
# INFERRING UNOBSERVED CO-OCCURRENCE EVENTS IN ANCHORED PACKED TREES

THOMAS HELMUT KOBER

Submitted for the degree of Doctor of Philosophy
University of Sussex
November 2017

SUPERVISORS:
David Weir
Julie Weeds

## DECLARATION

I hereby declare that this thesis has not been and will not be submitted in whole or in part to another University for the award of any other degree.

Signature:

Thomas Helmut Kober

Dedicated to the loving memory of my mum Astrid (1964 – 2007)

To Sophie and Julia

It was the best of times
It was the worst of times

It was the age of wisdom
It was the age of foolishness

It was the epoch of belief
It was the epoch of incredulity

It was the season of light
It was the season of darkness

It was the spring of hope
It was the winter of despair

– Charles Dickens, *A Tale of Two Cities*

# ACKNOWLEDGEMENTS

First of all, I want to thank my supervisors for their guidance and support throughout my PhD, for always being available for discussing a problem, commenting on yet another draft of a paper at 2am in the morning, 7 minutes before the submission deadline, and for giving me a good (metaphoric) beating whenever necessary.
David Weir for patiently listening through dozens of my half-baked ideas, only to shoot them down with a single question. For teaching me how to bring structure to my writing, how to precisely express an idea, and how to setup a narrative, and for not giving up on me when I referred to APTs as vectors yet again, or when I decided that now is the right time to ignore his suggestion in order to do whatever I thought was best.
Julie Weeds, for almost always being the *good cop* in supervision meetings, for usually being at least one step ahead of me in thinking about the implications and consequences of my own ideas, and for being the only one who generally reads through the whole draft of a paper.
Jeremy Reffin, for being the *chief-whip* when it comes to dissecting empirical work, who has the frightening ability to sniff out dodgy experimental results from 100 miles, and who has the slightly annoying habit of being right more often than would be his turn.
Thank you all very much for your insight and ideas, and for chaperoning me throughout my PhD journey.

It certainly takes a lab to raise a doctor and the TAG lab has always been a welcoming and inspiring environment from the first day I entered as a 2nd year undergraduate intern in July 2013, to the day I *high-fived* everybody before submitting this thesis in November 2017. A big thanks to all, including all de-facto[1], current and former members of the TAG lab, Colin Ashby, Tom Ball, Miro Batchkarov, Daoud Clarke, Justin Crow, Roland Davis, Chris Inskip, Matti Lyra, Hamish Morgan, Jack Pay, Andy Robertson, Aleksandar Savkov, David Sheldrick, David Spence, Joe Taylor, and Simon Wibberley. I'll miss you once I've moved up to Edinburgh, but as Jeremy likes to put it: *you can checkout anytime you like, but you can never leave* ♫.

Outside the lab, I'd like to especially thank Matti, Miro, Phil and Sasho for having spare beds available in Berlin and London on multiple occasions, and for many rants and discussions about politics, python, machine learning and NLP.
Dilyan, for also having a couch available when needed, and foremost

1 I'm looking at you Sasho!

for helping me move my furniture, alongside *a lot* of other useless stuff, from flat to flat and town to town (most recently to Edinburgh). I'm *sorry* that I always seem to move into the third floor flat of a building without elevator, but super-narrow staircases.
Antje, Hannah, Jakob and John, for engaging conversations about culture, media, sports and technology, and for always providing a poor and miserable PhD (and previously BSc) student with a cold drink, a hot meal and a warm bed. Those danish chairs in front of the fireplace are certainly one of the most comfortable places I know.
Finally, my best man Berni and his better half Agnes, for the, by far and wide, finest Maki imaginable, countless board-game and skateboarding sessions (recently much fewer than we would have liked to have . . . ), and for being the godfather of my daughter Sophie. I love you folks ♡!

I am grateful to my dad Heli and his (soon to be) wife Sonja, for their inexhaustible support over the years, and for being able to rely on them as a safety net if things would go horribly wrong (luckily they haven't so far!). Due to not quite being next door neighbours, we haven't seen each other as often as we would have liked to have, but once you retire, you'll have plenty of time to visit us in Edinburgh ☺.
I am furthermore indebted to my in-laws Anna and Fritz, for hosting us in Austria for extended, "temporary" periods, daily bakery services, fresh veg and fruit from their garden, adapting the *cuisine* to support a vegan, despite thinking this whole vegan thing is a bit over the top, infinitely many taxi services to and from various airports and train stations, and an enormous load of other things that I fail to mention here. Thank you! This thesis would not have been possible without your support.

My wife Julia for her unimaginable patience and support over all these years since I decided that what I really needed to do the next in life is to chuck away my job as a software engineer in 2011 in order to go back to university for a degree in AI somewhere in England. And then decide that, well actually, a PhD would be a really decent thing to do next! There are not enough words for describing how grateful I am for everything you have done for me. Thank you — I love you more than you can imagine! And let me just add . . .

*Liebe Julia, mein PhD hat uns die ganzen* 3 *Jahre über sehr viel an Kraft und Nerven gekostet, und es war eine ziemlich ereignisreiche Zeit. In meinem ersten Jahr haben wir geheiratet, im zweiten Jahr wurde unsere Tochter Sophie geboren, und im dritten Jahr hatten wir eine kleine Toddlermaus in der Wohnung herumdüsen die vor allem mischief im Sinn hatte. Danke, Danke, Danke für deine Geduld, dein Verständnis und deine Unterstützung. Ich bin überglücklich dich in meinem Leben zu haben! Ich liebe*

*dich! Ohne dich ist alles doof!*
Never forget that *even in the most uncertain times, which frankly is most of the time, the thing I am most certain of is you* ♡!

My daughter Sophie for teaching me to appreciate the beauty of everyday things and for showing me how very relaxing and entertaining watching horses, geese, ducks, rabbits, cats, dogs, and all sorts of other animals can be. And of course you are right, cats are the best. I also like how you literally rip my thesis drafts into pieces — much like my supervisors do metaphorically. I am unbelievably happy to have you, I love to watch you grow and learn, and when you take your time to explore new things. I admire how quickly you get back up after you fall, and how quietly you sneak out of sight when you're up to something mischievous. I appreciate your honesty, how you cry when you're sad, how you scream and shout when you're angry and how you giggle, laugh and smile when you're happy. I love you more than anything! I hope you'll enjoy Edinburgh and look forward to hearing you talk with a bit of a scottish accent.

Lastly, I want to thank my mum Astrid for everything. I miss you!

## ABSTRACT

Anchored Packed Trees (APTs) are a novel approach to distributional semantics that takes distributional composition to be a process of lexeme contextualisation. A lexeme's meaning, characterised as knowledge concerning co-occurrences involving that lexeme, is represented with a higher-order dependency-typed structure (the APT) where paths associated with higher-order dependencies connect vertices associated with weighted lexeme multisets. The central innovation in the compositional theory is that the APT's type structure enables the precise alignment of the semantic representation of each of the lexemes being composed.

Like other count-based distributional spaces, however, Anchored Packed Trees are prone to considerable data sparsity, caused by not observing all plausible co-occurrences in the given data. This problem is amplified for models like APTs, that take the grammatical type of a co-occurrence into account. This results in a very sparse distributional space, requiring a mechanism for inferring missing knowledge. Most methods face this challenge in ways that render the resulting word representations uninterpretable, with the consequence that distributional composition becomes difficult to model and reason about.

In this thesis, I will present a practical evaluation of the APT theory, including a large-scale hyperparameter sensitivity study and a characterisation of the distributional space that APTs give rise to. Based on the empirical analysis, the impact of the problem of data sparsity is investigated. In order to address the data sparsity challenge and retain the interpretability of the model, I explore an alternative algorithm — *distributional inference* — for improving elementary representations. The algorithm involves explicitly inferring unobserved co-occurrence events by leveraging the distributional neighbourhood of the semantic space. I then leverage the rich type structure in APTs and propose a generalisation of the distributional inference algorithm. I empirically show that distributional inference improves elementary word representations and is especially beneficial when combined with an intersective composition function, which is due to the complementary nature of inference and composition. Lastly, I qualitatively analyse the proposed algorithms in order to characterise the knowledge that they are able to infer, as well as their impact on the distributional APT space.

# PUBLICATIONS

Some of the ideas and figures in this thesis have appeared previously in the following publications:

D. Weir, J. Weeds, J. Reffin, T. Kober (2016). Aligning packed dependency trees: a theory of composition for distributional semantics. In *Computational Linguistics, special issue on Formal Distributional Semantics*, 42(4):727–761, December 2016.

T. Kober, J. Weeds, J. Reffin, D. Weir (2016). Improving sparse word representations with distributional inference for semantic composition. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1691–1702, Austin, Texas, November 2016.

T. Kober, J. Weeds, J. Wilkie, J. Reffin, D. Weir (2017). One representation per word - does it make sense for composition?. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 79–90, Valencia, Spain, April 2017.

J. Weeds, T. Kober, J. Reffin, D. Weir (2017). When a red herring is not a red herring: Using compositional methods to detect non-compositional phrases. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 529–534, Valencia, Spain, April 2017.

T. Kober, J. Weeds, J. Reffin, D. Weir (2017). Improving semantic composition with offset inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 433–440, Vancouver, BC, July 2017.

# CONTENTS

# TERMINOLOGY & NOTATION

| Notation | Explanation |
|---|---|
| $\alpha, \beta$ | Scalar variables |
| $x, y$ | Vectors |
| $\mathbf{A}, \mathbf{B}$ | Matrices |
| $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}$ | Tensors (of order 3 or higher) |
| $V$ | Set |
| $x_i$ | $i^{\text{th}}$ element of the vector $x$ |
| $\mathbf{M}_{i,j}$ | element in the $i^{\text{th}}$ row and $j^{\text{th}}$ column of the matrix $M$ |
| $\lvert V \rvert, \lvert x \rvert$ | size of the set $V$ and the vector $x$, respectively |
| $x + y, \mathbf{A} + \mathbf{B}$ | elementwise addition between the vectors $x$ and $y$, and the matrices $\mathbf{A}$ and $\mathbf{B}$, respectively |
| $x \odot y, \mathbf{A} \odot \mathbf{B}$ | elementwise (a.k.a. Hadamard) product between vectors $x$ and $y$, and matrices $\mathbf{A}$ and $\mathbf{B}$, respectively |
| $x \otimes y, \mathbf{A} \otimes \mathbf{B}$ | outer (a.k.a. Kronecker) product between vectors $x$ and $y$, and matrices $\mathbf{A}$ and $\mathbf{B}$, respectively |
| $x \oplus y, \mathbf{A} \oplus \mathbf{B}$ | concatenation of vectors $x$ and $y$, and matrices $\mathbf{A}$ and $\mathbf{B}$, respectively |
| $x \cdot y, \mathbf{A} \cdot x, \mathbf{B}\, y$ | dot product between vectors $x$ and $y$, and matrix vector product between $\mathbf{A}$ and $x$, and $\mathbf{B}$ and $y$, respectively |
| $x \circledast y$ | arbitrary function to combine vectors $x$ and $y$ |
| $f, g$ | functions, precise semantics will be c explained in the text |
| $\langle w, r, w' \rangle, \langle w, c \rangle$ | co-occurrence event between the word-type-word triple $w, r, w'$, and the |

| | |
|---|---|
| | word-context tuple $w, c$, respectively. The semantics of $w'$ and $c$ are such that $w'$ always refers to a single lexeme whereas $c$ can refer to any kind of context such as single lexemes, phrases, sentences, paragraphs or documents. The two will be used interchangably unless a distinction is important for the understanding of a statement. |
| $\langle w, r, w' \rangle, \langle w, \tau, w' \rangle$ | co-occurrence between words $w$ and $w'$ with type $r$ or $\tau$, respectively. $r$ is used to refer to any kind of relation that can be modelled between the two lexemes, whereas $\tau$ specifically refers to a grammatical relation between $w$ and $w'$. Both will be used interchangably unless the context requires a distinction. |
| $\#\langle w, r, w' \rangle$ | frequency of the co-occurrence event between words $w$ and $w'$, and relation $r$ |
| `dobj`, `nsubj` | dependency relations (denoted by their universal dependency label) |
| $\overline{\texttt{amod}}$, $\overline{\texttt{nsubjpass}}$ | inverse dependency relations |
| `amod.nsubj` | higher-order dependency relation (separated by a dot) |
| $\langle \overline{\texttt{amod}}, \textit{seagull} \rangle$, $\overline{\texttt{amod}}\texttt{:}\textit{seagull}$ | typed distributional features in tuple form (former) and key form (latter) |
| $\mathbf{A}$ | elementary Apt representation (same notation as for matrices, it will be made clear in the text whether an Apt or a matrix is being referred to) |
| $\textit{white}^{\texttt{amod}}$ | offset Apt, represents the noun offset view of the adjective *white* |
| $\vec{\mathbf{A}}$ | vectorised Apt |
| lexeme | A word type (e.g. in opposition to a word occurrence or token) and that uniquely identifies an entry in the Apt lexicon or any |

| | |
|---|---|
| | other distributional space |
| higher-order feature | Referring to the case where the dependency path in a co-occurrence between two words includes two or more dependency relations. |
| higher-order APT | APT lexicon that includes higher-order features. |
| type | Unless stated otherwise in the text, this refers to the label of a dependency relation. |
| path of order $n$ | $n$ refers to the length of a dependency path in a co-occurrence event between two words. |
| packing | Process that merges words with identical paths into the same APT node. |

# 1

# INTRODUCTION

*Representing Natural Language Meaning*

Representing natural language meaning has been a long standing open research problem in natural language processing (NLP) as it requires an answer to two central questions from philosophy and artificial intelligence: *what is meaning?* and *how can it be represented?*

One answer to these questions is based on formal semantics theory, which operates on the sentence level, and defines meaning as the truth value of a given sentence. This notion of meaning arguably dates back to the work of Frege (1892) and treats the derivation of the denotation of a sentence as a logical inference problem. A logic formalism is used to describe and represent the meaning of a sentence. The focus of formal semantics has predominantly been on modelling the role of closed class words such as determiners. Content — or open class — words such as adjectives, nouns and verbs, on the other hand, have received comparatively less attention and are still frequently treated as "unanaly[s]ed primitives" (Partee, 2016).

A contrasting answer to the question of meaning and its representation is based on lexical semantics research, which unlike formal semantics approaches, focuses on the meaning of content words. One popular approach to represent the meaning of individual lexemes is based on the distributional hypothesis, attributed to Harris (1954) and Firth (1957). The distributional hypothesis states that "difference of meaning correlates with difference of distribution" (Harris, 1954), which means that two lexemes have similar meaning if their associated co-occurrence patterns are similar.

*Distributional Semantics*

This hypothesis has been adopted by distributional semantics research, where meaning representations are derived from the co-occurrence statistics of words in a large corpus of text. Individual lexemes are typically represented by a high-dimensional vector that records the co-occurrences with the contexts in which the lexemes occur. The vector representation of any lexeme in isolation is of limited utility in determining the meaning of a word. However, the *distribu-*

*tional similarity* to all other lexemes in the same space gives rise to a *continuous model of meaning*. This enables the precise quantification of the representation for the lexeme *dog* being more similar to the representations for *cat* and *pet*, than to the vector representations for the lexemes *car* and *mug*.

While the technique of representing lexical items on the basis of their co-occurrence statistics has been very successful for individual lexemes, the same method quickly becomes infeasible for longer *n*-grams due to data sparsity. For example, in a cleaned[1] October 2013 Wikipedia dump, the lexeme *cat* occurs almost 25k times, whereas the bigram *black cat* has only 680 occurrences, and the trigram *big black cat* occurs only once. Collecting co-occurrence statistics therefore is already problematic for less frequent two-word combinations, and becomes infeasible beyond the bigram level. Furthermore, it is practically impossible to observe all plausible word combinations of any length in any text corpus[2].

### *Distributional Composition*

One approach that has been proposed, and has attracted a substantial amount of interest in the NLP research community in recent years, is to leverage distributional representations of individual lexemes, and *compose* them to create representations for longer phrases. This idea has frequently been motivated by the Fregean *principle of compositionality* (Frege, 1884), which states that the meaning of the whole is a function of the meaning of its parts and the way in which these are combined.

For distributional word representations, which are most commonly modelled as vectors, this means that some arithmetic function is applied to two or more vectors to derive a meaning representation for a phrase. In one of its simplest instantiations, this means retrieving the respective vectors for *white* and *clothes* from the vector space, and composing them by pointwise addition in order to create a representation for the phrase *white clothes*. However it is unclear what the vector for the phrase *white clothes* actually represents. Pointwise addition of the two constituent representations suggests that *white clothes* are as similar to something *white* as they are to *clothes*, which arguably is not the case. The adjective *white* is only modifying the noun *clothes*, hence

1 Articles with fewer than 20 pageviews on a given day have been removed, see Wilson (2015).

2 The bigram *angry cat*, for example, never occurs in the Wikipedia corpus mentioned above, however would represent a perfectly plausible phrase.

the representation for the composed phrase *white clothes* should still predominantly be governed by *clothes*. It therefore remains an open research question of how exactly the constituents in the phrase interact, what distributional features are being shared among them in the given context, and how such a phrase should be represented.

An early approach that addresses the problem of modelling the meaning of a phrase with distributional word representations has been put forward by Erk and Padó (2008). They propose that the key component to expressing the denotation of a phrase is a *contextualisation* mechanism that extracts the meaning of each individual lexeme given the context of the phrase it occurs in. This means that only features of *white* that are relevant to *clothes*, and features of *clothes* that are relevant to *white*, contribute to the meaning of the phrase *white clothes*. In Weir et al. (2016) we expanded on that idea by formalising distributional composition, which is interpreted as a process of lexeme contextualisation, into two steps. These involve the correct *alignment* and subsequent *integration* of the distributional knowledge of every lexeme in a phrase. The result of that process is that every lexeme's meaning reflects its bespoke use in the phrase it occurs in.

Distributional composition therefore acts as a mechanism of extracting the appropriate meaning of the lexemes involved in a phrase from each others contexts, and integrating them into a representation that models the semantics of the *whole* by leveraging the correct meaning of its *parts*. For the lexemes in a phrase "aligning" means "agreeing" on a set of distributional features that will be shared in the composed representation.

For example when considering the phrase *white clothes* once again, the meaning of *white* in the context of *clothes* and the meaning of *clothes* in the context of *white* should be reflected in the resulting composed representation. The contextualisation mechanism expresses the fact that the meaning of *white* in the phrase *white clothes* is different from the meaning of *white* in *white noise*. The alignment step achieves that the two representations agree on features that take *white things* as their direct object, as well as other verbs that *white clothes* are the subject of. While the alignment is driven by syntax as *white* is used as a modifier but needs to be expressed in terms of a noun — i.e. a *white thing* — the consequences of the alignment are semantic. This is because the representation of the adjective *white* changes from an uncontextualised modifier to a noun that expresses the semantics of a "thing that can be *white*". The subsequent composition operation in-

tegrates the agreed set of features into a single shared representation that constitutes the phrase *white clothes*.

Interpreting distributional composition as a two-step process of alignment and integration is crucial to appropriately modelling the dynamics between the various *meaning potentials* of distributional word representations, that are activated in a context-dependent manner. This interpretation of composition parallels the argument of Hanks (2000), who asserted that a lexeme only has a concrete meaning within a context. Outside of any context, a lexeme expresses a number of different meaning potentials, which are activated when put into context.

*Limitations of Vector-Based Representations for Distributional Composition*

Representing lexemes as vectors in a metric space is not suitable for modelling distributional composition as the process of *alignment* and *integration* outlined above. This is because a vector-based representation is static, and implicitly assumes that all word representations are correctly aligned *a priori*, outside of any context. Consequently, distributional composition is modelled as a *post-hoc* operation on top of a static vector space that results in a limited capacity to contextualise a lexeme.

Another problem of using a vector space as the basic data structure is that standard composition functions, such as pointwise addition or multiplication, are commutative. Remedying this shortcoming has frequently involved the use of some form of weighted addition (Mitchell and Lapata, 2008; Baroni and Zamparelli, 2010; Mitchell and Lapata, 2010; Zanzotto et al., 2010; Grefenstette et al., 2011; Grefenstette and Sadrzadeh, 2011a,b), with the weights frequently modelled by a neural network (Socher et al., 2011; Tsubaki et al., 2013; Kalchbrenner et al., 2014; Kim, 2014; Bowman et al., 2016; Hill et al., 2016). The process of agreeing on a set of shared features between two lexemes is "outsourced" to a set of learnable parameters that are determined in a task-specific manner. However, this in turn introduces the reliance on the availability of a sufficiently large amount of training data in order to learn the parameters of the composition function[3].

3 As Mou et al. (2016) have shown, transferring a neural network based composition function from one task to another is difficult, often brittle, and frequently results in relatively poor performance on the task the neural network has been transferred to.

*Anchored Packed Trees for Distributional Composition*

These above mentioned limitations for modelling the meaning of a phrase in an unsupervised and task-independent manner motivate the need for a novel way of representing distributional knowledge. This data structure should be able to encode the distributional semantics of a given lexeme out of context, as well as supporting a mechanism for appropriate alignment when used in context. The form of representation furthermore needs to support a way of integrating the aligned representations into a single unified structure that is capable of modelling longer phrases and sentences in the same space as individual lexemes[4].

One such proposal is Anchored Packed Trees (Weir et al., 2016) (APTs), which represent the distributional semantic space as a graph, where edges associated with dependency relations connect vertices associated with weighted lexeme multisets. By modelling the grammatical relation of a lexeme, APTs support a mechanism for aligning the semantics of the lexemes involved in a phrase. This precise alignment of the semantic representation of each of the lexemes being composed represents the central innovation underlying the compositional theory.

In Anchored Packed Trees, distributional composition is the core concept and the structure of the elementary distributional word representations is a direct consequence of the way that composition has been defined. By focusing on an effective mechanism for aligning the semantics of the lexemes in a phrase, the problem of commutativity in distributional composition can be avoided, while retaining the benefit of being an unsupervised model that does not require a labelled dataset to learn a task-specific composition function.

APTs leverage the syntactic structure of the text to model composition as the contextualisation of all the lexemes involved in a phrase. This results in composed phrasal representations that are distinct to the given context. The composition function acts as a mechanism to integrate the aligned meanings of the lexemes involved in the phrase into a single unified representation.

One central contribution of this thesis is a practical evaluation of the APT theory that analyses the performance of elementary and composed APT representations. The thesis also contributes a large-scale hyperparameter sensitivity analysis, and provides recommendations

4 While this thesis is only concerned with individual lexemes and short phrases, the theory of APTs goes well beyond that as outlined in Weir et al. (2016).

for deriving a set of robust parameters when using APTs in practice. In the following, the thesis presents a characterisation of the distributional space that the novel APT data structure gives rise to.

*The Problem of Data Sparsity in Distributional Representations*

The empirical evaluation of the APT theory, however, highlights a central practical difficulty when working with APTs. The typed nature of elementary representations, and the syntactically driven definition of distributional composition comes at the cost of increasing the negative effect of *data sparsity* — the problem of not observing all plausible co-occurrences between any two lexemes. The reason for the amplified negative effect of sparsity is the way the elementary APT representations are structured. Co-occurrences are not only collected between two lexemes, but between word-type-word triples, where the type is the grammatical relation holding between the two words.

For example, due to modelling the grammatical relation of a co-occurrence, APTs model the distinction between a noun being used as the object or subject in a phrase, or even more fine-grained distinctions such as the use of a subject in an active or passive voice construction. These distinctions would not be present in a distributional semantic model that neglects the type information and only models the co-occurrence of word-word tuples.

The problem of data sparsity extends to distributional composition, where a representation for a phrase needs to be built from two incomplete elementary structures. This subsequently results in very poor composed representations that do not adequately capture the semantics of a phrase. Therefore data sparsity is a central challenge for extending the continuous model of meaning from the lexical to the phrasal level.

The issue of data sparsity has been a long standing research problem in distributional semantics research, and has motivated the use of various techniques for dimensionality reduction such as Singular-Value Decomposition or Non-negative Matrix Factorisation, as well as the use of low-dimensional neural word embeddings, which aim to improve the generalisation capabilities of a model.

When analysing the sparsity problem in a distributional semantic space, however, it can be useful to distinguish between two different kinds of sparsity: *model sparsity* and *data sparsity*. A model can

be sparse *by design* as the consequence of explicitly representing the co-occurrence space in a distributional semantic model. This has the advantage of deriving very expressive and discriminative[5] representations, and has the effect of being straightforwardly interpretable. A disadvantage of model sparsity is that such representations tend to be unwieldy to use in downstream tasks due to their high dimensionality. This is one of the reasons why low-dimensional and dense neural word embeddings have become such a popular resource for a large range of different NLP tasks (Turian et al., 2010; Collobert et al., 2011).

Data sparsity on the other hand is a consequence of not observing all plausible co-occurrence events in any given text corpus. This is an effect of the richness and variety of natural language itself. There is almost always more than one way to express an idea or thought, hence the distributional features of a concept can be scattered across many different lexemes when collecting the co-occurrence information from a text collection. However, they might have occurred for only one lexeme without changing the meaning of the thought.

For example, the lexemes *bike* and *bicycle* are equally plausible in numerous and possibly disjoint contexts. This has the consequence of not observing all plausible co-occurrences for *either* lexeme when building distributional word representations, resulting in incomplete representations for *both* terms. Concretely, this can result in a situation where only *bicycles* are observed as being *old*, and only *bikes* are observed as being *stolen*, according to the representations of *bicycle* and *bike* in a given Apt space derived from the British National Corpus (Burnard, 2007).

The following two concrete examples illustrate how the problem of data sparsity manifests itself differently in dense and sparse distributional semantic models. In a dense model, such as neural word embeddings, sparse data frequently leads to increasingly high similarity scores for completely unrelated lexemes, or low similarity scores for related ones. For example in a 100-dimensional `word2vec` space derived from the British National Corpus, the nearest neighbours of the lexeme *dongle*[6] include misspelled terms such as *workflo*, or completely unrelated terms such as *pizza*. Data sparsity therefore causes semantic inconsistency in the distributional neighbourhood, that fails

5 Discriminative in the sense of discriminating between many different contexts.

6 A *dongle* is a small device that can be connected to a computer, such as a wireless broadband stick. See https://en.oxforddictionaries.com/definition/dongle for a definition. The lexeme *dongle* occurs only 10 times in the British National Corpus, so is relatively infrequent.

to capture interesting linguistic patterns, such as clusters of topically related words, that have been observed for neural word embeddings (Levy and Goldberg, 2014a; Pennington et al., 2014).

In a sparse model, on the other hand, less data increases the sparsity of already sparse representations, i.e. there are hardly any observed co-occurrences for a given word. This leads to very little contextual overlap and therefore very low similarity scores between any two lexemes. For example in an example APT space derived from the British National Corpus, the lexeme *dongle* has only 2 non-zero co-occurrence features (out of ≈800k dimensions), leading to essentially no feature overlap with any other lexeme. The low number of non-zero features is a consequence of applying a lexical association function such as PPMI (Church and Hanks, 1989). This results in effectively treating most other lexemes with which *dongle* co-occurs as "chance encounters" which are assigned negative PMI scores that are subsequently filtered by the PPMI threshold.

Sparse models typically suffer more from data sparsity because of explicitly representing all observed contextual dimensions instead of embedding the contexts into a latent space as in dense models. This is a direct consequence of the "curse of dimensionality" as there are fewer observations for each dimension in the available data.

Data sparsity also represents a major challenge for composing distributional word representations as it is difficult to accurately capture the meaning of a phrase when it has to be built from impoverished elementary representations. For APTS specifically, data sparsity impacts the alignment process during which a set of shared features is negotiated between the lexemes in a phrase. The problem stems from the "uncertainty of the 0" — the issue whether a co-occurrence event has not been observed because it is *actually* implausible, or because it just has not been seen in the given source corpus, while being perfectly plausible.

In APTS the contextualisation process is responsible for narrowing down the meaning of the lexemes in the current phrasal context, which means filtering out distributional features that do not fit in the given context. However, by not observing all plausible co-occurrences on the lexical level, the process of contextualisation will inevitably filter out too many features. Therefore, distributional composition in Anchored Packed Trees requires a supporting mechanism that is able to expand the current state of knowledge in order to better model the

semantics of a composed phrase.

Sparse and dense distributional semantic models are at somewhat opposite ends of the representation spectrum. Sparse models are highly transparent in terms of the knowledge they encode, however they tend to discriminate too many contexts, thereby reinforcing the data sparsity problem. Dense models, conversely, rely on an opaque optimisation process to merge contextual dimensions without much chance of encoding prior knowledge as to what should and should not be merged.

The second central contribution of this thesis is the proposal of an unsupervised algorithm that infers plausible knowledge by leveraging the distributional neighbourhood. The algorithm strikes a middle ground between the two extremes by offering the possibility to control what contextual dimensions can be combined, while retaining the discriminative and transparent nature of sparse count-based distributional semantic models. The proposed distributional inference algorithm can furthermore be seen as a first step towards overcoming the "uncertainty of the 0" problem, as any contextual dimension that remains unobserved (i.e. is still 0) after inference is much more likely to be actually implausible than before. This is because a particular context might not have been observed with any of the distributional neighbours of a lexeme either. Therefore, it can be disregarded with a much higher level of confidence than before. While the thesis is focused on enriching representations within the Anchored Packed Trees framework, the proposed algorithm is applicable to other count-based sparse distributional models as we have successfully demonstrated in Kober et al. (2016). An analysis of the impact of distributional inference on other count-based models is out of scope of this thesis.

*Mitigating the Sparsity Problem*

The necessity of an alternative approach to mitigating the data sparsity problem in APTs stems from the fact that typical approaches, such as reducing the dimensionality of the elementary representations, are not feasible for APTs. The reason is that the composition mechanism in the APT framework relies on explicit knowledge concerning the structure of the representations. This means that individual context dimensions need to be interpretable. However, while

applying dimensionality reduction techniques smoothes the elementary word representations and thereby — arguably[7] — improves its generalisation capabilities, it furthermore renders the individual contextual dimensions opaque due to embedding them into a latent space. This is not compatible with composition in Anchored Packed Trees.

The proposed algorithm leverages the distributional neighbourhood for enriching the representations with additional information. Explicitly inferring co-occurrence features from a set of nearest neighbours creates the potential for effectively transferring and sharing knowledge between distributionally similar lexemes. For example if the lexeme *bicycle* is among the nearest neighbours of the lexeme *bike*[8], it is possible to inject some of the knowledge of *bicycle* into the representation for *bike*, and *vice versa*, which originally might not have been observed in the corpus. This enriches the representation of both lexemes with additional information that has already been obtained elsewhere, and provides an effective intrinsic mechanism for addressing the data sparsity issue. Hence, the proposed algorithm provides an effective solution to the problem described earlier, where *bikes* have not been observed as *old*, and *bicycles* have never been *stolen*, due to inferring the corresponding missing distributional features for *bike* and *bicycle*, respectively.

The rich type structure in Apts allows the distributional inference process to go beyond the surface lexeme level and leverage the neighbourhood of so-called "offset" Apt representations. Offsetting is a fundamental part of the composition process in Apts and is responsible for aligning the semantic representations of two lexemes accordingly. An offset Apt representation therefore reflects the contextualised view of a given lexeme in a phrase. For example, offsetting is the process that expresses the semantics of the adjective *white* in terms of a noun — a "thing that can be *white*" — when used in the context of *white clothes*. This structure might well be similar to representations for "things that can be *blue*" or "things that can be *dark*", thus enabling higher-order inferences of knowledge for any given Apt representation. This form of knowledge expansion goes beyond what has been observed at the "surface-form" co-occurrence level and provides a

7 See Caron (2001); Bullinaria and Levy (2012); Lapesa and Evert (2014); Levy et al. (2015); Sahlgren and Lenci (2016); Lapesa and Evert (2017) for a detailed discussion.

8 Indeed in a tested Apt space derived from the BNC, *bicycle* is the nearest neighbour of *bike*.

mechanism for inferring knowledge from a more abstract conceptual level.

The inference process based on offset representations effectively leverages the rich type structure and exploits the higher-order semantics of a given lexeme that is encoded by an APT. This leads to a generalisation of the distributional inference procedure where elementary APT representations can be enriched with additional knowledge outside of any context, as well as in a contextualised manner. Offset views and offset inference will be discussed in detail in Chapters 4 and 5, respectively.

Furthermore, by generalising the inference process to offset representations, an important connection between distributional inference and distributional composition is uncovered as both are realised by the same operation. Their complementary use results in an effective mechanism for *co-occurrence embellishment* through distributional inference, and *co-occurrence filtering* through distributional composition, when composing a phrase as we have shown in Kober et al. (2017a) and as will also be outlined in Chapter 5.

## 1.1 CONTRIBUTIONS OF THIS THESIS

The contributions of this thesis are a combination of practical analyses based on empirical data, together with advancements on APT theory. On the practical side, this thesis presents an empirical evaluation of the APT proposal together with recommendations for a set of favourable hyperparameter settings on the basis of an extensive hyperparameter sensitivity analysis. Furthermore, the thesis provides a characterisation of the distributional semantics of elementary, offset and composed APT representations. This includes an analysis of their respective distributional neighbourhoods, as well as an assessment of the performance of APTs on a number of widely used word similarity datasets and a short phrase composition benchmark.

On the theoretical side, the thesis adapts the algorithm introduced by Essen and Steinbiss (1992) and Dagan et al. (1993) for smoothing language models, to the use with a distributional semantic model such as APTs. Subsequently, the algorithm is generalised to distributional offset inference by leveraging the rich type structure inherent in APT representations in order to address the data sparsity issue more effectively. The thesis shows that distributional inference and offset inference significantly improve the performance of well-tuned

APT models on a range of popular word similarity tasks as well as a short phrase composition benchmark dataset. Alongside the quantitative improvements, the proposed algorithm is studied qualitatively by analysing what kind of knowledge it is able to infer, and in how far the additionally inferred knowledge changes the characteristics of the distributional representations.

Lastly, the thesis highlights that the generalisation of the distributional inference algorithm to offset inference uncovers a latent relation between distributional composition and distributional inference. The thesis shows that both operations are concerned with inferring plausible co-occurrence counts between a given set of representations, and are realised by the same operation. This thesis furthermore proposes the use of distributional inference in a complementary way to distributional composition. The inference operation can be used as a process of *co-occurrence embellishment* to expand the current state of knowledge, and composition can be used as a process of *co-occurrence filtering* for filtering implausible co-occurrences.

## 1.2 STRUCTURE OF THIS THESIS

The thesis is structured as follows: Chapter 2 reviews related work on distributional semantics (§ 2.1), compositional distributional semantics (§ 2.2), modelling word meaning in context (§ 2.3), and inferring unobserved events (§ 2.4). Chapter 3 introduces the theory behind Anchored Packed Trees, and explains how elementary APT representations are created (§ 3.1) and how distributional composition is modelled (§ 3.2), as well as contributing a discussion on how APTS compare to previously proposed models (§ 3.3).

Chapter 4 introduces the preprocessing pipeline, datasets and evaluation methodology (§ 4.1) used throughout this thesis and contributes a practical evaluation of the APT theory (§ 4.2), on the basis of a large-scale hyperparameter sensitivity study that derives a set of parameter recommendations for using APTS in practice. In addition, Chapter 4 characterises the distributional semantics of APT representations (§ 4.3). Chapter 5 analyses the problem of data sparsity (§ 5.1), followed by the proposal of the distributional inference algorithm alongside a quantitative study to measure its performance as well as a qualitative analysis of its impact on the distributional semantic APT space (§ 5.2). The standard distributional inference algorithm is generalised to offset inference (§ 5.3) and its qualitative and quant-

itative impact on elementary and composed APT representations is measured. Lastly, the thesis explores the inherent relation between distributional inference and distributional composition (§ 5.4) and recommends the combined use of an inference mechanism alongside an intersective composition function. Finally, Chapter 6 highlights the main contributions (§ 6.1), summarises the thesis (§ 6.2) and outlines potential directions for future work (§ 6.3).

# 2

# RELATED WORK

The following chapter reviews 4 different NLP research branches, each with their own associated body of work. Section 2.1 discusses distributional representations of individual lexemes and distinguishes "bag-of-words" approaches (§ 2.1.1) from syntactically aware word representations (§ 2.1.2). This distinction is particularly important for Section 2.2, Compositional Distributional Semantics, and Section 2.3, Contextualisation —- Modelling Word Meaning in Context, as syntactically aware distributional word representations generally lead to more expressive composition functions and more flexibility to model the meaning of a word in context.

Section 2.2 is concerned with the different ways to compose individual distributional semantic word representations into longer phrases. One major difference between different approaches is whether the complexity of the model is encoded in the composition function or in the word representations. Approaches based on formal semantics principles, on the other hand, typically encode the majority of the complexity in tensor-based word representations and subsequently apply comparatively simple composition functions. Distributional composition with vector-based word representations is discussed in Section 2.2.1, and semantic composition on the basis of formal semantics theory is discussed in Section 2.2.2.

Section 2.3 reviews approaches to determining the meaning of a word in context. While being closely related to distributional composition, it has usually been treated as a task in its own right. Different approaches for contextualisation include multi-prototype and exemplar-based models (§ 2.3.1), contextualisation based on modelling selectional preferences (§ 2.3.2), and expressing the meaning of a word in context via latent sense modelling approaches (§ 2.3.3).

Lastly, Section 2.4 discusses ways to infer missing information in a model by leveraging its distributional neighbourhood. It addresses the problem of not observing all possible co-occurrences in a given corpus of text, which applies to all distributional semantic approaches that obtain distributional representations from data. Approaches that specifically aim to leverage an inference mechanism for distributional composition are discussed in Section 2.4.1.

## 2.1 DISTRIBUTIONAL SEMANTICS

Distributional semantics is an approach to lexical semantics, concerned with the study of the meaning of words on the basis of their co-occurrence statistics in language. It is based on the distributional hypothesis (Harris, 1954; Firth, 1957), stating that words appearing in similar contexts tend to have similar meaning. The idea can further be traced back to Saussurean structural linguistics (de Saussure, 1916), via the work of Firth (1935), with a clear distinction between a signifier (the string of characters making up the word *seagull*) and the signified (the distribution of co-occurrences for the concept denoted by the string *seagull*), which together represent the Saussurean "sign".

Early approaches to apply the distributional hypothesis in a computational semantic space model include Dale and Dale (1965) for clustering[1] of key words in information retrieval, Harper (1961, 1965) for determining the distributional similarity of Russian nouns, Spärck-Jones (1964) for thesaurus construction, and Rubenstein and Goodenough (1965) for empirically analysing the correspondence between similarity in meaning and similarity of contextual distributions.

Representing words as distributions of the contexts in which they occur started gaining more popularity through the works of Church and Hanks (1989) for lexicography, Deerwester et al. (1990) for information retrieval, and Hindle (1990) for classifying nouns into distributionally similar sets. A formalisation of a semantic space model, or vector space model, was proposed by Lowe (2001), defining it as a quadruple $\langle A, B, S, M \rangle$. In this definition $B$ is the set of basis context elements that forms the dimensionality of the space. These are typically word types, word lemmas or whole documents, but can also be $\langle r, w' \rangle$ relation-word tuples. $A$ is a lexical association function, converting raw co-occurrence frequencies of the elements in $B$ to weights or scores, in order to make the word representations more robust to frequency effects. $S$ denotes the similarity measure for the semantic space, expressing the similarity between the two context distributions of two lexemes as a real-valued number. Typical measures include metrics such as cosine or information theoretic measures such as proposed by Lin (1998); Lee (1999); Weeds and Weir (2003). Lastly, $M$ is a transformation function, mapping the semantic space onto another one, for example by applying a dimensionality reduction function.

1 Though the authors refer to their technique as "clumping" the methodology would be referred to as "clustering" in more modern terminology.

A substantial amount of research in the distributional semantics literature is concerned with improving and analysing any of the items in the quadruple $\langle A, B, S, M \rangle$, that has been defined by Lowe (2001). For example, one choice of the basis context elements $B$, together with their precise parameterisation such as the size of the associated context window, might be better at uncovering certain linguistic regularities than another choice. For example it is well established that very narrow context windows promote a semantic space governed by hypernymy and co-hyponymy, whereas wide context windows uncover more topical relations such as meronymy and semantic relatedness (Peirsman, 2008; Baroni and Lenci, 2011; Levy and Goldberg, 2014a). A large body of research is concerned with exploring and evaluating the vast hyperparameter space of distributional models (Bullinaria and Levy, 2007, 2012; Lapesa and Evert, 2014, 2017; Kiela and Clark, 2014; Sahlgren, 2006; Sahlgren and Lenci, 2016). Furthermore two recent surveys by Turney and Pantel (2010) and Erk (2012) provide a broad overview of current research topics.

Two major axes along which distributional models can be categorised are whether the word representations are *typed* or *untyped*, and whether contexts are *counted* or *predicted*. In the typed case, the grammatical relation holding between a co-occurrence is recorded, whereas untyped models follow the bag-of-words paradigm. The second categorisation is whether co-occurrences are explicitly counted, or whether they are predicted (Baroni et al., 2014) as in neural network models. In the following I will make a primary distinction along the *typed-untyped* axis and distinguish count-based and predict-based models for typed and untyped approaches individually.

### 2.1.1 *Untyped Distributional Semantic Models*

In an untyped count-based distributional semantic model (DSM), each lexeme is represented as a distribution over contextual items as observed in a large body of text. The definition of what exactly forms a context is a hyperparameter, however the most frequently adopted notion is that of a symmetric spatial window, with typical sizes between 1-10, around a given target word. Alternative interpretations of context involve full sentences, paragraphs or whole documents.

The basic methodology of creating an untyped distributional vector space model is very simple. In a count-based model, the co-

occurrences of each target word $w$ with every context item $c$ in a given corpus is counted and stored in a co-occurrence matrix.

More formally, a co-occurrence matrix $\mathbf{M} \in \mathbb{R}^{|V| \times |V|}$, with rows indexed by $i$, representing target words $w$, and columns, indexed by $j$ denoting context items $c$, where $|V|$ is the size of the vocabulary $V$, is populated with scores denoting the association between $w_i$ and $c_j$ as defined in Equation 2.1

$$\mathbf{M}_{i,j} = f(w_i, c_j) \tag{2.1}$$

where $\mathbf{M}_{i,j}$ denotes the co-occurrence matrix cell in the $i^{\text{th}}$ row and $j^{\text{th}}$ column, and $f$ is a lexical association function such as PMI and PPMI (Church and Hanks, 1989; Dagan et al., 1993), $t$-test (Curran, 2004), tf-idf (Spärck-Jones, 1972), or a function returning the raw co-occurrence frequency between $w_i$ and $c_j$ (Deerwester et al., 1990; Lund and Burgess, 1996), among many others proposed in the distributional semantics literature.

Figure 2.1 shows an example sentence with a symmetric context window of size 1, where the co-occurrences of *big* and *landed* for the current target word *seagull* are recorded. The increment $\alpha$ can be either a constant such as 1, or, for example, depend on the distance $d$ from the target word such as $\frac{1}{d}$, which would damp the contribution of words further away from the target lexeme (Sahlgren, 2006; Levy et al., 2015). The context window slides over the whole cor-

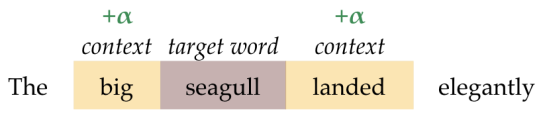


Figure 2.1: Sliding window of a DSM.

pus, resulting in a large and sparse co-occurrence matrix. In order to decrease the high dimensionality of the semantic space, a number of techniques have been proposed to embed the words into a latent, low-dimensional and dense feature space. These include methods such as Singular-Value Decomposition (Deerwester et al., 1990), Non-negative Matrix Factorisation (Lee and Seung, 2001), random indexing (Sahlgren and Karlgren, 2002), or the use of a neural network for matrix factorisation (Pennington et al., 2014), among others.

More recently, a class of models based on predicting co-occurrences rather than explicitly counting them, has gained a considerable amount of momentum in the NLP research community. Instead of modelling the context dimensions as word types, these methods dir-

ectly yield low-dimensional and dense representations, embedding words in a latent and distributed feature space (Bengio et al., 2003; Morin and Bengio, 2005; Collobert and Weston, 2008; Mnih and Hinton, 2009; Collobert et al., 2011; Mnih and Kavukcuoglu, 2013).

The perhaps most popular[2] method for creating low-dimensional word embeddings is `word2vec` by Mikolov et al. (2013). The proposed method comes in two variants, *Continuous Bag-of-Words* (CBOW), which aims to predict the target word from its surrounding context, and *Skip-Gram* (SG), which aims to predict the surrounding context from a given target word. Figure 2.2 illustrates the difference between the two algorithm variants. Where the objective of CBOW is to predict *seagull* from observing the surrounding context words *big* and *landed*, SG works the other way round, by aiming to predict *big* and *landed*, from its observation of *seagull*.

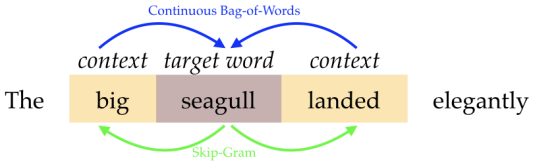


Figure 2.2: `word2vec` — Skip-Gram and Continuous Bag-of-Words models.

On the technical side, training proceeds in an online fashion, predicting a word either via a hierarchical softmax formulation (Morin and Bengio, 2005; Mnih and Hinton, 2009), or by a variant of noise contrastive estimation (Gutmann and Hyvärinen, 2012), called negative sampling[3], which aims to maximise the dot product of two similar words and minimise the dot product of two dissimilar words. A dissimilar word is drawn from a noise distribution (Mikolov et al., 2013).

While predict-based models do not operate on an explicit co-occurrence matrix, Levy and Goldberg (2014b) showed that SG with negative sampling implicitly factorises a word-context co-occurrence matrix, where the entries approximate their PPMI value, shifted by a global constant which is related to the number of negative samples drawn from the noise distribution.

Untyped DSMs — both count-based and predict-based models — make the simplifying assumption that word meaning can be represen-

2 Judging from the number of citations, which is approaching 4000 at the time of this writing.

3 The major difference is that noise contrastive estimation results in a probability distribution, whereas negative sampling does not. Negative sampling utilises unnormalised scores and is therefore computationally more efficient due to not normalising the distribution of scores to form a probability distribution.

ted on the basis of the spatial proximity between lexemes, effectively ignoring the grammatical structure of the given text.

### 2.1.2 *Typed Distributional Semantic Models*

In a typed distributional semantic model the lexical association between a target word and its context is extended to include the relation between the two co-occurring items. Furthermore, a contextual item is generally assumed to be another word. The predominant approach for modelling the relation is to use the grammatical structure of the text, such as from a dependency grammar (Tesniére, 1959; Hays, 1964). However, more complex definitions of a relation are possible (Baroni and Lenci, 2010). A typed co-occurrence between a given target word and a context is therefore a triple of the form $\langle w, r, w' \rangle$ and is defined as

$$\mathbf{M}_{i,j} = f(w_i, r, w'_j) \tag{2.2}$$

where $\mathbf{M}_{i,j}$, $w_i$, $w'_j$, and $f$ are defined as in Equation 2.1, and $r$ is the relation between the co-occurring word and its context. Typically, $r$ is mapped onto the context dimension which is thereby forming a $\langle r, w \rangle$ tuple, to keep the semantic space as a matrix.

Early work on typed DSMs focused on modelling nouns in simple predicate-argument structures in noun phrases such as adjective-noun and noun-noun compounds, and verb phrases such as subject-verb and verb-object constructs for noun classification (Hindle, 1990) and query expanson (Grefenstette, 1992). Due to the availability of improved broad coverage syntactic parsers, Lin (1998) was able to build a robust typed distributional space beyond just nouns, also modelling content words with other parts of speech such as adjectives, adverbs and verbs. The benefit of a more fine-grained typed distributional space has also been shown by Lin and Pantel (2001) for automatically acquiring inference rules from text, Curran and Moens (2002) for automatic thesaurus extraction, Rothenhäusler and Schütze (2009) for concept clustering, and Weeds et al. (2014a) and Roller and Erk (2016) for hypernymy detection.

A framework for creating typed distributional semantic spaces was introduced by Padó and Lapata (2007) who interpreted and formalised the type of a co-occurrence as the dependency relation between a target word $w$ and a context word $w'$. Padó and Lapata (2007) extend

prior work to include inverse and higher-order co-occurrences in a dependency tree.

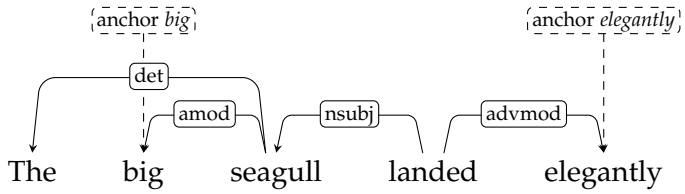


Figure 2.3: Anchored dependency-typed DSM.

An example of an inverse relation in the dependency tree in Figure 2.3 would be the reverse dependency arc `nsubj` from the noun *seagull* to the verb *landed*. A higher-order feature would be recording the co-occurrence between the noun *seagull* and the adverbial modifier *elegantly*, which is essentially expressing that "*seagulls* can do something *elegantly*".

Padó and Lapata (2007) also introduce the notion of an *anchor*, which marks the relative starting point for all paths anchored at a given lexeme. Figure 2.3 shows two possible anchor positions for the given dependency parsed sentence[4]. For example, placing the anchor at the adjective *big* gives rise to the typed co-occurrence, $\langle big, \overline{\texttt{amod}}.\overline{\texttt{nsubj}}, landed \rangle$, between *big* and *landed* by following the inverse `amod` and inverse `nsubj` relations. However, if the anchor is placed at *elegantly*, then its co-occurrence with *landed* would be via the inverse `advmod` relation, giving rise to the co-occurrence event $\langle elegantly, \overline{\texttt{advmod}}, landed \rangle$.

Instead of recording the co-occurrences of spatially adjacent words, a typed distributional semantic space stores the co-occurrences of syntactically related[5] lexemes. Figure 2.4 shows the co-occurring contexts for the lexeme *seagull* when taking first-order direct and inverse dependency relations into account. The number of co-occurring contexts is governed by the syntactic structure of a given sentence instead of a fixed (or sampled as in the case of `word2vec`) context window. For example, the lexeme *big* only co-occurs with the noun it modifies (*seagull*) in the typed model, whereas it would also co-occur with the article *The* in an untyped model.

Padó and Lapata (2007) note that their dependency-typed distributional semantic space is very sparse, and therefore notably remove the path information from the context definition, effectively turning

4 The anchor can be placed at any lexeme in the sentence.

5 Referring to a syntactic relation between two or more lexemes in a parse tree. For example a transitive verb is syntactically related to its subject and object.

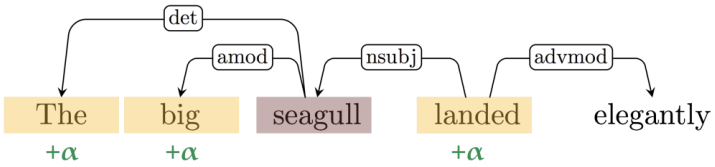


Figure 2.4: Co-occurrences in a count-based dependency-typed DSM.

the co-occurrence triple $\langle w, r, w' \rangle$ into a $\langle w, w' \rangle$ tuple. However, while the type information has been removed from the word representations, the co-occurrence information is still based on the syntactic structure of the text. Therefore the model remains distinct from an untyped model. In their experiments, Padó and Lapata (2007) have shown that their typed vector space model achieves strong performance on a range of tasks such as synonymy detection or word-sense disambiguation.

Baroni and Lenci (2010) generalised typed distributional semantic models beyond modelling a relation on the basis of the dependency path between two lexemes to include more complex lexico-syntactic links such as encoding past participles and auxiliaries as part of the relation $r$ in a $\langle w, r, w' \rangle$ triple. Their model, called *Distributional Memory*, is represented as a 3rd order tensor and is explicitly modelling the type information $r$, of a given co-occurrence of the form $\langle w, r, w' \rangle$, in its own dimension. This formulation enables Baroni and Lenci (2010) to instantiate a number of different 2-dimensional semantic spaces through matricisation[6] of the 3rd tensor along a given dimension.

For example, by encoding the relation in the contextual dimension, Baroni and Lenci (2010) derive a semantic space defined by $\langle w, r\, c \rangle$, which represents the most commonly used way of modelling the $\langle r, c \rangle$ tuple in a typed distributional semantic space. Other spaces include $\langle w\, r, c \rangle$, $\langle w\, c, r \rangle$ and $\langle r, w\, c \rangle$. Baroni and Lenci (2010) have shown the flexibility and utility of the different semantic spaces that their model gives rise to for a range of tasks, including word similarity, relation classification and modelling selectional preferences of verbs, among others.

All of the typed DSMs discussed above follow a count-based paradigm that models the underlying co-occurrence space explicitly. Levy and Goldberg (2014a) on the other hand, take a con-

6 Matricisation refers to the process of converting a tensor into a matrix along a specified dimension (Kolda and Bader, 2009).

text predict-based approach, and introduced a generalisation of the `word2vec` Skip-Gram model. Their proposed method takes a parsed corpus as input and, for a given target word, predicts its surrounding context words together with the dependency relations connecting them to the target word. Figure 2.5 illustrates how the model would predict the typed co-occurrences ⟨`det`, *The*⟩, ⟨`amod`, *big*⟩, and ⟨$\overline{\texttt{nsubj}}$, *landed*⟩, for the target word *seagull*.

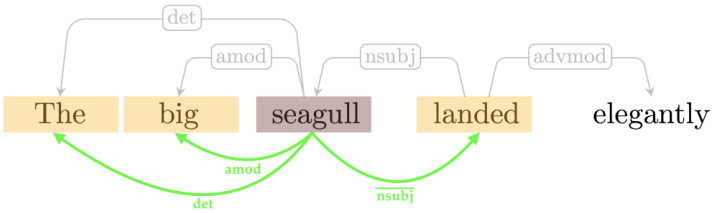


Figure 2.5: Co-occurrences in a predict-based dependency-typed DSM. The model's predicted dependency arcs are shown below the text.

Levy and Goldberg (2014a) conducted a qualitative analysis of the distributional neighbourhood of their model and found that their dependency-based word embeddings give rise to a functionally governed neighbourhood, dominated by co-hyponymy. This contrasts the neighbourhood of a standard window-based Skip-Gram model where the neighbourhood is governed by topical relatedness. This effect was confirmed in a ranking experiment where word pairs exhibiting a functional similarity are consistently ranked higher than word pairs being topically connected, in comparison to two standard `word2vec` Skip-Gram models with varying window sizes. These findings agree with the results of earlier studies by Peirsman (2008) and Baroni and Lenci (2011), who reached the same conclusion when investigating typed and untyped count-based VSMs.

### 2.1.3 *Untyped vs. Typed Distributional Semantic Models*

It is not possible to assert superiority to one kind of model over the other, *a priori* in the absence of a task. Untyped distributional semantic models generally give rise to a neighbourhood governed by meronymy and topical relatedness, making them a better fit for tasks such as topic classification (Kiela et al., 2015). In contrast, the distributional neighbourhood in typed models is governed by hypernymy and co-hyponymy, which was found to work better for thesaurus construction (Grefenstette, 1992; Lin, 1998; Curran, 2004; Kiela et al., 2015).

One advantage of untyped distributional models is their independence from the availability of broad-coverage syntactic parsers, making them more easily applicable to low-resource languages. On the other hand, the modelling of typed co-occurrences results in more expressivity for tasks involving distributional composition, and generally more flexibility for tasks requiring a fine-grained understanding of the semantics of a phrase or sentence (Padó and Lapata, 2007; Lewis and Steedman, 2013).

## 2.2 COMPOSITIONAL DISTRIBUTIONAL SEMANTICS

The major goal of compositional distributional semantics research is to create meaningful representation of longer phrases from lexical distributional representations by means of a composition function. Distributional word representations are well known for effectively encoding a large amount of linguistic knowledge, and providing a continuous model of meaning represented in a distributional space. Composition is therefore a way to extend the continuous model of meaning from the lexical to the phrasal level. Furthermore, distributional word representations are scalable to very large corpora, are language-agnostic, and can be obtained by unsupervised algorithms in an offline manner. Leveraging such existing resources to model longer units of text has therefore vast practical potential.

The idea of distributional composition is frequently motivated by the *principle of compositionality* of Frege (1884), which states that the meaning of a complex expression is a function of its parts together with the way in which they are combined. Frege's principle has been a hotly debated subject in the linguistics community (Pelletier, 1994a,b), and has furthermore given rise to a large body of research in the NLP community, resulting in many practical advancements.

An early theoretical framework for modelling composition in a distributional semantic space has been proposed by Mitchell and Lapata (2008)[7] who translated the Fregean principle of compositionality to distributional semantics:

$$z = f(x, y, R, K) \tag{2.3}$$

7 However their approach is pre-dated by the *predication* algorithm of Kintsch (2001) and the work of Widdows (2008), but neither introduces a general framework for modelling distributional composition. Furthermore both works are only based on small-scale and qualitative evaluations.

where $z$ is the representation of a composed phrase, consisting of two constituents, $x$ and $y$, connected by a syntactic relation $R$, subject to additional knowledge $K$, and composed by some function $f$.

A number of other theories of compositional distributional semantics have been proposed such as the Compositional Matrix-Space Model of Rudolph and Giesbrecht (2010), which models distributional composition on the basis of matrix multiplication. Rudolph and Giesbrecht (2010) show that the framework of Mitchell and Lapata (2008), as well as other models such as Holographic Reduced Representations (Plate, 1995) and symbolic approaches (Clark and Pulman, 2007), can be encoded by their framework. While Rudolph and Giesbrecht (2010) do not provide an implementation themselves, Yessenalina and Cardie (2011) show that the model can be used to estimate the sentiment of short phrases, but note that the model is difficult to train.

Another alternative theory for compositional distributional semantics has been proposed by Clarke (2012), who mathematically formalises the "meaning as context" hypothesis. Clarke (2012) primarily investigates the abstract properties of his framework, focusing on recognising textual entailment as a potential target application. Clarke (2012) shows that his mathematical formulation encodes several other theories of composition, including the categorical framework of Coecke et al. (2011), the Compositional Matrix-Space Model of Rudolph and Giesbrecht (2010), and the framework of Mitchell and Lapata (2008). However, a detailed discussion of the theories proposed by Rudolph and Giesbrecht (2010) and Clarke (2012) is out of scope of this work.

In the following, I will make a high-level distinction between distributional composition as a function between elementary word representations as vectors (§ 2.2.1), and composition based on formal semantic principles in a more general tensor space (§ 2.2.2).

Section 2.2.1 includes the discussion of simple pointwise algebraic composition models, and shows how these represent a special and simplified case of using a neural network as composition function. Section 2.2.2 reviews the body of research which can be embedded within, or derived from, the categorical framework of Coecke et al. (2011). Furthermore it is yet to be determined whether a general unsupervised composition function, suitable across a variety of tasks, can be defined, or whether composition is a task specific concept.

### 2.2.1 *Distributional Composition Based on Word Representations as Vectors*

The perhaps most widely adopted method for composing longer units of text is to represent individual words as vectors in a high-dimensional space and model distributional composition as simple pointwise algebraic operations between elements of this space. One major advantage of this approach — alongside its simplicity — is that words, phrases and sentences, independently of length and structure, are represented in the same semantic space. This has the effect that distributional similarity estimates between any constituents — elementary or composed — are as straightforward as in a distributional space solely consisting of individual words.

Two of the simplest algebraic composition functions are pointwise vector addition and multiplication (see Equation 2.4), which despite their simplicity, have been shown to work remarkably well in numerous studies (Mitchell and Lapata, 2008, 2010; Blacoe and Lapata, 2012; Hill et al., 2016; Kober et al., 2017b).

$$\begin{aligned} z &= x + y \\ z &= x \odot y \end{aligned} \tag{2.4}$$

Equation 2.4 above notably neglects the syntactic relation $R$ between $x$ and $y$, as well as any form of additional knowledge $K$ as defined in the general composition function defined by Mitchell and Lapata (2008) (see Equation 2.3). Pointwise addition, or averaging of word vectors[8], has furthermore been shown to be an effective composition function for creating vector representations of sentences and documents, used as input to neural network models for downstream processing tasks. For example, Iyyer et al. (2015) and Wieting et al. (2016) show that averaging word vectors is competitive, and can even outperform more complex neural network models such as LSTMs, for recognising textual entailment, judging textual similarity and sentiment analysis.

An important note is that composition by pointwise addition exhibits contrasting behaviour when used in an explicit high-dimensional co-occurrence space as in a count-based model, as opposed to a low-dimensional space in a predict-based model. Composition by pointwise addition corresponds to a union of the feature spaces of

8 Vector average refers to pointwise addition, followed by a normalisation step.

two constituent word vector representations in a count-based model[9]. However, it (approximately) corresponds to feature intersection in a predict-based model (Tian et al., 2017). Feature intersection in an explicit count-based model would be realised by pointwise multiplication[10].

A major weakness of the above approaches is that composition is a commutative operation, resulting in identical representations for the two phrases *author sues publisher* and *publisher sues author*, which is undesirable for a fine-grained understanding of the semantics of natural language. Especially complex tasks such as question answering or recognising textual entailment require a detailed notion of "what is being done to whom, how, where and when", where capturing the semantics of a predicate and its subject and object in the composition process is a crucial aspect. A further problem with pointwise algebraic composition functions is that all constituents in a phrase are assigned equal weight. This results in ignoring the role of modifiers in adjective-noun phrases present in the text, such as in the phrase *big seagull*, which is still more of a *seagull* than *something big*, and the composed phrasal representation should ideally reflect that.

A simple fix to both problems would be to use two scalars, $\alpha$ and $\beta$, where $\beta = 1 - \alpha$, to re-weight the contributions of the two constituents accordingly (Mitchell and Lapata, 2008, 2010) as shown in Equation 2.5.

$$\begin{aligned} z &= \alpha\, x + \beta\, y \\ z &= x^{\alpha} \odot y^{\beta} \end{aligned} \tag{2.5}$$

With $\alpha = \beta = 1$ the equivalent of the composition functions in Equation 2.4 would be recovered[11]. While the weighting problem can be adequately addressed, the issue of commutativity still somewhat prevails as only the magnitude of the contextual features in a composed representation changes, but not the semantics of the vector represent-

9 In case where some form of PMI is used as lexical association function, pointwise addition furthermore results in multiplying the probability distributions of two word representations due to the use of the log in PPMI (Ganesalingam and Herbelot, 2013).

10 There is very little research into what category pointwise multiplication in predict-based models falls. The lack of research on this composition function is presumably a consequence of the relatively poor performance of this operation on several tasks (Dinu et al., 2013; Hill et al., 2016; Kober et al., 2017b)

11 In case $\alpha = \beta = 0.5$ the weight averaged equivalent would be recovered for the pointwise additive composition function.

ations *per se*.

A more flexible weighting scheme has therefore been introduced by Guevara (2010, 2011) and Zanzotto et al. (2010), who generalise additive composition to be the sum of two matrix multiplications, where the word vector representations $x$ and $y$ are parameterised by matrices $\mathbf{A}$ and $\mathbf{B}$ respectively, as Equation 2.6 below shows.

$$z = \mathbf{A}\,x + \mathbf{B}\,y \tag{2.6}$$

Notably, the scalar model can be recovered if the weight matrices $\mathbf{A}$ and $\mathbf{B}$ are the identity matrix, with the diagonal entries scaled by $\alpha$ and $\beta$ respectively[12]. Obtaining the weights for the matrices $\mathbf{A}$ and $\mathbf{B}$ is more complex than for the simpler pointwise composition functions, and requires a supervised learning regime. To simplify the training process, Guevara (2010) restricts himself to only modelling adjective-noun phrases and extends his approach to verb-noun phrases in later work (Guevara, 2011).

Training the weight matrices $\mathbf{A}$ and $\mathbf{B}$ is a three step process. The first step involves building a standard count-based distributional space from a given corpus to obtain representations for individual lexemes. In the second step, a number of high-frequency adjective-noun and verb-noun phrases are extracted from the corpus and encoded as single tokens. For example a phrase like *big seagull* would become the "pseudo-lexeme" *big_seagull*. Subsequently, a distributional space with the encoded pseudo-lexemes is built. In the final step, the distributional vector representations of the encoded phrases serve as the targets in a partial least squares regression model which is trained with the original word vectors obtained in step 1 as input. Guevara (2011) uses a single weight matrix per phrase type, such that the obtained matrix represents an estimation of how an adjective or a noun modify their respective phrasal heads. By using a weight matrix to parameterise distributional composition, some amount of syntactic idiosyncrasy between the phrasal constituents can be captured, thereby representing a way to include $R$ from Equation 2.3 into the composition process.

Similarly Zanzotto et al. (2010) also focus on particular phrase types only, concentrating on adjective-noun, noun-noun and verb-noun pairs. Furthermore, Zanzotto et al. (2010) do not distinguish between adjective-nouns and noun-nouns, and learn a single weight

12 The standard additive model of Equation 2.4 can be recovered by setting $\mathbf{A} = \mathbf{B} = \mathbb{1}$.

matrix for both types of noun phrases. Their training regime resembles that of Guevara (2011), however instead of creating "pseudo-lexemes", Zanzotto et al. (2010) rely on a dictionary to extract a short equivalent expression for a given lexeme, such as *close interaction* for *contact*. Training proceeds by learning a mapping between individual lexemes as targets (e.g. *contact*) and their respective compositional representation of the equivalent description (e.g. *close interaction*), using a partial least squares regression model. Essentially, this formulation of training represents a paraphrase objective.

Following this line of reasoning, a further generalisation of the pointwise additive model can be formulated as a neural network. This can be achieved by concatenating the two weight matrices, $\mathbf{A}$ and $\mathbf{B}$, from Equation 2.6 into a single weight matrix $\mathbf{W} = [\mathbf{A}; \mathbf{B}]$, resulting in:

$$z = f\left(\mathbf{W}\begin{bmatrix} x \\ y \end{bmatrix} + b\right) \tag{2.7}$$

where $f$ represents an elementwise non-linearity such as `tanh`, and $b$ represents a bias term. This equation is equivalent to the one used by Socher et al. (2011) for defining their recursive neural network. If $f$ is the identify function and $b = 0$, the additive model of Guevara (2010, 2011) and Zanzotto et al. (2010) would be recovered, completing the direct link from simple and unweighted pointwise addition as in Equation 2.4 to a neural network formulation as in Equation 2.7 above.

One major difference between neural networks and general additive models for distributional composition is that the weights in a neural network model are optimised w.r.t. to a downstream task such as recognising textual entailment or sentiment analysis. This has the effect of learning a task specific composition function that is frequently difficult to transfer to a another task (Mou et al., 2016). Modelling distributional composition with neural networks as part of an end-to-end learning strategy for a downstream task has seen a surge of interest in recent years. Two of the main architectures used for modelling distributional composition are convolutional neural networks and recurrent/recursive neural networks[13]. Given their current pop-

13 The recursive neural network architecture represents a generalisation of recurrent neural networks from sequence structures to tree structures (Goldberg, 2017).

ularity I will briefly review these two model types, focusing on their mechanism to perform distributional composition.

*Distributional Composition with Convolutional Neural Networks*

Convolutional neural networks have been pioneered in computer vision (LeCun et al., 1989) and have been shown to achieve impressive performance on numerous image classification and object detection tasks (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2014). Somewhat surprisingly, convolutional neural network architectures have also been shown to achieve competitive performance for computer vision tasks without any training at all (Saxe et al., 2011).

In recent years, these models have been applied to a variety of NLP tasks, ranging from named entity recognition and semantic role labelling (Collobert et al., 2011) to language modelling (Pham et al., 2016) and machine translation (Kalchbrenner et al., 2016).

Their suitability as a model for performing distributional composition has predominantly been evaluated on sentence level text classification and sentiment analysis tasks (Kalchbrenner et al., 2014; Kim, 2014; Le and Zuidema, 2015; Mou et al., 2015).The models of Kalchbrenner et al. (2014) and Kim (2014) perform distributional composition within a convolutional neural network in a sequential bag-of-words fashion, whereas Le and Zuidema (2015) and Mou et al. (2015) compose distributional word representations on the basis of a parse tree input.

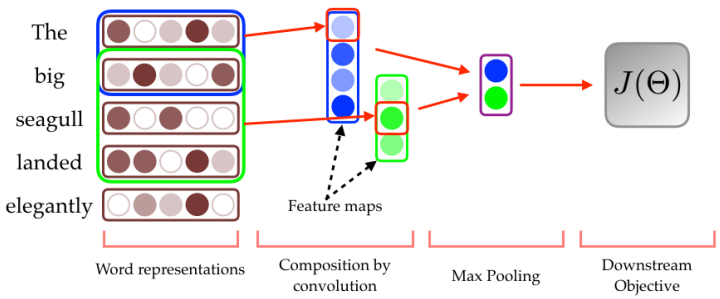


Figure 2.6: Composition operation in a Convolutional Neural Network.

Figure 2.6 shows how distributional composition is modelled in a convolutional neural network, following the architecture of Kim (2014). The input is formed of (usually low-dimensional and dense) distributional word representations, which are composed with a convolution filter[14] (second bracket in Figure 2.6), and which contain

14 Also referred to as “convolution kernel" or “convolution mask".

learnable weights. The filter width is a hyperparameter, with typical ranges spanning from bigrams to 5-grams, and most models use a varying number of filters per width which is usually on the order of ≈100-300 (Kim, 2014; Pham et al., 2016). In Figure 2.6 a bigram filter (blue frame) and a trigram filter (green frame) are applied to the given input sentence *The big seagull landed elegantly*.

Given a word vector space $V$ of dimensionality $\mathbb{R}^{|V|\times 5}$ for all words $x_i, \ldots, x_{|V|} \in \mathbb{R}^{1\times 5}$ in the vocabulary, and a bigram convolution filter, $c_{bi} \in \mathbb{R}^{10\times 1}$, composing any two adjacent[15] word representations $w_i$ and $w_{i+1}$ would work by horizontally concatenating $x_i$ and $x_{i+1}$ in the given input sentence to form a vector $x_{bi} \in \mathbb{R}^{1\times 10}$, building the dot product between $x_{bi}$ and $c_{bi}$, adding a bias term and applying a non-linearity. The resulting scalar, $z_i$, is then placed in the feature map corresponding to the convolution filter. More formally, given the horizontal concatenation of the word vectors in an input sentence $X_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n$ of length $n$, and a convolution filter $c \in \mathbb{R}^{l\times 1}$, the feature map entry $z_i \in \mathbb{R}$ is defined as:

$$z_i = f(X_{i:i+l-1} \cdot c + b) \tag{2.8}$$

where $f$ is an elementwise non-linearity, $b$ a bias term and $c$ the convolution filter containing the learnable weights.

Thus, convolving the example sentence with the bigram and trigram filter results in feature maps of size 4 and 3, respectively. Notably, any composed $n$-gram is compressed into a *scalar quantity* when stored in a feature map[16]. The size of the feature maps depends on the sentence length and the width of the convolution filter. In order to obtain a fixed length feature vector, a pooling operation is performed which selects the entry with the highest numerical magnitude[17] (see the max pooling operation in the third bracket in Figure 2.6) from each feature map. The resulting feature vector is subsequently used as input to a classifier (often referred to as "fully connected layer") and the whole network is trained via backpropagation (Rumelhart et al., 1986). Globally, the composition mechanism, represented by the convolution machinery, acts as a feature detector for the given task and is optimised on the downstream objective of that task.

15 Either spatially adjacent or adjacent in a parse tree.

16 An alternative, feature based convolution has been proposed by Kalchbrenner et al. (2014) who convolve elementwise over the given input sequence.

17 Alternatively, the mean or the top $n$ highest values could be chosen (Kalchbrenner et al., 2014).

*Distributional Composition with Recurrent and Recursive Neural Networks*

Recurrent neural network models operate on sequential input data (Elman, 1990) and have been generalised to tree structures by Pollack (1990). The uni-directional recurrent variant by Elman (1990) has been extended to a bi-directional variant, simultaneously processing data in a forward and backward mode, by Schuster and Paliwal (1997). Recently, bi-directional variants of recursive and recurrent neural networks have been shown to achieve strong performance for a variety of NLP tasks ranging from parsing (Dyer et al., 2016; Kiperwasser and Goldberg, 2016) to text classification and sentiment analysis (Teng and Zhang, 2017), and recognising textual entailment (Chen et al., 2016; Liu et al., 2016).

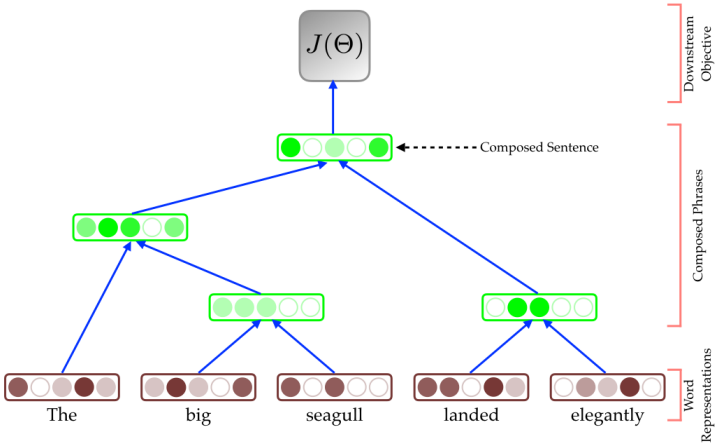


Figure 2.7: Composition operation in a Recursive Neural Network.

Figure 2.7 shows how distributional composition can be modelled with a simple uni-directional recursive neural network, operating on a binarised constituency parse tree. The inputs are usually low-dimensional and dense word vector representations (bottom row in Figure 2.7), which are composed in a bottom-up fashion, following the given parse tree structure. Intermediary nodes represent composed phrases (the green row vectors in Figure 2.7) of varying lengths, which share the characteristic of pointwise algebraic composition functions of having the same dimensionality as individual word vectors. The top green row vector in Figure 2.7 represents the composed sentence and is fed into a classifier for a downstream task. The network is optimised on the given downstream objective and trained through backpropagation.

An interesting consequence of the composition mechanism based on a binarised constituency tree is that the weight matrix $\mathbf{W}$ can be

factorised into two matrices $\mathbf{W_l}$ and $\mathbf{W_r}$ as shown in Equation 2.9 (and as already shown in Equations 2.6 and 2.7), resulting in *position dependent* parameterisations of whether a word (or composed phrase) is the left or the right constituent (Socher et al., 2014).

$$f\left(\mathbf{W}\begin{bmatrix}x\\y\end{bmatrix}+b\right) = f(\mathbf{W_l}\,x + \mathbf{W_r}\,y + b) \quad \text{if } \mathbf{W} = [\mathbf{W_l}; \mathbf{W_r}] \tag{2.9}$$

In order to overcome that issue and avoid the model's reliance on binarised trees, Socher et al. (2014) generalised the recursive neural network to $n$-ary trees such as resulting from a dependency parse of a given input sentence. Hermann and Blunsom (2013) and Socher et al. (2014) furthermore introduced weight tying between different grammatical relations, such that all phrase types would be parameterised by individual weight matrices. For example, all adjective-noun phrases would be parameterised by $\mathbf{W}_{\texttt{amod}}$, all verb-object phrases by $\mathbf{W}_{\texttt{dobj}}$, and so on. Interestingly, this idea follows Guevara (2011), but for any phrase type and with a downstream training objective instead of a pseudo-lexeme objective. Formally, distributional composition with tied weights in an $n$-ary tree can be represented by:

$$z = f(b_t + \sum_{x_i \in X} \mathbf{W_t} \cdot x_i) \tag{2.10}$$

where $x_i$ is the $i^{\text{th}}$ child of the parent node $X$ in a given dependency tree, $z$ the composed phrase of all child nodes of $X$, $\mathbf{W_t}$ and $b_t$ are the weight matrix and bias term of type $t$, representing the dependency relation between the lexeme at node $X$ and its $i^{\text{th}}$ child $x_i$, and $f$ is an elementwise non-linearity such as `tanh`. This formulation of distributional composition fully encodes the syntactic relation $R$ between two constituents.

Instead of using a simple recursive neural network, more complex neural network models can be used, such as a long-short term memory network (LSTM) (Hochreiter and Schmidhuber, 1997), which has been generalised to work on tree structures by Tai et al. (2015) and Zhu et al. (2015a). More complex networks have often been shown to perform better than their simpler counterparts due to their improved ability to model long-range dependencies. However, they have themselves been frequently outperformed by much simpler composi-

tion functions such as averaging word vectors (see Equation 2.5) in a number of studies (Iyyer et al., 2015; Wieting et al., 2016).

### 2.2.2 *Distributional Composition Based on Formal Semantics*

One of the major advancements of distributional semantics has been the development of a continuous model of meaning on the lexical level, resulting in rich representations for content words such as adjectives, nouns or verbs, and allowing fine grained distinctions in meaning between any two lexemes. However, the lack of structure in a continuous space makes it difficult to model more discrete linguistic phenomena such as negation, or representing operators for quantification. Furthermore, distributional semantics has been lacking a clear theory of how longer phrases can be modelled in a compositional way, extending the continuous model of meaning from the lexical to the phrasal level (Baroni, 2013).

On the other hand, semantic composition based on formal semantics (Montague, 1970), has provided a strong foundational theory of compositionality, focusing on recursive compositional rules to derive the meaning of complex expressions. Unlike distributional semantics, the form of representation of formal semantic expressions is not a continuous vector space, but a logic formalism such as higher-order predicate logic and the lambda calculus, where the meaning of a composed sentence is frequently modelled as its truth value. On the level of individual lexemes, formal semantics is primarily concerned with negation, quantification and the role of function words, but oftentimes treats content words as unanalysed primitives (Partee, 2016).

Hence, the two theories have frequently been described as complementary and a considerable amount of research effort has been spent on integrating the strengths of both approaches into a single unified model. For example, Lewis and Steedman (2013) represent natural language meaning as a combination of first-order logic for function words and distributional semantic representations for content words such as nouns and verbs. The logical form is obtained from a CCG parse of the input sentence and the relational terms in the logical expressions are represented by distributional representations. In order to improve the generality of their approach, Lewis and Steedman (2013) subsequently cluster the distributional representations of the relational terms in order to better capture the association between two

typed entities (e.g. a book and its author). The distributional clustering has the effect of abstracting the representation away from differences in sentence structure such as the use of relative clauses or passives. For example for the relational terms *wrote*, *was written by* and *is author of* in the respective phrases *Shakespeare wrote Macbeth*, *Macbeth was written by Shakespeare*, and *Shakespeare is the author of Macbeth*, would all be clustered together and represented by a relational identifier such as *relation42* that captures the relationship between a book and an author. Due to their use of CCG with its transparent syntax-semantics interface (Steedman, 2000), composition would be supported *per se*, however is not explored in this work. Lewis and Steedman (2013) showed that their approach achieves strong performance for question answering and recognising textual entailment.

A similar approach has been proposed by Beltagy et al. (2016) who also represent natural language meaning as a combination of logical and distributional semantics on the basis of Markov logic networks. The main difference between their approach and the model of Lewis and Steedman (2013) is the inference method. Where Lewis and Steedman (2013) use standard first-order inference, Beltagy et al. (2016) use probabilistic logic in order to model the uncertainty of inferences.

Asher et al. (2016) attempt to integrate Type Composition Logic with distributional semantic vectors where the use of types is aimed at appropriately restricting the kinds of arguments that a given predicate can take. When composing two constituent words, the types have the effect of *contextualising* the representations in the given phrase. The model of Asher et al. (2016) can furthermore be interpreted as a formal semantics flavoured version of the models of Guevara (2011) and Socher et al. (2012).

However, as neither of these approaches propose a theory of compositionality within a distributional framework *per se*, a more detailed discussion of the models of Lewis and Steedman (2013), Asher et al. (2016) and Beltagy et al. (2016), together with related approaches that aim to leverage the strengths of formal and distributional semantics, is out of scope of this work.

A general framework for semantic composition on the basis of Lambek pregroup grammar (Lambek, 2001, 2008) was introduced by Coecke et al. (2011) who proved that pregroups and vector spaces share the same abstract structure, referred to as a *compact closed category*. The approach is inspired by the idea of combining symbolic and distributional models of word meaning (Clark and Pulman, 2007).

The framework is based on the formal semantic notion of function application, modelling atomic units such as nouns as distributional semantic vectors, and relational terms such as adjectives or verbs as tensors. Distributional composition is achieved by a tensor product between the constituents in a phrase. The abstract formulation of distributional composition by Coecke et al. (2011) gives rise to a number of different strands of work, focused on providing a faithful implementation of the theory.

For the purpose of this thesis, I will distinguish three different strands of work, the first one being the body of work originating from the *lexical function (lf)* model (Baroni and Zamparelli, 2010). The concrete instantiation of the lexical function model is contained within the categorical framework of Coecke et al. (2011)[18]. I will therefore discuss the approach of Baroni and Zamparelli (2010) (§ *Composition Based on the Lexical Function Model*) as a derivative of the more formal definition of Coecke et al. (2011), despite the fact that the two ideas have been developed independently at approximately the same time.

The second strand of work comprises concrete instantiations of the categorical framework (Coecke et al., 2011) on the basis of the Lambek pregroup grammar and is discussed in Section *Composition based on Pregroup Grammar*. The third strand of work follows the abstract framework of Coecke et al. (2011), but uses Combinatory Categorical Grammar (CCG) instead of Lambek pregroup grammar as the underlying formalism, and is presented in Section *Composition based on Combinatory Categorical Grammar*.

While it is straightforward to build vector representations for atomic types such as nouns, these approaches share the practical bottleneck of deriving representations for relational terms such as adjectives and verbs which are modelled as higher-order tensors. Much of the research has therefore been focused on how these higher-order structures can be effectively extracted and built from the given data.

### *Composition Based on the Lexical Function Model*

The *lexical function (lf)* model of Baroni and Zamparelli (2010) focuses on adjective-noun phrases and aims to follow the formal semantic notion of modelling attributive adjectives as functions over content nouns, which are represented as distributional semantic vectors. Equation 2.11 shows how a phrase vector $z$ would be obtained

18 It is also contained within the general framework of Baroni et al. (2014), introduced a few years later.

by multiplying the adjective, represented by the matrix $\mathbf{X}$ with the noun vector $y$.

$$z = \mathbf{X} \cdot y \tag{2.11}$$

Unlike the composition model by Guevara (2011), the matrix $\mathbf{X}$ does not represent a weight acting on a content adjective, but aims to directly encode the semantics of how an adjective modifies a given noun. While vector representations of nouns are obtained in the "standard" distributional semantic way, the adjective matrices need to be learnt in a supervised setting.

Following Guevara (2010), Baroni and Zamparelli (2010) use a partial least squares regression model to predict a vector representation of an observed adjective-noun phrase $z$, that is obtained in the same way as in Guevara (2010), with just the noun vector $y$ as input. Thus, the matrix $\mathbf{X}$ encodes the modifier semantics of how an adjective changes its head noun. However, instead of training a single matrix for all adjectives, Baroni and Zamparelli (2010) learn one matrix per adjective.

Baroni and Zamparelli (2010) showed that their *lf* model produces composed adjective-noun representations that are closer, in terms of cosine similarity, to corpus observed representations than unsupervised baselines such as pointwise addition or multiplication. The *lf* model was later extended to verb phrases (Grefenstette et al., 2013), again achieving improvements over simple pointwise additive and multiplicative baselines. More recently, Vecchi et al. (2016) showed that the *lf* model is also effective at recognising nonsensical adjective-noun combinations, such as *parliamentary tomato*.

A major obstacle for scaling their model to other phrase types and longer expressions is the fact that the order of the predicate representations depends on the valency[19] of the given lexeme in context. For example, while adjectives or intransitive verbs can be represented by matrices acting on noun vectors, transitive verbs such as *eat* in the context of *seagulls eat fish*, would need to be represented by 3rd order tensors. Furthermore, *eat* in an intransitive context such as *seagulls eat* would be different from *eat* in a transitive context, as it would need to be modelled as a 2nd order tensor (a matrix) in the intransitive case and as a 3rd order tensor in the transitive case. This has the consequence of not being able to share distributional information between the two representations and furthermore having fewer ob-

19 Also referred to as *arity*, however I will be using *valency* throughout this thesis.

servations for each case in the source corpus. The *lf* model has been scaled to transitive phrases by Grefenstette et al. (2013) via a multi-step regression learning regime. In the first step the model learns predicate verb phrase matrices and then uses these to estimate the corresponding verb tensor, however this approach remains difficult to scale beyond short phrases.

To overcome the issues of estimating valence dependent tensors, Paperno et al. (2014) proposed the *practical lexical function (plf)* model, which relaxes the restriction that lexeme valency is modelled by tensor order. Instead, every lexical item is represented by a distributional semantic vector, encoding the given lexeme as "content word", and a number of matrices encoding its predicate semantics. The number of matrices for a given lexeme is dependent on its valency. This results in adjectives such as *big* being represented by a vector and one matrix, and transitive verbs such as *catch* being represented by a vector and two matrices, a distinct matrix for encoding the subject and object function-argument relations, respectively. Given an example transitive phrase such as *seagulls catch fish*, a phrasal representation would be obtained by:

$$z = x + \sum_{y_t^{(i)} \in Y, \mathbf{X_t} \in \mathcal{X}} \mathbf{X_t} \cdot y_t^{(i)} \tag{2.12}$$

where $x$ represents the vector representation for the verb *catch*, $\mathbf{X_t}$ represents the matrices encoding the object and subject function-argument semantics of *catch*, such that $\mathcal{X}^{\text{catch}} = \{\mathbf{X_{nsubj}}, \mathbf{X_{dobj}}\}$, and the set $Y = \{\texttt{seagulls}_{nsubj}, \texttt{fish}_{dobj}\}$ contains the typed content representations of the nouns *seagulls* and *fish*, respectively. Paperno et al. (2014) found that learning a set of matrices for encoding the functional semantics of predicates is much easier to scale, and achieves better generalisation, than the valency dependent tensor approach of the *lf* model. The *plf* model has furthermore been shown to be effective for modelling longer and more complex sentences such as relative clauses (Rimell et al., 2016).

An alternative generalisation of the *lf* model for adjective-noun and noun-noun compounds has been proposed by Bride et al. (2015), who learn a 3rd order tensor instead of individual matrices for representing the functional modifier semantics, relying on tensor decomposition for dimensionality reduction of the modifier data structure. Composition for an adjective-noun phrase such as *big seagull* is defined as:

$$z = (\boldsymbol{\mathcal{X}} \cdot x) \cdot y \tag{2.13}$$

where $\boldsymbol{\mathcal{X}}$ is the adjective tensor, encoding the semantics of adjectival modification, $x$ is the vector representation for the adjective *big* and $y$ is the vector representation of the noun *seagull*.

*Composition based on Pregroup Grammar*

A first implementation of the categorical framework by Coecke et al. (2011) on a toy dataset has shown that the model is able to capture interesting compositional distributional patterns such as plausible similarity scores between sentences with different structure, albeit no quantitative experimental evaluation has been conducted (Grefenstette et al., 2011).

A more realistic implementation on a real-world corpus and with a quantitative evaluation of the theory has been provided by Grefenstette and Sadrzadeh (2011a), who focus on modelling intransitive and transitive verb phrases. In their instantiation, they make the simplifying assumption that a transitive verb is modelled as a matrix ($2^{\text{nd}}$ order tensor) instead of a $3^{\text{rd}}$ order tensor. This has the consequence that sentences and phrases with different lengths or structure live in different vector spaces. For example while intransitive verb phrases would live in $\mathbb{R}^i$, transitive verb phrases would live in $\mathbb{R}^{i \times j}$ and ditransitive ones in $\mathbb{R}^{i \times j \times k}$. This characteristic severely limits the scalability and applicability of the concrete instantiation.

In their model, nouns are represented by standard count-based distributional semantic vectors, obtained in an unsupervised way from the given source corpus. Unlike Baroni and Zamparelli (2010), matrix representations for predicates are computed in a bottom-up manner from the given distributional information in the corpus, rather than in a top-down way. More concretely, a verb matrix is represented by the sum of the Kronecker products of all the individual subject and object pairs that co-occurred with the given verb in the source corpus (Grefenstette and Sadrzadeh, 2011a,b). Distributional composition for a transitive verb phrase is computed as the Hadamard product between the verb matrix and the Kronecker product of the distributional semantic vector representations of the constituent subject and object representations:

$$\mathbf{Z} = \mathbf{V} \odot (s \otimes o) \tag{2.14}$$

where $\mathbf{Z}$ represents the resulting phrase as a matrix $\in \mathbb{R}^{i \times j}$, $\mathbf{V}$ represents the verb matrix, and $s$ and $o$ the corresponding subject and object vector representations, respectively.

Despite the above mentioned shortcomings, Grefenstette and Sadrzadeh (2011a) showed that their implementation is able to outperform simple additive and multiplicative composition models on an intransitive and a transitive verb phrase similarity task. In subsequent work, Grefenstette and Sadrzadeh (2011b) compared several different ways to building a verb matrix and found that instead of computing the verb matrix from the distributional information of object and subject nouns, the Kronecker product between the basic distributional semantic vector representation of the given verb with itself derives a better representation.

$$\mathbf{V} = x \otimes x \tag{2.15}$$

Equation 2.15 shows how a verb matrix $\mathbf{V}$ is computed as the Kronecker product between the corresponding distributional semantic vectors of the given verb $x$ with itself[20]. While this method significantly improved performance for a transitive verb phrase similarity task, the above mentioned problem of phrases of different length and structure living in different vector spaces, remained unaddressed.

A solution to this fundamental problem has been proposed by Kartsaklis et al. (2012), who stipulated that a composed sentence $S$ needs to live in the same vector space as any other atomic units, $S \in \mathbb{R}^i$. The major challenge for this approach to work is to construct the required $3^{rd}$ order tensor for a transitive verb from the given matrix, resulting from the bottom-up construction approach for representing relational words. Kartsaklis et al. (2012) proposed two different methods for mapping the verb matrix into a $3^{rd}$ order tensor. The first approach, "copy-subject (CpSbj)", copies the dimension corresponding to the subject to form a $3^{rd}$ order tensor and "copy-object" (CpObj) copies the dimension corresponding to the object. $3^{rd}$ order tensors are built by applying the Kronecker product to subject-subject-object vectors (CpSbj) or to subject-object-object vectors (CpObj). Equation 2.16 shows the two methods more formally.

$$\begin{aligned} \mathcal{V}_{CpSbj} &= s \otimes s \otimes o \\ \mathcal{V}_{CpObj} &= s \otimes o \otimes o \end{aligned} \tag{2.16}$$

20 For two vectors, the Kronecker product is identical to the outer product.

$\mathcal{V}_{CpSbj}$ and $\mathcal{V}_{CpObj}$ are the verb tensors for each method respectively, and $s$ and $o$ represent the distributional semantic vectors for the subject and object. Composition of a transitive subject-verb-object phrase can then be achieved through tensor contraction between a subject vector, a verb tensor and an object vector:

$$z = s \cdot \mathcal{V} \cdot o \tag{2.17}$$

where $z$ is a representation of the phrase and has dimensionality $\mathbb{R}^i$, $s$ and $o$ are the distributional semantic vector representations of the subject and object, respectively, and $\mathcal{V}$ represents the verb tensor obtained via the copy-subject or the copy-object method. Kartsaklis et al. (2012) found the copy-object method generally outperforming its copy-subject counterpart for a transitive verb phrase similarity task and a definition classification task. A top-down alternative approach to constructing the necessary verb data structure for modelling a transitive verb phrase is presented by Grefenstette et al. (2013), who follow Baroni and Zamparelli (2010) and formulate the supervised training procedure as a multi-step regression problem.

Further work by Kartsaklis and Sadrzadeh (2013) and Kartsaklis et al. (2014) incorporated a contextualisation mechanism that disambiguates the sense of the given words in context and showed that this approach can further improve performance of the tensor based models for a variety of distributional composition tasks.

*Composition based on Combinatory Categorical Grammar*

An alternative to the Lambek pregroup grammar formulation of Coecke et al. (2011) has been proposed by Maillard et al. (2014) who show that the categorical framework of Coecke et al. (2011) can be seamlessly integrated with Combinatory Categorical Grammar (CCG) (Steedman, 2000). Unlike the “shallow" CCG based compositional distributional model of Hermann and Blunsom (2013), which represents all words as vectors and encodes the CCG rules and types on the basis of tied weights in a neural network, Maillard et al. (2014) follows the framework of Coecke et al. (2011) which ties the order of a lexeme representation to its valency, representing nouns as vectors, adjectives as matrices and transitive verbs as $3^{rd}$ order tensors. As with all practical implementations based on the abstract framework of Coecke et al. (2011), the major bottleneck of the formulation by Maillard et al. (2014) is to efficiently construct high-quality representations for higher-order tensors from the given data.

Distributional composition of a transitive subject-verb-object phrase is defined in the same way as in Equation 2.17 above. In order to overcome the sparsity issue in estimating the parameters of a 3rd order tensor for a transitive verb, Polajnar et al. (2014a) introduce 3 methods that approximate the tensor with a matrix, and model distributional composition as a combination of matrix and vector products. Their best performing method decouples the direct interaction between the subject and object vector representations and models composition as a concatenation of two matrix-vector products:

$$z = \mathbf{V}_{nsubj} \cdot s \oplus \mathbf{V}_{dobj} \cdot o \tag{2.18}$$

where $\oplus$ is vector concatenation, $s$ and $o$ are the distributional semantic vector representations of the subject and object, respectively, and $\mathbf{V}_{nsubj}$ and $\mathbf{V}_{dobj}$ are the corresponding verb matrices. While the approach of Polajnar et al. (2014a) is not fully faithful to the formulation of Coecke et al. (2011) in terms of modelling a transitive verb as a 3rd order tensor, their approach does not suffer from the shortcoming of representing different kinds of verb phrases in different vector spaces.

A more faithful implementation of the categorical framework by Coecke et al. (2011), and alternative to the proposed method by Polajnar et al. (2014a) has been proposed by Fried et al. (2015). Instead of decoupling the subject and object interaction, Fried et al. (2015) applied tensor decomposition (Kolda and Bader, 2009) to the obtained verb tensor in order to reduce its dimensionality. They created an initial verb tensor by multi-linear regression, similar to the method of Bride et al. (2015). While their approach vastly reduces the parameter space, tensor decomposition generally leads to inferior performance on two transitive verb phrase similarity tasks in comparison to modelling the full tensor.

Rimell et al. (2016) introduced a dataset for modelling relative clauses, which in the categorical framework of Coecke et al. (2011) is defined as a noun modifier, representing a mapping from a composed transitive verb phrase to a modifier of the head noun. Following the categorical framework, a relative pronoun would need to be represented as a 4th order tensor, however due to the limited amount of available training data, Rimell et al. (2016) introduce two simplifications in order to approximate the relative pronoun tensor. Firstly, following Paperno et al. (2014), relational transitive verbs are modelled as a pair of matrices rather than a 3rd order tensor, reducing the

required order of the relative pronoun tensor from 4 to 3. Secondly, the same approach is subsequently applied to model the relative pronoun as another pair of matrices instead of a 3rd order tensor.

For evaluation, Rimell et al. (2016) compare several proposed tensor based compositional models (Grefenstette and Sadrzadeh, 2011a; Paperno et al., 2014; Polajnar et al., 2014a), together with a number of simpler lexical baselines based on pointwise vector operations, for modelling a relative clause. They found that the *plf* model[21] achieves strong performance, although it did not outperform a simple lexical model based on adding predict-based word vectors. The best performing model in their study was a simplified version of the *plf* model, where the composed verb argument vector is directly combined with the head noun via pointwise addition, instead of first applying the verb predicate matrix to it (Rimell et al., 2016).

While the *plf* model of Paperno et al. (2014) is also governed by a CCG parse of a given sentence, and was developed at approximately the same time as the model of Maillard et al. (2014), it evolved out of the lexical function model, and has thus been discussed previously.

## 2.3 CONTEXTUALISATION — MODELLING WORD MEANING IN CONTEXT

Distributional semantic models do not make a "hard" distinction between word senses and thereby conflate the multiple senses of a polysemous lexeme into a single representation. While it can be argued that this approach is advantageous from a linguistics perspective (Ruhl, 1989), it has recently been subject to a lot of criticism in the NLP community (Chen et al., 2014; Li and Jurafsky, 2015; Iacobacci et al., 2015). However, integrating a word-sense disambiguation component *a priori* might not always be a feasible solution as the granularity and number of different senses per lexeme differs per source corpus and target application.

This reasoning follows Kilgarriff (1997) who argues that individual word senses do not exist *per se*, but only relative to a given task, and that the basic unit of word meaning is not an individual sense, but an occurrence of a word in context (Kilgarriff, 1997). A proposal along similar lines has been put forward by Hanks (2000), who argues that

21 Rimell et al. (2016) use the modified version of the *plf* model, proposed by Gupta et al. (2015), that does not add the vector representation of the verb to the final expression, which Rimell et al. (2016) found to be working better on their dataset.

words outside of any context do not have a specific meaning, but a number of different meaning potentials, which are activated on a continuous spectrum once they are used in context. A further argument against enumerating the senses of a word was put forward by Pustejovsky (1991) on the basis of the argument that additional meanings can arise as a product of composition.

A number of proposals have therefore been made to discriminate the sense of an ambiguous word on the basis of its usage in a given context. This *contextualisation* mechanism to model word meaning for distributional semantic representations in context has the aim to upweight features that the ambiguous word shares with the current context, and downweight incompatible features. Given the close similarity between contextualisation and composition, it has been argued that contextualisation *is* distributional composition (Weir et al., 2016; Kober et al., 2017b). Nonetheless, the two concepts have frequently been treated as related but separate methods, and therefore have their own associated body of work.

In the following, I will review approaches to contextualisation on the basis of multi-prototype and exemplar-based models (§ 2.3.1), models leveraging selectional preference for representing word meaning in context (§ 2.3.2), and approaches based on latent sense modelling (§ 2.3.3).

### 2.3.1 *Contextualisation via Multi-Prototype and Exemplar-Based Models*

A number of approaches in the literature attempt to overcome the issue of conflating multiple senses into a single word representation by creating multiple sense specific prototypes, based on the current context, or aim to leverage a set of exemplars from the given contexts. Prototype models represent a concept on the basis of an abstract instance, aimed to capture the typicality of a set of observations. In the case of a single-prototype model, all the senses of a polysemous lexeme are conflated in a single representation[22].

Multi-prototype models can be seen as a top-down approach, by first creating a single-prototype model and subsequently clustering the contexts of each target word type. A concept is thus represented

22 Most vector-based distributional semantic models — as well as APTS — follow the *one representation per lexeme* paradigm, which means that every lexeme (or word type) is encoded by a single representation, conflating the multiple different meanings of an ambiguous word.

by the cluster centroid of the given contexts. Exemplar-based models on the other hand follow a bottom-up approach by representing a target word type on the basis of a set of concrete observed instances. These are usually filtered upon contextualisation such that only exemplars that are similar to the current context contribute to the representation of a concept.

I will divide the multi-prototype and exemplar-based methods for contextualisation into two rough categories, ones that achieve contextualisation in a self-contained manner by e.g. exploiting the information from the distributional neighbourhood (§ *Self-Contained Contextualisation*) and approaches that require an additional external resource to model the meaning of a word occurrence in context (§ *Contextualisation with External Resources*).

*Self-Contained Contextualisation*

An early approach for modelling the sense of a word in context was attempted by Schütze (1992, 1998), who contextualised a given target word on the basis of its second-order co-occurrence statistics in a particular context. This is achieved by forming a centroid of the context representations of a target word. For example, given the target word *landed* in the sentence *The big seagull landed elegantly*, the context representation of *landed* would be formed by averaging the vector representations for the remaining words in the sentence. Schütze (1998) argues that leveraging second-order co-occurrence statistics in this way is more robust and is not as severely affected by sparsity as first-order co-occurrence information. The averaged context vector representations are subsequently clustered in order to discriminate the different usages of an ambiguous lexeme in context.

Different senses of a given target word are represented by the centroid of their corresponding clusters. A new occurrence of an ambiguous lexeme in a test corpus would be assigned to the sense cluster that is closest given an averaged vector representation of its context. Schütze (1998) shows the merit of his approach on the basis of a small-scale evaluation on discriminating the senses of 10 naturally ambiguous nouns, and 10 pseudo-ambiguous nouns, which are formed by concatenating two unrelated words, such as *seagull* and *turbine* into the single lexeme, *seagull_turbine*.

Contextualising a polysemous lexeme on the basis of leveraging second-order contextual information has also been shown to improve

performance for modelling distributional composition (Kartsaklis et al., 2013). Their approach to sense disambiguation is based on the idea of Schütze (1998), where the context representations of a given target word are averaged and subsequently clustered to represent the senses of the given target word. Following Schütze (1998), Kartsaklis et al. (2013) use a hierarchical agglomerative algorithm for clustering the context representations. They show that their approach is able to improve upon a baseline without disambiguation on two composition tasks with a standard count-based distributional semantic model. Subsequently Kartsaklis and Sadrzadeh (2013) showed that the same approach also improves performance for the categorical model.

An alternative approach has been proposed by Reisinger and Mooney (2010a), who create a standard count-based distributional vector space and subsequently cluster the collected contexts of each lexeme. Unlike the approaches of Schütze (1998) and Kartsaklis et al. (2013), the clustering is based on individual representations of context words rather than the centroid of context representations in a given sentence. This results in a set of clusters per word type that captures the different usages of that word in a given corpus. Each word in the vocabulary can then be described as a set of cluster centroids. Clustering of contexts is achieved by a method based on a mixture of von Mises-Fisher distributions (Banerjee et al., 2005), which like spherical $k$-means uses cosine to estimate the semantic similarity between two context representations.

Distributional similarity between two words can be estimated in an out-of-context fashion by either computing the average or maximum similarity between the respective centroids of two given words. For in-context distributional similarity estimates, Reisinger and Mooney (2010a) calculate the probability of a context belonging to a given cluster. Their model is evaluated on an out-of-context word-similarity task and an in-context near-synonym prediction task, where Reisinger and Mooney (2010a) observe performance improvements with their multi-prototype vector model.

In a subsequent paper Reisinger and Mooney (2010b) extend their approach by replacing the hard clustering of contexts with a Dirichlet process mixture model, allowing for soft cluster assignments of contexts, and a tiered model which combines the soft cluster mixture model with a single-prototype approach. They show that their

tiered model improves over a single-prototype and hard cluster multi-prototype for modelling selectional preferences and two word similarity tasks.

Huang et al. (2012) introduced a predict-based counterpart to the count-based models of Reisinger and Mooney (2010a,b), which operates over low-dimensional word embeddings from a neural network. In addition to the local context, captured by the word embeddings, Huang et al. (2012) included global document context from a tf-idf representation of the current document into the word representation. For clustering the context representations of a given lexeme, Huang et al. (2012) used spherical $k$-means (Dhillon and Modha, 2001). Distributional similarity estimates in- and out-of-context are computed the same way as in Reisinger and Mooney (2010a). Huang et al. (2012) showed that their multi-prototype model is able to improve performance over a single-prototype baseline for a word similarity task and a novel word similarity in context task that they introduce in the same work.

Instead of modelling word meaning in context with prototypes, Erk and Pado (2010) proposed an exemplar-based approach where each target word is modelled by the set of vector representations of sentences in which the target word occurs. Contextualisation is modelled by a process of selecting the most relevant exemplars for a given context. Relevancy between a context and a set of exemplars is calculated as the cosine similarity between the respective bag-of-words vector representations of the current context and the set of exemplars. Every exemplar representation exceeding a certain similarity threshold subsequently contributes to the contextualised representation of a target word.

Erk and Pado (2010) explore two different variants for modelling the threshold: a static approach that always includes the top $n$ nearest neighbours for every lexeme, and a density based approach that includes all exemplars exceeding a certain similarity. Erk and Pado (2010) showed that their simple exemplar-based approach is able to achieve strong performance on the lexical substitution task (McCarthy and Navigli, 2007) when used in a paraphrase ranking setup.

*Contextualisation with External Resources*

Unlike self-contained contextualisation, which exploits the intrinsic distributional space itself, the following approaches make use of additional external components such as systems for performing word-sense induction and disambiguation, or a language model.

Reddy et al. (2011) proposed two approaches for disambiguating a lexeme prior to composition in a standard count-based distributional semantic space. Their first approach builds upon a graph based word sense induction (WSI) component to derive multiple "static" prototype vectors for a given lexeme in context. The WSI system takes distributional word representations as input and builds a graph where lexemes are represented by vertices, and edges are based on the similarity between two distributional representations. The edge space is pruned upon creation of the graph, based on a pre-determined similarity threshold. The graph is subsequently clustered using the chinese-whispers algorithm (Biemann, 2006) and a number of sense clusters is returned.

The second approach is exemplar-based and builds a "dynamic" prototype by only activating features that are relevant to the current context. For determining the relevancy of a given context feature, Reddy et al. (2011) used Sketch Engine (Kilgarriff et al., 2004) to retrieve weighted collocations for a target lexeme from a corpus and apply a ranking to its output, based on the distributional similarity of ranked words to the given target lexeme. In an evaluation on the noun-noun composition subtask of Mitchell and Lapata (2010), Reddy et al. (2011) showed that their exemplar-based dynamic prototype substantially outperforms a baseline without disambiguation. However, they find their static prototype approach to be performing poorly.

In order to avoid dissecting a single word representation into multiple prototypes, or creating a contextualised representation in an exemplar-based model, Iacobacci et al. (2015) aimed to avoid the conflation of multiple senses into a single representation before constructing their semantic space. In order to achieve the construction of a disambiguated semantic space, Iacobacci et al. (2015) applied BabelNet (Navigli and Ponzetto, 2012), an external word-sense disambiguation system, to a given source corpus. This resulted in multiple vectors per lexeme, rendering contextualisation as a process of selecting the best fitting sense from an explicitly modelled list. Sense selection can be achieved by maximising the distributional similarity between

the sense representations of the target and context words, respectively. Iacobacci et al. (2015) show that their method improves performance on a number of out-of-context word similarity tasks, however has been shown to perform worse on in-context evaluations (Iacobacci et al., 2015; Kober et al., 2017b).

Instead of focusing on senses on the basis of word usage or dictionary definitions, Melamud et al. (2015) concentrated on modelling potential fillers for a given slot. Their approach is based on substitute vectors (Yatbaz et al., 2012), which represent plausible alternatives for a blank slot in a given context, weighted by their suitability for that slot. Melamud et al. (2015) used an $n$-gram language model to determine plausible filler words for a given empty slot. For example in the sentence *The __ seagull landed elegantly*, the language model would be used to determine suitable fillers for the blank space, such as *big*, *hungry* or *wicked*. A contextualised substitute vector representation for a given target word is simply the weighted average of all potential fillers, for a given context. This means that, for the example above, the representation for the empty slot would be the averaged vector of the distributional representations for the lexemes *big*, *hungry* and *wicked*. Melamud et al. (2015) evaluated their substitute vector models on a pseudo-disambiguation task as well as the lexical substitution task, and showed that their approach is able to outperform strong baselines such as `word2vec`.

### 2.3.2 *Contextualisation via Modelling Lexical Selectional Preferences*

The model of Erk and Padó (2008) represents every lexeme as a set of vectors, consisting of one vector encoding the lexical meaning of a word, and a number of vectors encoding the selectional preferences of that word in different syntactic relations. Equation 2.19 formalises the representation as a triple

$$w = (v, R, R^{-1}) \tag{2.19}$$

where $v$ is the distributional semantic vector representation for a lexeme $w$, such as *fish*, $R$ maps a syntactic relation onto a vector describing the selectional preferences of $w$, such as all the adjectival modifiers that appear with *fish*, and $R^{-1}$ maps a relation onto a vector denoting the inverse selectional preferences of $w$, such as all the direct object relations to verbs that *fish* occurs with.

Modelling word meaning in context is therefore a function of the meaning of a word *per se*, combined with its syntactic relation to one or more context words. For example, in order to derive a fully contextualised representation of the verb phrase *catch fish*, the meaning of the verb *catch* in the context of *fish* needs to be combined with the meaning of *fish* in the context of *catch*.

More formally, given the lexemes $x$ and $y$ in each others context, and given a dependency relation $r$ between $x$ and $y$, where $x$ would represent the head of the phrase, their contextualised representations $x'$ and $y'$ are defined as

$$\begin{aligned} x' &= (x \circledast R_y^{-1}(r), R_x - \{r\}, R_x^{-1}) \\ y' &= (y \circledast R_x(r), R_y, R_y^{-1} - \{r\}) \end{aligned} \tag{2.20}$$

where $\circledast$ denotes some vector combination function such as pointwise addition or multiplication, $x \circledast R_y^{-1}(r)$ denotes the combination of the standard distributional semantic representation of $x$ with the inverse selectional preferences of $y$, i.e. nouns in their direct object positions that take *catch* as their verb. Conversely, $y \circledast R_x(r)$, represents the combination of the lexical meaning of $y$ and the forward selectional preferences of $x$, i.e. verbs that take *fish* in their direct object slot. The current relation $r$ is removed from the set of forward and inverse selectional preferences of the resulting phrase as denoted by $R_a - \{r\}$ and $R_b^{-1} - \{r\}$.

Erk and Padó (2008) showed that their model achieves strong performance on the verb-noun short phrase composition task of Mitchell and Lapata (2008) as well as the lexical substitution task.

Instead of explicitly modelling the syntactic selectional preferences of lexemes in a separate set of vectors, Thater et al. (2010) generalised the method of Erk and Padó (2008) and directly encoded typed second-order co-occurrences in a shared vector space. Following Padó and Lapata (2007) and Baroni and Lenci (2010), they encoded $\langle w, r, w' \rangle$ triples in their first-order distributional semantic co-occurrence space.

Second order representations are obtained from the co-occurrence quadruple $\langle w, r, r'w' \rangle$, where $r'$ denotes the second-order syntactic relation between $w$ and $w'$, via $r$. Modelling word meaning in context for a verb phrase such as *catch fish* is achieved by contextualising the second-order representation of *catch* with the first-order representation of *fish*. In this setting the first-order representation has the role

of a weighted filter in order to extract the contextualised meaning of the verb.

Due to explicitly modelling the type in their distributional space[23], Thater et al. (2010) observed that lexemes with different syntactic roles, such as verbs and their direct objects, live in different areas of the typed distributional space. This has the consequence that composition requires an additional step to align the two representations, which Thater et al. (2010) modelled through a "lifting-map" that maps the co-occurrence space spanned by $\langle w, r, w' \rangle$ triples into the second-order space spanned by $\langle w, r, r', w' \rangle$ quadruples.

Thater et al. (2010) evaluated their model on the lexical substitution task where they showed improved performance over the model of Erk and Padó (2008) on ranking paraphrases for verbs.

In later work, Thater et al. (2011), extended their previous approach with a more focused re-weighting scheme for contextualising a lexeme. Instead of constructing first and second-order vector spaces for modelling the distributional semantics and syntactic selectional preferences of lexemes, Thater et al. (2011) only built a first-order vector space spanned by $\langle w, r, w' \rangle$ triples. Contextualisation is modelled by re-weighting individual dimensions on the basis of their current context.

For example for the verb phrase *catch fish*, with a direct object relation connecting *catch* with *fish*, only the $\langle r, w' \rangle$ tuple $\langle \texttt{dobj}, \textit{fish} \rangle$ would remain in the vector representation for *catch*. In order to avoid overly sparse representations, Thater et al. (2011) exploited the distributional similarity between *fish* and other nouns in the same syntactic role (i.e. other direct objects) to enrich the contextualised representation of *fish*.

Thater et al. (2011) evaluated their model on the lexical substitution task and achieved substantially improved performance in comparison to their previous approach and the selectional preferences based-model of Erk and Padó (2008), as well as other approaches by Erk and Pado (2010) and Dinu and Lapata (2010).

### 2.3.3 *Contextualisation via Latent Sense Modelling*

These approaches are based on the notion of matrix factorisation of the distributional space in order to obtain distributional word vector

23 Padó and Lapata (2007), and Erk and Padó (2008) remove the type label from their representation.

representations in terms of a distribution over senses instead of over context words. Contextualisation is subsequently modelled as a distribution over these latent senses that is modulated by the current context of a lexeme.

Dinu and Lapata (2010) introduced two models based on Latent Dirichlet Allocation (LDA) (Blei et al., 2003), and Non-negative Matrix Factorisation (NMF) (Lee and Seung, 2001), respectively. As Equation 2.21 shows, the goal of matrix factorisation is to approximate a given matrix $\mathbf{M} \in \mathbb{R}^{i \times j}$ by the product of two matrices of lower dimensionality, $\mathbf{W} \in \mathbb{R}^{i \times k}$ and $\mathbf{C} \in \mathbb{R}^{k \times j}$, where typically $k << i$ and $k << j$, on the basis of minimising an error criterion such as the Frobenius norm or the KL-divergence between the original matrix $\mathbf{M}$ and its factorisation into $\mathbf{W}$ and $\mathbf{C}$.

$$\mathbf{M}_{i \times j} \approx \mathbf{W}_{i \times k} \cdot \mathbf{C}_{k \times j} \tag{2.21}$$

Both matrix factorisation techniques are based on the same count-based distributional space. In the case of their LDA model, the words within a context window of the distributional model are treated as documents[24], and the senses are interpreted as topics. This leads to a factorisation of the original target word by context matrix $\mathbf{M}$ into the product of two smaller matrices $\mathbf{W}$ and $\mathbf{C}$, where $\mathbf{W}$ represents a target word by sense matrix and $\mathbf{C}$ a sense by context matrix.

The second model they introduced is based on NMF, which when used with KL-divergence as objective function, has a probabilistic interpretation (Gaussier and Goutte, 2005; Ding et al., 2008). The matrices, $\mathbf{M}$, $\mathbf{W}$ and $\mathbf{C}$, in the factorisation have the same interpretation as in the LDA based model where $\mathbf{M}$ is the original distributional semantic co-occurrence matrix, $\mathbf{W}$ is a target word by sense matrix, and $\mathbf{C}$ a sense by context matrix.

For both models, all words in the distributional space share the same $n$ senses globally — i.e. all words are comprised of a distribution over the same latent sense space. This distribution is accordingly modulated for a particular target word when contextualised with a specific context. This is in contrast to the approaches of Reisinger and Mooney (2010a) and Reisinger and Mooney (2010b), discussed in Section 2.3.1 above, where all senses of a lexeme are *local* to that lexeme and not shared globally for the whole vocabulary.

24 That is every possible sliding window of size $n$ over the whole corpus is interpreted as an individual document.

In their evaluation Dinu and Lapata (2010) found that both of their proposed models achieved comparable performance on an out-of-context evaluation on a word similarity task and an in-context evaluation on the lexical substitution task, with the NMF based model working slightly better than the LDA model. Both of their models substantially outperformed a count-based distributional semantic baseline.

Two generalisations of the models introduced by Dinu and Lapata (2010) have been proposed by Van de Cruys et al. (2011) and Ó Séaghdha and Korhonen (2011), respectively. Both approaches aim to integrate syntactic context from a dependency parse of a given sentence into their models.

The approach of Van de Cruys et al. (2011) is based on an extension of the NMF model of Dinu and Lapata (2010), where the syntactic dependency contexts are modelled as two additional matrices, $\mathbf{R}_w$, representing a target word by dependency relation matrix, and $\mathbf{R}_c$, representing context by dependency relation matrix, in addition to the standard target word by context matrix $\mathbf{M}$. NMF is subsequently applied in an interleaved fashion to these three matrices in the order of $\mathbf{R}_w \rightarrow \mathbf{M} \rightarrow \mathbf{R}_c$, where the result of the former is used to initialise the latter (Van de Cruys, 2008). Combining the bag-of-words contexts with the syntactic contexts is achieved in a straightforward way by a weighted linear combination, such as a simple pointwise average, of the two corresponding factorised word by context matrices.

Similarly, Ó Séaghdha and Korhonen (2011) extend the LDA based model of Dinu and Lapata (2010) by incorporating syntactic context, treating a target word at a given dependency node, together with its head and dependants, as a document for the LDA modelling procedure. When evaluated on the lexical substitution task, both Van de Cruys et al. (2011) and Ó Séaghdha and Korhonen (2011) found that incorporating syntactic context improves upon the pure bag-of-words approach of Dinu and Lapata (2010). Their results provide independent evidence that incorporating syntax is beneficial for modelling word meaning in context.

An alternative approach has been proposed by Moon and Erk (2013), who modelled word meaning in context as an undirected graphical model. Each lexeme is modelled as an observed node, representing the surface form of the word, and a hidden node, repres-

enting its contextualised meaning. Hidden nodes are modelled as distributions over paraphrases which take the role of latent senses of the given target word. Contextualisation is modelled as an inference problem with the aim of inferring a paraphrase distribution for every word in a given phrase or sentence. The graphical model allows for a probabilistic formulation, describing the interactions between paraphrase distributions of contextualised lexemes. The factors of the graphical model are learnt based on Maximum Likelihood Estimation from a given source corpus. Moon and Erk (2013) showed that their approach outperforms the comparable models of Dinu and Lapata (2010), Erk et al. (2010), and Thater et al. (2010) on the lexical substitution task.

## 2.4 INFERRING UNOBSERVED EVENTS

The need for mechanisms for inferring unobserved events in distributional representations arises from the simple reality of not having a large enough corpus to observe all plausible co-occurrence events between any two lexemes. While simple smoothing techniques such as add-1 smoothing are often regarded as "good enough" in text classification models because of their simplicity, they quickly exhibit their shortcomings[25] in more complex tasks such as word-sense disambiguation or language modelling.

The issue of data sparsity is particularly problematic for explicit count-based distributional semantic models, especially when combined with an intersective composition function. This is because composed distributional phrase representations become sparser with each composition operation, making it difficult to scale the approach beyond modelling short phrases and sentences (Polajnar et al., 2014b). Consequently, the need for a mechanism for distributional inference in count-based distributional semantic models becomes a necessity for constructing high-quality elementary and composed representations. The predominant approach to inferring unobserved events is based on leveraging the distributional neighbourhood by enriching elementary representations with information from distributionally similar terms. Mitchell and Lapata (2008) note that inferring knowledge by leveraging the distributional neighbourhood provides one way of integrating additional knowledge $K$ (see Equation 2.3) into

25 See Gale and Church (1994) for an overview of why add-1 smoothing is a problematic estimation technique.

the distributional composition function.

The earliest approaches of using distributional information for inferring unobserved co-occurrences date back to Essen and Steinbiss (1992), who use ideas based on an earlier model by Sugawara et al. (1985). The approach of Essen and Steinbiss (1992) is to estimate the contextual similarities of seen lexemes $w_i$ with the context of an unseen occurrence of $w'$ in bigrams $(w_i, w')$, in order to infer whether the unobserved lexeme $w'$ is similar to any of the observed occurrences. Essen and Steinbiss (1992) showed that their co-occurrence smoothing technique substantially improves a language model for speech processing.

The technique gained popularity in the NLP community through the work of Dagan et al. (1993) and Dagan et al. (1994), who applied it to word-sense disambiguation and language modelling, respectively. Unseen events are assigned the average pointwise mutual information score of their top $n$ distributionally most similar neighbours. Dagan et al. (1993) used a Jaccard based similarity measure to compare the two contextual distributions of two lexemes. They showed that their approach is significantly better at predicting association scores for two related lexemes than a simple frequency based baseline, leading to substantially improved results for a word-sense disambiguation component in a machine translation system.

Dagan et al. (1994) extended their earlier approach by embedding it into a fully probabilistic language modelling framework, that first allocates an appropriate amount of probability mass for unseen co-occurrence events, and subsequently re-distributes that mass to the distributionally most similar terms, based on relative entropy as similarity measure. They showed that by combining their similarity based smoothing model with a standard back-off smoothing strategy (Katz, 1987), they were able to achieve statistically significant improvements for a language model. The merit of their distributional inference approach has furthermore been confirmed in a controlled pseudo-disambiguation experiment by Dagan et al. (1997), where they showed that their similarity based approaches significantly outperform other smoothing techniques such as the back-off model of Katz (1987) used by itself.

Inferring unobserved events on the basis of nearest neighbour estimates has been a popular approach for mitigating the data sparsity problem. For example, Turney (2006) used a similar idea to increase the coverage of his Latent Relational Analysis model for estimating the similarity of semantic relations. A notably different approach has been proposed by Erk (2016) who argued that cognitively, similarity between two terms is based on the property overlap between the two corresponding concepts. Distributional inference should therefore be a mechanism for inferring properties of a known into an unknown concept. In a number of preliminary experiments, Erk (2016) showed that there is a linear relation between corpus-based distributional similarity and property overlap, as based on the McRae feature norms dataset (McRae et al., 2005), between two corresponding concepts.

### 2.4.1 *Distributional Inference for Composition*

An early approach to distributional composition, that also included a mechanism for distributional inference, has been proposed by Kintsch (2001), who focused on modelling intransitive verb phrases. The approach called *predication* is based on pointwise addition of distributional semantic vectors that aims to integrate additional knowledge into the composed representation of a subject-verb pair. This is achieved by choosing the $n$ most similar neighbours of the predicate verb, of which the top $k$, that are also most similar to the argument noun, are selected to enrich the final composed representation. Equation 2.22 formalises the composition function:

$$z = x + y + \sum_{n_i \in N} n_i \tag{2.22}$$

where $x$ and $y$ are the distributional semantic representations for the predicate and the argument, respectively, and $N$ is the set of neighbours most similar to $x$ and $y$.

In this model, both composition and inference are based on the union of the features of each constituent. This creates the danger of overflowing the distributional representations with noise from unrelated neighbours if the hyperparameters of the model — the number of neighbours $n$ and $k$, or alternatively the similarity thresholds — are not carefully tuned. The only discriminative aspect of the composition function is the constraint that any neighbours need to be similar to the predicate and the argument. However, this severely restricts

the usefulness of the algorithm for typed count-based distributional semantic models, where lexemes with different syntactic roles live in very different areas of the distributional space with the consequence of very little distributional commonality between them.

Kintsch (2001) showed the feasibility of his proposal on a small qualitative experiment, highlighting how the *predication* algorithm improves distributional similarity estimates between plausible and implausible verb phrases; however, he does not conduct a quantitative evaluation. Later work by Mitchell and Lapata (2008) and Mitchell and Lapata (2010), found the *predication* algorithm to be performing relatively poorly for adjective-noun, noun-noun and verb-noun compositions, whereas Utsumi (2009) and Utsumi (2012) found it to be competitive for modelling a different set of noun-noun compounds.

An algorithm similar to *predication*, called *comparison*, has been proposed by Utsumi (2009) for composing noun-noun compounds. Instead of retrieving the $n$ nearest neighbours of the predicate and then sub-selecting the $k$ most similar to the argument as in the *predication* algorithm, the *comparison* algorithm selects the top $n$ *common* neighbours of both constituents. Utsumi (2009) showed that his *comparison* algorithm outperforms a pointwise vector addition baseline as well as the *predication* algorithm for noun phrases that exhibit emergent meaning properties[26]. However, taken all noun-noun compounds in his dataset into account, the *predication* algorithm beats the *comparison* algorithm by a small margin.

The *comparison* algorithm can also be represented by Equation 2.22 above with the only difference being that the set $N$ of most similar neighbours has been constructed in a different way. The approach by Utsumi (2009) also models composition and inference as a feature union, thereby suffering from the same hyperparameter sensitivity problem as the approach of Kintsch (2001).

Further alternatives of the *predication* algorithm have been proposed by Utsumi (2012) in subsequent work, introducing a variant based on composition and inference by pointwise multiplication, as well as a variant based on composition by pointwise multiplication and inference based on pointwise addition. The latter variant is the most similar previously introduced approach to the distributional inference

26 For example, in the phrase *information gathering*, the meaning of *intelligence* emerges in the compound and is thus more than just the sum of its parts.

algorithm proposed in this thesis. Equation 2.23 formalises the two novel variants

$$z = x \odot y \odot \prod_{n_i \in N} n_i$$
$$z = x \odot y \odot \sum_{n_i \in N} n_i \tag{2.23}$$

where $x$ and $y$ are the constituent word vector representations, and $N$ is the set of nearest neighbours. The set of neighbours $N$ can be retrieved by the *predication* or the *comparison* algorithm. Utsumi (2012) furthermore tests two variants of Equation 2.22 where the geometric mean instead of the raw product is used. The performance of the newly introduced variants exhibited mixed results when evaluated on modelling noun-noun compounds in comparison to the *predication* algorithm, and simple additive and multiplicative baselines.

Utsumi (2012) addressed the problem of too little discriminatory power in the composition function when integrating distributional inference with composition, by modelling composition as an *intersective* operation and inference as a *unifying* operation. However, the mixed set of results suggest that the restrictive neighbour retrieval function, that requires the neighbours to be similar to both constituents, represents a bottleneck for achieving better performance. For example, given the noun phrase *bike race*, the *predication* and *comparison* algorithms might discard the lexeme *bicycle* as a neighbour of *bike*, because of its low similarity to *race*[27]. This results in missing a potentially large number of lexemes that could contribute a significant amount of plausible co-occurrence information to the representations. Furthermore, Utsumi (2012) only evaluated the algorithms on noun-noun compounds, leaving any other phrase types untested for.

Thater et al. (2011) proposed an alternative similarity-based approach for enriching elementary word representation during distributional composition[28]. Instead of probing the distributional space for similar words to add unobserved events, they leverage the distributional neighbourhood to *retain* distributional knowledge that has already been observed with the target word, but would be filtered

27 Indeed in an untyped VSM, similar to the ones used by Utsumi (2012), the lexeme *bicycle* is not among the top 100 neighbours for the lexeme *race*. This has the consequence that any co-occurrence information from *bicycle* would be neglected when composing the phrase *bike race* with the *comparison* or *predication* algorithm.

28 Or “conextualisation" in their terminology.

out during composition. For example, given some target word $x$ and some context word $y$, with the goal of creating a contextualised version of $x$, Thater et al. (2011) first calculate the nearest neighbours of the context word $y$. Subsequently all the neighbours of $y$ that have already been observed in some syntactic relation with the target word $x$ are retained in the contextualised representation of $x$. This has the effect of creating a richer contextualised representation and resulted in superior performance for paraphrase ranking and word-sense disambiguation in comparison to approaches proposed by Erk and Padó (2008) and Erk and Pado (2010) (see also § 2.3.1 and § 2.3.2 above).

# 3

# ANCHORED PACKED TREES

This chapter outlines the theory behind the Anchored Packed Trees framework, that has been published in Weir et al. (2016), focusing on the nature of elementary APT representations (§ 3.1) and how they are composed (§ 3.2) to form longer phrases. This chapter furthermore contributes a comparison between the Anchored Packed Trees framework and previously proposed models in the literature (§ 3.3).

The theoretical work has been due to the first three authors (David Weir, Julie Weeds & Jeremy Reffin), whereas my contributions have been the empirical work in Weir et al. (2016).

Anchored Packed Trees (APTS) are a framework for modelling distributional composition based on a typed distributional co-occurrence space. The APT represents a novel unified data structure which is able to capture the distributional semantics of phrases and full sentences in the same space as individual lexemes. The predominant approach to modelling distributional composition has been to apply a composition function to fixed meaning representations of individual lexemes. Rather than defining composition to be a post-hoc operation on top of a given static semantic space, APTS model distributional composition as an intrinsic component of the distributional framework. The meaning of individual lexemes and composed phrases is represented in the same shared space. The core of the proposal is a unified data structure that enables the precise alignment of the distributional features of the lexemes in a phrase that are bespoke to the current context. In the following I will outline the most important characteristics of the Anchored Packed Trees framework. A more detailed definition has been published in Weir et al. (2016).

APTS are a compositional distributional semantic model where composition is treated as a process of lexeme contextualisation. The effect of composition is the integration of the contextualised distributional knowledge of all lexemes in some phrase into a single unified data structure — the Anchored Packed Tree. The APT is an aggregation of all occurrences involving a given lexeme and uses a single representation to encode the distributional knowledge concerning that lexeme.

In a composed phrase, the contextualised meaning of each lexeme in the phrase is bespoke to the current context. This is achieved through a mechanism that reduces the contribution of distributional features unsuitable to the current context while increasing the contribution of compatible features.

For example, consider the adjective-noun phrase *white clothes*. Not everything that can be described as *white* is compatible with *clothes*, such as the distributional knowledge about *white* obtained from a sentence like *Some sort of machine hummed around the corner, breaking the silence with mindless white noise*. Things done to or with *white clothes* or *white shoes* are very different to things done to or with *white noise*. Contextualisation is therefore a mechanism for deriving a set of distributional features on which the representations for *white* and *clothes* agree on. This agreement process between the lexemes in a phrase distills their bespoke meaning in the current context while using a single representation per word type.

In general, a composition function based on the intersection of the features of two aligned lexeme representations in a phrase is more appropriate to contextualise the semantic content of the representations in a given phrase than a composition function based on feature union is. Thus, discussions involving the semantic contextualisation of two lexemes pre-suppose an intersective composition function.

## 3.1 ELEMENTARY APT REPRESENTATIONS

The structure of the distributional semantic space is a consequence of how composition has been specified in the framework but it is easier to first describe APTs as a distributional semantic model, and subsequently show how composition is defined. The distributional semantic space is built on the basis of typed[1] co-occurrences $\langle w, \tau, w' \rangle$, where $w$ and $w'$ are two co-occurring lexemes and $\tau$ denotes the dependency relation between the two. Following Padó and Lapata (2007), the string $\tau$ may represent a single dependency relation such as `amod`, denoting an adjectival modifier for some noun, however it may also represent a higher-order path such as `dobj.amod`, denoting the adjectival modifier of some noun that is the direct object of some verb. Furthermore, APTs include inverse dependency paths, denoting the relations from modifiers back to their respective heads.

1 The APT framework is agnostic to the concrete grammatical formalism used to build the model, however this work assumes relations between lexemes to be dependency types.

Throughout this work I will assume that any lexemes $w$ and $w'$ are elements from a finite vocabulary $V$, and all relations $r$ and their inverse counterparts $\overline{r}$ are from a finite set of dependency relations $R \cup \overline{R}$, such that for any given co-occurrence type $\tau = r_1 \cdot \ldots \cdot r_n$, each $r_i \in R \cup \overline{R}$ for $1 \leq i \leq n$. Furthermore the co-occurrence type $\tau$ is restricted to be in $\overline{R}^* R^*$ for all elementary, offset and composed APT representations. This means that the path $\tau$ between $w$ and $w'$ in general first travels up towards the root of the given dependency tree until an ancestor of $w'$ is reached. It subsequently travels down the tree until $w'$ is encountered.

Figure 3.1 shows 4 unaligned example dependency trees, where the extracted typed co-occurrence features for the lexeme *clean* from the tree depicted in Figure 3.1 (a) are shown in Table 3.1 below.

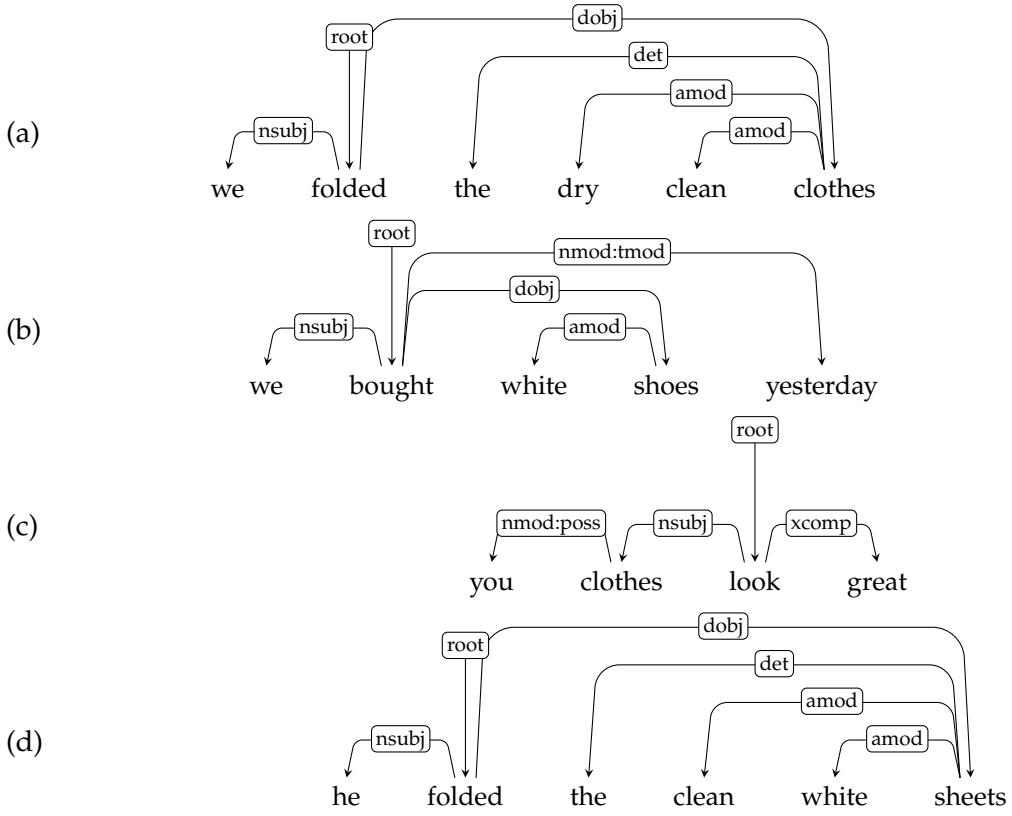


Figure 3.1: Unaligned example dependency trees.

Including the co-occurrence $\langle clean, \epsilon, clean \rangle$ results in a uniformity to the type system which is important for formulating distributional composition within the Anchored Packed Trees framework (Weir et al., 2016). The feature $\langle clean, \epsilon, dry \rangle$ highlights another important aspect of APT theory: canonicalisation of all co-occurrence types. Considering Figure 3.1 (a), the original path from the lexeme *clean* to *dry*

| Extracted Co-occurrence Features | |
|---|---|
| $\langle clean, \epsilon, clean \rangle$ | $\langle clean, \epsilon, dry \rangle$ |
| $\langle clean, \overline{\texttt{amod}}, clothes \rangle$ | $\langle clean, \overline{\texttt{amod}}\texttt{.det}, the \rangle$ |
| $\langle clean, \overline{\texttt{amod}}\texttt{.}\overline{\texttt{dobj}}, folded \rangle$ | $\langle clean, \overline{\texttt{amod}}\texttt{.}\overline{\texttt{dobj}}\texttt{.nsubj}, we \rangle$ |

Table 3.1: Typed co-occurrence features for the lexeme *clean*, extracted from the dependency tree shown in Figure 3.1 (a).

involves inversely travelling along the `amod` edge from *clean* to its head *clothes*, and from there, taking the forward `amod` edge to *dry*. The path between *clean* and *dry* would therefore be $\overline{\texttt{amod}}\texttt{.amod}$. However, through the canonicalisation of dependency paths, complementary adjacent edges are cancelled out. More formally:

$$\downarrow(\tau) = \begin{cases} \downarrow(\tau_1 \tau_2) & \text{if } \tau = \tau_1\, r \bar{r}\, \tau_2 \text{ or } \tau = \tau_1\, \bar{r} r\, \tau_2 \text{ for some } r \in R \\ \tau & \text{otherwise} \end{cases} \tag{3.1}$$

where $\downarrow(\tau)$ denotes the reduced co-occurrence type for some $\tau$. This results in the empty path $\epsilon$ in the co-occurrence triple $\langle clean, \epsilon, dry \rangle$ In the following, only reduced co-occurrence types are considered for any typed $\langle w, \tau, w' \rangle$ co-occurrence event.

*Constructing APTs*

Figure 3.2 illustrates the process of building APTs for the lexemes *white* and *clothes* from the example dependency trees in Figure 3.1. Step (1) in Figure 3.2 highlights how the path reduction procedure outlined above has placed — or "packed"[2] — the lexeme *dry* at the same node in the aligned tree as the adjective *clean*, which subsequently will hold other adjectives that modify the anchored lexeme *clothes*. As Figure 3.2 illustrates, the process of packing has the effect of losing the order information of lexemes with identical paths.

Step (2) shows how the third sentence is aligned with the first one, and (3) shows the aligned representation[3] for the APT anchored at the lexeme *white* for the second and last sentence of the example dependency trees shown in Figure 3.1. Weights associated with lexemes

2 The term "packed" in APTs refers to the process that merges words with identical paths into the same node as in the example for the two adjectives *dry* and *clean* in the sentence *we folded the dry clean clothes*. This notably differs from other usages of "packing" that refer to *packing* or *unpacking* individually quantified statements in logical form as used in Copestake and Herbelot (2012), among others.

3 The packing step that places the adjectival modifiers *clean* and *white* at the same node is identical as in (1) and has been omitted for brevity.

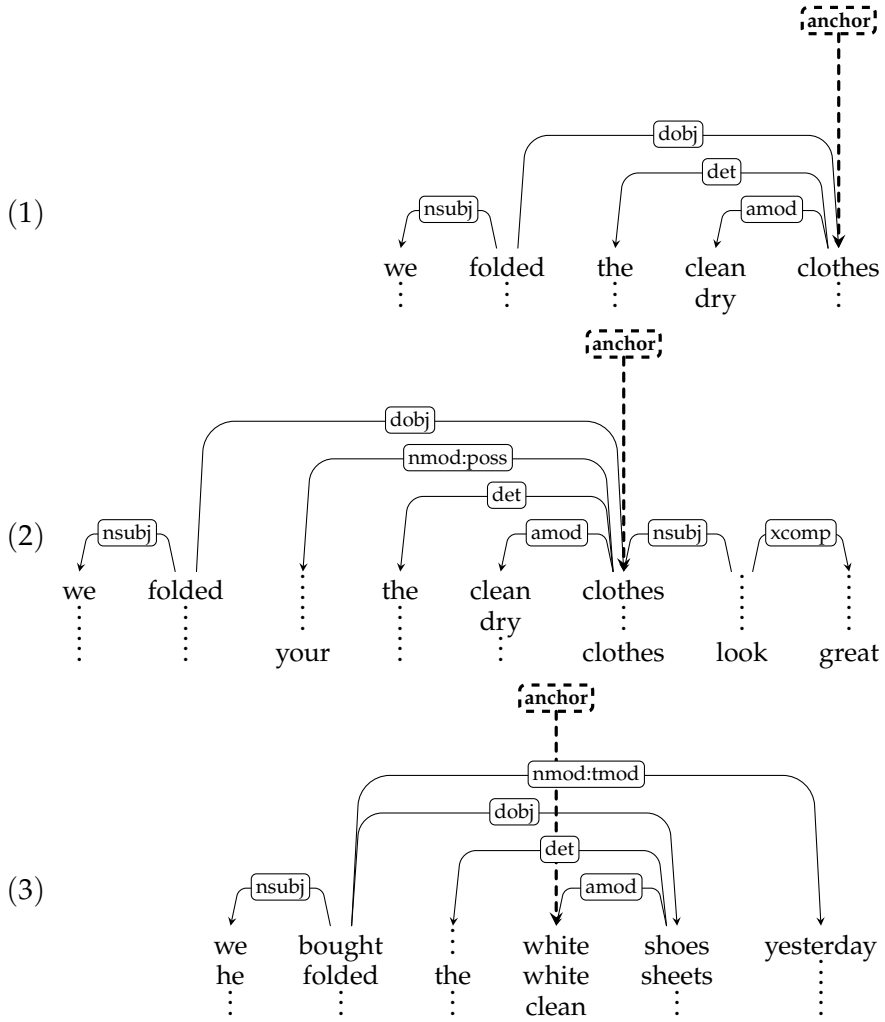


Figure 3.2: Alignment procedure of the dependency trees depicted in Figure 3.1. Tree (1) is the aligned representation of tree (a) in Figure 3.1, with the adjectival modifier *dry* merged into the same node as *clean*. Tree (2) is the final representation for the APT anchored at *clothes* and tree (3) is the aligned represenation of trees (b) and (c) of Figure 3.1 for the APT anchored at the adjective *white*.

are not shown in Figure 3.2, however their number of occurrences at a node is highlighted by having two occurrences of *clothes* in Figure 3.2 (2) and two occurrences of *white* in Figure 3.2 (3). The notion of an anchor in an APT generalises the anchor formulation of Padó and Lapata (2007) in their SVS model, as it represents the starting point for the paths for a whole APT structure, consisting of any number of aligned dependency trees, rather than an individual tree. The position of the anchor in an APT denotes the starting points of the paths. Moving the position of the anchor along some path in the APT is a

key component to modelling distributional composition within the Anchored Packed Trees framework as will be shown in Section 3.2.

The formulation of APTs as function (Weir et al., 2016) allows the retrieval of the weight associated with a specific co-occurrence event $\langle w, \tau, w' \rangle$ for APT $\mathbf{A}_w$, where $\mathbf{A}_w$ is the APT representation for the lexeme $w$, as $\mathbf{A}_w(\tau, w')$ as shown in Equation 3.2 below:

$$\mathbf{A}_w = f(w, \langle \tau, w' \rangle) \tag{3.2}$$

where $f$ represents some weighting function for the co-occurrence event $\langle w, \tau, w' \rangle$. For example, the number of occurrences of *white* at path $\epsilon$ in the APT in Figure 3.2 (3), denoted by $\mathbf{A}_{white}$, can be retrieved by $\mathbf{A}_{white}(\epsilon, white)$, resulting in 2. Equivalently, the number of occurrences of *shoes* at path $\overline{\texttt{amod}}$ can be retrieved as $\mathbf{A}_{white}(\overline{\texttt{amod}}, shoes)$, resulting in 1.

Given a large corpus, the elementary APTs for the lexemes *white* and *clothes* would give rise to the graph-like data structure in Figure 3.3.

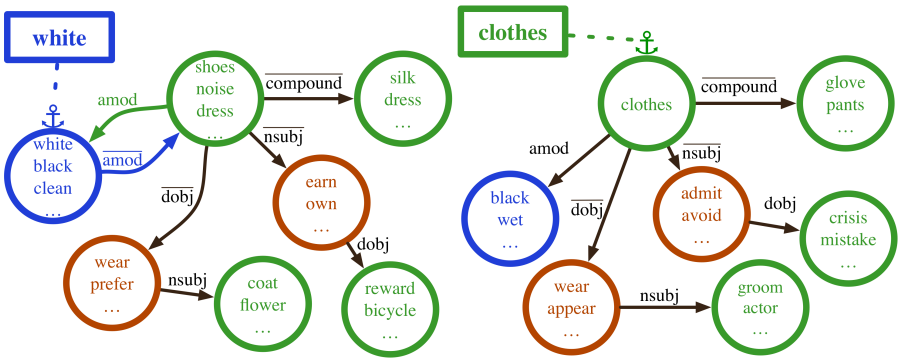


Figure 3.3: Structured distributional APT space. Different colours reflect different parts of speech. Boxes denote the lexeme at which the current APT is anchored. Circles represent nodes in the APT space, holding lexemes, and edges represent their relationship within the space.

All edges in the APTs are bi-directional as exemplified between the adjective node at which *white* appears (blue) and its corresponding head noun node at which *clothes* appears (green) for the APT anchored at *white* (see Figure 3.3, top left), however for reasons of readability, the figure only contains uni-directional edges.

*Vectorising APTs*

A vectorisation of the APT distributional space can be achieved by collecting all typed features up to some order for all lexemes in the space, and flattening them into a shared vector space such that any contextual dimension consists of a $\langle \tau, w' \rangle$ tuple. The whole space can subsequently be represented by a $\mathbf{M} \in \mathbb{R}^{|V| \times |C|}$ matrix, where $C$ represents the set of all $\langle \tau, w' \rangle$ contextual dimensions.

Formally, let $\mathbf{M} \in \mathbb{R}^{|V| \times |C|}$ be the adjacency matrix[4] of the APT lexicon where each row $i$ corresponds to a lexeme of the vocabulary $V$ and each column $j$ denotes a $\langle \tau, w' \rangle$ context tuple of the set of contexts $C$. Every co-occurrence event between a lexeme $w$ and a context $\langle \tau, w' \rangle$ is weighted by some score function $f$ as Equation 3.3 below shows, which gives rise to a matrix formulation of APTs that is equivalent to the functional definition of Equation 3.2.

$$\mathbf{M}_{i,j} = f(w, \langle \tau, w' \rangle) \tag{3.3}$$

In practice, the dependency path $\tau$ and the context word $w'$ are concatenated. For example, the concrete co-occurrence triple $\langle \textit{seagull}, \overline{\texttt{nsubj}}, \textit{landed} \rangle$ would be converted into the tuple $\langle \textit{seagull}, \overline{\texttt{nsubj}}\texttt{:}\textit{landed} \rangle$. This representation scheme follows Padó and Lapata (2007), Baroni and Zamparelli (2010), and Thater et al. (2010) by integrating the syntactic infomation as part of the context into the model.

A vectorised APT is denoted by $\vec{\mathbf{A}}$ and represents the $i^{\text{th}}$ row of the adjacency matrix $\mathbf{M}$ defined above, hence $\vec{\mathbf{A}} = \mathbf{M}_i$. For notational consistency with Weir et al. (2016) I will use the notation $\vec{\mathbf{A}}$ (rather than $\mathbf{M}_i$) to refer to a vectorised APT and explicitly denote any co-occurring context of the vectorised APT $\vec{\mathbf{A}}$ as just $\langle \tau, w' \rangle$, such that $\vec{\mathbf{A}}[\langle \tau, w' \rangle]$ refers to the specific co-occurrence between $\langle w, \tau, w' \rangle$.

The distributional similarity between two vectorised APTs[5], denoted as $\textsc{sim}(\vec{\mathbf{A}}_1, \vec{\mathbf{A}}_2)$, can be calculated with any distributional similarity measure such as cosine or euclidean distance as Equation 3.4 below shows.

4 APTs give rise to a weighted, directed and labelled graph as shown in Figure 3.3 from which the adjacency matrix can be obtained. Instead of representing an edge between two vertices by 0 or 1, the existence of an edge is weighted by the PMI score of the corresponding co-occurrence event associated with the two vertices and the labelled edge connecting them.

5 I.e. any two rows of $\mathbf{M}$.

$$\begin{aligned} \text{SIM}(\vec{\mathbf{A}}_1, \vec{\mathbf{A}}_2) &= \cos(\vec{\mathbf{A}}_1, \vec{\mathbf{A}}_2) \\ \text{SIM}(\vec{\mathbf{A}}_1, \vec{\mathbf{A}}_2) &= 1 - \|\vec{\mathbf{A}}_1 - \vec{\mathbf{A}}_2\| \end{aligned} \tag{3.4}$$

As the euclidean norm in the second line of Equation 3.4 denotes a *distance* rather than a *similarity*, it is necessary to express it as $1 - \|\vec{\mathbf{A}}_1 - \vec{\mathbf{A}}_2\|$.

In addition to the common practice of weighting all co-occurrence events by a lexical association score such as PMI, following Padó and Lapata (2007), it is furthermore possible to apply a path weighting function, reflecting the fact that path length is inversely proportional to the amount of direct distributional knowledge that a co-occurrence event provides about some lexeme. The weight of a given $\langle w, \tau, w' \rangle$ co-occurrence event for some vectorised APT $\vec{\mathbf{A}}$ can therefore be expressed as the product of a path weighting function $\phi(\tau, w)$ and a lexical association score $\mathrm{W}(w, \langle \tau, w' \rangle)$ as Equation 3.5 below shows[6].

$$\vec{\mathbf{A}}[\langle \tau, w' \rangle] = \phi(\tau, w)\, \mathrm{W}(w, \langle \tau, w' \rangle) \tag{3.5}$$

For example, using PMI as lexical association function would result in the formulation shown in Equation 3.6, which is following the PMI definition for a co-occurrence triple of Hindle (1990), that holds the path $\tau$ fixed:

$$\begin{aligned} p(w, w'; \tau) &= \frac{\#\langle w, \tau, w' \rangle}{\#\langle *, \tau, * \rangle} \\ p(w; \tau) &= \frac{\#\langle w, \tau, * \rangle}{\#\langle w * \tau, * \rangle} \\ p(w'; \tau) &= \frac{\#\langle *, \tau, w' \rangle}{\#\langle *, \tau, * \rangle} \end{aligned} \tag{3.6}$$

$$\mathrm{PMI}(w, w'; \tau) = \log \frac{p(w, w'; \tau)}{p(w; \tau) p(w'; \tau)}$$

where a "#" in front of a co-occurrence triple denotes the frequency of the event, and a "$*$" in any slot denotes the co-occurrences for any lex-

6 Hence, the function $f$ of Equation 3.3 is formed of the product of the lexical association function W and the path weighting function $\phi$.

eme in that slot. An alternative way to calculate PMI would be to treat the path $\tau$ as part of the context word $w'$ as used by Ó Séaghdha and Copestake (2007) and Kiela and Clark (2014). However in this work, I will consistently use the definition of Hindle (1990) as outlined in Equation 3.6 above.

Subsequently as shown in Equation 3.7, it is common to clamp any negative PMI values at 0 (Dagan et al., 1993; Niwa and Nitta, 1994), leading to Positive Pointwise Mutual Information (PPMI).

$$\text{PPMI}(w, w'; \tau) = \max(\text{PMI}(w, w'; \tau), 0) \tag{3.7}$$

## 3.2 COMPOSING APT REPRESENTATIONS

Before turning to distributional composition with APTs, the concept of so-called offset APTs and the procedure of offsetting needs to be introduced. Offsetting is the first step in deriving a set of agreeing features for the lexemes in a phrase and is governed by the *syntactic* context of a lexeme. Thus the process of alignment suppresses distributional features that do not fit into the current grammatical frame. The process of offsetting an APT representation according to its syntactic use is the key feature for aligning two or more elementary APT representations with different parts of speech.

### *Offset APTs*

By considering Figure 3.3 above, it can be seen that the APT structure can be traversed along the forward and inverse dependency links between the nodes. Furthermore, Figure 3.3 shows that words with different parts of speech live in very different feature spaces. For example the typed co-occurrence features of adjectives frequently start with $\overline{\texttt{amod}}$, the path connecting them to the nouns they modify (see Table 3.1 or Figure 3.3). Paths starting with $\overline{\texttt{amod}}$, however, cannot be observed for nouns or verbs as Figure 3.3 shows.

Thus, composing representations with different parts of speech without prior alignment would result in practically no feature overlap. This has the effect that any distributional commonalities between the representations cannot be leveraged. Therefore APTs require a mechanism to appropriately align the representations of the lexemes in a phrase. This process is called “offsetting" or “aligning", and causes a shift in anchor position along a given edge in the data structure. It is important to note that an anchor shift associated with offsetting does

not cause any structural changes *per se*. The only thing that is changing is the position of the anchor which denotes the starting point of the paths. For example by considering Figure 3.3, traversing along the $\overline{\texttt{amod}}$ edge from the adjective *white* to the adjacent noun node, results in a view in the APT for *white*, that constitutes a noun that has been modified by the adjective *white*. It therefore represents a "thing that can be white" structure. However, in accordance with the restriction that any dependency path $\tau$ in given an APT has to satisfy the constraint of being in the set $\overline{R}^* R^*$, not all co-occurrence events that are part of the elementary view of an APT are necessarily part of any of its offset views.

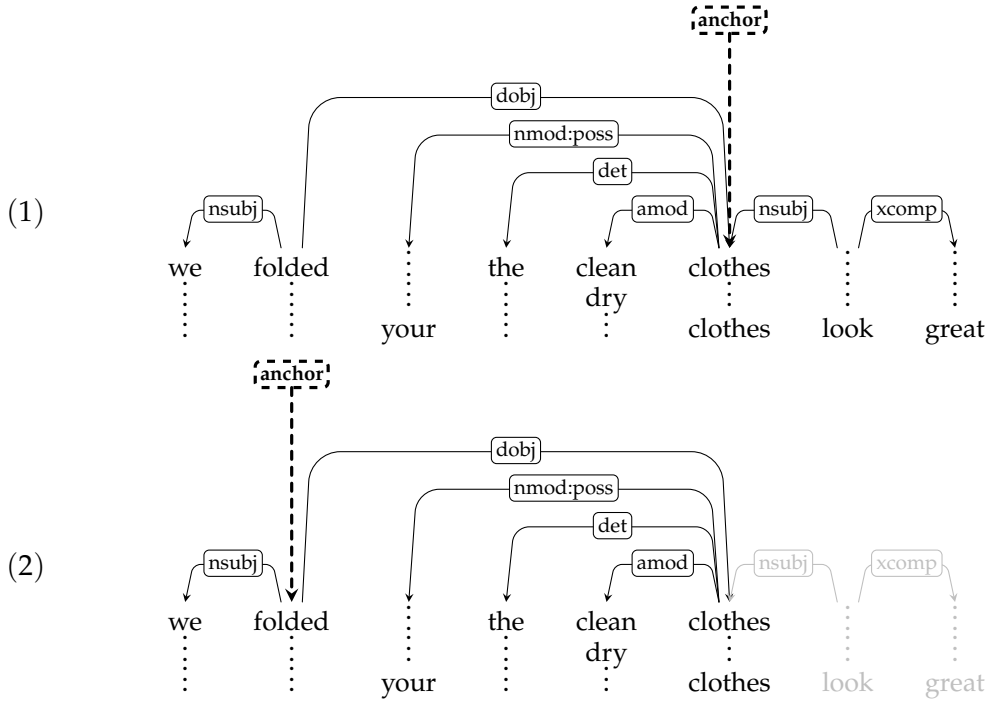


Figure 3.4: Offset procedure of the APT for the lexeme *clothes*. Tree (1) is the original APT representation for the lexeme *clothes* and tree (2) represents its `dobj` offset view $clothes^{\texttt{dobj}}$ which results in moving the anchor along the inverse direction of the `dobj` arc to the node where the lexeme *folded* occurs. The occurrences of *look* and *great* in faded text in tree (2) are not part of the representation of $clothes^{\texttt{dobj}}$ because they do not satisfy the constraint of being in $\overline{R}^* R^*$. This is because in order to reach the lexeme *look* (and subsequently *great*) from the anchored node at which the lexeme *folded* appears, one needs to traverse the `dobj` arc in a forward manner, however after which no more upwards traversals in the tree are allowed.

Figure 3.4 provides an example for this case where the lexemes *look* and *great* are part of the APT for the lexme *clothes* in tree (1), however are removed in tree (2) when moving the position of the anchor along

the inverse direction of the `dobj` arc from *clothes* to the node where the lexeme *folded* occurs.

In the case where an offset would be triggered along a path that has not been observed in the data and would therefore result in the anchor being placed at an empty node, the resulting APT structure would still adequately reflect the semantics of the resulting offset view. For example, as Figure 3.5 below shows, the `compound` offset view for the lexeme *clothes*, $clothes^{\texttt{compound}}$, does not have any observed co-occurrences for the path $\epsilon$, but is otherwise intact. If an offset would be triggered along a non-sensical path, such as *dobj* for an adjective, the resulting APT would still be non-empty, however it would not be expected to have any overlapping features with any other lexemes and would therefore be expected to yield distributional similarity scores of 0 in comparison with any other lexeme[7].

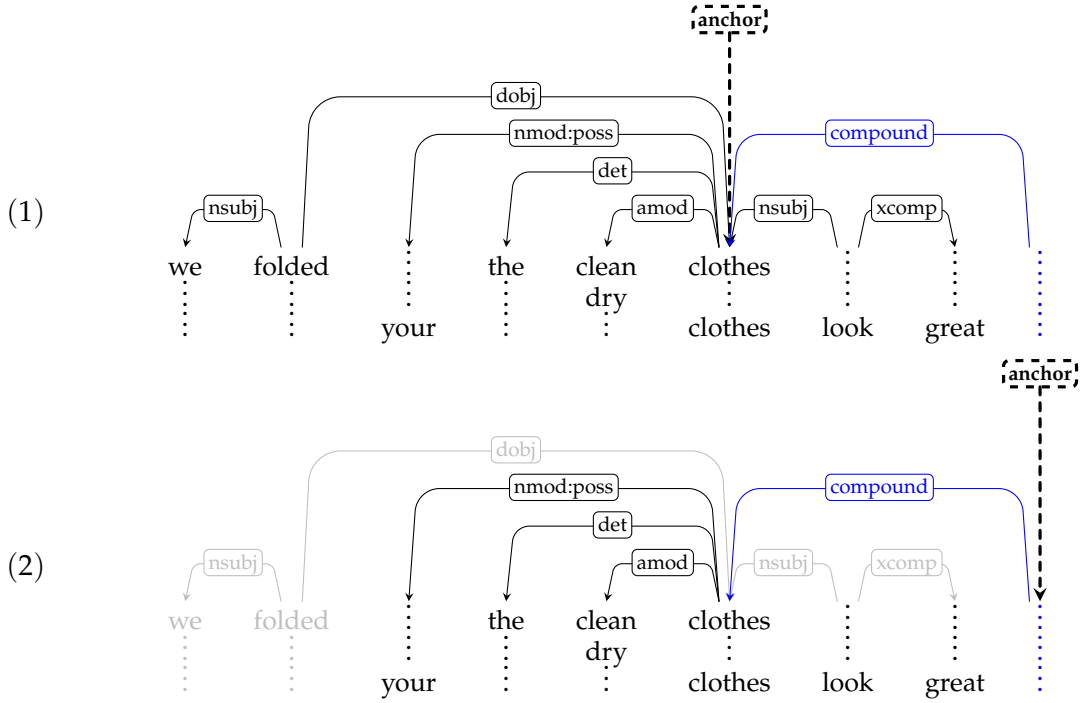


Figure 3.5: Offset procedure of the APT for the lexeme *clothes* along a path that has not been observed in the data. Tree (1) is the original APT representation for the lexeme *clothes* and tree (2) represents its `compound` offset view $clothes^{\texttt{compound}}$ which results in moving the anchor along the inverse direction of the `compound` arc. As in Figure 3.4 above, all paths not satisfying the constraint of being in $\overline{R}^* R^*$ have been removed.

7 Indeed, when tested with an order 2 APT space, derived from the BNC, comparisons between `dobj` offset views of adjectives such as *ancient*, *blonde*, *boring*, or *new* to any other adjectives, nouns or verbs alike, all yield a cosine similarity score of 0.

*Distributional Composition with APTs*

The lexeme that requires offsetting in a concrete phrase, and the path by which the lexeme needs to be offset is determined by the dependency tree of that phrase. For example, given the phrase *white clothes* with its associated dependency tree in Figure 3.6, the offset is carried out for the dependent[8] of the phrase, in the inverse direction of the dependency relation connecting it to its head.

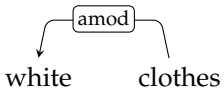


Figure 3.6: Dependency tree for the adjective-noun phrase *white clothes*.

For the adjective-noun phrase *white clothes* this means offsetting the modifier *white* by `amod`. The offset path `amod` results from the observation that the lexeme *white* is connected to its head via the relation $\overline{\texttt{amod}}$ and the inverse of $\overline{\texttt{amod}}$ is just `amod`. Hence traversing an edge, mathematically, is a process of path reduction as formulated in Equation 3.1 above, leading to the reduction $\downarrow(\texttt{amod}.\overline{\texttt{amod}}) = \epsilon$. This reduction reflects the fact that the data structure now provides a noun view of the adjective *white*.

Table 3.2 lists a number of example typed co-occurrence features for *white*, its offset noun view $white^{\texttt{amod}}$ and the noun *clothes*, highlighting the non-overlapping feature spaces between *white* and *clothes*, and the alignment between $white^{\texttt{amod}}$ and *clothes*, once the offset has been carried out for *white*.

| ***white*** | $\boldsymbol{white^{\texttt{amod}}}$ | ***clothes*** |
|---|---|---|
| `:`*clean* | `amod:`*clean* | `amod:`*wet* |
| $\overline{\texttt{amod}}$`:`*shoes* | `:`*shoes* | `:`*clothes* |
| $\overline{\texttt{amod}}.\overline{\texttt{dobj}}$`:`*wear* | $\overline{\texttt{dobj}}$`:`*wear* | $\overline{\texttt{dobj}}$`:`*wear* |
| $\overline{\texttt{amod}}.\overline{\texttt{dobj}}$`.nsubj:`*coat* | $\overline{\texttt{dobj}}$`.nsubj:`*coat* | $\overline{\texttt{dobj}}$`.nsubj:`*actor* |

Table 3.2: Sample of vectorised features for the APTs shown in Figure 3.3. Offsetting *white* by `amod` creates an offset view, $white^{\texttt{amod}}$, representing a noun, and has the consequence of aligning its feature space with *clothes*.

8 Alternatively, one could offset the head of a phrase, thereby creating an adjective view of the noun *clothes*. Either alignment approach produces the same resulting APT when the aligned elementary APTs are composed. For convenience Weir et al. (2016) generally illustrate the process by offsetting the dependent in a relation and this convention is followed here.

Once the typed distributional co-occurrence features of the lexemes in a phrase are appropriately aligned, composition can be carried out by merging the respective aligned nodes and their associated feature weights. More formally, Equation 3.8 below shows that composition is a function $f$ between $n$ aligned APTs, where $f$ is a pointwise arithmetic function such as min, max, addition or multiplication, among others, that combines the feature weights associated with their respective aligned nodes.

$$\bigsqcup_{Y}\{\mathbf{A}_1,\ldots,\mathbf{A}_n\}(\tau,w') = f_{1\leq i\leq n}[\mathbf{A}_i(\tau,w')] \tag{3.8}$$

The type of the merge function $\bigsqcup$, determined by Y, can be feature *intersection*, *union*, or any other merging operation. In this thesis I will restrict myself to either merging by *intersection*, denoted by $\bigsqcup_{\text{INT}}$, or merging by *union*, denoted by $\bigsqcup_{\text{UNI}}$. I will use pointwise addition[9] for combining the feature weights of two aligned APT nodes throughout this thesis and will use the terminology *composition by intersection* or *composition by union* as shorthand for referring to "merging aligned nodes by feature intersection/union, and combining their associated weights by pointwise addition". The use of pointwise addition to combine aligned APT nodes corresponds to multiplying the associated probability distributions for each node due to the use of log in PPMI (Ganesalingam and Herbelot, 2013). Furthermore, distributional composition between two or more APTs is always assumed to happen between *aligned* representations.

The composition function is responsible for integrating the distributional knowledge of the aligned lexemes in a phrase, and is governed by the *semantic* context of the lexemes in that phrase. Thus the distributional content of a composed phrase is determined by the alignment process, which contextualises the lexemes syntactically, and by composing them, which contextualises them semantically. This leads to the interpretation of distributional composition as a process of lexeme contextualisation.

Figure 3.7 illustrates the composition process for *white clothes*. First, as denoted by the red dashed arrow, the adjective *white* is aligned with its head *clothes* by offsetting it into the noun node, and subsequently the two aligned representations can be composed by intersection (bottom left) or union (bottom right). Notably, composition by intersection creates a much sparser representation for the whole

9 In preliminary experiments I found that pointwise addition was consistently among the top performing functions.

phrase than composition by union, however has the potential of more accurately reflecting the semantics of the phrase due to filtering features that are not compatible with both lexemes. On the downside, composition by intersection has the effect of making a sparse representation even sparser, therefore requiring a mechanism to ease that sparsity effect.

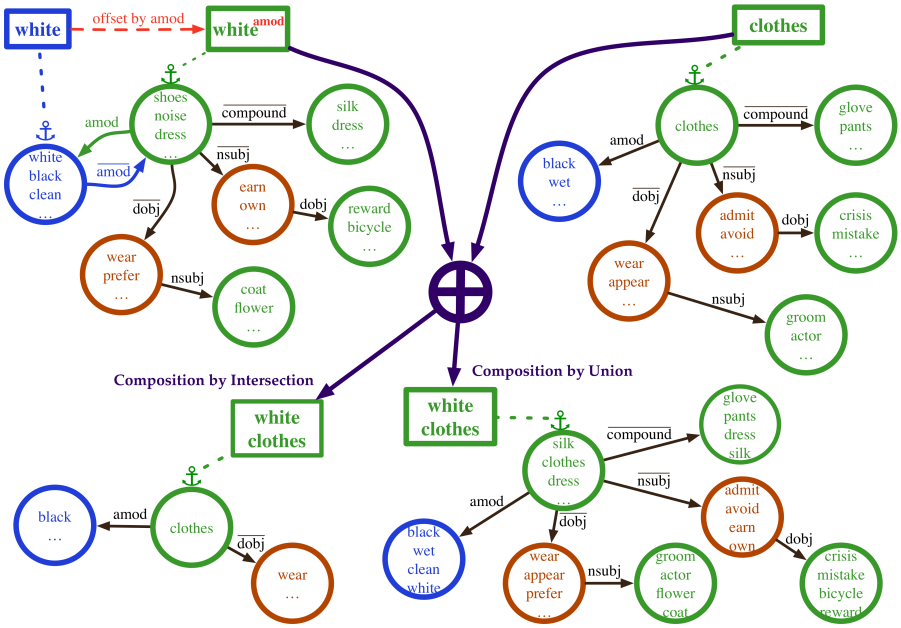


Figure 3.7: Distributional composition of two aligned APT representations by intersection (bottom left) or union (bottom right).

## 3.3 RELATION OF ANCHORED PACKED TREES TO PREVIOUS MODELS

Anchored Packed Trees bear some amount of resemblance to a number of previously proposed models. In the following I will discuss their similarities as well as their characteristic differences.

### 3.3.1 *Relation to Padó and Lapata (2007)*

As discussed in Section 2.1.2, Padó and Lapata (2007) proposed[10] a structured vector space (SVS) model on the basis of a dependency parsed corpus. The most important differences between the SVS model and APTs are that the former is purely a distributional

10 While originally published in an earlier paper (Padó and Lapata, 2003), the publication Padó and Lapata (2007) represents a more formal and exhaustive description of the model.

model and does not provide a theory of how composition should work within the distributional space. Secondly, APTs define a completely novel data structure, where elementary representations, only when *instantiated* as vectors, are similar to the structured vector space model of Padó and Lapata (2007).

Leaving the compositional component of APTs aside and assuming a vector based instantiation of the APT space, the models do bear a number of similarities. For example, the notion of an anchor is adopted in Anchored Packed Trees, however instead of only referring to the current starting point in a given dependency tree, its notion is extended to refer to the current starting point in an APT structure, which aggregates and aligns any number of dependency trees. Furthermore, like in the SVS model, APTs adopt the notion of a path weighting function in combination with a lexical association score for a co-occurrence event. APTs include inverse and higher-order dependencies when collecting co-occurrence events, however unlike Padó and Lapata (2007) who state that all paths in a $\langle w, \tau, w' \rangle$ co-occurrence triple are undirected, APT paths are labelled and directed[11]. Most notably, Padó and Lapata (2007) remove the type label from the co-occurrence events in order to reduce the sparsity of their model while in APTs all co-occurrences remain labelled[12].

### 3.3.2 *Relation to Baroni and Lenci (2010)*

Much like the structured vector space model of Padó and Lapata (2007), the Distributional Memory (DM) framework of Baroni and Lenci (2010) does not provide a theory of how composition can be modelled within the distributional space, whereas APTs are built around the concept of composition. Nonetheless, when focused on the distributional aspect of both models, the Distributional Memory framework and APTs exhibit a number of similarities. Firstly, the APT lexicon could be represented as a tensor by encoding the type of the relation in a co-occurrence event in its own dimension. The procedure of offsetting could be expressed in terms of matrix-matrix multiplication and distributional composition can also be expressed

11 While all paths in any APT are bi-directional, they cannot be reduced to undirected edges as the direction and label of a path are crucial for correctly offsetting and composing APTs.

12 The difference between *typed* and *labelled* is that a typed model follows some syntactic pattern when collecting the co-occurrences from a given source corpus as described in Section 2.1.2. Labelling refers to the fact that the typed co-occurrences retain their path information (the label) when building the co-occurrence space.

as functions between two matrices. Secondly, when the DM tensor is matricised by encoding the type of the $\langle w, r, w' \rangle$ co-occurrence event as part of the contextual dimension, $\langle w, r\, w' \rangle$, it yields the same matrix base representation[13] as when the APT lexicon is vectorised. While the representations of individual lexemes can be regarded as identical between the DM model and APTS, the former does not provide any machinery for modelling distributional composition. For example, it is unclear how lexemes with different grammatical roles could be composed in the Distributional Memory framework.

### 3.3.3 *Relation to Erk and Padó (2008)*

While not explicitly defining a theory of distributional composition, Erk and Padó (2008) proposed a model for representing the contextualised meaning of a word. A single lexeme is represented by a lexical vector for the given lexeme itself and an additional set of vectors expressing the selectional preferences of the given lexeme (also see Section 2.3.2).

Lexical vectors are either represented as standard untyped distributional semantic vectors, or as typed vectors by following the SVS model of Padó and Lapata (2007). The number of selectional preference vectors for a given lexeme is determined by the number of incoming and outgoing dependency edges that are adjacent to the given lexeme. A selectional preference vector is subsequently created for each dependency path and is represented as a weighted average of all lexical vectors that appear as $w'$ in a specific $\langle w, \tau, w' \rangle$ co-occurrence event for a fixed lexeme $w$ and dependency path $\tau$.

For example, the selectional preference vector for the relation `dobj` of the verb *catch*, is the weighted average of all word vector representations that co-occur as the direct object of *catch* across the given corpus. The weight is determined by the frequency of the specific co-occurrence event (Erk and Padó, 2008). The meaning of some phrase $w\, w'$ is then defined as the tuple of vectors for the meaning of $w$ in the context of $w'$ and the meaning of $w'$ in the context of $w$, where the contextualised representations of $w$ and $w'$ are subsequently not composed.

The major similarity to Anchored Packed Trees is an explicit mechanism to contextualise the meaning of a lexeme; however APTS provide a more flexible and general mechanism for modelling word

13 Assuming the typing in DM and in APTS are equivalent

meaning in context and distributional composition. The elementary lexical representations in the APT framework already encode the selectional preferences of a given lexeme due to explicitly modelling and retaining the grammatical type of all co-occurrences in the distributional space. Hence, there is no need for separate representations for modelling the lexical meaning and the selectional preferences of a lexeme.

Explicitly contextualising the selectional preferences of a lexeme in a given phrase is achieved through offsetting one of the lexemes in the dependency relation. Due to following the direction of the dependency tree of a phrase, there is no need for reciprocal contextualisation as the dependent in a relation is adapted to its phrasal head, which reflects its usage in the context of the head lexeme. Composition of the offset representation with the lexical representation of the head lexeme is subsequently carried out to integrate the combined meaning of the lexemes into a single representation.

While APTs and the model of Erk and Padó (2008) provide alternative ways of expressing the contextualised meaning of a word, Erk and Padó (2008) do not provide a way to represent the meaning of a phrase as a whole.

### 3.3.4 *Relation to Thater et al. (2010)*

The model of Thater et al. (2010) arguably shares the most commonalities with the Anchored Packed Trees framework, where globally, APTs can be seen as a formalised generalisation that subsumes the model proposed by Thater et al. (2010).

Thater et al. (2010) created a typed distributional vector space and observe, by explicitly retaining the type structure, that lexemes with different parts of speech, such as verbs and nouns, have a very different set of features and cannot be compared in a straightforward manner. For example, the distributional features of verbs frequently start with `dobj` or `nsubj`, denoting the direct objects and subjects that the verb co-occurs with. Nouns on the other hand will have have features starting with `amod`, denoting their adjectival modifiers or $\overline{\texttt{dobj}}$ and $\overline{\texttt{nsubj}}$ denoting the inverse relation to the verbs for which they occur as direct object or subject. In order to address that issue, Thater et al. (2010), created a *second* vector space that captures the second-order co-occurrences of lexemes.

This is achieved through the use of a so-called “lifting-map" which creates a second-order vector from a first-order vector, by prepending every typed feature of the given first-order vector with the dependency relation in which the lexemes co-occur in the given phrase. For example, if the first-order noun vector has the typed distributional features $\overline{\texttt{dobj}}:\textit{catch}$ and $\texttt{compound}:\textit{fish}$, and is the direct object of the verb *eat*, then the lifting operation creates the second-order typed features $\texttt{dobj}.\overline{\texttt{dobj}}:\textit{catch}$ and $\texttt{dobj}.\texttt{compound}:\textit{fish}$ from the given first-order representation.

The use of a “lifting-map" is very similar to the concept of offsetting in Apts, and they are indeed operations that are yielding equivalent outcomes: an alignment between two or more representations with different parts of speech. However there are a number of differences between the two alignment methods. Firstly, the lifting-map changes the first-order representation of a given lexeme itself. In Apts on the other hand, offsetting does not cause a structural change, but only a shift in the anchor position in the given Apt data structure that changes the starting points of the paths. Secondly, due to using separate vector spaces for first- and second-order representations, it is unclear how any higher-order liftings could be performed. In Apts higher-order offsets are supported due to the use of a unified data structure for distributional features of any order. Lastly, paths are not canonicalised, which has the consequence of, for example verbs, having paths such as $\texttt{dobj}.\overline{\texttt{dobj}}$ to other verbs. In Apts such paths would be reduced to an empty path by the reduction $\downarrow(\texttt{dobj}.\overline{\texttt{dobj}}) = \epsilon$. Any such verbs would subsequently be merged into the same node, which is enabled by the use of a unified data structure. Therefore, the offsetting procedure in Anchored Packed Trees represents a formalised and generalised variant of the lifting operation proposed by Thater et al. (2010).

By using an intersective composition function (pointwise multiplication) to combine the lifted first-order vector $w$ with the second-order vector $w'$ in the phrase $w\,w'$, the lifted representation for $w$ acts as a filter on $w'$, retaining non-zero values for only those features that have been observed for both $w$ and $w'$. Pointwise multiplication conflates the merging of aligned features and combination of their associated weights into a single operation. Instead, composition in Apts decouples the two steps by first merging aligned nodes by feature intersection, union, or a variant of the two, and subsequently combines the feature weights of the merged nodes by any arithmetic operation.

This decoupling of aligning and merging adds a layer of flexibility to the composition operation.

### 3.3.5 *Relation to Thater et al. (2011)*

Thater et al. (2011) proposed a simplified model of the approach published in Thater et al. (2010), which notably removes the modelling of second-order co-occurrences, while representing individual lexemes in a typed first-order vector space. The elimination of the second-order vector space abolishes the need for the lifting-map operation to align two lexemes with different parts of speech. Through the simplification of removing the requirement for aligning the representations of lexemes with different parts of speech., the model of Thater et al. (2011) is less similar to the Anchored Packed Trees framework.

Thater et al. (2011) compare two different contextualisation mechanisms. Given the phrase *catch fish*, where *fish* is the direct object of *catch*, the first contextualisation method would simply remove all features from the vector representations of *catch* and *fish*, apart from $\texttt{dobj}:\mathit{fish}$ for *catch* and $\overline{\texttt{dobj}}:\mathit{catch}$ for *fish*. Despite the simplicity and high degree of sparsity induced by this operation Thater et al. (2011) found the method to be working surprisingly well for ranking paraphrases.

The second method uses a mechanism to retain features of distributionally similar lexemes in the corresponding vectors, such that for example, the feature $\overline{\texttt{dobj}}:\mathit{trout}$ would be retained in the vector for *fish*, where *trout* and *fish* are assumed to be distributionally similar. This approach is not similar to the contextualisation process in APTs *per se*, however it is indeed similar to the use of the standard distributional inference algorithm introduced in Chapter 5.

# 4

# CHARACTERISING ELEMENTARY AND COMPOSED APT REPRESENTATIONS

This chapter analyses the Anchored Packed Trees framework as a distributional semantic and compositional distributional semantic model by characterising the distributional neighbourhood of elementary, offset and composed APT representations. The performance of APTs is evaluated on a number of word similarity tasks, as well as a short phrase composition benchmark dataset. This chapter contains an expanded version of the empirical work presented in Weir et al. (2016). The contributions of this chapter are:

- An empirical evaluation of the APT theory on word similarity and short phrase composition tasks.
- A hyperparameter sensitivity analysis of elementary and composed APT representations.
- Practical recommendations for favourable hyperparameter settings.
- An analysis of the effect of different hyperparameter settings on the semantic APT space.
- A qualitative and quantitative assessment of the distributional neighbourhood that elementary, offset and composed APT representations give rise to.

The chapter is structured as follows: the preprocessing pipeline (§ 4.1.1), evaluation methodology (§ 4.1.2) and datasets (§ 4.1.4) are summarised in Section 4.1. Section 4.2 reports the practical evaluation on word and phrase similarity tasks, introducing the investigated hyperparameters in Section 4.2.1, presenting the large-scale hyperparameter sensitivity analysis in Section 4.2.3, and providing practical recommendations in Section 4.2.4. Subsequently, Section 4.3 characterises the distributional space of elementary (§ 4.3.1), offset (§ 4.3.2), and composed (§ 4.3.3) APT representations.

## 4.1 PREPROCESSING, DATA AND EVALUATION

### 4.1.1 *Preprocessing Pipeline and Source Corpus*

The empirical work in this chapter predominantly relies on representations derived from the widely used British National Corpus (BNC) (Burnard, 2007), containing ≈100 million words. This primarily has practical reasons as the BNC is big enough to enable the creation of word representations of good quality, however is still compact enough to allow the exploration of a large number of model parameters while keeping the computational load associated with creating APT spaces manageable. Furthermore, the primary aim of this chapter is not to achieve state-of-the-art results on any of the datasets, but to provide a practical evaluation of the theory and to explore and quantify the impact of different parameterisations on elementary and composed APT representations.

The BNC was preprocessed with the Stanford NLP Toolkit[1] (Manning et al., 2014), using the default models for lemmatisation and part of speech tagging, and the standard version of the shift-reduce parser for dependency parsing. All lexemes have been lowercased before creating the APT spaces[2]. Any $\langle w, \tau, w' \rangle$ co-occurrence triple with a frequency of less than 10 has been discarded, and any APT representations with fewer than 50 non-zero features have been disposed of. All numbers[3] have been replaced by a "*#num*" token and all URLs[4] have been replaced by a "*#url*"[5] token prior to creating APTs. The motivation for this fixed set of hyperparameters is due to preliminary experiments, where an exhaustive investigation of the impact of simple transformations such as lowercasing or text normalisation are out of scope of this work. The adoption of a set of "good defaults" allows to shift the focus on parameters that are more idiosyncratic to APTs.

1 Version 3.5.2

2 Lowercasing has been shown to provide a small benefit in terms of processing time in preliminary experiments, while the results did not differ beyond the 4th or 5th significant digit.

3 Anything part of speech tagged as a cardinal number (CD), which can include lexemes such as *third*.

4 URLs have been identified on the basis of a regular expression.

5 While most of the texts in the BNC pre-date the world wide web, there are only very few occurrences of URLs (a search for string patterns ending with `*.com` yields ≈ 80 results), hence this normalisation step is expected to have relatively little impact, but is nonetheless applied to reduce the contextual noise of URLs for their surrounding words.

### 4.1.2 *Evaluation*

Evaluating compositional distributional semantic models is an active, and relatively recent, area of research. Unlike part of speech tagging or dependency parsing, there neither is a widely adopted benchmark such as the Penn Treebank (Marcus et al., 1993), nor a general agreement of what the best way of evaluating a given model actually is (Batchkarov, 2016).

A broad categorisation can be made between *intrinsic* and *extrinsic* evaluation strategies. An intrinsic evaluation generally involves a comparison to human judgements, such as in word similarity or phrase similarity tasks. An extrinsic evaluation on the other hand, is carried out on downstream tasks such as dependency parsing, sentiment analysis or recognising textual entailment.

An intrinsic evaluation such as word and phrase similarity tasks only provide part of the performance picture of a compositional distributional model as they focus on relatively frequent lexemes and ignore the wider context in which a lexeme might occur. However, these tasks are a convenient way for analysing the impact of a large number of parameters on the distributional space. If a model performs badly on the "easier" tasks in an intrinsic evaluation, it is unlikely that it will perform substantially better than other models in a more difficult task. Hence, intrinsic evaluations can provide important insights about the quality of a model.

### 4.1.3 *Statistical Significance*

The correlation coefficients of two models, in comparison to the same set of target variables, in general, are not independent. Therefore, the method of Steiger (1980) tests whether the similarity estimates of two different models are statistically different in comparison to the human judgements. One model is judged to perform statistically significantly better than another model if the method of Steiger (1980) judges the similarity distributions obtained from the respective models, in comparison to the human judgements, to be statistically different with a level of confidence of $p < 0.05$.

Testing for statistical significance has long been neglected when comparing distributional semantic models on word or phrase similarity datasets. The first work to perform a systematic and large scale analysis for statistical significance has been conducted by Shalaby

and Zadrozny (2015), who found that even seemingly large differences in correlation between two models are frequently not statistically significant due to the small size of some of the datasets such as MC30 Miller and Charles (1991) and RG65 Rubenstein and Goodenough (1965). Due to their small size and inconsistent behaviour with regards to noise as identified by Batchkarov et al. (2016), I refrain from an evaluation on these two datasets. Shalaby and Zadrozny (2015) furthermore recommend the usage of the statistical test of Steiger (1980) for its above mentioned properties. The method of Steiger (1980) has subsequently been adopted by Rastogi et al. (2015), Wieting et al. (2015) and Faruqui et al. (2016). Due to its popularity in recent work, I am also adopting the method of Steiger (1980) in this thesis. All statistical tests in this thesis use the two-tailed variant of the recommended test. The statistical test will generally be referred to as "the method of Steiger (1980)" throughout.

The null hypothesis for all tests is that the distributions of similarity estimates of two models, in comparison to the human judgements, are statistically equivalent — i.e. that the performance of the two models does not significantly differ. The null hypothesis will be rejected if a statistically significant difference between two models can be determined on the basis of the method of Steiger (1980). Rejecting the null hypothesis will not be explicitly mentioned when presenting the results. Repeated trials are not corrected for in the results, which means that each model instantiation with a specific set of hyperparameters is treated as a model in its own right rather than a permutation of some base model.

### 4.1.4 *Datasets*

For analysing the impact of the hyperparameter settings of different APT spaces I am using 3 commonly used word similarity datasets, WordSim-353 (Finkelstein et al., 2001), MEN (Bruni et al., 2014), SimLex-999 (Hill et al., 2015). Distributional composition is evaluated on the popular short phrase composition benchmark of Mitchell and Lapata (2010), containing adjective-noun, noun-noun, and verb-object pairs. The BLESS dataset (Baroni and Lenci, 2011) is used for characterising the distributional space of elementary, offset and composed APT representations.

*WordSim-353*

The WordSim-353 (WS353) dataset (Finkelstein et al., 2001) contains 353[6] word pairs, each annotated with 13-16 human similarity judgements on a scale of 0 (completely dissimilar) to 10 (maximally similar). The task is to compare the averaged human judgements to the distributional similarity between the two lexemes in the model by calculating Spearman's $\rho$ between the two sets of similarity judgements. Notably, the WordSim-353 dataset contains a mix of different semantic relations such synonymy, co-hyponymy, hypernymy, meronymy and topical relatedness, without clear annotation guidelines on what constitutes a similar pair (Batchkarov et al., 2016).

Therefore, I will use the partitioned variant of WordSim-353 by Agirre et al. (2009), who split the dataset into a *similarity* and *relatedness* subset. The name of the subsplit will be added in parentheses, as in *WordSim-353 (similarity)* or *WS353 (rel)*[7]. The similarity subset contains all pairs categorised as synonyms, antonyms, hypernym-hyponyms or identical, together with all dissimilar pairs that have an average human similarity rating of less than or equal to 5. The relatedness subset is the union of the same dissimilar pairs as the similarity subset, together with all pairs categorised as meronym-holonyms. After re-partitioning the dataset, the similarity subset contains 203 word pairs, and the relatedness subset contains 252 word pairs.

*MEN*

The MEN dataset (Bruni et al., 2014) consists of 3000 word pairs containing adjectives, nouns and verbs[8], and its annotation guidelines favour semantic relatedness. The word pairs have been created in a semi-automatic way on the basis of their distributional similarity and their frequency in a reference corpus. Bruni et al. (2014) used Amazon Mechanical Turk to collect the annotations, where the annotation task has been set up as a comparison between two word pairs, requiring annotators to select which of the two pairs shows a greater degree of relatedness. Each word pair has been rated 50 times and its score is

6 While in total it contains 353 pairs, the dataset contains only 352 *distinct* pairs as the pair *money - cash* occurs twice, with slightly different average similarity ratings (9.08 vs. 9.15).

7 Representing the abbreviation for *WordSime-353 (relatedness)*

8 Word pairs are not restricted to the same part of speech but are frequently across different parts of speech.

computed as the number of times the word pair has been identified as more related in a comparison, divided by 50. The task is to compare the human annotations to the similarity judgements of the distributional model by computing Spearman's $\rho$. The MEN dataset contains a separate development set which will be referred to as MEN (dev) throughout this work.

*SimLex-999*

The SimLex-999 dataset (Hill et al., 2015) contains 999 word pairs, consisting of adjectives, nouns and verbs. Unlike MEN, all comparisons between words are between pairs with the same part of speech. The word pairs in the dataset have been selected on the basis of the University of South Florida Free Association Database (Nelson et al., 2004) and WordNet (Fellbaum, 1998). Annotations have been collected from Amazon Mechanical Turk with careful annotation guidelines that aim to capture synonymy and near-synonymy, while regarding, for example antonyms, which are treated as related in MEN, as completely dissimilar[9]. Each annotator rated 20 groups of word pairs, where each group consisted of 6-7 pairs, on a scale between 0 (completely dissimilar) to 6 (maximally similar). In a post-processing step, the ratings have been averaged across all annotations per word pair, and linearly transformed to the interval $[0, 10]$ (from $[0, 6]$). The task is to compare the averaged human annotations to the similarity estimates of the distributional model by computing Spearman's $\rho$.

*Short Phrase Composition*

For evaluating distributional composition, I am using the dataset introduced by Mitchell and Lapata (2010), henceforth referred to as "ML2010". The dataset assesses the capability of a compositional distributional model to represent adjective-noun (AN), noun-noun (NN), and verb-object (VO) compounds. Each phrase type consists of 108 phrase pairs which have been annotated in a crowdsourced experiment on a Likert scale from 1 (completely dissimilar) to 7 (maximally similar). In total 162 annotators have been recruited, each one judging 36 phrase pairs for their similarity, leading to 5832 annotations in total, uniformly distributed across the three phrase types. The task is

9 For example the antonym pair *hot - cold* has a normalised similarity of 0.66 in MEN, whereas the antonym pair *old - new* only scores 0.16 (normalised) in SimLex-999.

to compose each of the two phrases in a pair and compare the resulting distributional similarity of the composed phrases to each human annotator *individually* by computing Spearman's $\rho$. Notably, this differs from the evaluation strategy used in the word similarity datasets where the similarity estimates of a distributional model are compared to *averaged* human similarity judgements.

### *BLESS*

The BLESS dataset (Baroni and Lenci, 2011) consists of 200 concrete single-word noun concepts which are paired with relata from the following set of semantic relations to the target concepts: co-hyponyms ("coord"), hypernyms ("hyper"), meronyms ("mero"), attributes ("attri") and events ("event"). Co-hyponyms, hypernyms and meronyms are all nouns, attributes are adjectives, and are selected to represent typical properties of the given target concept. Events are verbs that are frequently associated actions or happenings the target concept is involved in. For all 3 parts of speech, the BLESS dataset includes random distractors ("random-n", "random-j" and "random-v").

The target concepts have been chosen by using the McRae Feature Norms dataset (McRae et al., 2005) as primary source, and relata have been selected on the basis of WordNet (Fellbaum, 1998), ConceptNet (Liu and Singh, 2004), and the ukWaC corpus (Ferraresi et al., 2008), and have been filtered and validated using Amazon Mechanical Turk. The task is to calculate the similarity of every concept noun with all of its relata and choose the most similar relatum per relation. The distributions are subsequently converted into z-scores to account for concept-specific sparsity effects, and summarised in a box-and-whisker plot in order to visualise the preference of a given model for particular semantic relations (Baroni and Lenci, 2011).

## 4.2 PRACTICAL EVALUATION

A major challenge when putting a theoretical proposal into practice is finding a set of robust parameters that optimise the performance of the given model. As other compositional distributional models, Anchored Packed Trees specify a number of parameters that need to be set prior to deriving representations from a corpus. These hyperparameters can have a significant impact on the nature and characteristics of the resulting distributional space. This section aims to

identify the most important hyperparameters by quantifying their impact on 3 word similarity task and a short phrase composition task. On the basis of the hyperparameter sensitivity analysis, a set of favourable parameter settings is identified and will be adopted for further processing.

In the following, the hyperparameters are defined in Section 4.2.1, and the methodology and quantification of the contribution of each parameter is subsequently presented in Section 4.2.3. Section 4.2.4 provides recommendations for favourable parameter settings.

### 4.2.1 *Hyperparameters*

The investigated hyperparameters are categorised into three groups, *preprocessing parameters*, APT *parameters*[10], and *lexical association metric parameters*, each group concerning a different stage of the APT creation pipeline. A large body of previous work, such as Bullinaria and Levy (2012); Lapesa and Evert (2014); Kiela et al. (2014); Lapesa and Evert (2017) has already investigated the impact of various hyperparameterisations on typed and untyped distributional semantic models. The insights obtained by these studies provide a robust starting point for the parameters in this work. For example there is strong evidence that some form of mutual information (Lapesa and Evert, 2014; Kiela et al., 2014; Polajnar and Clark, 2014; Weeds et al., 2014b) or *t*-test (Curran, 2004; Kiela et al., 2014; Polajnar and Clark, 2014) as lexical association function consistently is among the top performers in recent studies. Furthermore, Sahlgren (2006); Levy et al. (2015) have shown that the weighting applied to the context window can have a substantial effect on the characteristics of the distributional space.

#### *Preprocessing Parameters*

The parameters investigated at the preprocessing stage include the use of lemmatisation: `{true, false}`, the use of part of speech tags: `{true, false}` and the granularity of the part of speech tags when they are used: `{1, full}`. When being set to 1, only the first letter of a PoS tag is considered for a given lexeme. This means that all tags associated with nouns, such as NN, NNS, or NNP would be mapped to just N. The option `full` uses the complete tag information.

10 While referred to as APT parameters they are also applicable to other typed distributional models such as the SVS model of Padó and Lapata (2007).

The preprocessing parameters can have a considerable impact on the dimensionality of the distributional space. For example a lemmatised order 3 APT space without PoS tags gives rise to ≈1.39M dimensions, whereas the same order 3 APT space without lemmatisation or PoS tags results in ≈1.51M dimensions — a difference of ≈120k contextual dimensions.

*APT Parameters*

In this section, the path length and its associated weight are investigated, representing two specific APT parameters.

Path length — or APT order — defines the maximum length of typed features in an APT, and has a similar role to the size of the sliding window in untyped distributional semantic models. Figure 4.1 shows the features that would be extracted for the verb *jumps* in the given example sentence when using an order 1 APT space (top) in comparison to an order 2 APT space (bottom). Notably, an order 3 APT would be sufficient to capture all lexemes in the sentence for the anchored word *jumps*.

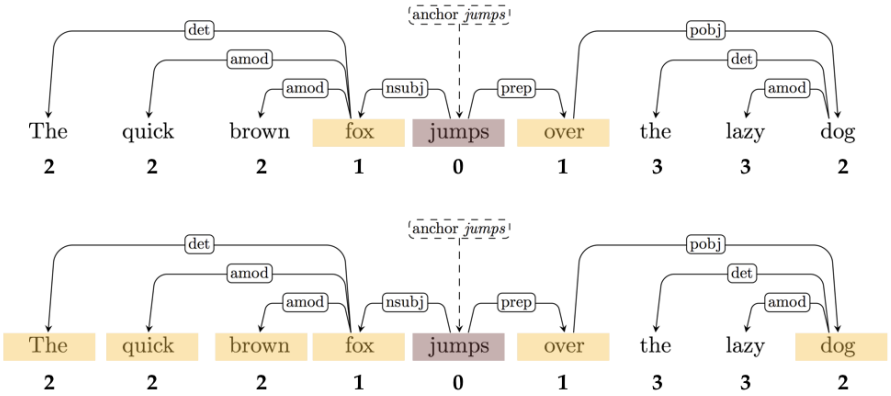


Figure 4.1: Co-occurrence features captured by an order 1 APT (top) and an order 2 APT bottom for the lexeme *jumps*. The numbers below each lexeme indicate the distance from the anchored word. For the given sentence, an order 3 APT would capture the full sentence for the lexeme *jumps*.

Figure 4.1 highlights that higher-order paths tend to provide less direct evidence about the semantics of a given lexeme. For example, the order 1 features for the lexeme *jumps* provide distributional evidence that *foxes* are able to jump and that it is possible to jump *over* something. The order 2 features of *jumps* provide evidence that *quick* and *brown* things can jump and that it is possible for *something* to jump over a *dog*. While the order 2 features still provide useful co-occurrence information, they tend to be less general and more specific

to the current context, i.e. not all *quick* or *brown* things might be able to jump. In this section, I am investigating the following path lengths for building APTs: `{1, 2, 3, 5, 7}`.

After defining the order of the APT space, the associated path weighting function needs to be defined next. This parameter is the equivalent of weighting the terms in the sliding window in untyped distributional semantic models. The following path weighting schemes are considered in this section: `{constant, harmonic, inverse harmonic, very aggressive, path probability}`.

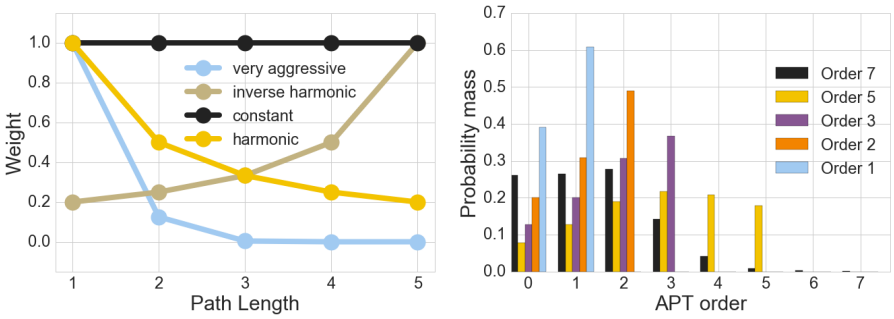


Figure 4.2: Overview of path weighting functions directly dependent on path length (left), and accumulation of probability mass per path length for the `path probability` weighting scheme (right).

Figure 4.2 (left) illustrates the effect of different path weighting schemes for an order 5 APT space. Path probability weighting is notably missing from Figure 4.2 (left) as it is the only path weighting scheme that is not a direct function of the length of a given path. Figure 4.2 (right) shows the accumulated probability mass of paths of different lengths across all words in the BNC with `path probability` as weighting scheme.

As Figure 4.2 (left) shows, `constant` path weighting is the simplest function and uniformly weights paths independent of their length. The `harmonic` path weighting scheme weights paths as $\phi = \frac{1}{l}$ where $l$ is the length of a given path such that paths of length 2 are assigned a weight of $\frac{1}{2}$, paths of length 3 are assigned a weight of $\frac{1}{3}$, and so on. A more extreme version of downweighting longer paths is represented by the `very aggressive`[11] path weighting scheme, where the weight assigned to a path is defined as $\phi = 2^{1-l^2}$. As Figure 4.2 (left) shows, this scheme causes paths beyond the length of 2 to be assigned a weight of nearly 0. For example a path of length 2 would be assigned a weight of $\frac{1}{8}$ and a path of length 3 would be assigned a weight of $\frac{1}{256}$.

11 I am using a more extreme variant of the "aggressive" weighting function, $\phi = 2^{1-l}$, suggested by Sahlgren (2006), as it would have been too similar to the `harmonic` path weighting function for the path lengths under consideration.

The general hypothesis is that features associated with longer paths contribute less salient information to the distributional semantics of a given lexemes than shorter and more direct paths (Padó and Lapata, 2007; Weir et al., 2016).

This section furthermore includes an `inverse harmonic` path weighting scheme that assigns higher weight to longer paths. For example in an order 5 APT space, direct relations are assigned a weight of $\frac{1}{5}$, whereas paths of length 3 would be assigned a weight of $\frac{3}{5}$.

The last path weighting scheme I am considering in this section is `path probability` that weights paths according to the probability that a randomly selected co-occurrence between the lexemes $w$ and $w'$ happens to be of type $\tau$ as shown in Equation 4.1 below:

$$p(\tau \mid w) = \frac{\#\langle w, \tau, * \rangle}{\#\langle w, *, * \rangle} \tag{4.1}$$

where $\#\langle w, \tau, * \rangle$ is the number of co-occurrence events of the lexeme $w$ with path $\tau$, and $\#\langle w, *, * \rangle$ denotes the number of co-occurrence events involving the lexeme $w$. Interestingly, as Figure 4.2 (right) shows, the majority of the probability mass for paths of length 2-5 is accumulated for longer paths. This is because there are considerably more 2nd order features than 1st order features, thus the higher-order features accumulate a larger proportion of the available probability mass[12]. For the order 7 APT space, however, the distribution is shifted towards shorter paths. One explanation for this behaviour is that the large amount of additional features of orders 6 and 7 caused a considerably larger number of negative PPMI scores which have subsequently been filtered due to the positive threshold, while retaining proportionally more features with shorter paths.

*Lexical Association Metric Parameters*

Lexical association functions are commonly applied to count-based distributional semantic spaces to transform raw counts to scores that better capture more expressive contexts of a given lexeme. For example common contexts for a noun include the articles *the* or *an*, which are very uninformative, and whose contribution to the semantics of a noun should therefore be decreased. The contribution of other content words such as verbs co-occurring with a given noun, on the other hand, should be increased.

12 Which, however, does not necessarily translate to a larger proportion of PPMI score mass as will be highlighted in Table 4.2 below.

In this section I am considering 3 lexical association functions which have been found to work well in a number of studies (Baroni and Lenci, 2010; Kiela and Clark, 2014; Polajnar and Clark, 2014): `{PPMI, PLMI, t-test}`.

Positive Pointwise Mutual Information (PPMI) (Church and Hanks, 1989; Dagan et al., 1993), is perhaps the most commonly used association score for count-based distributional semantic models and is a measure of how surprising a collocation between a context and a target word is. The PMI and PPMI formulas for APTs have been defined in the previous chapter in Equations 3.6 and 3.7, respectively, and are repeated in Equation 4.2 below for convenience.

$$\begin{aligned} p(w, w'; \tau) &= \frac{\#\langle w, \tau, w' \rangle}{\#\langle *, \tau, * \rangle} \\ p(w; \tau) &= \frac{\#\langle w, \tau, * \rangle}{\#\langle * \tau, * \rangle} \\ p(w'; \tau) &= \frac{\#\langle *, \tau, w' \rangle}{\#\langle *, \tau, * \rangle} \end{aligned} \tag{4.2}$$

$$\begin{aligned} \text{PMI}(w, w'; \tau) &= \log \frac{p(w, w'; \tau)}{p(w; \tau) p(w'; \tau)} \\ \text{PPMI}(w, w'; \tau) &= \max(\text{PMI}(w, w'; \tau), 0) \end{aligned}$$

For PPMI, I furthermore consider two more parameters that have been shown to have a major impact on the performance of a distributional model on numerous tasks. The first one is a negative shift of the PMI matrix (Levy and Goldberg, 2014b), which is applied before clamping all negative values at 0. The negative shift has the effect of promoting more prominent PMI scores, while zeroing out potentially noisy scores. Shifting the PMI scores is furthermore related to the context selection method introduced by Polajnar and Clark (2014) which selects the top $n$ largest PMI scores and zeroes out the rest. The two methods are achieving the same effect by different means. Where a shift of the PMI values discards all dimensions with a PMI score lower than some threshold, context selection explicitly retains the top $n$ highest scoring dimensions. Interestingly, a positive effect of considering only PMI scores above a certain threshold (albeit on a different task) has already been observed in earlier work (Church and Hanks, 1989). I am considering the following negative shifts of the PMI matrix: `{log 1, log 5, log 10, log 40, log 100}`. SPPMI alters the PPMI formula to Equation 4.3 below:

$$\text{SPPMI}(w, w'; \tau) = \max(\text{PMI}(w, w'; \tau) - \log k, 0) \tag{4.3}$$

where $k$ is the magnitude of the negative shift.

The second PPMI parameter under consideration is context distribution smoothing (cds) (Levy et al., 2015) which can be applied to reduce the contribution of more common contexts shown in Equation 4.4 below:

$$\text{PMI}_\alpha(w, w'; \tau) = \log \frac{p(w, w'; \tau)}{p(w; \tau)p(w'; \tau)^\alpha} \tag{4.4}$$

where $\alpha$ denotes the magnitude of the applied smoothing operation. I am considering the following values for $\alpha$: `{1.0, 0.75}`, where 1.0 means that no context distribution smoothing is applied and represents the standard PMI association function as shown in Equation 4.2[13].

The second lexical association function under consideration is Pointwise Localised Mutual Information (PLMI) (Scheible et al., 2013), which scales the PMI score by the joint probability[14] of two lexemes co-occurring together with a given path as shown in Equation 4.5 below.

$$\begin{aligned} p(w, w'; \tau) &= \frac{\#\langle w, \tau, w' \rangle}{\#\langle *, \tau, * \rangle} \\ \text{PLMI}(w, w'; \tau) &= p(w, w'; \tau) \times \text{PMI}(w, w'; \tau) \end{aligned} \tag{4.5}$$

The last lexical association function under consideration is the $t$-test, which is a hypothesis testing technique. The $t$-test requires the definition of a null hypothesis that contradicts the desired result, and is subsequently rejected on the basis of the outcome of the statistical test (Curran, 2004). For lexical association the null hypothesis would be that a co-occurrence between two lexemes $w$ and $w'$, given a path $\tau$, is independent or unrelated. This would mean that the joint probability of the co-occurrence event would be equal to the product of the respective marginals: $p(w, w'; \tau) = p(w; \tau)p(w'; \tau)$, in which case the association score between $w$ and $w'$, given $\tau$, would be 0 as Equation 4.6 below shows.

13 In preliminary experiments I found that no other value of $k$, either lower than 0.75 or higher than 1.0, had a positive effect.

14 PLMI notably differs from LMI (Evert, 2005) which scales PMI by the frequency of the target word $w$.

$$
\begin{aligned}
p(w, w'; \tau) &= \frac{\#\langle w, \tau, w' \rangle}{\#\langle *, \tau, * \rangle} \\
p(w; \tau) &= \frac{\#\langle w, \tau, w' \rangle}{\#\langle w, \tau, * \rangle} \\
p(w'; \tau) &= \frac{\#\langle *, \tau, w' \rangle}{\#\langle *, \tau, * \rangle}
\end{aligned} \tag{4.6}
$$

$$
\text{t-test}(w, w'; \tau) = \frac{p(w, w'; \tau) - p(w; \tau)p(w'; \tau)}{\sqrt{p(w; \tau)p(w'; \tau)}}
$$

A positive *t*-test score indicates some level of association between a lexeme and its context, where the level of dependence between the lexeme and its context is proportional to the magnitude of the *t*-test score.

The higher the score, the more dependence there is between a lexeme and its co-occurring context.

### 4.2.2 *The APT Baseline Model*

To better assess the performance difference between the various parameterisations, I am using a standard parameterisation of the Apt space as baseline that has been shown to achieve competitive results in previous studies (Weir et al., 2016; Kober et al., 2017a)[15] and is shown in Table 4.1 below[16].

This model will henceforth be referred to as "Apt baseline model" or simply "baseline model", which is shorthand for an Apt space, produced by a particular corpus (the BNC in this case), pre-processed with the pipeline outlined in Section 4.1.1 above, and parameterised as shown in Table 4.1.

#### *The Feature Space of the APT Baseline Model*

Table 4.2 below shows basic feature statistics of the Apt feature space. The data has been obtained on the basis of the MEN dataset, consisting of 57 unique adjectives, 656 unique nouns and 38 unique verbs. For all lexemes per PoS tag, the number of features per path length

15 The resulting models are not exactly the same, as the previously published results have been achieved with a slightly different preprocessing pipeline, such as the use of a pre-parsed version of the source corpus, without any number or url normalisation.

16 Due to not using any PoS tag information the granularity parameter is not applicable (`N.A.`) for the baseline model.

| Parameter | Value |
|---|---|
| Lexical Association Metric | `PPMI` |
| Path weight | `constant` |
| SPPMI shift | `log 1` |
| CDS | `1.0` |
| APT order | `2` |
| Lemma | `True` |
| PoS | `False` |
| PoS granularity | `N.A.` |

Table 4.1: Hyperparameter configuration of the APT baseline model.

have been counted and subsequently averaged by the number of unique lexemes per PoS tag. To obtain the PPMI score distribution, the PPMI scores for every lexeme per PoS tag have been tallied up and subsequently normalised.

| Path Length | Number of Features | PPMI Score Distribution |
|---|---|---|
| 0 | ≈34 (*JJ*), ≈12 (*NN*), ≈5 (*VB*) | 0.11 (*JJ*), 0.05 (*NN*), 0.01 (*VB*) |
| 1 | ≈167 (*JJ*), ≈199 (*NN*), ≈634 (*VB*) | 0.52 (*JJ*), 0.62 (*NN*), 0.60 (*VB*) |
| 2 | ≈223 (*JJ*), ≈229 (*NN*), ≈708 (*VB*) | 0.37 (*JJ*), 0.33 (*NN*), 0.39 (*VB*) |

Table 4.2: Average number of features and PPMI score distribution per path length and PoS tag for all lexemes occurring in the MEN dataset on the basis of the order 2 APT baseline model.

As Table 4.2 shows, the number of features generally increases with their path lengths, however, despite the larger number of features of order 2, the combined magnitude of their PPMI scores is substantially smaller than for the features with path length 1. Even in the absence of any path weighting function, this behaviour appears to capture the idea that features closer to the target word generally contribute more to the semantics of a word (Padó and Lapata, 2007; Weir et al., 2016). A further interesting observation is that verbs appear to have the richest contexts with more than 600 order 1 features on average, in comparison to only $\approx 200 - 230$ for adjectives and nouns.

### 4.2.3 *Hyperparameter Sensitivity Study*

In order to estimate and quantify the impact of any of the hyperparameters introduced above, I follow the evaluation strategy proposed by Lapesa and Evert (2014) who learn a linear regression model to predict the performance of a distributional model on a given task

from its parameterisation. More concretely, Lapesa and Evert (2014) treat the performance measure on the given task — Spearman's $\rho$ in the case of the tasks used in this section — as the dependent variable, and the hyperparameters of the given distributional model as the independent variables. Hence, running all distributional models on a given task creates a set of observations, and Lapesa and Evert (2014) subsequently aim to fit a linear regression model to predict the model performance from the hyperparameters. As the primary interest of this study is the impact of each hyperparameter *individually* — and as the individual parameters account for a sufficiently large amount of the variance — I do not add higher-order interactions as independent variables to the model as Lapesa and Evert (2014) do.

In the following, the method of Lapesa and Evert (2014) will be applied to quantify the impact of each individual parameter on the 3 word similarity tasks, as well as the short phrase composition task.

*Word Similarity*

Fitting a linear regression model to each of the word similarity datasets results in ≈70-80% of the variance explained by the model. Figure 4.3 shows the result of a feature ablation study on the basis of an ANOVA type 2 test. The ANOVA test shows that the choice of the lexical association metric has the largest impact on subsequent model performance, followed by the path weighting scheme and the magnitude of the negative SPPMI shift.

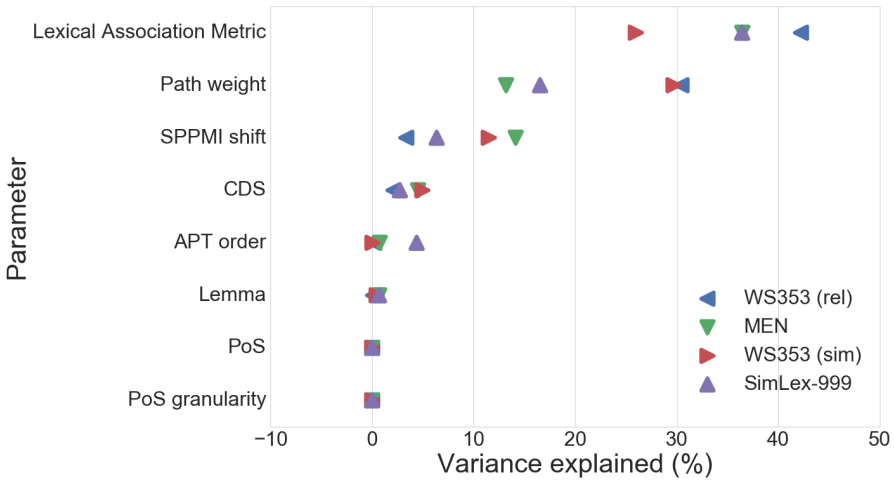


Figure 4.3: ANOVA type 2 test for quantifying the amount of variance explained for each parameter for the word similarity tasks.

Interestingly, the preprocessing parameters (lemmatisation, PoS tagging, PoS granularity) have very little impact on the quality of the

resulting model on the word similarity tasks. Furthermore the order of the APT space, as well as the application of context distribution smoothing, have a negligible effect as well.

For the MEN dataset the parameters have been tuned on its development set. For the other word similarity datasets, following the evaluation methodology of Levy et al. (2015), I have used 2-fold cross-validation to tune the hyperparameters and the results represent the averaged Spearman $\rho$ scores.

As the results in Table 4.3 show, tuning the parameters pays off, with statistically significant improvements on the two WordSim-353 datasets and MEN, and a more modest performance improvement on the SimLex-999 dataset, in comparison to the baseline model. Interestingly while the performance of APTS on MEN is substantially weaker than the performance of the comparable model of Kiela et al. (2014)[17], the reverse happens for the WordSim-353 dataset[18]. While both, APTS and the model of Kiela et al. (2014), have a comparable base setup they differ in a number of aspects such as the preprocessing pipeline with differing parser and the use of potentially different sets of dependency relations[19]. Furthermore, the way in which PMI is performed between the two models is different, as Kiela et al. (2014) treat the dependency relation as part of the context, whereas PMI weighting in this work follows Hindle (1990) by treating the path as fixed in a co-occurrence event.

The improvements on the MEN and the WordSim-353 (similarity) tasks are statistically significant at the $p < 0.01$ level. The improvement for the WordSim-353 (relatedness) subtask is statistically significant at the $p < 0.05$ level. Statistical significance has been determined using the method of Steiger (1980).

Table 4.3 also highlights some clear trends with regards to favourable parameterisations. In general, context distribution smoothing does not improve performance on the word similarity tasks, and top performance is usually achieved with a lemmatised APT model, without any PoS tag information. Furthermore, the best performing models for WordSim-353 (similarity), WordSim-353 (relatedness) and

17 Kiela et al. (2014) report a Spearman's $\rho$ score of $\approx 0.5$ on the MEN dataset, whereas the tuned APTS only achieve a score of 0.43.

18 Kiela et al. (2014) achieve a Spearman's $\rho$ score of $\approx 0.36$, whereas APTS achieve a Spearman's $\rho$ score of 0.40 on the whole dataset (not shown in Table 4.3).

19 APTS use universal dependencies, which are very fine grained, Kiela et al. (2014) do not specify explicitly which dependency annotation schema is used by their parser.

| | WS353 (sim) | WS353 (rel) | MEN | SimLex-999 |
|---|---|---|---|---|
| Lex. Assoc. | PPMI | PPMI | PPMI | PPMI |
| Path weight | constant | constant | constant | inv. harmonic |
| SPPMI shift | log 40 | log 40 | log 40 | log 5 |
| CDS | 1.0 | 1.0 | 1.0 | 1.0 |
| APT order | 1 | 1 | 1 | 7 |
| Lemma | true | true | true | true |
| PoS | false | false | false | false |
| PoS gran. | N.A. | N.A. | N.A. | N.A. |
| Result | **0.52**‡ (+/- 0.09) | **0.35**† (+/- 0.01) | **0.43**‡ (+/- 0.02) | **0.25** (+/- 0.01) |
| Baseline | 0.40 (+/- 0.14) | 0.24 (+/- 0.06) | 0.36 (+/- 0.01) | 0.22 (+/- 0.02) |

Table 4.3: Results and corresponding hyperparameters for the APT spaces with optimal parameterisation for the word similarity datasets in comparison to the APT baseline model. Performance is reported in terms of averaged Spearman $\rho$ across 2-fold cross-validation. The numbers in parentheses denote the standard deviation across the two runs. ‡ marks statistical significance at the $p < 0.01$ level and † marks statistical significance at the $p < 0.05$ level according to the method of Steiger (1980).

MEN are achieved with an order 1 APT space and a relatively high negative SPPMI shift of log 40. SimLex-999 is the only dataset breaking that pattern, by preferring a lower SPPMI shift of log 5 and an order 7 APT space. Interestingly, assigning a *higher* weight to distributional features with *longer* paths outperforms other path weighting schemes on SimLex-999, although the differences to other path weighting schemes are relatively small.

The high negative SPPMI shift of log 40 zeroes out the majority of features in the representation. The remaining features exhibit a relatively stringent set, focused on a particular sense of the lexeme. For example the features with the highest PPMI scores in the APT baseline model for the lexemes *bank* and *market* are relatively mixed between the different senses of the lexemes. In the model with the higher SPPMI shift on the other hand, the features of the two lexemes are dominated by the *financial institution* and *financial trading place* meanings of *bank* and *market*, respectively. The higher order of the best performing configuration on the SimLex-999 dataset causes a relatively larger number of modifier features in the representation in comparison to the baseline model. As the SimLex-999 is more difficult due to its strict focus on synonymy, the additional content words in the representation have a small positive effect on performance.

The best performing path weights for WS353 (sim), WS353 (rel) and MEN are slightly misleading because the top performing models are of order 1, meaning neither of the path weighting schemes, except

`path probability`, are relevant for this APT model. In the following, I will separately investigate the impact of the 3 parameters that account for the largest amount of variance, according to Figure 4.3.

Table 4.4 lists the best result involving a specific parameterisation. For example, the lexical association metric group lists the best result for any APT space per lexical association function, independent of all other parameters. Upwards pointing arrows such as ⇑ and ↑ mark generally superior (⇑) and mostly superior (↑) parameterisations, and a ⇓ arrow marks a generally unfavourable parameterisation.

| | | WS353 (sim) | WS353 (rel) | MEN | SL |
|---|---|---|---|---|---|
| Lex. Assoc. | **PPMI⇑** | **0.52** | **0.35** | **0.43** | **0.25** |
| | PLMI | 0.36 | 0.12 | 0.35 | 0.22 |
| | t-Test | 0.38 | 0.22 | 0.32 | 0.15 |
| Path weight | constant | 0.49 | 0.33 | **0.43** | 0.24 |
| | harmonic | 0.49 | 0.33 | **0.43** | 0.24 |
| | inv. harmonic | 0.49 | 0.33 | **0.43** | **0.25** |
| | path prob.⇓ | 0.20 | 0.11 | 0.19 | 0.11 |
| | **very aggr. ↑** | **0.50** | **0.34** | **0.43** | 0.24 |
| SPPMI shift | log 1 | 0.43 | 0.28 | 0.37 | 0.23 |
| | log 5 | 0.46 | 0.32 | 0.36 | **0.25** |
| | log 10 | 0.47 | 0.33 | 0.38 | 0.24 |
| | **log 40↑** | **0.52** | **0.35** | **0.43** | 0.20 |
| | log 100 | 0.48 | 0.27 | 0.39 | 0.16 |

Table 4.4: Overview of the impact of individual parameterisations. All results represent the best run for each respective set of parameters. Boldfaced numbers indicate the best result for the given parameter, underlined numbers indicate the worst result. Recommended parameterisations are marked with ⇑ (strongly recommended) and ↑ (recommended), and parameterisations marked with ⇓ are advised against and should be avoided.

As Table 4.4 shows, using `PPMI` as lexical association function outperforms the other options by a considerable margin. Surprisingly, the different path weighting schemes are all relatively similar in terms of their performance on the given word similarity task, except for `path probability` which performs much worse than any other path weighting function, and is the main driver of variance for this parameter. An explanation for its poor performance would be that the dependency-typed nature of the space is too fine grained, and potentially too sensitive to frequency effects, causing an increase in weight for common paths such as between a determiner and a noun, that would otherwise be assigned a much lower score due to PPMI. Us-

ing a weighting criterion that relies on the path label is less robust to parsing errors, causing an accumulation of probability mass for less informative or even implausible paths[20]. Furthermore, except for SimLex-999, a relatively high SPPMI shift of $\log 40$ is performing best for word similarity, which has already been observed in previously published work (Kober et al., 2016). In the following, I will refer to the best performing configuration on both WS353 subtasks and MEN as the "APT-WS-MEN" model, and to the best performing model on SimLex-999 as the "APT-SL-999" model.

*Measuring the Impact on the Distributional Space*

A different choice of hyperparameters can lead to significantly different distributional APT spaces. In order to investigate the effect of different configurations on the distributional neighbourhood, I am measuring the neighbour overlap between two APT spaces for a given set of lexemes. The neighbour overlap between two APT spaces, $\mathbf{A}$ and $\mathbf{A}_{\text{ref}}$, as defined in Equation 4.7 below, measures how many of the top $n$ neighbours, independent of their rank, are shared between $\mathbf{A}$ and $\mathbf{A}_{\text{ref}}$ for each lexeme in the set $W \subseteq V$, where $V$ denotes the vocabulary.

$$\text{overlap}(\mathbf{A}, \mathbf{A}_{\text{ref}}; W) = \frac{\sum_{w \in W} |\text{N}_{\mathbf{A}}(w) \cap \text{N}_{\mathbf{A}_{\text{ref}}}(w)|}{|W| \cdot n} \tag{4.7}$$

$\text{N}_{\mathbf{A}}(w)$ is a function returning the top $n$ neighbours for some APT $\mathbf{A}$, and $|W|$ denotes the size of the set $W$.

For a comparison I am using the APT baseline model introduced previously as $\mathbf{A}_{\text{ref}}$, all individual lexemes from the WordSim-353[21] dataset as the set $W$, and the top 100 neighbours for each lexeme ($n = 100$). The APT-SL-999 model has a neighbour overlap of approximately 60% with respect to the baseline model. The order 1 APT-WS-MEN model, on the other hand, has a neighbour overlap with respect to the baseline of only 37%[22].

20 Interestingly the highest scoring features, for a number of investigated nouns, in an order 2 APT space with path probability weighting are almost exclusively other nouns with a path of `compound` or $\overline{\texttt{compound}}$, but only very few $\overline{\texttt{amod}}$, `dobj` or `nsubj` features. This is in contrast to the APT baseline model, where more high scoring features are co-occurrences with adjectives and verbs.

21 The dataset consists of 437 unique words.

22 The ranks of the neighbours between models, on the basis of a comparison of the pairwise similarities of all 437 unique words in WS353, differ substantially, with a Kendall's $\tau$ of only $\approx$0.014, and a Spearman's $\rho$ of only $\approx$0.021 between the baseline model and the APT-SL-999 model. This provides further evidence that the spaces differ considerably between configurations, however the low correlation suggests that taking the ranks of the neighbours into account is a too sensitive measure. Therefore,

| **Lexeme** | APT **Baseline** | APT-**SL-999** | APT-**WS-MEN** |
|---|---|---|---|
| game | match, season, player, goal, team | match, program, season, goal, player | match, adventure, competition, clash, fun |
| government | authority, party, council, company, state | authority, council, party, state, policy | regime, parliament, cabinet, administration, policy |
| change | development, difference, increase, effect, alter | development, effect, increase, alter, shift | variation, privilege, alter, improvement, review |

Table 4.5: Nearest neighbours of the 3 different APT spaces for 3 example lexemes. While even the top neighbours do not necessarily overlap between the 3 spaces, the neighbours are all topically coherent and predominantly co-hyponyms of the respective target lexeme.

Table 4.5 show the 5 nearest neighbours of the 3 different APT spaces under consideration for 3 example lexemes, and provides a first hint that the distributional neighbourhood of all 3 APT spaces is governed by co-hyponymy. In general, previous research (Peirsman, 2008; Baroni and Lenci, 2011; Levy and Goldberg, 2014a) has found that typed distributional models generally favour co-hyponymy and hypernymy in their neighbourhood and 4.5 suggests that this trend also holds for APTs (a more exhaustive characterisation of the APT space is presented in Section 4.3). A more The neighbourhood for the lexeme *game* is dominated by the "sports" sense of *game* for the APT baseline and APT-SL-999 spaces, with all of their 5 nearest neighbours expressing that sense. The APT-WS-MEN space exhibits a second dominant factor in the distributional neighbourhood, the sense of *game* related to "children playing" with neighbours such as *fun* and *adventure*.

For the lexeme *government*, the APT-WS-MEN space appears to capture a more narrow sense of *government*, perhaps describable as the "legislative body" sense of *government*, with neighbours such as *cabinet* and *administration*. The other 2 spaces on the other hand capture a more abstract and general notion of *government* with neighbours such as *state* and *authority*.

An interesting effect happens for the lexeme *change*, where the distributional neighbourhoods for all 3 spaces seem to mix verbs, such as the lexeme *alter* appearing in all 3 APT spaces, and nouns. This is

I will be using the neighbour overlap measure as defined in Equation 4.7 to quantify in how far two APT spaces differ.

an artifact of working on a non PoS tagged source corpus, however, as shown in the ablation study above (see Figure 4.3), the inclusion of PoS tags did not have a significant effect on the performance of the APT spaces on the word similarity tasks, it might however, affect the distributional neighbourhood as shown in Table 4.5.

*Phrase Similarity*

I am using the ML2010 dataset for evaluating the impact of different model parameters for distributional composition tasks. The dataset consists of 108 pairs of adjective-noun, noun-noun and verb-object compounds, 324 phrase pairs in total, rated by multiple human annotators for similarity. All parameters are tuned on the development portion of the ML2010 dataset[23]. Hyperparameter effects are compared to the same APT baseline model as used for the word similarity tasks above.

Given that neither lemmatisation nor PoS tags had a significant effect on the results of the word similarity tasks, I am dropping these parameters from the hyperparameter sensitivity analysis for the phrase similarity task. I furthermore exclude all APT models with the `path probability` weighting scheme due to their poor performance on the word similarity tasks. For the experiments involving composition, an additional parameter — the composition function used to combine two aligned APT representations — is added to the study. I am considering the following composition functions for the ML2010 dataset: `{intersection, union}`.

Composition by intersection only keeps distributional features that occur in both APTS. This results in composed representations that are substantially sparser — and more discriminative — than representations obtained with composition by union, which merges all features of the two aligned APTS.

The same line fitting procedure of all first-order features, as outlined earlier in this section, results in an adjusted $R^2$ of $\approx$0.74-0.78 for the different phrase types, meaning that approximately 74-78% of the variance can be explained by the model. This represents a substantial amount of model variation explained and allows solid inferences regarding the importance of individual parameters.

23 The phrase pairs labelled by the first 108 participants are the test set and the phrase pairs labelled by the last 54 participants are the development set. Information based on personal communication with Douwe Kiela (18th April, 2016) who shared his communication with Jeff Mitchell w.r.t. the test/development split on the ML2010 dataset with me.

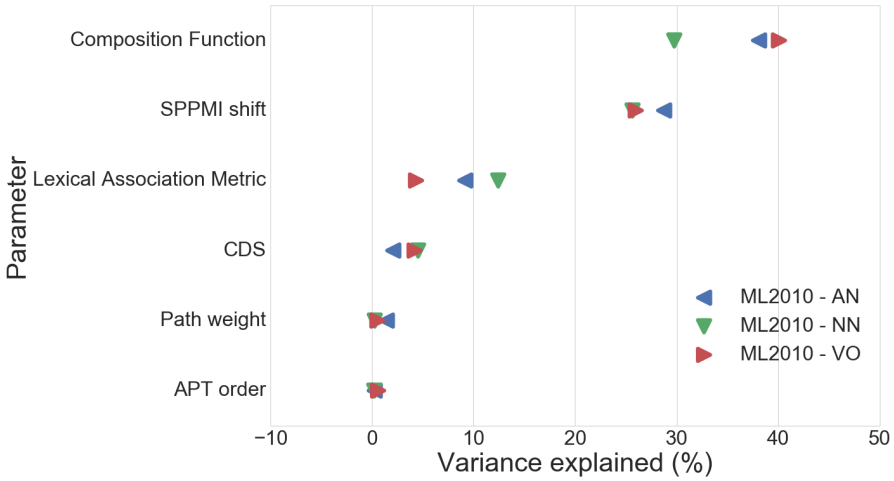


Figure 4.4: ANOVA type 2 test for quantifying the amount of variance explained for each parameter for the ML2010 dataset.

Figure 4.4 shows that the composition function is the parameter causing the largest proportion of variance in the data, followed the by the negative SPPMI shift, and to a lesser extend, the lexical association function. Interestingly the order of the APT space only has a small impact for most configurations. Due to the removal of models with the `path probability` weighting scheme, the path weight parameter has relatively little impact on model performance as well.

Table 4.6 shows the best performing configurations per composition function. Interestingly, the optimal choice of order for the APT space on the phrase similarity tasks appears to be 2 or 3, hence adding order 5 or even order 7 features does not appear to be beneficial. The optimal magnitude of the negative SPPMI shift is considerably lower for the phrase similarity task than for the word similarity tasks above. This suggests that for tasks involving distributional composition it is beneficial to keep distributional features relating to more than the predominant sense in the representations. Any filtering of unrelated features can be carried out by the composition function itself. For composition by union a negative shift of $\log 5$ consistently results in optimal performance, whereas for composition by intersection no shift at all is the best performing configuration for this parameter.

The optimal parameterisation for the path weight parameter appears to vary for each phrase type, however the relative differences in performance between configurations is very small. Furthermore, using context distribution smoothing for composition by union per-

| **Comp. by union** | **Adjective-Noun** | **Noun-Noun** | **Verb-Object** |
|---|---|---|---|
| Lexical Association | PPMI | PPMI | PPMI |
| Path weight | harmonic | inv. harmonic | very aggr. |
| SPPMI shift | log 5 | log 5 | log 5 |
| CDS | 0.75 | 0.75 | 0.75 |
| APT order | 3 | 2 | 2 |
| Result | **0.50**‡ | **0.45**‡ | **0.45**‡ |
| Baseline | 0.43 | 0.39 | 0.41 |
| **Comp. by intersection** | **Adjective-Noun** | **Noun-Noun** | **Verb-Object** |
| Lexical Association | t-test | PPMI | PPMI |
| Path weight | inv. harmonic | inv. harmonic | harmonic |
| SPPMI shift | log 1 | log 1 | log 1 |
| CDS | 1.0 | 1.0 | 1.0 |
| APT order | 2 | 2 | 3 |
| Result | **0.39** | **0.43**‡ | **0.36** |
| Baseline | **0.39** | 0.41 | 0.35 |

Table 4.6: Results and corresponding hyperparameters for the APT spaces with optimal parameterisation for the test set of the ML2010 phrase similarity tasks in comparison to a standard APT-baseline for composition by union and composition by intersection, respectively. ‡ marks statistical significance at the $p < 0.01$ level according to the method of Steiger (1980).

forms slightly better than models without applying context distribution smoothing.

The most interesting observation, however, is that all of the improvements due to hyperparameter tuning for composition by union cause a statistically significant performance boost at the $p < 0.01$ level using the method of Steiger (1980). For composition by intersection on the other hand, the performance improvements are minimal and only the improvements for modelling noun-noun compounds have a statistically significant effect. Hence, despite extensive parameter tuning, composition by intersection still performs poorly — even in comparison to the untuned composition by union baseline. The reason for its bad performance is therefore not due to a bad choice of parameters, but due to data sparsity, and the discriminative effect of the composition function. This characteristic will be investigated in further detail in Chapter 5.

An overview of the best performing configurations per composition function for the 3 most important hyperparameters[24] is shown in Table 4.7.

| Composition by union | | Adjective-Noun | Noun-Noun | Verb-Object |
|---|---|---|---|---|
| SPPMI shift | log 1 | 0.54 | 0.42 | 0.38 |
| | **log 5⇑** | **0.55** | **0.43** | **0.41** |
| | log 10 | 0.53 | **0.43** | 0.40 |
| | log 40 | 0.41 | **0.43** | 0.34 |
| | log 100⇓ | 0.30 | 0.37 | 0.31 |
| Lex. Assoc. | **PPMI⇑** | **0.55** | **0.43** | **0.41** |
| | PLMI⇓ | 0.26 | 0.29 | 0.27 |
| | t-Test | 0.53 | 0.42 | 0.37 |
| CDS | 1.0 | 0.53 | 0.42 | **0.41** |
| | **0.75⇑** | **0.55** | **0.43** | **0.41** |
| **Composition by intersection** | | **Adjective-Noun** | **Noun-Noun** | **Verb-Object** |
| SPPMI shift | **log 1⇑** | **0.42** | **0.41** | **0.32** |
| | log 5 | 0.31 | 0.35 | 0.28 |
| | log 10 | 0.25 | 0.31 | 0.20 |
| | log 40 | 0.11 | 0.35 | 0.01 |
| | log 100⇓ | NaN | 0.02 | 0.07 |
| Lex. Assoc. | **PPMI↑** | 0.41 | **0.41** | **0.32** |
| | PLMI⇓ | 0.31 | 0.27 | 0.28 |
| | t-Test | **0.42** | 0.34 | 0.27 |
| CDS | **1.0⇑** | **0.42** | **0.41** | **0.32** |
| | 0.75 | 0.38 | 0.38 | 0.28 |

Table 4.7: Overview of the impact of individual parameterisations. All results represent the best run for each respective set of parameters on the ML2010 development set. Boldfaced numbers indicate the best result for the given parameter, underlined numbers indicate the worst result. Recommended parameterisations are marked with ⇑ (strongly recommended) and ↑ (recommended), and parameterisations marked with ⇓ are advised against and should be avoided.

The table highlights the trends observed for the best performing parameterisations in Table 4.6 above, by showing the tendency of composition by union to prefer a low negative SPPMI shift of $\log 5$, and of composition by intersection to work best without any SPPMI shift. The best performing lexical association function is PPMI, with the exception of using composition by intersection for adjective-noun phrases where *t*-test works slightly better. Lastly while the use of context distribution smoothing generally hurts performance for compos-

24 As results are shown for each composition function individually, the 3 most important parameters are therefore SPPMI shift, the lexical association function and the use of context distribution smoothing.

ition by intersection, its use with composition by union can improve the results by a small margin. In the following, the best performing configuration for composition by union will be referred to as the "APT union" model.

### 4.2.4 *Practical Recommendations*

Before starting to tune the hyperparameters it is important to choose an APT baseline model, representing a default choice of parameters. For both, word similarity and phrase similarity tasks, the chosen order 2 APT baseline model represented a solid lower bound, while not degenerating into a "trashline"[25].

As the different optimal parameterisations for word similarity and phrase similarity show, tuning the parameter space can significantly boost performance, however different tasks require different parameterisations, and it is therefore advised against to tune the model parameters on a word similarity task if the goal is to achieve strong performance on phrase similarity or another downstream task[26].

In general, it is recommended to use PPMI as lexical association function and to *avoid* `path probability` as path weighting function. Furthermore, lemmatisation appears to perform slightly better than an unlemmatised APT space, and PoS tagging does not appear to have a significant performance impact and can therefore be omitted. Furthermore lower order APT spaces (1-3) have been found to perform better than higher-order spaces.

If the aim is to optimise the APT space for word similarity tasks, it is recommended to start with a larger negative SPPMI shift ($\log 40$), and gradually decrease it if the performance remains below the chosen baseline. The path weighting scheme can relatively safely be set to `constant`, however trying `very aggressive` or even `inverse harmonic` has the potential to give marginally better results.

For optimising the parameters for a task involving composition, it is advisable to start with a low negative SPPMI shift ($\log 5$), or to not apply a shift at all. Furthermore, applying context distri-

25 The term has been coined by Twitter user `@deliprao` and refers to "A poorly constructed baseline, possibly done to make one's model look good", see https://twitter.com/deliprao/status/908987429528334336.

26 A similar observation has been made by Schnabel et al. (2015), who found that the best performing model on an intrinsic task is rarely the best performing model on a given extrinsic downstream task.

bution smoothing might slightly improve performance. The use of `constant` path weighting is recommended as a starting point, however the use of any of the other path weighting schemes (except `path probability`!) can lead to small performance improvements. Table 4.8 summarises the hyperparameter recommendations for tasks assessing word and phrase similarity, respectively.

| **Parameter** | **Elementary APTs** | **Composition** | |
|---|---|---|---|
| | | *Union* | *Intersection* |
| `Lex. Assoc.` | `PPMI` | `PPMI` | `PPMI` |
| `Lemma` | `true` | `true` | `true` |
| `PoS` | `false` | `false` | `false` |
| `PoS gran.` | `N.A.` | `N.A.` | `N.A.` |
| `Path weight` | `very aggressive` | `constant` | `constant` |
| `SPPMI shift` | `log 40` | `log 5` | `log 1` |
| `Apt order` | `1` | `2` | `2` |
| `CDS` | `1.0` | `0.75` | `1.0` |

Table 4.8: Recommended parameterisations for word similarity and phrase similarity tasks, respectively.

## 4.3 CHARACTERISING THE DISTRIBUTIONAL SPACE

In order to quantitatively describe the characteristics of the distributional APT space, I am using the BLESS dataset[27] (Baroni and Lenci, 2011) to determine which semantic relation is generally favoured by a given APT model — i.e. whether hypernyms or co-hyponyms are generally more similar to a given lexeme than meronyms or other related terms. Previous studies have concluded that typed distributional models, as well as untyped models with a small sliding window size, give rise to a distributional space governed by hypernymy and co-hyponymy (Peirsman, 2008; Baroni and Lenci, 2011; Levy and Goldberg, 2014a). In the same studies untyped distributional semantic models with a wider sliding window have been found to give rise to a space favouring meronymy and topical relatedness.

In the following, Section 4.3.1 will describe the distributional semantics of elementary APT representations, followed by the characterisation of offset APT representations and composed APT representations in sections 4.3.2 and 4.3.3, respectively.

27 See Section 4.1.4 for further details about the BLESS dataset.

### 4.3.1 *The Distributional Semantics of Elementary APT Representations*

When analysing the neighbour overlap between different APT spaces above, Table 4.5 has hinted that most neighbours of a given APT model are co-hyponyms. Previous research has found that typed distributional semantic models in general give rise to a distributional neighbourhood predominantly governed by co-hyponymy and hypernymy. The following section aims to provide empirical evidence in favour of that claim for Anchored Packed Trees.

Figure 4.5 shows the similarity distributions of semantic relations for the APT baseline model (left) and the APT-WS-MEN model (right) on the BLESS dataset. The boxplot captures the distribution of similarities across the semantic relations. Each plot represents the distribution of similarity estimates of the given APT model for each of the 200 target concepts in the BLESS dataset.

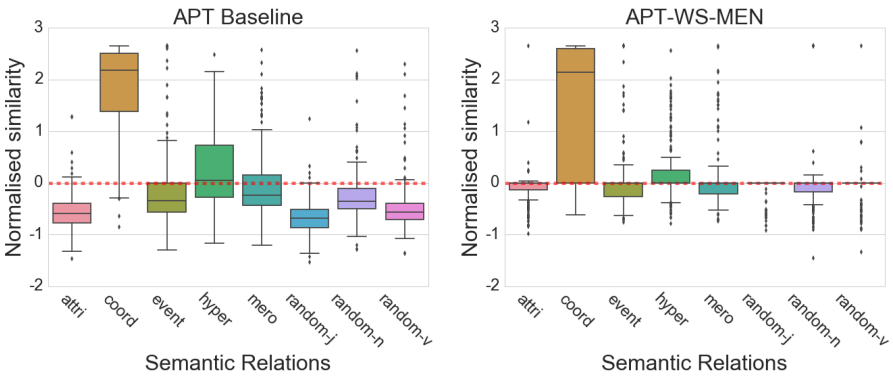


Figure 4.5: Distribution of similarities of the order 2 APT baseline model without SPPMI shift (left) and the order 1 APT-WS-MEN model with an SPPMI shift of $\log 40$ (right) on the BLESS dataset.

The plots show the median of the distribution as a horizontal line inside the box, and following the setup of Baroni and Lenci (2011), the whiskers cover 1.5 of the interquartile range beyond the boxes, with outliers plotted as diamonds outside the range covered by the whiskers. In accordance with previous research, the APT spaces exhibit a strong preference towards co-hyponymy with hypernyms representing a second, but considerably weaker, dominant factor.

The similarities of the target concepts to meronyms, however, are rarely higher than the similarities to random nouns. Other relations such as attributes and events also exhibit a relatively low average similarity to the given target concepts. The low average similarity of APT representations for target concepts in comparison to random

relata and relata with parts of speech other than nouns, furthermore suggests that the APT model represents a very stable and coherent semantic space.

Interestingly, as Figure 4.5 (right) shows, the preference of the APT-WS-MEN model is even more skewed towards co-hyponyms, however the distribution of the similarities is considerably less peaky and more spread out. One reason for this behaviour is that the negative SPPMI shift of $\log 40$ for the order 1 APT space results in fewer non-zero dimensions in the resulting representations. This in turn results in less feature overlap when calculating the cosine similarity, and hence a wider distribution of possible similarity scores. Lastly, Figure 4.5 (right) suggests that the contextual dimensions with the highest PPMI scores belong to a set of features that predominantly occur with co-hyponyms of the given concept[28].

The bias towards "taxonomic similarity"[29] of the APT distributional spaces is further highlighted by Figure 4.6 which is showing a precision/recall curve of ranking taxonomically similar lexemes above topically related ones.

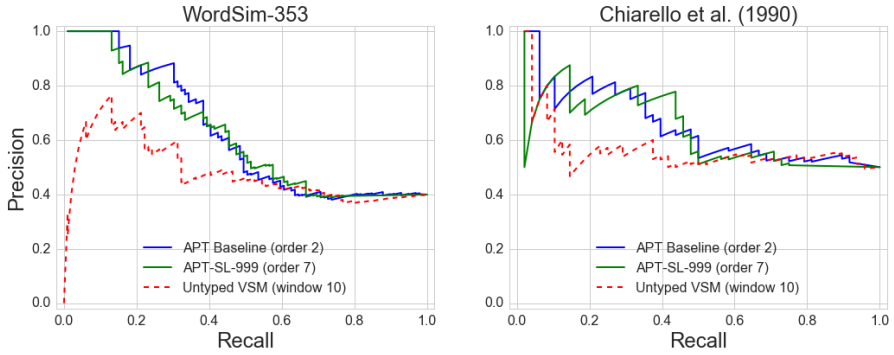


Figure 4.6: Precision / Recall curves on ranking the taxonomically similar word pairs above the topically related ones on the WordSim-353 and Chiarello et al. (1990) datasets.

The word pairs are ranked by their corresponding cosine similarity where the similarity between each word pair represents a threshold at which the precision and recall of the model are evaluated. For ex-

28 Indeed a quick check of the highest scoring PPMI features of 3 BLESS concepts (*pub*, *train*, *lettuce*) and their co-hyponyms reveals some characteristic commonalities, especially with regards to their function. For example the lexemes *restaurant*, *pub* and *cafe* all share features related to *eating* and *drinking*, whereas the lexemes *train*, *car* and *bus* share features related to *driving* and *transportation*, and the lexemes *lettuce*, *celery* and *cucumber* co-occur frequently with other vegetables.

29 I am using the phrase "taxonomic similarity" in the sense of any given model's neighbourhood being predominantly governed by lexemes that are taxonomically close, i.e. hypernyms, hyponyms or co-hyponyms.

ample, if the highest cosine similarity between any word pair is 0.6, then this value represents the first threshold where all pairs $\geq 0.6$ are classified as taxonomically similar and all pairs $< 0.6$ are classified as topically related. This results in the first precision and recall value that can be plotted on the curve. Then, the second highest cosine similarity is used as the next threshold value, and the process subsequently repeats for all word pairs.

Following Levy and Goldberg (2014a), I use the WordSim-353 similarity and relatedness subsplits of Agirre et al. (2009) as well as the dataset of Chiarello et al. (1990), containing pairs of words exhibiting taxonomic similarity and topical relatedness[30]. Both of the WordSim-353 subsplits share the low similarity word pairs, which are neither taxonomically similar, nor topically related, and which therefore are removed prior to creating the precision/recall curve. I furthermore removed the pair *tiger - tiger* from the WordSim-353 similarity subset.

Figure 4.6 shows that both the order 2 APT baseline model as well as the order 7 APT-SL-999 model exhibit a bias towards ranking taxonomically similar lexemes above topically related ones for both datasets. In comparison, the red dashed line shows an untyped distributional semantic VSM[31] with a window size of 10, which is showing a stronger tendency towards ranking topically related lexemes above taxonomically similar ones. While the effect is lesser on the Chiarello et al. (1990) dataset due to its small size, it is very pronounced on the WordSim-353 dataset.

Another observation in Figure 4.6 is that increasing the order of the APTs does not appear to have the effect of biasing the representations away from taxonomic similarity and more towards topical relatedness as observed in untyped vector space models (Peirsman, 2008; Levy and Goldberg, 2014a). The precision/recall curve of the order 7 APT-SL-999 model closely follows the curve of the APT baseline model, and does not deviate much towards the curve of the untyped VSM.

Further evidence for the vastly different distributional spaces between the APT models and the untyped VSM is provided by the low neighbour overlap scores between the two types of models. The neighbour overlap among the top 100 neighbours of all lexemes in the WordSim-353 dataset between the APT-baseline model and the untyped VSM is 21%, between the APT-SL-999 model and the

30 The dataset also conatins a list of word pairs that exhibit both — taxonomic similarity and topical relatedness — which, however, I do not use in this study.

31 The model has been obtained from the same lowercased and lemmatised version of the BNC as the APT models.

untyped VSM is 17%, and between the APT-WS-MEN model and the untyped VSM is only 14%.

*Estimating the Semantic Relations of Nearest Neighbours*

In the absence of labelled gold standard data, quantifying the distribution of semantic relations of the *neighbours* of some lexeme is a more difficult problem. In this thesis, I am using WordNet (Fellbaum, 1998) in order to get a rough estimate of which semantic relation holds between a target concept and its neighbours. Given that any distributional neighbour of a concept might not be in WordNet, or might be in a far away branch in the WordNet hierarchy, a direct classification of a neighbour in any of the semantic relations under consideration is not a feasible option.

Thus, I am applying a more indirect way of determining which semantic relation any given neighbour has to its target concept. This is achieved by comparing every neighbour of a given BLESS concept to the average similarity of all hypernyms, hyponyms, meronyms, and direct co-hyponyms of the target concept, across all of the target concept's synsets[32]. The semantic relation with the highest average similarity to any given neighbour is tallied up and collected into a histogram.

The target concepts are taken from the BLESS dataset and for each target concept the top 10 neighbours are analysed. Characterising the *neighbours* of a concept, rather than some pre-defined relata as in the BLESS dataset, is important for estimating what kind of knowledge will be inferred from the distributional neighbourhood in Chapter 5.

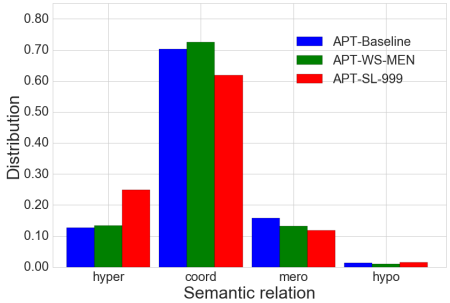


Figure 4.7: Histogram of the semantic relations that the nearest neighbours of the BLESS concepts exhibit.

32 The BLESS dataset does not contain any synset information, hence it is necessary to take all synsets of a concept into account.

Figure 4.7 shows the distribution of neighbours for the APT baseline model as well as APT-WS-MEN and APT-SL-999 models. While there is a relatively low neighbour overlap between the APT-WS-MEN and APT-SL-999 models with the APT-baseline space (37% with the APT-WS-MEN space and 60% with the APT-SL-999 space, respectively), the distributions in Figure 4.7 are relatively similar among the three different spaces. This suggests that while the neighbours themselves can vary considerably, the distribution of semantic relations they have to the given target concept remains stable. Interestingly, nearest neighbours are very rarely hyponyms of the target concept. One explanation for this effect is that hyponyms represent more specialised words and are thus often less frequent.

### 4.3.2 *The Distributional Semantics of Offset APT representations*

Offset APT-representations are the key ingredient for distributional composition, as the offset procedure aligns two otherwise incompatible representations with each other. In this section, I will investigate what offset APTs represent, in what neighbourhood they are embedded, and more generally, what preference (if any) towards specific semantic relations they exhibit.

The focus in this section is on 3 prominent first-order order offset paths: `amod`, `dobj` and `nsubj`. The `amod` offset study uses a number of frequent adjectives and creates respective noun offset views. For example, when creating the noun view $white^{\texttt{amod}}$ for the adjective *white*, $white^{\texttt{amod}}$ represents a "thing that can be *white*". The `dobj` and `nsubj` offset views create a verb view from a noun. For example the `nsubj` offset representation of *father*, $father^{\texttt{nsubj}}$, represents a typical action carried out by a *father*.

For qualitatively investigating the neighbours of offset APTs, I have created `amod` APT offset representations from all adjectives in the adjective-noun subset of the Mitchell and Lapata (2010) dataset, and manually added[33] 20 additional adjectives. These include antonyms, such as *old* and *new*, or *boring* and *exciting*, as well as some common colour terms such as *red*, *green* or *blue*. Furthermore, I have created `dobj` and `nsubj` offset representations for all nouns that appear as direct objects or subjects in the datasets of Mitchell and Lapata (2008, 2010) and Kartsaklis and Sadrzadeh (2014).

33 See Appendix A.

| Offset Representation | Neighbours |
|---|---|
| ancient$^{amod}$ | monument, civilization, legend, temple, rome |
| black$^{amod}$ | dark$^{amod}$, red$^{amod}$, blue$^{amod}$, green$^{amod}$, jacket |
| pretty$^{amod}$ | ugly$^{amod}$, clever$^{amod}$, smart$^{amod}$, blonde$^{amod}$, sexy$^{amod}$ |
| hot$^{amod}$ | cold$^{amod}$, stove, snack, tea, cylinder |
| right$^{amod}$ | left$^{amod}$, good$^{amod}$, gentleman, friend, better$^{amod}$ |
| door$^{dobj}$ | unlock, shut, slam, push, fling |
| father$^{dobj}$ | mother$^{dobj}$, parent$^{dobj}$, wife$^{dobj}$, family$^{dobj}$, girl$^{dobj}$ |
| gentleman$^{dobj}$ | assure, refer, remind, congratulate, thank |
| book$^{dobj}$ | letter$^{dobj}$, read, example$^{dobj}$, word$^{dobj}$, publish |
| requirement$^{dobj}$ | satisfy, need$^{dobj}$, demand$^{dobj}$, meet, fulfill |
| tongue$^{nsubj}$ | dart, lick, flick, tooth$^{nsubj}$, twist |
| researcher$^{nsubj}$ | author$^{nsubj}$, graduate$^{nsubj}$, discover, conclude, writer$^{nsubj}$ |
| father$^{nsubj}$ | mother$^{nsubj}$, parent$^{nsubj}$, wife$^{nsubj}$, girl$^{nsubj}$, family$^{nsubj}$ |
| gentleman$^{nsubj}$ | appreciate, acknowledge, recall, doctor$^{nsubj}$, aware |
| hand$^{nsubj}$ | clasp, hand$^{dobj}$, caress, eye$^{nsubj}$, clutch |

Table 4.9: 5 nearest neighbours of `amod`, `dobj` and `nsubj` representations, using the APT baseline model.

Table 4.9 shows the 5 nearest neighbours, according to their cosine similarity, of a number of `amod`, `dobj` and `nsubj` offset representations for the APT baseline model .

The noun offset view for the lexeme *ancient*, *ancient*$^{amod}$, represents a typical "thing that can be *ancient*", with neighbours easily associated with the property *ancient*. Neighbours of offset APT representations are frequently other offset APTs, showing that *black things* are most similar to *dark things*, but also to *red*, *blue* and *green things*, providing empirical evidence that "things that can be *coloured*"[34] share a considerable amount of features. The two nearest neighbours for the lexemes *pretty* and *hot*, respectively, show that the woes of distributional semantic models, of having antonyms as close neighbours, are extended to offset APT-representations. Another interesting case is the ambiguous lexeme *right*, where the neighbours of its `amod` offset representation cover the directional (or political) sense, with neighbours such as *left*$^{amod}$, as well as the moral sense ("doing the *right* thing"), with neighbours such as *good*$^{amod}$ and *better*$^{amod}$.

The group of `dobj` offset APTs, representing "actions typically *done to* a concept", and the group of `nsubj` offsets, representing "actions *carried out by* a concept" in Table 4.9 exhibit a similar and often comple-

34 More precisely, "things for which *colour* is typically mentioned in the text". For example, while *red onion* has several mentions in the BNC, *brown onion* does not occur. This means that distributional models tend to pick up atypical properties more frequently than typical ones that would not require a mention in the text but are *presupposed*.

mentary behaviour. For example actions done *to* a *father* and actions done *by* a *father* tend to be similar to actions done *to* and *by* a *mother*, *parent* or *wife*. Interestingly a *gentleman* typically tends to be *assured*, *referred to*, *reminded*, *congratulated* and *thanked*, while he himself tends to *appreciate* and *acknowledge* things. Another interesting example are the neighbours of $hand^{\texttt{nsubj}}$, where an action done *by* a *hand*, is very similar to an action done *to* a hand, as the offset view $hand^{\texttt{dobj}}$ among the top neighbours of $hand^{\texttt{nsubj}}$ shows.

Table 4.9 furthermore highlights that a single APT representation can give rise to different offset views and is not restricted to a static vector space. In terms of the neighbour overlap between different APT models, I compared the neighbourhoods of the offset representations of the APT baseline model to the APT-WS-MEN and APT-SL-999 models, respectively. Following the trend from just standard elementary APT representations, the overlap between the baseline and APT-WS-MEN was 31%, whereas the overlap between the baseline and the APT-SL-999 model is 53%. While the overlap among the offset representations is slightly lower than that of the elementary APT representations, a substantial amount of the distributional neighbourhood of offset views is shared across different APT parameterisations.

For a quantitative evaluation of APT offset representations, I am using the BLESS dataset. Instead of taking the target concepts as they are, I am extracting the most frequent adjectival modifiers for each target concept from the BNC and retaining target concepts where the adjectival modifier occurs at least 50 times with the target concept in the BNC[35]. Any target concept might be represented by more than one offset if it occurs with more modifiers of frequency $\geq 50$ in the BNC. Each target concept is subsequently represented by its `amod` offset representation.

For example, if the most frequent adjectival modifier for the target concept *bear* would be *polar*, then *bear* would be represented by $polar^{\texttt{amod}}$ — a *polar thing*. The underlying question I am addressing is, if a target concept were to be represented by a noun view of its most frequent modifiers, would the APT space still be biased towards co-hyponymy, or would there be a shift towards a different semantic relation such as meronymy? While the task might not be perfect for characterising offset APTs, as for example a *polar thing* is quite dif-

35 Appendix B contains a list of all modifiers used for this study.

ferent from a *bear*, it nonetheless provides a first estimation of what semantic relation offset APTs prefer.

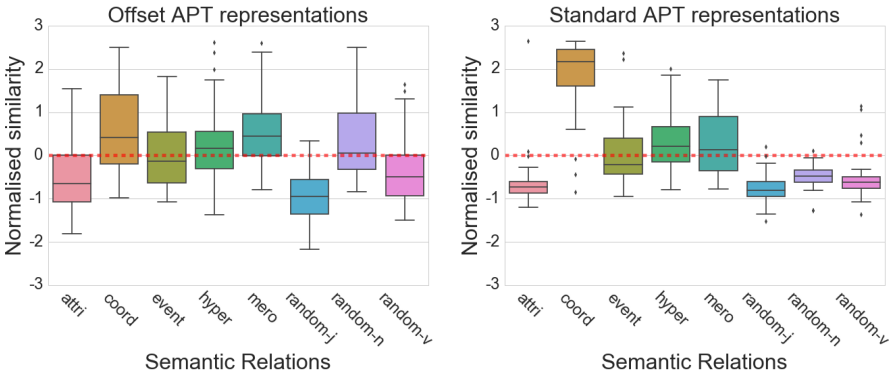


Figure 4.8: Comparison of the distribution of similarities between target concepts represented as the offset representations of their most frequent modifiers (left), and target concepts represented with "standard" elementary APT representations (right), using the baseline model.

Figure 4.8 shows a comparison between the similarity distributions of semantic relations of offset representations (left) and "standard" elementary representations (right) from the same subset of target concepts, using the APT baseline model. The offset representations are less biased towards co-hyponyms. In general, the resulting space appears to be less coherent and more fuzzy, judging from its relatively higher similarity to random concepts. Furthermore, due to using the `amod` offset view of the given noun, the space is more tipped towards meronymy and attributes, which will be further investigated in Chapter 5.

The results on the BLESS dataset suggest that offset APT representations offer a complementary view of a given lexeme, and might be used for biasing a representation towards a specific semantic relation as the offset inference algorithm proposed in Chapter 5 will show.

### 4.3.3 *The Distributional Semantics of Composed APT Representations*

One goal of distributional composition is to extend the continuous model of meaning from the word level to the phrase level. This opens the potential for inferences about paraphrases and entailment relations in compositional distributional semantic models. This section aims to provide a preliminary overview about the distributional neighbourhood of composed representations that APTs give rise to. Further-

more, the question whether composed APT representations are still biased towards co-hyponymy is addressed.

For investigating the distributional neighbourhood of composed APTs, I am using the ML2010 dataset, consisting of 108 each of adjective-noun, noun-noun and verb-object compounds, 324 phrase pairs in total. Table 4.10 lists the 5 nearest neighbours of a number of adjective-noun, noun-noun and verb-object phrases, using the APT baseline model and *composition by intersection*. Interestingly, most neighbours exhibit some degree of topical relatedness to the query phrase as in the case of the adjective-noun compound *federal assembly* which is among other government related terms and phrases. Other examples of the topical coherence of composed APT representations are the noun-noun compound *health minister*, which is embedded between other governmental secretaries, such as *environment secretary*, and the verb-object phrase *raise head*, which is located in a "body movement" neighbourhood.

| **Phrase** | **Neighbours** |
|---|---|
| elderly woman | elderly lady, older man, old person, pensioner, carer |
| federal assembly | legislature, assembly, presidency, state control, bureaucracy |
| vast amount | large quantity, quantity, wealth, bulk, amount |
| hot weather | weather, cold air, pants, sunshine, vacation |
| further evidence | evidence, indication, particular case, explanation, consideration |
| health minister | defence minister, health service, environment secretary, education officer, government leader |
| tax credit | tax charge, credit, tax rate, exemption, compensation |
| league match | match, football club, fixture, game, football |
| bedroom window | kitchen door, window, door, suite, floor |
| family allowance | allowance, housing benefit, tax credit, pension, motto |
| send message | relay, convey, communicate, send, delete |
| buy land | sell property, buy home, leave house, purchase, cultivate |
| consider matter | discuss issue, address question, complicate, consider, express view |
| raise head | lift hand, wave hand, stretch arm, raise, close eye |
| pose problem | pose, present problem, face difficulty, anticipate, require attention |

Table 4.10: 5 nearest neighbours of composed adjective-noun, noun-noun and verb-object phrases. All phrases have been composed with composition by intersection using the APT baseline model.

While the topical coherence of the phrasal neighbours might be an artifact of the ML2010 dataset itself, it is nonetheless an interesting

observation that the general topic of a phrase is preserved. Furthermore, Table 4.10 highlights the contextualisation capabilities of an intersective composition function. For example, due to its composition with *weather*, the lexeme *hot* is disambiguated and promotes contexts such as *sunshine* and *vacation* instead of neighbours associated with its meaning in *hot sauce*. The same effect can be observed for the phrase *raise head*, where *raise* is fully embedded in a "body movement" neighbourhood, where contexts associated with the financial sense of *raise*, such as in *raise money*, are suppressed.

The table shows that the APT model is able to capture paraphrases as close neighbours, as in the case of *fixture*, as a close neighbour of the phrase *league match*. Further examples are the verb *communicate* as a neighbour for *send message*, and *legislature* as the nearest neighbour of the phrase *federal assembly*.

As a first step towards studying the entailment characteristics of Anchored Packed Trees, I am extracting the most frequent adjectival modifiers of all target concepts from the BLESS dataset that occurred at least 50 times with the target concept in the BNC[36]. Subsequently, I compose the target concept with the respective extracted modifiers and represent the BLESS concept as the composed construct. For example, if *polar* is extracted as a modifier for the concept *bear*, it would be represented by the adjective-noun phrase *polar bear*.

Most of the extracted modifiers fall into the class of subsective adjectives, hence the resulting adjective-noun compound represents a more specific concept than the noun by itself (Kamp and Partee, 1995). The question I am addressing is whether a composed adjective-noun phrase is able to retain the characteristics of its head noun in terms of the distribution of semantic relations, or whether it distorts the distribution towards random behaviour.

Figure 4.9 shows a comparison of the distributions of semantic relations between the composed APT representations (left) and the corresponding subset of concepts represented by "standard" elementary APTs (right), using the APT baseline model.

These experiments provide preliminary evidence that adjective-noun composition — at least when the adjectives are subsective — retain the characteristics of the distributional space of the head noun, while being able to retrieve paraphrases from the distributional space as shown in Table 4.10. Hence, composition in Anchored Packed

36 See Appendix B for a complete overview of the BLESS subset used in this study.

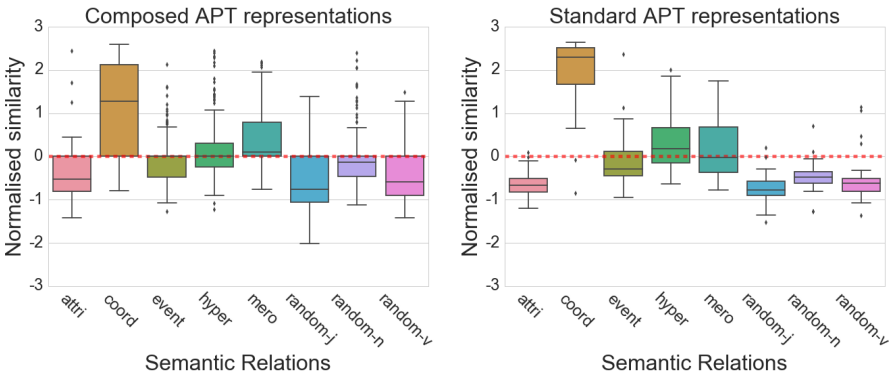


Figure 4.9: Comparison of the distribution of similarities between target concepts, composed with their most frequent adjectival modifiers (left), and target concepts represented with "standard" elementary APT representations (right), using the APT baseline model.

Trees has the potential to successfully extend the continuous model of meaning from individual words to phrases, as well as contextualising the meaning of a given lexeme in some phrase.

## 4.4 SUMMARY

This chapter has presented an empirical evaluation of the APT theory on word similarity and short phrase composition tasks. The chapter also contributed a characterisation of the distributional semantic space that Anchored Packed Trees give rise to. It has furthermore been shown that the choice of hyperparameters is a task-dependent problem, and does not transfer from word similarity to composition. In particular

- Optimising the hyperparameters in a task specific manner can significantly improve performance.
- While the performance of elementary APT representations and for composition by union could be significantly improved through optimising the various model hyperparameters, the performance of composition by intersection remained poor.
- The distributional space of elementary and composed APT representations is predominantly governed by co-hyponyms.
- Offset APT representations can encode higher-order semantic relations between lexemes and can offer a complementary view of a given lexeme.

- Composed APT representations retain a strong topical coherence, primarily caused by the contextualisation effect of an intersective composition function.
- Distributional composition in APTS can extend the continuous model of meaning to the phrase level, as well as achieve contextualisation of ambiguous lexemes.
- The distributional semantics of offset APT representations capture a number of interesting higher-order semantic phenomena, such as *black things* being similar to *dark things*, and actions carried out by a *father* being similar to actions carried out by a *mother*.

One central observation in this chapter was that even with extensive hyperparameter tuning, the performance of composition by intersection on short phrase composition tasks remained poor. The next chapter will investigate the reason for the weak performance and propose a method for improving composition by intersection that retains its desirable effect of contextualising the meaning of the lexemes in some phrase.

# 5

# INFERRING UNOBSERVED CO-OCCURRENCE EVENTS IN ANCHORED PACKED TREES

This chapter investigates the problem of data sparsity in the context of Anchored Packed Trees. Data sparsity leads to incomplete elementary representations due to not observing all plausible co-occurrences for any given lexeme. This problem is amplified for intersective composition functions, that rely on observing plausible co-occurrences between *all* lexemes in a phrase. In addition, an algorithm for explicitly inferring unobserved co-occurrence events is proposed, and its utility for improving elementary APT representations and distributional composition is shown. The algorithm is generalised in order to leverage the rich type structure in APTS, yielding further performance improvements for distributional composition. Lastly, the complementary nature between distributional inference and distributional composition is highlighted.

The following chapter is based on, and extends, the work published in Kober et al. (2016), which leveraged the idea of distributional inference of Dagan et al. (1993) to improve elementary distributional semantic word representations and distributional composition, and Kober et al. (2017a) which proposed the generalisation of distributional inference to offset inference and highlighted the similarity between distributional composition and distributional inference.

The previous chapter has shown that considerable improvements can be gained from optimising the hyperparameters for a given APT model. This chapter shows that even well tuned models suffer from the problem of not observing all plausible co-occurrences. This chapter shows that statistically significant improvements can be achieved by leveraging the distributional neighbourhood and explicitly inferring unobserved co-occurrence events in the distributional space.

The contributions of this chapter are:

- An analysis of the issue of data sparsity and its consequences for Anchored Packed Trees.

- The proposal of *distributional inference* — an unsupervised algorithm for explicitly inferring unobserved co-occurrence events in distributional semantic models that extends the algorithm of Dagan et al. (1993) to a mechanism for augmenting elementary word representations with unseen but plausible co-occurrence events.
- The subsequent generalisation of distributional inference to *offset inference*, in order to leverage the type structure in APTs.
- A characterisation of the kind of knowledge that can be learnt from the distributional neighbourhood as well as a qualitative and quantitative assessment of how the distributional inference algorithms affect the semantic space of APTs.
- An empirical validation of the algorithm on a range of word similarity tasks and a short phrase composition benchmark dataset, showing that distributional inference is successfully closing the performance gap between low-dimensional uninterpretable models and high-dimensional interpretable models on the short phrase composition task.
- An analysis of the relation and complementary nature between distributional composition and distributional inference.

This chapter is structured as follows: the issue of data sparsity is discussed in Section 5.1, followed by the proposal and analysis of the distributional inference algorithm in Section 5.2. Subsequently, the standard distributional inference algorithm is generalised to offset inference in Section 5.3. Finally, Section 5.4 discusses the relation between distributional inference and distributional composition.

## 5.1 THE ISSUE OF DATA SPARSITY

Data sparsity is the problem of not observing all plausible co-occurrences that a lexeme might co-occur with. For example, while the two lexemes *bike* and *bicycle* might be used interchangably in many cases, their co-occurrence with possibly independent contexts leads to unobserved co-occurrences for *both* lexemes. Indeed, in the distributional space of the APT baseline model, a *bicycle* is never observed as being *bought*, *used* or *stolen*, whereas a *bike* is.

The missing information in the distributional representations leads to less feature overlap when the similarity of two lexemes is com-

puted. This in turn has the consequence that the similarity between two given words is frequently *underestimated*, or even results in no feature overlap at all.

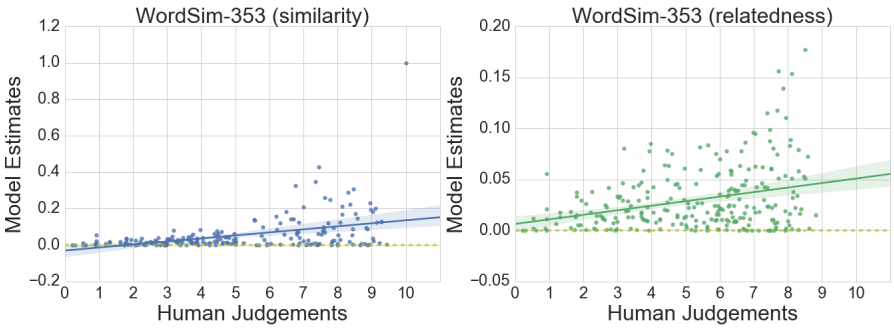


Figure 5.1: Scatterplots of human judgements in comparison to model estimates on the WordSim-353 similarity (left) and relatedness (right) subsplits using the APT baseline model.

Figure 5.1 show the scatter plots resulting from the APT baseline model on the WordSim-353 similarity (left) and relatedness (right) subsets between the similarity estimates of the human annotators ($x$-axis) and the distributional model ($y$-axis).

The plots highlight that even for many word pairs given high similarity scores by the human annotators, the model estimates are often very close to 0. This is highlighted by the large proportion of dots along the $x$-axis in Figure 5.1. Therefore, missing information about many plausible co-occurrence events is likely the primary factor that is causing the weak performance of the APT models on the word similarity tasks.

The problem is amplified when composing two distributional word representations with an intersective composition function because two incomplete representations are used to build a phrasal representation. The result is a phrasal representation that is missing a large proportion of plausible features. This in turn leads to increasingly semantically incoherent behaviour of the composed constructs that manifests in low similarity estimates between similar phrases.

Figure 5.2 shows the distribution of model similarity estimates for each step on the 1-7 Likert scale as rated by human annotators, for composition by union (left) and composition by intersection (right) on the adjective-noun subtask of the Mitchell and Lapata (2010) dataset. While the distributions of similarity estimates are monotonically increasing with the human judgments for composition by union, the estimates for an APT model using composition by intersection are levelling off and remain constant from a human similarity rating of 4

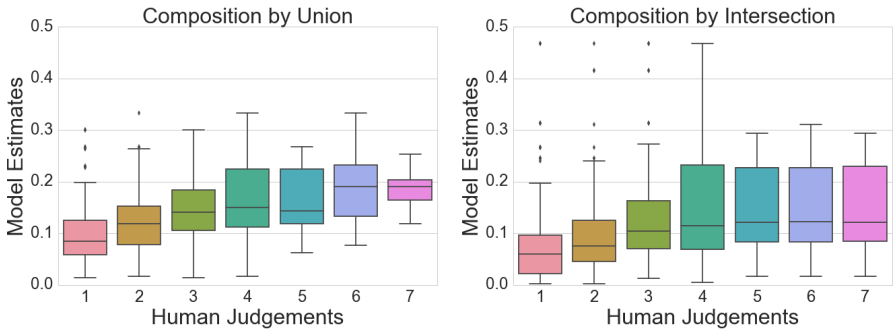


Figure 5.2: Distribution of similarity estimates by the APT baseline model in comparison to human judgements on the ML2010 adjective-noun subtask for composition by union (left) and composition by intersection (right).

onwards. Even phrase pairs judged with a similarity of 6 or 7 by the human annotators frequently receive very low similarity estimates by the model, as the corresponding whiskers in Figure 5.2 show. The reason for this problem with model estimates using composition by intersection is the larger impact of data sparsity due to the discriminative nature of the composition function as the comparison to the same APT model, using composition by union, in Figure 5.2 shows.

## 5.2 IMPROVING SPARSE APT REPRESENTATIONS WITH DISTRIBUTIONAL INFERENCE

The previous section has identified data sparsity as one of the root causes for the weak performance of APTs on word similarity, and in particularly in conjunction with composition by intersection, on composition tasks. This section proposes a simple algorithm that infers missing co-occurrence information by leveraging the distributional neighbourhood of a lexeme. Furthermore, the algorithm generalises earlier approaches by Kintsch (2001); Utsumi (2009, 2012) which can be represented as special cases of the proposed algorithm.

In the following, after the distributional inference (DI) algorithm will be described, the kind of knowledge that can be inferred will be characterised in Section 5.2.1. Subsequently, Section 5.2.2 shows the positive effect of using distributional inference on the word similarity datasets and short phrase composition tasks, yielding statistically significant improvements, and being especially beneficial to composition by intersection. A detailed analysis of the properties of distributional inference is presented in Section 5.2.3, showing that the performance

improvements come from inferring missing information rather than just mitigating sparsity. Section 5.2.4 analyses how much data distributional inference can make up for, followed by Section 5.2.5 which is putting the DI algorithm in context with earlier approaches. Finally, Section 5.2.6 assesses the limitations of distributional inference.

Algorithm 1 below outlines how distributional inference works. The input to the algorithm is a source distributional model $M$, the representation for the lexeme $w$ on which distributional inference will be performed, and the number of neighbours to consider $k$. The algorithm returns an enriched version of $w$, $w'$, as output. The distributional features in $w$ are scaled by the number of neighbours $k$ (see line 2 in Algorithm 1) to prevent the representation for the original lexeme $w$ from being overwhelmed by the information inferred from its neighbours. The algorithm subsequently loops through all of the $k$ neighbour representations of $w$ in $M$, and then merges them with $w'$.

**Algorithm 1** Distributional Inference

```
1: procedure DISTRIBUTIONAL_INFERENCE(M, w, k)
2:     w' ← w × k
3:     for all n in neighbours(M, w, k) do
4:         w' ← merge(w', n)
5:     end for
6:     return w'
7: end procedure
```

The time complexity of the algorithm broadly follows that of the $k$-Nearest Neighbours algorithm (Manning et al., 2008). It linearly depends on the size of the vocabulary $|V|$, the number of lexemes for which to perform distributional inference for $q$, the number of neighbours $k$, and the time it takes to calculate the similarity between two items $s$. Merging two elements is dependent on the dimensionality of the distributional space. However given a sparse representation of the distributional APT space, a tighter bound can be achieved by using the average number of non-zero features per lexeme $d_{\text{Avg}}$.

Each iteration of the main loop in Algorithm 1 requires $\mathcal{O}(|V|\, s + d_{\text{Avg}})$ runtime, where $\mathcal{O}(|V|\, s)$ is the time it takes to retrieve a neighbour, and $d_{\text{Avg}}$ is the time required to merge two elements. The main loop is executed $n$ times for $q$ words, thus the overall runtime of the algorithm can be estimated as $\mathcal{O}(k\, q\, (|V|\, s + d_{\text{Avg}}))$.

The largest impact on runtime is the time it takes to calculate the similarities of a query word to all other lexemes in the distributional

space, which is dependent on the size of the vocabulary $|V|$. In order to speed up the algorithm, it is possible to calculate all sorted pairwise similarities upfront, and thereby reduce the work of the main loop to a lookup operation, requiring constant time, and the merge operation, requiring $\mathcal{O}(d_{\text{Avg}})$. Overall this would result in a runtime of the algorithm of $\mathcal{O}(k\,q\,d_{\text{Avg}})$[1].

Distributional inference can be seen as a non-parametric soft-clustering algorithm, where every cluster is formed by the given lexeme $w$ as the centroid, and its distributional neighbours as members of the cluster. However, any lexeme can be part of any number of clusters, differentiating it from hard-clustering methods such as $k$-means. Every lexeme for which distributional inference has been performed is subsequently represented as a weighted average of its respective cluster.

The algorithm is agnostic to the distributional model $M$ used for querying neighbours. For example, it is possible to use a `word2vec` model to infer knowledge for an APT model, or even have an ensemble of different source distributional models to query the neighbours. Throughout this thesis, however, I will use the same APT model for querying neighbours and for which to perform distributional inference for.

Algorithm 1 has two degrees of freedom: the method to query neighbours for a given lexeme and the method to merge the inferred information into a given word representation. For the former I will use the "static top $n$" neighbour retrieval function that uses the top $n$ neighbours of any lexeme for inference. This method has been found to consistently perform well across tasks in (Kober et al., 2016), while at the same time requiring only the number of neighbours $n$ to be tuned for the task at hand. For the latter, I will use pointwise addition to merge the aligned distributional features of two or more word representations.

### 5.2.1 *What kind of knowledge can be inferred?*

Section 4.3 characterised the distributional space of APTs, and it was shown that the distributional APT space is predominantly governed by co-hyponymy. Any inferred co-occurrence events for a given lex-

1 The constant factor $\mathcal{O}(1)$ is subsumed by the time required by the larger factor $\mathcal{O}(d_{\text{Avg}})$ in the main loop, allowing the runtime to be estimated as $\mathcal{O}(k\,q\,d_{\text{Avg}})$ instead of $\mathcal{O}(k\,q + k\,q\,d_{\text{Avg}})$.

eme will therefore primarily be drawn from its co-hyponym neighbours.

| Lexeme | Neighbours | Inferred Co-occurrences |
|---|---|---|
| magazine | **newspaper**, journal, papers | $\overline{\texttt{dobj}}$:*sell*, $\overline{\texttt{nsubj}}$:*report*, amod:*daily* |
| cafe | **pub**, restaurant, lounge | $\overline{\texttt{dobj}}$:*leave*, $\overline{\texttt{nsubj}}$:*close*, amod:*famous* |
| cat | **dog**, rabbit, pet | $\overline{\texttt{dobj}}$:*walk*, $\overline{\texttt{nsubj}}$:*bark*, amod:*hot* |
| car | **vehicle**, lorry, bus | amod:*four-wheel*, amod:*horse-drawn*, amod:*military* |
| house | building, **room**, home | $\overline{\texttt{dobj}}$:*brighten*, $\overline{\texttt{dobj}}$:*book*, amod:*stuffy* |

Table 5.1: Example co-occurrence inferences from the boldfaced neighbour for a given lexeme on the basis of the APT baseline model. The illustrated features have been observed with the (boldfaced) neighbours, but not with the target lexeme itself. For the lexemes *magazine*, *cafe* and *cat*, inferred co-occurrences from co-hyponym neighbours are exemplified, the lexeme *car* shows example inferences from a hypernym neighbour, and for the lexeme *house* example inferences from a meronym neighbour are shown.

Table 5.1 shows the nearest neighbours of a number of lexemes, together with co-occurrences that have been observed with the neighbours, but not with the lexeme itself. The table shows that leveraging co-occurrence events from co-hyponyms can lead to many plausible inferences, for example for the lexeme *magazine*, one would learn that they can be *sold*, that they *report* things and that they might be published on a *daily* basis, by observing its top neighbour *newspaper*. By considering the nearest neighbour for *cafe*, which is one of its co-hyponyms *pub*, it can be inferred that *cafes* can be *left*, that they *close*, and that they might be *famous*.

However, inferring co-occurrence events from co-hyponyms can also lead to implausible inferences. For example for the lexeme *cat*, whose nearest neighbour is its co-hyponym *dog*, one would infer that *cats* might be taken for a *walk*, that they are able to *bark* and that they might be *hot*[2]. While co-hyponyms share a large number of distributional properties, for example that it is possible to have lunch in both, a *cafe* and a *pub*, they are distinctive in many other respects. The current distributional inference algorithm, however, does not make any assumptions about what can plausibly be inferred from a neighbour and what represents an implausible feature for the given lexeme.

2 An artefact from many occurrences of *hot dog*, which has been parsed as an adjective-noun phrase.

An interesting case is the inferences that can be made for *car* from its nearest neighbour — a hypernym — *vehicle*. For example, it has not been observed in the BNC that *cars* might be *four-wheeled*, suggesting that distributional word representations are frequently missing obvious common-sense properties. A further example of common-sense information being missing from a distributional model is that neither *cats* nor *dogs* have *tails* in the APT baseline model. This stems from the fact that there are only 3 occurrences of the phrase *dog's tail*, and 2 occurrences of the phrase *cat's tail* in the BNC — not enough occurrences to pass through the PPMI filtering (and potentially not enough to even pass through the preprocessing stage).

As the neighbour *room* for the lexeme *house* shows, meronyms can also contribute useful co-occurrence information to a given lexeme. For example, it has not been observed in the BNC that a *house* might be *booked* or that it can be *brightened*, or maybe be *stuffy*.

*Estimating the Change in the Distributional Neighbourhood*

| **Lexeme** | 0 **neighbours** | 100 **neighbours** | 1000 **neighbours** |
|---|---|---|---|
| mug | tray, pot, glass, cup, gulp | jar, tray, glass, bottle, bowl | tray, jar, glass, pot, cup |
| rock | punk, stone, pop, roll, jazz | surface, jazz, sea, wood, hill | surface, punk, stone, sea, jazz |
| train | bus, taxi, boat, car, lorry | bus, car, journey, taxi, flight | bus, taxi, car, boat, journey |

Table 5.2: Nearest Neighbours for the APT baseline model with and without distributional inference. The increasing number of neighbours for DI highlights its effect on the distributional space.

Table 5.2 shows the nearest neighbours for the APT baseline model without distributional inference (0 neighbours) and with distributional inference, using 100 and 1000 neighbours, respectively. While the use of DI preserves the general nature of the distributional neighbourhood, it is able to shift the meaning of a given lexeme towards a different sense.

For example, for the lexeme *rock*, the semantics seems to shift from a "music" dominated topic to a more mixed "music" and "surface" related neighbourhood as distributional inference seems to predominantly promote the "surface" meaning of *rock*. Similarly, for the lexemes *mug* and *train* a refinement of its semantics is visible. Where

the meaning of *mug* seems to shift from a “drinking" related to a “glass container" related neighbourhood, the meaning of *train* seems to move from a “vehicle" related to a “travelling" related neighbourhood. An interesting effect happens for the lexeme *apple*, where distributional inference with 100 neighbours seems to promote a second sense, which however, gets suppressed again when using 1000 neighbours. When using 100 neighbours, DI blends the distributional neighbourhood with some more “fruitiness", which is weighted down when using 1000 neighbours.

The effects of topicality shifts that distributional inference cause are a consequence of reinforcing the predominant — or majority — sense of its nearest neighbours. For example, while many of the top neighbours for *train* are other vehicles, there is a large number of neighbours associated with the primary function of *trains* — travelling. This subsequently results in top neighbours such as *flight* or *journey* when distributional inference is applied.

Interestingly, the neighbour overlap scores on the words from the WordSim-353 dataset between a space with and without the use of distributional inference are generally higher than with different hyperparameterisations. For example, the overlap between the APT baseline model without DI and the same model with DI, using 100 neighbours, is 65%. Between the baseline model without DI and a model with DI and 1000 neighbours, the overlap is still 51%. Hence, the use of DI better preserves the characteristics of the distributional space in comparison to using a different hyperparameterisation of the APT space.

### *Characterising the Effect of Distributional Inference*

Inferring missing knowledge furthermore has a substantial effect on the distribution of semantic relations. Figure 5.3 compares the distribution of semantic relations on the BLESS dataset for the APT-WS-MEN model without distributional inference (left), and the same APT-WS-MEN model with DI, using the top 100 neighbours for the inference process. Whilst the tendency of the semantic space to favour co-hyponymy prevails, distributional inference is able to increase the similarities to hypernyms, and to a lesser extend meronyms, while preserving the semantic coherence of the space and keeping the median similarities to any random concepts well below 0.

The widening of the similarity distributions for relations other than co-hyponymy is caused by the fact that distributional inference is able

to add features associated with hypernyms or meronyms back into the elementary representations. These features have previously been filtered out due to applying a high negative SPPMI shift of $\log 40$.

The tightening of the similarity distribution for co-hyponyms can be explained by the same effect. The widening of the co-hyponym similarity distribution in Figure 5.3 (left) has been caused by applying a negative SPPMI shift of $\log 40$, which filtered out many features associated with co-hyponymy as well. This caused a wider spread in similarity scores as the elementary representations have been "thinned out". Distributional inference is able to add a large amount of features indicating co-hyponymy back into the representations which is causing the tightening of the similarity distributions for co-hyponyms in Figure 5.3 (right).

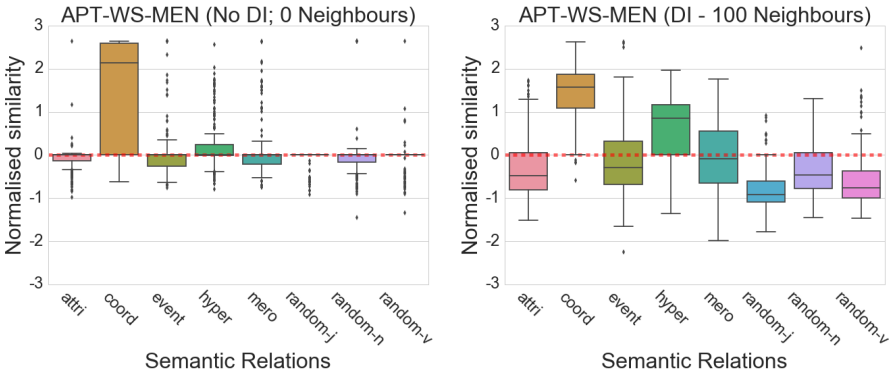


Figure 5.3: Distribution of semantic similarities in the BLESS dataset for the APT-WS-MEN model without distributional inference (left) and the same model with distributional inference (right) using 100 neighbours.

### 5.2.2 *Quantitative Analysis*

#### *Word Similarity*

Table 5.3 summarises the results of using distributional inference with the tuned APT-WS-MEN and APT-SL-999 models on the word similarity tasks in comparison to the standard APT baseline model without the use of DI, and the two tuned APT-WS-MEN and APT-SL-999 models without DI, respectively. The use of distributional inference can significantly improve performance on the optimised APT models, showing that the algorithm is able to successfully infer relevant and missing knowledge from the distributional neighbourhood. An overview of the optimal number of neighbours for each dataset for distri-

butional inference is listed in Table 5.8 further below. The number of neighbours for distributional inference for MEN has been tuned on its development set. For the other datasets, the number of neighbours was determined by 2-fold cross-validation and the results reported in Table 5.3 represent averaged Spearman $\rho$'s.

| | **WS353 (sim)** | **WS353 (rel)** | **MEN** | **SimLex-999** |
|---|---|---|---|---|
| Baseline (No DI) | 0.40 (+/- 0.14) | 0.24 (+/- 0.06) | 0.36 (+/- 0.01) | 0.22 (+/- 0.02) |
| Tuned APT models | $0.52^{\ddagger}$ (+/- 0.09) | $\mathbf{0.35}^{\dagger}$ (+/- 0.01) | $0.43^{\ddagger}$ (+/- 0.02) | 0.25 (+/- 0.01) |
| Tuned APT models + DI | $\mathbf{0.54}^{\ddagger}$ (+/- 0.06) | $\mathbf{0.35}^{\dagger}$ (+/- 0.06) | $\mathbf{0.48}^{\ddagger\spadesuit}$ (+/- 0.02) | $\mathbf{0.30}^{\ddagger\spadesuit}$ (+/- 0.01) |

Table 5.3: Results on the word similarity tasks for a standard APT baseline model, the tuned models (APT-WS-MEN for both WordSim-353 subtasks and MEN, and APT-SL-999 for the SimLex-999 dataset) without DI and the tuned APT models with distributional inference. Performance is reported in terms of averaged Spearman $\rho$ across 2-fold cross-validation. The numbers in parentheses denote the standard deviation across the two runs. Results marked with † and ‡ are statistically significant at the $p < 0.05$ and $p < 0.01$ level in comparison to the APT baseline model, respectively. Results marked with ♠ are statistically significant at the $p < 0.01$ level in comparison to the respective Tuned APT models. Statistical significance has been determined using the method of Steiger (1980).

Furthermore, the unimproved performance on the WordSim-353 (relatedness) subtask when using distributional inference suggests that simply alleviating the issue of sparsity alone does not guarantee improved performance. If the "wrong" distributional knowledge for the task at hand — features indicating taxonomic similarity rather than topical relatedness for WS353 (rel) — is inferred, then DI might not lead to improved performance.

### *Phrase Similarity*

The use of distributional inference has a particularly positive effect in conjunction with composition by intersection as the results in Table 5.4 show. The table shows a comparison between the hyperparameter-optimised APT models without DI for composition by intersection and composition by union, respectively, compared to the same model with the use of distributional inference.

The table shows that performance is vastly improved for composition by intersection, performing *on par* with composition by union, for which the performance remained at the same level. As the hyperparameter sensitivity study for the short phrase composition dataset in the previous chapter has shown, no tweaking of model parameters

| **Composition by Intersection** | **AN** | **NN** | **VO** | **Average** |
|---|---|---|---|---|
| Tuned APT model | 0.39 | 0.41 | 0.35 | 0.38 |
| Tuned APT model + DI | **0.48**‡ | **0.46**‡ | **0.44**‡ | **0.46**‡ |
| **Composition by Union** | **AN** | **NN** | **VO** | **Average** |
| Tuned APT model | **0.50** | **0.45** | 0.44 | **0.46** |
| Tuned APT model + DI | **0.50** | 0.44 | **0.45**† | **0.46** |

Table 5.4: Comparison between tuned APT models without DI and the same models with the use of distributional inference on the ML2010 composition task. Results marked with ‡ are statistically significant at the $p < 0.01$ level, and results marked with † are statistically significant at the $p < 0.05$ level, in comparison to the respective baseline without distributional inference for composition by union and composition by intersection. Statistical significance has been determined using the method of Steiger (1980).

can alleviate the fact that there is too much missing information in the elementary representations. This is subsequently leading to very little feature overlap in the composition process and distributional similarity estimation.

For composition by union on the other hand, data sparsity is less problematic as the composition function does not discard any features. This, however, leads to the problem where implausible features, either introduced by the composition process or by distributional inference, cannot be filtered out, thus decreasing the composition functions capability of performing semantic contextualisation. An overview of the optimal number of neighbours for distributional inference, which has been determined on the ML2010 development set is listed in Table 5.11 further below.

*Alleviating the Problem of Data Sparsity for Composition by Intersection*

Figure 5.4 shows the distribution of similarity estimates of the APT baseline model in a boxplot for each similarity band on the Likert scale from 1-7 for composition by intersection on the adjective-noun subtask of the ML2010 dataset. Without the use of distributional inference (left), there are a large number of phrase pairs with very few distributional features after composition, resulting in a large number of 0-similarity estimates and very low scores for even high similarity pairs.

The use of distributional inference is able to remedy this shortcoming as the right-hand side plot in Figure 5.4 shows. With distributional inference, the distribution of the APT model's similarity estimates is increasing with higher human similarity judgements. This res-

ults in substantially fewer 0-overlap comparisons, which for medium- and high-similarity pairs could be avoided altogether as the whiskers in Figure 5.4 (right) show. The additionally inferred knowledge furthermore leads to a significant improvement in Spearman $\rho$ correlation as shown in Table 5.4 above.

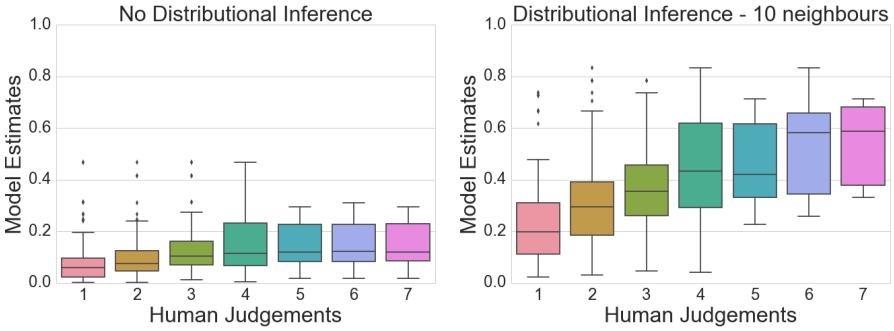


Figure 5.4: Distribution of similarity estimates by the APT baseline model in comparison to human judgements on the ML2010 adjective-noun subtask for composition by intersection without distributional inference (left) and composition by intersection with distributional inference (right) using 10 neighbours.

### 5.2.3 *Inferring Missing Knowledge vs. Reducing Sparsity*

An important question is whether the performance improvements of the APT models are due to inferring missing information, or whether they are simply an effect of the decrease in sparsity in the distributional space.

In order to address this question I am comparing the scatterplots of the similarity judgements between the human annotators and the APT models on the two WordSim-353 subtasks, with and without the use of distributional inference, in Figure 5.5.

The two top plots show how the number of (near) 0-overlap comparisons, resulting in a large number of dots along the $x$-axis corresponding to a model similarity estimate of close to 0, substantially decreases with the use of distributional inference on the WordSim-353 (similarity) subtask (5.5, top right). Furthermore, the inference of missing knowledge leads to an improvement of fit for the corresponding regression line in Figure 5.5 as well as to an improved Spearman $\rho$ correlation measure as shown in Figure 5.6 as well as Table 5.3 in the previous section.

For WordSim-353 (relatedness) on the other hand, distributional inference reduces sparsity, as shown by a substantial decrease in near 0

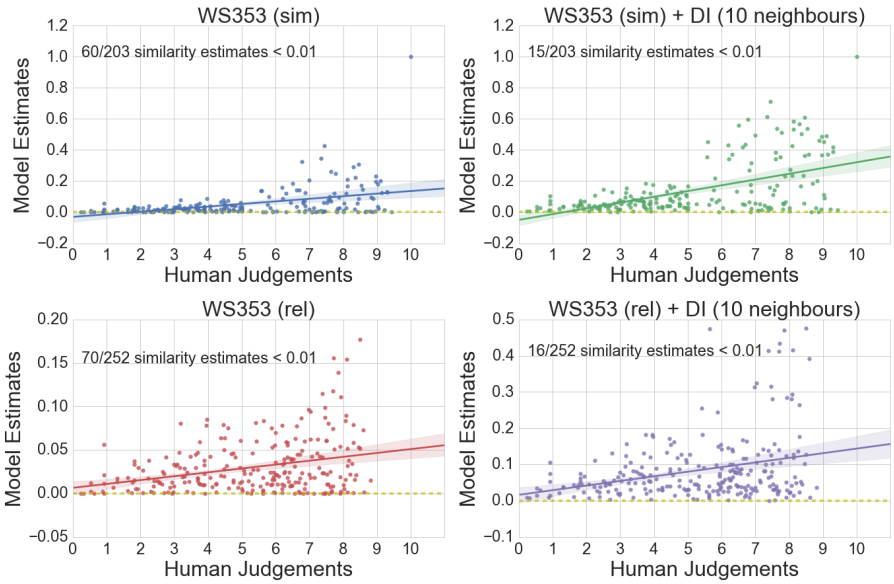


Figure 5.5: Scatterplots for WordSim-353 (similarity; top) and WordSim-353 (relatedness; bottom) showing the reduction of (near) 0-overlap comparisons after distributional inference (plots on the right) in comparison to no distributional inference (plots on the left), using the APT baseline model.

similarity estimates by the model, but is unable to improve the fit of the regression line, and therefore does not result in improved Spearman $\rho$ correlation. One reason for this behaviour is that the distributional APT space is primarily governed by co-hyponymy and inferring knowledge from a large amount of co-hyponym neighbours does not lead to representations better suited for a task favouring topical relatedness.

A further question relates to the number of neighbours. If the only factor that is improving performance would be a decrease in sparsity in the representations, then model performance must be linearly increasing with the number of neighbours. However, this is not the case.

Figure 5.6 shows the Spearman $\rho$ correlation on the ML2010 development set between the human similarity judgements and the model estimates as a function of the number of neighbours used for distributional inference for all 3 ML2010 subtasks (left), using composition by intersection, as well as the word similarity tasks (right) used in this thesis. The dashed lines show the model performance without the use of distributional inference, whereas the solid lines show the performance trajectory of an APT model with distributional inference.

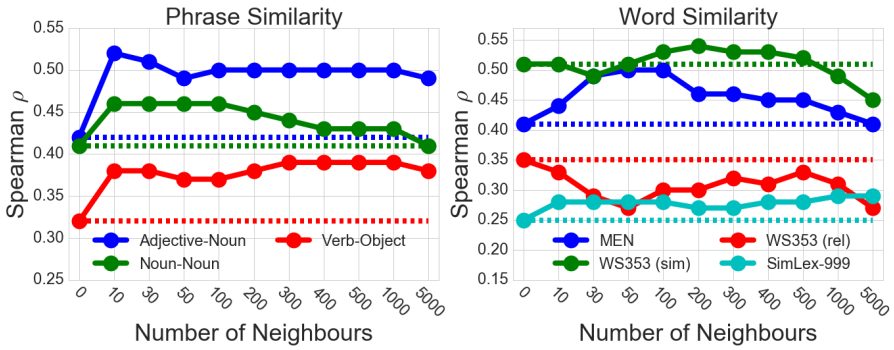


Figure 5.6: Spearman $\rho$ correlation between human similarity judgements and model estimates on the adjective-noun, noun-noun and verb-object subtasks of the ML2010 dataset (left), using the APT baseline model and composition by intersection, and the WS353 (sim), WS353 (rel), MEN and SimLex-999 word similarity tasks (right) as a function of the number of neighbours used for distributional inference, using the respective tuned APT models. The dashed lines represent model performance without the use of distributional inference.

The figure therefore highlights that the improved performance is due to inferring missing distributional knowledge rather than just a decrease in sparsity, because otherwise more neighbours would *always* lead to better performance. Every neighbour might contributed noise in addition to missing knowledge, however as long as the amount of actual knowledge being inferred is larger than the amount of noise added to the representations, the performance of distributional inference is better than the performance of an APT model without distributional inference. The peaks in the performance trajectories in Figure 5.6 show that there generally appears to be an optimal number of neighbours for a given task.

While composition by intersection appears to be relatively robust to overestimating the number of neighbours required due to the composition function's ability to filter out a substantial amount of noise, the performance trajectories for the word similarity tasks show that adding information from too many neighbours can hurt performance. This is supporting evidence that closer neighbours contribute actually missing information, while "oversmoothing" with too many neighbours is akin to add-1 smoothing and is decreasing sparsity but overflowing the representations with noise.

An interesting observation for the word similarity trajectories in Figure 5.6 is that certain bands of neighbours seem to be better suited for the given task than others. For example for the WordSim-353 (similarity) subtask the first 30 neighbours do not appear to contribute

much useful information, whereas the next ≈170 neighbours do. This characteristic opens the potential for a more sophisticated neighbour selection procedure in future work as briefly outlined in Section 6.3.2.

### 5.2.4 *How much Data can Distributional Inference make up for?*

One way to assess how much missing information distributional inference is able to contribute is to compare the performance of an APT space with distributional inference to APT representations without DI on samples of different size of the source corpus. In addition to the full BNC corpus, I created 20 independently drawn samples of size 1%, 5%, 10%, 25%, 50%, and 75% from the BNC and created order 2 APT representations, using the baseline configuration, for this experiment. Figure 5.7 shows the performance trajectories of the APT-baseline model on the MEN (dev) and WordSim-353 (relatedness) datasets[3]. The similarity judgements of each APT model for each sample have been concatenated and Spearman $\rho$ correlation has subsequently been calculated between the concatenated model estimates and the corresponding concatenated human similarity judgements[4].

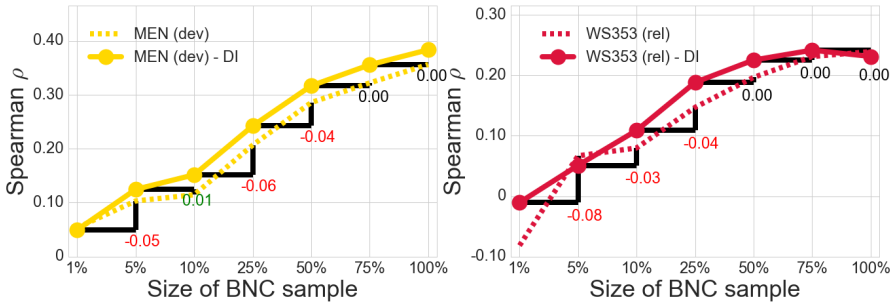


Figure 5.7: Spearman $\rho$ correlation between human similarity judgements and estimates by the APT baseline model on MEN (left) and WS353 (rel; right) on increasing sample sizes of the BNC. The dashed line is the performance without distributional inference, the solid line shows the model performance with the use of distributional inference.

The dashed lines show the performance of the APT-baseline model without the use of distributional inference and the solid lines show the Spearman $\rho$ correlation between the human judgements and the baseline model with distributional inference. For the MEN (dev) data-

3 These two datasets have been chosen because they best illustrate the contributions of distributional inference for the APT baseline model.

4 The human similarity judgements are identical for, and independent of, each sample, hence they have been duplicated and stacked in order to match the size of the concatenated model estimates.

set, the use of distributional inference consistently improves the fit between the human judgements and the model's similarity estimates. For the WS353 (rel) subtask, distributional inference has a positive effect for up to a sample size of 75% and then decreases when the whole BNC is used. The black angled lines illustrate the performance difference between an APT model *with* distributional inference at sample size $i-1$ and an APT model *without* distributional inference at sample size $i$, where $i$ indexes the list `[1, 5, 10, 25, 50, 75, 100]`, representing the sample sizes. For example, as shown in Figure 5.7, for the MEN (dev) dataset, the performance of the APT model with distributional inference for 50% of the BNC is approximately at level with the performance of an APT model *without* distributional inference at a sample size of 75% of the BNC.

Figure 5.8 shows the impact of distributional inference on all word similarity tasks with bootstrapped confidence intervals (Efron and Tibshirani, 1994) over samples of increasing size of the BNC. 20 samples were drawn for each size without replacement and the individual cosine similarity scores of the APT model were concatenated and compared to the human similarity judgements by calculating Spearman's $\rho$.

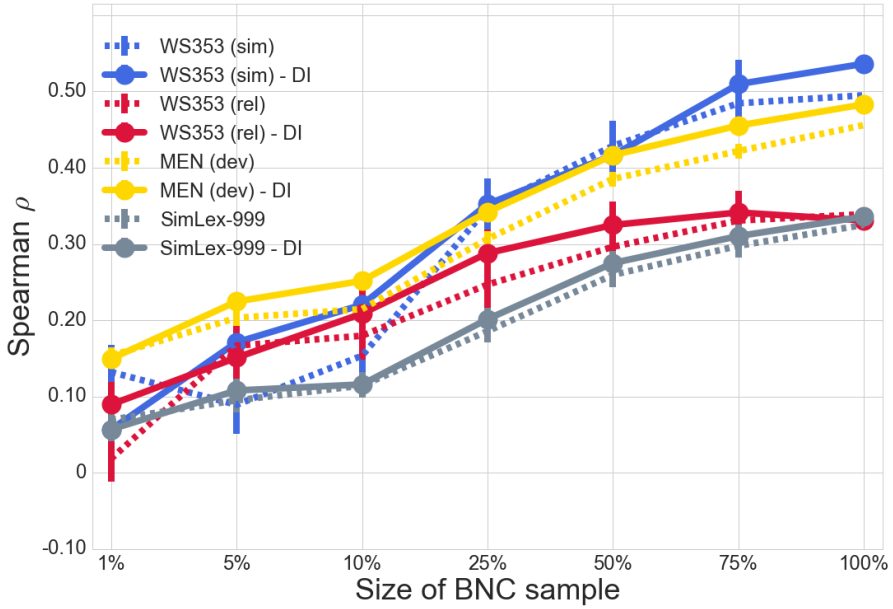


Figure 5.8: Spearman $\rho$ correlation between human similarity judgements and model estimates on WS353 (sim), WS353 (rel), MEN and SimLex-999 with and without distributional inference using the APT baseline model on samples of increasing size of the BNC.

Figure 5.7 illustrates that the use of distributional inference can make up for a substantial amount of data. The positive effect of the

algorithm is amplified for tasks involving distributional composition with an intersective composition function. Figure 5.9 shows the performance gains due to distributional inference on the adjective-noun and verb-object subtasks of the development portion of the ML2010 dataset on the same samples of the BNC. As for the sampling experiments with the word similarity datasets, the APT model's similarity estimates have been concatenated for all 20 samples and compared to the corresponding concatenated human similarity judgements by calculating Spearman's $\rho$.

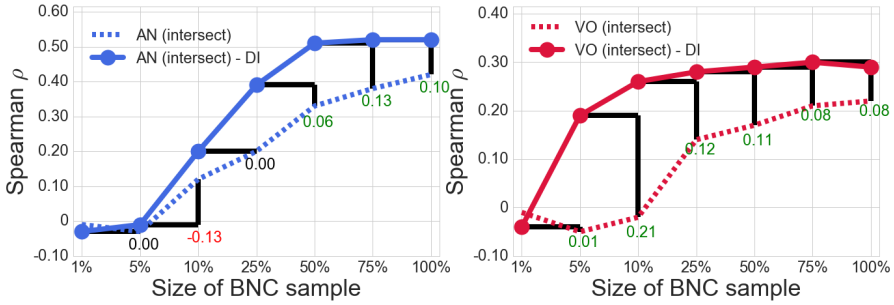


Figure 5.9: Spearman $\rho$ correlation between human similarity judgements and estimates by the APT baseline model on the adjective-noun and verb-object subtasks of the development portion of the ML2010 dataset on increasing sample sizes of the BNC. The dashed line is the performance without distributional inference, the solid line shows the model performance with the use of distributional inference. Composition by intersection has been used as the composition function.

The use of distributional inference results in a substantial performance boost which is proportionally larger for smaller amounts of data. For example by only using 25% of the BNC in conjunction with distributional inference, it is possible to achieve approximately the same performance as if *all* of the BNC had been used without distributional inference on the adjective-noun subtask. The performance improvement is even higher for verb-object compounds, where distributional inference is able to achieve comparable performance with only 5% of the BNC in comparison to using the whole BNC without DI.

Interestingly for both adjective-noun and verb-object compounds, the improvements due to distributional inference are exhausted earlier than the improvements due to adding more data. On the adjective-noun task, the performance of distributional inference remains relatively stable from 50% of the BNC onwards, whereas without the use of DI, the correlation between the human judgements and the model's estimates keeps increasing. For verb-object phrases, peak performance for distributional inference is reached even earlier at about

25% of the BNC.

Given that the BNC is a relatively small corpus, I have merged it with several other corpora in order to test whether the positive effect of distributional inference carries over to much larger amounts of data and, more importantly, whether or when the positive effect of using DI starts levelling off.

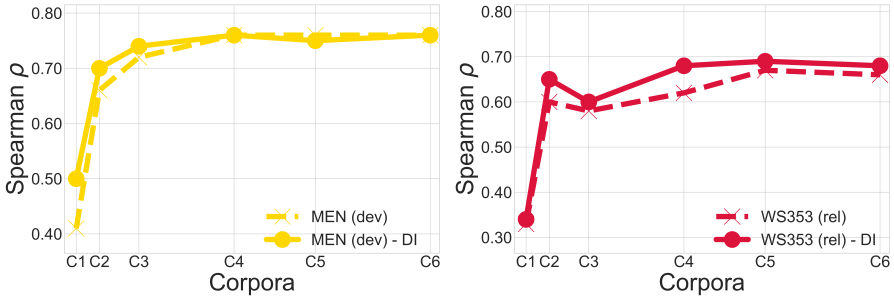


Figure 5.10: Spearman $\rho$ correlation between human similarity judgements and model estimates on MEN (left) and WS353 (rel; right) on corpora of increasing size. The APT model follows the baseline configuration, but uses a negative SPPMI shift of $\log 40$. The dashed line is the performance without distributional inference, the solid line shows the model performance with the use of distributional inference. The spacing on the $x$-axis is proportional to the (cumulative) size of the corpora. Corpora explanation: `C1=BNC, C2=C1+Wikipedia, C3=C2+Toronto, C4=C3+ukWaC, C5=C4+Gigaword, C6=C5+Gutenberg`

In addition to the BNC (`C1=BNC`; ≈0.1bn tokens), I have used a cleaned version of an October 2013 dump of Wikipedia[5] (Wilson, 2015), (`C2=C1+Wikipedia`; ≈0.6bn tokens), the Toronto books corpus (Zhu et al., 2015b), (`C3=C2+Toronto`; ≈1.45bn tokens), consisting of ≈11k books from unpublished authors, ukWaC (Ferraresi et al., 2008), (`C4=C3+ukWaC`; ≈3.5bn tokens), consisting of scraped web pages in the `.co.uk` domain space, the english Gigaword corpus (Parker et al., 2011), (`C5=C4+Gigaword`; ≈5.25bn tokens), consisting of newswire text, and all books electronically available from Project Gutenberg[6], (`C6=C5+Gutenberg`; ≈7.73bn tokens). The largest corpus, `C6`, is ≈77 times larger than the BNC.

Figure 5.10 shows the Spearman $\rho$ performance trajectory on increasing amounts of data, using an order 2 APT space with a negative SPPMI shift of $\log 40$ and constant path weighting. The cleaned Wikipedia corpus is more than 5 times the size of the BNC, and as Fig-

5 Articles with fewer than 20 page views on a particular day have been removed (Wilson, 2015).

6 https://www.gutenberg.org

ure 5.10, when going from C1 to C2, for both datasets shows, adding such a large amount of additional data results in a tremendous surge in performance for the elementary APT representations on MEN (dev) and WS353 (rel). In general, distributional inference is able to improve over the performance of an APT model without DI in the majority of cases, however its contributions become smaller with larger amounts of available data and starts levelling off on MEN and WS353 (rel) from C4 onwards.

The previous chapter has highlighted that an order 2 APT space does not appear to be the optimal choice for the MEN and WordSim-353 datasets[7], hence the performance in Figure 5.10 when using only the BNC (C1) is generally worse than previously published results with count-based distributional semantic models (Kiela et al., 2014). Thus, distributional inference cannot only boost the performance of a well-tuned model as shown in Section 5.2.2, but furthermore remedy some of the shortcomings of a non-optimal parameterisation.

On the MEN (dev) dataset, going from C2 to C3 improves the Spearman $\rho$ correlation for an APT model without distributional inference from 0.66 to 0.72; however the same APT model using only C2 but in conjunction with DI, already achieves a Spearman $\rho$ of 0.70 — with less than half the amount of data. On the WordSim-353 (relatedness) subtask, the APT model with distributional inference, obtained from C2 outperforms the APT model without DI using C4 and achieves comparable performance to the APT model without DI on C5. Hence, distributional inference is able to obtain equal or better performance with only a fraction of the data.

In general, the performance contributions of using distributional inference start diminishing with larger corpora, where beyond C4, no further gains are achieved on the two word similarity tasks[8].

The performance trajectories for the APT baseline model with composition by intersection on the adjective-noun and verb-object subtasks of the development portion of the ML2010 dataset in Figure 5.11 show that distributional inference is able to substantially improve upon a baseline without DI at any level of available data. This provides empirical evidence that distributional inference is a signific-

7 Section 4.2.3 has highlighted that an order 1 APT space with a high SPPMI shift is indeed working substantially better.

8 An interesting side note is that adding general fiction corpora appears to harm the APT models on the WordSim-353 (relatedness) subtask as the two dents (one at C3 when the Toronto books corpus is added, and a smaller one at C6 when the Project Gutenberg corpus is added) in the performance trajectory in Figure 5.10 (right) show.

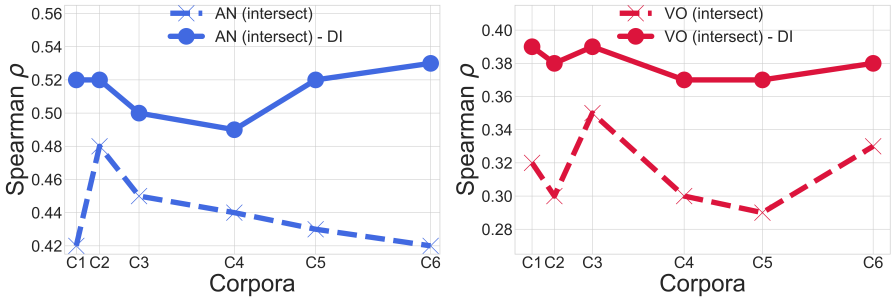


Figure 5.11: Spearman $\rho$ correlation between human similarity judgements and model estimates on ML2010 - Adjective-Noun (left) and ML2010 - Verb-Object (right) on corpora of increasing size. The APT model follows the baseline configuration. The dashed line is the performance without distributional inference, the solid line shows the model performance with the use of distributional inference. The spacing on the $x$-axis is proportional to the (cumulative) size of the corpora. Corpora explanation: `C1=BNC, C2=C1+Wikipedia, C3=C2+Toronto, C4=C3+ukWaC, C5=C4+Gigaword, C6=C5+Gutenberg`

ant ingredient to achieving competitive performance when using an intersective composition function with sparse word representations. As Figure 5.11 shows, the *worst* APT model *with* distributional inference is still better than the *best* model *without* distributional inference for both composition tasks, independent of the amount of available data for an APT model without DI.

Using larger amounts of data for the ML2010 dataset exhibits an interesting corpus-dependent effect, where more data frequently results in *lower* model performance as Figure 5.11 shows. One explanation for this behaviour is that the ML2010 dataset has been constructed on the basis of co-occurrence statistics derived from the BNC (Mitchell and Lapata, 2010). Therefore, all words and bigrams in the ML2010 are ensured to a) occur with sufficient frequency and b) are generally within the same frequency band. This potentially has the effect of factoring out any frequency effects when a model is subsequently learnt from the BNC[9].

Overall, the use of distributional inference almost always has a positive effect on the performance of an APT model and works exceptionally well with little data. It has its largest positive contribution in composition tasks, in conjunction with an intersective composition function — a characteristic that will be analysed in more depth in

9 Frequency effects in word representations are known to have a substantial impact on performance, for example for lexical entailment (Weeds and Weir, 2003) or sequence labelling tasks (Schnabel et al., 2015).

Section 5.4. The positive effect of using DI tends to become smaller with larger amounts of data for the word similarity tasks, but remains a solid method for getting more value out of the available data. Furthermore, if peak performance for an APT model with distributional inference is approximately the same as for an APT model without DI, using distributional inference reaches that peak earlier and frequently with a significantly smaller amount of data. This would make the algorithm particularly well suited for improving distributional semantic models in low-resource settings.

### 5.2.5 *Relation of Distributional Inference to Previous Work*

Leveraging the use of information from the distributional neighbourhood for composing two or more word representations has already been attempted in previous work, although none of that work has investigated the kind of information that is inferred in detail. Furthermore, previous work has primarily used the inference mechanism to augment distributional composition rather than to improve elementary representations.

Perhaps the earliest proposal dates back to the *predication* algorithm of Kintsch (2001), who used an additive composition function together with an additive inference mechanism to enrich the representation of a composed phrase. Kintsch (2001) focused on intransitive verb phrases and his proposal was to choose the top $n$ neighbours of the predicate verb and from this set, select the top $k$ also most similar to the argument noun in a phrase.

A similar proposal has been brought forward by Utsumi (2009) who introduced the *comparison* algorithm which adds the information of the top $n$ *common* distributional neighbours of both constituents to the resulting composed phrasal representation.

Both algorithms incorporate the constraint that the neighbours added to the resulting composed representation must be compatible with all lexemes in a given phrase. This constraint can be integrated into the DI algorithm, thereby generalising the distributional inference algorithm and subsuming the proposals of Kintsch (2001) and Utsumi (2009). The generalisation can be achieved by integrating an additional conditional statement, representing a neighbour selection step, that governs whether a given neighbour $n$ of the current lexeme $w$ will be added to the final representation depending on its member-

ship in a given set of neighbours $N$. Algorithm 2 below shows the modified algorithm with the additional constraint.

**Algorithm 2** Generalised Distributional Inference

```
1: procedure DISTRIBUTIONAL_INFERENCE(M, w, k, N)
2:     w' ← w × k
3:     for all n in neighbours(M, w, k) do
4:         if N = ∅ or n ∈ N then        ▷ Require that n is a member of N
5:             w' ← merge(w', n)
6:         end if
7:     end for
8:     return w'
9: end procedure
```

The original distributional inference algorithm can be retrieved by passing an empty set, $N = \emptyset$, to the algorithm. Interestingly the algorithms by Kintsch (2001) and Utsumi (2009) can be represented by the same pseudo-code, the only difference being the size of the set $N$, which for the *comparison* algorithm by Utsumi (2009) needs to be larger in order to account for the fact that the two lexemes being composed are very dissimilar and so the number of neighbours required from one lexeme must be sufficiently large to satisfy the overall constraint of using the top $n$ common neighbours for inference.

The additional constraint in the algorithm has no impact on its runtime as set membership checks can be very efficiently implemented and executed in constant time.

The conditional spanning lines 4-6 in Algorithm 2 can be incorporated into the neighbour retrieval method, extending it to a *neighbour selection* routine, which is highlighted in the pseudo-code snippet below that shows the modified main loop of the algorithm.

```
1: for all n in neighbour_selection(M, w, k, N) do
2:     w' ← merge(w', n)
3: end for
```

The change of name from `neighbours(...)` to `neighbour_selection(...)` highlights that the purpose of the function has been extended from pure retrieval to retrieval and selection.

*Distributional Inference as Data Augmentation*

Distributional inference can also be interpreted as a form of *data augmentation* in the distributional space. Data augmentation has been

a commonly used technique in computer vision for creating additional data by rotating or randomly cropping the original images (Krizhevsky et al., 2012; Chatfield et al., 2014). Recently, this approach has been adopted for natural language processing tasks such as machine translation (Sennrich et al., 2016; Fadaee et al., 2017) or morphological inflection generation (Bergmanis et al., 2017; Silfverberg et al., 2017).

Instead of augmenting the raw input data, which would correspond to adding additional sentences to the BNC by modifying existing ones, it is the representations themselves that are augmented with additional knowledge from similar instances. For example, for data augmentation, all sentences in the BNC with an occurrence of *bike* would be copied and all occurrences of *bike* in the copied sentences would be replaced by *bicycle*, in order to create more training data for the lexeme *bicycle*. The resulting representation for *bicycle* would essentially correspond to the case where unobserved knowledge from the representation for *bike* has been added to *bicycle* with distributional inference. Thus, distributional inference represents a more direct approach of enriching representations without the need for generating new input data.

### 5.2.6 *Inferring Noise - The Limitations of Distributional Inference*

As previously pointed out in Section 5.2.1, which concerned the knowledge that can be learnt from the distributional neighbourhood, inferring information from the nearest neighbours has the risk of inferring implausible co-occurrences — for example that *cats* might *bark*. Furthermore, the more neighbours are used to infer information from, the less reliable the information becomes. For example, one might infer that *cats* can be *four-wheeled* or that there are *military cats*, because the lexeme *vehicle* happened to be among the top $n$ neighbours for *cat*. There is no immediately obvious way to prevent this from happening while retaining the unsupervised nature of the algorithm[10]. In order to estimate the impact of noise in the inference process, I compare the distributional inference algorithm with the standard "top $n$ neighbour" retrieval function to a version where neighbours are retrieved

10 One possible solution would be to use an ensemble of several distributional models and only use a neighbour for inference if it occurs in the nearest neighbour list in the majority of models. However, this approach is out-of-scope for this thesis and represents an idea for future work.

on the basis of their occurrence as synonyms, hypernyms, hyponyms, meronyms or direct co-hyponyms in WordNet[11].

| | **WS353 (sim)** | **WS353 (rel)** | **MEN** | **SimLex-999** |
|---|---|---|---|---|
| Best top $n$ | 0.54 (+/- 0.06) | **0.35** (+/- 0.06) | 0.48 (+/- 0.02) | 0.30 (+/- 0.01) |
| WordNet | **0.57** (+/- 0.14) | 0.31 (+/- 0.00) | **0.52**$^{\dagger}$ (+/- 0.00) | **0.50**$^{\ddagger}$ (+/- 0.01) |

Table 5.5: Comparison between the best unsupervised DI variant using the "static top $n$" neighbour retrieval method, and a WordNet-based neighbour retrieval function. The APT baseline model has been used with both DI variants. Performance is reported in terms of averaged Spearman $\rho$ across 2-fold cross-validation. The numbers in parentheses denote the standard deviation across the two runs. Tasks favouring taxonomic similarity benefit more from using a hand crafted resource such as WordNet. Results marked with † are statistically significant at the $p < 0.05$ level, and results marked with ‡ are statistically significant at the $p < 0.01$ level according to the method of Steiger (1980).

Table 5.5 shows that for similarity tasks such as the WordSim-353 similarity subset and SimLex-999, that benefit proportionally more from inferring information from hypernyms, hyponyms and co-hyponyms, the performance improvements by using a "clean" source such as WordNet can be substantial. Especially the surge by 20 points for the SimLex-999 dataset is remarkable[12].

Interestingly, using WordNet as neighbour retrieval function does not result in the same performance improvements for tasks that focus on the topical relatedness of two lexemes rather than their taxonomic similarity. For example the performance improvement for MEN is only marginal in comparison to the purely unsupervised variant of the DI algorithm. For the WordSim-353 (relatedness) task, using WordNet even underperfoms the standard DI algorithm[13].

Table 5.5 furthermore highlights that inferring unobserved co-occurrence events is a fine-grained task specific problem and depends on the specific kind of similarity (taxonomic, topical, etc.) that a given task is testing. As the cases of MEN and WordSim-353 (relatedness) show, the utility of inferring purely taxonomic knowledge from WordNet into a task geared towards testing topical relatedness is limited.

11 The WordNet retrieval function has already been tested in Kober et al. (2016), where however, only synonyms have been used for inference. The current comparison represents an extension of the version of Kober et al. (2016).

12 One explanation for this huge surge is that the dataset construction of SimLex-999 made use of WordNet to extract similar word pairs.

13 A further disadvantage in using WordNet might be in low-resource settings or in very specialised domains where a data-driven approach would be expected to work better.

The results on the word similarity tasks show that if a handcrafted resource is available, and perhaps more importantly, suitable for a given task, its performance is able to exceed the standard unsupervised distributional inference algorithm. However, the results show that the inference of potentially noisy co-occurrences is not a dominant problem and that just leveraging the distributional neighbourhood results in robust knowledge inferences that improve the performance of a distributional model significantly.

*The Effect of Too Many Neighbours*

While the use of distributional inference with a "good" number of neighbours can substantially improve the performance of a distributional model, using too many neighbours for the inference process can have a severe negative effect. In order to analyse that effect, I am using the BLESS dataset and compare an APT baseline model without DI to the same model with distributional inference, using 1000 neighbours for the inference process. Figure 5.12 shows the distribution of semantic relations among the target concepts from the BLESS dataset between the two APT spaces.

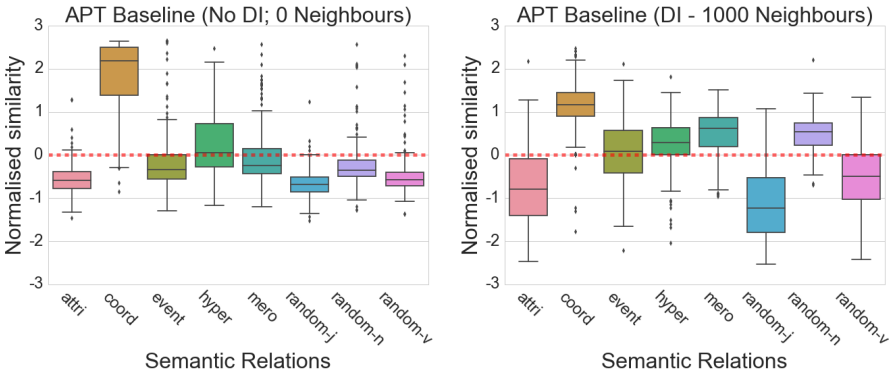


Figure 5.12: Comparison of the distribution of semantic relations of the APT baseline model without distributional inference (left) and the same APT model with distributional inference, using 1000 neighbours (right). The large number of neighbours substantially decreases the specificity of the distributional space, for example by causing random nouns to be more similar to the BLESS target concepts than their hypernyms.

While co-hyponyms remain the dominant semantic relation in the distributional space with DI, the large number of neighbours caused an increase in similarity of the target concepts to any other relation as well. This has the consequence that for example the median similarity between random nouns and BLESS concepts is higher than

the median similarity between BLESS concepts and their hypernyms. The relative differences between target concepts and lexemes of any semantic similarity have become substantially smaller when distributional inference with (too) many neighbours is used. This is an effect of "oversmoothing" the semantic space and is the result of using a large proportion of the same neighbours for the inference process, thereby rendering all representations more similar to each other.

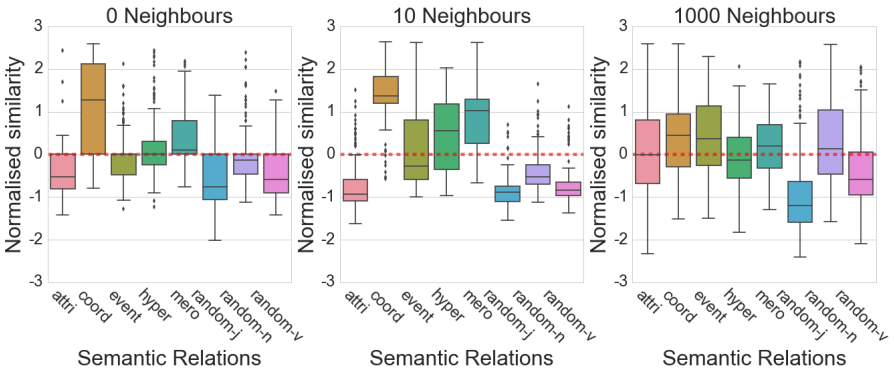


Figure 5.13: Comparison of the distribution of semantic relations of the APT baseline model without distributional inference (left) and the same APT model with distributional inference, using 10 neighbours (middle) and 1000 neighbours (right), respectively. All APT models use composition by intersection as composition function. While a small number of neighbours increases the specificity of the space, too many neighbours leads to oversmoothing resulting in a distributional space where random relata are difficult to distinguish from co-hyponyms, hypernyms or meronyms for any given target concept.

Oversmoothing can also have a negative impact on the distributional space after composition as Figure 5.13 shows. The figure compares the distribution of semantic relations of the APT baseline model without distributional inference (left), to the same APT model with distributional inference using 10 neighbours (middle), and 1000 neighbours (right) on the same subset of BLESS target concepts, composed with their most frequent adjectival modifiers as in Section 4.3.3 when characterising the distributional semantics of composed APT representations. For all APT models, composition by intersection is used.

Using distributional inference with 10 neighbours has a clear positive effect on the nature of the distributional space, for example by making the distribution of similarities of co-hyponyms peakier, while on average, substantially decreasing the similarity to attributes and random relata. However, with 1000 neighbours, the distributional space is devoid of almost all of its specificity, resulting in a space where random nouns frequently cannot be distinguished from co-

hyponyms, meronyms or hypernyms based on their feature distribution alone.

## 5.3 OFFSET INFERENCE

The previous section has shown that the proposed distributional inference algorithm can successfully infer missing knowledge into distributional word representations and thereby — to a substantial extent — overcome the issue of data sparsity in APTs. However, the existing algorithm performs inferences on the basis of the "surface form" of a word. For example, it is possible to learn things about a *bicycle*, such that it can be *stolen*[14], by observing its neighbour *bike*, but it is not possible to infer knowledge from other things that can be *stolen*.

This section describes the generalisation of the distributional inference algorithm to offset inference, followed by a characterisation of the kind of knowledge that can be learnt from the distributional space (§ 5.3.1). Section 5.3.2 presents the experimental results, showing that APTs, together with offset inference achieve a new state-of-the-art on the short phrase composition dataset of Mitchell and Lapata (2010).

Inferring knowledge from higher-order interactions between lexemes can be achieved by leveraging the rich type structure in APTs that give rise to offset representations. Offset APTs describe the semantics of a concept on a more abstract level. Figure 5.14 illustrates the capacity of an APT structure to represent different concepts on the basis of shifting its anchor position. For example, considering the adjective *old*, and shifting its anchor position along the `amod` edge[15], creates a noun view that describes the semantics of "something that can be *old*", such as a *bicycle* or a *desk*, as the top left illustration in Figure 5.14 shows.

Similarly, by considering the verb *steal*, offsetting its anchor along the $\overline{\texttt{dobj}}$ edge results in a structure of "something that can be *stolen*", such as a *bike* or a *wallet* as highlighted by the top right example in Figure 5.14. APTs also support higher-order offsets as the bottom example in Figure 5.14 shows. The anchor of the adjective *old* is first

14 Indeed, the distributional feature $\overline{\texttt{dobj}}$:*steal* has not been observed with the lexeme *bicycle* in the representations derived from the BNC, but it has been observed with *bike*.

15 Recalling from Chapter 3 that traversing an edge is a process of type reduction of the form $\downarrow(\overline{\texttt{r}}.\texttt{r}) = \epsilon$, where r is some dependency relation, such as in the concrete reduction $\downarrow(\overline{\texttt{amod}}.\texttt{amod}) = \epsilon$. Thus, traversing an edge requires an offset by its inverse.

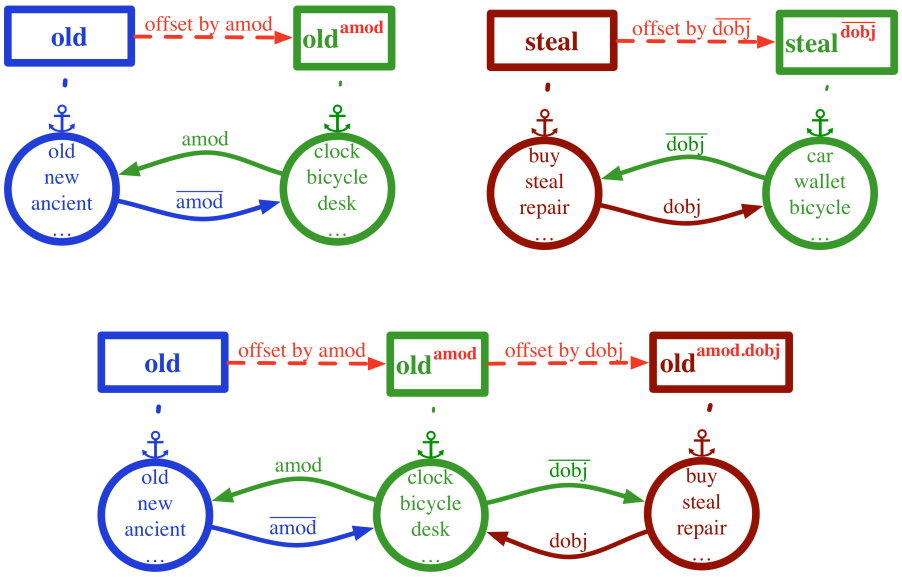


Figure 5.14: Illustration of 3 example offsets. The top left example creates a noun offset view of the adjective *old*, the top right example creates a noun offset view of the verb *repair*, and the bottom example shows a higher-order offset, creating a verb view for the adjective *old*.

shifted along the `amod` edge, creating a "thing that can be *old* structure" as in the top right example. Subsequently, its anchor is further shifted along the `dobj` edge, creating a "something that can be done to an *old* thing", which represents a verb.

This creates the potential for a much richer inference mechanism, that is able to infer features from the "surface form" distributional APT representation of a lexeme, as well as from its higher-order interactions. For example, it is possible to learn that a *bicycle* can be stolen from its distributional neighbour *bike*, and subsequently infer further knowledge from "other things that can be *stolen*", such as that they might be *expensive* or *valuable*.

Offset inference therefore offers the possibility to learn new knowledge at a more abstract *concept* level and goes beyond the shallow inferences that can be made from the co-occurrences that have been observed at the lexeme level. In order to achieve higher-order inferences, I extend the distributional inference algorithm to include an additional step, offsetting the current lexeme $w$ by a specified path $p$, to infer knowledge from neighbours of offset representations of the given lexeme.

The pseudo-code for offset inference is shown in Algorithm 3 below. The input to the algorithm is a source distributional model to query neighbours from, $M$, a distributional representation of the lex-

eme $w$ for which offset inference should be performed, an offset path $p$ indicating on which offset view of $w$ offset inference should be performed on, the number of neighbours used for offset inference $k$, and an optional set of neighbours constraining the neighbour space for inference $N$.

**Algorithm 3** Offset Inference

```
1: procedure OFFSET_INFERENCE(M, w, p, k, N)
2:     w' ← offset(w, p)
3:     for all n in neighbour_selection(M, w', k, N) do
4:         w'' ← merge(w'', n)
5:     end for
6:     return w''
7: end procedure
```

The original distributional inference algorithm can be recovered by passing the empty path, $p = \epsilon$, as offset to the algorithm. A further constraint to Algorithm 3 is the need of the source distributional model $M$ to provide a meaningful mechanism for representing offset representations.

The additional offset step in the algorithm does impact its runtime. The runtime of offsetting depends on the dimensionality of the distributional space — or with a tighter bound — on the average number of non-zero dimensions across all word representations in the space, $d_{\text{Avg}}$. The algorithm performs inference for a single given offset at a time, however it is possible to infer additional distributional features for any number of offset paths. For example, for any noun, one could perform distributional inference on the noun itself, as well as on its `amod`, $\overline{\texttt{dobj}}$ and $\overline{\texttt{nsubj}}$ offset views, executing the algorithm 4 times altogether. A single run of the algorithm now consumes $\mathcal{O}(d_{\text{Avg}} + n\,q\,d_{\text{Avg}})$ runtime due to the offset operation requiring $\mathcal{O}(d_{\text{Avg}})$ additional work. Denoting the number of offset paths for a given lexeme as $z$, the running time of the algorithm increases to $\mathcal{O}(z\,(d_{\text{Avg}} + n\,q\,d_{\text{Avg}}))$[16].

### 5.3.1 *What kind of Knowledge can be Inferred?*

Table 5.6 shows a number of example inferences that can be made from offset representations. All example co-occurrences in the table have been observed with the neighbours, but not with the target lexeme itself. Offset representations exhibit a less specific semantic space,

16 Presupposing that the pairwise similarities have been computed upfront.

as shown for its distribution of semantic relations in the previous chapter, because they conflate several different semantic meaning potentials into one view. For example, the offset representation $old^{\texttt{amod}}$ — an *old thing* — blends together fragments of meaning from its usage in *old car*, *old friend*, *old job*, amongst many others.

| Offset | Neighbours | Inferred Co-occurrences |
|---|---|---|
| magazine$^{\overline{\texttt{amod}}}$ | newspaper$^{\overline{\texttt{amod}}}$, glossy, **monthly** | $\overline{\texttt{amod}}$:*column*, $\overline{\texttt{amod}}$:*report* $\overline{\texttt{amod}}$:*payment* |
| magazine$^{\texttt{dobj}}$ | newspaper$^{\texttt{dobj}}$, edit, **read** | dobj:*paragraph*, dobj:*fiction*, dobj:*message* |
| magazine$^{\texttt{nsubj}}$ | newspaper$^{\texttt{nsubj}}$, journal$^{\texttt{nsubj}}$, **publish** | nsubj:*press*, nsubj:*researcher*, nsubj:*government* |
| cafe$^{\overline{\texttt{amod}}}$ | pub$^{\overline{\texttt{amod}}}$, restaurant$^{\overline{\texttt{amod}}}$, **comfortable** | $\overline{\texttt{amod}}$:*lounge*, $\overline{\texttt{amod}}$:*house* $\overline{\texttt{amod}}$:*accommodation* |
| cafe$^{\texttt{dobj}}$ | pub$^{\texttt{dobj}}$, restaurant$^{\texttt{dobj}}$, **room**$^{\texttt{dobj}}$ | dobj:*hotel*, dobj.amod:*furnished* dobj.amod:*elegant* |
| cafe$^{\texttt{nsubj}}$ | pub$^{\texttt{nsubj}}$, restaurant$^{\texttt{nsubj}}$, **room**$^{\texttt{nsubj}}$ | nsubj:*library*, nsubj:*office* nsubj.amod:*spacious* |
| cat$^{\overline{\texttt{amod}}}$ | dog$^{\overline{\texttt{amod}}}$, stray, **wild** | $\overline{\texttt{amod}}$:*party*, $\overline{\texttt{amod}}$:*flower* $\overline{\texttt{amod}}$:*creature* |
| cat$^{\texttt{dobj}}$ | dog$^{\texttt{dobj}}$, cat$^{\texttt{nsubj}}$, **feed** | dobj:*cattle*, dobj:*population*, dobj.amod:*hungry* |
| cat$^{\texttt{nsubj}}$ | dog$^{\texttt{nsubj}}$, cat$^{\texttt{dobj}}$, **jump** | nsubj:*heart*, dobj:*fence* dobj:*cliff* |

Table 5.6: Example offset inferences from the boldfaced neighbours for a given lexeme using the APT baseline model. The distributional features have been observed with the (boldfaced) neighbours, but not with the offset representation itself.

Table 5.6 reflects that characteristic. For example, a close neighbour of the adjective offset view for the noun *magazine* is the adjective *monthly* which indicates that magazines tend to occur in a monthly interval. However, *something monthly* is less specific than an attribute of a *newspaper*, i.e. it could also be a *bill* or a *salary*, hence the offset inference algorithm infers co-occurrences such as $\overline{\texttt{amod}}$:*report* or $\overline{\texttt{amod}}$:*payment* for the lexeme *magazine*.

However, performing inferences on offset representations opens the possibility to expand the knowledge of a lexeme beyond the features that can be learnt from a direct distributional neighbour. For example, Table 5.6 shows that from observing the neighbours of $cafe^{\texttt{dobj}}$ — verbs that are specifying actions that take *cafe* as their direct object — it is possible to infer that cafes are usually *furnished* and often *elegant* from its neighbour $room^{\texttt{dobj}}$. Another example is that through the offset representation $cat^{\texttt{dobj}}$ — things done to a *cat* — it can be learnt

that cats can be *hungry* from its neighbour *feed*. Thus, offset inference is able to learn attributive and behavioural *common sense knowledge* about a given lexeme that distributional semantic models frequently struggle with (Rubinstein et al., 2015) as the relevant information is often not mentioned explicitly in the text.

*Characterising the Effect of Offset Inference on the Distributional Space*

Figure 5.15 shows the impact of offset inference on the BLESS dataset, using the APT-WS-MEN model (top row in Figure 5.15) and the APT baseline model (bottom row in Figure 5.15). The top row shows how the distribution of similarities is changed from an APT-WS-MEN space without distributional distributional inference (top left) to the same APT-WS-MEN model with offset inference (top right) on the $\overline{\texttt{amod}}$, `dobj` and `nsubj` views for each target concept.

The use of offset inference has the effect of making the similarities of co-hyponyms much peakier, and comparatively more similar to the target concepts, than the APT space without distributional inference. At the same time, offset inference *decreases* the similarities to all other relations for this APT space. This result is remarkable, as it provides empirical evidence that in its combination, co-hyponyms share a substantial amount of properties as well as events in which they participate as agent or patient. The combination of these properties and events, however, is not shared by hypernyms or meronyms.

The bottom row in Figure 5.15 highlights the change in similarities between an APT baseline model without DI (bottom left) and the same APT baseline model with combined standard distributional inference and offset inference on the `advmod` view of all target concepts (bottom right). Enriching the elementary representations with the combination of standard and offset distributional inference has the effect of *decreasing* the similarities to co-hyponyms, while at the same time substantially *increasing* the similarities to hypernyms, meronyms, and — somewhat undesirably — events and random nouns. Offsetting by the `advmod` relation shows the versatility of offset inference, especially in comparison to offsetting an APT representation by $\overline{\texttt{amod}}$, `dobj` and `nsubj`. Where the offset inference on the `advmod` view biases the semantics of the representation away from co-hyponyms and towards hypernyms and meronyms, offset inference on $\overline{\texttt{amod}}$, `dobj` and `nsubj` offset views biases the semantics of the APTs towards co-hyponyms exclusively.

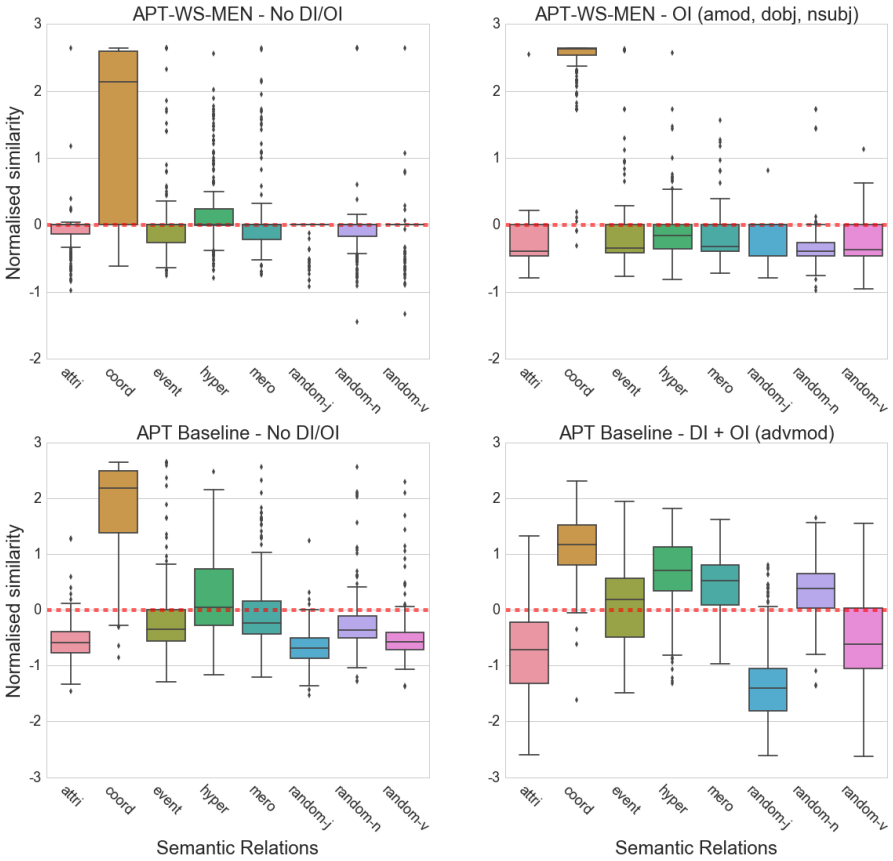


Figure 5.15: Comparison of the APT-WS-MEN models (top row) and APT baseline models (bottom row) on the BLESS dataset with and without offset inference. The top row compares the APT-WS-MEN model without any form of distributional inference to the same APT-WS-MEN model with offset inference on its $\overline{\texttt{amod}}$, `dobj` and `nsubj` offset views. The bottom row compares the APT baseline model without DI to the same APT baseline model with combined standard distributional inference and offset inference on the `advmod` offset views of each target concept.

This behaviour creates an interesting avenue for future work where offset inference in conjunction with Anchored Packed Trees can potentially be leveraged for distinguishing co-hyponyms and hypernyms in text. For example, the boxplot for offset inference on the $\overline{\texttt{amod}}$, `dobj` and `nsubj` offset views (Figure 5.15, top right) shows that the combination of properties of target concepts, actions done to a target concept, and actions carried out by a target concept, tend to be very similar to their respective co-hyponyms, but far less similar to their hypernyms.

### 5.3.2 *Quantitative Analysis*

In order to evaluate the performance of offset inference, I am using the same datasets — WS353 (sim), WS353 (rel), MEN, SimLex-999, and ML2010 — as for the evaluation of the standard distributional inference algorithm, and reusing the same optimised APT models. For the two WordSim-353 subtasks and the MEN dataset, I use the APT-WS-MEN model, for SimLex-999 I use the APT-SL-999 model, and for the ML2010 dataset I use the APT baseline model for composition by intersection and the APT union model for composition by union. In the following, given the ambiguity of the term *distributional inference*, I will use "standard distributional inference" to refer to generalised distributional inference (see Algorithm 2) and "offset inference" to refer to the algorithm introduced in this section (see Algorithm 3).

#### *Word Similarity*

For the word similarity tasks, I restrict the evaluation to nouns only and infer knowledge from neighbours from the respective $\overline{\texttt{amod}}$, `dobj`, `nsubj` offset representations of a given lexeme. I furthermore test the combination of offset inference and standard distributional inference, which infers co-occurrences from "surface-form" distributional neighbours in addition to its offset neighbours. Essentially, combining offset inference and standard distributional inference amounts to executing the offset inference algorithm an additional time with $p = \epsilon$ as the offset path (see Algorithm 3 above).

Table 5.7 shows an overall comparison between the APT baseline model without any form of distributional inference, the respective tuned APT models from the previous chapter without distributional inference, and the tuned APT models with standard distributional inference, offset inference, and the combination of standard distributional inference and offset inference. As the results show, using offset inference by itself results in worse performance than using standard distributional inference and generally performs barely better than the tuned APT model without DI. Combining standard distributional inference and offset inference appears to benefit the WordSim-353 (similarity) subtask, but is hurting performance on the WordSim-353 (relatedness) and MEN datasets.

This suggests that higher-order knowledge inferences from offset neighbours are less beneficial for word similarity tasks. A likely explanation is that the offset inference process introduces too much

| | WS353 (sim) | WS353 (rel) | MEN (N) | SL-999 (N) |
|---|---|---|---|---|
| APT baseline | 0.40 (+/- 0.14) | 0.24 (+/- 0.06) | 0.399 (+/- 0.01) | 0.24 (+/- 0.01) |
| Tuned APTs | 0.52‡ (+/- 0.09) | **0.35**† (+/- 0.01) | 0.440‡ (+/- 0.01) | 0.25 (+/- 0.01) |
| Tuned APTs + DI | 0.54‡ (+/- 0.06) | **0.35**† (+/- 0.06) | **0.492**‡♠ (+/- 0.01) | **0.32**‡♠ (+/- 0.00) |
| Tuned APTs + OI | 0.53‡ (+/- 0.08) | 0.29 (+/- 0.06) | 0.435† (+/- 0.01) | 0.28† (+/- 0.01) |
| Tuned APTs + comb. | **0.59**‡♢ (+/- 0.02) | 0.26 (+/- 0.08) | 0.440‡ (+/- 0.01) | 0.27 (+/- 0.01) |

Table 5.7: Comparison between distributional inference (DI), offset inference (OI), combined offset and standard distributional inference (combined), with the tuned APT models without any distributional inference and the APT baseline model without distributional inference on the two WordSim-353 subtasks as well as the noun-only subsets of MEN and SimLex-999. Performance is reported in terms of averaged Spearman $\rho$ across 2-fold cross-validation. The numbers in parentheses denote the standard deviation across the two runs. Results marked with † and ‡ are statistically significant at the $p < 0.05$ and $p < 0.01$ levels, respectively, in comparison to the APT baseline model. Results marked with ♢ and ♠ are statistically significant at the $p < 0.05$ and $p < 0.01$ levels, respectively, in comparison to the Tuned APT model without any form of distributional inference. Statistical significance has been calculated according to the method of Steiger (1980). For readability reasons, a third significant digit is only added when necessary.

noise into the elementary APT representations, which are lacking a mechanism, such as composition by intersection, to filter out the implausible co-occurrences.

Table 5.8 lists the optimal number of neighbours for the word similarity tasks, determined by 2-fold cross-validation for standard distributional inference, offset inference, and combined distributional and offset inference.

| Dataset | DI | OI | DI+OI |
|---|---|---|---|
| WordSim-353 (similarity) | 200 | 10 | 10/50 |
| WordSim-353 (relatedness) | 500 | 100 | 25/100 |
| MEN | 50 | 10 | 10/50 |
| SimLex-999 | 500 | 10 | 10/50 |

Table 5.8: Number of neighbours used for the word similarity tasks. The notation $x/y$ for combined distributional and offset inference means means that $x$ is the number of neighbours used for standard distributional inference and $y$ is the number of offset neighbours for offset inference.

*Phrase Similarity*

The ML2010 dataset allows for a more focused offset inference process because composition requires the explicit instantiation of a spe-

cific offset view in order to align the APT representations in a given phrase. For any two constituents in a phrase, the head of the phrase is enriched with standard distributional inference. For the dependent, its appropriate offset view is instantiated, and subsequently enriched with offset inference. For the combination of the two inference methods, the head of the phrase is smoothed with standard distributional inference as before. The dependent is smoothed with standard distributional inference prior to offsetting, and after its offset view has been instantiated from its enriched representation, offset inference is performed on the instantiated offset view before composition.

A comparison on the ML2010 dataset between an APT model, using composition by intersection and composition by union, without distributional inference, the same model with distributional inference, offset inference, and its combination is shown in Table 5.9. As the table highlights, any form of distributional inference significantly improves over a baseline without a mechanism to enrich the elementary representations for composition by intersection. For composition by union, standard distributional inference does not improve performance across all three composition tasks on average, however offset inference or combined standard distributional inference and offset inference does.

The benefit of using offset inference with composition by union is smaller in magnitude in comparison to a baseline without DI, but consistent and furthermore consistently better than the standard distributional inference algorithm. However, due to lacking a mechanism for filtering implausible or noisy co-occurrence events, the performance cannot be improved much beyond the level of a well tuned APT space without any distributional inference.

With the APT baseline model, using composition by intersection, offset inference is able to improve upon a baseline without any distributional inference, however its performance lags behind that of standard distributional inference. The reason for this is that the APT baseline model represents a non-optimal parameterisation for offset inference. If the number of neighbours for all distributional inference algorithms is optimised jointly with the other APT hyperparameters, offset inference, together with composition by intersection, achieves state-of-the-art performance on the ML2010 dataset. For standard distributional inference, the APT baseline model indeed represents the optimal parameterisation, whereas for offset inference, the APT union model is a significantly better configuration. Jointly optim-

| **Composition by Intersection** | **AN** | **NN** | **VO** | **Average** |
|---|---|---|---|---|
| Tuned APT model | 0.39 | 0.41 | 0.35 | 0.38 |
| Tuned APT model + DI | **0.48**$^{\ddagger}$ | **0.46**$^{\ddagger}$ | **0.44**$^{\ddagger}$ | **0.46**$^{\ddagger}$ |
| Tuned APT model + OI | 0.46$^{\ddagger}$ | 0.43$^{\ddagger}$ | 0.41$^{\ddagger}$ | 0.43$^{\ddagger}$ |
| Tuned APT model + combined | 0.47$^{\ddagger}$ | 0.43$^{\ddagger}$ | 0.41$^{\ddagger}$ | 0.44$^{\ddagger}$ |
| **Composition by Union** | **AN** | **NN** | **VO** | **Average** |
| Tuned APT model | 0.503 | **0.454** | 0.445 | 0.467 |
| Tuned APT model + DI | 0.499 | 0.444 | 0.452$^{\dagger}$ | 0.465 |
| Tuned APT model + OI | 0.503 | 0.445 | **0.459**$^{\ddagger}$ | **0.469**$^{\dagger\spadesuit}$ |
| Tuned APT model + combined | **0.504**$^{\diamondsuit}$ | 0.445 | **0.459**$^{\ddagger}$ | **0.469**$^{\dagger\spadesuit}$ |

Table 5.9: Comparison between tuned APT models without DI and the same models with the use of distributional inference on the ML2010 composition task. Results denoted with † and ‡ mark statistical significance at the $p < 0.05$ and $p < 0.01$ level in comparison to the Tuned APT model without distributional inference, per composition function, respectively. Results marked with ♢ and ♠ mark statistical significance at the $p < 0.05$ and $p < 0.01$ level in comparison to the Tuned APT models with standard distributional inference, per composition function, respectively. Statistical significance has been determined using the method of Steiger (1980). For readability reasons, a third significant digit is only added when necessary — the use of 3 significant digits is justified when using the original evaluation regime of Mitchell and Lapata (2010), which treats every human judgement as an individual data point, resulting in almost 3.9k data points in the test set.

ising the number of neighbours with the other APT parameters did not result in further improvements for composition by union.

Table 5.10 compares the optimised APT spaces without distributional inference, standard distributional inference, offset inference, and combined standard and offset distributional inference, using composition by intersection, to other state-of-the-art model that reported results on the ML2010 dataset using the BNC as source corpus[17].

17 For example, other approaches that reported state-of-the-art performance on the ML2010 dataset such as Wieting et al. (2015) used the Paraphrase Database (Ganitkevitch et al., 2013) as source corpus, while we used a cleaned version of Wikipedia in Kober et al. (2016), and the concatenation of ukWaC, Wackypedia and the BNC in Kober et al. (2017a).

18 The results reported here differ from the ones reported in Table 3 in Hashimoto et al. (2014) because they did not evaluate their model against just the human judgements from the test set. Unfortunately, private communication with Kazuma Hashimoto could not resolve the issue. The reported results in this thesis use the published resources of Hashimoto et al. (2014) and re-evaluted their model on just the test set of the ML2010 dataset.

| Model | AN | NN | VO | Average |
|---|---|---|---|---|
| Kiela et al. (2014)* (cosine) | 0.57 | 0.56 | 0.52 | 0.55 |
| Kiela et al. (2014)* (correlation) | 0.66 | 0.60 | 0.53 | 0.60 |
| APT* + combined (*intersect* & cosine) | **0.76** | **0.64** | **0.64** | **0.68** |
| APT* + combined (*intersect* & correlation) | 0.73 | **0.64** | 0.58 | 0.65 |
| Mitchell and Lapata (2010) | 0.46 | 0.49 | 0.37 | 0.44 |
| Blacoe and Lapata (2012) | 0.48 | 0.50 | 0.35 | 0.44 |
| Hashimoto et al. (2014)[18] | 0.49 | 0.45 | **0.46** | 0.47 |
| Weir et al. (2016) | 0.45 | 0.42 | 0.42 | 0.43 |
| APT + DI (*union*) | 0.50 | 0.44 | 0.45 | 0.46 |
| APT + OI (*union*) | 0.50 | 0.45 | **0.46** | 0.47 |
| APT + combined (*union*) | 0.50 | 0.45 | **0.46** | 0.47 |
| APT + DI (*intersect*) | 0.48 | 0.46 | 0.44 | 0.46 |
| APT + OI (*intersect*) | 0.50 | 0.50‡♣ | 0.45 | 0.48 |
| APT + combined (*intersect*) | **0.52**‡ | **0.51**‡♣ | 0.45 | **0.49**‡♠ |
| Human agreement | 0.52 | 0.49 | 0.55 | 0.52 |

Table 5.10: Comparison between the best performing APT model in this thesis with the state-of-the-art results from the literature. Models marked with * denote that the result has been obtained by comparing to *averaged* human judgements (i.e. the same evaluation regime as used for the word similarity tasks) whereas models without an asterisk are evaluated on the basis of comparing to *individual* human judgements (i.e. the original scheme of Mitchell and Lapata (2008)). Results marked with ‡ are statistically significant at the $p < 0.01$ level in comparison to the APT + DI (*intersect*) model, and ♠ denotes statistical significance at the $p < 0.01$ level in comparison to the APT + OI (*intersect*) model. Results marked with ♣ denote statistical significance at the $p < 0.01$ level in comparison to the state-of-the-art neural network model of Hashimoto et al. (2014). The method of Steiger (1980) is used as a statistical test.

As the results in Table 5.10 show, the APT models[19] with distributional inference outperform comparable sparse untyped count-based models of Mitchell and Lapata (2010) and Blacoe and Lapata (2012), using pointwise multiplication as composition function, and Kiela et al. (2014) using pointwise addition. Furthermore, APTs in combination with combined standard distributional inference and offset inference outperform the neural network model of Hashimoto et al. (2014), and achieve a new state-of-the-art on the ML2010 dataset. This shows that distributional inference in conjunction with Anchored Packed Trees is able to bridge the performance gap between interpretable high-dimensional explicit word representations and low-dimensional

19 In Kober et al. (2016) we showed that the positive effect of distributional inference furthermore transfers over to untyped count-based sparse word representations, albeit the benefit is smaller than for APTs.

distributed neural network models for the ML2010 short phrase composition tasks.

Table 5.11 lists the optimal number of neighbours for all tasks of the ML2010 dataset, determined on the respective development sets for standard distributional inference, offset inference, and combined distributional and offset inference.

| Dataset | DI | OI | DI+OI |
|---|---|---|---|
| ML2010 (adjective-nouns) | 10/30 | 30/30 | 10/30 |
| ML2010 (noun-nouns) | 50/5000 | 10/1000 | 10/1000 |
| ML2010 (verb-objects) | 400/10 | 10/400 | 10/400 |

Table 5.11: Number of neighbours used for each task of the ML2010 dataset. The notation $x/y$ refers to the number of neighbours used for composition by intersection $x$, and the number of neighbours used for composition by union $y$.

## 5.4 DISTRIBUTIONAL COMPOSITION AND DISTRIBUTIONAL INFERENCE

If untyped distributional semantic word representations are composed with a pointwise arithmetic composition function[20], then distributional inference and distributional composition are modelled by the same algebraic mechanism as in the algorithms of Kintsch (2001) and Utsumi (2009). With the generalisation of the offset inference algorithm, distributional composition and distributional inference in Anchored Packed Trees are also both realised by the same mechanism — an offset followed by merging the features of two or more representations.

In essence, distributional composition can be interpreted as an inference process that, when given two or more lexemes in some grammatical construction, needs to *infer* the distributional features that are plausible in the combined expression. This relation creates an interesting dynamic between distributional inference and composition by intersection when used in a complementary manner. The inference process can be used as a method for *co-occurrence embellishment*, which adds missing information to a representation, however with the risk of introducing co-occurrences implausible for the current context. For example, given the phrase *river bank*, distributional inference will likely infer knowledge for the lexeme *bank* that concerns its "financial institution" meaning, which however, is not relevant to the

20 For example pointwise min, max, addition or multiplication.

phrase *river bank*. Therefore, an intersective composition function can be used as a process of *co-occurrence filtering*, that is leveraging the enriched representations, while filtering out co-occurrences unsuitable for the current context.

Table 5.12 below highlights the interplay between distributional composition and inference. The table shows the 10 nearest neighbours, corresponding to the meaning expressed by the subsequent use in a phrase, for a number of ambiguous lexemes. The superscripted annotation denotes whether the neighbour corresponds to the first (1) or second (2) phrase. The number in parentheses denotes the rank of the given neighbour. The last column shows the 5 nearest neighbours of the composed phrases. For this experiment, the APT baseline model, together with composition by intersection and 50-100 neighbours for the offset inference algorithm, has been used.

| Lexeme | Neighbours | Phrase | Neighbours |
|---|---|---|---|
| *bank* | company$^1$(1), firm$^1$(2), fund$^1$(5), office$^1$(7), business$^1$(13), bridge$^2$(66), shore$^2$(68), coast$^2$(69), beach$^2$(97), trip$^2$(98) | *bank account*<br>*river bank* | account, bank, deposit, loan, customer<br>bank, thames, bridge, valley, river |
| *novel* | article$^1$(5), papers$^1$(9), notion$^1$(21), journal$^1$(23), innovative$^1$(25), poem$^2$(1), fiction$^2$(2), poetry$^2$(3), story$^2$(4), essay$^2$(10) | *novel method*<br>*romantic novel* | method, technique, approach, concept, procedure<br>novel, fiction, poem, poetry, story |
| *plant* | tree$^1$(1), shrub$^1$(2), flower$^1$(3), grass$^1$(13), crop$^1$(14), factory$^2$(4), station$^2$(7), equipment$^2$(8), building$^2$(10), industry$^2$(16) | *plant cell*<br>*power plant* | cell, tissue, plant, species, extract<br>plant, station, factory, equipment, unit |
| *tear* | anger$^1$(2), scream$^1$(21), cry$^1$(22), disappointment$^1$(24), yell$^1$(26), rip$^2$(1), drag$^2$(4), pull$^2$(8), sweep$^2$(10), toss$^2$(11) | *false tear*<br>*tear apart* | tear, accusation, promise, hurry, surprise<br>tear, pull, drag, push, shake |

Table 5.12: 10 neighbours, with their ranks in parentheses, for a number of ambiguous lexemes and their use in phrases which disambiguates their meaning. Despite inferring co-occurrences from different meanings of an ambiguous lexeme, composition by intersection is able to appropriately contextualise the meaning of the given ambiguous lexeme as shown by the 5 nearest neighbours of each phrase. Superscripts for the neighbours of the lexemes denote whether the neighbour is indicative of the meaning expressed by the first (1) or second (2) phrase.

As the example for the ambiguous lexeme *bank* shows, the nearest neighbour expressing the “sloping land" sense of *bank* has only rank 66. This means that the distributional inference process is predominantly enriching the APT representation of *bank* with knowledge concerning its “financial institution" meaning. Nonetheless, composition

by intersection is able to recover the correct meaning of *bank* in the phrase *river bank*, as its 5 nearest neighbours show. For the other ambiguous lexemes, the respective neighbours are relatively balanced between the meanings expressed in the short phrases. This leads to a distributional inference process that embellishes a given elementary APT representation with co-occurrences from all meanings[21] of an ambiguous lexeme. However, composition by intersection is able to filter most implausible co-occurrences and thereby recover the intended meanings of the lexemes in context.

While composition by intersection has the capability to filter out unrelated co-occurrences by itself, its strong discriminatory nature leads to the issue of data sparsity. Therefore, it requires a support mechanism that provides additional data to ease the sparsity effect. As the empirical work in this thesis has shown, distributional inference represents such a supporting mechanism that is able to overcome the sparsity issue while maintaining the discriminatory power of the composition function.

## 5.5 SUMMARY

This chapter has provided an analysis of the issue of data sparsity which is the result of not observing all plausible co-occurrences for any given lexeme. Subsequently, an unsupervised algorithm has been proposed to explicitly infer missing co-occurrence events and thereby provide a mechanism to substantially ease the data sparsity problem. Furthermore, the standard distributional inference algorithm has been generalised to offset inference in the scope of Anchored Packed Trees.

The distributional inference algorithms have been qualitatively analysed in order to characterise the knowledge that is being inferred, and have been put into context with earlier work as well as recent developments using data augmentation. The merit of distributional inference has been empirically validated on a number of popular word similarity tasks, as well as a short phrase composition dataset. The results demonstrate substantial and statistically significant improvements over a baseline without distributional inference, and are closing the performance gap between high-dimensional interpretable models and low-dimensional uninterpretable models for short phrase

21 All meanings present among the top $n$ nearest neighbours in the given source corpus.

composition tasks. While for the short phrase composition tasks significant performance improvements could be observed with any amount of available data, the improvements due to distributional inference become smaller with more available data on the word similarity tasks.

Furthermore, the limitations of the proposed algorithms have been analysed, highlighting that the number of neighbours used for inference has a significant impact on the resulting distributional characteristics of the enriched representations as well as on task performance.

Lastly, the chapter has uncovered a latent relation between distributional inference and distributional composition, and has highlighted their complementary nature with an intersective composition function.

# 6

# CONCLUSION

This section highlights the main contributions of this thesis (§ 6.1), summarises the thesis as a whole (§ 6.2) and provides an overview of possible directions for future work (§ 6.3).

## 6.1 MAIN CONTRIBUTIONS

This thesis contributed a practical evaluation of the APT theory, together with a characterisation of the distributional semantic space that elementary, offset and composed APT representations give rise to. Then, the thesis analysed and addressed the data sparsity problem in the Anchored Packed Trees framework. Data sparsity represented a central challenge to composition in APTs because of their rich type structure, which results in a very sparse and high-dimensional distributional space. The proposed algorithms for explicitly inferring missing co-occurrence events from the distributional neighbourhood have been shown to successfully alleviate the sparsity problem. The thesis examined the kind of knowledge that can be inferred from the distributional neighbourhood, as well as quantifying the impact of distributional inference on the semantic space. Subsequently, the thesis showed that the use of any form of distributional inference resulted in statistically significant performance improvements on a range of word similarity tasks as well as a short phrase composition task.

## 6.2 SUMMARY

To summarise, the thesis contextualised the APT framework and the distributional inference algorithm with related work concerning distributional semantics, compositional distributional semantics, modelling word meaning in context and inferring unobserved events in Chapter 2. The Anchored Packed Trees framework (Weir et al., 2016) has been reviewed in Chapter 3. Subsequently, a practical evaluation of the APT theory on the basis of a large-scale hyperparameter sensitivity analysis was provided in Chapter 4. In addition, Chapter 4

derived a robust set of favourable parameter settings which have been shown to work well on a number of popular word similarity datasets as well as on a short phrase composition benchmark. Furthermore, Chapter 4 contributed a characterisation of the distributional space that APTs give rise to and confirmed previous findings in the literature that the neighbourhood of typed distributional semantic models tends to be governed by co-hyponymy. Lastly, Chapter 4 studied the distributional semantics of offset APT representations as well as composed APT representations, showing that offset APTs provide a complementary view of the semantics of a lexeme and that adjective-noun composition preserves the general characteristics of the head noun.

Chapter 5 analysed the issue of data sparsity which is inherent in natural language, and stems from not observing all plausible co-occurrences for any given lexeme in a source corpus. Instead of using various dimensionality reduction techniques which would render the distributional space uninterpretable, Chapter 5 proposed an unsupervised algorithm to learn about unobserved co-occurrence events in distributional space by explicitly inferring them from their neighbours. The distributional inference algorithm is based on smoothing approaches in the language modelling community (Essen and Steinbiss, 1992; Dagan et al., 1993) and can be interpreted as a soft-clustering algorithm where any given lexeme is the centroid of the cluster formed by the weighted average of its distributional neighbours.

In the following Chapter 5 showed that the distributional inference algorithms successfully alleviates the data sparsity problem, resulting in statistically significant performance improvements for all datasets used in this thesis. Furthermore, the distributional inference algorithm has been generalised within the APT framework to offset inference in order to effectively leverage the rich type structure in APTs. Furthermore, Chapter 5 investigated how much data the distributional inference algorithm can make up for and found that DI is able to improve performance at any (reasonably large) amount of data, and is especially beneficial in combination with an intersective composition function.

In addition to the quantitative studies of the distributional and offset inference algorithms, Chapter 5 contributed a qualitative analysis that explored *how* inferring additional knowledge changes the charac-

teristics of the distributional space, and furthermore *what* knowledge can possibly be inferred.

Lastly, Chapter 5 highlighted the close relation between distributional composition and distributional inference which both are realised by the same mechanism. This insight also provides an explanation why an intersective composition function benefits relatively more from an inference mechanism, as the two methods can be used in a complementary manner.

## 6.3 FUTURE WORK

The work in this thesis opened up a number of avenues for future work and I will briefly give an overview of possible routes. Directions for future work are categorised in work concerning the APT framework itself (§ 6.3.1), work involving the distributional inference algorithm (§ 6.3.2), and potential next tasks for applying APTs together with distributional inference (§ 6.3.3).

### 6.3.1 *Future Work on APTs*

The current instantiation of the Anchored Packed Trees framework uses a dependency grammar to model relations between lexemes, but the theory itself is agnostic to the concrete grammatical formalism used. One problem with using syntactic dependencies in a semantic framework is that the type structure is frequently too fine-grained. For example for a typed distributional semantic representation, distinguishing the active from the passive voice as in *geese chase ducks* vs. *ducks are chased by geese*, is often undesirable as it leads to further scattering the knowledge about the entities involved. Therefore, one possible strand of future work would be to employ a different grammatical formalism such as Minimal Recursion Semantics (Copestake et al., 2005), or Combinatory Categorical Grammar (Steedman, 2000), which are able to abstract over such semantically equivalent classes.

An alternative strand of future work would be to either learn syntactic relations from data directly, or to derive a more coarse grained set of relations from the existing dependency structure, by e.g. clustering similar dependency relations. Creating a more compact distributional space for APTs has the potential for improving the handling of longer phrases.

### 6.3.2 *Future Work on Distributional Inference*

Currently, the distributional inference algorithm suffers from the "cold-start" problem as it aims to create improved representations from a knowingly incomplete distributional model. One potential solution for the problem would be to leverage an existing lexical resource such as WordNet to provide an initial set of "good" neighbours, and subsequently switch to the unsupervised mode and infer knowledge from the distributional neighbourhood.

An alternative approach would be to apply an iterative inference algorithm that does not consume all of the top $n$ neighbours at once, but only uses the first 3-5 neighbours per iteration to gradually enrich the elementary representations. A further alternative would be to use a different source distributional model, or indeed an ensemble of distributional models, to query an initial set of neighbours from. However, this would only be applicable to the offset inference algorithm if the source distributional models are APTs themselves.

Another strand of future work would be to employ more sophisticated neighbour selection algorithms. Kintsch (2001) and Utsumi (2009) used relatively simple constraints to select neighbours, and it might be feasible to improve the distributional inference algorithm with e.g. knowledge-based neighbour selection constraints by leveraging resources such as WordNet. An example would be to restrict that any distributional neighbour to be within some edge distance in WordNet, or even to impose the constraint that any neighbour must be in a particular semantic relation, such as hypernymy, to the target lexeme. A further option to extending the distributional inference algorithm would be through a clustering of the neighbours of a lexeme. Subsequently, only a subset of the (highest-ranking) features from each cluster would be selected for inference. This has the potential of achieving a tighter inference process that is reducing the amount of inferred noise due to relying on a larger amount of evidence for a co-occurrence event.

### 6.3.3 *Task-based Future Work*

There are numerous tasks to which Anchored Packed Trees in general and distributional inference in particular would be well suited. For example, potential tasks for assessing the contextualisation beha-

viour in a more large-scale setting would be the lexical substitution task (McCarthy and Navigli, 2007), setup as a paraphrase ranking problem as in Erk and Padó (2008) and Thater et al. (2010, 2011), the "Usage Similarity (USIM)" task of Erk et al. (2013) or the dictionary definition task that we proposed in Kober et al. (2017b).

A very interesting route for further study would be the analysis of the entailment properties of distributional inference, optionally in conjunction with distributional composition, in further detail. A particularly fruitful avenue for further research would be the automatic distinction of co-hyponyms and hypernyms based on the findings in Chapter 5 (see Section 5.3.2). These show that offset inference together with a large SPPMI shift leads to a distributional space with very high similarity scores for co-hyponyms, and very low similarity scores for all other relations.

Lastly, measuring the impact of distributional inference on the selectional preferences of common verbs would be a further feasible task for evaluating the impact of DI on the distributional semantics of APT representations.

# A

# APPENDIX

Table A.1 lists the 20 additional adjectives that have been manually added to the list of 55 adjectives occurring in the 72 distinct adjective-noun phrases in the Mitchell and Lapata (2010) dataset, alongside their frequency in the BNC, in order to study the distributional semantics of offset representations in section 4.3.2, and specifically to investigate their distributional neighbours (see Table 4.9).

| **Lexeme** | **Frequency in BNC** |
|---|---|
| ancient | 4 946 |
| blonde | 1 062 |
| blue | 10 042 |
| boring | 1 481 |
| clever | 2 238 |
| disgusting | 464 |
| dumb | 728 |
| exciting | 3 261 |
| green | 14 863 |
| long | 56 301 |
| nasty | 1 809 |
| new | 124 114 |
| old | 53 171 |
| pretty | 7 570 |
| red | 14 960 |
| sexy | 634 |
| shiny | 692 |
| short | 19 721 |
| smart | 1 833 |
| ugly | 1 302 |

Table A.1: List of adjectives additionally added to the study of the distributional semantics of offset APT representations.

# B

## APPENDIX

Table B.1 lists the 72 BLESS concepts alongside their most frequent adjectival modifiers where the adjective-noun compound had more than 50 occurrences in the BNC. The extraction of adjective-noun compounds was done on the basis of an `amod` relation between the two constituents. As the table below shows, some of the extracted adjective-noun pairs are due to parsing errors, because for example the phrase *Edinburgh castle* represents a named entity — or at least a noun-noun compound. In order to be consistent between the APT space[1] and the dataset, the parsing errors have not been corrected manually. This subset of BLESS has been used for characterising the distributional semantics of offset APT representations (see section 4.3.2) and composed APT representations (see section 4.3.3).

| **BLESS concept** | **Modifier (AN frequency in BNC)** |
|---|---|
| ambulance | john (60) |
| apple | big (51) |
| bag | carrier (186), paper (152), piping (55), plastic (339), polythene (86), shopping (91), sleep (184), tea (80) |
| bear | polar (80), teddy (186) |
| bed | double (251), flower (112), four-poster (58), hospital (166), main (67), own (95), river (77), sea (81), single (152) |
| birch | silver (50) |
| blouse | white (62) |
| bomb | atom (54), atomic (137), car (117), hydrogen (54), ira (122), mortar (50), nuclear (58), petrol (60) |
| bottle | empty (58), glass (57), milk (91), plastic (70), water (97) |
| bowl | large (74), rose (53) |

1 The creation of the APT spaces relied on the same parsed corpus as the extraction of frequent adjective-noun compounds.

| | |
|---|---|
| box | ballot (108), black (134), cardboard (248), deposit (50), dialogue (58), dispatch (61), letter (138), little (91), metal (61), phone (117), po (247), signal (114), small (66), telephone (105), window (69), witness (67), wooden (87) |
| bull | groupe (51), machines (66), pit (58) |
| bus | local (126), school (65), scottish (54) |
| car | big (74), black (53), british (63), cable (78), company (239), diesel (70), dining (67), electric (57), estate (60), european (55), family (56), fast (80), first (59), hire (84), japanese (78), little (61), luxury (57), many (53), motor (500), new (534), old (114), other (168), own (156), park (130), patrol (82), police (370), private (130), racing (69), second-hand (60), small (91), sport (270), steal (202) |
| castle | barnard (219), edinburgh (68), medieval (56), windsor (148) |
| cat | big (63), black (77), domestic (60), pussy (73), wild (54) |
| chair | comfortable (55), deputy (68), easy (70), high (51), kitchen (55), wooden (62) |
| coat | black (56), fur (103), white (170) |
| cottage | country (67), little (52), rose (57), small (56), tie (57) |
| cow | dairy (70) |
| deer | red (106) |
| desk | reception (138) |
| dolphin | river (53) |
| dress | black (122), blue (58), cotton (50), evening (88), fancy (120), new (64), red (50), silk (59), wedding (134), white (73), golden (113) |
| elephant | african (52) |
| fighter | fire (67) |

| | |
|---|---|
| glove | rubber (59) |
| goat | mountain (85) |
| guitar | acoustic (87), electric (82) |
| gun | big (74), machine (233) |
| hat | black (66), bowler (78), hard (51), straw (116), top (109) |
| herring | red (70) |
| horse | black (51), other (109), white (183), young (74) |
| hospital | college (50), cross (55), day (81), district (52), general (588), local (172), london (119), maternity (78), memorial (163), mental (230), new (72), nhs (53), other (66), park (69), private (104), psychiatric (161), radcliffe (79), royal (95), teaching (94), university (61), victoria (54) |
| hotel | grand (161), house (160), london (58), luxury (77), park (68), savoy (54), small (86), star (93) |
| jacket | dinner (63), leather (182), tweed (64) |
| jar | jam (71) |
| jet | jumbo (65) |
| knife | kitchen (53), sharp (104) |
| library | academic (65), bodleian (54), british (301), cdna (55), central (63), college (60), local (87), national (146), public (461), reference (72), research (52), school (249), university (172) |
| lion | british (90), red (89) |
| missile | ballistic (107), cruise (133), nuclear (70), scud (50) |
| oak | royal (81) |
| onion | spring (108) |
| oven | microwave (87) |
| owl | barn (240), brown (127), eagle (111), little (52), tawny (94) |
| phone | mobile (177) |
| pig | guinea (134) |

| | |
|---|---|
| pistol | sex (141) |
| potato | baked (57), jacket (77), new (71) |
| pub | local (146) |
| radio | bbc (237), car (54), community (58), local (255), national (66) |
| restaurant | chinese (67), italian (53), staff (51) |
| robin | sir (79) |
| salmon | smoked (93) |
| scarf | silk (53) |
| sheep | black (103) |
| shirt | cotton (54), silk (79), striped (50), white (214) |
| spoon | silver (50), wooden (76) |
| swan | black (50) |
| table | bedside (137), breakfast (87), coffee (158), dining (127), dinner (88), dress (100), follow (94), high (58), kitchen (282), league (279), little (60), long (59), low (58), negotiating (56), next (56), pool (53), round (241), side (61), small (130), trestle (56), water (93), wooden (76) |
| tanker | oil (73) |
| television | bbc (92), british (63), cable (85), central (53), circuit (52), colour (124), commercial (50), independent (122), national (66), satellite (75) |
| train | freight (67), passenger (100), royal (50), special (89), steam (111) |
| trout | brown (62) |
| van | transit (57) |
| villa | aston (340) |
| whale | killer (56), minke (53) |
| yacht | royal (63) |

Table B.1: List of adjectival modifiers of the 72 BLESS concepts used to characterise the distributional semantics of offset APT representations and composed APT representations.

# BIBLIOGRAPHY


E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado, June 2009. Association for Computational Linguistics. (Cited on pages 85 and 110.)

N. Asher, T. Van de Cruys, A. Bride, and M. Abrusán. Integrating type theory and distributional semantics: A case study on adjective–noun compositions. *Computational Linguistics*, 42(4):703–725, 2016. (Cited on page 37.)

A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382, Dec. 2005. (Cited on page 48.)

M. Baroni. Composition in distributional semantics. *Language and Linguistics Compass*, 7(10):511–522, 2013. (Cited on page 36.)

M. Baroni and A. Lenci. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4): 673–721, December 2010. (Cited on pages 22, 24, 52, 76, and 92.)

M. Baroni and A. Lenci. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, GEMS '11, pages 1–10, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-16-9. (Cited on pages 19, 25, 84, 87, 101, 107, and 108.)

M. Baroni and R. Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, MA, October 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D10-1115. (Cited on pages 7, 38, 39, 41, 43, and 68.)

M. Baroni, G. Dinu, and G. Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June 2014. Association for Computational Linguistics. (Cited on pages 19 and 38.)

M. Batchkarov. *Evaluating distributional models of compositional semantics*. PhD thesis, University of Sussex, 2016. (Cited on page 83.)

M. Batchkarov, T. Kober, J. Reffin, J. Weeds, and D. Weir. A critique of word similarity as a method of evaluating distributional semantic models. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 7–12. Association for Computational Linguistics, 2016. (Cited on pages 84 and 85.)

I. Beltagy, S. Roller, P. Cheng, K. Erk, and R. J. Mooney. Representing meaning with a combination of logical and distributional models. *Computational Linguistics*, 42(4):763–808, Dec. 2016. (Cited on page 37.)

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3: 1137–1155, March 2003. (Cited on page 21.)

T. Bergmanis, K. Kann, H. Schütze, and S. Goldwater. Training data augmentation for low-resource morphological inflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 31–39, Vancouver, August 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/K17-2002. (Cited on page 143.)

C. Biemann. Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80, New York City, June 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W06/W06-3812. (Cited on page 50.)

W. Blacoe and M. Lapata. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D12-1050. (Cited on pages 28 and 157.)

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003. (Cited on page 54.)

S. R. Bowman, J. Gauthier, A. Rastogi, R. Gupta, C. D. Manning, and C. Potts. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1466–1477, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1139. (Cited on page 7.)

A. Bride, T. Van de Cruys, and N. Asher. A generalisation of lexical functions for composition in distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 281–291, Beijing, China, July 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-1028. (Cited on pages 40 and 44.)

E. Bruni, N. K. Tran, and M. Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47, 2014. (Cited on pages 84 and 85.)

J. A. Bullinaria and J. P. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, pages 510–526, 2007. (Cited on page 19.)

J. A. Bullinaria and J. P. Levy. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd. *Behavior Research Methods*, 44(3):890–907, 2012. (Cited on pages 13, 19, and 88.)

L. Burnard. The british national corpus. http://www.natcorp.ox.ac.uk/XMLedition/URG/, 2007. URL http://www.natcorp.ox.ac.uk/XMLedition/URG/. (Cited on pages 10 and 82.)

J. Caron. Experiments with lsa scoring: Optimal rank and basis. In *In Proceedings of the SIAM Computational information Retrieval Workshop*, pages 157–169, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics. (Cited on page 13.)

K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014. (Cited on page 143.)

Q. Chen, X. Zhu, Z. Ling, S. Wei, and H. Jiang. Enhancing and combining sequential and tree LSTM for natural language inference. *CoRR*, abs/1609.06038, 2016. URL http://arxiv.org/abs/1609.06038. (Cited on page 34.)

X. Chen, Z. Liu, and M. Sun. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, Doha, Qatar, October 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1110. (Cited on page 45.)

C. Chiarello, C. Burgess, L. Richards, and A. Pollock. Semantic and associative priming in the cerebral hemispheres: Some words do, some words don't ... sometimes, some places. *Brain and Language*, 38:75–104, 1990. (Cited on pages 109 and 110.)

K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 76–83, Vancouver, British Columbia, Canada, June 1989. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P89-1010. (Cited on pages 11, 18, 20, and 92.)

S. Clark and S. Pulman. Combining symbolic and distributional models of meaning. In *AAAI Spring Symposium: Quantum Interaction*, pages 52–55. AAAI, 2007. (Cited on pages 27 and 37.)

D. Clarke. A context-theoretic framework for compositionality in distributional semantics. *Computational Linguistics*, 38, 2012. (Cited on page 27.)

B. Coecke, M. Sadrzadeh, and S. Clark. Mathematical foundations for a compositional distributed model of meaning. *Linguistic Analysis*, 36(1-4):345–384, 2011. (Cited on pages 27, 37, 38, 41, 43, and 44.)

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167, New York, NY, USA, 2008. (Cited on page 21.)

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, November 2011. (Cited on pages 10, 21, and 32.)

A. Copestake and A. Herbelot. Lexicalised compositionality, 2012. (Cited on page 65.)

A. Copestake, D. Flickinger, C. Pollard, and I. A. Sag. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2):281–332, July 2005. (Cited on page 164.)

J. Curran. *From Distributional to Semantic Similarity*. PhD thesis, University of Edinburgh, 2004. (Cited on pages 20, 25, 88, and 93.)

J. R. Curran and M. Moens. Improvements in automatic thesaurus extraction. In *ACL-SIGLEX Workshop on Unsupervised Lexical Acquisition*, Philadelphia, 2002. (Cited on page 22.)

I. Dagan, S. Marcus, and S. Markovitch. Contextual word similarity and estimation from sparse data. In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, pages 164–171, Stroudsburg, PA, USA, 1993. Association for Computational Linguistics. (Cited on pages 14, 20, 57, 70, 92, 120, 121, and 163.)

I. Dagan, F. Pereira, and L. Lee. Similarity-based estimation of word cooccurrence probabilities. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 272–278, Las Cruces, New Mexico, USA, June 1994. Association for Computational Linguistics. (Cited on page 57.)

I. Dagan, L. Lee, and F. Pereira. Similarity-based methods for word sense disambiguation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 56–63, Madrid, Spain, July 1997. Association for Computational Linguistics. (Cited on page 57.)

A. G. Dale and N. Dale. Some clumping experiments for associative document retrieval. *American Documentation*, 16(1):5–9, January 1965. (Cited on page 18.)

F. de Saussure. *Cours de linguistique générale*. v.C. Bally and A. Sechehaye (eds.), Paris/Lausanne, 1916. English translation: Course in General Linguistics. London: Peter Owen, 1960. (Cited on page 18.)

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. (Cited on pages 18 and 20.)

I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1-2):143–175, January 2001. (Cited on page 49.)

C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927, Apr. 2008. (Cited on page 54.)

G. Dinu and M. Lapata. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1162–1172, Cambridge, MA, October 2010. Association for Computational Linguistics. (Cited on pages 53, 54, 55, and 56.)

G. Dinu, N. T. Pham, and M. Baroni. General estimation and evaluation of compositional distributional semantic models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 50–58, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-3206. (Cited on page 29.)

C. Dyer, A. Kuncoro, M. Ballesteros, and N. A. Smith. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California, June 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N16-1024. (Cited on page 34.)

B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. CRC press, 1994. (Cited on page 136.)

J. L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. (Cited on page 34.)

K. Erk. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653, 2012. (Cited on page 19.)

K. Erk. What do you know about an alligator when you know the company it keeps? *Semantics and Pragmatics*, 9(17):1–63, April 2016. (Cited on page 58.)

K. Erk and S. Padó. A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 897–906, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D08-1094. (Cited on pages 6, 51, 52, 53, 61, 77, 78, and 166.)

K. Erk and S. Pado. Exemplar-based models for word meaning in context. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 92–97, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P10-2017. (Cited on pages 49, 53, and 61.)

K. Erk, S. Padó, and U. Padó. A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4):723–763, 2010. (Cited on page 56.)

K. Erk, D. McCarthy, and N. Gaylord. Measuring word meaning in context. *Computational Linguistics*, 39(3):511–554, 2013. (Cited on page 166.)

U. Essen and V. Steinbiss. Co-occurrence smoothing for stochastic language modeling. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-92)*, pages 161–164. IEEE, March 1992. (Cited on pages 14, 57, and 163.)

S. Evert. *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. PhD thesis, University of Stuttgart, 2005. (Cited on page 93.)

M. Fadaee, A. Bisazza, and C. Monz. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-2090. (Cited on page 143.)

M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35. Association for Computational Linguistics, 2016. (Cited on page 84.)

C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998. (Cited on pages 86, 87, and 111.)

A. Ferraresi, E. Zanchetta, M. Baroni, and S. Bernardini. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the WAC4 Workshop at LREC*, 2008. (Cited on pages 87 and 138.)

L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*, pages 406–414, New York, NY, USA, 2001. ACM. (Cited on pages 84 and 85.)

J. R. Firth. The technique of semantics. *Transactions of the Philological Society*, 34(1):36–73, 1935. (Cited on page 18.)

J. R. Firth. A synopsis of linguistic theory, 1930–1955. In *Studies in Linguistic Analysis*, pages 1–32. Oxford: Philological Society, 1957. (Cited on pages 4 and 18.)

G. Frege. *Die Grundlagen der Arithmetik: Eine logisch mathematische Untersuchung über den Begriff der Zahl*. W. Koebner, 1884. (Cited on pages 5 and 26.)

G. Frege. Über sinn und bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50, 1892. (Cited on page 4.)

D. Fried, T. Polajnar, and S. Clark. Low-rank tensors for verbs in compositional distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 731–736, Beijing, China, July 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-2120. (Cited on page 44.)

W. A. Gale and K. W. Church. What's wrong with adding one. In *Corpus-Based Research into Language*. Rodolpi, 1994. (Cited on page 56.)

M. Ganesalingam and A. Herbelot. Composing distributions: mathematical structures and their linguistic interpretation. Technical report, University of Cambridge, 2013. (Cited on pages 29 and 74.)

J. Ganitkevitch, B. Van Durme, and C. Callison-Burch. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL http://cs.jhu.edu/~ccb/publications/ppdb.pdf. (Cited on page 156.)

E. Gaussier and C. Goutte. Relation between plsa and nmf and implications. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 601–602, New York, NY, USA, 2005. ACM. (Cited on page 54.)

Y. Goldberg. *Neural Network Methods for Natural Language Processing*, volume 37 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool, San Rafael, CA, 2017. ISBN 978-1-62705-298-6. (Cited on page 31.)

E. Grefenstette and M. Sadrzadeh. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404, Edinburgh, Scotland, UK., July 2011a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D11-1129. (Cited on pages 7, 41, 42, and 45.)

E. Grefenstette and M. Sadrzadeh. Experimenting with transitive verbs in a discocat. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 62–66, Edinburgh, UK, July 2011b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-2507. (Cited on pages 7, 41, and 42.)

E. Grefenstette, M. Sadrzadeh, S. Clark, B. Coecke, and S. Pulman. Concrete sentence spaces for compositional distributional models of meaning. *Proceedings of the 9th International Conference on Computational Semantics (IWCS 2011)*, pages 125–134, 2011. (Cited on pages 7 and 41.)

E. Grefenstette, G. Dinu, Y. Zhang, M. Sadrzadeh, and M. Baroni. Multi-step regression learning for compositional distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 131–142, Potsdam, Germany, March 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-0112. (Cited on pages 39, 40, and 43.)

G. Grefenstette. Use of syntactic context to produce term association lists for text retrieval. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 89–97, New York, NY, USA, 1992. ACM. (Cited on pages 22 and 25.)

E. Guevara. A regression model of adjective-noun compositionality in distributional semantics. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, pages 33–37, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W10-2805. (Cited on pages 30, 31, and 39.)

E. Guevara. Computing semantic compositionality in distributional semantics. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 135–144, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. (Cited on pages 30, 31, 35, 37, and 39.)

A. Gupta, J. Utt, and S. Padó. Dissecting the practical lexical function model for compositional distributional semantics. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 153–158, Denver, Colorado, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S15-1017. (Cited on page 45.)

M. U. Gutmann and A. Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(1):307–361, February 2012. (Cited on page 21.)

P. Hanks. Do word meanings exist? *Computers and the Humanities*, 34 (1–2):205–215, 2000. URL http://www.coli.uni-sb.de/~kowalski/senseval/hanks.pdf. (Cited on pages 7 and 45.)

K. E. Harper. Procedures for the determination of distributional classes. In *Proceedings of the International Conference on Machine Translation of Languages and Applied Language Analysis*, pages 688–698, Teddington, UK, September 1961. National Physical Laboratory. (Cited on page 18.)

K. E. Harper. Measurement of similarity between nouns. In *Proceedings of Coling*, pages 1–21, Bonn, Germany, 1965. Association for Computational Linguistics. (Cited on page 18.)

Z. Harris. Distributional structure. *Word*, 10:146–162, 1954. (Cited on pages 4 and 18.)

K. Hashimoto, P. Stenetorp, M. Miwa, and Y. Tsuruoka. Jointly learning word representations and composition functions using predicate-argument structures. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1544–1555, Doha, Qatar, October 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1163. (Cited on pages 156 and 157.)

D. G. Hays. Dependency theory: A formalism and some observations. *Language*, 40(4):511–525, 1964. (Cited on page 22.)

K. M. Hermann and P. Blunsom. The role of syntax in vector space models of compositional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 894–904, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P13-1088. (Cited on pages 35 and 43.)

F. Hill, R. Reichart, and A. Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, December 2015. (Cited on pages 84 and 86.)

F. Hill, K. Cho, A. Korhonen, and Y. Bengio. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30, 2016. URL http://www.aclweb.org/anthology/Q/Q16/Q16-1002.pdf. (Cited on pages 7, 28, and 29.)

D. Hindle. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 268–275, Pittsburgh, Pennsylvania, USA, June 1990. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P90-1034. (Cited on pages 18, 22, 69, 70, and 97.)

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, nov 1997. (Cited on page 35.)

E. Huang, R. Socher, C. Manning, and A. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea, July 2012. Association for Computational Linguistics. (Cited on page 49.)

I. Iacobacci, M. T. Pilehvar, and R. Navigli. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 95–105, Beijing, China, July 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-1010. (Cited on pages 45, 50, and 51.)

M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China, July 2015. Association for Computational Linguistics. (Cited on pages 28 and 36.)

N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June 2014. Association for Computational Linguistics. (Cited on pages 7, 32, and 33.)

N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu. Neural machine translation in linear time. *CoRR*, abs/1610.10099, 2016. URL http://arxiv.org/abs/1610.10099. (Cited on page 32.)

H. Kamp and B. Partee. Prototype theory and compositionality. *Cognition*, 57(2):129–191, 1995. (Cited on page 117.)

D. Kartsaklis and M. Sadrzadeh. Prior disambiguation of word tensors for constructing sentence vectors. In *Proceedings of EMNLP*, pages 1590–1601. Association for Computational Linguistics, 2013. URL http://www.aclweb.org/anthology/D13-1166. (Cited on pages 43 and 48.)

D. Kartsaklis and M. Sadrzadeh. A study of entanglement in a categorical framework of natural language. In *Proceedings of the 11th Workshop on Quantum Physics and Logic (QPL)*, 2014. (Cited on page 112.)

D. Kartsaklis, M. Sadrzadeh, and S. Pulman. A unified sentence space for categorical distributional-compositional semantics: Theory and experiments. In *Proceedings of COLING 2012: Posters*, pages 549–558, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL http://www.aclweb.org/anthology/C12-2054. (Cited on pages 42 and 43.)

D. Kartsaklis, M. Sadrzadeh, and S. Pulman. Separating disambiguation from composition in distributional semantics. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 114–123, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-3513. (Cited on page 48.)

D. Kartsaklis, N. Kalchbrenner, and M. Sadrzadeh. Resolving lexical ambiguity in tensor regression models of meaning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 212–217, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P14-2035. (Cited on page 43.)

S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. In *IEEE Transactions on Acoustics, Speech and Singal processing*, volume ASSP-35, pages 400–401, March 1987. (Cited on page 57.)

D. Kiela and S. Clark. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. (Cited on pages 19, 70, and 92.)

D. Kiela, F. Hill, A. Korhonen, and S. Clark. Improving multi-modal representations using image dispersion: Why less is sometimes more. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 835–841, Baltimore, Maryland, June 2014. Association for Computational

Linguistics. URL http://www.aclweb.org/anthology/P14-2135. (Cited on pages 88, 97, 139, and 157.)

D. Kiela, F. Hill, and S. Clark. Specializing word embeddings for similarity or relatedness. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2044–2048, Lisbon, Portugal, September 2015. Association for Computational Linguistics. (Cited on page 25.)

A. Kilgarriff. "i don't believe in word senses". *Computers and the Humanities*, 31(2):91–113, 1997. (Cited on page 45.)

A. Kilgarriff, P. Rychlý, P. Smrž, and D. Tugwell. The sketch engine. *Information Technology*, 2004. (Cited on page 50.)

Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. (Cited on pages 7, 32, and 33.)

W. Kintsch. Predication. *Cognitive Science*, 25, 2001. (Cited on pages 26, 58, 59, 123, 141, 142, 158, and 165.)

E. Kiperwasser and Y. Goldberg. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327, 2016. URL https://transacl.org/ojs/index.php/tacl/article/view/885. (Cited on page 34.)

T. Kober, J. Weeds, J. Reffin, and D. Weir. Improving sparse word representations with distributional inference for semantic composition. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1691–1702, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1175. (Cited on pages 12, 100, 120, 125, 144, 156, and 157.)

T. Kober, J. Weeds, J. Reffin, and D. Weir. Improving semantic composition with offset inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 433–440, Vancouver, Canada, July 2017a. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-2069. (Cited on pages 14, 94, 120, and 156.)

T. Kober, J. Weeds, J. Wilkie, J. Reffin, and D. Weir. One representation per word - does it make sense for composition? In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 79–90, Valencia, Spain, April 2017b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W17-1910. (Cited on pages 28, 29, 46, 51, and 166.)

T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, Aug. 2009. (Cited on pages 24 and 44.)

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. (Cited on pages 32 and 143.)

J. Lambek. Type grammars as pregroups. *Grammars*, 4(1):21–39, 2001. (Cited on page 37.)

J. Lambek. From word to sentence: A computational algebraic approach to grammar. Technical report, McGill University, 2008. (Cited on page 37.)

G. Lapesa and S. Evert. A large scale evaluation of distributional semantic models: Parameters, interactions and model selection. *TACL*, 2:531–545, 2014. (Cited on pages 13, 19, 88, 95, and 96.)

G. Lapesa and S. Evert. Large-scale evaluation of dependency-based dsms: Are they worth the effort? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 394–400, Valencia, Spain, April 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E17-2063. (Cited on pages 13, 19, and 88.)

P. Le and W. Zuidema. The forest convolutional network: Compositional distributional semantics with a neural chart and without binarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1155–1164, Lisbon, Portugal, September 2015. Association for Computational Linguistics. (Cited on page 32.)

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, December 1989. (Cited on page 32.)

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001. (Cited on pages 20 and 54.)

L. Lee. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, College Park, Maryland, USA, June 1999. Association for Computational Linguistics. doi: 10.3115/1034678.1034693. (Cited on page 18.)

O. Levy and Y. Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June 2014a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P14-2050. (Cited on pages 11, 19, 24, 25, 101, 107, and 110.)

O. Levy and Y. Goldberg. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc., 2014b. (Cited on pages 21 and 92.)

O. Levy, Y. Goldberg, and I. Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015. ISSN 2307-387X. (Cited on pages 13, 20, 88, 93, and 97.)

M. Lewis and M. Steedman. Combined distributional and logical semantics. In *Transactions of the Association for Computational Linguistics*, pages 179–192, 2013. (Cited on pages 26, 36, and 37.)

J. Li and D. Jurafsky. Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1722–1732, Lisbon, Portugal, September 2015. Association for Computational Linguistics. (Cited on page 45.)

D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 768–774, Montreal, Quebec, Canada, August 1998. Association for Computational Linguistics. (Cited on pages 18, 22, and 25.)

D. Lin and P. Pantel. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360, Dec. 2001. ISSN 1351-3249. doi: 10.1017/S1351324901002765. (Cited on page 22.)

H. Liu and P. Singh. Conceptnet — a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, Oct. 2004. (Cited on page 87.)

Y. Liu, C. Sun, L. Lin, and X. Wang. Learning natural language inference using bidirectional LSTM model and inner-attention. *CoRR*, abs/1605.09090, 2016. URL http://arxiv.org/abs/1605.09090. (Cited on page 34.)

W. Lowe. Towards a theory of semantic space. In *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society*, pages 576–581, 2001. (Cited on pages 18 and 19.)

K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, 1996. (Cited on page 20.)

J. Maillard, S. Clark, and E. Grefenstette. A type-driven tensor-based semantics for ccg. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 46–54, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-1406. (Cited on pages 43 and 45.)

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715. (Cited on page 124.)

C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. (Cited on page 82.)

M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, June 1993. (Cited on page 83.)

D. McCarthy and R. Navigli. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S/S07/S07-1009. (Cited on pages 49 and 166.)

K. McRae, G. S. Cree, M. S. Seidenberg, and C. Mcnorgan. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547–559, November 2005. (Cited on pages 58 and 87.)

O. Melamud, I. Dagan, and J. Goldberger. Modeling word meaning in context with substitute vectors. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 472–482, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1050. (Cited on page 51.)

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. (Cited on page 21.)

G. A. Miller and W. G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991. (Cited on page 84.)

J. Mitchell and M. Lapata. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P08/P08-1028. (Cited on pages 7, 26, 27, 28, 29, 52, 56, 59, 112, and 157.)

J. Mitchell and M. Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429, 2010. (Cited on pages 7, 28, 29, 50, 59, 84, 86, 112, 122, 140, 147, 156, 157, and 167.)

A. Mnih and G. E. Hinton. A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1081–1088. Curran Associates, Inc., 2009. (Cited on page 21.)

A. Mnih and K. Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2265–2273. Curran Associates, Inc., 2013. (Cited on page 21.)

R. Montague. English as a formal language. In B. Visentini, editor, *Linguaggi nella società e nella tecnica*, pages 189–223. 1970. (Cited on page 36.)

T. Moon and K. Erk. An inference-based model of word meaning in context as a paraphrase distribution. *ACM Transactions on Intelligent Systems and Technology*, 4(3):42:1–42:28, July 2013. (Cited on pages 55 and 56.)

F. Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of AISTATS*, pages 246–252. Society for Artificial Intelligence and Statistics, 2005. (Cited on page 21.)

L. Mou, H. Peng, G. Li, Y. Xu, L. Zhang, and Z. Jin. Discriminative neural sentence modeling by tree-based convolution. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2315–2325, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL http://aclweb.org/anthology/D15-1279. (Cited on page 32.)

L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin. How transferable are neural networks in nlp applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1046. (Cited on pages 7 and 31.)

R. Navigli and S. P. Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012. (Cited on page 50.)

D. L. Nelson, C. L. McEvoy, and T. A. Schreiber. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407, August 2004. (Cited on page 86.)

Y. Niwa and Y. Nitta. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, pages 304–309, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. doi: 10.3115/991886.991938. (Cited on page 70.)

D. Ó Séaghdha and A. Copestake. Co-occurrence contexts for noun compound interpretation. In *Proceedings of the Workshop on a Broader Perspective on Multiword Expressions*, pages 57–64. Association for Computational Linguistics, 2007. (Cited on page 70.)

D. Ó Séaghdha and A. Korhonen. Probabilistic models of similarity in syntactic context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1047–1057, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D11-1097. (Cited on page 55.)

S. Padó and M. Lapata. Constructing semantic space models from parsed corpora. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Sapporo, Japan, July 2003. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P03-1017. (Cited on page 75.)

S. Padó and M. Lapata. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199, June 2007. (Cited on pages 22, 23, 24, 26, 52, 53, 63, 66, 68, 69, 75, 76, 77, 88, 91, and 95.)

D. Paperno, N. T. Pham, and M. Baroni. A practical and linguistically-motivated approach to compositional distributional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 90–99, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P14-1009. (Cited on pages 40, 44, and 45.)

R. Parker, D. Graff, J. Kong, K. Chen, and K. Maeda. English gigaword fifth edition ldc2011t07. DVD, June 2011. (Cited on page 138.)

B. Partee. Lexical semantics in formal semantics: History and challenges. http://esslli2016.unibz.it/wp-content/uploads/2015/

10/ParteeESSLLI2016RefSemPlus.slides.pdf, August 2016. (Cited on pages 4 and 36.)

Y. Peirsman. Word space models of semantic similarity and relatedness. In *In Proceedings of the 13th ESSLLI Student Session*, pages 143–152, 2008. (Cited on pages 19, 25, 101, 107, and 110.)

F. J. Pelletier. The principle of semantic compositionality. *Topoi*, 13: 11–24, 1994a. (Cited on page 26.)

F. J. Pelletier. On an argument agains semantic compositionality. *Logic and Philosophy of Science*, pages 599–610, 1994b. (Cited on page 26.)

J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. (Cited on pages 11 and 20.)

N.-Q. Pham, G. Kruszewski, and G. Boleda. Convolutional neural network language models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1153–1162, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1123. (Cited on pages 32 and 33.)

T. A. Plate. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641, 1995. (Cited on page 27.)

T. Polajnar and S. Clark. Improving distributional semantic vectors through context selection and normalisation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 230–238, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. (Cited on pages 88 and 92.)

T. Polajnar, L. Fagarasan, and S. Clark. Reducing dimensions of tensors in type-driven distributional semantics. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1036–1046, Doha, Qatar, October 2014a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1111. (Cited on pages 44 and 45.)

T. Polajnar, L. Rimell, and S. Clark. Evaluation of simple distributional compositional operations on longer texts. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4440–4443, Reykjavik, Iceland, May 2014b. European Language Resources Association (ELRA). ISBN 978-2-9517408-8-4. URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/110_Paper.pdf. ACL Anthology Identifier: L14-1076. (Cited on page 56.)

J. B. Pollack. Recursive distributed representations. *Artificial Intelligence*, 46:77–105, 1990. (Cited on page 34.)

J. Pustejovsky. The generative lexicon. *Computational Linguistics*, 17 (4):409–441, Dec. 1991. (Cited on page 46.)

P. Rastogi, B. Van Durme, and R. Arora. Multiview lsa: Representation learning via generalized cca. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 556–566, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1058. (Cited on page 84.)

S. Reddy, D. McCarthy, and S. Manandhar. An empirical study on compositionality in compound nouns. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 210–218, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL http://www.aclweb.org/anthology/I11-1024. (Cited on page 50.)

J. Reisinger and R. J. Mooney. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117, Los Angeles, California, June 2010a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N10-1013. (Cited on pages 48, 49, and 54.)

J. Reisinger and R. J. Mooney. A mixture model with sharing for lexical semantics. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Cambridge, MA, October 2010b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D10-1114. (Cited on pages 48, 49, and 54.)

L. Rimell, J. Maillard, T. Polajnar, and S. Clark. Relpron: A relative clause evaluation data set for compositional distributional semantics. *Computational Linguistics*, 42(4):661–701, Dec. 2016. (Cited on pages 40, 44, and 45.)

S. Roller and K. Erk. Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2163–2172, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1234. (Cited on page 22.)

K. Rothenhäusler and H. Schütze. Unsupervised classification with dependency based word spaces. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 17–24, Athens, Greece, March 2009. Association for Computational

Linguistics. URL http://www.aclweb.org/anthology/W09-0203. (Cited on page 22.)

H. Rubenstein and J. B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965. (Cited on pages 18 and 84.)

D. Rubinstein, E. Levi, R. Schwartz, and A. Rappoport. How well do distributional models capture different types of semantic knowledge? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 726–730, Beijing, China, July 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-2119. (Cited on page 151.)

S. Rudolph and E. Giesbrecht. Compositional matrix-space models of language. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 907–916, Uppsala, Sweden, July 2010. Association for Computational Linguistics. (Cited on page 27.)

C. Ruhl. *On Monosemy: A Study in Linguistic Semantics*. SUNY series in linguistics. State University of New York Press, 1989. ISBN 9780887069468. (Cited on page 45.)

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 0-262-68053-X. (Cited on page 33.)

M. Sahlgren. *The Word-space model*. PhD thesis, University of Stockholm (Sweden), 2006. (Cited on pages 19, 20, 88, and 90.)

M. Sahlgren and J. Karlgren. Vector-based semantic analysis using random indexing for cross-lingual query expansion. *Lecture Notes in Computer Science*, 2406:169–176, 2002. (Cited on page 20.)

M. Sahlgren and A. Lenci. The effects of data size and frequency range on distributional semantic models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 975–980, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1099. (Cited on pages 13 and 19.)

A. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Ng. On random weights and unsupervised feature learning. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1089–1096, New York, NY, USA, June 2011. (Cited on page 32.)

S. Scheible, S. Schulte im Walde, and S. Springorum. Uncovering distributional differences between synonyms and antonyms in a word space model. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 489–497, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing. (Cited on page 93.)

T. Schnabel, I. Labutov, D. Mimno, and T. Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal, September 2015. Association for Computational Linguistics. (Cited on pages 106 and 140.)

M. Schuster and K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, Nov. 1997. (Cited on page 34.)

H. Schütze. Dimensions of meaning. In *Proceedings of ACM/IEEE Conference on Supercomputing*, pages 787–796. IEEE Computer Society Press, 1992. (Cited on page 47.)

H. Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, mar 1998. ISSN 0891-2017. (Cited on pages 47 and 48.)

R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1009. (Cited on page 143.)

W. Shalaby and W. Zadrozny. Measuring semantic relatedness using mined semantic analysis. *CoRR*, abs/1512.03465, 2015. URL http://arxiv.org/abs/1512.03465. (Cited on pages 83 and 84.)

M. Silfverberg, A. Wiemerslage, L. Liu, and L. J. Mao. Data augmentation for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99, Vancouver, August 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/K17-2010. (Cited on page 143.)

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL http://arxiv.org/abs/1409.1556. (Cited on page 32.)

R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 151–161, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. (Cited on pages 7 and 31.)

R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D12-1110. (Cited on page 37.)

R. Socher, A. Karpathy, Q. Le, C. Manning, and A. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014. (Cited on page 35.)

K. Spärck-Jones. *Synonymy and Semantic Classification*. PhD thesis, University of Cambridge, 1964. (Cited on page 18.)

K. Spärck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972. (Cited on page 20.)

M. Steedman. *The Syntactic Process*. MIT Press, 2000. (Cited on pages 37, 43, and 164.)

J. H. Steiger. Tests for comparing elements of a correlation matrix. *Psychological Bulletin*, 87(2):245, 1980. (Cited on pages 83, 84, 97, 98, 104, 130, 131, 144, 154, 156, and 157.)

K. Sugawara, M. Nishimura, and K. Toshioka. Isolated word recognition using hidden markov models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-92)*, pages 1–4. IEEE, March 1985. (Cited on page 57.)

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842. (Cited on page 32.)

K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July 2015. Association for Computational Linguistics. (Cited on page 35.)

Z. Teng and Y. Zhang. Head-lexicalised bidirectional tree lstms. *Transactions of the Association for Computational Linguistics*, 5:163–177, 2017. (Cited on page 34.)

L. Tesniére. *Eléments de Syntaxe Structurale*. Klincksieck, Paris, 1959. (Cited on page 22.)

S. Thater, H. Fürstenau, and M. Pinkal. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 948–957, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P10-1097. (Cited on pages 52, 53, 56, 68, 78, 79, 80, and 166.)

S. Thater, H. Fürstenau, and M. Pinkal. Word meaning in context: A simple and effective vector model. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1134–1143, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL http://www.aclweb.org/anthology/I11-1127. (Cited on pages 53, 60, 61, 80, and 166.)

R. Tian, N. Okazaki, and K. Inui. The mechanism of additive composition. *Machine Learning*, 106(7):1083–1130, 2017. ISSN 1573-0565. doi: 10.1007/s10994-017-5634-8. URL http://dx.doi.org/10.1007/s10994-017-5634-8. (Cited on page 29.)

M. Tsubaki, K. Duh, M. Shimbo, and Y. Matsumoto. Modeling and learning semantic co-compositionality through prototype projections and neural networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 130–140, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1014. (Cited on page 7.)

J. Turian, L.-A. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden, July 2010. Association for Computational Linguistics. (Cited on page 10.)

P. D. Turney. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416, Sept. 2006. (Cited on page 58.)

P. D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1): 141–188, jan 2010. ISSN 1076-9757. (Cited on page 19.)

A. Utsumi. Computational semantics of noun compounds in a semantic space model. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 1568–1573, Pasadena, California, USA, July 2009. (Cited on pages 59, 123, 141, 142, 158, and 165.)

A. Utsumi. Extending and evaluating a multiplicative model for semantic composition in a distributional semantic model. In *Proceedings of the 2011 International Conference on Cognitive Modeling*, pages 243–248, Berlin, Germany, April 2012. (Cited on pages 59, 60, and 123.)

T. Van de Cruys. Using three way data for word sense discrimination. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 929–936, Manchester, UK, August 2008. Coling 2008 Organizing Committee. URL http://www.aclweb.org/anthology/C08-1117. (Cited on page 55.)

T. Van de Cruys, T. Poibeau, and A. Korhonen. Latent vector weighting for word meaning in context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1012–1022, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. (Cited on page 55.)

E. M. Vecchi, M. Marelli, R. Zamparelli, and M. Baroni. Spicy adjectives and nominal donkeys: Capturing semantic deviance using compositionality in distributional spaces. *Cognitive Science*, pages 102–136, 2016. (Cited on page 39.)

J. Weeds and D. Weir. A general framework for distributional similarity. In M. Collins and M. Steedman, editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 81–88, 2003. (Cited on pages 18 and 140.)

J. Weeds, D. Clarke, J. Reffin, D. Weir, and B. Keller. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland, August 2014a. Dublin City University and Association for Computational Linguistics. URL http://www.aclweb.org/anthology/C14-1212. (Cited on page 22.)

J. Weeds, D. Weir, and J. Reffin. Distributional composition using higher-order dependency vectors. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 11–20, Gothenburg, Sweden, April 2014b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-1502. (Cited on page 88.)

D. Weir, J. Weeds, J. Reffin, and T. Kober. Aligning packed dependency trees: a theory of composition for distributional semantics. *Computational Linguistics, special issue on Formal Distributional Semantics*, 42(4):727–761, December 2016. (Cited on pages 6, 8, 46, 62, 64, 67, 68, 73, 81, 91, 94, 95, 157, and 162.)

D. Widdows. Semantic vector products: Some initial investigations. In *Proceedings of the Second Symposium on Quantum Interaction, Oxford, UK*, pages 1–8, 2008. (Cited on page 26.)

J. Wieting, M. Bansal, K. Gimpel, and K. Livescu. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, 3:345–358, 2015. URL http://www.aclweb.org/anthology/Q/Q15/Q15-1025.pdf. (Cited on pages 84 and 156.)

J. Wieting, M. Bansal, K. Gimpel, and K. Livescu. Towards universal paraphrastic sentence embeddings. In *Proceedings of the International Conference on Learning Representations*, 2016. (Cited on pages 28 and 36.)

B. Wilson. The unknown perils of mining wikipedia. https://blog.lateral.io/2015/06/the-unknown-perils-of-mining-wikipedia/, June 2015. (Cited on pages 5 and 138.)

M. A. Yatbaz, E. Sert, and D. Yuret. Learning syntactic categories using paradigmatic representations of word context. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 940–951, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D12-1086. (Cited on page 51.)

A. Yessenalina and C. Cardie. Compositional matrix-space models for sentiment analysis. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 172–182, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D11-1016. (Cited on page 27.)

F. M. Zanzotto, I. Korkontzelos, F. Fallucchi, and S. Manandhar. Estimating linear models for compositional distributional semantics. In *Proceedings of Coling*, pages 1263–1271, 2010. URL http://www.aclweb.org/anthology/C10-1142. (Cited on pages 7, 30, and 31.)

X. Zhu, P. Sobihani, and H. Guo. Long short-term memory over recursive structures. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1604–1612, Lille, France, 07–09 Jul 2015a. PMLR. URL http://proceedings.mlr.press/v37/zhub15.html. (Cited on page 35.)

Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, pages 19–27. IEEE Computer Society, 2015b. (Cited on page 138.)