UNIVERSITY OF CAMBRIDGE

Representing Meaning in Continuous Space: From Words to Sentences

Felix Hill

This dissertation is submitted for the degree of Doctor of Philosophy

Abstract

My abstract ...

Declaration

This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except where specified in the text. This dissertation is not substantially the same as any that I have submitted for a degree or diploma or other qualification at any other university. This dissertation does not exceed the prescribed limit of 60 000 words.

Felix Hill
April 2016

Acknowledgements

My acknowledgements ...

Contents

Chapter 1

Introduction

Some introduction text

Chapter 2

Neural Language Models

Some NLM text

Chapter 3

Distributed Representations of Words

Some word embedding text

3.1 Learning Distributed Word Representations from Text

A brief history of how this was done...Bengio - Collobert - Turian - Mikolov

Contrast with 'predicting' models: LSA, Sahlgren, Baroni

Levy two results.

3.2 Modelling Word Acquisition with Multi-Modal Data and Neural Language Models

3.2.1 Introduction

Multi-modal models that learn semantic representations from both language and information about the perceptible properties of concepts were originally motivated by parallels with human word learning [6] and evidence that many concepts are grounded in perception [14]. The perceptual information in such models is generally mined directly from images [53, 28] or from data collected in psychological studies [156, 147].

By exploiting the additional information encoded in perceptual input, multi-modal models can outperform language-only models on a range of semantic NLP tasks, including modelling similarity [29] and free association [156], predicting compositionality [147] and concept categorization [154]. However, to date, this superiority has only been established when evaluating on concrete words such as *cat* or *dog*, rather than abstract concepts, such as *curiosity* or *loyalty*. Indeed, differences between abstract and concrete processing and representation [133, 78] suggest that conclusions about concrete concept learning may not necessarily hold in the general case. In this paper, we therefore focus on multi-modal models for learning both abstract and concrete concepts.

Although concrete concepts might seem more basic or fundamental, the vast majority of open-class, meaning-bearing words in everyday language are in fact abstract. 72% of the noun or verb tokens in the British National Corpus [99] are rated by human judges[1] as more abstract than the noun *war*, for instance, a concept many would already consider to be quite abstract. Moreover, abstract concepts by definition encode higher-level (more general) principles than concrete concepts, which typically reside naturally in a single semantic category or domain [43]. It is therefore likely that abstract representations may prove highly applicable for multi-task, multi-domain or transfer learning models, which aim to acquire 'general-purpose' conceptual knowledge without reference to a specific objective or task [40, 117].

Motivated by these observations, we introduce an architecture for learning both abstract and concrete representations that generalizes the skipgram model of [118] from corpus-based to multi-modal learning. The extended model is designed to reflect aspects of human word learning, in that it introduces more perceptual information about commonly-occurring concrete concepts and less information about rarer concepts.

We train our model on running-text language and two sources of perceptual descriptors for concrete nouns: the ESPGame dataset of annotated images [169] and the CSLB set of concept property norms [46]. We find that our model *combines* information from the different modalities more effectively than previous methods, resulting in an improved ability to model the USF free association gold standard [130] for concrete nouns. In addition, the architecture *propagates* the extra-linguistic input for concrete nouns to improve representations of abstract concepts more effectively than alternative methods. While this propagation can effectively extend the advantage of the multi-modal approach to many more concepts than simple concrete nouns, we observe that the benefit of adding perceptual input appears to decrease as target concepts become more abstract. Indeed, for the most abstract concepts of all, language-only models still

[1] Contributors to the USF dataset [130]

provide the most effective learning mechanism.

Finally, we investigate the optimum quantity and type of perceptual input for such models. Between the most concrete concepts, which can be effectively represented directly in the perceptual modality, and the most abstract concepts, which cannot, we identify a set of concepts that cannot be represented effectively directly in the perceptual modality, but still benefit from perceptual input propagated in the model via concrete concepts.

The motivation in designing our model and experiments is both practical and theoretical. Taken together, the empirical observations we present are potentially important for optimizing the learning of representations of concrete and abstract concepts in multi-modal models. In addition, they offer a degree of insight into the poorly understood issue of how abstract concepts may be encoded in human memory.

3.2.2 Model Design

Before describing how our multi-modal architecture encodes and integrates perceptual information, we first describe the underlying corpus-based representation learning model.

Language-only Model Our multi-modal architecture builds on the continuous log-linear skip-gram language model proposed by [118]. This model learns lexical representations in a similar way to neural-probabilistic language models (NPLM) but without a non-linear hidden layer, a simplification that facilitates the efficient learning of large vocabularies of dense representations, generally referred to as *embeddings* [163]. Embeddings learned by the model achieve state-of-the-art performance on several evaluations including sentence completion and analogy modelling [118].

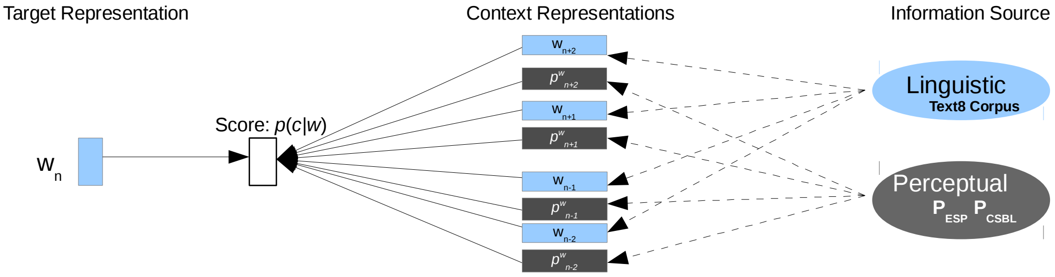


Figure 3.1: Our multi-modal model architecture. Light boxes are elements of the original [118] model. For target words w_n in the domain of $\mathbf{P}$, the model updates based on corpus context words w_{n+i} then on words p^w_{n+i} in perceptual psuedo-sentences. Otherwise, updates are based solely on the w_{n+i}.

For each word type w in the vocabulary V, the model learns both a 'target-embedding' $r_w \in \mathbb{R}^d$ and a 'context-embedding' $\hat{r}_w \in \mathbb{R}^d$ such that, given a target word, its ability to predict nearby context words is maximized. The probability of seeing context word c given target w is defined as:

$$p(c|w) = \frac{\mathrm{e}^{\hat{r}_c \cdot r_w}}{\sum_{v \in V} \mathrm{e}^{\hat{r}_v \cdot r_w}}$$

The model learns from a set of target-word, context-word pairs, extracted from a corpus of sentences as follows. In a given sentence S (of length N), for each position $n \leq N$, each word w_n is treated in turn as a target word. An integer $t(n)$ is then sampled from a uniform distribution on $\{1, \dots k\}$, where $k > 0$ is a predefined maximum context-window parameter. The pair tokens $\{(w_n, w_{n+j}) : -t(n) \leq j \leq t(n), w_i \in S\}$ are then appended to the training data. Thus, target/context training pairs are such that (i) only words within a k-window of the target are selected as context words for that target, and (ii) words closer to the target are more likely to be selected than those further away.

The training objective is then to maximize the log probability T across of all such examples from S, and then across all sentences in the corpus:

$$T = \frac{1}{N} \sum_{n=1}^{N} \sum_{-t(n) \leq j \leq t(n), j \neq 0} log(p(w_{n+j}|w_n))$$

The model free parameters (target-embeddings and context-embeddings of dimension d for each word in the corpus with frequency above a certain threshold f) are updated according to stochastic gradient descent and backpropation, with learning rate controlled by Adagrad [49]. For efficiency, the output layer is encoded as a hierarchical softmax function based on a binary Huffman tree [127].

As with other distributional architectures, the model captures conceptual semantics by exploting the fact that words appearing in similar linguistic contexts are likely to have similar meanings. Informally, the model adjusts its embeddings to increase the 'probability' of seeing the language in the training corpus. Since this probability increases with the $p(c|w)$, and the $p(c|w)$ increase with the dot product $\hat{r}_v \cdot r_c$, the updates have the effect of moving each target-embedding incrementally 'closer' to the context-embeddings of its collocates. In the target-embedding space, this results in embeddings of concept words that regularly occur in similar contexts moving closer together.

Multi-modal Extension We extend the [118] architecture via a simple means of introducing perceptual information that aligns with human language learning. Based on the assumption that frequency in domain-general linguistic corpora correlates with the likelihood of 'experiencing' a concept in the world [32, 34], perceptual information is introduced to the model whenever designated concrete concepts are encountered in the running-text linguistic input. This has the effect of introducing more perceptual input for commonly experienced concrete concepts and less input for rarer concrete concepts.

To implement this process, perceptual information is extracted from external sources and encoded in an associative array $\mathbf{P}$, which maps (typically concrete) words w to bags of perceptual features $\mathbf{b}(w)$. The construction of this array depends on the perceptual information source; the process for our chosen sources is detailed in Section 3.2.3.

Training our model begins as before on running-text. When a sentence S_m containing a word w in the domain of $\mathbf{P}$ is encountered, the model completes training on S_m and begins learning from a perceptual pseudo-sentence $\hat{S}(w)$. $\hat{S_m}(w)$ is constructed by randomly sampling features from $\mathbf{b}(w)$ to occupy positions before and instances of w, so that $\hat{S_m}(w)$ is the same length as S_m (see Figure 3.2). Once training on $\hat{S_m}(w)$ is completed, the model reverts to the next 'real' (linguistic) sentence S_{m+1}, and the process continues. Thus, when a concrete concept is encountered in the corpus, its embedding is first updated based on language (moved incrementally closer to concepts appearing in similar linguistic contexts), and then on perception (moved incrementally closer to concepts with the same or similar perceptual features).

For greater flexibility, we introduce a parameter α reflecting the raw quantity of perceptual information relative to linguistic input. When $\alpha = 2$, two pseudo-sentences are generated and inserted for every corpus occurrence of a token from the domain of $\mathbf{P}$. For non-integral α, the number of sentences inserted is $\lfloor \alpha \rfloor$, and a further sentence is added with probability $\alpha - \lfloor \alpha \rfloor$.

In all experiments reported in the following sections we set the window size parameter $k = 5$ and the minimum frequency parameter $f = 3$, which guarantees that the model learns embeddings for all concepts in our evaluation sets. While the model learns both target and context-embeddings for each word in the vocabulary, we conduct our experiments with the target embeddings only. We set the dimension parameter $d = 300$ as this produces high quality embeddings in the language-only case [118].

$\hat{S}(crocodile)$ = **Crocodile** legs **crocodile** teeth **crocodile** teeth **crocodile** scales **crocodile** green **crocodile**.

$\hat{S}(screwdriver)$ = **Screwdriver** handle **screwdriver** flat **screwdriver** long **screwdriver** handle **screwdriver** head.

Figure 3.2: Example pseudo-sentences generated by our model.

3.2.3 Information Sources

We construct the associative array of perceptual information $\mathbf{P}$ from two sources typical of those typically used for multi-modal semantic models.

ESPGame Dataset The ESP-Game dataset (ESP) [169] consists of 100,000 images, each annotated with a list of lexical concepts that appear in that image.

For any concept w identified in an ESP image, we construct a corresponding bag of features $\mathbf{b}(w)$. For each ESP image I that contains w, we append the other concept tokens identified in I to $\mathbf{b}(w)$. Thus, the more frequently a concept co-occurs with w in images, the more its corresponding lexical token occurs in $\mathbf{b}(w)$. The array $\mathbf{P_{ESP}}$ in this case then consists of the $(w, \mathbf{b}(w))$ pairs.

CSLB Property Norms The Centre for Speech, Language and the Brain norms (CSLB) [46] is a recently-released dataset containing semantic properties for 638 concrete concepts produced by human annotators. The CSLB dataset was compiled in the same way as the [114] property norms used widely in multi-modal models [156, 147]; we use CSLB because it contains more concepts. For each concept, the proportion of the 30 annotators that produced a given feature can also be employed as a measure of the strength of that feature.

When encoding the CSLB data in $\mathbf{P}$, we first map properties to lexical forms (e.g. *is_green* becomes *green*). By directly identifying perceptual features and linguistic forms in this way, we treat features observed in the perceptual data as (sub)concepts to be acquired via the same multi-modal input streams and stored in the same domain-general memory as the evaluation concepts. This design decision in fact corresponds to a view of cognition that is sometimes disputed [57]. In future studies we hope to compare the present approach to architectures with domain-specific conceptual memories.

For each concept w in CSLB, we then construct a feature bag $\mathbf{b}(w)$ by append-

ESPGame		**CSLB**	
Image 1	Image 2	Crocodile	Screwdriver
red	wreck	has 4 legs (7)	has handle (28)
chihuaua	cyan	has tail (18)	has head (5)
eyes	man	has jaw (7)	is long (9)
little	crash	has scales (8)	is plastic (18)
ear	accident	has teeth (20)	is metal (28)
nose	street	is green (10)	
small		is large (10)	

Table 3.1: Concepts identified in images in the ESP Game (left) and features produced for concepts by human annotators in the CSLB dataset (with feature strength, max=30).

ing lexical forms to $\mathbf{b}(w)$ such that the count of each feature word is equal to the strength of that feature for w. Thus, when features are sampled from $\mathbf{b}(w)$ to create pseudo-sentences (as detailed previously) the probability of a feature word occuring in a sentence reflects feature strength. The array $\mathbf{P_{CSLB}}$ then consists of all $(w, \mathbf{b}(w))$ pairs.

Linguistic Input The linguistic input to all models is the 400m word Text8 Corpus[2] of Wikipedia text, split into sentences and with punctuation removed.

3.2.4 Evaluation

We evaluate the quality of representations by how well they reflect *free association* scores, an empirical measure of cognitive conceptual proximity. The University of South Florida Norms (USF) [130] contain free association scores for over 40,000 concept pairs, and have been widely used in NLP to evaluate semantic representations [6, 53, 156, 147]. Each concept that we extract from the USF database has also been rated for conceptual concreteness on a Likert scale of 1-7 by at least 10 human annotators. Following previous studies [83, 156], we measure the (Spearman ρ) correlation between association scores and the cosine similarity of vector representations.

We create separate abstract and concrete concept lists by ranking the USF concepts according to concreteness and sampling at random from the first and fourth quartiles. We also introduce a complementary noun/verb dichotomy,[3] on the intu-

[2] From http://mattmahoney.net/dc/textdata.html

[3] Based on the majority POS-tag of words in the lemmatized British National Corpus [99]

Concept 1	**Concept 2**	**Assoc.**
abdomen (6.83)	stomach (6.04)	0.566
throw (4.05)	ball (6.08)	0.234
hope (1.18)	glory (3.53)	0.192
egg (5.79)	milk (6.66)	0.012

Table 3.2: Example concept pairs (with mean concreteness rating) and free-association scores from the USF dataset.

Concept Type	**List**	**Pairs**	**Examples**
concrete nouns	541	1418	*yacht, cup*
abstract nouns	100	295	*fear, respect*
all nouns	666	1815	*fear, cup*
concrete verbs	50	66	*kiss, launch*
abstract verbs	50	127	*differ, obey*
all verbs	100	221	*kiss, obey*

Table 3.3: Details the subsets of USF data used in our evaluations, downloadable from our website.

ition that information propagation may occur differently from noun to noun or from noun to verb (because of their distinct structural relationships in sentences). The abstract/concrete and noun/verb dichotomies yield four distinct concept lists. For consistency, the concrete noun list is filtered so that all concrete noun concepts w have perceptual representations **b**(w) in both $\mathbf{P_{ESP}}$ and $\mathbf{P_{CSLB}}$. For each of the four resulting concept lists C (concrete/abstract, noun/verb), a corresponding set of evaluation pairs $\{(w_1, w_2) \in USF : w_1, w_2 \in C\}$ is extracted (see Table 3 for details).

3.2.5 Results and Discussion

Our experiments were designed to answer four questions, outlined in the following subsections: (1) Which model architectures perform best at *combining* information pertinent to multiple modalities when such information exists explicitly (as common for concrete concepts)? (2) Which model architectures best propagate perceptual information to concepts for which it does not exist explicitly (as is common for abstract concepts)? (3) Is it preferable to include all of the perceptual input that can be obtained from a given source, or to filter this input stream in some way? (4) How much perceptual vs. linguistic input is optimal for learning various concept types?

3.2.6 Combining information sources

To evaluate our approach as a method of information combination we compared its performance on the concrete noun evaluation set against alternative methods. When implementing the alternatives, we first encoded the perceptual input directly into sparse feature vectors, with coordinates for each of the 2726 features in CSLB and for each of the 100,000 images in ESP.

The first alternative is simple concatenation of these perceptual vectors with linguistic vectors embeddings learned by the [118] model on the Text8 Corpus. In the second alternative, proposed for multi-modal models by [156], *canonical correlation analysis* (CCA) [68] was applied to the vectors of both modalities. This yields reduced-dimensionality representations that preserve underlying inter-modal correlations, which are then concatenated. The final alternative, proposed by [29] involves applying Singular Value Decomposition (SVD) to the matrix of concatenated multimodal representations, yielding smoothed representations.[4]

We compare these alternatives to our proposed model with $\alpha = 1$. In The CSLB and ESP models, all training pseudo-sentences are generated from the arrays $\mathbf{P_{CSLB}}$ and $\mathbf{P_{ESP}}$ respectively. In the models classed as *CSLB&ESP*, a random choice between $\mathbf{P_{CSLB}}$ and $\mathbf{P_{ESP}}$ is made every time perceptual input is included (so that the overall quantity of perceptual information is the same).

As shown in Figure 3.3 (left side), the embeddings learned by our model achieve a higher correlation with the USF data than simple concatenation, CCA and SVD regardless of perceptual input source. With the optimal perceptual source (ESP only), for instance, the correlation is 11% higher that the next best alternative method, CCA.

One possible factor behind this improvement is that, in our model, the learned representations fully integrate the two modalities, whereas for both CCA and the concatenation method each representation feature (whether of reduced dimension or not) corresponds to a particular modality. This deeper integration may help our architecture to overcome the challenges inherent in information combination such as inter-modality differences in information content and representation sparsity.

[4] CCA was implemented using the *CCA* package in R. SVD was implemented using the Python *sparsesvd* package, with truncation factor $k = 1024$ as per [29].

3.2.7 Propagating input to abstract concepts

To test the process of information propagation in our model, we evaluated the learned embeddings of more abstract concepts. We compared our approach with two recently-proposed alternative methods for inferring perceptual features when explicit perceptual information is unavailable.

Johns and Jones In the method of [88], pseudo-perceptual representations for target concepts without a perceptual representations (uni-modal concepts) are inferred as a weighted average of the perceptual representations of concepts that do have such a representation (bi-modal concepts).

In the first step of their two-step method, for each uni-modal concept $\mathbf{k}$, a quasi-perceptual representation is computed as an average of the perceptual representations of bi-modal concepts, weighted by the proximity between each of these concepts and $\mathbf{k}$

$$\mathbf{k}^p = \sum_{\mathbf{c} \in \bar{C}} S(\mathbf{k}^l, \mathbf{c}^l)^\lambda \cdot \mathbf{c}^p$$

where $\bar{C}$ is the set of bi-modal concepts, $\mathbf{c}^p$ and $\mathbf{k}^p$ are the perceptual representations for $\mathbf{c}$ and $\mathbf{k}$ respectively, and $\mathbf{c}^l$ and $\mathbf{k}^l$ the linguistic representations. The exponent parameter λ reflects the learning rate.

In step two, the initial quasi-perceptual representations are inferred for a second time, but with the weighted average calculated over the perceptual or initial quasi-perceptual representations of all other words, not just those that were orignally bi-modal. As with [88], we set the learning rate parameter λ to be 3 in the first step and 13 in the second.

Ridge Regression An alternative, proposed for the present purpose by [ref. withdrawn for review], uses *ridge regression* [128]. Ridge regression is a variant of least squares regression in which a regularization term is added to the training objective to favor solutions with certain properties.

For bi-modal concepts of dimension n_p, we apply ridge regression to learn n_p linear functions $f_i : \mathbb{R}^{n_l} \rightarrow \mathbb{R}$ that map the linguistic representations (of dimension n_l) to a particular perceptual feature i. These functions are then applied together to map the linguistic representations of uni-modal concepts to full quasi-perceptual representations.

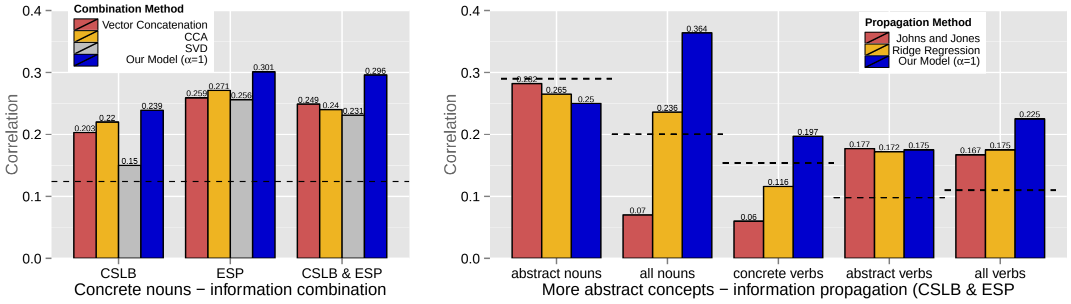


Figure 3.3: The proposed approach compared with other methods of information combination (left) and propagation. Dashed lines indicate language-only model baseline.

Following [ref. withdrawn for review], we take the Euclidian l_2 norm of the inferred parameter vector as the regularization term. This ensures that the regression favors lower coefficients and a smoother solution function, which should provide better generalization performance than simple linear regression. The objective for learning the f_i is then to minimize

$$||\mathbf{a}X - Y_i||_2^2 + ||\mathbf{a}||_2^2$$

where $\mathbf{a}$ is the vector of regression coefficients, X is a matrix of linguistic representations and Y_i a vector of the perceptual feature i for the set of bi-modal concepts.

Comparisons We applied the Johns and Jones method and ridge regression starting from linguistic embeddings acquired by the [118] model on the Text8 Corpus, and concatenated the resulting pseudo-perceptual and linguistic representations. As with the implementation of our model, the perceptual input for these alternative models was limited to concrete nouns (i.e. concrete nouns were the only bi-modal concepts in the models).

Figure 3.3 (right side) illustrates the propagation performance of the three models. While the correlations overall may seem somewhat low, this is a consequence of the difficulty of modeling the USF data. In fact, the performance of both the language-only model and our multi-modal extension across the concept types, ranging from .18–.36, is equal to or higher than equivalent models evaluated on the same data previously [53, 156, 155].

For learning representations of concrete verbs, our approach achieves a 69% increase in performance over the next best alternative. The performance of the model on abstract verbs is marginally inferior to Johns and Jones' method. Nevertheless, the clear advantage for concrete verbs makes our model the best choice for learning rep-

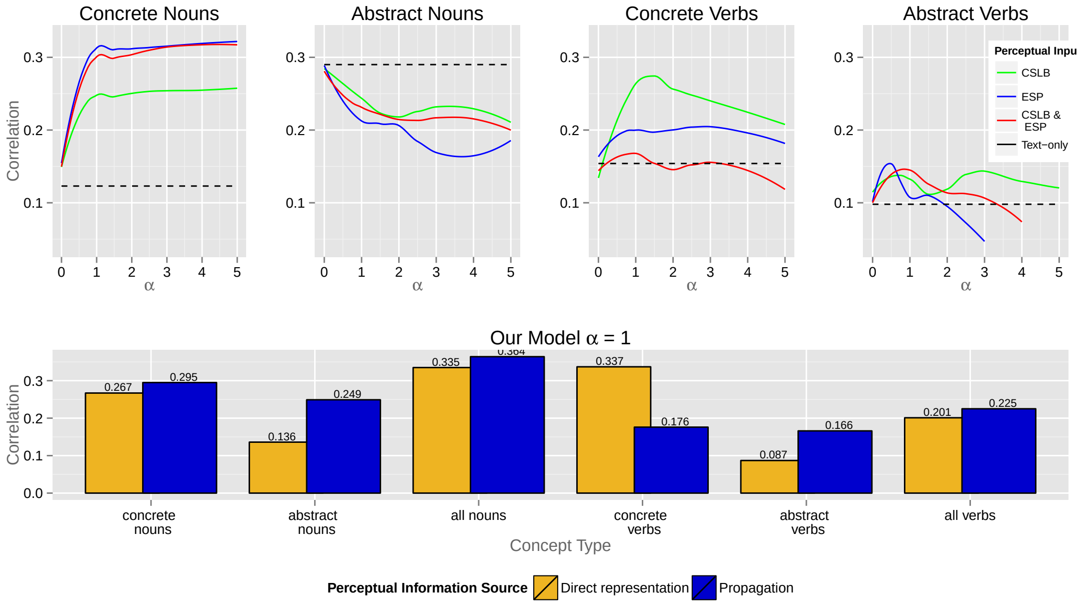


Figure 3.4: Top: Comparing the strategy of directly representing abstract concepts from perceptual information where available (yellow bars) vs. propagating via concrete concepts. Bottom: The effect of increasing α on correlation with USF pairs (Spearman ρ) for each concept type. Horizontal dashed lines indicate language-only model baseline.

resentations of verbs in general, as shown by performance on the set *all verbs*, which also includes mixed abstract-concrete pairs.

Our model is also marginally inferior to alternative approaches in learning representations of abstract nouns. However, in this case, no method improves on the linguistic-only baseline. It is possible that perceptual information is simply so removed from the core semantics of these concepts that they are best acquired via the linguistic medium alone, regardless of learning mechanism. The moderately inferior performance of our method in such cases is likely caused by its greater inherent intermodal dependence compared with methods that simply concatenate uni-modal representations. When the perceptual signal is of low quality, this greater inter-modal dependence allows the linguistic signal to be obscured. The trade-off, however, is the higher quality joint representations when the perceptual signal is of higher-quality, exemplified by the fact that our proposed approach outperforms alternatives on the set *all nouns*, which includes the more concrete nouns.

3.2.8 Direct representation vs. propagation

Although property norm datasets such as the CSLB data typically consist of perceptual feature information for concrete nouns only, image-based datasets such as ESP do contain information on more abstract concepts, which was omitted from the previous experiments. Indeed, image banks such as Google Images contain millions of photographs portraying quite abstract concepts, such as *love* or *war*. On the other hand, encodings or descriptions of abstract concepts are generally more subjective and less reliable than those of concrete concepts [176]. We therefore investigated whether or not it is preferable to include this additional information as model input or to restrict perceptual input to concrete nouns as previously.

Of our evaluation sets, it was possible to construct from ESP (and add to $\mathbf{P_{ESP}}$) representations for all of the concrete verbs, and for approximately half of the abstract verbs and abstract nouns. Figure 3.4 (top), shows the performance of a our model trained on all available perceptual input versus the model in which the perceptual input was restricted to concrete nouns.

The results reflect a clear manifestation of the abstract/concrete distinction. Concrete verbs behave similarly to concrete nouns, in that they can be effectively represented directly from perceptual information sources. The information encoded in these representations is beneficial to the model and increases performance. In contrast, constructing 'perceptual' representations of abstract verbs and abstract nouns directly from perceptual information sources is clearly counter-productive (to the extent that performance also degrades on the combined sets *all nouns* and *all verbs*). It appears in these cases that the perceptual input acts to obscure or contradict the otherwise useful signal inferred from the corpus.

As shown in the previous section, the inclusion of any form of perceptual input inhibits the learning of abstract nouns. However, this is not the case for abstract verbs. Our model learns higher quality representations of abstract verbs when perceptual input is restricted to concrete nouns than when no perceptual input is included whatsoever *and* when perceptual input is included for both concrete nouns and abstract verbs. This supports the idea of a gradual scale of concreteness: the most concrete concepts can be effectively represented directly in the perceptual modality; somewhat more abstract concepts cannot be represented directly in the perceptual modality, but have representations that are improved by propagating perceptual input from concrete concepts via language; and the most abstract concepts are best acquired via language alone.

3.2.9 Source and quantity of perceptual input

For different concept types, we tested the effect of varying the proportion of perceptual to linguistic input (the parameter α). Perceptual input was restricted to concrete nouns as in Sections 3.1-3.2.

As shown in Figure 3.4, performance on concrete nouns improves (albeit to a decreasing degree) as α increases. When learning concrete noun representations, linguistic input is apparently redundant if perceptual input is of sufficient quality and quantity. For the other concept types, in each case there is an optimal value for α in the range .5–2, above which perceptual input obscures the linguistic signal and performance degrades. The proximity of these optima to 1 suggests that for optimal learning, when a concrete concept is experienced approximately equal weight should be given to available perceptual and linguistic information.

3.2.10 Conclusions

Motivated by the notable prevalence of abstract concepts in everyday language, and their likely importance to flexible, general-purpose representation learning, we have investigated how abstract and concrete representations can be acquired by multi-modal models. In doing so, we presented a simple and easy-to-implement architecture for acquiring semantic representations of both types of concept from linguistic and perceptual input.

While neuro-probabilistic models have been applied to the problem of multi-modal representation learning previously [158, 178] our model and experiments develop this work in several important ways. First, we address the problem of learning abstract concepts. By isolating concepts of different concreteness and part-of-speech in our evaluation sets, and separating the processes of information combination and propagation, we demonstrate that the multi-modal approach is indeed effective for some, but perhaps not all, abstract concepts. In addition, our model introduces a clear parallel with human language learning. Perceptual input is introduced precisely when concrete concepts are ‘experienced’ by the model in the corpus text, much like a language learner experiencing concrete entities via sensory perception.

Taken together, our findings indicate the utility of distinguishing three concept types when learning representations in the multi-modal setting.

Type I Concepts that can be effectively represented directly in the perceptual modality. For such concepts, generally concrete nouns or concrete verbs, our proposed approach provides a simple means of combining perceptual and linguistic input. The resulting multi-modal representations are of higher quality than those learned via other approaches, resulting in a performance improvement of over 10% in modelling free association.

Type II Concepts, including abstract verbs, that cannot be effectively represented directly in the perceptual modality, but whose representations can be improved by joint learning from linguistic input and perceptual information about related concepts. Our model can effectively propagate perceptual input (exploiting the relations inferred from the linguistic input) from Type I concepts to enhance the representations of Type II concepts above the language-only baseline. Because of the frequency of abstract concepts, such propagation extends the benefit of the multi-modal approach to a far wider range of language than models based solely in the concrete domain.

Type III Concepts, such as abstract nouns, which are more effectively learned via language-only models than multi-modal models. Neither the model we introduce here nor other proposed propagation methods achieve an improvement in representation quality for these concepts over the language-only baseline. Of course, it is an empirical question whether a multi-modal approach could ever enhance the representation learning of these concepts, one with potential implications for cognitive theories of grounding (a topic of much debate in psychology [64, 12]).

Additionally, we investigated the optimum type and quantity of perceptual input for learning concepts of different types. We showed that too much perceptual input can result in degraded representations. For concepts of type I and II, the optimal quantity resulted from setting $\alpha = 1$; i.e. whenever a concrete concept was encountered, the model learned from an equal number of language-based and perception-based examples. While we make no formal claims here, such observations may ultimately provide insight into human language learning and semantic memory.

In future we will address the question of whether Type III concepts can ever be enhanced via multi-modal learning, and investigate multi-modal models that optimally learn concepts of each type. This may involve filtering the perceptual input stream for concepts according to concreteness, and possibly more elaborate model architectures that facilitate distinct representational frameworks for abstract and concrete concepts.

3.3 Multi-modal fusion based on image dispersion

3.4 Improving the Evaluation of Word Representations

There is very little similar about coffee and cups. *Coffee* refers to a plant, which is a living organism or a hot brown (liquid) drink. In contrast, a *cup* is a man-made solid of broadly well-defined shape and size with a specific function relating to the consumption of liquids. Perhaps the only clear trait these concepts have in common is that they are concrete entities. Nevertheless, in what is currently the most popular evaluation gold standard for semantic similarity, WordSim(WS)-353 [55], *coffee* and *cup* are rated as more 'similar' than pairs such as *car* and *train*, which share numerous common properties (function, material, dynamic behaviour, wheels, windows etc.). Such anomalies also exist in other gold standards such as the MEN dataset [28]. As a consequence, these evaluations effectively penalise models for learning the evident truth that *coffee* and *cup* are dissimilar.

Although clearly different, *coffee* and *cups* are very much related. The psychological literature refers to the conceptual relationship between these concepts as *association*, although it has been given a range of names including *relatedness* [31, 1], *topical similarity* [71] and *domain similarity* [164]. Association contrasts with *similarity*, the relation connecting *cup* and *mug* [166]. At its strongest, the similarity relation is exemplified by pairs of *synonyms*; words with identical referents.

Computational models that effectively capture similarity as distinct from association have numerous applications. Such models are used for the automatic generation of dictionaries, thesauri, ontologies and language correction tools [38, 22, 106]. Machine translation systems, which aim to define mappings between fragments of different languages whose meaning is similar, but not necessarily associated, are another established application [72, 113]. Moreover, since, as we establish, similarity is a cognitively complex operation that can require rich, structured conceptual knowledge to compute accurately, similarity estimation constitutes an effective proxy evaluation for general-purpose representation-learning models whose ultimate application is variable or unknown [40, 11].

As we show in Section 2, the predominant gold standards for semantic evaluation in NLP do not measure the ability of models to reflect similarity. In particular, in both WS-353 and MEN, pairs of words with associated meaning, such as *coffee* and *cup* (rating = 6.8) *telephone* and *communication* (7.5) or *movie* and *theater* (7.7), receive a high rating regardless of whether or not their constituents are similar. Thus, the utility of such resources to the development and application of similarity models is limited, a problem exacerbated by the fact that many researchers appear unaware of what their evaluation resources actually measure.[5]

While certain smaller gold standards, those of [150] (RG) and [1] (WS-Sim), do focus clearly on similarity, these resources suffer from other important limitations. For instance, as we show, and as is also the case for WS-353 and MEN, state-of-the-art models have reached the average performance of a human annotator on these evaluations. It is common practice in NLP to define the upper limit for automated performance on an evaluation as the average human performance or inter-annotator agreement [180, 44, 146]. Based on this established principle and the current evaluations, it would therefore be reasonable to conclude that the problem of representation learning, at least for similarity modelling, is approaching resolution. However, circumstantial evidence suggests that distributional models are far from perfect. For instance, we are some way from automatically-generated dictionaries, thesauri or ontologies that can be used with the same confidence as their manually-created equivalents.

Motivated by these observations, in Section 3 we present *SimLex-999*, a gold standard resource for evaluating the ability of models to reflect similarity. SimLex-999 was produced by 500 paid native English speakers, recruited via Amazon Mechanical Turk,[6] who were asked to rate the similarity, as opposed to association, of concepts via a simple visual interface. The choice of evaluation pairs in SimLex-999 was motivated by empirical evidence that humans represent concepts of distinct part-of-speech (POS) [58] and conceptual concreteness [78] differently. While existing gold standards contain only concrete noun concepts (MEN) or cover only some of these distinctions via a random selection of items (WS-353, RG), SimLex-999 contains a principled selection of adjective, verb and noun concept pairs covering the full concreteness spectrum. This design enables more nuanced analyses of how computational models overcome

[5] For instance, [83] (pages 1,4,10) and [144] (page 4) refer to MEN and/or WS-353 as 'similarity datasets'. Others evaluate on both these association-based and genuine similarity-based gold standards with no reference to the fact that they measure different things [116, 105].

[6] `www.mturk.com/`

the distinct challenges of representing concepts of these types.

In Section 4 we present quantitative and qualitative analyses of the SimLex-999 ratings, which indicate that participants found it unproblematic to quantify consistently the similarity of the full range of concepts and to distinguish it from association. Unlike existing datasets, SimLex-999 therefore contains a significant number of pairs, such as [*movie*, *theater*], which are strongly associated but receive low similarity scores.

The second main contribution of this paper, presented in Section 5, is the evaluation of state-of-the-art distributional semantic models using SimLex-999. These include the well known neural language models (NLMs) of [83], [40] and [118], which we compare with traditional vector-space co-occurrence models (VSMs) [165] with and without dimensionality reduction (SVD) [97]. Our analyses demonstrate how SimLex-999 can be applied to uncover substantial differences in the ability of models to represent concepts of different types.

Despite these differences, the models we consider each share the characteristic of being better able to capture association than similarity. We show that the difficulty of estimating similarity is driven primarily by those strongly-associated pairs with a high (association) rating in gold standards such as WS-353 and MEN, but a low similarity rating in SimLex-999. As a result of including these challenging cases, together with a wider diversity of lexical concepts in general, current models achieve notably lower scores on SimLex-999 than on existing gold standard evaluations, and well below the SimLex-999 inter-human agreement ceiling.

Finally, we explore ways in which distributional models might improve on this performance in similarity modelling. To do so, we evaluate the models on the SimLex-999 subsets of adjectives, nouns and verbs, as well as on abstract and concrete subsets and subsets of more and less strongly associated pairs (Sections 5.2.2-5.2.4). As part of these analyses, we confirm the hypothesis [1, 100] that models learning from input informed by dependency parsing, rather than simple running-text input, yield improved similarity estimation and, specifically, clearer distinction between similarity and association. In contrast, we find no evidence for a related hypothesis [1, 91], that smaller context windows improve the ability of models to capture similarity. We do, however, observe clear differences in model performance on the distinct concept types included in SimLex-999. Taken together, these experiments demonstrate the benefit of the diversity of concepts included in SimLex-999; it would not have been possible to derive similar insights by evaluating on existing gold standards.

We conclude by discussing how observations such as these can guide future research into distributional semantic models. By facilitating better-defined evaluations and finer-grained analyses, we hope that SimLex-999 will ultimately contribute to the development of models that accurately reflect human intuitions of similarity for the full range of concepts in language.

3.5 Design Motivation

In this section, we motivate the design decisions made in developing SimLex-999. We begin (2.1) by examining the distinction between similarity and association. We then show that for a meaningful treatment of similarity it is also important to take a principled approach to both part-of-speech (POS) and conceptual concreteness (2.2). We finish by reviewing existing gold standards, and show that none enables a satisfactory evaluation of the capability of models to capture similarity (2.3).

3.5.1 Similarity and Association

The difference between association and similarity is exemplified by the concept pairs [*car, bike*] and [*car, petrol*]. *Car* is said to be (semantically) similar to *bike* and associated with (but not similar to) *petrol*. Intuitively, *car* and *bike* can be understood as similar because of their common physical features (e.g. wheels), their common function (transport), or because they fall within a clearly definable category (modes of transport). In contrast, *car* and *petrol* are associated because they frequently occur together in space and language, in this case as a result of a clear functional relationship [140, 115].

Association and similarity are neither mutually exclusive nor independent. *Bike* and *car*, for instance, are related to some degree by both relations. Since it is common in both the physical world and in language for distinct entities to interact, it is relatively easy to conceive of concept pairs, such as *car* and *petrol*, that are strongly associated but not similar. Identifying pairs of concepts for which the converse is true is comparatively more difficult. One exception is common concepts paired with low frequency synonyms, such as *camel* and *dromedary*. Since the essence of association is co-occurrence (linguistic or otherwise [115]), such pairs can seem, at least intuitively, to be similar but not strongly associated.

To explore the interaction between the two cognitive phenomena quantitatively, we exploited perhaps the only two existing large-scale means of quantifying similarity and association. To estimate similarity, we considered proximity in the WordNet taxonomy [52]. Specifically, we applied the measure of [179] (henceforth *WupSim*), which approximates similarity on a [0,1] scale reflecting the minimum distance between any two synsets of two given concepts in WordNet. WupSim has been shown to correlate well with human judgements on the similarity-focused RG dataset [179]. To estimate association, we extracted ratings directly from the University of South Florida Free Association Database (USF) [130]. These data were generated by presenting human subjects with one of 5000 cue concepts and asking them to write the *first word that comes into their head that is associated with or meaningfully related to that concept*. Each cue concept c was normed in this way by over 10 participants, resulting in a set of associates for each cue, and a total of over 72,000 (c, a) pairs. Moreover, for each such pair, the proportion of participants who produced associate a when presented with cue c can be used as a proxy for the strength of association between the two concepts.

By measuring WupSim between all pairs in the USF dataset, we observed, as expected, a high correlation between similarity and association strength across all USF pairs (Spearman $\rho = 0.65, p < 0.001$). However, in line with the intuitive ubiquity of pairs such as *car* and *petrol*, of the USF pairs (all of which are associated to a greater or lesser degree) over 10% had a WupSim score of less than 0.25. These include pairs of ontologically different entities with a clear functional relationship in the world [*refrigerator, food*], which may be of differing concreteness [*lung, disease*], pairs in which one concept is a small concrete part of a larger abstract category [*sheriff, police*], pairs in a relationship of modification or subcategorization [*gravy, boat*] and even those whose principal connection is phonetic [*wiggle, giggle*]. As we show in Section 2.2, these are precisely the sort of pairs that are not contained in existing evaluation gold standards. Table 1 lists the USF noun pairs with the lowest similarity scores overall, and also those with the largest additive discrepancy between association strength and similarity.

3.5.1.1 Association and similarity in NLP

As noted in the Introduction, the similarity/association distinction is not only of interest to researchers in psychology or linguistics. Models of similarity are particularly applicable to various NLP tasks, such as lexical resource building, semantic parsing and machine translation [72, 67, 113, 15]. Models of association, on the other hand, may be better suited to tasks such as word-sense disambiguation [129], and applica-

Concept 1	Concept 2	USF	WupSim
hatchet	*murder*	0.013	0.091
robbery	*jail*	0.020	0.100
lung	*disease*	0.014	0.105
burglar	*robbery*	0.020	0.105
sheriff	*police*	0.333	0.133
colonel	*army*	0.303	0.111
quart	*milk*	0.462	0.235
refrigerator	*food*	0.424	0.235

Table 3.4: Top: Concept pairs with the lowest WupSim scores in the USF dataset overall. Bottom: Pairs with the largest discrepancy in rank between association strength (high) and WupSim (low).

tions such as text classification [139] in which the target classes correspond to topical domains such as *agriculture* or *sport* [149].

Much recent research in *distributional semantics* does not distinguish between association and similarity in a principled way (see e.g. [83, 144, 108]).[7] One exception is [164], who constructs two distributional models with different features and parameter settings, explicitly designed to capture either similarity or association. Using the output of these two models as input to a logistic regression classifier, Turney predicts whether two concepts are associated, similar or both, with 61% accuracy. However, in the absence of a gold standard covering the full range of similarity ratings (rather than a list of pairs identified as being similar or not) Turney cannot confirm directly that the similarity-focused model does indeed effectively quantify similarity.

[1] explicitly examine the distinction between association and similarity in relation to distributional semantic models. Their study is based on the partition of WS-353 into a subset focused on similarity, which we refer to as *WS-Sim*, and a subset focused on association, which we term *WS-Rel*. More precisely, WS-Sim is the union of the pairs in WS-353 judged by three annotators to be similar and the set U of entirely unrelated pairs, and WS-Rel is the union of U and pairs judged to be associated but not similar. Agirre et al. confirm the importance of the association/similarity distinction by showing that certain models perform relatively well on WS-Rel while others perform comparatively better on WS-Sim. However, as shown in the following section, a model need not be an exemplary model of similarity in order to perform well on WS-Sim since an important class of concept pair (associated but not similar entities) is not represented

[7] Several papers that take a knowledge-based or symbolic approach to meaning do address the similarity/association issue [31].

in this dataset. Therefore the insights that can be drawn from the results of the [1] study are limited.

Several other authors touch on the similarity/association distinction in inspecting the output of distributional models [6, 91, 100]. While the strength of the conclusions that can be drawn from such qualitative analyses is clearly limited, there appear to be two broad areas of consensus concerning similarity and distributional models:

- Models that learn from input annotated for syntactic or dependency relations better reflect similarity, whereas approaches that learn from running-text or bag-of-words input better model association [1, 100].

- Models with larger context windows may learn representations that better capture association, whereas models with narrower windows better reflect similarity [1, 91].

3.5.2 Concepts, part-of-speech and concreteness

Empirical studies have shown that the performance of both humans and distributional models depends on the POS category of the concepts learned. [59] showed that children find verb concepts harder to learn than noun concepts, and [112] present evidence that different cognitive operations are employed when comparing two nouns or two verbs. [79] demonstrate differences in the ability of distributional models to acquire noun and verb semantics. Further, they show that these differences are greater for models that learn from both text and perceptual input (as with humans).

In addition to POS category, differences in human and computational concept learning and representation have been attributed to the effects of *concreteness*, the extent to which a concept has a directly perceptible physical referent. On the cognitive side, these 'concreteness effects' are well established, even if the causes are still debated [133, 78]. Concreteness has also been associated with differential performance in computational text-based [76] and multi-modal semantic models [92].

3.5.3 Existing gold standards and evaluation resources

For brevity, we do not exhaustively review all methods that have been employed to evaluate semantic models, but instead focus on the similarity or association-based gold standards that are most commonly-applied in recent work in NLP. In each case, we consider how well the dataset satisfies one of the three following criteria:

Representative The resource should cover the full range of concepts that occur in natural language. In particular, it should include cases representing the different ways in which humans represent or process concepts, and cases that are both challenging and straightforward for computational models.

Clearly-defined In order for a gold standard to be diagnostic of how well a model can be applied to downstream applications, a clear understanding is needed of what exactly the gold standard measures. In particular, it must clearly distinguish between dissociable semantic relations such as association and similarity.

Consistent and reliable Untrained native speakers must be able to quantify the target property consistently, without requiring lengthy or detailed instructions. This ensures that the data reflect a meaningful cognitive or semantic phenomenon, and also enables the dataset to be scaled up or transferred to other languages at minimal cost and effort.

We begin our review of existing evaluation with the gold standard most commonly-applied in current NLP research.

WordSim-353 WS-353 [55] is perhaps the most commonly-used evaluation gold standard for semantic models. Despite its name, and the fact that it is often referred to as a 'similarity gold standard',[8] in fact, the instructions given to annotators when producing WS-353 were ambiguous with respect to similarity and association. Subjects were asked to: "*Assign a numerical similarity score between 0 and 10 (0 = words totally unrelated, 10 = words VERY closely related) ... when estimating similarity of antonyms, consider them "similar" (i.e., belonging to the same domain or representing features of the same concept), not "dissimilar".*"

As we confirm analytically in Section 5.2, these instructions result in pairs being rated according to association rather than similarity.[9] WS-353 consequently suffers two important limitations as an evaluation of similarity (which also afflict other resources to a greater or lesser degree):

1. Many dissimilar word pairs receive a high rating.

[8] See e.g. [83, 8]

[9] This fact is also noted by the dataset authors. See `www.cs.technion.ac.il/~gabr/resources/data/wordsim353/`.

2. No associated but dissimilar concepts receive low ratings.

As noted in the Introduction, an arguably more serious third limitation of WS-353 is low inter-annotator agreement, and the fact that state-of-the-art models such as those of [40] and [83] reach, or even surpass, the inter-annotator agreement ceiling in estimating the WS-353 scores. [83] report a Spearman correlation of $\rho = 0.713$ between their model output and WS-353. This is ten percentage points higher than inter-annotator agreement ($\rho = 0.611$) when defined as the average pairwise correlation between two annotators, as is common in NLP work [132, 143, 154]. It could be argued that a different comparison is more appropriate: Since the model is compared to the gold-standard average across all annotators, we should compare a single annotator with the (almost) gold-standard average over all other annotators. Based on this metric the average performance of an annotator on WS-353 is $\rho = 0.756$, which is still only marginally better than the best automatic method.[10]

Thus, at least according to the established wisdom in NLP evaluation [180, 44, 146], the strength of the conclusions that can be inferred from improvements on WS-353 is limited. At the same time, however, state-of-the-art distributional models are clearly not perfect representation-learning or even similarity estimation engines, as evidenced by the fact they cannot yet be applied, for instance, to generate flawless lexical resources [4].

WS-Sim WS-Sim is the set of pairs in WS-353 identified by [1] as either containing similar or unrelated (neither similar nor associated) concepts. The ratings in WS-Sim are mapped directly from WS-353, so that all concept pairs in WS-Sim that receive a high rating are associated and all pairs that receive a low rating are unassociated. Consequently, any model that simply reflects association would score highly on WS-Sim, irrespective of how well it captures similarity.

Such a possibility could be excluded by requiring models to perform well on WS-Sim and poorly on WS-Rel, the subset of WS-353 identified by [1] as containing no pairs of similar concepts. However, while this would exclude models of pure association, it would not test the ability of models to quantify the similarity of the pairs in WS-Sim. Put another way, the WS-Sim/WS-Rel partition could in theory resolve limitation (1) of WS-353 but it would not resolve limitation (2): models are not tested on their ability to attribute low scores to associated but dissimilar pairs.

[10] Individual annotator responses for WS-353 were downloaded from www.cs.technion.ac.il/~gabr/resources/data/wordsim353.

In fact, there are more fundamental limitations of WS-Sim as a similarity-based evaluation resource. It does not, strictly-speaking, reflect similarity at all, since the ratings of its constituent pairs were assigned by the WS-353 annotators, who were asked to estimate association, not similarity. Moreover, it inherits the limitation of low inter-annotator agreement from WS-353. The average pairwise correlation between annotators on WS-Sim is $\rho = 0.667$, and the average correlation of a single annotator with the gold standard is only $\rho = 0.651$, both below the performance of automatic methods [1]. Finally, the small size of WS-Sim renders it poorly representative of the full range of concepts that semantic models may be required to learn.

Rubenstein & Goodenough Prior to WS-353, the smaller RG dataset, consisting of 65 pairs, was often used to evaluate semantic models. The 15 raters employed in the data collection were asked to rate the 'similarity of meaning' of each concept pair. Thus RG does appear to reflect similarity rather than association. However, while limitation (1) of WS-353 is therefore avoided, RG still suffers from limitation (2): By inspection, it is clear that the low similarity pairs in RG are not associated. A further limitation is that distributional models now achieve better performance on RG (correlations of up to Person $r = 0.86$ [70]) than the reported inter-annotator agreement of $r = 0.85$ [150]. Finally, the size of RG renders it an even less comprehensive evaluation than WS-Sim.

The MEN Test Collection A larger dataset, MEN [28], is used in a handful of recent studies [30, 21]. As with WS-353, both of the terms *similarity* and *relatedness* are used by the authors when describing MEN, although the annotators were expressly asked to rate pairs according to relatedness.[11]

The construction of MEN differed from RG and WS-353 in that each pair was only considered by one rater, who ranked it for relatedness relative to 50 other pairs in the dataset. An overall score out of 50 was then attributed to each pair corresponding to how many times it was ranked as more related than an alternative. However, because these rankings are based on relatedness, with respect to evaluating similarity MEN necessarily suffers from both of the limitations (1) and (2) that apply to WS-353. Further, there is a strong bias towards concrete concepts in MEN because the concepts were originally selected from those identified in an image-bank [28].

[11] http://clic.cimec.unitn.it/ elia.bruni/MEN.html

Synonym detection sets Multiple-choice synonym detection tasks, such as the TOEFL test questions [97], are an alternative means of evaluating distributional models. A question in the TOEFL task consists of a cue word and four possible answer words, only one of which is a true synonym. Models are scored on the number of true synonyms identified out of 80 questions. The questions were designed by linguists to evaluate synonymy, so, unlike the evaluations considered thus far, TOEFL-style tests effectively discriminate between similarity and association. However, since they require a zero-one classification of pairs as synonymous or not, they do not test how well models discern pairs of medium or low similarity. More generally, in opposition to the fuzzy, statistical approaches to meaning predominant in both cognitive psychology [66] and NLP [165], they do not require similarity to be measured on a continuous scale.

3.6 The SimLex-999 Dataset

Having considered the limitations of existing gold standards, in this section we describe the design of SimLex-999 in detail.

3.6.1 Choice of Concepts

Separating similarity from association To create a test of the ability of models to capture similarity as opposed to association, we started with the $\approx 72,000$ pairs of concepts in the USF dataset. As the output of a free-association experiment, each of these pairs is associated to a greater or lesser extent. Importantly, inspecting the pairs revealed that a good range of similarity values are represented. In particular, there were many examples of hypernym / hyponym pairs [*body, abdomen*] cohyponym pairs [*cat, dog*], synonyms or near synonyms [*deodorant, antiperspirant*] and antonym pairs [*good, evil*]. From this cohort, we excluded pairs containing a multiple-word item [*hot dog, mustard*], and pairs containing a capital letter [*Mexico, sun*]. We ultimately sampled 900 of the SimLex-999 pairs from the resulting cohort of pairs according to the stratification procedures outlined in the following sections.

To complement this cohort with entirely unassociated pairs, we paired up the concepts from the 900 associated pairs at random. From these random parings, we excluded those that coincidentally occurred elsewhere in USF (and therefore had a degree of association). From the remaining pairs, we accepted only those in which both concepts had been subject to the USF norming procedure, ensuring that these non-USF pairs were indeed unassociated rather than simply not normed. We sampled the remaining 99 SimLex-999 pairs from this resulting cohort of unassociated pairs.

POS category In light of the conceptual differences outlined in Section 2.2, SimLex-999 includes subsets of pairs from the three principle meaning-bearing POS categories, nouns, verbs and adjectives. To classify potential pairs according to POS, we counted the frequency with which the items in each pair occurred with the three possible tags in the POS-tagged British National Corpus [99]. To minimise POS ambiguity, which could lead to inconsistent rating, we excluded pairs containing a concept with lower than 75% tendency towards one particular POS. This yielded three sets of potential pairs : [A,A] pairs (of two concepts whose majority tag was Adjective), [N,N] pairs and [V,V] pairs.

Given the likelihood that different cognitive operations are employed in estimating the similarity between items of different POS-category (Section 2.2), concept pairs were presented to raters in batches defined according to POS. Unlike both WS-353 and MEN, pairs of concepts of mixed POS ([*white, rabbit*], [*run,marathon*]) were excluded. POS categories are generally considered to reflect very broad ontological classes [52]. We thus felt it would be very difficult, or even counter-intuitive, for annotators to quantify the similarity of mixed POS pairs according to our instructions.

Concreteness Although a clear majority of pairs in gold standards such as MEN and RG contain concrete items, perhaps surprisingly, the vast majority of adjective, noun and verb concepts in everyday language are in fact abstract [79, 92].[12] To facilitate the evaluation of models for both concrete and abstract concept meaning, and in light of the cognitive and computational modelling differences between abstract and concrete concepts noted in Section 2.2, we aimed to include both concept types in SimLex-999.

Unlike the POS distinction, concreteness is generally considered to be a gradual phenomenon. One benefit of sampling pairs for SimLex-999 from the USF dataset is that most items have been rated according to concreteness on a scale of 1-7 by at least 10 human subjects. As Figure 1 demonstrates, concreteness (as the average over these ratings) interacts with POS on these concepts: nouns are on average more concrete than verbs which are more concrete than adjectives. However, there is also

[12] According to the USF concreteness ratings, 72% of noun or verb types in the British National Corpus are more abstract than the concept *war*, a concept many would already consider quite abstract.

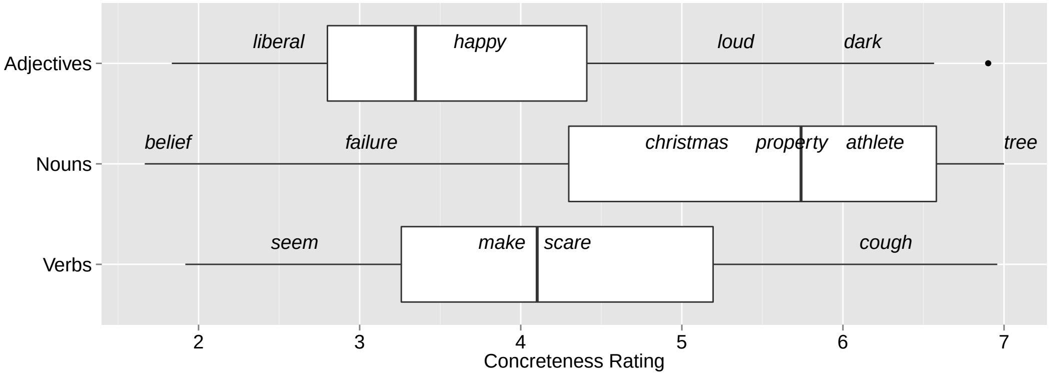


Figure 3.5: Boxplots showing the interaction between concreteness and POS for concepts in USF. The white boxes range from the first to third quartiles and the central vertical line indicates the median.

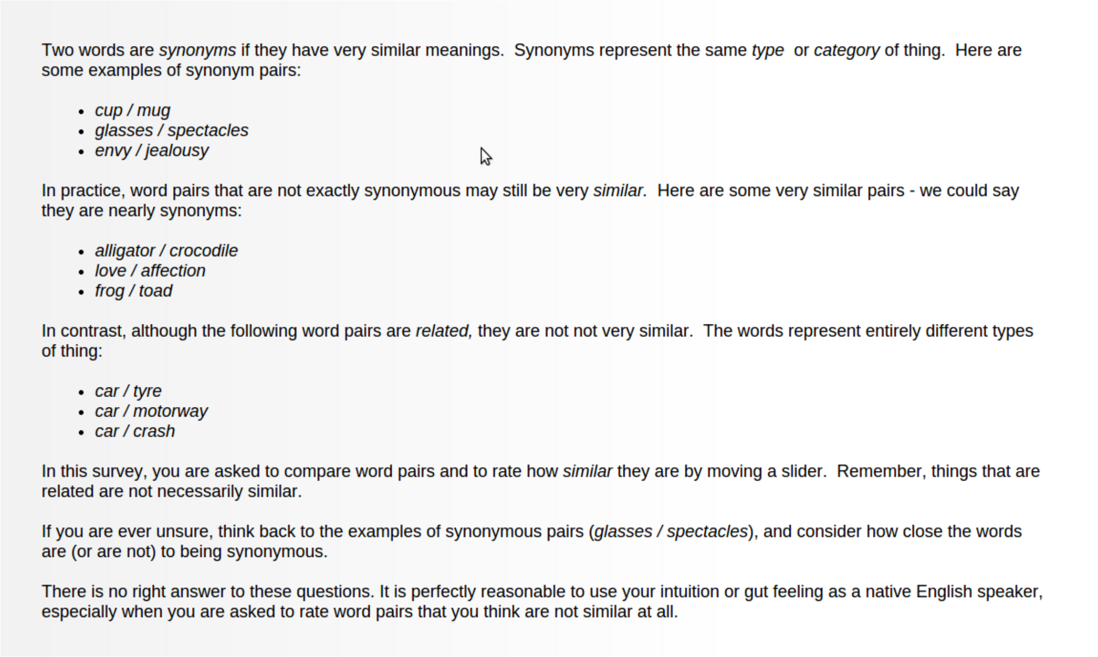
clear variation in concreteness within each POS category. We therefore aimed to select pairs for SimLex-999 that spanned the full abstract-concrete continuum within each POS category.

After excluding any pairs that contained an item with no concreteness rating, for each potential SimLex-999 pair we considered both the concreteness of the first item and the additive difference in concreteness between the two items. This enabled us to stratify our sampling equally across four classes: (C_1) concrete first item (rating > 4) with below-median concreteness difference, (C_2) concrete first item (rating> 4), second item of lower concreteness and the difference being greater than the median, (C_3) abstract first item (rating ≤ 4) with below-median concreteness difference, and (C_4) abstract first item (rating ≤ 4) with the second item of greater concreteness and the difference being greater than the median.

Final sampling From the associated (USF) cohort of potential pairs we selected 600 noun pairs, 200 verb pairs and 100 adjective pairs, and from the unassociated (non-USF) cohort, we sampled 66 nouns pairs, 22 verb pairs and 11 adjective pairs. In both cases, the sampling was stratified such that, in each POS subset, each of the four concreteness classes $C_1 - C_4$ was equally represented.

3.6.2 Question Design

The annotator instructions for SimLex-999 are shown in Figure 2. We did not attempt to formalise the notion of similarity, but rather introduce it via the well-understood idea of synonymy, and in contrast to association. Even if a formal characterisation of similarity existed, the evidence in Section 2 suggests that the instructions would need separate cases to cover different concept types, increasing the difficulty of the rating task. Therefore we preferred to appeal to intuition on similarity, and to verify post-hoc that subjects were able to interpret and apply the informal characterization consistently for each concept type.



Two words are *synonyms* if they have very similar meanings. Synonyms represent the same *type* or *category* of thing. Here are some examples of synonym pairs:

- *cup / mug*
- *glasses / spectacles*
- *envy / jealousy*

In practice, word pairs that are not exactly synonymous may still be very *similar*. Here are some very similar pairs - we could say they are nearly synonyms:

- *alligator / crocodile*
- *love / affection*
- *frog / toad*

In contrast, although the following word pairs are *related*, they are not not very similar. The words represent entirely different types of thing:

- *car / tyre*
- *car / motorway*
- *car / crash*

In this survey, you are asked to compare word pairs and to rate how *similar* they are by moving a slider. Remember, things that are related are not necessarily similar.

If you are ever unsure, think back to the examples of synonymous pairs (*glasses / spectacles*), and consider how close the words are (or are not) to being synonymous.

There is no right answer to these questions. It is perfectly reasonable to use your intuition or gut feeling as a native English speaker, especially when you are asked to rate word pairs that you think are not similar at all.

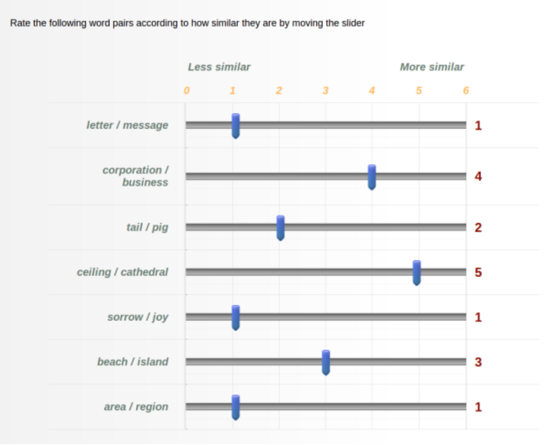
Figure 3.6: Instructions for SimLex-999 annotators.

Immediately following the instructions in Figure 2, participants were presented with two ’checkpoint’ questions, one with abstract examples and one with concrete examples. In each case the participant was required to identify the *most similar* pair from a set of three options, all of which were associated, but only one of which was clearly similar (e.g. [*bread, butter*] [*bread, toast*] [*stale, bread*]). After this, the participants began rating pairs in groups of 6 or 7 pairs by moving a slider, as shown in Figure 3.

This group size was chosen because the (relative) rating a set of pairs implicitly requires pairwise comparisons between all pairs in that set. Therefore, larger groups would have significantly increased the cognitive load on the annotators. Another advantage of grouping was the clear break (submitting a set of ratings and moving to the next page) between the tasks of rating adjective, noun and verb pairs. For better inter-group calibration, from the second group onwards the last pair of the previous group became the first pair of the present group, and participants were asked to re-assign the rating previously attributed to the first pair before rating the remaining new items.

3.6.3 Context-free rating

As with MEN, WS-353 and RG, SimLex-999 consists of pairs of concept words together with a numerical rating. Thus, unlike in the small evaluation constructed by [83], words are not rated in a phrasal or sentential context. Such meaning-in-context evaluations are motivated by a desire to disambiguate words that otherwise might be considered to have multiple senses.



Rate the following word pairs according to how similar they are by moving the slider

letter / message 1

corporation / business 4

tail / pig 2

ceiling / cathedral 5

sorrow / joy 1

beach / island 3

area / region 1

Figure 3.7: A group of noun pairs to be rated by moving the sliders. The rating slider was initially at position 0, and it was possible to attribute a rating of 0, although it was necessary to have actively moved the slider to that position to proceed to the next page.

We did not attempt to construct an evaluation based on meaning-in-context for several reasons. First, determining the set of senses for a given word, and then the set of contexts that represent those senses, introduces a high degree of subjectivity into the design process. Second, ensuring that a model has learned a high quality representation of a given concept would have required evaluating that concept in each of its given contexts, necessitating many more cases and a far greater annotation effort. Third, in the (infrequent) case that some concept c_1 in an evaluation pair (c_1, c_2) is genuinely (etymologically) polysemous, c_2 can provide sufficient context to disambiguate c_1.[13] Finally, the POS grouping of pairs in the survey can also serve to disambiguate in the case that the conflicting senses of the polysemous concept

[13] This is supported by the fact that the WordNet-based methods that perform best at modeling human ratings model the similarity between concepts c_1 and c_2 as the minimum of all pairwise distances between the senses of c_1 and the senses of c_2 [145, 136].

are of differing POS category.

3.6.4 Questionnaire structure

Each participant was asked to rate 20 groups of pairs on a 0-6 scale of integers (non-integral ratings were not possible). Checkpoint multiple-choice questions were inserted at points between the 20 groups in order to ensure the participant had retained the correct notion of similarity. In addition to the checkpoint of three noun pairs presented before the first group (which contained noun pairs), checkpoint questions containing adjective pairs were inserted before the first adjective group and checkpoints of three verb pairs were inserted before the first verb group.

From the 999 evaluation pairs, 14 noun pairs, 4 verb pairs and 2 adjective pairs were selected as a *consistency set*. The dataset of pairs was then partitioned into 10 tranches, each consisting of 119 pairs, of which 20 were from the consistency set and the remaining 99 unique to that tranche. To reduce workload, each annotator was asked to rate the pairs in a single tranche only. The tranche itself was divided into 20 groups, with each group consisting of 7 pairs (with the exception of the last group of the 20, which had 6). Of these 7 pairs, the first pair was the last pair from the previous group, and the second pair was taken from the consistency set. The remaining pairs were unique to that particular group and tranche. The design enabled control for possible systematic differences between annotators and tranches, which could be detected by variation on the consistency set.

3.6.5 Participants

500 residents of the USA were recruited from Mechanical Turk, each with at least 95% approval rate for work on the web service. Each participant was required to check a box confirming that he or she was a native speaker of English and warned that work would be rejected if the pattern of responses indicated otherwise. The participants were distributed evenly to rate pairs in one of the ten question tranches, so that each pair was rated by approximately 50 subjects. Participants took between 8 and 21 minutes to rate the 119 pairs across the 20 groups, together with the checkpoint questions.

3.6.6 Post-processing

In order to correct for systematic differences in the overall calibration of the rating scale between respondents, we measured the average (mean) response of each rater on the consistency set. For 32 respondents, the absolute difference between this average and the mean of all such averages was greater than one (though never greater than two); i.e. 32 respondents demonstrated a clear tendency to rate pairs as either more or less similar than the overall rater population. To correct for this bias, we increased (or decreased) the rating of such respondents for each pair by one, except in cases where they had given the maximum rating, six (or minimum rating, zero). This adjustment, which ensured that the average response of each participant was within one of the mean of all respondents on the consistency set, resulted in a small increase to the inter-rater agreement on the dataset as a whole.

After controlling for systematic calibration differences, we imposed three conditions for the responses of a rater to be included in the final data collation. First, the average pairwise Spearman correlation of responses with all other responses for a participant could not be more than one standard deviation below the mean of all such averages. Second, the increase in inter-rater agreement when a rater was excluded from the analysis needed to be smaller than at least 50 other raters (i.e. 10% of raters were excluded on this criterion). Third, we excluded the 6 participants who got one or more of the checkpoint questions wrong. A total of 99 participants were excluded based on one or more of these conditions, but no more than 16 from any one tranche (so that each pair in the final dataset was rated by a minimum of 36 raters). Finally, we computed average (mean) scores for each pair, and transformed all scores linearly from the interval $[0, 6]$ to the interval $[0, 10]$.

3.7 Analysis of Dataset

In this section we analyse the responses of the SimLex-999 annotators and the resulting ratings. First, by considering inter-annotator agreement we examine the consistency with which annotators were able to apply the characterization of similarity outlined in the instructions to the range of concept types in SimLex-999. Second, we verify that a valid notion of similarity was understood by the annotators, in that they were able to accurately separate similarity from association.

3.7.1 Inter-annotator agreement

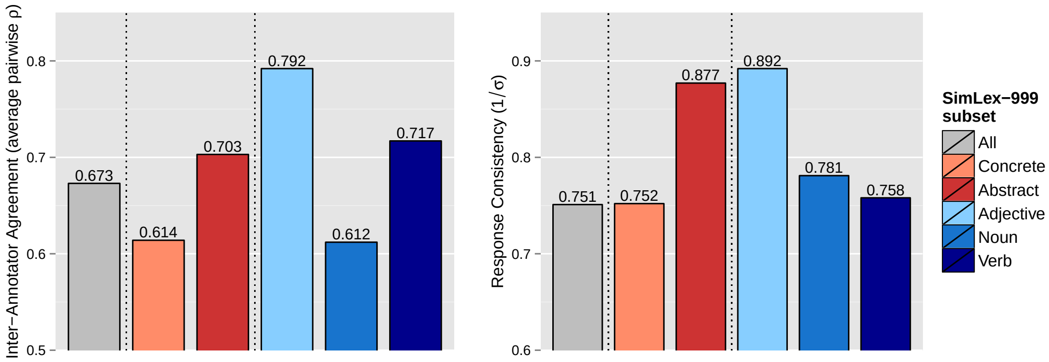


Figure 3.8: Left: Inter-annotator agreement, measured by average pairwise Spearman ρ correlation, for ratings of concept types in SimLex-999. **Right:** Response consistency, reflecting the standard deviation of annotator ratings for each pair, averaged over all pairs in the concept category.

As in previous annotation or data collection for computational semantics [132, 143, 154] we computed the inter-rater agreement as the average of pairwise Spearman ρ correlations between the ratings of all respondents. Overall agreement was $\rho = 0.67$. This compares favourably with the agreement on WS-353 ($\rho = 0.61$ using the same method). The design of the MEN rating system precludes a conventional calculation of inter-rater agreement [30]. However, two of the creators of MEN who independently rated the dataset achieved an agreement of $\rho = 0.68$.[14]

The SimLex-999 inter-rater agreement suggests that participants were able to understand the (single) characterization of similarity presented in the instructions and to apply it to concepts of various types consistently. This conclusion was supported by inspection of the brief feedback offered by the majority of annotators in a final text field in the questionnaire: 78% expressed sentiment that the test was clear, easy to complete or some similar sentiment.

Interestingly, as shown in Figure 4 (left), agreement was not uniform across the concept types. Contrary to what might be expected given established concreteness effects [133], we observed not only higher inter-rater agreement but also less per-pair variability for abstract rather than concrete concepts[15].

[14] Reported at http://clic.cimec.unitn.it/ elia.bruni/MEN. It is reasonable to assume that actual agreement on MEN may be somewhat lower than 0.68 given the small sample size and the expertise of the raters.

[15] Per-pair variability was measured by calculating the standard deviation of responses for each pair, and averaging these scores across the pairs of each concept type.

C1	C2	POS	USF*	USF rank (of 999)	SimLex	SimLex rank (of 999)
dirty	*narrow*	A	0.00	999	0.30	996
student	*pupil*	N	6.80	12	9.40	12
win	*dominate*	V	0.41	364	5.68	361
smart	*dumb*	A	2.10	92	0.60	947
attention	*awareness*	N	0.10	895	8.73	58
leave	*enter*	V	2.16	89	1.38	841

Table 3.5: Top: Similarity aligns with association Pairs with a small difference in rank between USF (association) and SimLex-999 (similarity) scores for each POS category. **Bottom: Similarity contrasts with association** Pairs with a high difference in rank for each POS category. *Note that the distribution of USF association scores on the interval [0,10] is highly skewed towards the lower bound in both SimLex-999 and the USF dataset as a whole.

Strikingly, the highest inter-rater consistency and lowest per-pair variation (defined as the inverse of the standard deviation of all ratings for that pair) was observed on adjective pairs. While we are unsure exactly what drives this effect, a possible cause is that many pairs of adjectives in SimLex-999 cohabit a single salient, one-dimensional scale (*freezing ¿ cold ¿ warm ¿ hot*). This may be a consequence of the fact that many pairs in SimLex-999 were selected (from USF) to have a degree of association. On inspection, pairs of nouns and verbs in SimLex-999 do not appear to occupy scales in the same way, possibly since concepts of these POS categories come to be associated via a more diverse range of relations. It seems plausible that humans are able to estimate the similarity of scale-based concepts more consistently than pairs of concepts related in a less uni-dimensional fashion.

Regardless of cause, however, the high agreement on adjectives is a satisfactory property of SimLex-999. Adjectives exhibit various aspects of lexical semantics that have proved challenging for computational models, including antonymy, polarity [177] and sentiment [174]. To approach the high level of human confidence on the adjective pairs in SimLex-999, it may be necessary to focus particularly on developing automatic ways to capture these phenomena.

3.7.2 Response validity: Similarity not association

Inspection of the SimLex-999 ratings indicated that pairs were indeed evaluated according to similarity rather than association. Table 2 includes examples that demonstrate a clear dissociation between the two semantic relations.

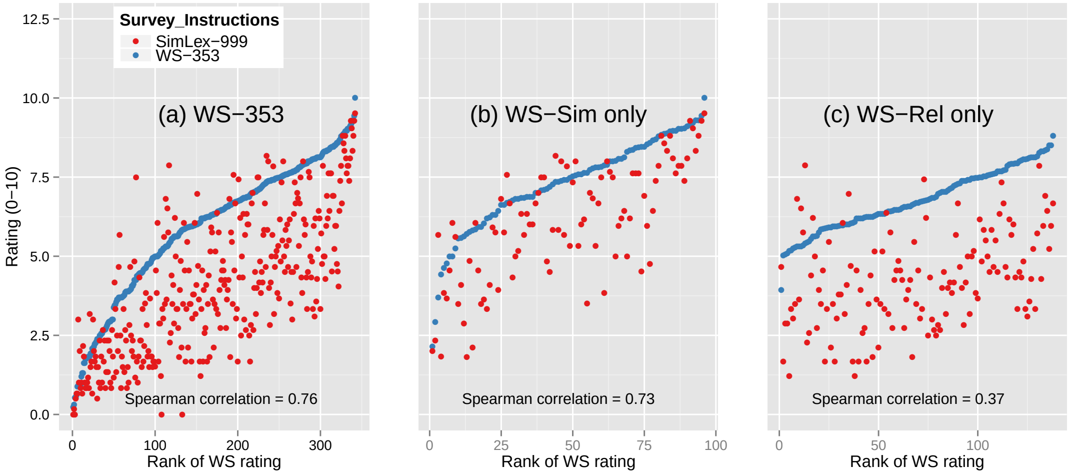


Figure 3.9: (a) Pairs rated by WS-353 annotators (blue points, ranked by rating) and the corresponding rating of annotators following the SimLex-999 instructions (red points). **(b-c)** The same analysis, restricted to pairs in the WS-Sim or WS-Rel subsets of WS-353.

To verify this effect quantitatively, we recruited 100 additional participants to rate the WS-353 pairs, but following the SimLex-999 instructions and question format. As shown in Fig 5(a), there were clear differences between these new ratings and the original WS-353 ratings. In particular, a high proportion of pairs was given a lower rating by subjects following the SimLex-999 instructions than those following the WS-353 guidelines: The mean SimLex rating was 4.07 compared with 5.91 for WS-353.

This was consistent with our expectations that pairs of associated but dissimilar concepts would receive lower ratings based on the SimLex-999 than on the WS-353 instructions while pairs that were both associated and similar would receive similar ratings in both cases. To confirm this, we compared the WS-353 and SimLex-999-based ratings on the subsets WS-Rel and WS-Sim, which were hand-sorted by [1] to include pairs connected by association (and not similarity) and those connected by similarity (but possibly also association) respectively.

As shown in Figure 5(b-c), the correlation between the SimLex-999-based and WS-353 ratings was notably higher ($\rho = 0.73$) on the WS-Sim subset than the WS-Rel subset ($\rho = 0.38$). Specifically, the tendency of subjects following the SimLex-999 instructions to assign lower ratings than those following the WS-353 instructions was far more pronounced for pairs in WS-Sim (Figure 5(b)) than for those in WS-Rel (5(c)). This observation suggest that the associated but dissimilar pairs in WS-353 were an important driver of the overall lower mean for SimLex-999-based ratings, and thus provide strong evidence that the SimLex-999 instructions do indeed enable subjects to distinguish similarity from association effectively.

3.7.3 Finer-grained Semantic Relations

We have established the validity of similarity as a notion understood by human raters and distinct from association. However, much theoretical semantics focuses on relations between words or concepts that are finer-grained than similarity and association. These include *meronymy* (a part to its whole, e.g. *blade - knife*), *hypernymy* (a category concept to a member of that category, e.g. *animal - dog*) and *cohyponymy* (two members of the same implicit category, e.g. the pair of animals *dog - cat*) [42]. Beyond theoretical interest, these relations can have practical relevance. For instance, hypernymy can form the basis of semantic entailment and therefore textual inference: the proposition *a cat is on the table* entails that *an animal is on the table* precisely because of the hypernymy relation from animal to cat.

We chose not to make these finer-grained relations the basis of our evaluation for several reasons. At present, detecting relations such as hypernymy using distributional methods is challenging, even when supported by supervised classifiers with access to labelled pairs [103]. Such a designation can seem to require specific world-knowledge (is a *snale* a *reptile*?), can be gradual, as evidenced by typicality effects [148], or simply highly subjective. Moreover, a fine-grained relation R will only be attested (to any degree) between a small subset of all possible word pairs, whereas similarity can in theory be quantified for any two words chosen at random. We thus considered a focus on fine-grained semantic relations to be less appropriate for a general-purpose evaluation of representation quality.

Nevertheless, post-hoc analysis of the SimLex annotator responses and fine-grained relation classes, as defined by lexicographers, yields further interesting insights into the nature of both similarity and association. Of the 999 word pairs in SimLex, 382 are also connected by one of the common finer-grained semantic relations in WordNet. For each of these relations, Figure 6 shows the average similarity rating and average USF free association score for all pairs that exhibit that relation.

In cases where a relationship of hypernymy/hyponymy exists between the words in a pair (not necessarily immediate - $1\_hypernym, 2\_hypernym$ etc.) similarity and association coincide. Hyper/hyponym pairs that are separated by fewer levels in the WordNet hierarchy are both more strongly associated and rated as more similar. However, there are also interesting discrepancies between similarity and association. Unsurprisingly, pairs that are classed as synonyms in WordNet (i.e. having at least one sense in some common synset) are rated as more similar than pairs of any other relation type by SimLex annotators. In contrast, antonyms are the most strongly-associated word pairs among these finer-grained relations. Further, pairs consisting of a meronym and holonym (part and whole) are comparatively strongly associated but not judged to be similar.

The analysis also highlights a case that can be particularly problematic when rating similarity; cohyponyms, or members of the same salient category (such as *knife* and *fork*). We gave no specific guidelines for how to rate such pairs in the SimLex annotator instructions, and whether they are considered similar or not seems to be a matter of perspective. On one hand, their membership of a common category could make them appear similar, particularly if the category is relatively specific. On the other hand, in the case of *knife* and *fork*, for instance, the underlying category *cultery* might provide a backdrop against which the differences of distinct members become particularly salient.

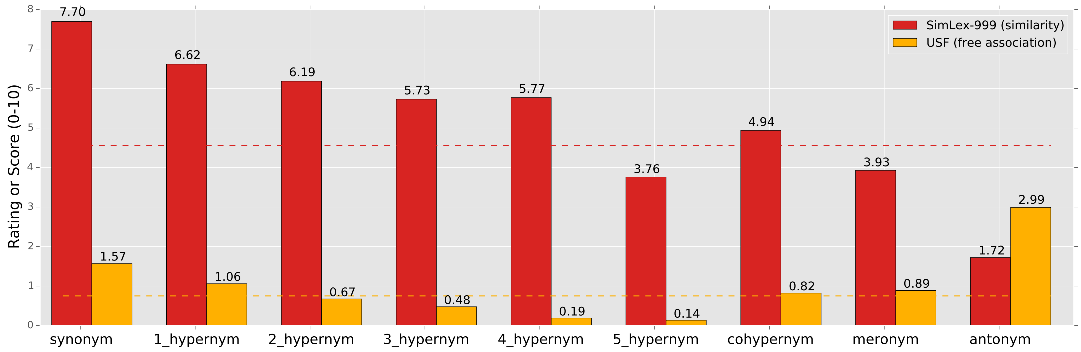


Figure 3.10: Average SimLex and USF free association scores across pairs representing different fine-grained semantic relations. All relations were extracted from WordNet. $n\_hypernym$ refers to a direct hypernymy path of length n. Note that the average SimLex rating across all 999 word pairs (dashed red line) is much higher than the average USF rating (dashed golden line) because of differences in the rating procedure. The more interesting differences concern the relative strength of similarity vs. association across the different relation types.

3.8 Evaluating Models with SimLex-999

In this section, we demonstrate the applicability of SimLex-999 by analysing the performance of various distributional semantic models in estimating the new ratings. The models were selected to cover the main classes of representation learning architectures [10]: Vector space co-occurrence (counting) models and neural language models (NLM)s [16]. We first show that SimLex-999 is notably more difficult for state-of-the-art models to estimate than existing gold standards. We then conduct more focused analyses on the various concept subsets defined in SimLex-999, exploring possible causes for the comparatively low performance of current models and, in turn, demonstrating how SimLex-999 can be applied to investigate such questions.

3.8.1 Semantic models

Collobert & Weston [40] apply the architecture of an NLM to learn a word representations v_w for each word w in some corpus vocabulary V. Each sentence s in the input text is represented by a matrix containing the vector representations of the words in s in order. The model then computes output scores $f(s)$ and $f(s^w)$, where s^w denotes an 'incorrect' sentence created from s by replacing its last word with some other word w from V. Training involves updating the parameters of the function f and the entries of the vector representations v_w such that $f(s)$ is larger than $f(s^w)$ for any w in V, other than the correct final word of s. This corresponds to minimising the sum of the following sentence objectives C_s over all sentences in the input corpus, which is achieved via (mini-batch) stochastic gradient descent:

$$C_s = \sum_{w \in V} max(0, 1 - f(s) + f(s^w)).$$

The relatively low-dimension, dense (vector) representations learned by this model and the other NLMs introduced in this section are sometimes referred to as *embeddings* [163]. [40] train their models on 852 million words of text from a 2007 dump of Wikipedia and the RCV1 Corpus [104] and use their embeddings to achieve state-of-the-art results on a variety of NLP tasks. We downloaded the embeddings directly from the authors' webpage.[16]

Huang et al. [83] present a NLM that learns word embeddings to maximise the likelihood of predicting the last word in a sentence s based on (i) the previous words in that sentence (local context - as with [40]) and (ii) the document d in which that word occurs (global context). As with [40], the model represents input sentences as a ma-

[16] http://ml.nec-labs.com/senna/

trix of word embeddings. In addition, it represents documents in the input corpus as single-vector averages over all word embeddings in that document. It can then compute scores $g(s, d)$ and $g(s^w, d)$, where as before s^w is a sentence with an 'incorrect' randomly-selected last word. Training is again by stochastic gradient descent, and corresponds to minimising the sum of the sentence objectives $C_{s,d}$ over all of the sentences in the corpus:

$$C_{s,d} = \sum_{w \in V} max(0, 1 - g(s, d) + g(s^w, d)).$$

The combination of local and global contexts in the objective encourages the final word embeddings to reflect aspects of both the meaning of nearby words and of the documents in which those words appear. When learning from 990m words of wikipedia text, Huang et al. report a Spearman correlation of $\rho = 71.3$ between the cosine similarity of their model embeddings and the WS-353 scores, which constitutes state-of-the-art performance for a NLM model on that dataset. We downloaded these embeddings from the authors' webpage.[17]

Mikolov et al. [118] present an architecture that learns word embeddings similar to those of standard NLMs but with no non-linear hidden layer (resulting in a simpler scoring function). This enables faster representation learning for large vocabularies. Despite this simplification, the embeddings achieve state-of-the-art performance on several semantic tasks including sentence completion and analogy modelling [118, 121].

For each word type w in the vocabulary V, the model learns both a 'target-embedding' $r_w \in \mathbb{R}^d$ and a 'context-embedding' $\hat{r}_w \in \mathbb{R}^d$ such that, given a target word, its ability to predict nearby context words is maximised. The probability of seeing context word c given target w is defined as:

$$p(c|w) = \frac{e^{\hat{r}_c \cdot r_w}}{\sum_{v \in V} e^{\hat{r}_v \cdot r_w}}.$$

The model learns from a set of (target-word, context-word) pairs, extracted from a corpus of sentences as follows. In a given sentence s (of length N), for each position $n \leq N$, each word w_n is treated in turn as a target word. An integer $t(n)$ is then sampled from a uniform distribution on $\{1, \ldots k\}$, where $k > 0$ is a predefined maximum

[17] `www.socher.org/index.php/Main/ImprovingWordRepresentationsViaGlobalContextAndMul`

context-window parameter. The pair tokens $\{(w_n, w_{n+j}) : -t(n) \leq j \leq t(n), w_i \in s\}$ are then appended to the training data. Thus, target/context training pairs are such that (i) only words within a k-window of the target are selected as context words for that target, and (ii) words closer to the target are more likely to be selected than those further away.

The training objective is then to maximise the log probability T, defined below, across of all such examples from s, and then across all sentences in the corpus. This is achieved by stochastic gradient descent.

$$T = \frac{1}{N} \sum_{n=1}^{N} \sum_{-t(n) \leq j \leq t(n), j \neq 0} log(p(w_{n+j}|w_n))$$

As with other NLMs, Mikolov et al.'s model captures conceptual semantics by exploiting the fact that words appearing in similar linguistic contexts are likely to have similar meanings. Informally, the model adjusts its embeddings to increase the probability of observing the training corpus. Since this probability increases with $p(c|w)$, and $p(c|w)$ increases with the dot product $\hat{r}_c \cdot r_w$, the updates have the effect of moving each target-embedding incrementally 'closer' to the context-embeddings of its collocates. In the target-embedding space, this results in embeddings of concept words that regularly occur in similar contexts moving closer together.

We use the author's Word2vec software in order to train their model and use the target embeddings in our evaluations. We experimented with embeddings of dimension 100, 200, 300, 400 and 500 and found that 200 gave the best performance on both WS-353 and SimLex-999.

Vector Space Model (VSM) As an alternative to NLMs, we constructed a vector space model following the guidelines for optimal performance outlined by [91]. After extracting the 2000 most frequent word tokens in the corpus that are not in a common list of stopwords[18] as features, we populated a matrix of co-occurrence counts with a row for each of the concepts in some pair in our evaluation sets, and a column for each of the features. Co-occurrence was counted within a specified window size, although never across a sentence boundary. This resulting matrix was then weighted according to Pointwise Mutual Information (PMI) [142]. The rows of the resulting matrix constitute the vector representations of the concepts.

[18] Taken from the Python Natural Language Toolkit [25].

SVD As proposed initially in [97], we also experimented with models in which Singular Value Decomposition (SVD) [62] is applied to the PMI-weighted VSM matrix, reducing the dimension of each concept representation to 300 (which yielded best results after experimenting, as before, with 100-500 dimension vectors).

For each model described in this section, we calculate similarity as the cosine similarity between the (vector) representations learned by that model.

3.8.2 Results

In experimenting with different models on SimLex-999, we aimed to answer the following questions: (i) How well do the established models perform on SimLex-999 versus on existing gold standards? (ii) Are any observed differences caused by the potential of different models to measure similarity vs. association? (iii) Are there interesting differences in ability of models to capture similarity between adjectives vs nouns vs verbs? (iv) In this case, are the observed differences driven by concreteness, and its interaction with POS, or are other factors also relevant?

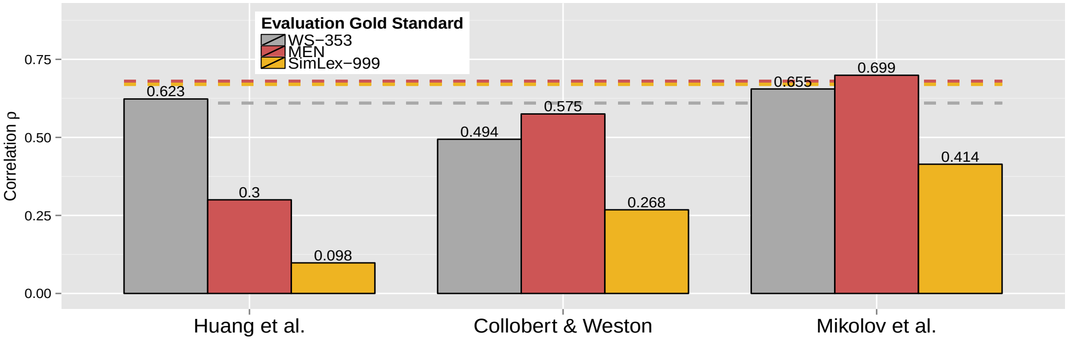


Figure 3.11: Performance of NLMs on WS-353, MEN and SimLex-999. All models are trained on Wikipedia; note that as Wikipedia is constantly growing, the [118] model exploited slightly more training data (≈1000m tokens) than the [83] model (≈990m), which in turn exploited more than the [40] model (≈852m). Dashed horizontal lines indicate the level of inter-annotator agreement for the three datasets.

Overall performance on SimLex-999 Figure 7 shows the performance of the NLMs on SimLex-999 versus on comparable datasets, measured by Spearman's ρ correlation. All models estimate the ratings of MEN and WS-353 more accurately than SimLex-

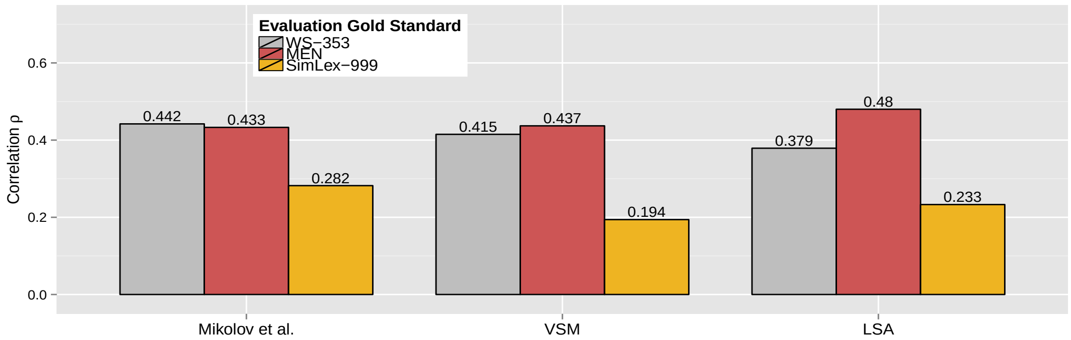


Figure 3.12: Comparison between the leading NLM, *Mikolov et al.*, the vector space model, *VSM*, and the *SVD* model. All models were trained on the ≈150m word RCV1 Corpus [104].

999. The [83] model performs well on WS-353,[19] but is not very robust to changes in evaluation gold standard, and performs worst of all the models on SimLex-999. Given the focus of the WS-353 ratings, it is tempting to explain this by concluding that the global context objective leads the [83] model to focus on association rather than similarity. However, the true explanation may be less simple, since the [83] model performs weakly on the association-based MEN dataset. The [40] model is more robust across WS-353 and MEN, but still does not match the performance of the [118] model on SimLex-999.

Figure 8 compares the best performing NLM model [118] with the VSM and SVD models.[20] In contrast to recent results that emphasise the superiority of NLMs over alternatives [10], we observed no clear advantage for the NLM over the VSM or SVD when considering the association-based gold standards WS-353 and MEN together. While the NLM is the strongest performer on WS-353, SVD is the strongest performer on MEN. However, the NLM model performs notably better than the alternatives at modelling similarity, as measured by SimLex-999.

Comparing all models in Figures 7 and 8 suggests that SimLex-999 is notably more challenging to model than the alternative datasets, with correlation scores ranging from 0.098 to 0.414. Thus, even when state-of-the-art models are trained for several days on massive text corpora, [21] their performance on SimLex-999 is well below the inter-annotator agreement (Figure 7). This suggests that there is ample scope for SimLex-

[19] This score, based on embeddings downloaded from the authors' webpage, is notably lower than the score reported in [83] mentioned in Section 5.1.

[20] We conduct this comparison on the smaller RCV1 Corpus [104] because training the VSM and SVD models is comparatively slow.

[21] Training times reported by [83] and for [40] at `http://ronan.collobert.com/senna/`.

999 to guide the development of improved models.

Modeling similarity vs. association The comparatively low performance of NLM, VSM and SVD models on SimLex-999 compared with MEN and WS-353 is consistent with our hypothesis that modelling similarity is more difficult than modelling association. Indeed, given that many strongly-associated but dissimilar pairs, such as [*coffee, cup*], are likely to have high co-occurrence in the training data, and that all models infer connections between concepts from linguistic co-occurrence in some form or another, it seems plausible that models may overestimate the similarity of such pairs because they are 'distracted' by association.

To test this hypothesis more precisely, we compared the performance of models on the whole of SimLex-999 versus its 333 most associated pairs (according to the USF free association scores). Importantly, pairs in this strongly-associated subset still span the full range of possible similarity scores (min similarity = 0.23 [*shrink, grow*], max similarity = 9.80 [*vanish, disappear*]).

As shown in Figure 9, all models performed worse when the evaluation was restricted to pairs of strongly-associated concepts, which was consistent with our hypothesis. The [40] model was better than the [83] model at estimating similarity in the face of high association. This not entirely surprising given the global-context objective in the latter model, which may have encouraged more association-based connections between concepts. The Mikolov et al. model, however, performed notably better than both other NLMs. Moreover, this superiority is proportionally greater when evaluating on the most associated pairs only (as indicated by the difference between the red and grey bars), suggesting that the improvement is driven at least in part by an increased ability to 'distinguish' similarity from association.

To understand better how the architecture of models captures information pertinent to similarity modelling, we performed two additional experiments using SimLex-999. These comparisons were also motivated by the hypotheses, made in previous studies and outlined in Section 2.1.2, that both dependency-informed input and smaller context windows encourage models to capture similarity rather than association.

We tested the first hypothesis using the embeddings of [100], whose model extends the [118] model so that target-context training instances are extracted based on dependency-parsed rather than simple running text. As illustrated in Figure 9, the dependency-based embeddings outperform the original (running text) embeddings trained on the same corpus. Moreover, the comparatively large increase in the red

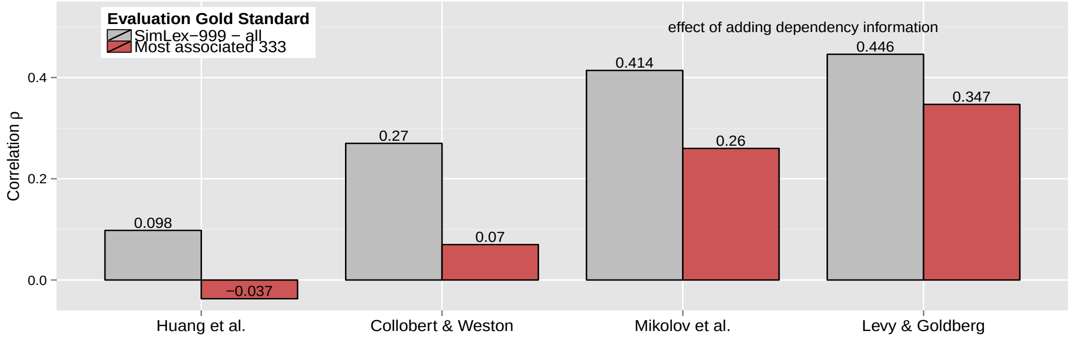


Figure 3.13: The ability of NLMs to model the similarity of highly-associated concepts versus concepts in general. The two models on the right hand side also demonstrate the effect of training an NLM (the [118] model) on running-text (*Mikolov et al.*) vs. on dependency-based input (*Levy & Goldberg*).

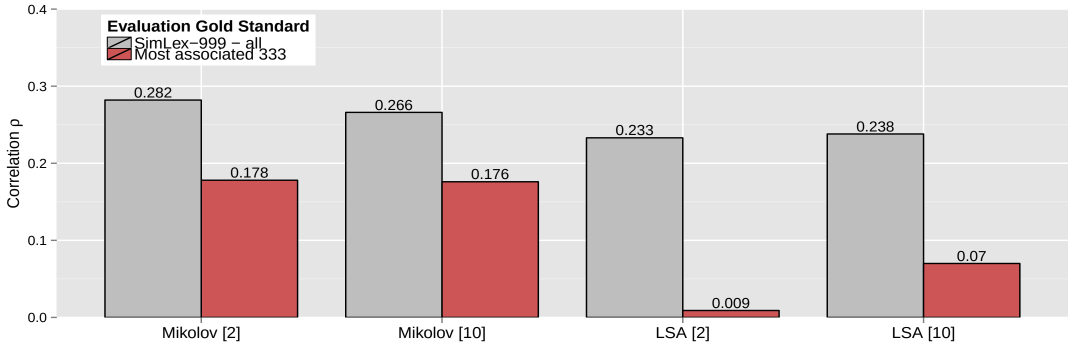


Figure 3.14: The effect of different window sizes (indicated in square brackets [ ]) on NLM and SVD models.

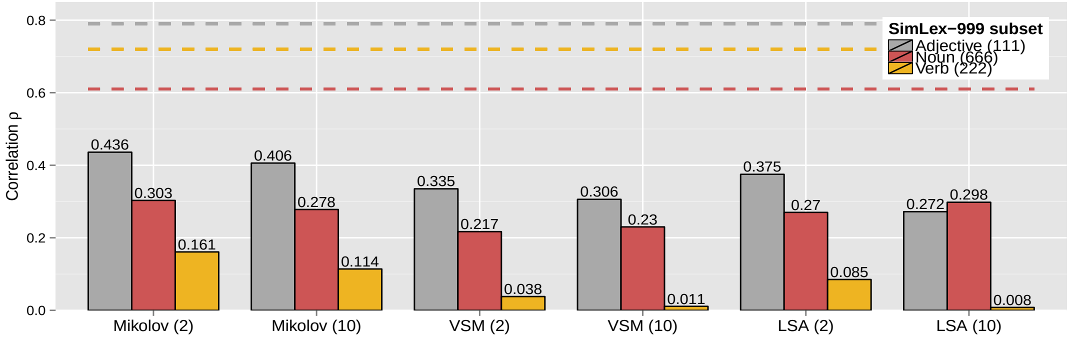


Figure 3.15: Performance of models on POS-based subsets of SimLex-999. The window size for each model is indicated in parentheses. Inter-annotator agreement for each POS is indicated by the dashed horizontal line.

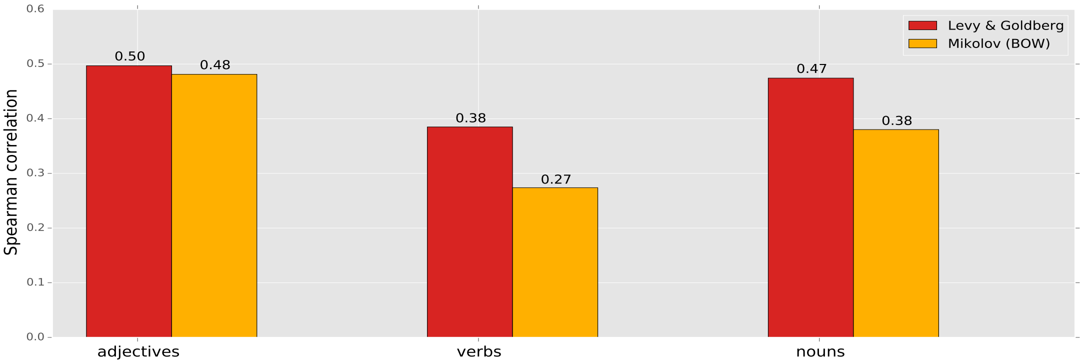


Figure 3.16: The importance of dependency-focussed contexts (in the Levy & Goldberg model) for capturing concepts of different POS, when compared to a standard Skipgram (BOW) model trained on the same Wikipedia corpus.

bar compared to the grey bar suggests that an important part of the improvement of the dependency-based model derives from a greater ability to discern similarity from association.

Our comparisons provided less support for the second (window size) hypothesis. As shown in Figure 10, there is a negligible improvement in the performance of the [] model when the window size is reduced from 10 to 2. However, for the SVD model we observed the converse. The SVD model with window size 10 slightly outperforms the SVD model with window 2, and this improvement is quite pronounced on the most associated pairs in SimLex-999.

Learning concepts of different POS Given the theoretical likelihood of variation in model performance across POS categories noted in Section 2.2, we evaluated the [118], VSM and SVD models on the subsets of SimLex-999 containing adjective, noun and verb concept pairs.

The analyses yield two notable conclusions, as shown in Figure 11. First, perhaps contrary to intuition, all models estimate the similarity of adjectives better than other concept categories. This aligns with the (also unexpected) observation that humans rate the similarity of adjectives more consistently and with more agreement than other parts of speech (see the dashed lines). However, the parallels between human raters and the models do not extend to verbs and nouns; verb similarity is rated more consistently than noun similarity by humans, but models estimate these ratings more accurately for nouns than for verbs.

To better understand the linguistic information exploited by models when acquiring concepts of different POS, we also computed performance on the POS subsets of SimLex-999 of the dependency-based model of [100] and the standard skipgram model, in which linguistic contexts are encoded as simple bags-of-words (BOW) [118] (trained on the same Wikipedia text). As shown in Figure 12, dependency-aware contexts yield the largest improvements for capturing verb similarity. This aligns with the cognitive theory of verbs as *relational concepts* [112] whose meanings rely on their interaction with (or dependency on) other words or concepts. It is also consistent with research on the automatic acquisition of verb semantics, in which syntactic features have proven particularly important [159]. While a deeper exploration of these effects is beyond the scope of this work, this preliminary analysis again highlights the how the word classes integrated into SimLex-999 are pertinent to a range of questions concerning lexical semantics.

Learning concrete and abstract concepts Given the strong interdependence between POS and conceptual concreteness (Figure 1), we aimed to explore whether the variation in model performance on different POS categories was in fact driven by an underlying effect of concreteness. To do so, we ranked each pair in the SimLex-999 dataset according to the sum of the concreteness of the two words, and compared performance of models on the most concrete and least concrete quartiles according to this ranking (Figure 13).

Interestingly, the performance of models on the most abstract and most concrete pairs suggests that the distinction characterised by concreteness is at least partially independent of POS. Specifically, while the Mikolov et al. model was the highest performer on all POS categories, its performance was worse than both the simple VSM

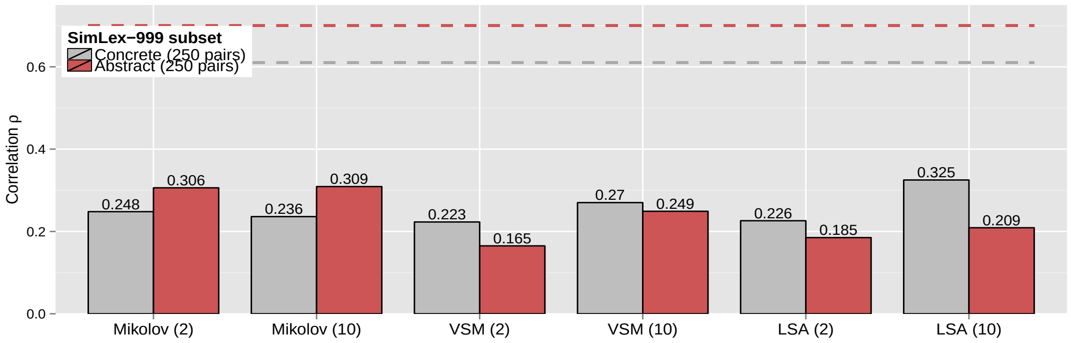


Figure 3.17: Performance of models on concreteness-based subsets of SimLex-999. Window size is indicated in parentheses. Horizontal dashed lines indicate inter-annotator agreement between SimLex-999 annotators on the two subsets.

and SVD models (of window size 10) on the most concrete concept pairs.

This finding supports the growing evidence for systematic differences in representation and/or similarity operations between abstract and concrete concepts [76], and suggests that at least part of these concreteness effects are independent of POS. In particular, it appears that models built from underlying vectors of co-occurrence counts, such as VSMs and SVD, are better equipped to capture the semantics of concrete entities, whereas the embeddings learned by NLMs can better capture abstract semantics.

3.9 Conclusion

Although the ultimate test of semantic models should be their utility in downstream applications, the research community can undoubtedly benefit from ways to evaluate the general quality of the representations learned by such models, prior to their integration in any particular system. We have presented SimLex-999, a gold standard resource for the evaluation of semantic representations containing similarity ratings of word pairs of different POS categories and concreteness levels.

The development of SimLex-999 was principally motivated by two factors. First, as we demonstrated, several existing gold standards measure the ability of models to capture association rather than similarity, and others do not adequately test their ability to discriminate similarity from association. This is despite the many potential applications for accurate similarity-focussed representation learning models. Analysis of the ratings of the 500 SimLex-999 annotators showed that subjects can consistently quantify similarity, as distinct from association, and apply it to various concept types, based on minimal intuitive instructions.

Second, as we showed, state-of-the-art models trained solely on running-text corpora have now reached or surpassed the human agreement ceiling on WordSim-353 and MEN, the most popular existing gold standards, as well as on RG and WS-Sim. These evaluations may therefore have limited use in guiding or moderating future improvements to distributional semantic models. Nevertheless, there is clearly still room for improvement in terms of the use of distributional models in functional applications. We therefore consider the comparatively low performance of state-of-the-art models on SimLex-999 to be one of its principal strengths. There is clear room under the inter-rating ceiling to guide the development of the next generation of distributional models.

We conducted a brief exploration of how models might improve on this performance, and verified the hypotheses that models trained on dependency-based input capture similarity more effectively than those trained on running-text input. The evidence that smaller context windows are also beneficial for similarity models was mixed, however. Indeed, we showed that the optimal window size depends on both the general model architecture and the part-of-speech and concreteness of the target concepts.

Our analysis of these hypotheses illustrates how the design of SimLex-999 - covering a principled set of concept categories and including meta-information on concreteness and free-association strength - enables fine-grained analyses of the performance and parameterization of semantic models. However, these experiments only scratch the surface in terms of the possible analyses. We hope that researchers will adopt the resource as a robust means of answering a diverse range of questions pertinent to similarity modelling, distributional semantics and representation learning in general.

In particular, for models to learn high-quality representations for all linguistic concepts, we believe that future work must uncover ways to explicitly or implicitly infer 'deeper', more general, conceptual properties such as intentionality, polarity, subjectivity or concreteness [60]. However, while improving corpus-based models in this direction is certainly realistic, models that learn exclusively via the linguistic modality may never reach human-level performance on evaluations such as SimLex-999. This is because much conceptual knowledge, and particularly that which underlines similarity computations for concrete concepts, appears to be grounded in the perceptual modalities as much as in language [13].

Whatever the means by which the improvements are achieved, accurate concept-level representation is likely to constitute a necessary first step towards learning informative, language-neutral phrasal and sentential representations. Such representations would be hugely valuable for fundamental NLP applications such as language understanding tools and machine translation.

Distributional semantics aims to infer the meaning of words based on the *company they keep* [56]. However, while words that occur together in text often have associated meanings, these meanings may be very similar or indeed very different. Thus, possibly excepting the population of Argentina, most people would agree that, strictly speaking, *Maradona* is not synonymous with *football* (despite their high rating of 8.62 in WordSim-353). The challenge for the next generation of distributional models may therefore be to infer what is useful from the co-occurrence signal and to overlook what is not. Perhaps only then will models capture most, or even all, of what humans know when they know how to use a language.

3.10 Sequence-to-Sequence Learning of Word Representations From Bilingual Data

It is well known that word representations can be learned from the distributional patterns in corpora. Originally, such representations were constructed by counting word co-occurrences, so that the features in one word's representation corresponded to other words [97, 165]. Neural language models, an alternative method for learning word representations, use language data to optimise (latent) features with respect to a language modelling objective. The objective can be to predict either the next word given the initial words of a sentence [16, 125, 40], or simply a nearby word given a single cue word [121, 137]. The representations learned by neural models (sometimes called *embeddings*) perform very effectively when applied as pre-trained features in a range of NLP applications and tasks [10].

Despite these clear results, it is not well understood how the architecture of neural models affects the information encoded in their embeddings. Here we contribute to this understanding by considering the embeddings learned by architectures with a very different objective function: *neural machine translation* (NMT) *models*. NMT models have recently emerged as an alternative to statistical, phrase-based translation models, and are beginning to achieve impressive translation performance [89, 47, 161].

We show that NMT models are not only a potential new direction for machine translation, but are also an effective means of learning word embeddings. Specifically, translation-based embeddings encode information relating to conceptual similarity (rather than non-specific relatedness or association) and lexical syntactic role more effectively than embeddings from monolingual neural language models. We demonstrate that these properties persist when translating between different language pairs (English-French and English-German). Further, based on the observation of subtle language-specific effects in the embedding spaces, we conjecture as to why similarity dominates over other semantic relations in translation embedding spaces. Finally, we discuss a potential limitation of the application of NMT models for embedding learning - the computational cost of training large vocabularies of embeddings - and show that a novel method for overcoming this issue preserves the aforementioned properties of translation-based embeddings.

3.11 Learning Embeddings with Neural Language Models

All neural language models, including NMT models, learn real-valued embeddings (of specified dimension) for words in some pre-specified vocabulary, V, covering many or all words in their training corpus. At each training step, a 'score' for the current training example (or batch) is computed based on the embeddings in their current state. This score is compared to the model's objective function, and the error is backpropagated to update both the model weights (affecting how the score is computed from the embeddings) and the embedding features themselves. At the end of this process, the embeddings should encode information that enables the model to optimally satisfy its objective.

3.11.1 Monolingual Models

In the original neural language model [16] and subsequent variants [40], training examples consist of an ordered sequence of n words, with the model trained to predict the n-th word given the first $n-1$ words. The model first represents the input as an ordered sequence of embeddings, which it transforms into a single fixed length 'hidden' representation, generally by concatenation and non-linear projection. Based on this representation, a probability distribution is computed over the vocabulary, from which the model can sample a guess for the next word. The model weights and embeddings are updated to maximise the probability of correct guesses for all sentences in the training corpus.

More recent work has shown that high quality word embeddings can be learned via simpler models with no nonlinear hidden layer [121, 137]. Given a single word or unordered window of words in the corpus, these models predict which words will occur nearby. For each word w in V, a list of training cases $(w, c) : c \in V$ is extracted from the training corpus according to some algorithm. For instance, in the *skipgram* approach [121], for each 'cue word' w the 'context words' c are sampled from windows either side of tokens of w in the corpus (with c more likely to be sampled if it occurs closer to w).[22] For each w in V, the model initialises both a cue-embedding, representing the w when it occurs as a cue-word, and a context-embedding, used when w occurs as a context-word. For a cue word w, the model uses the corresponding cue-embedding and all context-embeddings to compute a probability distribution over V that reflects the probability of a word occurring in the context of w. When a training example (w, c) is observed, the model updates both the cue-word embedding of w and the context-word embeddings in order to increase the conditional probability of c.

3.11.2 Bilingual Representation-learning Models

Various studies have demonstrated that word representations can also be effectively learned from bilingual corpora, aligned at the document, paragraph or word level [67, 171, 120, 74, 33]. These approaches aim to represent the words from two (or more) languages in a common vector space so that words in one language are close to words with similar or related meanings in the other. The resulting multilingual embedding spaces have been effectively applied to bilingual lexicon extraction [67, 171, 120] and document classification [95, 74, 33, 96].

We focus our analysis on two representatives of this class of (non-NMT) bilingual model. The first is that of [74], whose embeddings improve on the performance of [95] in document classification applications. As with the NMT models introduced in the next section, this model can be trained directly on bitexts aligned only at the sentence rather than word level. When training, for aligned sentences S_E and S_F in different languages, the model computes representations R_E and R_F by summing the

[22] Subsequent variants use different algorithms for selecting the (w, c) from the training corpus [77, 100]

embeddings of the words in S_E and S_F respectively. The embeddings are then updated to minimise the divergence between R_E and R_F (since they convey a common meaning). A noise-contrastive loss function ensures that the model does not arrive at trivial (e.g. all zero) solutions to this objective. [74] show that, despite the lack of prespecified word alignments, words in the two languages with similar meanings converge in the bilingual embedding space.[23]

The second model we examine is that of [51]. Unlike the models described above, [51] showed explicitly that projecting word embeddings from two languages (learned independently) into a common vector space can favourably influence the orientation of word embeddings when considered in their monolingual subspace; i.e relative to other words in their own language. In contrast to the other models considered in this paper, the approach of [51] requires bilingual data to be aligned at the word level.

3.11.3 Neural Machine Translation Models

The objective of NMT is to generate an appropriate sentence in a target language S_t given a sentence S_s in the source language (see e.g. [89, 161]). As a by-product of learning to meet this objective, NMT models learn distinct sets of embeddings for the vocabularies V_s and V_t in the source and target languages respectively.

Observing a training case (S_s, S_t), these models represent S_s as an ordered sequence of embeddings of words from V_s. The sequence for S_s is then encoded into a single representation R_S.[24] Finally, by referencing the embeddings in V_t, R_S and a representation of what has been generated thus far, the model decodes a sentence in the target language word by word. If at any stage the decoded word does not match the corresponding word in the training target S_t, the error is recorded. The weights and embeddings in the model, which together parameterise the encoding and decoding process, are updated based on the accumulated error once the sentence decoding is complete.

Although NMT models can differ in their low-level architecture [89, 37, 7], the translation objective exerts similar pressure on the embeddings in all cases. The source language embeddings must be such that the model can combine them to form single

[23] The models of [33] and [74] both aim to minimise the divergence between source and target language sentences represented as sums of word embeddings. Because of these similarities, we do not compare with both in this paper.

[24] Alternatively, subsequences (phrases) of S_s may be encoded at this stage in place of the whole sentence [7].

representations for ordered sequences of multiple words (which in turn must enable the decoding process). The target language embeddings must facilitate the process of decoding these representations into correct target-language sentences.

3.12 Experiments

To learn translation-based embeddings, we trained two different NMT models. The first is the RNN encoder-decoder, *RNNenc* [37], which uses a recurrent-neural-network to encode all of the source sentence into a single vector on which the decoding process is conditioned. The second is the *RNN Search* architecture [7], which was designed to overcome limitations exhibited by the RNN encoder-decoder when translating very long sentences. RNN Search includes a *attention* mechanism, an additional feed-forward network that learns to attend to different parts of the source sentence when decoding each word in the target sentence.[25] Both models were trained on a 348m word corpus of English-French sentence pairs or a 91m word corpus of English-German sentence pairs.[26]

To explore the properties of bilingual embeddings learned via objectives other than direct translation, we trained the *BiCVM* model of [74] on the same data, and also downloaded the projected embeddings of [51], *FD*, trained on a bilingual corpus of comparable size (≈ 300 million words per language).[27] Finally, for an initial comparison with monolingual models, we trained a conventional skipgram model [121] and its *Glove* variant [137] for the same number of epochs on the English half of the bilingual corpus.

To analyse the effect on embedding quality of increasing the quantity of training data, we then trained the monolingual models on increasingly large random subsamples of Wikipedia text (up to a total of 1.1bn words). Lastly, we extracted embeddings from a full-sentence language model, *CW*, [40], which was trained for several months on the same Wikipedia 1bn word corpus. Note that increasing the volume of training

[25] Access to source code and limited GPU time prevent us from training and evaluating the embeddings from other NMT models such as that of [89], [47] and [161]. The underlying principles of encoding-decoding also apply to these models, and we expect the embeddings would exhibit similar properties to those analysed here.

[26] These corpora were produced from the WMT '14 parallel data after conducting the data-selection procedure described by [37].

[27] Available from `http://www.cs.cmu.edu/~mfaruqui/soft.html`. The available embeddings were trained on English-German aligned data, but the authors report similar to for English-French.

data for the bilingual (and NMT) models was not possible because of the limited size of available sentence-aligned bitexts.

3.12.1 Similarity and relatedness modelling

As in previous studies [2, 29, 10], our initial evaluations involved calculating pairwise (cosine) distances between embeddings and correlating these distances with (gold-standard) human judgements of the strength of relationships between concepts. For this we used three different gold standards: WordSim-353 [2], MEN [29] and SimLex-999 [80]. Importantly, there is a clear distinction between WordSim-353 and MEN, on the one hand, and SimLex-999, on the other, in terms of the semantic relationship that they quantify. For both WordSim-353 and MEN, annotators were asked to rate how *related* or *associated* two concepts are. Consequently, pairs such as [*clothes-closet*], which are clearly related but ontologically dissimilar, have high ratings in WordSim-353 and MEN. In contrast, such pairs receive a low rating in SimLex-999, where only genuinely *similar* concepts, such as [*coast- shore*], receive high ratings.

To reproduce the scores in SimLex-999, models must thus distinguish pairs that are similar from those that are merely related. In particular, this requires models to develop sensitivity to the distinction between synonyms (similar) and antonyms (often strongly related, but highly dissimilar).[28]

Table 3.6 shows the correlations of NMT (English-French) embeddings, other bilingually-trained embeddings and monolingual embeddings with these three lexical gold-standards. NMT outperform monolingual embeddings, and, to a lesser extent, the other bilingually trained embeddings, on SimLex-999. However, this clear advantage is not observed on MEN and WordSim-353, where the projected embeddings of [51], which were tuned for high performance on WordSim-353, perform best. Given the aforementioned differences between the evaluations, this suggests that bilingually-trained embeddings, and NMT based embeddings in particular, better capture similarity, whereas monolingual embedding spaces are orientated more towards relatedness.

To test this hypothesis further, we ran three more evaluations designed to probe the sensitivity of models to similarity as distinct from relatedness or association. In the first, we measured performance on SimLex-Assoc-333 [80]. This evaluation comprises the 333 most related pairs in SimLex-999, according to an independent empirical measure of relatedness (free associate generation [130]). Importantly, the pairs

[28] For a more detailed discussion of the similarity/relatedness distinction, see [80].

		Monolingual models			Biling. models		NMT models	
		Skipgram	**Glove**	**CW**	**FD**	**BiCVM**	**RNNenc**	**RNNsearch**
WordSim-353	ρ	0.52	0.55	0.51	**0.69**	0.50	0.57	0.58
MEN	ρ	0.44	0.71	0.60	**0.78**	0.45	0.63	0.62
SimLex-999	ρ	0.29	0.32	0.28	0.39	0.36	**0.52**	0.49
SimLex-333	ρ	018	0.18	0.07	0.24	0.34	**0.49**	0.45
TOEFL	%	0.75	0.78	0.64	0.84	0.87	**0.93**	**0.93**
Syn/antonym	%	0.69	0.72	0.75	0.76	0.70	**0.79**	0.74

Table 3.6: NMT embeddings (RNNenc and RNNsearch) clearly outperform alternative embedding-learning architectures on tasks that require modelling similarity (below the dashed line), but not on tasks that reflect relatedness. Bilingual embedding spaces learned without the translation objective are somewhere between these two extremes.

	Skipgram	**Glove**	**CW**	**FD**	**BiCVM**	**RNNenc**	**RNNsearch**
teacher	*vocational*	*student*	*student*	*elementary*	*faculty*	*professor*	*instructor*
	in-service	*pupil*	*tutor*	*school*	*professors*	*instructor*	*professor*
	college	*university*	*mentor*	*classroom*	*teach*	*trainer*	*educator*
eaten	*spoiled*	*cooked*	*baked*	*ate*	*eating*	*ate*	*ate*
	squeezed	*eat*	*peeled*	*meal*	*eat*	*consumed*	*consumed*
	cooked	*eating*	*cooked*	*salads*	*baking*	*tasted*	*eat*
Britain	*Northern*	*Ireland*	*Luxembourg*	*UK*	*UK*	*UK*	*England*
	Great	*Kingdom*	*Belgium*	*British*	*British*	*British*	*UK*
	Ireland	*Great*	*Madrid*	*London*	*England*	*America*	*Syria*

Table 3.7: Nearest neighbours (excluding plurals) in the embedding spaces of different models. All models were trained for 6 epochs on the translation corpus except CW and FD (as noted previously). NMT embedding spaces are oriented according to similarity, whereas embeddings learned by monolingual models are organized according to relatedness. The other bilingual model BiCVM also exhibits a notable focus on similarity.

in SimLex-Assoc-333, while all strongly related, still span the full range of similarity scores.[29] Therefore, the extent to which embeddings can model this data reflects their sensitivity to the similarity (or dissimilarity) of two concepts, even in the face of a strong signal in the training data that those concepts are related.

The TOEFL synonym test is another similarity-focused evaluation of embedding spaces. This test contains 80 cue words, each with four possible answers, of which one is a correct synonym [97]. We computed the proportion of questions answered correctly by each model, where a model's answer was the nearest (cosine) neighbour to the cue word in its vocabulary.[30] Note that, since TOEFL is a test of synonym recog-

[29] The most dissimilar pair in SimLex-Assoc-333 is [*shrink,grow*] with a score of 0.23. The highest is [*vanish,disappear*] with 9.80.

[30] To control for different vocabularies, we restricted the effective vocabulary of each model to the

nition, it necessarily requires models to recognise similarity as opposed to relatedness.

Finally, we tested how well different embeddings enabled a supervised classifier to distinguish between synonyms and antonyms, since synonyms are necessarily similar and people often find antonyms, which are necessarily dissimilar, to be strongly associated. For 744 word pairs hand-selected as either synonyms or antonyms,[31] we presented a Gaussian SVM with the concatenation of the two word embeddings. We evaluated accuracy using 10-fold cross-validation.

As shown in Table 3.6, with these three additional similarity-focused tasks we again see the same pattern of results. NMT embeddings outperform other bilingually-trained embeddings which in turn outperform monolingual models. The difference is particularly striking on SimLex-Assoc-333, which suggests that the ability to discern similarity from relatedness (when relatedness is high) is perhaps the most clear distinction between the bilingual spaces and those of monolingual models.

These conclusions are also supported by qualitative analysis of the various embedding spaces. As shown in Table 3.7, in the NMT embedding spaces the nearest neighbours (by cosine distance) to concepts such as *teacher* are genuine synonyms such as *professor* or *instructor*. The bilingual objective also seems to orientate the non-NMT embeddings towards semantic similarity, although some purely related neighbours are also oberved. In contrast, in the monolingual embedding spaces the neighbours of *teacher* include highly related but dissimilar concepts such as *student* or *college*.

3.12.2 Importance of training data quantity

In previous work, monolingual models were trained on corpora many times larger than the English half of our parallel translation corpus. Indeed, the ability to scale to large quantities of training data was one of the principal motivations behind the skipgram architecture [121]. To check if monolingual models simply need more training data to capture similarity as effectively as bilingual models, we therefore trained them on increasingly large subsets of Wikipedia.[32] As shown in Figure 3.18, this is not in fact the case. The performance of monolingual embeddings on similarity tasks remains well below the level of the NMT embeddings and somewhat lower than the non-MT

intersection of all model vocabularies, and excluded all questions that contained an answer outside of this intersection.

[31] Available online at `http://www.cl.cam.ac.uk/~fh295/`.

[32] We did not do the same for our translation models because sentence-aligned bilingual corpora of comparable size do not exist.

bilingual embeddings as the amount of training data increases.

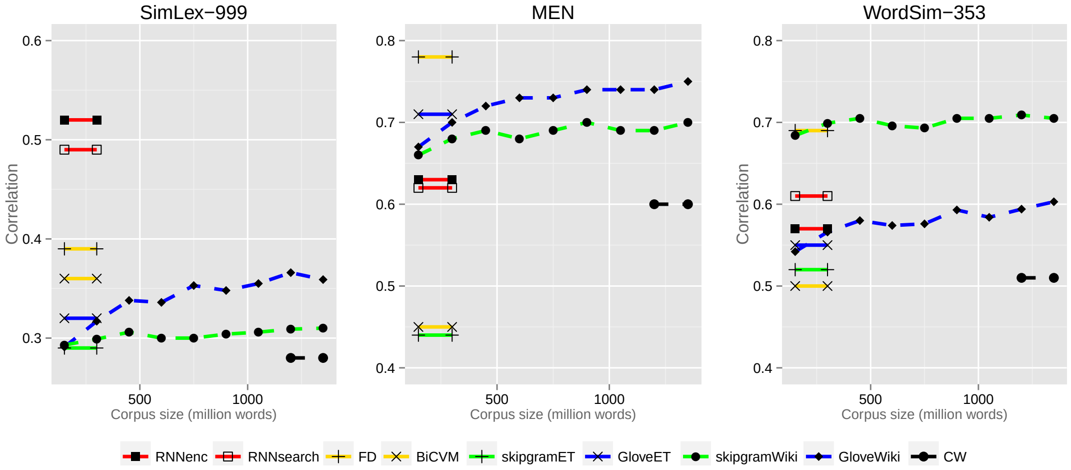


Figure 3.18: The effect of increasing the amount of training data on the quality of monolingual embeddings, based on similarity-based evaluations (SimLex-999) and two relatedness-based evaluations (MEN and WordSim-353). *ET* in the legend indicates models trained on the English half of the translation corpus. *Wiki* indicates models trained on Wikipedia.

3.12.3 Analogy Resolution

Lexical analogy questions have been used as an alternative way of evaluating word representations. In this task, models must identify the correct answer (*girl*) when presented with analogy questions such as ‘*man* is to *boy* as *woman* is to ?’. It has been shown that Skipgram-style models are surprisingly effective at answering such questions [121]. This is because, if $\mathbf{m}, \mathbf{b}$ and $\mathbf{w}$ are skigram-style embeddings for *man*, *boy* and *woman* respectively, the correct answer is often the nearest neighbour in the vocabulary (by cosine distance) to the vector $\mathbf{v} = \mathbf{w} + \mathbf{b} - \mathbf{m}$.

We evaluated embeddings on analogy questions using the same vector-algebra method as [121]. As in the previous section, for fair comparison we excluded questions containing a word outside the intersection of all model vocabularies, and restricted all answer searches to this reduced vocabulary. This left 11,166 analogies. Of these, 7219 are classed as ‘syntactic’, in that they exemplify mappings between parts-of-speech or syntactic roles (e.g. *fast* is to *fastest* as *heavy* is to *heaviest*), and 3947 are classed as ‘semantic‘ (*Ottawa* is to *Canada* as *Paris* is to *France*), since successful answering seems to rely on some (world) knowledge of the concepts themselves.

As shown in Fig. 3.19, NMT embeddings yield relatively poor answers to semantic analogy questions compared with monolingual embeddings and the bilingual embeddings *FD* (which are projections of similar monolingual embeddings).[33] It appears that the translation objective prevents the embedding space from developing the same linear, geometric regularities as skipgram-style models with respect to semantic organisation. This also seems to be true of the embeddings from the full-sentence language model *CW*. Further, in the case of the Glove and FD models this advantage seems to be independent of both the domain and size of the training data, since embeddings from these models trained on only the English half of the translation corpus still outperform the translation embeddings.

On the other hand, NMT embeddings are effective for answering syntactic analogies using the vector algebra method. They perform comparably to or even better than monolingual embeddings when trained on less data (albeit bilingual data). It is perhaps unsurprising that the translation objective incentivises the encoding of a high degree of lexical syntactic information, since coherent target-language sentences could not be generated without knowledge of the parts-of-speech, tense or case of its vocabulary items. The connection between the translation objective and the embedding of lexical syntactic information is further supported by the fact that embeddings learned by the bilingual model BiCVM do not perform comparably on the syntactic analogy task. In this model, sentential semantics is transferred via a bag-of-words representation, presumably rendering the precise syntactic information less important.

When considering the two properties of NMT embeddings highlighted by these experiments, namely the encoding of semantic similarity and lexical syntax, it is worth noting that items in the similarity-focused evaluations of the previous section (SimLex-999 and TOEFL) consist of word groups or pairs that have identical syntactic role. Thus, even though lexical semantic information is in general pertinent to conceptual similarity [100], the lexical syntactic and conceptual properties of translation embeddings are in some sense independent of one another.

3.13 Effect of Target Language

To better understand why a translation objective yields embedding spaces with particular properties, we trained the RNN Search architecture to translate from English to German.

[33] The performance of the FD embeddings on this task is higher than that reported by [51] because we search for answers over a smaller total candidate vocabulary.

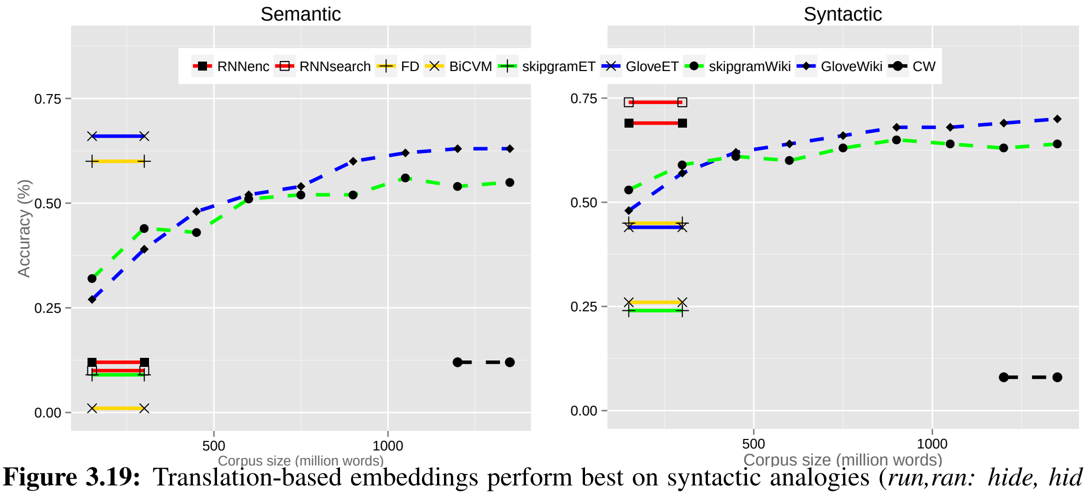


Figure 3.19: Translation-based embeddings perform best on syntactic analogies (*run,ran: hide, hid*). Monolingual skipgram/Glove models are better at semantic analogies (*father, man; mother, woman*)

		EN-FR	EN-DE
WordSim-353	ρ	0.60	**0.61**
MEN	ρ	0.61	**0.62**
SimLex-999	ρ	0.49	**0.50**
SimLex-Assoc-333	ρ	0.45	**0.47**
TOEFL	%	0.90	**0.93**
Syn/antonym	%	**0.72**	0.70
Syntactic analogies	%	**0.73**	0.62
Semantic analogies	%	0.10	**0.11**

	'earned'	'castle'	'money'
	gained	*chateau*	***silver***
EN-FR	***won***	*palace*	*funds*
	acquired	*fortress*	*cash*
	gained	*chateau*	*funds*
EN-DE	*deserved*	*palace*	*cash*
	accummulated	***padlock***	*resources*

Table 3.8: Comparison of embeddings learned by RNN Search models translating between English-French (EN-FR) and English-German (EN-DE) on all semantic evaluations (left) and nearest neighbours of selected cue words (right). Bold italics indicate target-language-specific effects. Evaluation items and vocabulary searches were restricted to words common to both models.

As shown in Table 3.8 (left side), the performance of the source (English) embeddings learned by this model was comparable to that of those learned by the English-to-French model on all evaluations, even though the English-German training corpus (91 million words) was notably smaller than the English-French corpus (348m words). This evidence shows that the desirable properties of translation embeddings highlighted thus far are not particular to English-French translation, and can also emerge when translating to a different language family, with different word ordering conventions.

3.14 Overcoming the Vocabulary Size Problem

A potential drawback to using NMT models for learning word embeddings is the computational cost of training such a model on large vocabularies. To generate a target language sentence, NMT models repeatedly compute a softmax distribution over the target vocabulary. This computation scales with vocabulary size and must be repeated for each word in the output sentence, so that training models with large output vocabularies is challenging. Moreover, while the same computational bottleneck does not apply to the encoding process or source vocabulary, there is no way in which a translation model could learn a high quality source embedding for a word if the plausible translations were outside its vocabulary. Thus, limitations on the size of the target vocabulary effectively limit the scope of NMT models as representation-learning tools. This contrasts with the shallower monolingual and bilingual representation-learning models considered in this paper, which efficiently compute a distribution over a large target vocabulary using either a hierarchical softmax [127] or approximate methods such as negative sampling [121, 74], and thus can learn large vocabularies of both source and target embeddings.

A recently proposed solution to this problem enables NMT models to be trained with larger target vocabularies (and hence larger meaningful source vocabularies) at comparable computational cost to training with a small target vocabulary [86]. The algorithm uses (biased) importance sampling [17] to approximate the probability distribution of words over a large target vocabulary with a finite set of distributions over subsets of that vocabulary. Despite this element of approximation in the decoder, extending the effective target vocabulary in this way significantly improves translation performance, since the model can make sense of more sentences in the training data and encounters fewer unknown words at test time. In terms of representation learning, the method provides a means to scale up the NMT approach to vocabularies as large as those learned by monolingual models. However, given that the method replaces an exact calculation with an approximate one, we tested how the quality of source embeddings is affected by scaling up the target language vocabulary in this way.

As shown in Table 3.9, there is no significant degradation of embedding quality when scaling to large vocabularies with using the approximate decoder. Note that for a fair comparison we filtered these evaluations to only include items that are present in the smaller vocabulary. Thus, the numbers do not directly reflect the quality of the additional 470k embeddings learned by the extended vocabulary models, which

		RNN Search	**RNN Search**	**RNN Search-LV**	**RNN Search-LV**
		EN-FR	**EN-DE**	**EN-FR**	**EN-DE**
WordSim-353	ρ	0.60	**0.61**	0.59	0.57
MEN	ρ	0.61	**0.62**	**0.62**	0.61
SimLex-999	ρ	0.49	0.50	**0.51**	0.50
SimLex-Assoc-333	ρ	0.45	**0.47**	**0.47**	0.46
TOEFL	%	0.90	0.93	0.93	**0.98**
Syn/antonym	%	0.72	0.70	**0.74**	0.71
Syntactic analogies	%	**0.73**	0.62	0.71	0.62
Semantic analogies	%	0.10	0.11	0.08	**0.13**

Table 3.9: Comparison of embeddings learned by the original (RNN Search - 30k French words, 50k German words) and extended-vocabulary (RNN Search-LV -500k words) models translating from English to French (EN-FR) and from English to German (EN-DE). For fair comparisons, all evaluations were restricted to the intersection of all model vocabularies.

one would expect to be lower since they are words of lower frequency. All embeddings can be downloaded from `http://www.cl.cam.ac.uk/~fh295/`, and the embeddings from the smaller vocabulary models can be interrogated at `http://lisa.iro.umontreal.ca/mt-demo/embs/`.[34]

3.15 How Similarity Emerges

Although NMT models appear to encode both conceptual similarity and syntactic information for any source and target languages, it is not the case that embedding spaces will always be identical. Interrogating the nearest neighbours of the source embedding spaces of the English-French and English-German models reveals occasional language-specific effects. As shown in Table 3.8 (right side), the neighbours for the word *earned* in the English-German model are as one might expect, whereas the neighbours from the English-French model contain the somewhat unlikely candidate *won*. In a similar vein, while the neighbours of the word *castle* from the English-French model are unarguably similar, the neighbours from the English-German model contain the word *padlock*.

These infrequent but striking differences between the English-German and English-French source embedding spaces indicate how similarity might emerge effectively in NMT models. Tokens of the French verb *gagner* have (at least) two possible English

[34] A different solution to the rare-word problem was proposed by [109]. We do not evaluate the effects on the resulting embeddings of this method because we lack access to the source code.

translations (*win* and *earn*). Since the translation model, which has limited encoding capacity, is trained to map tokens of *win* and *earn* to the same place in the target embedding space, it is efficient to move these concepts closer in the source space. Since *win* and *earn* map directly to two different verbs in German, this effect is not observed. On the other hand, the English nouns *castle* and *padlock* translate to a single noun (*Schloss*) in German, but different nouns in French. Thus, *padlock* and *castle* are only close in the source embeddings from the English-German model.

Based on these considerations, we can conjecture that the following condition on the semantic configuration between two language is crucial to the effective induction of lexical similarity.

(1) For s_1 and s_2 in the source language, there is some t in the target language such that there are sentences in the training data in which s_1 translates to t and sentences in which s_2 translates to t.

if and only if

(2) s_1 and s_2 are semantically similar.

Of course, this condition is not true in general. However, we propose that the extent to which it holds over all possible word pairs corresponds to the quality of similarity induction in the translation embedding space. Note that strong polysemy in the target language, such as *gagner* = *win, earn*, can lead to cases in which 1 is satisfied but 2 is not. The conjecture claims that these cases are detrimental to the quality of the embedding space (at least with regards to similarity). In practice, qualitative analyses of the embedding spaces and native speaker intuitions suggest that such cases are comparatively rare. Moreover, when such cases are observed, s_1 and s_2, while perhaps not similar, are not strongly dissimilar. This could explain why related but strongly dissimilar concepts such as antonym pairs do not converge in the translation embedding space. This is also consistent with qualitative evidence presented by [51] that projecting monolingual embeddings into a bilingual space orientates them to better reflect the synonymy/antonymy distinction.

3.16 Conclusion

In this work, we have shown that the embedding spaces from neural machine translation models are orientated more towards conceptual similarity than those of monolingual models, and that translation embedding spaces also reflect richer lexical syntactic information. To perform well on similarity evaluations such as SimLex-999, embeddings must distinguish information pertinent to what concepts *are* (their function or ontology) from information reflecting other non-specific inter-concept relationships. Concepts that are strongly related but dissimilar, such as antonyms, are particularly challenging in this regard [80]. Consistent with the qualitative observation made by [51], we suggested how the nature of the semantic correspondence between the words in languages enables NMT embeddings to distinguish synonyms and antonyms and, more generally, to encode the information needed to reflect human intuitions of similarity.

The language-specific effects we observed in Section 3.13 suggest a potential avenue for improving translation and multi-lingual embeddings in future work. First, as the availability of fast GPUs for training grows, we would like to explore the embeddings learned by NMT models that translate between much more distant language pairs such as English-Chinese or English-Arabic. For these language pairs, the word alignment will less monotonic and may result in even more important semantic and syntactic information being encoded in the lexical representation. Further, as observed by both [74] and [51], the bilingual representation learning paradigm can be naturally extended to update representations based on correspondences between multiple languages (for instance by interleaving English-French and English-German training examples). Such an approach should smooth out language-specific effects, leaving embeddings that encode only language-agnostic conceptual semantics and are thus more generally applicable. Another related challenge is to develop smaller or less complex representation-learning tools that encode similarity with as much fidelity as NMT models but without the computational overhead. One promising approach for this is to learn word alignments and word embeddings jointly [96]. This approach is effective for cross-lingual document classification, although the authors do evaluate the monolingual subspace induced by the model.[35]

Not all word embeddings learned from text are born equal. Depending on the ap-

[35] These embeddings are not publicly available and we were unable to re-train them using the source code.

plication, those learned by NMT models may have particularly desirable properties. For decades, distributional semantic models have aimed to exploit Firth's famous *distributional hypothesis* to induce word meanings from (monolingual) text. However, the hypothesis also betrays the weakness of the monolingual distributional approach when it comes to learning humah-quality concept representations. For while it is undeniable that "words which are similar in meaning appear in similar distributional contexts" [56], the converse assertion, which is what really matters, is only sometimes true.

Chapter 4

Learning to Represent Phrases

4.1 Introduction

Much recent research in computational semantics has focussed on learning representations of arbitrary-length phrases and sentences. This task is challenging partly because there is no obvious gold standard of phrasal representation that could be used in training and evaluation. Consequently, it is difficult to design approaches that could learn from such a gold standard, and also hard to evaluate or compare different models.

In this work, we use dictionary definitions to address this issue. The composed meaning of the words in a dictionary definition (*a tall, long-necked, spotted ruminant of Africa*) should correspond to the meaning of the word they define (*giraffe*). This bridge between lexical and phrasal semantics is useful because high quality vector representations of single words can be used as a target when learning to combine the words into a coherent phrasal representation.

This approach still requires a model capable of learning to map between arbitrary-length phrases and fixed-length continuous-valued word vectors. For this purpose we experiment with two broad classes of neural language models (NLMs): Recurrent Neural Networks (RNNs), which naturally encode the order of input words, and simpler (feedforward) bag-of-words (BOW) embedding models. Prior to training these NLMs, we learn target lexical representations by training the Word2Vec software [121] on billions of words of raw text.

We demonstrate the usefulness of our approach by building and releasing two applications. The first is a *reverse dictionary* or *concept finder*: a system that returns words based on user descriptions or definitions [182]. Reverse dictionaries are used by copywriters, novelists, translators and other professional writers to find words for notions or ideas that might be on the tip of their tongue. For instance, a travel-writer might look to enhance her prose by searching for examples of a *country that people associate with warm weather* or *an activity that is mentally or physically demanding*. We show that an NLM-based reverse dictionary trained on only a handful of dictionaries identifies novel definitions and concept descriptions comparably or better than commercial systems, which rely on significant task-specific engineering and access to much more dictionary data. Moreover, by exploiting models that learn bilingual word representations [172, 95, 73, 63], we show that the NLM approach can be easily extended to produce a potentially useful cross-lingual reverse dictionary.

The second application of our models is as a general-knowledge crossword question answerer. When trained on both dictionary definitions and the opening sentences of Wikipedia articles, NLMs produce plausible answers to (non-cryptic) crossword clues, even those that apparently require detailed world knowledge. Both BOW and RNN models can outperform bespoke commercial crossword solvers, particularly when clues contain a greater number of words. Qualitative analysis reveals that NLMs can learn to relate concepts that are not directly connected in the training data and can thus generalise well to unseen input. To facilitate further research, all of our code, training and evaluation sets (together with a system demo) are published online with this paper.[1]

4.2 Neural Language Model Architectures

The first model we apply to the dictionary-based learning task is a recurrent neural network (RNN). RNNs operate on variable-length sequences of inputs; in our case, natural language definitions, descriptions or sentences. RNNs (with LSTMs) have achieved state-of-the-art performance in language modelling [119], image caption generation [93] and approach state-of-the-art performance in machine translation [7].

During training, the input to the RNN is a dictionary definition or sentence from an encyclopedia. The objective of the model is to map these defining phrases or sentences to an embedding of the word that the definition defines. The target word embeddings are learned independently of the RNN weights, using the Word2Vec software [121].

The set of all words in the training data constitutes the vocabulary of the RNN.

[1] https://www.cl.cam.ac.uk/~fh295/

For each word in this vocabulary we randomly initialise a real-valued vector (input embedding) of model parameters. The RNN 'reads' the first word in the input by applying a non-linear projection of its embedding v_1 parameterised by input weight matrix W and b, a vector of biases.

$$A_1 = \phi(Wv_1 + b)$$

yielding the first internal activation state A_1. In our implementation, we use $\phi(x) = \tanh(x)$, though in theory ϕ can be any differentiable non-linear function. Subsequent internal activations (after time-step t) are computed by projecting the embedding of the t^{th} word and using this information to 'update' the internal activation state.

$$A_t = \phi(UA_{t-1} + Wv_t + b).$$

As such, the values of the final internal activation state units A_N are a weighted function of all input word embeddings, and constitute a 'summary' of the information in the sentence.

4.2.1 Long Short Term Memory

A known limitation when training RNNs to read language using gradient descent is that the error signal (gradient) on the training examples either vanishes or explodes as the number of time steps (sentence length) increases [18]. Consequently, after reading longer sentences the final internal activation A_N typically retains useful information about the most recently read (sentence-final) words, but can neglect important information near the start of the input sentence. LSTMs [81] were designed to mitigate this long-term dependency problem.

At each time step t, in place of the single internal layer of units A, the LSTM RNN computes six internal layers i^w, g^i, g^f, g^o, h and m. The first, g^w, represents the core information passed to the LSTM unit by the latest input word at t. It is computed as a simple linear projection of the input embedding v_t (by input weights W_w) and the *output state* of the LSTM at the previous time step h_{t-1} (by update weights U_w):

$$i^w_t = W_w v_t + U_w h_{t-1} + b_w$$

The layers g^i, g^f and g^o are computed as weighted sigmoid functions of the input embeddings, again parameterised by layer-specific weight matrices W and U:

$$g_t^s = \frac{1}{1 + \exp(-(W_s v_t + U_s h_{t-1} + b_s))}$$

where s stands for one of i, f or o. These vectors take values on $[0, 1]$ and are often referred to as *gating activations*. Finally, the *internal memory state*, m_t and new output state h_t, of the LSTM at t are computed as

$$\begin{aligned} m_t =& i_t^w \odot g_t^i + m_{t-1} \odot g_t^f \\ h_t =& g_t^o \odot \phi(m_t), \end{aligned}$$

where $\odot$ indicates elementwise vector multiplication and ϕ is, as before, some non-linear function (we use $tanh$). Thus, g^i determines to what extent the new *input* word is considered at each time step, g^f determines to what extent the existing state of the internal memory is retained or *forgotten* in computing the new internal memory, and g^o determines how much this memory is considered when computing the output state at t.

The sentence-final memory state of the LSTM, m_N, a 'summary' of all the information in the sentence, is then projected via an extra non-linear projection (parameterised by a further weight matrix) to a target embedding space. This layer enables the target (defined) word embedding space to take a different dimension to the activation layers of the RNN, and in principle enables a more complex definition-reading function to be learned.

4.2.2 Bag-of-Words NLMs

We implement a simpler linear bag-of-words (BOW) architecture for encoding the definition phrases. As with the RNN, this architecture learns an embedding v_i for each word in the model vocabulary, together with a single matrix of input projection weights W. The BOW model simply maps an input definition with word embeddings $v_1 \dots v_n$ to the sum of the projected embeddings $\sum_{i=1}^n W v_i$. This model can also be considered a special case of an RNN in which the update function U and nonlinearity ϕ are both the identity, so that 'reading' the next word in the input phrase updates the current representation more simply:

$$A_t = A_{t-1} + Wv_t.$$

4.2.3 Pre-trained Input Representations

We experiment with variants of these models in which the input definition embeddings are pre-learned and fixed (rather than randomly-initialised and updated) during training. There are several potential advantages to taking this approach. First, the word embeddings are trained on massive corpora and may therefore introduce additional linguistic or conceptual knowledge to the models. Second, at test time, the models will have a larger effective vocabulary, since the pre-trained word embeddings typically span a larger vocabulary than the union of all dictionary definitions used to train the model. Finally, the models will then map to and from the same space of embeddings (the embedding space will be closed under the operation of the model), so conceivably could be more easily applied as a general-purpose 'composition engine'.

4.2.4 Training Objective

We train all neural language models M to map the input definition phrase s_c defining word c to a location close to the the pre-trained embedding v_c of c. We experiment with two different cost functions for the word-phrase pair (c, s_c) from the training data. The first is simply the cosine distance between $M(s_c)$ and v_c. The second is the rank loss

$$\max(0, m - cos(M(s_c), v_c) - \cos(M(s_c), v_r))$$

where v_r is the embedding of a randomly-selected word from the vocabulary other than c. This loss function was used for language models, for example, in [83]. In all experiments we apply a margin $m = 0.1$, which has been shown to work well on word-retrieval tasks [27].

4.2.5 Implementation Details

Since training on the dictionary data took 6-10 hours, we did not conduct a hyper-parameter search on any validation sets over the space of possible model configurations such as embedding dimension, or size of hidden layers. Instead, we chose these parameters to be as standard as possible based on previous research. For fair comparison, any aspects of model design that are not specific to a particular class of model were kept constant across experiments.

The pre-trained word embeddings used in all of our models (either as input or target) were learned by a continuous bag-of-words (CBOW) model using the Word2Vec software on approximately 8 billion words of running text.[2] When training such models on massive corpora, a large embedding length of up to 700 have been shown to yield best performance (see e.g. [50]). The pre-trained embeddings used in our models were of length 500, as a compromise between quality and memory constraints.

In cases where the word embeddings are learned during training on the dictionary objective, we make these embeddings shorter (256), since they must be learned from much less language data. In the RNN models, and at each time step each of the four LSTM RNN internal layers (gating and activation states) had length 512 – another standard choice (see e.g. [36]). The final hidden state was mapped linearly to length 500, the dimension of the target embedding. In the BOW models, the projection matrix projects input embeddings (either learned, of length 256, or pre-trained, of length 500) to length 500 for summing.

All models were implemented with Theano [20] and trained with minibatch SGD on GPUs. The batch size was fixed at 16 and the learning rate was controlled by *adadelta* [181].

4.3 Reverse Dictionaries

The most immediate application of our trained models is as a *reverse dictionary* or *concept finder*. It is simple to look up a definition in a dictionary given a word, but professional writers often also require suitable words for a given idea, concept or definition.[3] Reverse dictionaries satisfy this need by returning candidate words given a phrase, description or definition. For instance, when queried with the phrase *an activity that requires strength and determination*, the OneLook.com reverse dictionary returns the concepts *exercise* and *work*. Our trained RNN model can perform a similar function, simply by mapping a phrase to a point in the target (Word2Vec) embedding

[2] The Word2Vec embedding models are well known; further details can be found at `https://code.google.com/p/word2vec/` The training data for this pre-training was compiled from various online text sources using the script *demo-train-big-model-v1.sh* from the same page.

[3] See the testimony from professional writers at `http://www.onelook.com/?c=awards`

space, and returning the words corresponding to the embeddings that are closest to that point.

Several other academic studies have proposed reverse dictionary models. These generally rely on common techniques from information retrieval, comparing definitions in their internal database to the input query, and returning the word whose definition is ‘closest’ to that query [23, 24, 182]. Proximity is quantified differently in each case, but is generally a function of hand-engineered features of the two sentences. For instance, [153] propose a method in which the candidates for a given input query are all words in the model’s database whose definitions contain one or more words from the query. This candidate list is then ranked according to a query-definition similarity metric based on the hypernym and hyponym relations in WordNet, features commonly used in IR such as *tf-idf* and a parser.

There are, in addition, at least two commercial online reverse dictionary applications, whose architecture is proprietary knowledge. The first is the Dictionary.com reverse dictionary [4], which retrieves candidate words from the Dictionary.com dictionary based on user definitions or descriptions. The second is **OneLook.com**, whose algorithm searches 1061 indexed dictionaries, including all major freely-available online dictionaries and resources such as Wikipedia and WordNet.

4.3.1 Data Collection and Training

To compile a bank of dictionary definitions for training the model, we started with all words in the target embedding space. For each of these words, we extracted dictionary-style definitions from five electronic resources: *Wordnet, The American Heritage Dictionary, The Collaborative International Dictionary of English, Wiktionary* and *Webster’s*. We chose these five dictionaries because they are freely-available via the WordNik API,[5] but in theory any dictionary could be chosen. Most words in our training data had multiple definitions. For each word w with definitions $\{d_1 \dots d_n\}$ we included all pairs $(w, d_1) \dots (w, d_n)$ as training examples.

To allow models access to more factual knowledge than might be present in a dictionary (for instance, information about specific entities, places or people, we supplemented this training data with information extracted from Simple Wikipedia. [6] For

[4] Available at http://dictionary.reference.com/reverse/

[5] See http://developer.wordnik.com

[6] https://simple.wikipedia.org/wiki/Main_Page

every word in the model's target embedding space that is also the title of a Wikipedia article, we treat the sentences in the first paragraph of the article as if they were (independent) definitions of that word. When a word in Wikipedia also occurs in one (or more) of the five training dictionaries, we simply add these pseudo-definitions to the training set of definitions for the word. Combining Wikipedia and dictionaries in this way resulted in $\approx 900,000$ word-'definition' pairs of $\approx 100,000$ unique words.

To explore the effect of the quantity of training data on the performance of the models, we also trained models on subsets of this data. The first subset comprised only definitions from Wordnet (approximately 150,000 definitions of 75,000 words). The second subset comprised only words in Wordnet and their *first* definitions (approximately 75,000 word, definition pairs).[7]. For all variants of RNN and BOW models, however, reducing the training data in this way resulted in a clear reduction in performance on all tasks. For brevity, we therefore do not present these results in what follows.

		Dictionary definitions								
Test Set		**Seen** (500 WN defs)			**Unseen** (500 WN defs)			**Concept descriptions** (200)		
Unsup. models	W2V add	-	-	-	923	.04/.16	163	339	.07/.30	150
	W2V mult	-	-	-	1000	.00/.00	10*	1000	.00/.00	27*
	OneLook	**0**	**.89/.91**	**67**	-	-	-	**18.5**	**.38**/.58	153
NLMs	RNN cosine	12	.48/.73	103	22	.41/.70	116	69	.28/.54	157
	RNN w2v cosine	19	.44/.70	111	19	.44/.69	126	26	**.38**/.66	111
	RNN ranking	18	.45/.67	128	24	.43/.69	103	25	.34/.66	102
	RNN w2v ranking	54	.32/.56	155	33	.36/.65	137	30	.33/.69	**77**
	BOW cosine	22	.44/.65	129	19	.43/.69	103	50	.34/.60	99
	BOW w2v cosine	15	.46/.71	124	**14**	**.46/ .71**	104	28	.36/.66	99
	BOW ranking	17	.45/.68	115	22	.42/.70	**95**	32	.35/.69	101
	BOW w2v rankng	55	.32/.56	155	36	.35/.66	138	38	.33/**.72**	85
					median rank	*accuracy@10/100*	*rank variance*			

Table 4.1: Performance of different reverse dictionary models in different evaluation settings. *Low variance in *mult* models is due to consistently poor scores, so not highlighted.

4.3.2 Comparisons

As a baseline, we also implemented two entirely unsupervised methods using the neural (Word2Vec) word embeddings from the target word space. In the first (**W2V add**),

[7] As with other dictionaries, the first definition in WordNet generally corresponds to the most typical or common sense of a word.

we compose the embeddings for each word in the input query by pointwise addition, and return as candidates the nearest word embeddings to the resulting composed vector.[8] The second baseline, (**W2V mult**), is identical except that the embeddings are composed by elementwise multiplication. Both methods are established ways of building phrase representations from word embeddings [124].

None of the models or evaluations from previous academic research on reverse dictionaries is publicly available, so direct comparison is not possible. However, we do compare performance with the commercial systems. The Dictionary.com system returned no candidates for over 96% of our input definitions. We therefore conduct detailed comparison with OneLook.com, which is the first reverse dictionary tool returned by a Google search and seems to be the most popular among writers.

4.3.3 Reverse Dictionary Evaluation

To our knowledge there are no established means of measuring reverse dictionary performance. In the only previous academic research on English reverse dictionaries that we are aware of, evaluation was conducted on 300 word-definition pairs written by lexicographers [153]. Since these are not publicly available we developed new evaluation sets and make them freely available for future evaluations.

The evaluation items are of three types, designed to test different properties of the models. To create the **seen** evaluation, we randomly selected 500 words from the WordNet training data (seen by all models), and then randomly selected a definition for each word. Testing models on the resulting 500 word-definition pairs assesses their ability to recall or decode previously encoded information. For the **unseen** evaluation, we randomly selected 500 words from WordNet and excluded all definitions of these words from the training data of all models.

Finally, for a fair comparison with OneLook, which has both the seen and unseen pairs in its internal database, we built a new dataset of **concept descriptions** that do not appear in the training data for any model. To do so, we randomly selected 200 adjectives, nouns or verbs from among the top 3000 most frequent tokens in the British National Corpus [99] (but outside the top 100). We then asked ten native English speakers to write a single-sentence ‘description’ of these words. To ensure the resulting descriptions were good quality, for each description we asked two participants

[8] Since we retrieve all answers from embedding spaces by cosine similarity, addition of word embeddings is equivalent to taking the mean.

who did not produce that description to list any words that fitted the description (up to a maximum of three). If the target word was not produced by one of the two checkers, the original participant was asked to re-write the description until the validation was passed.[9] These concept descriptions, together with other evaluation sets, can be downloaded from our website for future comparisons.

Test set	Word	Description
Dictionary definition	*valve*	”control consisting of a mechanical device for controlling fluid flow”
Concept description	*prefer*	”when you like one thing more than another thing”

Table 4.2: Style difference between *dictionary definitions* and *concept descriptions* in the evaluation.

Given a test description, definition, or question, all models produce a ranking of possible word answers based on the proximity of their representations of the input phrase and all possible output words. To quantify the quality of a given ranking, we report three statistics: the *median rank* of the correct answer (over the whole test set, lower better), the proportion of training cases in which the correct answer appears in the top 10/100 in this ranking (*accuracy@10/100* - higher better) and the variance of the rank of the correct answer across the test set (*rank variance* - lower better).

4.3.4 Results

Table 4.1 shows the performance of the different models in the three evaluation settings. Of the unsupervised composition models, elementwise addition is clearly more effective than multiplication, which almost never returns the correct word as the nearest neighbour of the composition. Overall, however, the supervised models (RNN, BOW and OneLook) clearly outperform these baselines.

The results indicate interesting differences between the NLMs and the OneLook dictionary search engine. The Seen (WN first) definitions in Table 4.1 occur in both the training data for the NLMs and the lookup data for the OneLook model. Clearly the OneLook algorithm is better than NLMs at retrieving already available information (returning 89% of correct words among the top-ten candidates on this set). However, this is likely to come at the cost of a greater memory footprint, since the model requires access to its database of dictionaries at query time.[10]

[9] Re-writing was required in 6 of the 200 cases.

[10] The trained neural language models are approximately half the size of the six training dictionar-

The performance of the NLM embedding models on the (unseen) concept descriptions task shows that these models can generalise well to novel, unseen queries. While the median rank for OneLook on this evaluation is lower, the NLMs retrieve the correct answer in the top ten candidates approximately as frequently, within the top 100 candidates more frequently and with lower variance in ranking over the test set. Thus, NLMs seem to generalise more 'consistenly' than OneLook on this dataset, in that they generally assign a reasonably high ranking to the correct word. In contrast, as can also be verified by querying our we demo, OneLook tends to perform either very well or poorly on a given query.[11]

When comparing between NLMs, perhaps the most striking observation is that the RNN models do not significantly outperform the BOW models, even though the BOW model output is invariant to changes in the order of words in the definition. Users of the online demo can verify that the BOW models recover concepts from descriptions strikingly well, even when the words in the description are permuted. This observation underlines the importance of lexical semantics in the interpretation of language by NLMs, and is consistent with some other recent work on embedding sentences [85].

It is difficult to observe clear trends in the differences between NLMs that learn input word embeddings and those with pre-trained (Word2Vec) input embeddings. Both types of input yield good performance in some situations and weaker performance in others. In general, pre-training input embeddings seems to help most on the concept descriptions, which are furthest from the training data in terms of linguistic style. This is perhaps unsurprising, since models that learn input embeddings from the dictionary data acquire all of their conceptual knowledge from this data (and thus may overfit to this setting), whereas models with pre-trained embeddings have some semantic memory acquired from general running-text language data and other knowledge acquired from the dictionaries.

4.3.5 Qualitative Analysis

Some example output from the various models is presented in Table 4.3. The differences illustrated here are also evident from querying the web demo. The first example shows how the NLMs (BOW and RNN) generalise beyond their training data. Four of

ies stored as plain text, so would be hundreds of times smaller than the OneLook database of 1061 dictionaries if stored this way.

[11] We also observed that the *mean* ranking for NLMs was lower than for OneLook on the concept descriptions task.

Input Description	OneLook	W2V add	RNN	BOW
”a native of a cold country”	1:*country* 2:*citizen* 3:*foreign* 4:*naturalize* 5:*cisco*	1:*a* 2.*the* 3:*another* 4:*of* 5:*whole*	1:*eskimo* 2:*scandinavian* 3:*arctic* 4:*indian* 5:*siberian*	1:*frigid* 2:*cold* 3:*icy* 4:*russian* 5:*indian*
”a way of moving through the air”	1:*drag* 2:*whiz* 3:*aerodynamics* 4:*draught* 5:*coefficient of drag*	1:*the* 2:*through* 3:*a* 4:*moving* 5:*in*	1:*glide* 2:*scooting* 3:*glides* 4:*gliding* 5:*flight*	1:*flying* 2:*gliding* 3:*glide* 4:*fly* 5:*scooting*
”a habit that might annoy your spouse”	1:*sisterinlaw* 2:*fatherinlaw* 3:*motherinlaw* 4:*stepson* 5:*stepchild*	1:*annoy* 2:*your* 3:*might* 4:*that* 5:*either*	1:*bossiness* 2:*jealousy* 3:*annoyance* 4:*rudeness* 5:*boorishness*	1:*infidelity* 2:*bossiness* 3:*foible* 4:*unfaithfulnes* 5:*adulterous*

Table 4.3: The top-five candidates for example queries (invented by the authors) from different reverse dictionary models. Both the RNN and BOW models are without Word2Vec input and use the cosine loss.

the top five responses could be classed as appropriate in that they refer to inhabitants of cold countries. However, inspecting the WordNik training data, there is no mention of *cold* or anything to do with climate in the definitions of *Eskimo*, *Scandinavian*, *Scandinavia* etc. Therefore, the embedding models must have learned that *coldness* is a characteristic of Scandinavia, Siberia, Russia, relates to Eskimos etc. via connections with other concepts that are described or defined as *cold*. In contrast, the candidates produced by the OneLook and (unsupervised) W2V baseline models have nothing to do with coldness.

The second example demonstrates how the NLMs generally return candidates whose linguistic or conceptual function is appropriate to the query. For a query referring explicitly to a means, method or process, the RNN and BOW models produce verbs in different forms or an appropriate deverbal noun. In contrast, OneLook returns words of all types (*aerodynamics, draught*) that are arbitrarily related to the words in the query. A similar effect is apparent in the third example. While the candidates produced by the OneLook model are the correct part of speech (Noun), and related to the query topic, they are not semantically appropriate. The dictionary embedding models are the only ones that return a list of plausible *habits*, the class of noun requested by the input.

Input description	RNN EN-FR	W2V add	RNN + Google
”an emotion that you might feel after being rejected”	*triste, pitoyable* *répugnante, épouvantable*	*insister, effectivement* *pourquoi, nous*	*sentiment, regrett* *peur, aversion*
”a small black flying insect that transmits disease and likes horses”	*mouche, canard* *hirondelle, pigeon*	*attentivement, pouvions* *pourrons, naturellement*	*voler, faucon* *mouches, volant*

Table 4.4: Responses from cross-lingual reverse dictionary models to selected queries. Underlined responses are ‘correct’ or potentially useful for a native French speaker.

4.3.6 Cross-Lingual Reverse Dictionaries

We now show how the RNN architecture can be easily modified to create a *bilingual reverse dictionary* - a system that returns candidate words in one language given a description or definition in another. A bilingual reverse dictionary could have clear applications for translators or transcribers. Indeed, the problem of attaching appropriate words to concepts may be more common when searching for words in a second language than in a monolingual context.

To create the bilingual variant, we simply replace the Word2Vec target embeddings with those from a bilingual embedding space. Bilingual embedding models use bilingual corpora to learn a space of representations of the words in two languages, such that words from either language that have similar meanings are close together [73, 33, 63]. For a test-of-concept experiment, we used English-French embeddings learned by the state-of-the-art BilBOWA model [63] from the Wikipedia (monolingual) and Europarl (bilingual) corpora.[12] We trained the RNN model to map from English definitions to English words in the bilingual space. At test time, after reading an English definition, we then simply return the nearest French word neighbours to that definition.

Because no benchmarks exist for quantitative evaluation of bilingual reverse dictionaries, we compare this approach qualitatively with two alternative methods for mapping definitions to words across languages. The first is analogous to the W2V Add model of the previous section: in the bilingual embedding space, we first compose the embeddings of the English words in the query definition with elementwise addition, and then return the French word whose embedding is nearest to this vector sum. The second uses the RNN monolingual reverse dictionary model to identify an English word from an English definition, and then translates that word using Google Translate.

Table 4.4 shows that the RNN model can be effectively modified to create a cross-

[12] The approach should work with any bilingual embeddings. We thank Stephan Gouws for doing the training.

lingual reverse dictionary. It is perhaps unsurprising that the W2V Add model candidates are generally the lowest in quality given the performance of the method in the monolingual setting. In comparing the two RNN-based methods, the RNN (embedding space) model appears to have two advantages over the RNN + Google approach. First, it does not require online access to a bilingual word-word mapping as defined e.g. by Google Translate. Second, it less prone to errors caused by word sense ambiguity. For example, in response to the query *an emotion you feel after being rejected*, the bilingual embedding RNN returns emotions or adjectives describing mental states. In contrast, the monolingual+Google model incorrectly maps the plausible English response *regret* to the verbal infinitive *regretter*. The model makes the same error when responding to a description of a fly, returning the verb *voler* (to fly).

4.3.7 Discussion

We have shown that simply training RNN or BOW NLMs on six dictionaries yields a reverse dictionary that performs comparably to the leading commercial system, even with access to much less dictionary data. Indeed, the embedding models consistently return syntactically and semantically plausible responses, which are generally part of a more coherent and homogeneous set of candidates than those produced by the commercial systems. We also showed how the architecture can be easily extended to produce bilingual versions of the same model.

In the analyses performed thus far, we only test the dictionary embedding approach on tasks that it was trained to accomplish (mapping definitions or descriptions to words). In the next section, we explore whether the knowledge learned by dictionary embedding models can be effectively transferred to a novel task.

4.4 General Knowledge (crossword) Question Answering

The automatic answering of questions posed in natural language is a central problem of Artificial Intelligence. Although web search and IR techniques provide a means to find sites or documents related to language queries, at present, internet users requiring a specific fact must still sift through pages to locate the desired information.

Systems that attempt to overcome this, via fully open-domain or general knowledge question-answering (open QA), generally require large teams of researchers, modular design and powerful infrastructure, exemplified by IBM's Watson [54]. For this reason, much academic research focuses on settings in which the scope of the task is reduced. This has been achieved by restricting questions to a specific topic or domain [126], allowing systems access to pre-specified passages of text from which the answer can be inferred [84, 173], or centering both questions and answers on a particular knowledge base [19, 26].

In what follows, we show that the dictionary embedding models introduced in the previous sections may form a useful component of an open QA system. Given the absence of a knowledge base or web-scale information in our architecture, we narrow the scope of the task by focusing on general knowledge crossword questions. General knowledge (non-cryptic, or quick) crosswords appear in national newspapers in many countries. Crossword question answering is more tractable than general open QA for two reasons. First, models know the length of the correct answer (in letters), reducing the search space. Second, some crossword questions mirror definitions, in that they refer to fundamental properties of concepts (*a twelve-sided shape*) or request a category member (*a city in Egypt*).[13]

4.4.1 Evaluation

General Knowledge crossword questions come in different styles and forms. We used the Eddie James crossword website to compile a bank of sentence-like general-knowledge questions.[14] Eddie James is one of the UK's leading crossword compilers, working for several national newspapers. Our **long** question set consists of the first 150 questions (starting from puzzle #1) from his general-knowledge crosswords, excluding clues of fewer than four words and those whose answer was not a single word (e.g. *kingjames*).

To evaluate models on a different type of clue, we also compiled a set of **short**er questions based on the Guardian Quick Crossword. Guardian questions still require general factual or linguistic knowledge, but are generally shorter and somewhat more cryptic than the longer Eddie James clues. We again formed a list of 150 questions, beginning on 1 January 2015 and excluding any questions with multiple-word answers.

[13] As our interest is in the language understanding, we do not address the question of fitting answers into a grid, which is the main concern of end-to-end automated crossword solvers [107].

[14] `http://www.eddiejames.co.uk/`

For clear contrast, we excluded those few questions of length greater than four words. Of these 150 clues, a subset of 30 were **single-word** clues. All evaluation datasets are available online with the paper.

As with the reverse dictionary experiments, candidates are extracted from models by inputting definitions and returning words corresponding to the closest embeddings in the target space. In this case, however, we only consider candidate words *whose length matches the length specified in the clue*.

Test set	**Word**	**Description**
Long (150)	*Baudelaire*	”French poet and key figure in the development of Symbolism.”
Short (120)	*satanist*	”devil devotee”
Single-Word (30)	*guilt*	”culpability”

Table 4.5: Examples of the different question types in the crossword question evaluation dataset.

Question Type	*avg rank -accuracy@10/100 - rank variance*								
	Long (150)			**Short (120)**			**Single-Word (30)**		
One Across		.39 /			**.68 /**			.70 /	
Crossword Maestro		.27 /			.43 /			.73 /	
W2V add	42	.31/.63	92	11	.50/.78	66	**2**	**.79/.90**	45
RNN cosine	15	.43/.69	108	22	.39/.67	117	72	.31/.52	187
RNN w2v cosine	4	.61/.82	60	**7**	.56/.79	60	12	.48/.72	116
RNN ranking	6	.58/.84	**48**	10	.51/.73	57	12	.48/.69	67
RNN w2v ranking	**3**	.62/.80	61	8	.57/.78	49	12	.48/.69	114
BOW cosine	4	.60/.82	54	**7**	.56/.78	51	12	.45/.72	137
BOW w2v cosine	4	.60/.83	56	**7**	.54/.80	48	3	.59/.79	111
BOW ranking	5	**.62/.87**	50	8	.58/**.83**	37	8	.55/.79	**39**
BOW w2v ranking	5	.60/.86	**48**	8	.56/.83	**35**	4	.55/.83	43

Table 4.6: Performance of different models on crossword questions of different length. The two commercial systems are evaluated via their web interface so only accuracy@10 can be reported in those cases.

4.4.2 Benchmarks and Comparisons

As with the reverse dictionary experiments, we compare RNN and BOW NLMs with a simple unsupervised baseline of elementwise addition of Word2Vec vectors in the embedding space (we discard the ineffective *W2V mult* baseline), again restricting candidates to words of the pre-specified length. We also compare to two bespoke online crossword-solving engines. The first, One Across (`http://www.oneacross.com/`) is the candidate generation module of the award-winning *Proverb* crossword system [107]. Proverb, which was produced by academic researchers, has featured in national media such as New Scientist, and beaten expert humans in crossword solving tournaments. The second comparison is with Crossword Maestro (`http://www.crosswordmaestro.com/`), a commercial crossword solving system that handles both cryptic and non-cryptic crossword clues (we focus only on the non-cryptic setting), and has also been featured in national media.[15] We are unable to compare against a third well-known automatic crossword solver, *Dr Fill* [61], because code for Dr Fill's candidate-generation module is not readily available. As with the RNN and baseline models, when evaluating existing systems we discard candidates whose length does not match the length specified in the clue.

Certain principles connect the design of the existing commercial systems and differentiate them from our approach. Unlike the NLMs, they each require query-time access to large databases containing common crossword clues, dictionary definitions, the frequency with which words typically appear as crossword solutions and other hand-engineered and task-specific components [107, 61].

4.4.3 Results

The performance of models on the various question types is presented in Table 4.6. When evaluating the two commercial systems, One Across and Crossword Maestro, we have access to web interfaces that return up to approximately 100 candidates for each query, so can only reliably record membership of the top ten (accuracy@10).

On the long questions, we observe a clear advantage for all dictionary embedding models over the commercial systems and the simple unsupervised baseline. Here, the best performing NLM (RNN with Word2Vec input embeddings and ranking loss) ranks the correct answer third on average, and in the top-ten candidates over 60% of the time.

As the questions get shorter, the advantage of the embedding models diminishes. Both the unsupervised baseline and One Across answer the short questions with comparable accuracy to the RNN and BOW models. One reason for this may be the dif-

[15] See e.g. `http://www.theguardian.com/crosswords/crossword-blog/2012/mar/08/crossword-blog-computers-crack-cryptic-clues`

Input Description	One Across	Crossword Maestro	BOW	RNN
”Swiss mountain peak famed for its north face (5)”	1:*noted* 2:*front* 3:**Eiger** 4:*crown* 5:*fount*	1:*after* 2:*favor* 3:*ahead* 4:*along* 5:*being*	1:**Eiger** 2.*Crags* 3:*Teton* 4:*Cerro* 5:*Jebel*	1:**Eiger** 2:*Aosta* 3:*Cuneo* 4:*Lecco* 5:*Tyrol*
”Old Testament successor to Moses (6)”	1:**Joshua** 2:*Exodus* 3:*Hebrew* 4:*person* 5:*across*	1:*devise* 2:*Daniel* 3:*Haggai* 4: *Isaiah* 5:*Joseph*	1:*Isaiah* 2:*Elijah* 3:**Joshua** 4:*Elisha* 5:*Yahweh*	1:**Joshua** 2:*Isaiah* 3:*Gideon* 4:*Elijah* 5:*Yahweh*
”The former currency of the Netherlands (7)”	1:*Holland* 2:*general* 3:*Lesotho*	1:*Holland* 2:*ancient* 3:*earlier* 4:*onetime* 5:*qondam*	1:**Guilder** 2:*Holland* 3:*Drenthe* 4:*Utrecht* 5:*Naarden*	1:**Guilder** 2:*Escudos* 3:*Pesetas* 4:*Someren* 5:*Florins*
”Arnold, 20th Century composer pioneer of atonality (10)”	1:*surrealism* 2:*laborparty* 3:*tonemusics* 4:*introduced* 5:**Schoenberg**	1:*disharmony* 2:*dissonance* 3:*bringabout* 4:*constitute* 5:*triggeroff*	1:**Schoenberg** 2:*Christleib* 3:*Stravinsky* 4:*Elderfield* 5:*Mendelsohn*	1:*Mendelsohn* 2:*Williamson* 3:*Huddleston* 4:*Mandelbaum* 5:*Zimmerman*

Table 4.7: Responses from different models to example crossword clues. In each case the model output is filtered to exclude any candidates that are not of the same length as the correct answer. BOW and RNN models are trained without Word2Vec input embeddings and cosine loss.

ference in form and style between the shorter clues and the full definitions or encyclopedia sentences in the dictionary training data. As the length of the clue decreases, finding the answer often reduces to generating synonyms (*culpability* - *guilt*), or category members (*tall animal* - *giraffe*). The commercial systems can retrieve good candidates for such clues among their databases of entities, relationships and common crossword answers. Unsupervised Word2Vec representations are also known to encode these sorts of relationships (even after elementwise addition for short sequences of words) [121]. This would also explain why the dictionary embedding models with pre-trained (Word2Vec) input embeddings outperfom those with learned embeddings, particularly for the shortest questions.

4.4.4 Qualitative Analysis

A better understanding of how the different models arrive at their answers can be gained from considering specific examples, as presented in Table 4.7. The first three examples show that, despite the apparently superficial nature of its training data (definitions and introductory sentences) embedding models can answer questions that require factual knowledge about people and places. Another notable characteristic of these model is the consistent semantic appropriateness of the candidate set. In the first case, the top five candidates are all mountains, valleys or places in the Alps; in the second, they are all biblical names. In the third, the RNN model retrieves currencies, in this case performing better than the BOW model, which retrieves entities of various type associated with the Netherlands. Generally speaking (as can be observed by the web demo), the ‘smoothness’ or consistency in candidate generation of the dictionary embedding models is greater than that of the commercial systems. Despite its simplicity, the unsupervised W2V addition method is at times also surprisingly effective, as shown by the fact that it returns *Joshua* in its top candidates for the third query.

The final example in Table 4.7 illustrates the surprising power of the BOW model. In the training data there is a single definition for the correct answer *Schoenberg*: *United States composer and musical theorist (born in Austria) who developed atonal composition*. The only word common to both the query and the definition is ’composer’ (there is no tokenization that allows the BOW model to directly connect *atonal* and *atonality*). Nevertheless, the model is able to infer the necessary connections between the concepts in the query and the definition to return Schoenberg as the top candidate.

Despite such cases, it remains an open question whether, with more diverse training data, the world knowledge required for full open QA (e.g. secondary facts about *Schoenberg*, such as his family) could be encoded and retained as weights in a (larger) dynamic network, or whether it will be necessary to combine the RNN with an external memory that is less frequently (or never) updated. This latter approach has begun to achieve impressive results on certain QA and entailment tasks [26, 65, 173].

4.5 Conclusion

Dictionaries exist in many of the world’s languages. We have shown how these lexical resources can constitute valuable data for training the latest neural language models to interpret and represent the meaning of phrases and sentences. While humans use the phrasal definitions in dictionaries to better understand the meaning of words, machines can use the words to better understand the phrases. We used two dictionary embedding architectures - a recurrent neural network architecture with a long-short-term memory, and a simpler linear bag-of-words model - to explicitly exploit this idea.

On the reverse dictionary task that mirrors its training setting, NLMs that embed all known concepts in a continuous-valued vector space perform comparably to the best known commercial applications despite having access to many fewer definitions. Moreover, they generate smoother sets of candidates and require no linguistic preprocessing or task-specific engineering. We also showed how the description-to-word objective can be used to train models useful for other tasks. NLMs trained on the same data can answer general-knowledge crossword questions, and indeed outperform commercial systems on questions containing more than four words. While our QA experiments focused on crosswords, the results suggest that a similar embedding-based approach may ultimately lead to improved output from more general QA and dialog systems and information retrieval engines in general.

We make all code, training data, evaluation sets and both of our linguistic tools publicly available online for future research. In particular, we propose the reverse dictionary task as a comparatively general-purpose and objective way of evaluating how well models compose lexical meaning into phrase or sentence representations (whether or not they involve training on definitions directly).

In the next stage of this research, we will explore ways to enhance the NLMs described here, especially in the question-answering context. The models are currently not trained on any question-like language, and would conceivably improve on exposure to such linguistic forms. We would also like to understand better how BOW models can perform so well with no 'awareness' of word order, and whether there are specific linguistic contexts in which models like RNNs or others with the power to encode word order are indeed necessary. Finally, we intend to explore ways to endow the model with richer world knowledge. This may require the integration of an external memory module, similar to the promising approaches proposed in several recent papers [65, 173].

Chapter 5

Learning to Represent Sentences

5.0.1 Introduction

Distributed representations - dense real-valued vectors that encode the semantics of linguistic units - are ubiquitous in today's NLP research. For single-words or word-like entities, there are established ways to acquire such representations from naturally occurring (unlabelled) training data based on comparatively task-agnostic objectives (such as predicting adjacent words). These methods are well understood empirically [10] and theoretically [101]. The best word representation spaces reflect consistently-observed aspects of human conceptual organisation [80], and can be added as features to improve the performance of numerous language processing systems [41].

By contrast, there is comparatively little consensus on the best ways to learn distributed representations of phrases or sentences.[1] With the advent of deeper language processing techniques, it is relatively common for models to represent phrases or sentences as continuous-valued vectors. Examples include machine translation [161], image captioning [110] and dialogue systems [152]. While it has been observed informally that the internal sentence representations of such models can reflect semantic intuitions [36], it is not known which architectures or objectives yield the 'best' or most useful representations. Resolving this question could ultimately have a significant impact on language processing systems. Indeed, it is phrases and sentences, rather than individual words, that encode the human-like general world knowledge (or 'common sense') [131] that is a critical missing part of most current language understanding systems.

[1] See the contrasting conclusions in [123, 39, 9, 122] among others.

We address this issue with a systematic comparison of cutting-edge methods for learning distributed representations of sentences. We constrain our comparison to methods that do not require labelled data gathered for the purpose of training models, since such methods are more cost-effective and applicable across languages and domains. We also propose two new phrase or sentence representation learning objectives - *Sequential Denoising Autoencoders* (SDAEs) and *FastSent*, a sentence-level log-linear bag-of-words model. We compare all methods on two types of task - *supervised* and *unsupervised evaluations* - reflecting different ways in which representations are ultimately to be used. In the former setting, a classifier or regression model is applied to representations and trained with task-specific labelled data, while in the latter, representation spaces are directly queried using cosine distance.

We observe notable differences in approaches depending on the nature of the evaluation metric. In particular, deeper or more complex models (which require greater time and resources to train) generally perform best in the supervised setting, whereas shallow log-linear models work best on unsupervised benchmarks. Specifically, SkipThought Vectors [94] perform best on the majority of supervised evaluations, but SDAEs are the top performer on paraphrase identification. In contrast, on the (unsupervised) SICK sentence relatedness benchmark, FastSent, a simple, log-linear variant of the SkipThought objective, performs better than all other models. Interestingly, the method that exhibits strongest performance across both supervised and unsupervised benchmarks is a bag-of-words model trained to compose word embeddings using dictionary definitions [75]. Taken together, these findings constitute valuable guidelines for the application of phrasal or sentential representation-learning to language understanding systems.

5.0.2 Distributed Sentence Representations

To constrain the analysis, we compare neural language models that compute sentence representations from unlabelled, naturally-ocurring data, as with the predominant methods for word representations.[2] Likewise, we do not focus on 'bottom up' models where phrase or sentence representations are built from fixed mathematical operations on word vectors (although we do consider a canonical case - see CBOW below); these were already compared by [122]. Most space is devoted to our novel approaches, and

[2] This excludes innovative supervised sentence-level architectures including [157, 90] and many others.

we refer the reader to the original papers for more details of existing models.

5.0.3 Existing Models Trained on Text

SkipThought Vectors For consecutive sentences S_{i-1}, S_i, S_{i+1} in some document, the **SkipThought** model [94] is trained to predict target sentences S_{i-1} and S_{i+1} given source sentence S_i. As with all *sequence-to-sequence* models, in training the source sentence is 'encoded' by a Recurrent Neural Network (RNN) (with Gated Recurrent uUnits [36]) and then 'decoded' into the two target sentences in turn. Importantly, because RNNs employ a single set of update weights at each time-step, both the encoder and decoder are sensitive to the order of words in the source sentence.

For each position in a target sentence S_t, the decoder computes a softmax distribution over the model's vocabulary. The cost of a training example is the sum of the negative log-likelihood of each correct word in the target sentences S_{i-1} and S_{i+1}. This cost is backpropagated to train the encoder (and decoder), which, when trained, can map sequences of words to a single vector.

ParagraphVector [98] proposed two log-linear models of sentence representation. The **DBOW** model learns a vector s for every sentence S in the training corpus which, together with word embeddings v_w, define a softmax distribution optimised to predict words $w \in S$ given S. The v_w are shared across all sentences in the corpus. In the **DM** model, k-grams of consecutive words $\{w_i \dots w_{i+k} \in S\}$ are selected and s is combined with $\{v_{w_i} \dots v_{w_{i+k}}\}$ to make a softmax prediction (parameterised by additional weights) of w_{i+k+1}.

We used the Gensim implementation,[3] treating each sentence in the training data as a 'paragraph' as suggested by the authors. During training, both DM and DBOW models store representations for every sentence (as well as word) in the training corpus. Even on large servers it was therefore only possible to train models with representation size 200, and DM models whose combination operation was averaging (rather than concatenation).

Bottom-Up Methods We train **CBOW** and **SkipGram** word embeddings [121] on the Books corpus, and compose by elementwise addition as proposed by [124].[4]

We also compare to **C-PHRASE** [138], an approach that exploits a (supervised) parser to infer distributed semantic representations based on a syntactic parse of sen-

[3] `https://radimrehurek.com/gensim/`

[4] We also tried multiplication but this gave very poor results.

tences. C-PHRASE achieves state-of-the-art results for distributed representations on several evaluations used in this study.[5]

Non-Distributed Baseline We implement a **TFIDF BOW** model in which the representation of sentence S encodes the count in S of a set of feature-words weighted by their *tfidf* in C, the corpus. The feature-words are the 200,000 most common words in C.

5.0.4 Models Trained on Structured Resources

The following models rely on (freely-available) data that has more structure than raw text.

DictRep [75] trained neural language models to map dictionary definitions to pre-trained word embeddings of the words defined by those definitions. They experimented with **BOW** and **RNN** (with LSTM) encoding architectures and variants in which the input word embeddings were either learned or pre-trained (**+embs.**) to match the target word embeddings. We implement their models using the available code and training data.[6]

CaptionRep Using the same overall architecture, we trained (**BOW** and **RNN**) models to map captions in the COCO dataset [35] to pre-trained vector representations of images. The image representations were encoded by a deep convolutional network [162] trained on the ILSVRC 2014 object recognition task [151]. Multi-modal distributed representations can be encoded by feeding test sentences forward through the trained model.

NMT We consider the sentence representations learned by neural MT models. These models have identical architecture to SkipThought, but are trained on sentence-aligned translated texts. We used a standard architecture [36] on all available **En-Fr** and **En-De** data from the 2015 Workshop on Statistical MT (WMT).[7]

[5] Since code for C-PHRASE is not publicly-available we use the available pre-trained model (`http://clic.cimec.unitn.it/composes/cphrase-vectors.html`). Note this model is trained on 3× more text than others in this study.

[6] `https://www.cl.cam.ac.uk/~fh295/`. Definitions from the training data matching those in the WordNet STS 2014 evaluation (used in this study) were excluded.

[7] `www.statmt.org/wmt15/translation-task.html`

5.0.5 Novel Text-Based Models

We introduce two new approaches designed to address certain limitations with the existing models.

Sequential (Denoising) Autoencoders The SkipThought objective requires training text with a coherent inter-sentence narrative, making it problematic to port to domains such as social media or artificial language generated from symbolic knowledge. To avoid this restriction, we experiment with a representation-learning objective based on *denoising autoencoders* (DAEs). In a DAE, high-dimensional input data is corrupted according to some noise function, and the model is trained to recover the original data from the corrupted version. As a result of this process, DAEs learn to represent the data in terms of features that explain its important factors of variation [167]. Transforming data into DAE representations (as a 'pre-training' or initialisation step) gives more robust (supervised) classification performance in deep feedforward networks [168].

The original DAEs were feedforward nets applied to (image) data of fixed size. Here, we adapt the approach to variable-length sentences by means of a noise function $N(S|p_o, p_x)$, determined by free parameters $p_o, p_x \in [0, 1]$. First, for each word w in S, N deletes w with (independent) probability p_o. Then, for each non-overlapping bigram $w_i w_{i+1}$ in S, N swaps w_i and w_{i+1} with probability p_x. We then train the same LSTM-based encoder-decoder architecture as NMT, but with the denoising objective to predict (as target) the original source sentence S given a corrupted version $N(S|p_o, p_x)$ (as source). The trained model can then encode novel word sequences into distributed representations. We call this model the *Sequential Denoising Autoencoder* (**SDAE**). Note that, unlike SkipThought, SDAEs can be trained on sets of sentences in arbitrary order.

We label the case with no noise (i.e. $p_o = p_x = 0$ and $N \equiv id$) **SAE**. This setting matches the method applied to text classification tasks by [45]. The 'word dropout' effect when $p_o \geq 0$ has also been used as a regulariser for deep nets in supervised language tasks [85], and for large p_x the objective is similar to word-level 'debagging' [160]. For the SDAE, we tuned p_o, p_x on the validation set (see Section 5.0.9).[8] We also tried a variant (**+embs**) in which words are represented by (fixed) pre-trained embeddings.

FastSent The performance of SkipThought vectors shows that rich sentence seman-

[8] We searched $p_o, p_x \in \{0.1, 0.2, 0.3\}$ and observed best results with $p_o = p_x = 0.1$.

tics can be inferred from the content of adjacent sentences. The model could be said to exploit a type of *sentence-level Distributional Hypothesis* [69, 141]. Nevertheless, like many deep neural language models, SkipThought is very slow to train (see Table 5.1). FastSent is a simple additive (log-linear) sentence model designed to exploit the same signal, but at much lower computational expense. Given a BOW representation of some sentence in context, the model simply predicts adjacent sentences (also represented as BOW) .

More formally, FastSent learns a source u_w and target v_w embedding for each word in the model vocabulary. For a training example S_{i-1}, S_i, S_{i+1} of consecutive sentences, S_i is represented as the sum of its source embeddings $\mathbf{s_i} = \sum_{w \in S_i} u_w$. The cost of the example is then simply:

$$\sum_{w \in S_{i-1} \cup S_{i+1}} \phi(\mathbf{s_i}, v_w) \tag{5.1}$$

where $\phi(v_1, v_2)$ is the softmax function.

We also experiment with a variant (**+AE**) in which the encoded (source) representation must predict its own words as target in addition to those of adjacent sentences. Thus in FastSent+AE, (5.1) becomes

$$\sum_{w \in S_{i-1} \cup S_i \cup S_{i+1}} \phi(\mathbf{s_i}, v_w). \tag{5.2}$$

At test time the trained model (very quickly) encodes unseen word sequences into distributed representations with $\mathbf{s} = \sum_{w \in S} u_w$.

5.0.6 Training and Model Selection

Unless stated above, all models were trained on the Toronto Books Corpus,[9] which has the inter-sentential coherence required for SkipThought and FastSent. The corpus consists of 70m ordered sentences from over 7,000 books.

Specifications of the models are shown in Table 5.1. The log-linear models (SkipGram, CBOW, ParagraphVec and FastSent) were trained for one epoch on one CPU core. The representation dimension d for these models was found after tuning $d \in \{100, 200, 300, 400, 500\}$ on the validation set.[10] All other models were trained on one

[9] http://www.cs.toronto.edu/~mbweb/

[10] For ParagraphVec only $d \in \{100, 200\}$ was possible due to the high memory footprint.

	OS	R	WO	SD	WD	TR	TE
S(D)AE			✓	2400	100	72*	640
ParagraphVec				100	100	4	1130
CBOW				500	500	2	145
SkipThought	✓		✓	4800	620	336*	890
FastSent	✓			100	100	2	140
DictRep		✓	✓	500	256	24*	470
CaptionRep		✓	✓	500	256	24*	470
NMT		✓	✓	2400	512	72*	720

Table 5.1: Properties of models compared in this study **OS:** requires training corpus of sentences in order. **R:** requires structured resource for training. **WO:** encoder sensitive to word order. **SD:** dimension of sentence representation. **WD:** dimension of word representation. **TR:** approximate training time (hours) on the dataset in this paper. * indicates trained on GPU. **TE:** approximate time (s) taken to encode 0.5m sentences.

Dataset		Sentence 1	Sentence 2
STS 2014	News	*Mexico wishes to guarantee citizens' safety.*	*Mexico wishes to avoid more violence.*
	Forum	*The problem is simpler than that.*	*The problem is simple.*
	WordNet	*A social set or clique of friends.*	*An unofficial association of people or groups.*
	Twitter	*Taking Aim #Stopgunviolence #Congress #NRA*	*Obama, Gun Policy and the N.R.A.*
	Images	*A woman riding a brown horse.*	*A young girl riding a brown horse.*
	Headlines	*Iranians Vote in Presidential Election.*	*Keita Wins Mali Presidential Election.*
SICK (test+train)		*A lone biker is jumping in the air.*	*A man is jumping into a full pool.*

Table 5.2: Example sentence pairs and 'similarity' ratings from the unsupervised evaluations used in this study.

GPU. The S(D)AE models were trained for one epoch (≈ 8 days). The SkipThought model was trained for two weeks, covering just under one epoch.[11] For CaptionRep and DictRep, performance was monitored on held-out training data and training was stopped after 24 hours after a plateau in cost. The NMT models were trained for 72 hours.

5.0.7 Evaluating Sentence Representations

In previous work, distributed representations of language were evaluated either by measuring the effect of adding representations as features in some classification task - *supervised evaluation* [41, 118, 94] - or by comparing with human relatedness judgements - *unspervised evaluation* [75, 10, 102]. The former setting reflects a scenario in

[11] Downloaded from `https://github.com/ryankiros/skip-thoughts`

which representations are used to inject general knowledge (sometimes considered as *pre-training*) into a supervised model. The latter pertains to applications in which the sentence representation space is used for direct comparisons, lookup or retrieval. Here, we apply and compare both evaluation paradigms.

Data	Model	MSRP (Acc / F1)	MR	CR	SUBJ	MPQA	TREC
	SAE	74.3 / 81.7	62.6	68.0	86.1	76.8	80.2
	SAE+embs.	70.6 / 77.9	73.2	75.3	89.8	86.2	80.4
Unordered	SDAE	**76.4 / 83.4**	67.6	74.0	89.3	81.3	77.6
Sentences	SDAE+embs.	73.7 / 80.7	**74.6**	**78.0**	**90.8**	**86.9**	78.4
(Toronto Books:	ParagraphVec DBOW	72.9 / 81.1	60.2	66.9	76.3	70.7	59.4
70m sents,	ParagraphVec DM	73.6 / 81.9	61.5	68.6	76.4	78.1	55.8
0.9B words)	Skipgram	69.3 / 77.2	73.6	77.3	89.2	85.0	82.2
	CBOW	67.6 / 76.1	73.6	7730	89.1	85.0	82.2
	Unigram TFIDF	**73.6 / 81.7**	73.7	79.2	90.3	82.4	**85.0**
Ordered	SkipThought	**73.0 / 82.0**	**76.5**	**80.1**	**93.6**	**87.1**	**92.2**
Sentences	FastSent	72.2 / 80.3	70.8	78.4	88.7	80.6	76.8
(Toronto Books)	FastSent+AE	71.2 / 79.1	71.8	76.7	88.8	81.5	80.4
	NMT En to Fr	69.1 / 77.1	64.7	70.1	84.9	81.5	**82.8**
Other	NMT En to De	65.2 / 73.3	61.0	67.6	78.2	72.9	81.6
structured	CaptionRep BOW	73.6 / 81.9	61.9	69.3	77.4	70.8	72.2
data	CaptionRep RNN	72.6 / 81.1	55.0	64.9	64.9	71.0	62.4
resource	DictRep BOW	**73.7 / 81.6**	71.3	75.6	86.6	82.5	73.8
	DictRep BOW+embs.	68.4 / 76.8	**76.7**	**78.7**	**90.7**	**87.2**	81.0
	DictRep RNN	73.2 / 81.6	67.8	72.7	81.4	82.5	75.8
	DictRep RNN+embs.	66.8 / 76.0	72.5	73.5	85.6	85.7	72.0
2.8B words	CPHRASE	72.2 / 79.6	75.7	78.8	91.1	86.2	78.8

Table 5.3: Performance of sentence representation models on **supervised** evaluations (Section 5.0.8). Bold numbers indicate best performance in class. Underlined indicates best overall.

5.0.8 Supervised Evaluations

Representations are applied to 6 sentence classification tasks: paraphrase identification (MSRP) [48], movie review sentiment (MR) [135], product reviews (CR) [82], subjectivity classification (SUBJ) [134], opinion polarity (MPQA) [175] and question type classification (TREC) [170]. We follow the procedure (and code) of [94]: a logistic regression classifier is trained on top of sentence representations, with 10-fold cross-validation used when a train-test split is not pre-defined.

	STS 2014							SICK
Model	News	Forum	WordNet	Twitter	Images	Headlines	All	Test + Train
SAE	17/.16	.12/.12	.30/.23	.28/.22	.49/.46	.13/.11	.12/.13	.32/.31
SAE+embs.	.52/.54	.22/.23	.60/.55	.60/.60	. 64/.64	.41/.41	.42/.43	.47/.49
SDAE	.07/.04	.11/.13	.33/.24	.44/.42	.44/.38	.36/.36	.17/.15	.46/.46
SDAE+embs.	.51/.54	.29/.29	.56/.50	.57/.58	.59/.59	.43/.44	.37/.38	.46/.46
ParagraphVec DBOW	.31/.34	.32/.32	.53/.5	.43/.46	.46/.44	.39/.41	.42/.43	.42/.46
ParagraphVec DM	.42/.46	.33/.34	.51/.48	.54/.57	.32/.30	.46/.47	.44/.44	.44/.46
Skipgram	.56/.59	.42/.42	**.73/.70**	**.71**/.74	.65/.67	**.55**/.58	.62/.63	**.60/.69**
CBOW	**.57/.61**	**.43/.44**	.72/.69	**.71/.75**	.71/.73	**.55/.59**	**.64/.65**	**.60/.69**
Unigram TFIDF	.48/.48	.40/.38	.60/.59	.63/.65	**72/.74**	.49/.49	.58/.57	.52/.58
SkipThought	.44/.45	.14/.15	.39/.34	.42/.43	.55/.60	.43/.44	.27/.29	.57/.60
FastSent	**.58/.59**	**.41**/.36	**.74/.70**	.63/.66	**.74/.78**	.57/.59	**.63/.64**	**.61/.72**
FastSent+AE	.56/ **.59**	**.41/.40**	.69/.64	**.70/.74**	.63/.65	**.58/.60**	.62/.62	.60/.65
NMT En to Fr	.35/.32	.18/.18	.47/.43	.55/.53	.44/.45	.43/.43	.43/.42	.47/.49
NMT En to De	.47/.43	.26/.25	.34/.31	.49/.45	.44/.43	.38/.37	.40/.38	.46/46
CaptionRep BOW	.26/.26	.29/.22	.50/.35	.37/.31	**.78**/.81	.39/.36	.46/.42	.56/.65
CaptionRep RNN	.05/.05	.13/.09	.40/.33	.36/.30	.76/**.82**	.30/.28	.39/.36	.53/.62
DictRep BOW	.62/.67	.42/.40	.81/.81	.62/.66	.66/.68	.53/.58	.62/.65	.57/.66
DictRep BOW+embs.	**.65/.72**	**.49/.47**	**.85/.86**	**.67/.72**	.71/.74	**.57/.61**	**.67/.70**	**.61**/.70
DictRep RNN	.40/.46	.26/.23	.78/.78	.42/.42	.56/.56	.38/.40	.49/.50	.49/.56
DictRep RNN+embs.	.51/.60	.29/.27	.80/.81	.44/.47	.65/.70	.42/.46	.54/.57	.49/.59
CPHRASE	**.69**/.71	.43/.41	.76/.73	.60/.65	.75/.79	**.60/.65**	.65/.67	.60/**.72**

Table 5.4: Performance of sentence representation models (Spearman/Pearson correlations) on **unsupervised** (relatedness) evaluations (Section 5.0.9). Models are grouped according to training data as indicated in Table 5.3.

5.0.9 Unsupervised Evaluations

We also measure how well representation spaces reflect human intuitions of the semantic sentence relatedness, by computing the cosine distance between vectors for the two sentences in each test pair, and correlating these distances with gold-standard human judgements. The SICK dataset [111] consists of 10,000 pairs of sentences and relatedness judgements. The STS 2014 dataset [3] consists of 3,750 pairs and ratings from six linguistic domains. Example ratings are shown in Table 5.2. All available pairs are used for testing apart from the 500 SICK 'trial' pairs, which are held-out for tuning hyperparameters (representation size of log-linear models, and noise parameters in SDAE). The optimal settings on this task are then applied to both supervised and unsupervised evaluations.

5.0.10 Results

Performance of the models on the supervised evaluations (grouped according to the data required by their objective) is shown in Table 5.3. Overall, SkipThought vectors perform best on three of the six evaluations, the BOW DictRep model with pre-trained word embeddings performs best on two, and the SDAE on one. SDAEs perform notably well on the paraphrasing task, going beyond SkipThought by three percentage points and approaching state-of-the-art performance of models designed specifically for the task [87]. SDAE is also consistently better than SAE, which aligns with other findings that adding noise to AEs produces richer representations [167].

Results on the unsupervised evaluations are shown in Table 5.4. The same DictRep model performs best on four of the six STS categories (and overall) and is joint-top performer on SICK. Of the models trained on raw text, simply adding CBOW word vectors works best on STS. The best performing raw text model on SICK is FastSent, which achieves almost identical performance to C-PHRASE's state-of-the-art performance for a distributed model [138]. Further, it uses less than a third of the training text and does not require access to (supervised) syntactic representations for training. Together, the results of FastSent on the unsupervised evaluations and SkipThought on the supervised benchmarks provide strong support for the sentence-level distributional hypothesis: the context in which a sentence occurs provides valuable information about its semantics.

Across both unsupervised and supervised evaluations, the BOW DictRep with pre-trained word embeddings exhibits by some margin the most consistent performance. Ths robust performance suggests that DictRep representations may be particularly valuable when the ultimate application is non-specific or unknown, and confirms that dictionary definitions (where available) can be a powerful resource for representation learning.

5.0.11 Discussion

Many additional conclusions can be drawn from the results in Tables 5.3 and 5.4.

Different objectives yield different representations It may seem obvious, but the results confirm that different learning methods are preferable for different intended applications (and this variation appears greater than for word representations). For instance, it is perhaps unsurprising that SkipThought performs best on TREC because the labels in this dataset are determined by the language immediately following the represented question (i.e. the answer) [170]. Paraphrase detection, on the other hand, may be better served by a model that focused entirely on the content *within* a sentence, such as SDAEs. Similar variation can be observed in the unsupervised evaluations. For instance, the (multimodal) representations produced by the CaptionRep model do not perform particularly well apart from on the Image category of STS where they beat all other models, demonstrating a clear effect of the well-studied modality differences in representation learning [29].

The nearest neighbours in Table 5.5 give a more concrete sense of the representation spaces. One notable difference is between (AE-style) models whose semantics come from within-sentence relationships (CBOW, SDAE, DictRep, ParagraphVec) and SkipThought/FastSent, which exploit the context around sentences. In the former case, nearby sentences generally have a high proportion of words in common, whereas for the latter it is the general concepts and/or function of the sentence that is similar, and word overlap is often minimal. Indeed, this may be a more important trait of FastSent than the marginal improvement on the SICK task. Readers can compare the CBOW and FastSent spaces at `http://45.55.60.98/`.

Differences between supervised and unsupervised performance Many of the best performing models on the supervised evaluations do not perform well in the unsupervised setting. In the SkipThought, S(D)AE and NMT models, the cost is computed based on a non-linear decoding of the internal sentence representations, so, as also observed by [5], the informative geometry of the representation space may not be reflected in a simple cosine distance. The log-linear models generally perform better in this unsupervised setting.

Differences in resource requirements As shown in Table 5.1, different models require different resources to train and use. This can limit their possible applications. For instance, while it was easy to make an online demo for fast querying of near neighbours in the CBOW and FastSent spaces, it was not practical for other models owing to memory footprint, encoding time and representation dimension.

The role of word order is unclear The average scores of models that are sensitive to word order (76.3) and of those that are not (76.6) are approximately the same across

Query	***If he had a weapon, he could maybe take out their last imp, and then beat up Errol and Vanessa.***	***An annoying buzz started to ring in my ears, beco louder and louder as my vision began to swim.***
CBOW	*Then Rob and I would duke it out, and every once in a while, he would actually beat me.*	*Louder.*
Skip Thought	*If he could ram them from behind, send them saling over the far side of the levee, he had a chance of stopping them.*	*A weighty pressure landed on my lungs and my vis at the edges, threatening my consciousness altoget*
FastSent	*Isak's close enough to pick off any one of them, maybe all of them, if he had his rifle and a mind to.*	*The noise grew louder, the quaking increased as th sidewalk beneath my feet began to tremble even mo*
SDAE	*He'd even killed some of the most dangerous criminals in the galaxy, but none of those men had gotten to him like Vitktis.*	*I smile because I'm familiar with the knock, pausing to take a deep breath before dashing dowr*
DictRep (FF+embs.)	*Kevin put a gun to the man's head, but even though he cried, he couldn't tell Kevin anything more.*	*Then gradually I began to hear a ringing in my ear*
Paragraph Vector (DM)	*I take a deep breath and open the doors.*	*They listened as the motorcycle-like roar of an engine got louder and louder then stopped.*

Table 5.5: Sample nearest neighbour queries selected from a randomly sampled 0.5m sentences of the Toronto Books Corpus.

Supervised (combined $\alpha = 0.90$)						Unsupervised (combined $\alpha = 0.93$)						
MSRP	MR	CR	SUBJ	MPAQ	TREC	News	Forum	WordNet	Twitter	Images	Headlines	All STS
0.94 (6)	0.85 (1)	0.86 (4)	0.85 (1)	0.86 (3)	0.89 (5)	0.92 (4)	0.92 (3)	0.92 (4)	0.93 (6)	0.95 (8)	0.92 (2)	0.91 (1)

Table 5.6: Internal consistency (Chronbach's α) among evaluations when individual benchmarks are left out of the (supervised or unsupervised) cohorts. Consistency rank within cohort is in parentheses (1 = most consistent with other evaluations).

supervised evaluations. Across the unsupervised evaluations, however, BOW models score 0.55 on average compared with 0.42 for RNN-based (order sensitive) models. This seems at odds with the widely held view that word order plays an important role in determining the meaning of English sentences. One possibility is that order-critical sentences that cannot be disambiguated by a robust conceptual semantics (that could be encoded in distributed lexical representations) are in fact relatively rare. However, it is also plausible that current available evaluations do not adequately reflect order-dependent aspects of meaning (see below). This latter conjecture is supported by the comparatively strong performance of TFIDF BOW vectors, in which the effective lexical semantics are limited to simple relative frequencies.

The evaluations have limitations The internal consistency (Chronbach's α) of all evaluations considered together is 0.81 (just above 'acceptable').[12] Table 5.6 shows that consistency is far higher ('excellent') when considering the supervised or unsupervised tasks as independent cohorts. This indicates that, with respect to common characteristics of sentence representations, the supervised and unsupervised benchmarks do indeed prioritise different properties. It is also interesting that, by this metric, the properties measured by MSRP and image-caption relatedness are the furthest

[12] wikipedia.org/wiki/Cronbach's_alpha

removed from other evaluations in their respective cohorts.

While these consistency scores are a promising sign, they could also be symptomatic of a set of evaluations that are all limited in the same way. The inter-rater agreement is only reported for one of the 8 evaluations considered (MPQA, 0.72 [175]), and for MR, SUBJ and TREC, each item is only rated by one or two annotators to maximise coverage. Table 5.2 illustrates why this may be an issue for the unsupervised evaluations; the notion of sentential 'relatedness' seems very subjective. It should be emphasised, however, that the tasks considered in this study are all frequently used for evaluation, and, to our knowledge, there are no existing benchmarks that overcome these limitations.

5.0.12 Conclusion

Advances in deep learning algorithms, software and hardware mean that many architectures and objectives for learning distributed sentence representations from unlabelled data are now available to NLP researchers. We have presented the first (to our knowledge) systematic comparison of these methods. We showed notable variation in the performance of approaches across a range of evaluations. Among other conclusions, we found that the optimal approach depends critically on whether representations will be applied in supervised or unsupervised settings - in the latter case, fast, shallow BOW models can still achieve the best performance. Further, we proposed two new objectives, FastSent and Sequential Denoising Autoencoders, which perform particularly well on specific tasks (MSRP and SICK sentence relatedness respectively).[13] If the application is unknown, however, the best all round choice may be DictRep: learning a mapping of pre-trained word embeddings from the word-phrase signal in dictionary definitions. While we have focused on models using naturally-occurring training data, in future work we will also consider supervised architectures (including convolutional, recursive and character-level models), potentially training them on multiple supervised tasks as an alternative way to induce the 'general knowledge' needed to give language technology the elusive human touch.

[13] We make all code for training and evaluating these new models publicly available, together with pre-trained models and an online demo of the FastSent sentence space.

Chapter 6

Representation Learning

Chapter 7

Conclusion

Some conclusive learning text

Bibliography

[1] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proccedings of NAACL*. Association for Computational Linguistics, 2009.

[2] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of NAACL-HLT 2009*, 2009.

[3] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, 2014.

[4] Enrique Alfonseca and Suresh Manandhar. Extending a lexical ontology by a combination of distributional semantics signatures. In *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, pages 1–7. Springer, 2002.

[5] Amjad Almahairi, Kyle Kastner, Kyunghyun Cho, and Aaron Courville. Learning distributed representations from reviews for collaborative filtering. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 147–154. ACM, 2015.

[6] Mark Andrews, Gabriella Vigliocco, and David Vinson. Integrating experiential and distributional data to learn semantic representations. *Psychological Review*, 116(3):463, 2009.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceeding of ICLR*, 2015.

[8] Mohit Bansal, Kevin Gimpel, and Karen Livescu. Tailoring continuous word representations for dependency parsing. In *Proceedings of ACL*, 2014.

[9] Marco Baroni, Raffaela Bernardi, and Roberto Zamparelli. Frege in space: A program of compositional distributional semantics. *Linguistic Issues in Language Technology*, 9, 2014.

[10] Marco Baroni, Georgiana Dinu, and Germán 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, 2014.

[11] Marco Baroni and Alessandro Lenci. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721, 2010.

[12] Lawrence W Barsalou. Grounded cognition: past, present, and future. *Topics in Cognitive Science*, 2(4):716–724, 2010.

[13] Lawrence W Barsalou, W Kyle Simmons, Aron K Barbey, and Christine D Wilson. Grounding conceptual knowledge in modality-specific systems. *Trends in Cognitive Sciences*, 7(2):84–91, 2003.

[14] Lawrence W Barsalou and Katja Wiemer-Hastings. Situating abstract concepts. *Grounding Cognition: The Role of Perception and Action in Memory, Language, and Thought*, pages 129–163, 2005.

[15] Islam Beltagy, Katrin Erk, and Raymond Mooney. Semantic parsing using distributional semantics and probabilistic logic. In *ACL 2014 Workshop on Semantic Parsing*, 2014.

[16] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003.

[17] Yoshua Bengio and Jean-Sébastien Sénécal. Quick training of probabilistic neural nets by importance sampling. In *Proceedings of AISTATS 2003*, 2003.

[18] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

[19] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of the Association for Computational Linguistics*, 2014.

[20] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[21] Raffaella Bernardi, Georgiana Dinu, Marco Marelli, and Marco Baroni. A relatedness benchmark to test the role of determiners in compositional distributional semantics. *Proceedings of ACL*, 2013.

[22] Chris Biemann. Ontology learning from text: A survey of methods. In *LDV forum*, volume 44, pages 75–93, 2005.

[23] Slaven Bilac, Timothy Baldwin, and Hozumi Tanaka. Improving dictionary accessibility by maximizing use of available knowledge. *Traitement Automatique des Langues*, 44(2):199–224, 2003.

[24] Slaven Bilac, Wataru Watanabe, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka. Dictionary search based on the target word description. In *Proceedings of NLP 2014*, 2004.

[25] Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.

[26] Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. *Proceedings of EMNLP*, 2014.

[27] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*, 2015.

[28] Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 136–145. Association for Computational Linguistics, 2012.

[29] Elia Bruni, Nam Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47, 2014.

[30] Elia Bruni, Jasper Uijlings, Marco Baroni, and Nicu Sebe. Distributional semantics with eyes: Using image analysis to improve computational representations of word meaning. In *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012.

[31] Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[32] Joan L Bybee and Paul J Hopper. *Frequency and the Emergence of Linguistic Structure*, volume 45. John Benjamins Publishing, 2001.

[33] Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C. Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems*, pages 1853–1861, 2014.

[34] Nick Chater and Christopher D Manning. Probabilistic models of language processing and acquisition. *Trends in Cognitive Sciences*, 10(7):335–344, 2006.

[35] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[36] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of EMNLP*, 2014.

[37] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of EMNLP*, 2014.

[38] Philipp Cimiano, Andreas Hotho, and Steffen Staab. Learning concept hierarchies from text corpora using formal concept analysis. *J. Artif. Intell. Res.(JAIR)*, 24:305–339, 2005.

[39] Stephen Clark and Stephen Pulman. Combining symbolic and distributional models of meaning. In *AAAI Spring Symposium: Quantum Interaction*, pages 52–55, 2007.

[40] Ronan Collobert and Jason 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. ACM, 2008.

[41] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[42] D Alan Cruse. *Lexical semantics*. Cambridge University Press, 1986.

[43] Sebastian J Crutch and Elizabeth K Warrington. Abstract and concrete concepts have structurally different representational frameworks. *Brain*, 128(3):615–627, 2005.

[44] Hamish Cunningham. Information extraction, automatic. *Encyclopedia of language and linguistics,*, pages 665–677, 2005.

[45] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, pages 3061–3069, 2015.

[46] Barry J Devereux, Lorraine K Tyler, Jeroen Geertzen, and Billi Randall. The centre for speech, language and the brain (cslb) concept property norms. *Behavior Research Methods*, pages 1–9, 2013.

[47] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the Association for Computational Linguistics*, 2014.

[48] Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics, 2004.

[49] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[50] Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2014.

[51] Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of EACL*, volume 2014, 2014.

[52] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1999.

[53] Yansong Feng and Mirella Lapata. Visual information in semantic representation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 91–99. Association for Computational Linguistics, 2010.

[54] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. Building Watson: An overview of the DeepQA project. In *AI magazine*, volume 31(3), pages 59–79, 2010.

[55] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on the World Wide Web*, pages 406–414. ACM, 2001.

[56] R. Firth, J. *A synopsis of linguistic theory 1930-1955*, pages 1–32. Oxford: Philological Society, 1957.

[57] Jerry A Fodor. *The modularity of mind: An essay on faculty psychology*. MIT press, 1983.

[58] Dedre Gentner. On relational meaning: The acquisition of verb meaning. *Child development*, pages 988–998, 1978.

[59] Dedre Gentner. Why verbs are hard to learn. *Action meets word: How children learn verbs*, pages 544–564, 2006.

[60] Yulia Tsvetkov Leonid Boytsov Anatole Gershman and Eric Nyberg Chris Dyer. Metaphor detection with cross-lingual model transfer. In *Proceedings of ACL*, 2014.

[61] Matthew L. Ginsberg. Dr. FILL: Crosswords and an implemented solver for singly weighted CSPs. In *Journal of Artificial Intelligence Research*, pages 851–886, 2011.

[62] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970.

[63] Stephan Gouws, Yoshua Bengio, and Greg Corrado. BilBOWA: Fast bilingual distributed representations without word alignments. In *Proceedings of NIPS Deep Learning Workshop*, 2014.

[64] Scott T Grafton. Embodied cognition and the simulation of action to understand others. *Annals of the New York Academy of Sciences*, 1156(1):97–117, 2009.

[65] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[66] Thomas L Griffiths, Mark Steyvers, and Joshua B Tenenbaum. Topics in semantic representation. *Psychological review*, 114(2):211, 2007.

[67] Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. Learning bilingual lexicons from monolingual corpora. In *ACL*, volume 2008, pages 771–779, 2008.

[68] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[69] Zellig S Harris. Distributional structure. *Word*, 1954.

[70] Samer Hassan and Rada Mihalcea. Semantic relatedness using salient semantic analysis. In *AAAI*, 2011.

[71] Vasileios Hatzivassiloglou, Judith L Klavans, Melissa L Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen McKeown. Simfinder: A flexible clustering tool for summarization. In *NAACL Workshop on Automatic Summarization*, 2001.

[72] Xiaodong He, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. Indirect-hmm-based hypothesis alignment for combining outputs from machine translation systems. In *Proceedings of EMNLP*, pages 98–107. Association for Computational Linguistics, 2008.

[73] Karl Moritz Hermann and Phil Blunsom. Multilingual distributed representations without word alignment. In *Proceedings of ICLR*, 2013.

[74] Karl Moritz Hermann and Phil Blunsom. Multilingual Distributed Representations without Word Alignment. In *Proceedings of ICLR*, 2014.

[75] Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 2015.

[76] Felix Hill, Douwe Kiela, and Anna Korhonen. Concreteness and corpora: A theoretical and practical analysis. *CMCL 2013*, page 75, 2013.

[77] Felix Hill and Anna Korhonen. Learning abstract concepts from multi-modal data: Since you probably can't see what i mean. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*, October 2014.

[78] Felix Hill, Anna Korhonen, and Christian Bentz. A quantitative empirical analysis of the abstract/concrete distinction. *Cognitive Science*, 2013.

[79] Felix Hill, Roi Reichart, and Anna Korhonen. Multi-modal models for concrete and abstract concept meaning. *Transactions of the Association for Computational Linguistics (TACL)*, 2014.

[80] Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 2015.

[81] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[82] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[83] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.

[84] Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. A neural network for factoid question answering over paragraphs. In *Proceedings of EMNLP*, 2014.

[85] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the Association for Computational Linguistics*, 2015.

[86] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. *Proceedings of ALC*, 2015.

[87] Yangfeng Ji and Jacob Eisenstein. Discriminative improvements to distributional sentence similarity. In *EMNLP*, pages 891–896, 2013.

[88] Brendan T Johns and Michael N Jones. Perceptual inference through global lexical similarity. *Topics in Cognitive Science*, 4(1):103–120, 2012.

[89] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, October 2013. Association for Computational Linguistics.

[90] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of EMNLP*, 2014.

[91] Douwe Kiela and Stephen 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)@ EACL*, pages 21–30, 2014.

[92] Douwe Kiela, Felix Hill, Anna Korhonen, and Stephen Clark. Improving multi-modal representations using image dispersion: Why less is sometimes more. In *Proceedings of ACL*. ACL, 2014.

[93] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *Transactions of the Association for Computational Linguistics*, 2015. to appear.

[94] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3276–3284, 2015.

[95] A. Klementiev, I. Titov, and B. Bhattarai. Inducing crosslingual distributed representations of words. In *Proceedings of COLING*, 2012.

[96] Tomáš Kočiský, Karl Moritz Hermann, and Phil Blunsom. Learning Bilingual Word Representations by Marginalizing Alignments. In *Proceedings of ACL*, jun 2014.

[97] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211, 1997.

[98] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of ICML*, 2014.

[99] Geoffrey Leech, Roger Garside, and Michael Bryant. Claws4: the tagging of the British National Corpus. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pages 622–628. Association for Computational Linguistics, 1994.

[100] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, 2014.

[101] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185, 2014.

[102] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

[103] Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics â AS Human Language Technologies (NAACL HLT 2015), Denver, CO*, 2015.

[104] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

[105] Changliang Li, Bo Xu, Gaowei Wu, Xiuying Wang, Wendong Ge, and Yan Li. Obtaining better word representations via language transfer. In *Computational Linguistics and Intelligent Text Processing*, pages 128–137. Springer, 2014.

[106] Mu Li, Yang Zhang, Muhua Zhu, and Ming Zhou. Exploring distributional similarity based models for query spelling correction. In *Proceedings of ALC*, pages 1025–1032. Association for Computational Linguistics, 2006.

[107] Michael L. Littman, Greg A. Keim, and Noam Shazeer. A probabilistic approach to solving crossword puzzles. *Artificial Intelligence*, 134(1):23–55, 2002.

[108] Minh-Thang Luong, Richard Socher, and Christopher D Manning. Better word representations with recursive neural networks for morphology. *CoNLL-2013*, 104, 2013.

[109] Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. *Proceedings of the Association for Computational Linguistics*, 2015.

[110] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan Yulle. Deep captioning with multimodal recurrent neural networks (m-rnn). In *Proceedings of ICLR*, 2015.

[111] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC*, pages 216–223. Citeseer, 2014.

[112] Arthur B Markman and Edward J Wisniewski. Similar and different: The differentiation of basic-level categories. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23(1), 1997.

[113] Yuval Marton, Chris Callison-Burch, and Philip Resnik. Improved statistical machine translation using monolingually-derived paraphrases. In *Proceedings of EMNLP*, pages 381–390. Association for Computational Linguistics, 2009.

[114] Ken McRae, George S Cree, Mark S Seidenberg, and Chris McNorgan. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547–559, 2005.

[115] Ken McRae, Saman Khalkhali, and Mary Hare. Semantic and associative relations in adolescents and young adults: Examining a tenuous dichotomy. In Valerie F Reyna, Sandra B Chapman, Michael R Dougherty, and Jere Ed Confrey, editors, *The adolescent brain: Learning, reasoning, and decision making.* American Psychological Association, 2012.

[116] Olena Medelyan, David Milne, Catherine Legg, and Ian H Witten. Mining meaning from wikipedia. *International Journal of Human-Computer Studies*, 67(9):716–754, 2009.

[117] Grégoire Mesnil, Yann Dauphin, Xavier Glorot, Salah Rifai, Yoshua Bengio, Ian J Goodfellow, Erick Lavoie, Xavier Muller, Guillaume Desjardins, David Warde-Farley, et al. Unsupervised and transfer learning challenge: a deep learning approach. *Journal of Machine Learning Research-Proceedings Track*, 27:97–110, 2012.

[118] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of International Conference of Learning Representations*, Scottsdale, Arizona, USA, 2013.

[119] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Proceedings of INTERSPEECH 2010*, 2010.

[120] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *CORR*, 2013.

[121] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[122] Dmitrijs Milajevs, Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Matthew Purver. Evaluating neural word representations in tensor-based compositional settings. In *Proceedings of EMNLP*, 2014.

[123] Jeff Mitchell and Mirella Lapata. Vector-based models of semantic composition. In *ACL*, pages 236–244, 2008.

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

[125] Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. In *Advances in Neural Information Processing Systems*, pages 1081–1088, 2009.

[126] Diego Mollá and José Luis Vicedo. Question answering in restricted domains: An overview. *Computational Linguistics*, 33(1):41–61, 2007.

[127] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *Proceedings of the international Workshop on Artificial Intelligence and Statistics*, pages 246–252, 2005.

[128] Raymond H Myers. *Classical and Modern Regression with Applications*, volume 2. Duxbury Press Belmont, CA, 1990.

[129] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10, 2009.

[130] Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. The University of South Florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407, 2004.

[131] Donald A Norman. Memory, knowledge, and the answering of questions. *ERIC*, 1972.

[132] Sebastian Padó, Ulrike Padó, and Katrin Erk. Flexible, corpus-based modelling of human plausibility judgements. In *Proceedings of EMNLP-CoNLL*, pages 400–409, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[133] Allan Paivio. Dual coding theory: Retrospect and current status. *Canadian Journal of Psychology*, 45(3):255, 1991.

[134] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.

[135] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124. Association for Computational Linguistics, 2005.

[136] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics, 2004.

[137] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*, October 2014.

[138] Nghia The Pham, Germán Kruszewski, Angeliki Lazaridou, and Marco Baroni. Jointly optimizing word representations for lexical and sentential tasks with the c-phrase model. In *Proceedings of ALC*, 2015.

[139] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100. ACM, 2008.

[140] David C Plaut. Semantic and associative priming in a distributed attractor network. In *Proceedings of CogSci*, volume 17, pages 37–42, 1995.

[141] Tamara Polajnar, Laura Rimell, and Stephen Clark. An exploration of discourse-based sentence spaces for compositional distributional semantics. In *Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (LSD-Sem)*, page 1, 2015.

[142] Gabriel Recchia and Michael N Jones. More data trumps smarter algorithms: Comparing pointwise mutual information with latent semantic analysis. *Behavior research methods*, 41(3):647–656, 2009.

[143] Joseph Reisinger and Raymond Mooney. A mixture model with sharing for lexical semantics. In *Proceedings of EMNLP*, pages 1173–1182. Association for Computational Linguistics, 2010.

[144] Joseph Reisinger and Raymond 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. Association for Computational Linguistics, 2010.

[145] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI*, 1995.

[146] Philip Resnik and Jimmy Lin. 11 evaluation of nlp systems. *The handbook of computational linguistics and natural language processing*, 57:271, 2010.

[147] Stephen Roller and Sabine Schulte im Walde. A multimodal LDA model integrating textual, cognitive and visual modalities. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1146–1157, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[148] Eleanor Rosch, Carol Simpson, and R Scott Miller. Structural bases of typicality effects. *Journal of Experimental Psychology: Human perception and performance*, 2(4):491, 1976.

[149] Tony Rose, Mark Stevenson, and Miles Whitehead. The reuters corpus volume 1-from yesterday's news to tomorrow's language resources. In *LREC*, volume 2, pages 827–832, 2002.

[150] Herbert Rubenstein and John B Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

[151] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pages 1–42, 2014.

[152] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. *arXiv preprint arXiv:1507.04808*, 2015.

[153] Ryan Shaw, Anindya Datta, Debra VanderMeer, and Kaushik Dutta. Building a scalable database-driven reverse dictionary. *Knowledge and Data Engineering, IEEE Transactions on*, 25(3):528–540, 2013.

[154] C Silberer and M Lapata. Learning grounded meaning representations with autoencoders. In *Proceedings of Association for Computational Linguistics*. ACL, 2014.

[155] Carina Silberer, Vittorio Ferrari, and Mirella Lapata. Models of semantic representation with visual attributes. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, August*, 2013.

[156] Carina Silberer and Mirella Lapata. Grounded models of semantic representation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1423–1433. Association for Computational Linguistics, 2012.

[157] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics, 2011.

[158] Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*, pages 2231–2239, 2012.

[159] Lin Sun, Anna Korhonen, and Yuval Krymolowski. Verb class discovery from rich syntactic data. In *Computational linguistics and intelligent text processing*, pages 16–27. Springer, 2008.

[160] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, 2011.

[161] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[162] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[163] Joseph Turian, Lev Ratinov, and Yoshua 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. Association for Computational Linguistics, 2010.

[164] Peter D Turney. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585, 2012.

[165] Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

[166] Amos Tversky. Features of similarity. *Psychological review*, 84(4):327, 1977.

[167] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[168] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

[169] Luis Von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 319–326. ACM, 2004.

[170] Ellen M Voorhees. Overview of the trec 2001 question answering track. *NIST special publication*, pages 42–51, 2002.

[171] Ivan Vulić, Wim De Smet, and Marie-Francine Moens. Identifying word translations from comparable corpora using latent topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 479–484. Association for Computational Linguistics, 2011.

[172] Ivan Vulic, Wim De Smet, and Marie-Francine Moens. Identifying word translations from comparable corpora using latent topic models. In *Proceedings of the Association for Computational Linguistics*, 2011.

[173] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards ai-complete question answering: a set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[174] Janyce Wiebe. Learning subjective adjectives from corpora. In *AAAI/IAAI*, pages 735–740, 2000.

[175] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210, 2005.

[176] Katja Wiemer-Hastings and Xu Xu. Content differences for abstract and concrete concepts. *Cognitive Science*, 29(5):719–736, 2005.

[177] Gbolahan K Williams and Sarabjot Singh Anand. Predicting the polarity strength of adjectives using wordnet. In *ICWSM*, 2009.

[178] Pengcheng Wu, Steven CH Hoi, Hao Xia, Peilin Zhao, Dayong Wang, and Chunyan Miao. Online multimodal deep similarity learning with application to image retrieval. In *Proceedings of the 21st ACM International Conference on Multimedia*, pages 153–162. ACM, 2013.

[179] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of ACL*, pages 133–138. Association for Computational Linguistics, 1994.

[180] Chung Yong and Shou King Foo. A case study on inter-annotator agreement for word sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Standardizing Lexical Resources (SIGLEX99)*, 1999.

[181] Matthew D. Zeiler. Adadelta: An adaptive learning rate method. In *arXiv preprint arXiv:1212.5701*, 2012.

[182] Michael Zock and Slaven Bilac. Word lookup on the basis of associations: From an idea to a roadmap. In *Proceedings of the ACL Workshop on Enhancing and Using Electronic Dictionaries*, 2004.

Appendix A

Extra Information

Some more text ...