that this bias is most efficiently corrected by leaving out the verb vector, resulting in:

$$\vec{\mathrm{h}} = \overrightarrow{dog} \cdot \overline{chase}^{\mathrm{s}} + \overrightarrow{cat} \cdot \overline{chase}^{\mathrm{o}}$$

The decoupling of the verb is based on the intuition that modelling its interaction with the subject and object as independent is an acceptable simplification. Initial testing seems to confirm this notion (Paperno et al., 2014).

To solve questions regarding what models of composition are effective, and what methods of dimensionality reduction are acceptable, most research has focused on specific local structures. Many of the above discussed studies focused on transitive verbs. A more specific structure where the semantics of transitive verbs are studied, are relative clauses containing a single transitive verb (Sadrzadeh, Clark & Coecke, 2013; Rimell et al., 2016a, Moortgat & Wijnholds, 2017).

One advantage of studying relative clauses is that they can have a descriptive function, allowing the semantic representation of an NP containing a relative clause to be compared to that of a noun. For example, Rimell et al. (2016a) focus on $\langle \textit{noun}, \textit{property} \rangle$ pairs like in (4). This example also provides the terminology for such phrases that will be used henceforth.

(4) $\underbrace{\text{expert}}_{\text{term}} : \underbrace{\underbrace{\text{a person}}_{\text{head noun}}\ \overbrace{\text{that}\ \underbrace{\text{a lawyer}}_{\text{argument}}\ \underbrace{\text{consults}}_{\text{verb}}}^{\text{relative clause}}}_{\text{property}}$ (Rimell et al., 2016a, p. 671)

Issues in modelling such clauses partially reflect issues in modelling the verb. Another issue is the representation of the relative pronoun. Its CCG type is (NP\NP)/(S\NP) or (NP\NP)/(S/NP) for subject and object relative clause respectively, requiring a fourth-order tensor. Using a PLF representation of verbs allows the pronoun to be modelled as a third-order tensor (Rimell et al., 2016a), but this is still significant. Additionally, it is debatable whether a distributional representation of a function word like a relative pronoun is valid. Therefore, some models do not involve a trained distributional tensor for the relative pronoun. The semantics of *a person that a laywer consults* are then often assumed to be comparable to that of the declarative sentence *a person consults a lawyer*. For the sake of completeness, the relative pronoun may be represented as a tensor that effectively combines the head noun, argument and verb as if they were combined in a declarative sentence. In the interest of simplicity and implementation, this non-distributional representation of the relative pronoun is typically omitted. Thus, with a PLF representation of a transitive verb, the meaning of the property would be modelled as:

$$\overrightarrow{lawyer} \cdot \overline{consult}^{\mathrm{s}} + \overrightarrow{person} \cdot \overline{consult}^{\mathrm{o}}$$

Regardless of the workings of specific models for relative clauses, any model based in the categorial framework is dependent upon the role assignment of the transitive verb, i.e. whether the clause is object relative or subject relative. This raises the question what the importance of this role assignment is. In a language like Dutch, which uses SOV word order in subclauses, the null argument in relative clauses makes such sentences derivationally ambiguous. As such, a translation of the example in (4) has both a subject-relative and object-relative meaning, as shown in (5).

(5) (a) een persoon die $\underbrace{\text{een advocaat}}_{\text{SBJ}}$ $\underbrace{\emptyset}_{\text{OBJ}}$ raadpleegt [*a person that a lawyer consults*]

(b) een persoon die $\underbrace{\emptyset}_{\text{SBJ}}$ $\underbrace{\text{een advocaat}}_{\text{OBJ}}$ raadpleegt [*a person that consults a lawyer*]

Such sentences may be disambiguated by the verb inflection if the subject and object differ in number or by their agreement in number and grammatical gender with the relative pronoun (*die* for masculine, feminine and plural nouns, *dat* for neuter, singular nouns)[1]. Additionally, semantic requirements of the verb, such as animacy, may dictate semantic role assignment. Nonetheless, a sentence like (5) where such restrictions do not apply, the derivation is ambiguous and selection of the correct role assignment would have to be based on context.

A theoretical discussion of how to model the derivation of relative clauses and link this to a compositional distributional model is provided in Moortgat & Wijnholds (2017). This study points out that typelogical grammars for English, such as CCG, model subject and object relative clauses as having relative pronouns of different types (mentioned above). Yet in Dutch, subject and object relative pronouns should have identical types since the surface form of relative clauses is identical, and the representation of the relative pronoun should imply derivational ambiguity. As mentioned, syntactic representations and the mapping of these representations to tensor representations are taken as given in the current study. Therefore, this section will focus on the distributional representation of Dutch relative clauses proposed by Moortgat & Wijnholds. This proposal features a tensor representation in $N \otimes N \otimes S$ (as is expected without PLF decoupling), and a nondistributional representation of the relative pronoun. The relative pronoun is represented as a third-order tensor in $N \otimes N \otimes N$ with all 0 entries except for a 1 diagonal, together with an all-ones vector in S. The proposed composition of the relative clause (after Moortgat & Wijnholds, p. 8) is given in (6), where $\odot$ represents elementwise multiplication.

(6) (a) persoon die advocaat$_{SBJ}$ raadpleegt [*person that consults lawyer*]

[1] The current study is limited to relative clauses using the relative pronouns *die* and *dat*. These are assumed to have equivalent semantics, since their distinction is based on morphological agreement. The use of interrogative pronouns in similar clauses is not investigated.

$$\overrightarrow{persoon} \odot \left(\left(\sum_{s} \overline{\overline{raadplegen}} \right)^{\mathrm{T}} \overrightarrow{advocaat} \right)$$

(b) persoon die advocaat$_{OBJ}$ raadpleegt [*person that lawyer consults*]

$$\overrightarrow{persoon} \odot \left(\left(\sum_{s} \overline{\overline{raadplegen}} \right) \overrightarrow{advocaat} \right)$$

For the sake of comparing the account by Moortgat & Wijnholds to the PLF model given on page 9, this paper proposes that adapting the model proposed by Moortgat & Wijnholds for a PLF representation of the verb would give the composition in (7).

(7) (a) persoon die advocaat$_{\{\mathrm{SBJ}\}}$ raadpleegt [*person that consults lawyer*]

$$\overrightarrow{persoon} \odot \left(\left(\sum_{s} \overline{raadplegen}^{\mathrm{sbj}} \right) + \left(\left(\sum_{s} \overline{raadplegen}^{\mathrm{obj}} \right) \odot \overrightarrow{advocaat} \right) \right)$$

(b) persoon die advocaat$_{\{\mathrm{OBJ}\}}$ raadpleegt [*person that lawyer consults*]

$$\overrightarrow{persoon} \odot \left(\left(\sum_{s} \overline{raadplegen}^{\mathrm{obj}} \right) + \left(\left(\sum_{s} \overline{raadplegen}^{\mathrm{sbj}} \right) \odot \overrightarrow{advocaat} \right) \right)$$

Note that summing the verb tensor over the S dimension allows for a representation in the N space, while the resulting vector of the PLF method given on page 9 is in S. Additionally, it is worth noting that this model uses elementwise multiplication to combine the head noun with the elements of the relative clause, with the motivation that the semantics of this combination are intersective, and elementwise multiplication is suitable to represent that intersection.

The use of elementwise multiplication in this model raises the question whether semantics that constitute the intersection of sets representing *identity* (e.g. the set denoted by *red car* is the intersection of the set of red things and of cars), would necessarily implicate an intersection of *distributions* (meaning the co-occurrence of *red car* with a word *X* is the intersection of the co-occurrence of *X* with *red* and *X* with *car*). Such issues arise in the selection of nondistributional representations of words, in this case of the relative pronoun. Training a distributional representation of the pronoun avoids such assumptions, but, as mentioned, comes with issues in implementation. More empirical research into relative pronoun representation may provide insight into whether a distributional representation is worth the training required, or if not, which nondistributional representation is most suitable.

The current study will not investigate complex issue, as the main focus is verb representation. Even solely in the interest of verb representation however, Dutch relative clauses are a suitable case study. First, any implementation of the discussed composition models in Dutch in such clauses would require addressing the issue of ambiguous structure. More generally, derivational ambiguity is a hallmark of dealing with natural language data. An implementation of compositional distributional models relying on machine parsing would inevitably contain errors. As such, it is important to verify the stability of compositional models in ambiguous context. Second, investigating the performance of a compositional model in ambiguous clauses may provide insight into the role that syntactic structure plays in such models.

Research Question & Hypothesis

The aim of the current study is to provide insight into the role of syntactic structure, particularly argument structure, in the composition of vectors representing relative clauses. This will be done by investigating how the performance of models of composition for distributional data is affected by the argument structure in relative clauses containing transitive verbs. This issue will be investigated for Dutch, where such sentences are ambiguous.

There is no existing research testing the stability of vector composition under real ambiguity. In general, the relatively good performance of structure-blind models like vector addition and elementwise multiplication (Clark et al., 2016; Rimell et al., 2016a), suggests that a well-performing model is not necessarily affected by syntactic structure.

Testing the performance of compositional distributional models of semantics for Dutch data will also require the training of distributional verb matrices for Dutch (a more detailed account of available resources for Dutch is provided below). As such, the creation of such matrices is a secondary goal of this study.

Method

The aim of this study is to perform an evaluation of models of vector composition in Dutch relative clauses. Since the primary focus will be on the semantic role assignment of the verb, the representation of the relative pronoun is not investigated. As in Rimell et al. (2016a), it is assumed that the relative clause can be treated as a declarative sentence. Given this simplification, three elements are necessary for this evaluation: a dataset of relative clauses, vector representations of individual words and tensor representations of verbs. For the verb tensors, it was decided to use the PLF-style matrices as outlined by Paperno et al. (2014), which not only reduces the complexity of the verb representation (allowing for a matrix rather than a third-order tensor) but also allows the interaction with the head noun and the argument of the relative clause to be evaluated separately. Detailed below is the acquisition of each of these three elements (a relative clause dataset, word vectors and verb matrices), followed by an account of the implementation of composition models on the dataset.[2]

To assemble a dataset of relative clauses, a loose translation was created of the English RELPRON dataset by Rimell, Maillard, Polajnar & Clark (2016b), which is generated from corpus data. Each item in this dataset consist of a term and a property containing a relative clause with a transitive verb, as in (4). For each term, the dataset contains about ten different properties. Properties use the same head noun for all instances of a term. It should be noted that relative clauses are intended to describe some action or relation involving the term, not to give a definition of it.

Items in the dataset were translated when a suitable translation was available, meaning the translation maintained syntactic structure and seemed plausible in use. Items for which no suitable translation existed were left out of the database. In most cases, this was because there was no translation that preserved syntactic structure, or for which a structure-preserving translation resulted in a highly unusual phrase. In some cases, items were excluded because there was no clear translation for the term. If the properties of a single homonym described different senses of it and the term was not ambiguous in Dutch, the sense with the most properties was chosen, while the other was omitted. Lastly, a few cases were left out since they referred to aspects of American culture. As a result, the translated dataset is smaller than the original, containing 728 relative clauses out of the 1086 in the original set. A sample of the translated dataset is provided below:

(8) (a) *OBJ* telescoop: instrument dat astronoom gebruikt/gebruik

[2] The dataset of relative clauses, the code used to create verb matrices and the matrices themselves can all be found at github.com/lukavdplas/dutch-verb-matrices

(b) *OBJ* telescoop: instrument dat sterrenwacht heeft/heb

(c) *OBJ* telescoop: instrument dat waarnemer richt/richt

(d) *SBJ* telescoop: instrument dat spiegel heeft/heb

(e) *SBJ* telescoop: instrument dat lens gebruikt/gebruik

(f) *SBJ* telescoop: instrument dat planeet bespeurt/bespeur

(g) *SBJ* telescoop: instrument dat sterren/ster bekijkt/bekijk

(h) *SBJ* telescoop: instrument dat licht opvangt/vang_op

Regarding the vectors representing nouns, it was decided that these could be imported rather than generated. Two projects have trained distributional word vectors for Dutch. The Polyglot project (Al-Rfou, Perozzi & Skiena, 2013) created language-independent software for training word vectors, and provides trained vectors for a large number of languages, including Dutch. Another project that created word vectors for Dutch is the study by Tulkens, Emmery & Daelemans (2016), which trained vectors based on several Dutch corpora. As described by Tulkens et al., this project used the polyglot vectors as a baseline when testing performance and achieved significantly higher results. Therefore, it was decided to import the vectors trained by Tulkens et al.. Of the models that this study created, it was decided to use those trained on the SoNaR-500 corpus (Oostdijk, Reynaert, Hoste & Schuurman, 2013). Besides performing well in evaluation, this corpus also has the advantage that it makes up the majority of the Lassy Groot corpus, which provides the large collection of dependency trees needed for verb training.

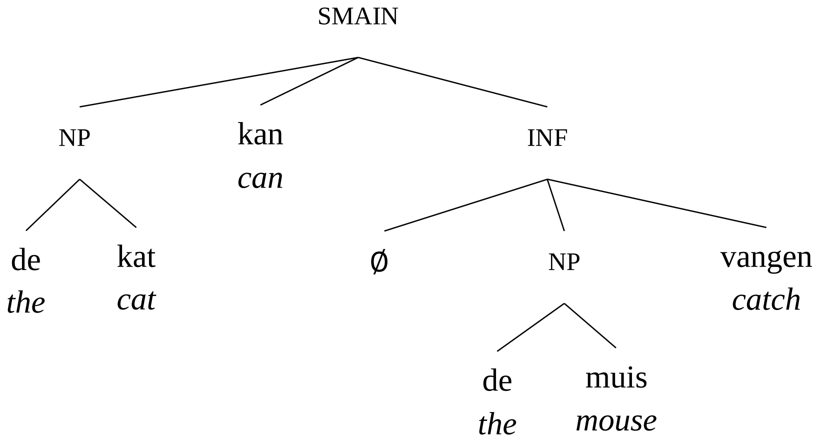
Regarding the verb matrices, these had to be trained since no existing projects provide such matrices for Dutch. Using the translated database, a list of approximately 300 transitive verbs to be trained was assembled[3]. This assembled list of verbs could then be used to collect distribution data and create a holistic vector for each observed verb-argument pair.

Distribution data were retrieved from the Lassy Groot corpus (Van Noord et al., 2013). This large corpus consists of Dutch sentences from a variety of sources, including the SoNaR-500 corpus. Crucially, sentences in the corpus include syntactic trees generated by Alpino (Van Noord, 2006). There is admittedly a decreased accuracy when relying on automatic parsing, but this allows for a corpus of the size needed for the current study.

Since the provided tools to search through the Lassy corpus are fairly limited, data retrieval was done using Python to search through the corpus's XML files. For each sentence in the corpus, any instances of the target verbs were identified. The syntactic structure was then examined to find the arguments of the verb. The object was searched within the domain projected by the verb, i.e. amongst the children of the verb's mother node. Since the subject is often raised out of this domain (e.g. in sentences with auxiliary verbs), the search was extended to the whole sentence if this domain had no overt subject. In such cases, the Lassy data provide an index number for the subject within the domain projected by the verb, and the sentence was searched for a constituent with the same index number and an overt realisation. After selecting the constituents that made up the verb's subject and object, the head of the constituent was selected and the root of this word (as provided by the Lassy corpus) was selected to represent the subject or object. To illustrate this process, the structure of an example sentence as represented in the Lassy corpus is given in (9).

[3] As a minor note, the verb *hebben* (*to have*) would later be left out of training due to its extremely large number of occurrences and broad semantics.

(9)



For this example, the identified verb-argument pairs would be ⟨*vang*, *kat*⟩ and ⟨*vang*, *muis*⟩ for the subject and object respectively. As a brief note, the simplification is made that the sentence above (*the cat can catch the mouse*) is an instance of *cat catches*, i.e. that the modal verb can be ignored. This simplification was made because it increases the amount of retrieved observations, especially for subjects. Since the test sentences only contain main verbs, this is found acceptable. However, a more complete model of composition would feature a distributional representation of the modal verb, which would mean the verb and subject would not interact directly.

Regarding the retrieval of verb instances and their context, two things are worth pointing out. First, the context of the verb was the sentence containing it and did not include the context of the sentence, since the Lassy corpus stores sentences separately and contains no metadata on context. Second, an instance of the verb was stored for either argument that could be identified, but there was no requirement that both needed to be found. This was done largely so as not to limit the amount of data, especially in case not identifying an argument was due to an error in parsing or searching through the tree, rather than the structure of the sentence. This typically resulted in verbs having more samples for objects than for subjects.

Once either argument was identified, a copy of the sentence was exported for each identified argument, with a tag marking the verb, relation and the root of the head of the argument constituent. The assembled training data were then used to assess distribution counts for all verb-argument combinations. As preparation for this, a vocabulary was assembled of the 10.000 most frequent words in the Lassy Groot corpus. Punctuation marks and numerals were stored as '<PUNCT>' and '<NUM>' respectively. After establishing a vocabulary, this was used to count the co-occurrence of verb-argument combinations with vocabulary items. The distribution was counted in a window of 8 words on either side of the verb. Such a large window typically encompassed the whole sentence. Words in the sentence were matched with the vocabulary file. As with assembling the vocabulary, punctuation and numeral items were substituted by general tokens. Words not in the vocabulary were counted as an '<UNK>' token. In addition, '<S>' and '</S>' tokens were added to mark sentence beginnings and endings. This resulted in a matrix of the absolute distribution counts for all verb-argument pairs. Verb-argument pairs with a frequency of 1 were omitted from the data.

To prepare these rudimentary holistic vectors for training verb matrices, PPMI weighting was applied to the holistic vectors using the dissect toolkit (Dinu, Pham & Baroni, 2013). After this, the dimensionality of the holistic vectors was reduced using Singular Value Decomposition from 10.000 to 200 dimensions. Lastly, since holistic vectors would be matched up with argument vectors created by Tulkens et al. (2016), it was necessary to match the argument strings as found in the Lassy corpus with the items in the imported vector set. As mentioned above, verb-argument combinations were grouped based on the root of the argument, not its inflected form. As such, arguments were matched to the vector of their root form, though diminutive suffixes were preserved if a diminutive form was found in the Tulkens et al. data. In addition, numeral strings (e.g. "15") were converted to their literal form (e.g. "vijftien"), since the Tulkens et al. data does not include vectors for numbers. Nonetheless, around 5% of holistic vectors had to be discarded because there was no corresponding vector for their argument.

The imported noun vectors and generated holistic vectors for each verb-relation pair were used to train verb-relation matrices using ridge regression, using scikit-learn (Pedregosa et al., 2011). These matrices will henceforth be called the verb matrices, though they represent a verb and a particular relation. The training is based on the idea that for a holistic vector **h** of length *k* and an argument vector **a** of length *l*, the verb matrix is is an *k* x *l* matrix **V** defined by

$$\mathbf{a} \cdot \mathbf{V} = \mathbf{h}$$

This means that for $1 < \mathrm{i} \leq k$, a feature $\mathrm{h_i}$ of the holistic vector is the product of the argument vector and a single row of the matrix:

$$\sum_{\mathrm{j}=1}^{l} \left(\mathrm{a_j} \cdot \mathrm{V_{ij}}\right) = \mathrm{h_i}$$

Since the calculation of each holistic feature is independent, each row of the verb matrix could be trained independently based on a set of argument vectors and a single dimension of each holistic vector for a verb. As mentioned, training was done using ridge regression, which minimises an error function containing the mean square error of the verb matrix's predictions on the holistic vectors in the train set and the weights of the verb matrix. Pairs of argument vectors and holistic features were weighed according to the logarithm of their frequency, resulting in the following error function for a verb with sample size *N*:

$$\mathrm{E} = \frac{1}{N}\sum_{\mathrm{j}=1}^{N}\left(\log\left(\mathit{freq}(\mathbf{a})\right)\left(\mathrm{h_i} - \sum_{\mathrm{j}=1}^{l}\mathrm{a_j}\cdot\mathrm{V_{ij}}\right)^2\right) + \alpha\left(\sum_{\mathrm{j}=1}^{l}\mathrm{V_{ij}}\right)^2$$

Here, *freq*(**a**) equals the token frequency of a verb-argument pair.

To evaluate the regression, both R^2 and the angle between the predicted and observed holistic vector were calculated. However, the angle was found to have no relation with sample size in initial tests, suggesting it was unsuitable for evaluation. Further development of the training model was based solely on R^2.

Some initial testing was performed to determine certain parameters of the regression. First, it was found that optimal performance was achieved when argument were scaled so as to have zero mean and fixed variance. Second, the value of the sparsity regularisation parameter α was numerically optimised for all verbs simultaneously, as the one resulting the best evaluation for a sample of 54 verbs with a sample size (N) of at least 500. Note that the sample size for the regression was the number of argument types a verb combined with, not the token count. It was found that optimal performance was reached for $\alpha = 50$.

To test the validity of using Ridge regression, training was also performed using LASSO (i.e. L1 regularisation) regression. This was not found to improve performance and will therefore not be reported on below. Unregularised regression was not attempted, since this would be identical to ridge regression with $\alpha = 0$ and low values of α resulted in poor performance. This indicates that regularisation improves performance by suppressing overfitting behaviour on training data.

After creating the verb matrices, a brief analysis of their effectiveness was performed using the translated RELPRON dataset. Five models of composition were implemented, namely the following:

- *Addition:* vector addition of the verb, head noun and argument vector.
- *Varg:* the product of the argument within the clause and the relevant verb-relation matrix.

- *Vhn:* the product of the head noun and the relevant verb-relation matrix.
- *PLF:* the sum of *Varg* and *Vhn*.
- *iPLF:* "inverted" PLF. This model is identical to the *PLF* model, except object-relative clauses are treated as subject-relative and vice versa.

All of these models are also implemented in Rimell et al. (2016a), except for iPLF, which was designed for this study. This model is added to investigate the effect of argument assignment in the verb matrix, by testing the performance if argument structure is incorrectly assigned. The example in (10) shows an example of how PLF and iPLF representations for subject and object relative clauses are composed.

(10) (a) persoon die advocaat$_{\{OBJ\}}$ raadpleegt [*person that consults lawyer*]

PLF: $\overrightarrow{persoon} \cdot \overline{raadplegen}^{s} + \overrightarrow{advocaat} \cdot \overline{raadplegen}^{o}$

iPLF: $\overrightarrow{persoon} \cdot \overline{raadplegen}^{o} + \overrightarrow{advocaat} \cdot \overline{raadplegen}^{s}$

(b) persoon die advocaat$_{\{SBJ\}}$ raadpleegt [*person that lawyer consults*]

PLF: $\overrightarrow{persoon} \cdot \overline{raadplegen}^{o} + \overrightarrow{advocaat} \cdot \overline{raadplegen}^{s}$

iPLF: $\overrightarrow{persoon} \cdot \overline{raadplegen}^{s} + \overrightarrow{advocaat} \cdot \overline{raadplegen}^{o}$

Note that the PLF representation of each reading is equivalent to the iPLF representation of its counterpart. To clarify, the iPLF model is added as a control element, not as a plausible model of composition. The composition models of *full PLF* and *simplified PLF* used by Rimell et al. are not implemented, since these require a distributive tensor for the relative pronoun and an identical N an S space, respectively. The adaptation of the model by Moortgat & Wijnholds illustrated in (7) can be implemented with the given verb representations, but this is left for future research to limit the scope of the current study. The implemented models are illustrated in figure 1.

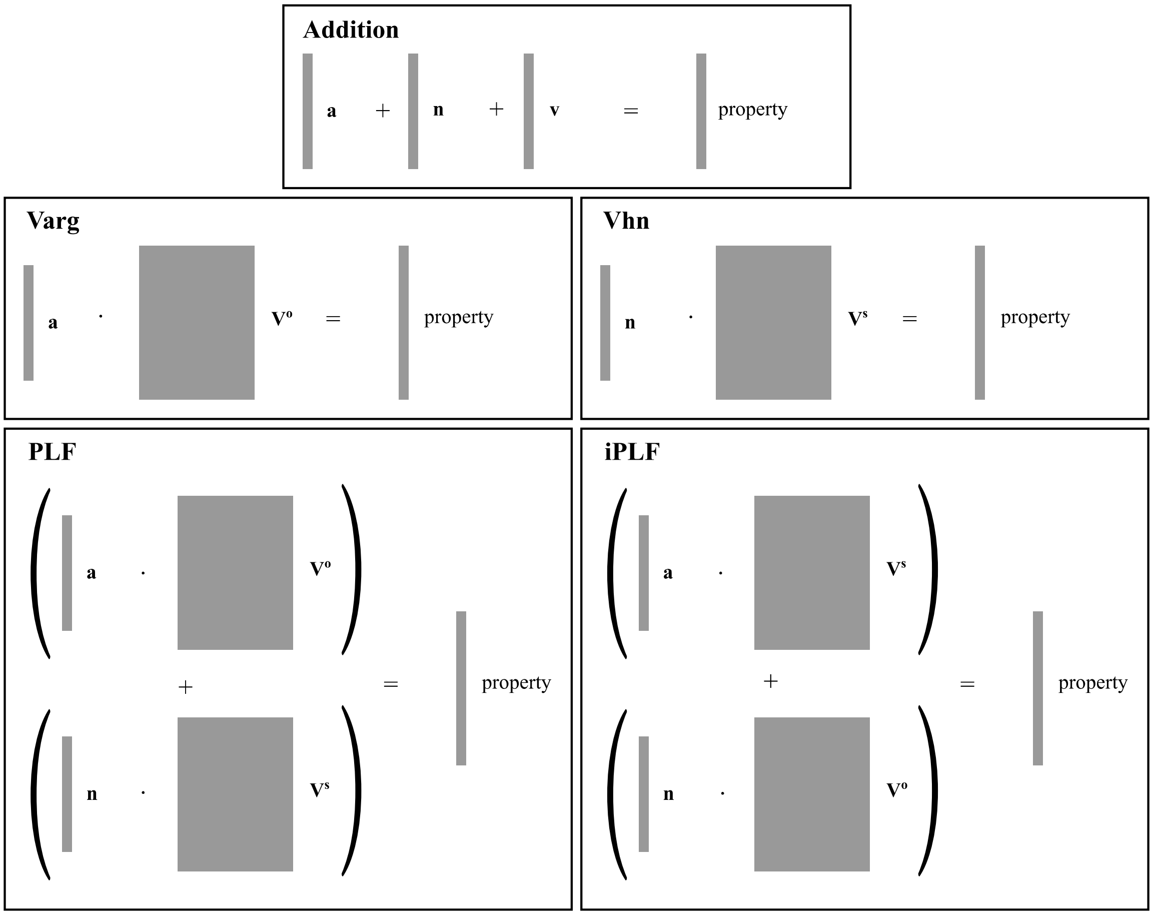


Figure 1: *Illustration of models of composition for relative clauses. To illustrate the use of the verb matrices, this illustration assumes a clause where the head noun is the subject of the verb. In an object relative clause, the choice of verb matrices would be swapped.*

Results

The training of verb matrices was evaluated using different parameter settings, resulting in the method reported on above. These will not be individually reported on. The evaluation of the final selection of parameters is shown in figure 2, which plots the performance of the model against sample size.

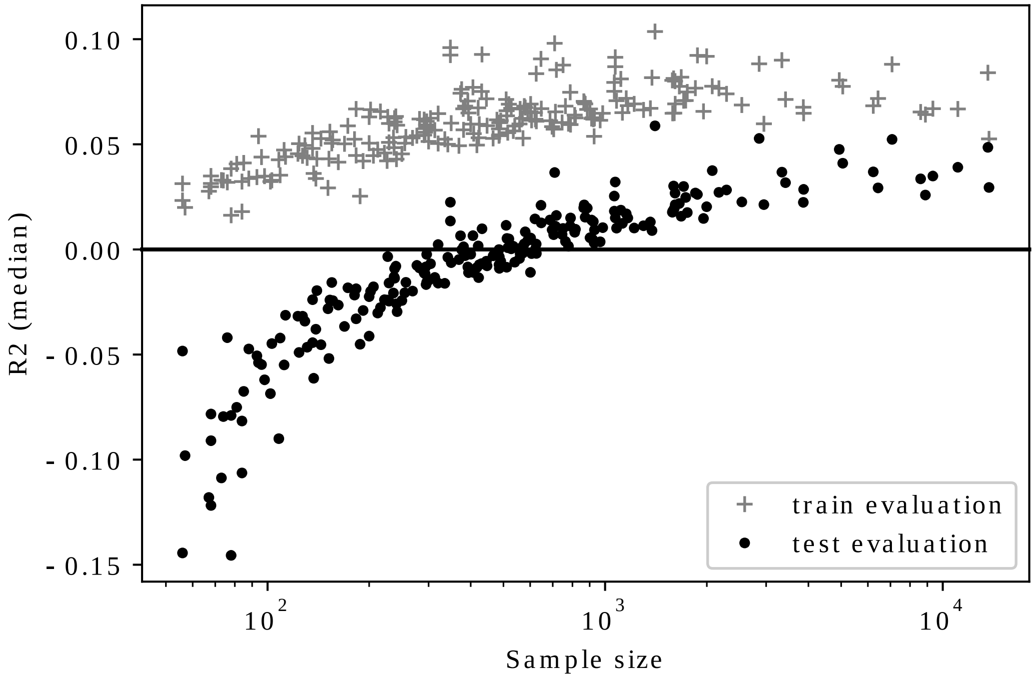


Figure 2: *Evaluation score (median of R^2 for each feature of the holistic vector) on both train and test data set, for each verb matrix. Evaluation score for both sets is plotted against the verb matrices' sample size, which represents the number of type arguments they were combined with. Data represent a sample of 218 verb matrices.*

As can be seen, the R^2 value for test evaluations becomes consistently positive for N > 500, indicating the trained matrix partly predicts the holistic feature variance. The positive effect of verb sample size on test evaluations plateaus for N ≳ 4000. Furthermore, there seems to be a slight positive trend between sample size and train evaluations.

The value for the α parameter in the ridge regression was optimized to maximize R^2 performance. However, initial testing with low α values revealed some different patterns in performance, particularly in relation to sample size, as shown in figure 3.

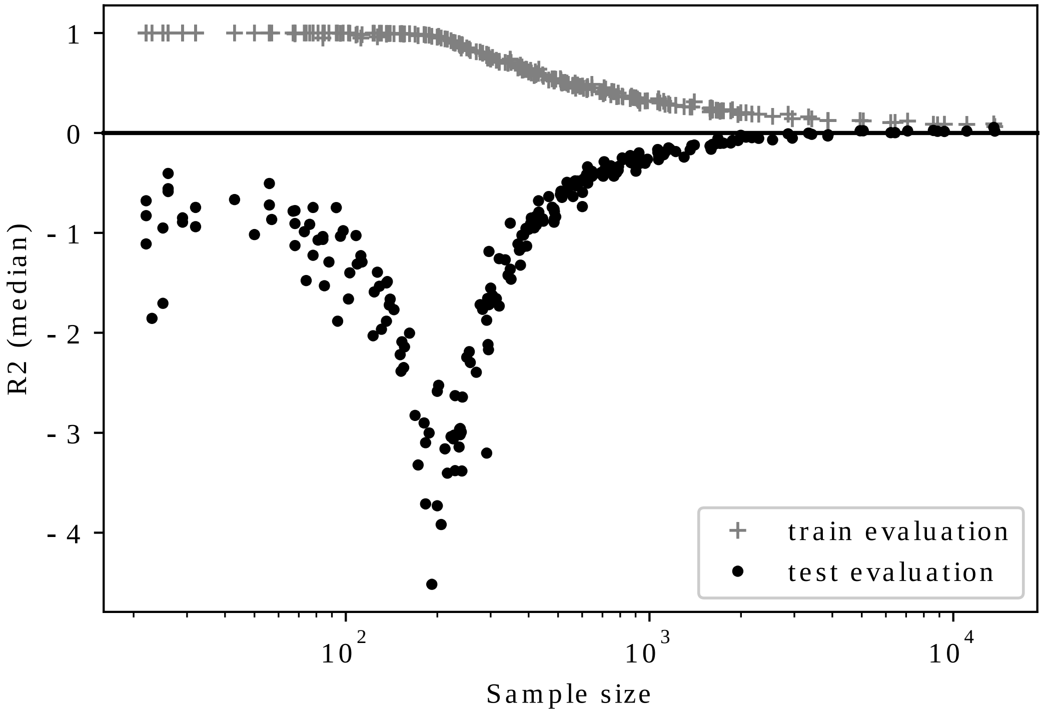


Figure 3: *Evaluation score (median of R^2 for each feature of the holistic vector) on both train and test data set, for each verb matrix. Evaluation score is plotted against the verb matrices' sample size. Performance is evaluated on the same sample of verbs as in figure 2, but regression is performed with α = 0.005.*

As shown, tuning with weak regularization (i.e. low α) results in clear overfitting on training data for low sample sizes, with a near-perfect evaluation score on training data if N < 200[4]. A more interesting trend in the performance on the test set, is that for verbs with a low sample size, evaluation performance degrades as sample sizes increase. This is not an effect of the regression model, but apparently reflects some quality of the data. For N < 200, the severe degree of overfitting means there is little reason for the model's performance on an independent test set to be any reflection of the relation between argument and holistic vectors. However, one explanation for the relatively decent performance in verbs with very low sample sizes (N < 80) is

[4] Since the training set made up 80% of the data, N < 200 means the training set had fewer than 160 samples. Since each holistic feature requires training 160 weights, this mean the model has at least as many degrees of freedom as samples. In short, a near-perfect score is trivial.

that argument test and training vector sets may not be wholly independent if the argument vectors for that verb are similar enough. This would mean that the model seems relatively stable under heavy overfitting, since its test vectors are extremely similar to training vectors.

This hypothesis was tested by comparing the argument vectors for each verb. First, arguments were controlled for their mean and variance (as for the regression). Then, for each verb, the variance between vectors over each feature was calculated. Figure 4 compares the mean of these variances for each verb with its sample size.

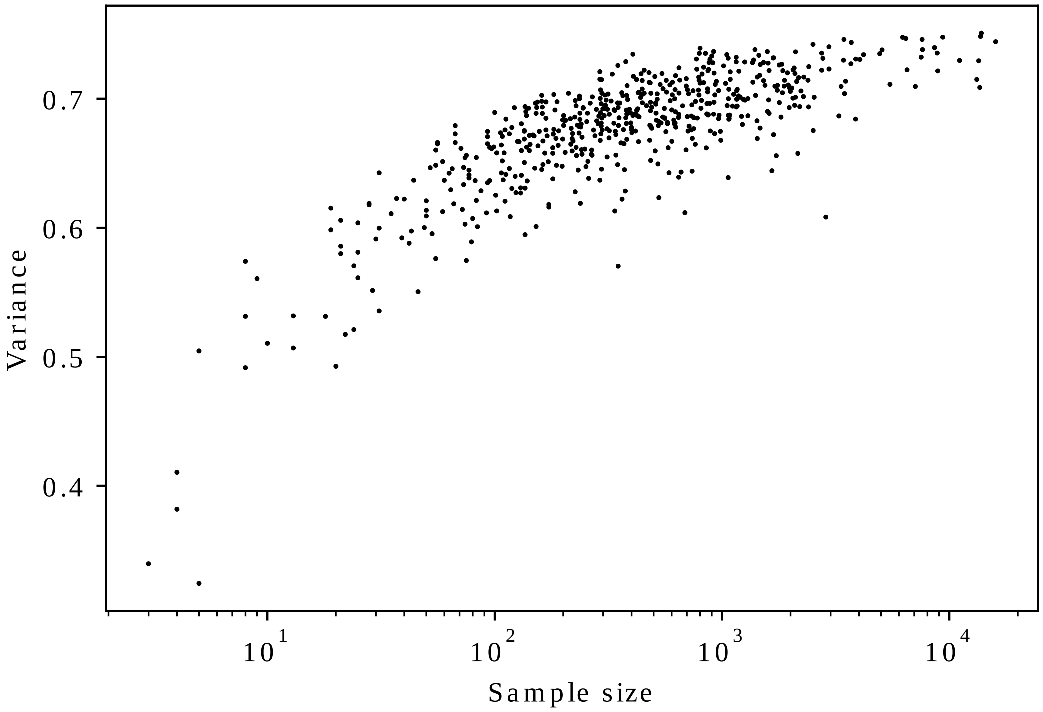


Figure 4: *Mean variance over features of argument vectors plotted against sample size of the corresponding verb.*

As is visible in the figure, there is a positive correlation between argument variance and sample size (ρ = 0,31, p < 0,001). This shows that apparently, the arguments of verbs with fewer argument types are also more similar to each other. This correlation would cause verbs with low sample sizes to be biased towards an increased test performance.

After these evaluations on the training of verbs, the analysis on relative clauses was performed. Because the holistic vectors were trained in their own vector space, it is not possible to compare the product of an argument vector and verb matrix (a vector in the S space) with the noun vector for the term (a vector in the N space) directly. Instead, the analysis is based on the notion that the vectors for two NPs describing the *same* term should be more alike than for NPs describing *different* terms.

For each of the composition models mentioned in the method section, the vector was calculated for each clause in the translated RELPRON database for which the both the $\mathbf{V^o}$ and $\mathbf{V^s}$ matrices had been trained on sample sizes of at least 400 samples. These vectors were then compared by calculating the correlation between each pair. These correlation values were divided in those between vectors describing the same term and vectors describing different terms. The mean of these correlations for each model is summarised in table 1.

As the distribution of both sets of correlations was observed to be roughly symmetrical, an independent t-test was performed to compare the correlation scores in both sets. As can be seen in table 1, the correlation between vectors representing the same term was higher for all composition models. This effect was found to be significant for all models ($p < 0{,}001$ for all cases).

	Same term		Different terms	
	M	SD	M	SD
Addition	0,611	0,185	0,530	0,150
PLF	0,752	0,098	0,685	0,100
Varg	0,674	0,122	0,606	0,121
Vhn	0,734	0,107	0,655	0,104
iPLF	0,749	0,096	0,687	0,095

Table 1: Mean correlation coefficients between vectors describing the same and different terms for each model of composition.

To compare the relative performance of models, a mixed ANOVA was performed to investigate the effect of model and target term equivalence. This revealed a significant interaction effect between model and term equivalence ($F = 7{,}80$, $p < 0.001$). While the correlation scores seem to be mostly predicted by the simple main effect of model ($F = 2{,}60 \cdot 10^3$, $p < 0{,}001$) and term (reported above), there are still minor differences between the degree to which different models distinguish terms.

Interpreting this data, it should be pointed out that the relatively high same term correlation for *Vhn* is trivial, since clauses referring to the same term always used the same head noun. It is, however, noteworthy that the results for PLF and iPLF are only marginally different. This suggest that the verb matrices are not particularly sensitive to argument structure. To further investigate this point of observation, the correlation between the resulting vector for the *PLF* and *iPLF* methods for each property was tested. The results of this comparison are displayed in figure 5, which plots this correlation against the correlation between the subject and object vectors of the verb.

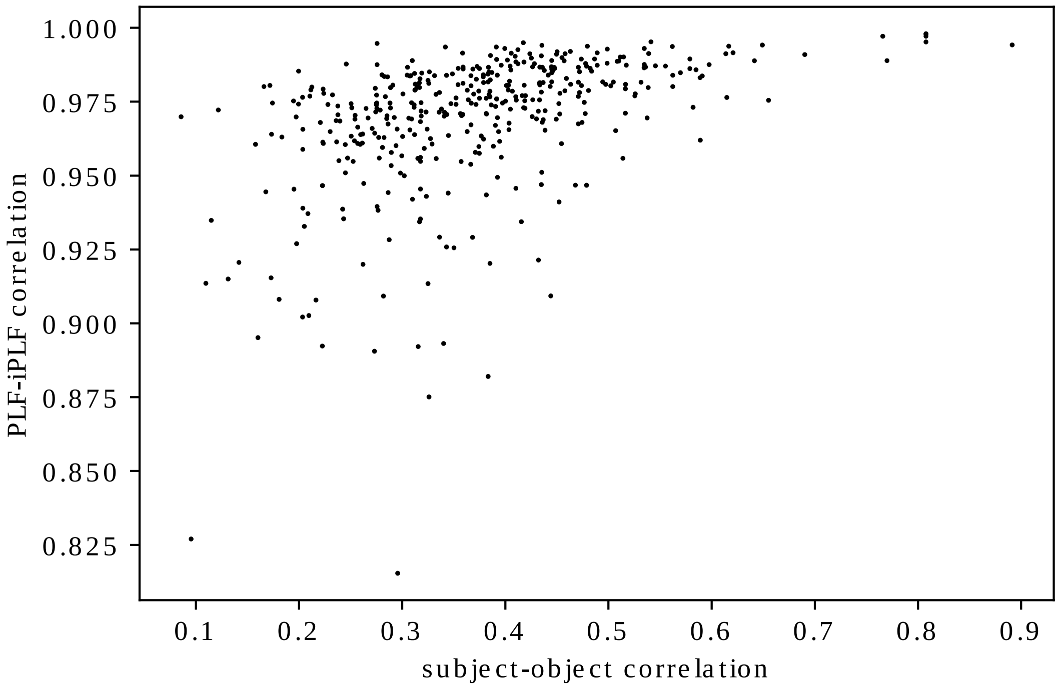


Figure 5: *For each property in the translated RELPRON dataset, the correlation between its vector representation as calculated using the PLF and iPLF methods is plotted against the correlation between the vectors representing the subject and object of the verb.*

As can be seen, the resulting vectors of the PLF and iPLF methods show very high correlations overall. In addition, there is a positive trend between subject-object correlation and PLF-iPLF correlation, which is expected, since the two models receive more similar input if the subject and object are more alike.

Discussion

The evaluation of the verb matrices in the translated RELPRON database shows that composed vectors for properties are more closely related if they resemble the same term, which indicates that the matrices are generally sound. However, such a distinction is also made by the vector addition model, suggesting that the choice of lexical items in the RELPRON properties already creates clusters of terms. The increased similarity between properties of the same term, then, is preserved in the implementation of argument structure, not revealed by it.

Surprisingly, the output of the iPLF model, which uses inverse argument structure, is almost identical to that of the PLF model. This may be an effect of the same reason that the vector addition model reports high similarity between properties describing the same term, namely that these similarities are strongly determined by the set of words in the property, and relations between them play only a minor role. Note that this would nonetheless indicate that a verb matrix applied to incorrectly assigned arguments still returns an adequate representation of the expected context. This is not an inherent property of the matrix. In an example like (11), the training of the subject matrix for *rule* would likely not have contained any instance of *building* as a subject, but, crucially, also no words with similar semantics, since *rule* requires a sentient subject.

(11) *OBJ* klooster: gebouw dat abt leidt [*abbey: building that abbot rules*]

As such, there is little data to infer the map from words like *building* to *building rules*, since the regression was performed on a subset of the N space that did not include *gebouw* or anything close to it.

One reason why verb matrices might give relatively stable performance in incorrect role assignment, is that their output needs to at least somewhat mirror their input, since the distribution of *dog chases cat* will have significant overlap with that of *dog* and *cat*. In addition, the distribution of the sentence would partially reflect the distribution of *chase*, independent of its arguments. On a conceptual level, one might imagine the distribution of *dog chases cat* to be composed as

$$\text{distr}(\textit{dog chases cat}) =$$

$$\alpha \cdot \text{distr}(\textit{dog}) + \beta \cdot \text{distr}(\textit{cat}) + \gamma \cdot \text{distr}(\textit{chase}) + \delta \cdot f\left(\text{distr}(\textit{chase}), \text{distr}(\textit{dog}), \text{distr}(\textit{cat})\right)$$

where the function f handles the interaction between the verb, subject and object, and α, β, γ and δ are real numbers representing the relative weight of each distribution[5]. The vector addition model makes the simplifying assumptions that $\delta = 0$ and $\alpha = \beta = \gamma$. Training verb tensors involves the assumption that the function f is linear and determined by the verb. The PLF model further assumes that f can be decoupled as

$$f_{\textit{chase}}\left(\text{distr}(\textit{dog}), \text{distr}(\textit{cat})\right) = f_{\textit{chase, sbj}}\left(\text{distr}(\textit{dog})\right) + f_{\textit{chase, obj}}\left(\text{distr}(\textit{cat})\right)$$

Since no intercept weights were trained during regression, it is assumed that $\gamma = 0$ (cf. Gupta et al., 2015). With that assumption, the composition above can be reduced to a pair of linear transformations $\mathbf{V^s}$ and $\mathbf{V^o}$ on the $\overrightarrow{dog}$ and $\overrightarrow{cat}$ vectors.

Based on this conceptual breakdown of the phrase distribution, one theory for the similarity of the PLF and iPLF methods would be that the value of α and β are very high, whereas δ is relatively low. In this case, the matrices $\mathbf{V^s}$ and $\mathbf{V^o}$ would closely resemble the identity matrix[6]. As such, the PLF vector for the sentence would resemble the sum of the object and subject vector, and thus have a high correlation with its iPLF counterpart.

Whether the lack of distinction between PLF and iPLF is positive or negative depends on the intended goal of the matrix. In practical implementations of compositional distributional matrices, it may be an advantage that parsing errors do not completely derail the results.

[5] With $\alpha + \beta + \gamma + \delta = 1$.

[6] In the case of the current study, where $N \neq S$, a verb matrix would resemble the linear transformation from N to S that would map a noun to a sentence with the same distribution.

However, a revaluation of the categorial approach as a whole might be required if verb matrices cannot properly distinguish between syntactic structures, since the approach is based on the intuition that incorporating syntactic relations improves the composition of vectors. If the real distribution of a sentence like *dog chases cat* actually features a very low δ value, such relations hardly come into play. This would raise the question whether it is worth the effort to model such syntactic relations in the first place. However, it is prudent to first examine whether this lack of distinction may be a result of issues in training.

Reflecting on the creation of verb matrices, there are several limitations of the current study which are worth pointing out. First, training data was retrieved from a machine-parsed corpus, inevitably resulting in some degree of inaccuracy. While this provided the required size of training data, parsing errors may contribute to verb matrices distinguishing argument structure less clearly. It should be noted, however, that verb data were retrieved from any instances of the verb, not just relative clauses, and argument structure is not typically ambiguous for Dutch sentences. The Alpino parser generally achieves fairly high accuracy, evaluated at around 90% by Van Noord (2006), but there are currently no quantitative data available on the accuracy on verb argument assignment specifically. Given the generally high accuracy of Alpino, machine parsing errors cannot completely explain the similarity between PLF and iPLF results, their correlation being as high as it is. Nonetheless, a more extensive evaluation of parsed training data may provide insight into the role such errors play.

Another limitation is that the minimum frequency to include holistic vectors in training data was low, requiring only two samples. Sharpening this restriction leads to a trade-off between the quality and quantity of holistic vectors, which could most likely have benefited from more testing and optimisation. However, the use of frequency-based weights for holistic vectors in regression has a similar effect, reducing the relative weight of low-quality vectors. Their inclusion was found to have little effect on performance. Nonetheless, further optimisation in the future might still be beneficial.

Regarding the sample size for training verb matrices, figure 2 shows that for verbs in the upper region of sample sizes, performance hardly increases with sample size. For such high-frequency verbs, improvement would have to come from better data selection and training algorithms, but a larger corpus is not necessary. However, many verbs had fewer than 160 training samples, which makes a regression training of 160 weights for the matrix ineffectual. Results show that a sample size of 400 training samples is a valid lower bar, which excludes about half of the verbs. The verbs in the RELPRON corpus are not particularly obscure, so for any application on more than a small selection of high-frequency verbs, it would be necessary to use a larger corpus.

The last limitation that is worth noting is that the N and S vector spaces were not identical. Using separate vector spaces meant that is was possible to assemble only the holistic vectors, while importing noun vectors. However, this means that it is not possible to make a direct comparison between terms and properties in the RELPRON dataset, which limits the analysis.

Besides these limitations which may be revisited in further research, two points of interest for further investigation will be pointed out. First, verb training evaluations suggested a relationship between the quantity and quality of training data that may be worth investigating further. In short, the suggestion is that the data for high-frequency verbs and low-frequency verbs is different in *quality* as well as quantity. Insight into this relationship may explain why in figure 2, the evaluations on training data increases with larger sample sizes, while they would be expected to decrease. This question was not investigated further since the focus was on test performance, but an explanation may provide insight into more general patterns in the training data.

A second pattern that reveals a relationship between the quality and quantity of training data is the evaluated similarity of argument vectors. As discussed, this may explain some relatively high results under heavy overfitting. The increased similarity between the arguments of low-frequency verbs conforms with a casual observation about the training data, namely that the arguments of low-frequency verbs are often thematically linked. For example, out of the 80 objects in the training data for the verb *aanbid* [*revere*], 20 were unambiguously linked to religion. For high-frequency verbs like *give* and *see*, one would expect virtually no thematic link between their arguments. These examples are relatively straightforward, but it seems worthwhile to investigate such effects. For example, an evaluation of trained matrices like the one performed here is by necessity restricted to relatively high-frequency verbs. If the quality of data for low-frequency and high-frequency verbs is significantly different, however, it is questionable whether it is valid to extend the results of these evaluations to low-frequency verbs.

Another point of interest for future investigation is the effect of inverting argument structure for the PLF verb matrices. It may prove interesting to investigate why the iPLF and PLF simulations achieved such similar output. If future investigation into decoupled verb matrices confirms the findings of the current study, that the $\mathbf{V}^{\mathbf{o}}$ and $\mathbf{V}^{s}$ matrices achieve extremely similar results, it raises the question why it is necessary to train two matrices at all. More generally, this may require an evaluation of how the interaction between verbs and arguments is conceptualised. Further investigation is required to understand the interaction between verbs, arguments and their syntactic relationship.

Conclusion

This study investigated the role of argument structure in PLF verb matrices, specifically by testing the performance of such matrices in relative clauses and comparing this with their performance if argument structure was incorrectly assigned. It was found that sentence vectors composed using proper PLF composition and vectors based on incorrect argument assignment returned similar output, and reflected a higher similarity between relative clauses describing the same term than between relative clauses describing different terms. The reasons for the relatively good performance of verb matrices under incorrect argument assignment are suggested as a topic of further study.

Acknowledgements

I would like to thank Michael Moortgat for the introduction to this topic and for his prolonged supervision. Furthermore, I would like to thank Thijs van der Plas for his assistance with the regression and proofreading, and Fons van der Plas for help with software installation.

References

Al-Rfou, R., Perozzi, B. & Skiena, S. (2013). Polyglot: distributed word presentations for multilingual NLP. In: *Proceedings Seventeenth Conference on Computational Natural Language Learning.*

Baroni, M., Bernardi, R. & Zamparelli, R. (2014). Frege in space: a program for compositional distributional semantics. *Linguistic Issues in Language Technologies, 9*(6), 5-110.

Clark, S. (2013). Type-driven syntax and semantics for composing meaning vectors. In: C. Heunen, M. Sadrzadeh, and E. Grefenstette (Eds.), *Quantum Physics and Linguistics: A Compositional, Diagrammatic Discourse*. Oxford: Oxford University Press, 2013.

Clark, S., Rimell, L., Polajnar, T. & Maillard, J. (2016). The categorial framework for compositional distributional semantics. University of Cambridge Computer Laboratory.

Coecke, B., Sadrzadeh, M. & Clark, S. (2010). Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, *36*, 345-384.

Dinu, G., Pham, N.T. & Baroni, M. (2013). Dissect: DIStributional SEmantics Composition Toolkit. In: M. Butt, S. Hussain (Eds.), *Proceedings of the 51st Annual Meeting of the Association of Computational Linguistics: the System Demonstrations* (31-36).

Gupta, A., Utt, J. & Padó, S. (2015). Dissecting the practical lexical function model for compositional distributional semantics. In: M. Palmer, G. Boleda & P. Rosso (Eds.), *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics* (153-158).

Kartsaklis, D., Sadrzadeh, M. & Pulman, S. (2012). A unified sentence space for categorial distributional-compositiona semantics: Theory and experiments. In: M. Kay, C. Boitet (Eds.), *Proceedings of COLING 2012* (549-558).

Maillard, J., Clark, S. & Grefenstette, E. (2014). A type-driven tensor-based semantics for CCG. In: R. Cooper, S. Dobnik, S. Lappin & S. Larsson (Eds.), *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics*, 46-54.

McCarthy, D., Koeling, R., Weeds, J. & Carroll, J. (2004). Finding predominant word senses in untagged text. In: *Proceedings of the 42nd Meeting on Association for Computational Linguistics.* doi:10.3115/1218955.1218991

McDonald, S. & Ramscar, M. (2001). Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*, *23*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani & K.Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems, 26* (3111-3119).

Mitchell, J. & Lapata, M. (2008). Vector-based models of semantic composition. In: *Proceedings of ALC-08: HLT*, 236-244.

Moortgat, M. & Wijnholds, G. (2017). Lexical and derivational meaning in vector-based models of relativisation. *Computing Research Repository.* arXiv:1711.11513

Noord, G. van (2006). At last parsing is now operational. In: P. Mertens, C. Fairon, A. Dister & P. Watrin (Eds.), *Verbum Ex Machina: Actes de la 13^e Conference sur le Traitement Automatique des Langues Naturelles (TALN06)* (20-42). Leuven: Presses Universitaires de Louvain.

Noord, G. van, Bouma, G., Eynde, F. van, Kok, D. de, Linde, J. van der, Schuurman, I., Tjon Kim Sang, E. & Vandeghinste, V. (2013). Large scale syntactic annotation of written Dutch: Lassy. In: P. Spyns & J. Odijk (Eds.), *Essential Speech and Language Technology for Dutch* (147-164). doi:10.1007/978-3-642-30910-6_9

Oostdijk, N., Reynaert, M., Hoste, V. & Schuurman, I. (2013). The construction of a 500-million-word reference corpus of contemporary written Dutch. In : P. Spyns & J. Odijk (Eds.), *Essential Speech and Language Technology for Dutch* (219-247). doi:10.1007/978-3-642-30910-6_13

Paperno, D., Pham, N. T. & Baroni, M. (2014). A practical and linguistically-motivated approach to compositional distributional semantics. In: K. Toutanova, H. Wu (Eds.), *Proceedings of* *the 52nd Annual Meeting of the Association for Computational Linguistics* (90-99). Baltimore: Association for Computational Linguistics.

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. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

Polajnar, T. Făgărăşan, L. & Clark, S. (2014). Reducing dimensions of tensors in type-driven distributional semantics. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (1036-1046).

Rimell, L., Maillard, J., Polajnar, T. & Clark, S. (2016a). RELPRON: a relative clause evaluation set for compositional distributional semantics. *Computational Linguistics*, *42*(4), 661-701.

Rimell, L., Maillard, J., Polajnar, T. & Clark, S. (2016b). Research Data Supporting “RELPRON: A Relative Clause Evaluation Dataset for Compositional Distributional Semantics” [dataset]. doi:10.17863/CAM.298

Sadrzadeh, M., Clark, S. & Coecke, B. (2013). The Frobenius anatomy of word meanings I: subject and object relative pronouns. *Journal of Logic and Computation, 23*(6), 1293-1317. doi:10.1093/logcom/ext044

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, *24*(1), 97-123.

Tulkens, S., Emmery, C. & Daelemans, W. (2016). Evaluating unsupervised Dutch word embeddings as a linguistic resource. In: N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk & S. Piperidis (Eds.), *Proceedings of the Tenth International Conference on Language Resources and*

Evaluation (LREC 2016) (4130-4136). Portorož: European Language Resources Association (ELRA).