UNIVERSITY OF
CAMBRIDGE

Representing Linguistic Meaning in Distributed Memories

From Words to Sentences and Beyond

Felix Hill

St John's College

This dissertation is submitted for the degree of Doctor of Philosophy

Abstract

In recent years, models of language understanding built on distributed semantic representations have advanced the state-of-the-art in various application areas, from text classification to machine translation. However, in many cases, this success was contingent on large-scale task-specific labelled training data. Unlike such models, humans are capable of acquiring linguistic and semantic knowledge from a diverse range of unlabelled language and sensory input for application to multiple tasks. This thesis develops computational methods for acquiring general-purpose semantic representations of linguistic units in distributed memories, to provide a basis for more robust, human-like language understanding that does not require task-specific or labelled data. The proposed techniques include novel methods for encoding the semantics of words, phrases, and sentences (in isolation and in context). In each case, the techniques result in at or near state-of-the-art performance on multiple benchmarks, including comparisons with human data and downstream classification or question-answering tasks. Moreover, resources are developed for better understanding the properties of semantic representations, enabling analysis of how they emerge from raw input as a consequence of model learning objectives. The results highlight the advantage of representing the linguistic signal in distributed memories, and indicate that we ultimately might be able develop a seamless (human-like) interface between such representations and more abstract semantic knowledge encoded in a similar way.

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
June 2016

Acknowledgements

After 3 months I didn't think I would get here, but most of the rest was a constant pleasure, and that was because I learned so much from so many people. In broadly chronological order, these are my advisor, Anna Korhonen, Christian Bentz, Douwe Kiela, Roi Reichart, Yoshua Bengio, Kyunghyun Cho, Antoine Bordes and Jason Weston. I'm also indebted to St John's College, Cambridge, Google and Facebook AI Research for making this research possible. On the personal side, I couldn't have done it without Becky and the Wallers, the Worcester crew and Hannah pushing me over the line. But most of all, this is an achievement for my parents, Zelda and Simon, whose support and encouragement allowed me to do what I dreamed and believed in.

Contents

Chapter 1

Introduction

The question of how children come to be competent language users is vigorously debated by researchers across the human sciences. However, there is a clear consensus that children do not learn language because they are taught it.[1] While instances of caregivers or teachers instructing children how to pronounce or use words are not uncommon, there is little or no evidence that such interventions are necessary for fluent language use to emerge (Tomasello and Tomasello, 2009; Pinker, 2009). On the other hand, two aspects of the child's environment do appear critical for language to flourish. The first of these is the massive amounts of correct language to which the developing brain is exposed. The second is the fact that many of the things that humans naturally desire or aspire to require linguistic interaction with other humans. This latter fact explains the motivation that may be necessary for learning, while the external information or signal required is a consequence of both.

This thesis is about computational models that learn to represent or understand language under similar conditions to a child listening to adults speak. That is, by making sense of everything that is observed, rather than by asking for additional information when it is uncertain, or receiving explicit corrections or explanations. This sort of learning is often referred to as *unsupervised*. The models in this thesis mimic human language acquisition in that they learn passively from massive amounts of data, and allow linguistic concepts to emerge rather than being instructed according to some oracle or theory. On the other hand, they are a less faithful reflection of the post-verbal interactive stage of human language learning, in which ongoing passive learning is combined with goal-driven linguistic interaction. Realistically modelling this combination

[1] See e.g. `http://www.linguisticsociety.org/resource/faq-how-do-we-learn-language`

poses many practical challenges, and research in this direction is left for immediate future work.

1.1 Distributed representations of knowledge

The way in which the models considered in this thesis effect unsupervised learning is by acquiring *distributed representations* of linguistic units (words, phrases, sentences and eventually short passages) (Hinton, 1986). Given a set S of s semantic (in this case, linguistic) concepts, we can encode each concept according to how strongly it activates each of a set of n potentially independent features. If each feature takes k possible values, we can unambiguously encode a maximum of k^n concepts in such a representation space. This contrasts with a situation in which the presence of each concept is represented by the activation of one of s discrete features. Such an encoding strategy is typically referred to as a *symbolic* or *localist representation*. The advantage of a symbolic approach is that, for a given view of the world (and thus a given set features and their activations) we can know immediately and unambiguously which concepts are present. On the other hand, in a system capable of representing n such features, a symbolic approach can only deal with n distinct concepts as possible inputs, whereas a system based on distributed representations could, in theory, reason about k^n concepts.

There are additional reasons why systems built on distributed representations are popular when aiming to replicate core human capabilities such as object recognition or language understanding. First, distributed models naturally generalise because of shared attributes. While we may not be able to associate words or easily-delimited meaning with individual features in a distributed representation, we can observe that representations of related or similar concepts such as dog and $wolf$ share common patterns of activation. A model that is trained exclusively on images of dogs (not wolves) eating meat may learn that concepts with a dog-like pattern of activation are carnivorous, from which it might infer that wolves are also carnivorous. Such generalisation is not possible using purely symbolic models, which do not naturally induce a similarity space on the concepts they encounter. It is, on the other hand, highly characteristic of human reasoning and cognition (Rosch et al., 1976a).

Second, a distributed model can acquire new concepts (beyond the original set S) without requiring additional memory, since concepts can 'emerge' as (relatively) stable patterns of activation across its feature set. In the present example, the concept of *carnivore* might be acquired if the model observes sufficient examples of animals eating meat (and different animals not eating meat). Over time, *carnivore* would come to behave just like any other concept in the memory of the model (i.e. associated with a stable pattern of feature activations). This is not the case with symbolic models. While the attribute *carnivore* could be associated with a subset of concepts in a symbolic model, it would not have the status of concept in itself unless the model added to its memory (and feature space). The existence of emergent concepts, of varying degree of stability or salience, is also a commonly-observed trait of human cognition (Patterson et al., 2007).

Third, like human semantic memory, knowledge encoded in distributed representations is content-addressable. If humans see a partially obscured object or hear an incomplete word, they may be able to infer the correct entity quickly based on this information. Similarly, if a distributed model receives partial information about a concept (in the form of an incomplete activation pattern), it may be able to access the full concept using this information (and indeed, may be able to learn to organise its memory such that its ability to perform such reconstructions improves). By contrast, in a symbolic model it is not generally possible to search for knowledge based on aspects of the information in question. Additional information about its location in memory is normally required.

Finally, if we encode concepts via symbols, or even pre-specified semantic features (sub-concepts), we introduce a large degree of bias or ad hocness into our representations (Miikkulainen and Dyer, 1991). The world is a large, chaotic continuum, and we have very little idea of how the brain rationalises the enormous stream of input data it receives. By training and then interrogating models that acquire representations of this data that are optimal for achieving realistic linguistic objectives, we can learn in a comparatively unbiased way which aspects of the data are important for realising the particular objective. This in turn can provides at least circumstantial evidence that the human brain exploits the same characteristics of the signal in order to achieve its learning goals.

These natural effects of computing with distributed representations are particularly appropriate for modelling linguistic cognition. Semantic categories such as the referents of a word are notoriously hard to delimit in terms of defining properties (Fauconnier, 1994). English speakers can easily use and understand a word like *chair*, and a seat with four legs and a back would typically be classed as a chair, but these properties are neither necessary nor sufficient for classifying an object as such. A longer seat with four legs and a back is not really a chair, but rather a *bench*. Similarly, an armchair does not typically have four legs, and we might call a tree stump a chair if people sat on it regularly enough (Prinz, 2004).

Western linguistic theory and education often involves more abstract categories referring to language itself, such as *noun* or *subject*, but even these can prove frustratingly intangible upon thorough consideration. Nouns are often said to refer to entities or objects, but *failure* or *unravelling* do not fall into either category. The type of word considered to be an *verb* or an *adjective* can be very different to speakers of different languages (Anward et al., 1997). How can we realise effective communication with others using and referring to concepts if we ourselves do not know how to unambiguously define or describe them? The solution advocated in this thesis is that we do so in the same way as a computational models that perform computations on distributed representations. This hypothesis also seems to be consistent with our (limited) knowledge the biological hardware of the brain (Kiefer and Pulvermüller, 2012).

1.2 Deep learning

The terms *artificial neural networks*, *connectionism* and, now, *deep learning* refer to a family of computational models that store knowledge in distributed representations and perform computations on these representations. In these approaches, distributed representations of the concepts observed in data are learned as part of the same process as learning how to interpret or manipulate these representations in some optimal way (e.g. to perform accurate prediction or classification based on the input). In many deep learning models, ‘manipulating’ representations corresponds to computing additional layers of representation of the same input (via continuous but non-linear mathematical operations). In theory, these representations then encode increasingly abstract or sophisticated characteristics of the input concepts. For example, in a deep convolutional network trained to classify images of objects into (semantic) object categories, features in the initial representations correspond to crude visual properties such as the edges of objects, whereas higher-level representations (which are the ones on which serve as input to the output classifier) seem to encode properties that can be much more directly connected to the model’s objective (such as the presence of eyes) (Zeiler and Fergus, 2014).

Every deep learning model acquires (at least one layer of) distributed representations for each concept that it observes. Unlike representations or encodings defined manually, the representations learned for a particular model are likely to be optimal or close to optimal with respect to objective of the model (given the constraints of its architecture). On the other hand, post-hoc analysis may be required to understand the representational strategy learned by the model, particularly since features in the model do not generally correspond to concepts that humans can label with words.

1.2.1 Neural language models

Deep learning models whose objective corresponds to linguistic tasks, such as predicting missing words, answering questions or classifying the sentiment of statements, are referred to as *neural language models* (NLMs) (Bengio et al., 2003b). The first NLM built on the scale of today's deep learning models was developed for the task of *language modelling*; estimating the probability of a word sequence, or the related objective of predicting the next word in a text.[2] In recent years, neural language models have begun to achieve state-of-the-art performance on a range of language processing tasks, including summarisation (Rush et al., 2015), visual question answering (Antol et al., 2015), machine comprehension (Hill et al., 2016), factoid question answering (Bordes et al., 2014), entailment detection (Rocktäschel et al., 2015) and machine translation (Bahdanau et al., 2015). In all of these cases, NLMs achieve at or close to state-of-the-art performance on the established benchmarks.

Beyond this clear empirical case for focusing on NLMs (which was much less compelling at the beginning of my doctoral research) there are important scientific reasons for addressing language processing problems within the connectionist paradigm. Symbolic approaches that are able to learn and generalise about language typically rely on a pre-specified categorisation of linguistic input (for instance into classes like *noun*, *subject* or *adjunct*). Such syntactic or semantic categories are not instantiated in the physical world, and in many cases there can be significant disagreement among language users regarding category membership (Anward et al., 1997). NLMs avoid this minefield by inducing such categories as (soft) clusterings of concepts in the similarity spaces of their distributed representations, inferring principles about these categories as computations on the spaces. By working with NLMs, we allow a (relatively) unbi-

[2] This should not discredit the various smaller-scale connectionist models of written language proposed before then (See e.g. Elman 1990; Miikkulainen and Dyer 1991)

ased model to teach us more about the data, rather than relying on a human view of the data - and its inevitable social and cultural biases - to inform the design of the model.

1.3 Unsupervised learning

The term *supervised learning* typically refers to a setting for a computational or statistical inference problem in which a model is presented with data consisting of a set of inputs and corresponding outputs. Based on these training data, the model learns to adjust its internal computation so that it can predict outputs from inputs as accurately as possible. Models trained in this way are typically evaluated on their ability to generalise by testing their accuracy on a subset of input and output pairs that are deliberately excluded from the training data.

In the vast majority cases to date, NLMs have been trained and applied in this supervised way. This thesis, however, focuses on applications of NLMs that do not align with this paradigm. That said, the designation of an approach as *unsupervised* is not unambiguous, The aforementioned case of language modelling being a notable case in point. Language models are supervised in that they learn by considering some words in a text, making a prediction about neighbouring words and then adjusting their internal computations based on what they observe the actual neighbouring words to be in the text. Moreover, as with other supervised methods, they are evaluated by measuring the performance of the model on similar text that was deliberately excluded from the training data. Nevertheless, they can also be considered unsupervised because the training data (text) is not explicitly divided into inputs and outputs (words play the role of both), and a correct label is not explicitly associated (e.g. by some annotation process) with the correct answer for a given input. A child listening to two adults converse could presumably implement his or her own language model-style learning (by making and correcting predictions about future words), but we would be unlikely to say that the child's learning is being supervised by the adults in this case.

Beyond language modelling, methods in NLP that are normally considered unsupervised include semantic models that build representations from the statistics of word occurrences in raw text (such as LSA, LDA and related approaches (Landauer and Dumais, 1997; Blei, 2012)) or clustering approaches to problems like word sense induction (Agirre and Soroa, 2007) and lexical category acquisition (Sun and Korhonen, 2009). Two characteristics stand out as distinguishing these methods from others more often described as supervised. First, (and like language modelling) in these methods the training data (usually raw text) would exist without the intervention of NLP researchers and/or annotators. Second, unlike the canonical language modelling setting, these methods are not evaluated on a task that mirrors the objective guiding their learning.

In this thesis, I therefore consider a method or approach to be unsupervised if **it is applied to tasks that differ from the training objective *and* does not require training data produced with machine learning in mind**. From a practical perspective, the first of these criteria is desirable, in theory at least, because it indicates that the model has learned something that may ultimately be useful for more than one particular tasks or application. The advantage of the second is less esoteric; it simply implies negligible cost for generating training data and a greater likelihood that such data exists for various domains and/or languages. Longer term (and more speculatively) the benefit of unsupervised learning may lie in the fact that it appears to be an important part of human learning and development. Human behaviour is not just characterised by strong performance on tasks that are well-practiced, but rather on the ability to adapt and apply knowledge acquired in a diverse range of settings to tasks that may resemble, but are not necessarily identical to, tasks that the human has attempted previously.

This characterisation of unsupervised methods does not preclude the involvement of humans whatsoever in the production of training data. In so much as natural languages are by definition a product of human activity, the training data for any language application will likely involve humans. Neither does it preclude learning from knowledge of the right answer. Indeed, most neural networks learn by backpropagating an error associated (via a mathematical cost function) with the ‘correctness’ of their predictions.

Structure of the thesis

Chapter 2 concerns what is to-date the most popular and effective unsupervised application of NLMs, namely the acquisition of distributed word representations or *embeddings*. The principal contribution is a novel resource, SimLex-999, that enables more principled evaluation and analysis of these representations. In Chapter 3, I present a novel method for learning word representations with NLMs, one that exploits a new sequence-to-sequence learning architecture and parallel bilingual corpora, and show that these representations exhibit potentially valuable properties that are not observed when learning via previous methods. In Chapter 4, I propose an objective for learning distributed representations of phrases with NLMs that exploits the structured textual information in dictionaries and encyclopedias. I show that the encoding of the information in dictionaries into a distributed semantic memory enables models to generalise and transfer their lexical knowledge to crossword question-answering. In Chapter 5, I place these observations in wider perspective by systematically comparing various ways of learning encode phrases and sentences in a distributed memory. This analysis includes two novel methods for acquiring such distributed representations, each of which exhibits certain advantages over existing approaches. Finally, in Chapter 6, I consider a new class of NLMs, *memory networks*, which couple the process of encoding phrases or sentences in a semantic memory with a particular language prediction task. By allowing us to measure the effect of particular representational strategies on an extrinsic prediction task, this end-to-end architecture enables a more functionally-grounded undertanding of the most useful strategies for encoding linguistic content in distributed memories.

Chapter 2

Understanding and evaluating distributed word representations

In many applications of machine learning, unsupervised learning has proved an intangible and impractical goal. However, computational lexical semantics is in some sense an exception. There are various established methods for acquiring word representations from unlabelled data. The reason that unsupervised learning was more tractable for lexical semantics than for other AI problems is largely down to a linguistic principle known as the *Distributional Hypothesis* (Firth, 1957). This is the idea that the meaning of a word can be inferred from any coherent text corpus based on its pattern of co-occurrence with other words in the corpus.

Algorithms that exploit the distributional hypothesis to learn semantic representations of words have existed for almost as long as the machines on which their realisation depends. Methods in which the semantics of a word is encoded in a vector representation by counting its co-occurrences with other words were proposed as early as the 1960s (Cordier, 1965; Harper, 1965). In the mid-1990s it was observed that these typically sparse representations can often be improved by dimensionality reduction techniques involving matrix factorisation (Landauer and Dumais, 1997). The 2000s saw the emergence of generative graphical models that learn representations of documents in terms of a finite number of latent random variables (distributions over word types) corresponding to semantic ‘topics’ (Blei et al., 2003). If words are represented in a low-dimensional space spanned by the topic variables of a trained model, the resulting semantic space reflects human semantic judgements (Griffiths et al., 2007).

As early as 1988 it had been observed that connectionist models trained via backpropagation could learn plausible distributed word representations from raw text Miikkulainen and Dyer (1989), although their properties were only analysed via rudimentary qualitative analyses. Owing to practical, and perhaps cultural, factors, however, this approach to representation learning remained comparatively unknown compared with corpus enumeration and matrix factorisation methods. This began to change with the first large-scale neural (probabilistic) language models (Bengio et al., 2003b). Neural language models (NLMs) learn low-dimensional distributed word representations (popularly known as 'embeddings') by optimising an objective concerning the prediction of words in texts. The original NLM architectures predicted words in documents based on an ordered (finite) sequence of previous words, and were designed with the aim of better language modelling. However, it was also observed that such models naturally acquire word representations with particularly rich semantics (Collobert and Weston, 2008). These word representations were successfully applied as features to improve performance on a wide range of classification tasks requiring sensitivity to lexical semantics, such as POS-tagging and word-sense disambiguation (Collobert and Weston, 2008).

The evident applicability of neural word embeddings led to the development of simpler NLMs whose explicit purpose was to learn high-quality distributed word representations (Mikolov et al., 2013a). In these shallow architectures, all non-lexical parameters (i.e. feedforward or recurrent update weights) are eschewed, and word representations are optimised for the direct prediction of neighbouring words. Moreover, in the very simplest variants, the probabilistic objective in which the likelihood of correct neighbour prediction is maximised can be replaced by a (heuristic) ranking loss in which the model simply distinguishes between 'correct' and 'incorrect' training examples. These models have inspired numerous modifications and tweaks (such as Levy and Goldberg 2014a; Pennington et al. 2014b; Hill and Korhonen 2014) but are established as the de facto standard for distributed word representations at the time of writing, thanks to their robust performance and fast, memory efficient training algorithm.

The Challenge of Evaluation Despite this history of clear progress in algorithm design, unsupervised models of word semantics are not immune from a critical issue faced by all representation learning research: the challenge of effective evaluation. The contribution presented in this chapter is designed to address this issue by providing a tool, *SimLex-999*, for robust analyses and evaluation of word representations. Like various existing evaluations created for this purpose, including WordSim-353 (Finkelstein et al., 2001) and MEN (Bruni et al., 2014), SimLex-999 works by comparing representation spaces acquired by distributional models with an independent and external measure of human semantic intuitions. However, SimLex-999 was designed specifically to overcome limitations of the existing methods of evaluation.

First, it provides *better coverage*. While existing resources contain only concrete noun concepts (MEN) or cover only some of these distinctions via a random selection of items (WS-353), SimLex-999 contains a principled selection of adjective, verb and noun concept pairs that span the full range of lexical concreteness. This design was informed by empirical evidence that humans represent concepts of distinct part-of-speech (POS) (Gentner, 1978) and conceptual concreteness (Hill et al., 2013b) differently. Second, the property of representation spaces measured by SimLex-999 is *clearly defined*. Existing evaluations test the extent to which models reflect a broad, ill-defined notion of semantic relatedness, whereas SimLex-999 requires that models capture a more specific phenomenon that is well understood by cognitive psychologists, namely conceptual similarity (Tversky, 1977). Third, SimLex-999 measures a more *robust* cognitive phenomenon, as evidenced by the high inter-annotator agreement not reflected with other evaluations. Finally, SimLex-999 is *challenging* for computational models. While annotators find it unproblematic to consistently quantify conceptual similarity, this aspect of human cognition is not easy for distributional models to replicate. In contrast, for existing evaluations that focus on semantic relatedness, the best distributional models already surpass inter-human agreement, leaving little scope for meaningful evaluation as models improve further.

A second main contribution in this chapter, presented in Section 2.5, is the evaluation and analysis of the main classes of distributional semantic models using SimLex-999. These include a representative selection of neural language models, together with longer standing approaches based on counting lexical co-occurrences and linear dimensionality reduction. This application of SimLex-999 reveals substantial differences in the ability of models to represent concepts of different types. Such insight in turn suggests ways in which distributional models might improve on their current ability to capture human semantic intuitions. Taken together, these analyses 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.

Finally, I discuss the potentially crucial role to be played by robust evaluations such as SimLex-999 as we move towards models and systems with more human-like general semantic awareness, and discuss some clear limitations and challenges for future research in lexical representation learning.

2.1 Reproducing human semantic knowledge

For an exponent of machine learning, the answer to the question *what makes a good representation?* may be simply *one that facilitates good prediction, classification or regression*. However, language is a uniquely human phenomenon, and when evaluating representation-learning algorithms, researchers in NLP have tended to downplay these overtly practical considerations in favour of the requirement that representation spaces directly reflect human conceptual organisation. Thus, with certain notable exceptions (Collobert and Weston, 2008; Turian et al., 2010), linear metrics such as Euclidean or cosine distance are used to determine which concepts are close or distant or to otherwise quantify the relative orientation of the space. This organisation is then compared to human semantic intuitions, as captured in established semantic resources (thesauri, dictionaries, taxonomies) or via direct experimentation. In this respect, SimLex-999 is no different from previous evaluation benchmarks. Its innovation lies in the specifics of its design and, in particular, its strong emphasis on conceptual (or semantic) *similarity* rather than a broader semantic relation that I refer to here as *association*.

2.1.1 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 (Plaut, 1995; McRae et al., 2012).

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 (McRae et al., 2012)), such pairs can seem, at least intuitively, to be similar but not strongly associated.

The association/similarity distinction had been the object of philosophical (Grigg, 2009), psychological (Crutch et al., 2009) and neuroscientific (Lucas, 2000) studies. Nevertheless the conclusions drawn from these investigations were based on relatively small populations of concepts. As a first attempt at taking a more data-driven approach to understanding association and similarity, I interrogated two existing large-data resources. To estimate similarity, I considered proximity in the WordNet taxonomy (Fellbaum, 1999). Specifically, I applied the measure of Wu and Palmer (1994) (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 of similarity (Wu and Palmer, 1994). To estimate association, I extracted ratings directly from the University of South Florida Free Association Database (USF) (Nelson et al., 2004). 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, I 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 I show in Section 2.2.2, these are precisely the sort of pairs that are not contained in existing evaluation gold

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 2.1: 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).

standards. Table 2.1 lists the USF noun pairs with the lowest similarity scores overall, and also those with the largest additive discrepancy between association strength and similarity.[1]

Association and similarity in NLP As noted above, 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 (He et al., 2008; Haghighi et al., 2008; Marton et al., 2009; Beltagy et al., 2014). Models of association, on the other hand, may be better suited to tasks such as word-sense disambiguation (Navigli, 2009), and applications such as text classification (Phan et al., 2008) in which the target classes correspond to topical domains such as *agriculture* or *sport* (Rose et al., 2002).

Despite this intuitive importance, the majority of research into unsupervised learning of semantic representations in NLP makes no principled distinction between association and similarity (see e.g. (Huang et al., 2012; Reisinger and Mooney, 2010b; Luong et al., 2013)).[2] A notable exception is Turney (2012), 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 cov-

[1] Hill et al. (2013b) present additional large-scale analyses of similarity and association and show how these relations interact with conceptual concreteness.

[2] Several papers that take a knowledge-based or symbolic approach to meaning do address the similarity/association issue (Budanitsky and Hirst, 2006).

ering 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.

Agirre et al. (2009b) also 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 I refer to as *WS-Sim*, and a subset focused on association, which I 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. (2009b) 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 in this dataset. Therefore the insights that can be drawn from the results of the Agirre et al. (2009b) study are limited.

2.2 Motivation for SimLex-999

In this section, I motivate the design decisions made in developing SimLex-999. I begin by examining the distinction between similarity and association. I 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. I finish by reviewing existing gold standards, and show that none enables a satisfactory evaluation of the capability of models to capture similarity.

2.2.1 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. Gentner (2006) showed that children find verb concepts harder to learn than noun concepts, and Markman and Wisniewski (1997) present evidence that different cognitive operations are employed when comparing two nouns or two verbs. Hill et al. (2014) 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 (Paivio, 1991; Hill et al., 2013b). Concreteness has also been associated with differential performance in computational text-based (Hill et al., 2013a) and multimodal semantic models (Kiela et al., 2014).

2.2.2 Existing gold standards and evaluation resources

An important part of the motivation for the design of SimLex-999 derives from limitations in the existing evaluation resources that were most commonly used in research on word representation learning. In discussing these evaluations, I consider how well each satisfies the following three 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.

The review of existing evaluations begins with the gold standard most commonly-applied in NLP research prior to SimLex-999.

WordSim-353 WS-353 (Finkelstein et al., 2001) 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',[3] 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 I confirm analytically in Section 2.5.3, these instructions result in pairs being rated according to association rather than similarity.[4] 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.
2. No associated but dissimilar concepts receive low ratings.

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 Collobert and Weston (2008) and Huang et al. (2012) reach, or even surpass, the inter-annotator agreement ceiling in estimating the WS-353 scores. Huang et al. (2012) 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 (Padó et al., 2007; Reisinger and Mooney, 2010a; Silberer and Lapata, 2014). 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.[5]

Thus, at least according to the established wisdom in NLP evaluation (Yong and Foo, 1999; Cunningham, 2005; Resnik and Lin, 2010), the strength of the conclusions

[3] See e.g. Huang et al. (2012); Bansal et al. (2014)

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

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

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 (Alfonseca and Manandhar, 2002).

WS-Sim WS-Sim is the set of pairs in WS-353 identified by Agirre et al. (2009b) 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 Agirre et al. (2009b) 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.

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 (Agirre et al., 2009b). 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 resource produced by Rubenstein and Goodenough (1965) (henceforth *RG*), 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 $r = 0.86$ (Hassan and Mihalcea, 2011)) than the reported inter-annotator agreement of $r = 0.85$ (Rubenstein and Goodenough, 1965). Finally, the size of RG renders it an even less comprehensive evaluation than WS-Sim.

The MEN Test Collection A larger dataset, MEN (Bruni et al., 2014), is used in a handful of recent studies (Bruni et al., 2012b; Bernardi et al., 2013). 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.[6]

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 Bruni et al. (2012a).

Synonym detection sets Multiple-choice synonym detection tasks, such as the TOEFL test questions (Landauer and Dumais, 1997), 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 (Griffiths et al., 2007) and NLP (Turney and Pantel, 2010), they do not require similarity to be measured on a continuous scale.

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

2.3 The SimLex-999 dataset

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

2.3.1 Choice of concepts

Separating similarity from association To create a test of the ability of models to capture similarity as opposed to association, I 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, I excluded pairs containing a multiple-word item [*hot dog, mustard*], and pairs containing a capital letter [*Mexico, sun*]. I 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, I paired up the concepts from the 900 associated pairs at random. From these random parings, I excluded those that coincidentally occurred elsewhere in USF (and therefore had a degree of association). From the remaining pairs, I 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. I 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, I counted the frequency with which the items in each pair occurred with the three possible tags in the POS-tagged British National Corpus (Leech et al., 1994). To minimise POS ambiguity, which could lead to inconsistent rating, I 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 (Fellbaum, 1999). I thus felt it would be very difficult, or even counter-intuitive, for annotators to quantify the similarity of mixed POS pairs according to the 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 (Hill et al., 2014; Kiela et al., 2014).[7] 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, I aimed to include both concept types in SimLex-999.

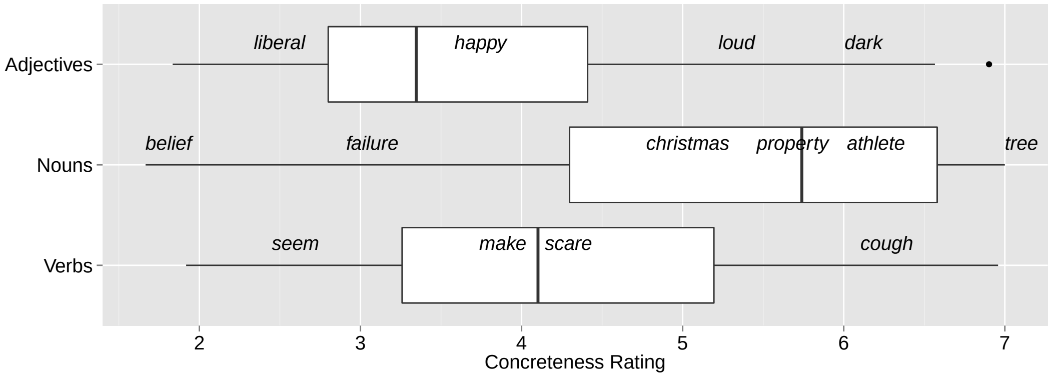


Figure 2.1: Boxplots showing the interaction between concreteness and POS for concepts in USF. The white boxes range from the first to third quartiles and the central vertical line indicates the median.

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 2.1 demonstrates, concreteness (as the average over these ratings) interacts with POS on these concepts: nouns are on average more

[7] 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.

concrete than verbs which are more concrete than adjectives. However, there is also clear variation in concreteness within each POS category. I therefore aimed to select pairs for SimLex-999 that spanned the full abstract-concrete continuum within each POS category.

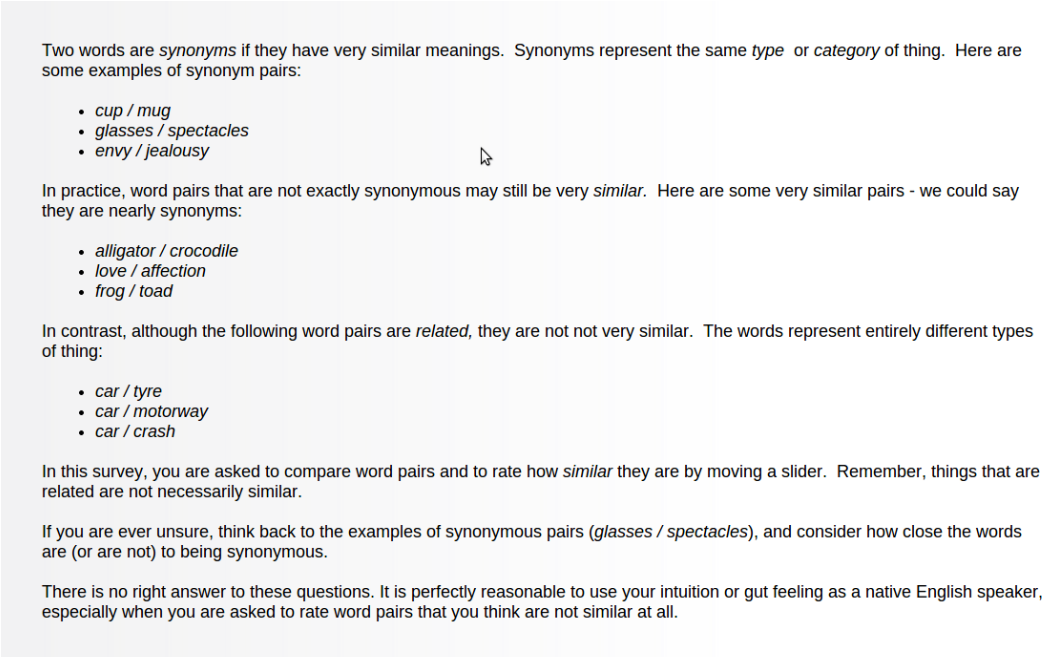
After excluding any pairs that contained an item with no concreteness rating, for each potential SimLex-999 pair I considered both the concreteness of the first item and the additive difference in concreteness between the two items. This enabled the sampling to be stratified equally across four classes: (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 I selected 600 noun pairs, 200 verb pairs and 100 adjective pairs, and from the unassociated (non-USF) cohort, I 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.

2.3.2 Question design

The annotator instructions for SimLex-999 are shown in Figure 2.2. I 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 2.2 suggests that the instructions would need separate cases to cover different concept types, increasing the difficulty of the rating task. Therefore I 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.

Immediately following the instructions in Figure 2.2, 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



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.

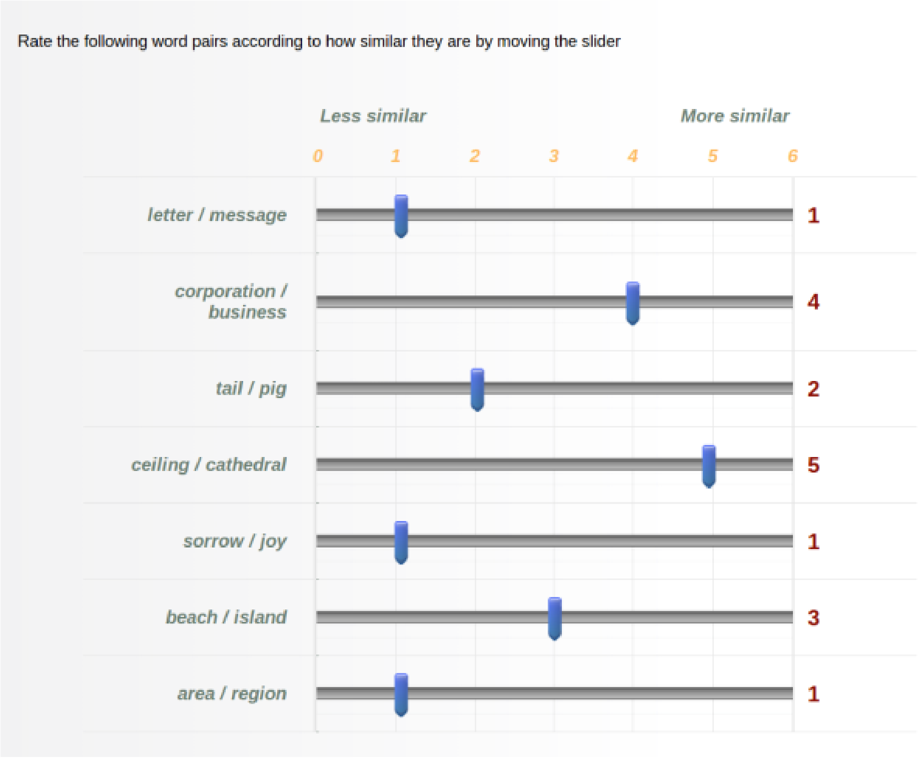
Figure 2.2: Instructions for SimLex-999 annotators.

began rating pairs in groups of 6 or 7 pairs by moving a slider, as shown in Figure 2.3.

This group size was chosen because the (relative) rating of a set of pairs implicitly requires pairwise comparisons between all pairs in that set. Therefore, using larger groups would have increased the cognitive load on the annotators exponentially. 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 reassign the rating previously attributed to the first pair before rating the remaining new items.

2.3.3 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 Huang et al. (2012), words are not rated in a phrasal or sentential context. Such



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 2.3: 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 clicked on the slider in the zero position to assign that rating and proceed to the next page.

meaning-in-context evaluations are motivated by a desire to disambiguate words that otherwise might be considered to have multiple senses.

I 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.[8] 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 are of differing POS category.

[8] 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 (Resnik, 1995; Pedersen et al., 2004).

2.3.4 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 corresponding to a single page on the web survey and consisting of 7 pairs (with the exception of the last group of the 20, which had 6). Of the 7 pairs in each group, 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.

2.3.5 Participants

500 residents of the USA were recruited from Mechanical Turk, each with at least 95% approval rate for previous work. 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.

2.3.6 Post-processing

In order to correct for systematic differences in the overall calibration of the rating scale between respondents, I 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, I 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, I 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, I 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 34 raters). Finally, I computed average (mean) scores for each pair, and transformed all scores linearly from the interval $[0, 6]$ to the interval $[0, 10]$.

2.4 Analysis of dataset

In this section I analyse the responses of the SimLex-999 annotators and the resulting ratings. First, by considering inter-annotator agreement I 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, I verify that a valid notion of similarity was understood by the annotators, in that they were able to accurately separate similarity from association.

2.4.1 Inter-annotator agreement

As in previous annotation or data collection for computational semantics (Padó et al., 2007; Reisinger and Mooney, 2010a; Silberer and Lapata, 2014) I computed the inter-

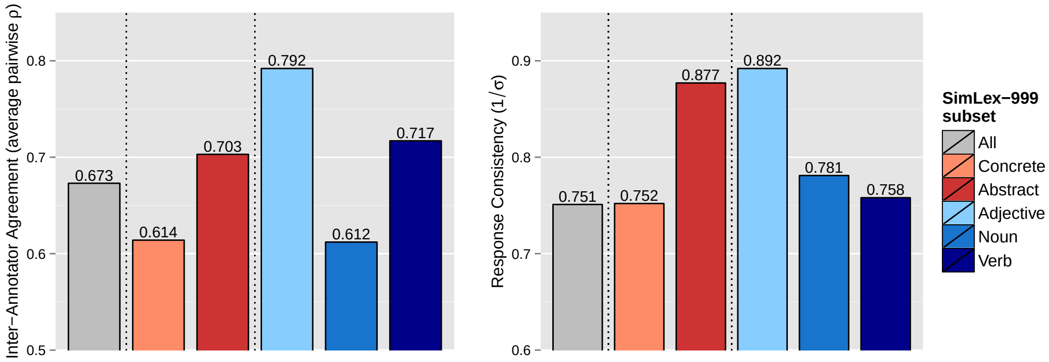


Figure 2.4: Left: Inter-annotator agreement, measured by average pairwise Spearman ρ correlation, for ratings of concept types in SimLex-999. **Right:** Response consistency, reflecting the standard deviation of annotator ratings for each pair, averaged over all pairs in the concept category.

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 (Bruni et al., 2012b). However, two of the creators of MEN who independently rated the dataset achieved an agreement of $\rho = 0.68$.[9]

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 2.4 (left), agreement was not uniform across the concept types. Contrary to what might be expected given established concreteness effects (Paivio, 1991), I observed not only higher inter-rater agreement but also less per-pair variability for abstract rather than concrete concepts[10].

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 the cause of this effect is not obvious, a possible cause is that

[9] 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.

[10] 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 /999	SimLex	SimLex rank /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 2.2: 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.

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 (Williams and Anand, 2009) and sentiment (Wiebe, 2000). 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.

2.4.2 Response validity: similarity not association

Inspection of the SimLex-999 ratings indicated that pairs were indeed evaluated according to similarity rather than association. Table 2.2 includes examples that demonstrate a clear dissociation between the two semantic relations.

To verify this effect quantitatively, I 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

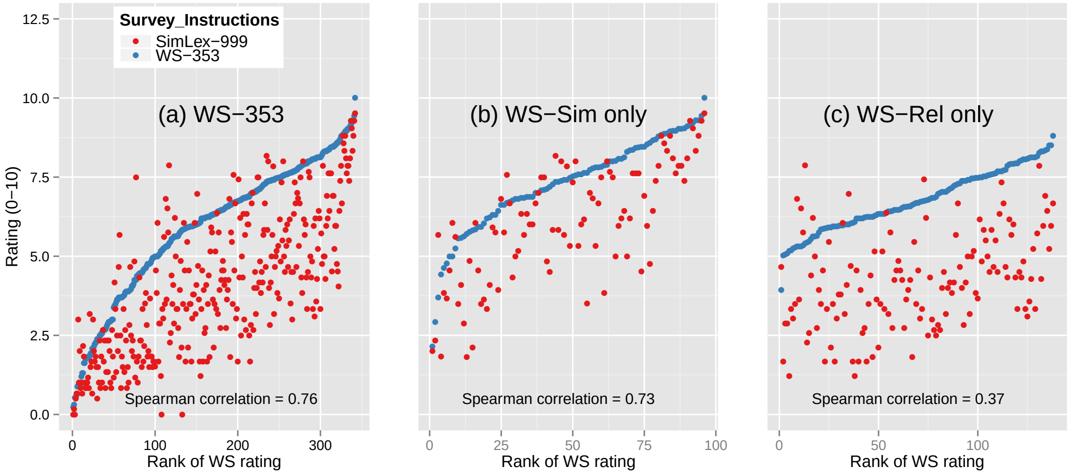


Figure 2.5: **(a)** Pairs rated by WS-353 annotators (blue points, ranked by rating) and the corresponding rating of annotators following the SimLex-999 instructions (red points). **(b-c)** The same analysis, restricted to pairs in the WS-Sim or WS-Rel subsets of WS-353.

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 the 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, I compared the WS-353 and SimLex-999-based ratings on the subsets WS-Rel and WS-Sim, which were hand-sorted by Agirre et al. (2009b) to include pairs connected by association (and not similarity) and those connected by similarity (but possibly also association) respectively.

As shown in Figure 2.5(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 2.5(b)) than for those in WS-Rel (2.5(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.

2.4.3 Finer-grained semantic relations

I 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*) (Cruse, 1986). 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.

I chose not to make these finer-grained relations the basis of the 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 (Levy et al., 2015b). Such a designation can seem to require specific world-knowledge (is a *snake* a *reptile*?), can be gradual, as evidenced by typicality effects (Rosch et al., 1976b), 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. I 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 2.6 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*). I 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 *cutlery* might provide a backdrop against which the differences of distinct members become particularly salient.

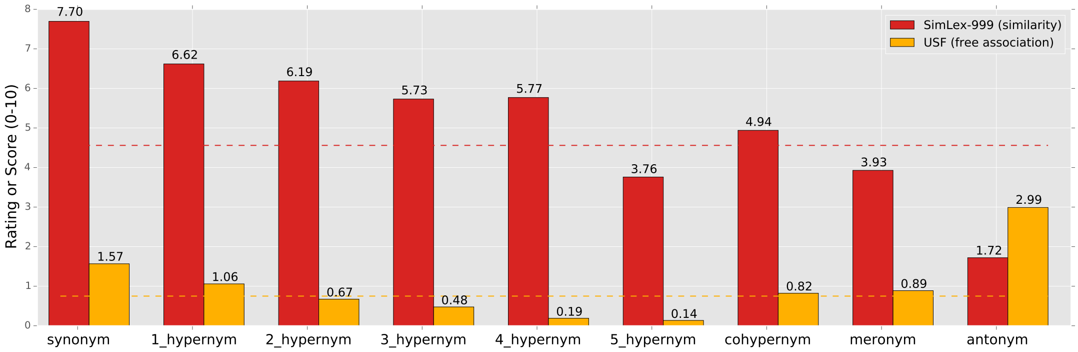


Figure 2.6: Average SimLex and USF free association scores across pairs representing different fine-grained semantic relations. All relations were extracted from WordNet. $n\_hypernym$ refers to a direct hypernymy path of length n. Note that the average SimLex rating across all 999 word pairs (dashed 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.

2.5 Evaluating models with SimLex-999

In this section, I 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 (Baroni et al., 2014b): vector space co-occurrence (counting) models and neural language models (NLM)s (Bengio et al., 2003a). I first show that SimLex-999 is in general notably more difficult for models to estimate than existing gold standards. I 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.

2.5.1 Neural language models for word representation

Collobert & Weston Collobert and Weston (2008) apply a ranking-based NLM to learn 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, distributed representations learned by this model, and the other NLMs, are usually referred to as *embeddings* (Turian et al., 2010). Collobert and Weston (2008) train their models on 852 million words of text from a 2007 dump of Wikipedia and the RCV1 Corpus (Lewis et al., 2004) and use their embeddings to achieve state-of-the-art results on a variety of NLP tasks. I downloaded the embeddings directly from the authors' webpage.[11]

Huang et al. Huang et al. (2012) 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 Collobert and Weston (2008)) and (ii) the document d in which that word occurs (global context). As with Collobert and Weston (2008), the model represents input sentences as a matrix of word embeddings. In addition, it represents documents in the input corpus as single-vector averages

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

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. Embeddings were again downloaded from the authors' webpage.[12]

Log-linear models Mikolov et al. (2013a) propose a framework for learning word embeddings using neural language models that are much shallower than those of standard NLMs. This enables faster representation learning for large vocabularies. Despite this simplification, the resulting embeddings achieve state-of-the-art performance on several semantic tasks including sentence completion and analogy modelling (Mikolov et al., 2013a,c). In fact, Mikolov et al. (2013a) present two related architectures, *Skipgram* and *CBOW*. Their experiments and those carried out since (Baroni et al., 2014b) reveal the performance of these two approaches to be similar, so we focus our analyses on the (marginally) simpler variant, Skipgram.

For each word type w in the vocabulary V, the Skipgram model learns both a 'source-embedding' $r_w \in \mathbb{R}^d$ and a 'context-embedding' $\hat{r}_w \in \mathbb{R}^d$ such that, given a source word, its ability to predict nearby context words is maximised. The probability of seeing context word c given source 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 (source-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 source word. An integer $t(n)$ is then sam-

[12] `www.socher.org`.

pled 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, source/context training pairs are such that (i) only words within a k-window of the source are selected as context words for that source, and (ii) words closer to the source are more likely to be selected than those further away.

The training objective is then to maximise 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 CBOW architecture differs from the Skipgram in that, for each position in the corpus, the current word is taken to be the object of prediction (the context), and the source from which it is predicted is the combination (via either average or sum) of words in a surrounding window. Both the CBOW and Skipgram models are optimsed via stochastic gradient descent using a linearly decaying learning rate.

As with other NLMs, the Skipgram and CBOW models capture conceptual semantics by exploiting the fact that words appearing in similar linguistic contexts are likely to have similar meanings. During training, the model adjusts its embeddings to increase the probability of observing the training corpus. For the Skipgram model, 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 source embedding incrementally 'closer' to the context-embeddings of its collocates. In the source embedding space, this results in embeddings of concept words that regularly occur in similar contexts moving closer together. It is in this way that Skipgram implicitly exploits the Distributional Hypothesis.[13]

I use the author's Word2vec software in order to train their model and use the source embeddings in the evaluations. I 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.

[13] See Section 2.6 for more about the connections between NLMs and traditional vector space models.

2.5.2 Vector space (counting) models

To place the performance of the NLMs in context, I compared their performance with vector space models, following the guidelines for optimal performance outlined by Kiela and Clark (2014). After extracting the 2000 most frequent word tokens in the corpus that are not in a common list of stopwords[14] as features, I populated a matrix of co-occurrence counts with a row for each of the concepts in some pair in the 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) (Recchia and Jones, 2009). The rows of the resulting matrix constitute the vector representations of the concepts.

LSA As proposed initially by Landauer and Dumais (1997), I also experimented with models in which Singular Value Decomposition (SVD) (Golub and Reinsch, 1970) 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, similarity was calculated as the cosine similarity between the (vector) representations learned by that model.

2.5.3 Results

In experimenting with different models on SimLex-999, I 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?

Overall performance on SimLex-999 Figure 2.7 shows the performance of the NLMs on SimLex-999 versus on comparable datasets, measured by Spearman's ρ cor-

[14] Taken from the Python Natural Language Toolkit (Bird, 2006).

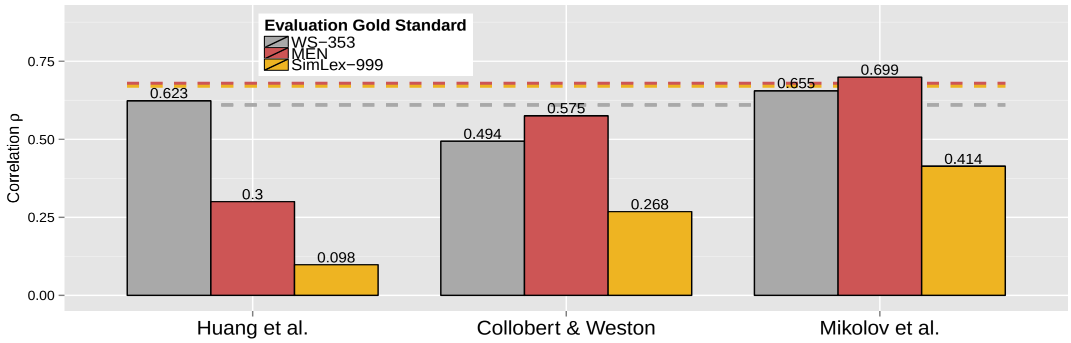


Figure 2.7: Performance of NLMs on WS-353, MEN and SimLex-999. All models are trained on Wikipedia; note that as Wikipedia is constantly growing, the Mikolov et al. (2013a) model exploited slightly more training data (≈1000m tokens) than the Huang et al. (2012) model (≈990m), which in turn exploited more than the Collobert and Weston (2008) model (≈852m). Dashed horizontal lines indicate the level of inter-annotator agreement for the three datasets.

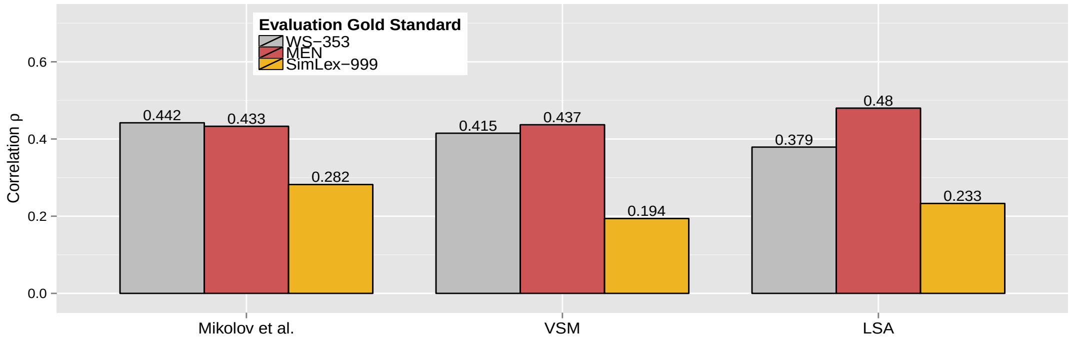


Figure 2.8: Comparison between the leading NLM, *Mikolov et al.*, the vector space model, *VSM*, and the *LSA* model. All models were trained on the ≈150m word RCV1 Corpus (Lewis et al., 2004).

relation. All models estimate the ratings of MEN and WS-353 more accurately than SimLex-999. The Huang et al. (2012) model performs well on WS-353,[15] 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 Huang et al. (2012) model to focus on association rather than similarity. However, the true explanation may be less simple, since the Huang et al. (2012) model performs weakly on the association-based MEN dataset. The Collobert and Weston (2008) model is more robust across WS-353 and MEN, but still does not match the performance of the Mikolov et al.

[15] This score, based on embeddings downloaded from the authors' webpage, is notably lower than the score reported by Huang et al. (2012) mentioned in Section 2.5.1.

(2013a) model on SimLex-999.

Figure 2.8 compares the best performing NLM model (Mikolov et al., 2013a) with the VSM and LSA models.[16] In contrast to recent results that emphasise the superiority of NLMs over alternatives (Baroni et al., 2014b), I observed no clear advantage for the NLM over the VSM or LSA when considering the association-based gold standards WS-353 and MEN together. While the NLM is the strongest performer on WS-353, LSA 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 2.7 and 2.8 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,[17] their performance on SimLex-999 is well below the inter-annotator agreement (Figure 2.7). This suggests that there is ample scope for SimLex-999 to guide the development of improved models.

Modeling similarity vs. association The comparatively low performance of NLM, VSM and LSA models on SimLex-999 compared with MEN and WS-353 is consistent with the 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, I 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 2.9, all models performed worse when the evaluation was restricted to pairs of strongly-associated concepts, which was consistent with the hypothesis. The Collobert and Weston (2008) model was better than the Huang et al.

[16] I conduct this comparison on the smaller RCV1 Corpus (Lewis et al., 2004) because training the VSM and LSA models is comparatively slow.

[17] Training times reported by Huang et al. (2012) and by Collobert and Weston (2008) at `http://ronan.collobert.com/senna/`.

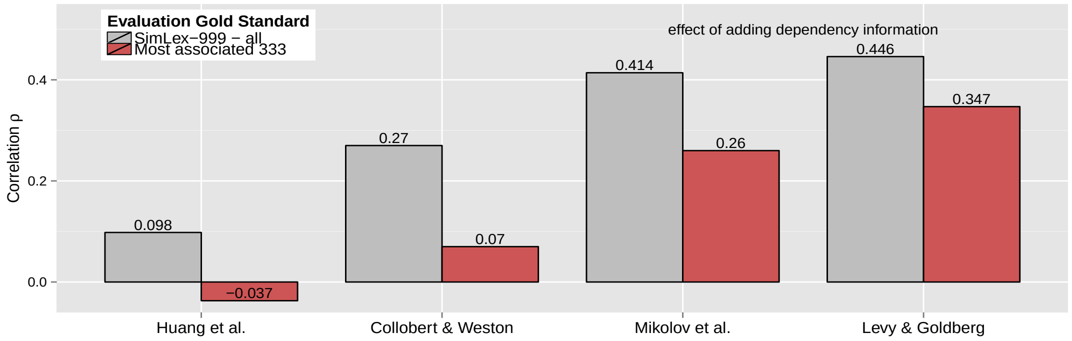


Figure 2.9: The ability of NLMs to model the similarity of highly-associated concepts versus concepts in general. The two models on the right hand side also demonstrate the effect of training an NLM (the Mikolov et al. (2013a) model) on running-text (*Mikolov et al.*) vs. on dependency-based input (*Levy & Goldberg*).

(2012) 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.

In a futher analysis designed to shed light on how distributional models captures information pertinent to similarity, I compared the modification of the Skipgram of Levy and Goldberg (2014a), in which source/context pairs are restricted to those in a (syntactic) dependency relationship. It was already suggested by Levy and Goldberg (2014a) that such a modification could yield a semantic space better ordered to semantic equivalence or similarity, although their demonstration of this effect was somewhat informal

As illustrated in Figure 2.9, the dependency-based embeddings outperform the original (running text) embeddings trained on the same corpus. Moreover, the comparatively large increase in the red 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.

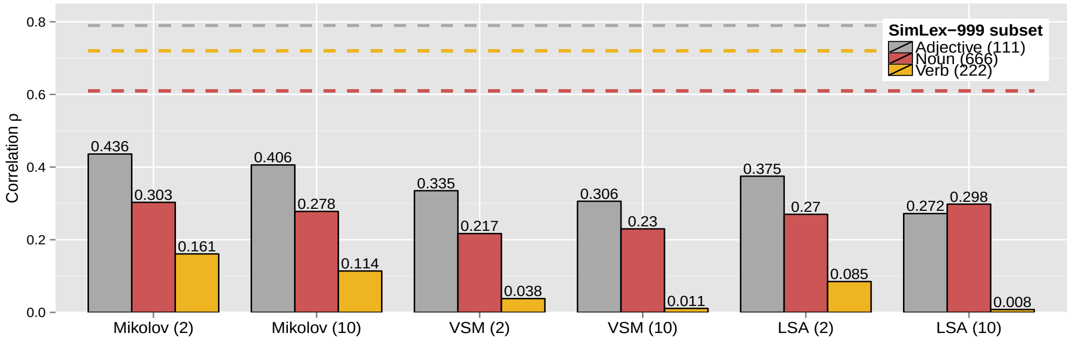


Figure 2.10: Performance of models on POS-based subsets of SimLex-999. The window size for each model is indicated in parentheses. Inter-annotator agreement for each POS is indicated by the dashed horizontal line.

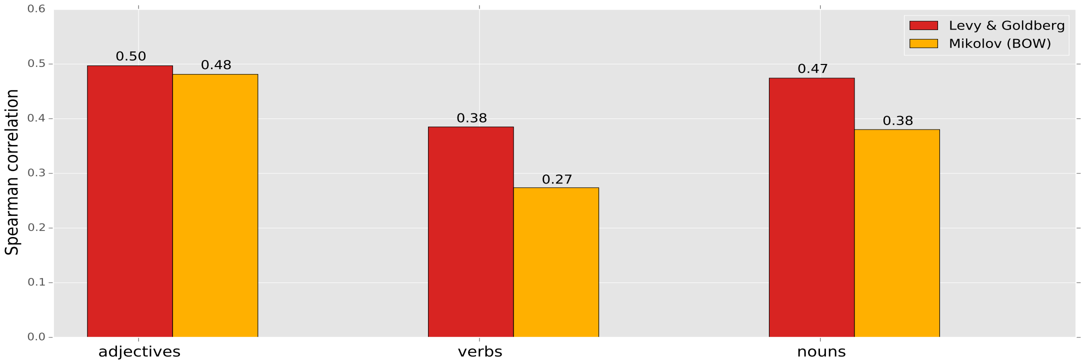


Figure 2.11: The importance of dependency-focussed contexts (in the Levy & Goldberg model) for capturing concepts of different POS, when compared to a standard Skipgram (BOW) model trained on the same Wikipedia corpus.

Learning concepts of different POS Given the theoretical likelihood of variation in model performance across POS categories noted in Section 2.2, I evaluated the Mikolov et al. (2013a), VSM and LSA models on the subsets of SimLex-999 containing adjective, noun and verb concept pairs.

The analyses yield two notable conclusions, as shown in Figure 2.10. 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, I also computed performance on the POS subsets of SimLex-999 of the dependency-based model of Levy and Goldberg (2014a) and the standard skipgram model, in which linguistic contexts are encoded as simple bags-of-words (BOW) (Mikolov et al., 2013a) (trained on the same Wikipedia text). As shown in Figure 2.11, dependency-aware contexts yield the largest improvements for capturing verb similarity. This aligns with the cognitive theory of verbs as *relational concepts* (Markman and Wisniewski, 1997) 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 (Sun et al., 2008). 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 2.1), I 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, I 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 2.12).

Interestingly, the performance of models on the most abstract and most concrete

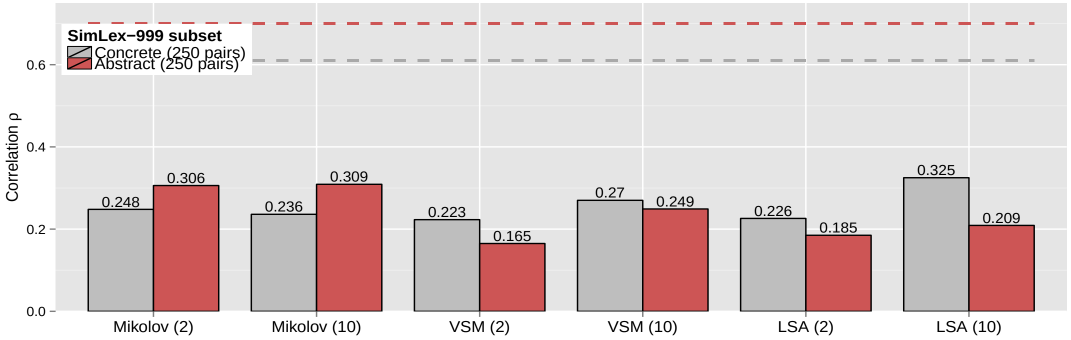


Figure 2.12: Performance of models on concreteness-based subsets of SimLex-999. Window size is indicated in parentheses. Horizontal dashed lines indicate inter-annotator agreement between SimLex-999 annotators on the two subsets.

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 and LSA 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 (Hill et al., 2013a), 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 LSA, are better equipped to capture the semantics of concrete entities, whereas the embeddings learned by NLMs can better capture abstract semantics.

2.6 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. I 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 I demonstrated, many existing were not clearly-constructed in terms of a well-defined semantic relation (for instance, measuring a combination of similarity and 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 I 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. I therefore consider the comparatively low performance of state-of-the-art models on SimLex-999 to be one of its principal strengths. Overcoming the deficit with the inter-rating ceiling can be a target to guide the development of the next generation of distributional models. Indeed, since its release, researchers have adopted SimLex-999, both as a way of comparing the performance of models and as a means of answering a diverse range of questions pertinent to similarity modelling, distributional semantics and representation learning in general (see e.g. (Levy et al., 2015a; Wang et al., 2015)).

What is so special about neural word embeddings? Since the analyses in this chapter were conducted, a clearer understanding has emerged of the connection between the word embeddings learned by shallow (log-linear) NLMs and VSMs. The original consensus, based on systematic comparisons (Baroni et al., 2014b), was that shallow neural language models learned 'better quality' representations than counting (VSM) approaches. This conclusion is also supported by the analyses presented in this chapter, although Table 2.8 suggests that the difference ($\rho = 0.28$ vs. $\rho = 0.23$) is not enormous on SimLex-999, and negligible on certain types of concept such as nouns (Table 2.10). However, Levy and Goldberg (2014b) have since shown that a non-probabilistic variant of the shallow NLMs (*Skipgram with negative sampling*) effectively minimises the same objective function as a (counting) vector-space model in which sparse count vectors are transformed with SVD. This result formalises some of the intuition expressed in Section 2.5.1 concerning how both types of model ultimately exploit the distributional hypothesis.

The equivalence, or at least close relationship, between shallow NLMs and VSMs made it unclear why studies, including this one, should have observed empirical differences in performance. Thankfully, Levy et al. (2015a) produced a very plausible explanation for this uncertaintly. The aspects of the Skipgram (or CBOW) algorithm that are most critical for the improved performance over VSMs had been excluded from the formal demonstration of equivalence because they seem to be peripheral to the main Skipgram (or CBOW) architecture. For instance, their experiments showed that the position-dependent random sampling of context-words within the fixed window (as described in Section 2.5.1) was an important factor in improved representations in the Skipgram model. Of course, this 'stochastic context window' property can also be easily applied to the VSM algorithm. In this way, Levy et al. (2015a) showed that VSMs can be modified to produce representations that are equally rich as those of shallow NLMs. In practice, however, Skipgram and CBOW remain the most popular algorithms for learning word representations from text, perhaps because of the available fast implementations and the much lower memory footprint of the algorithms.

The future of word representations Distributional semantics aims to infer the meaning of words based on the *company they keep* (Firth, 1957). However, while words that occur together in text often have associated meanings, these meanings may be very similar or very different. Thus, *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.

While text-based methods for inferring word representations have undoubtedly improved over the last 50 years, the underlying connections between the algorithms suggests that the principal cause of this improvement is simply access to larger text corpora. Looking further ahead, however, there are at least intuitive reasons why models that learn exclusively from static text corpora may never display a human-level mastery of lexical semantics. Much conceptual knowledge, and particularly that concerning concrete concepts, is grounded in the perceptual modalities as much as in language (Barsalou et al., 2003). At the same time, inferring deeper conceptual properties of abstract words, such as their subjectivity or affect, may require a much more active language learning environment in which the examples observed by a model depends on its previous output (allowing it to learn about the communicative or behavioural consequences of words). Whatever the means by which the improvements are achieved, the ability to acquire concept-level representations that closely align with human cognition is likely to be a crucial part of progress in many directions of NLP and language understanding research.

Chapter 3

Embedding word similarity with neural machine translation

Can text-based models of word representation be improved in order to achieve human-like performance on evaluations like SimLex-999? It is certainly plausible that models may never acquire human-quality word representations from text, however efficient the learning algorithm and however much data they observe. This is because text data as a learning resource lacks various characteristics of the information available to human learners. For instance, much of conceptual acquisition and word learning, particularly at the early stages, apparently involves the unification of linguistic concepts with those acquired via the perceptual system (Barsalou and Wiemer-Hastings, 2005). In addition, as the conceptual system develops, humans actively learn via interactions that depend on the current output of the learner, or explicit explanations targeted at the learner, or following a curriculum. Information in this form is not available to the text-based learning algorithms described in the previous chapter.

In this and the following chapters, I consider whether the distributed representations acquired by NLMs can be improved via training with more sophisticated prediction tasks, in may cases leveraging training data with more inherent structure than plain raw text. This chapter focuses specifically on word representations, before I extend the scope to phrases and sentences in chapters 4-refCH6.

In the remainder of this chapeter, I show how enhanced word representations can be acquired from bilingual text data, using recently-developed deep sequence-to-sequence learning architectures trained to translate between pairs of European languages. These architectures constitute the first end-to-end neural machine translation (NMT) systems.

NMT models have emerged in the past couple of years as an alternative to statistical, phrase-based translation models, and are beginning to achieve impressive translation performance (Kalchbrenner and Blunsom, 2013; Devlin et al., 2014; Sutskever et al., 2014).

As I show, NMT models are an effective tool for learning word embeddings with particular semantic properties, as well as a potential new direction for machine translation. Specifically, NMT word embeddings are orientated to reflect semantic similarity (rather than non-specific relatedness or association) and lexical-syntactic role more effectively than embeddings acquired by monolingual NLMs. I 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, I conjecture as to why similarity dominates over other semantic relations in translation embedding spaces. Finally, I 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.

3.1 Encoder-decoder models for Neural Machine Translation

The objective of NMT models is to generate an appropriate sentence in a target language S_t given a sentence S_s in the source language (see e.g. (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014)). 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.[1] 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

[1] Alternatively, subsequences (phrases) of S_s may be encoded at this stage in place of the whole sentence (Bahdanau et al., 2015).

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 (Kalchbrenner and Blunsom, 2013; Cho et al., 2014b; Bahdanau et al., 2015), the translation objective exerts similar pressure on the word embeddings in all cases. The source language embeddings must be such that the model can combine them to form single 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.

3.2 Other bilingual models of learning word representations

Before the advent of effective end-to-end NMT systems, several models had been developed with the specific goal of acquiring distributed word representations from bilingual corpora, aligned at the document, paragraph or word level (Haghighi et al., 2008; Vulić et al., 2011; Mikolov et al., 2013b; Hermann and Blunsom, 2014; Chandar et al., 2014). While these approaches rely on the same training data as NMT models, one important difference is that they represent the words from two different languages in single common vector space so that words in one language are close to words with similar or related meanings in the other (this is not the case for NMT models, where the source and target language word embeddings inhabit distinct vector spaces). The resulting multilingual embedding spaces have been effectively applied to bilingual lexicon extraction (Haghighi et al., 2008; Vulić et al., 2011; Mikolov et al., 2013b) and document classification (Klementiev et al., 2012; Hermann and Blunsom, 2014; Chandar et al., 2014; Kočiský et al., 2014).

For comparison, we focus on two representatives of this class of (non-NMT) bilingual model. The first is that of Hermann and Blunsom (2014), whose embeddings improve on the performance of Klementiev et al. (2012) in document classification applications. As with NMT models, 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 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. Hermann and Blunsom (2014) show that, despite the lack of prespecified word alignments, words in the two languages with similar meanings converge in the bilingual embedding space.[2]

The second model I examine is that of Faruqui and Dyer (2014). Unlike the models described above, Faruqui and Dyer (2014) 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 Faruqui and Dyer (2014) requires bilingual data to be aligned at the word level.

3.3 Experiments

To learn translation-based embeddings, I trained two different NMT models. The first is the RNN encoder-decoder, *RNNenc* (Cho et al., 2014b), which uses a recurrent neural network (RNN) 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 (Bahdanau et al., 2015), which was designed to overcome limitations exhibited by the RNN encoder-decoder when translating very long sentences. RNN Search includes an *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.[3] Both models were trained on a 348m word corpus of English-French sentence pairs or a 91m word corpus of English-German sentence pairs.[4]

To explore the properties of bilingual embeddings learned via objectives other than direct translation, I trained the *BiCVM* model of Hermann and Blunsom (2014) on the same data, and also downloaded the projected embeddings of Faruqui and Dyer

[2] The models of Chandar et al. (2014) and Hermann and Blunsom (2014) both aim to minimise the divergence between source and target language sentences represented as sums of word embeddings. Because of these similarities, I do not compare with both in this paper.

[3] Access to source code and limited GPU time prevented me from training and evaluating the embeddings from other NMT models such as that of Kalchbrenner and Blunsom (2013), Devlin et al. (2014) and Sutskever et al. (2014). The underlying principles of encoding-decoding also apply to these models, and I expect the embeddings would exhibit similar properties to those analysed here.

[4] These corpora were produced from the WMT '14 parallel data after conducting the data-selection procedure described by Cho et al. (2014b).

(2014), *FD*, trained on a bilingual corpus of comparable size (≈ 300 million words per language).[5] Finally, to compare with monolingual word embedding models, I trained a conventional skipgram model (Mikolov et al., 2013c) and its *Glove* variant (Pennington et al., 2014a) 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, I then trained the monolingual models on increasingly large random subsamples of Wikipedia text (up to a total of 1.1bn words). Lastly, I extracted embeddings from a full-sentence language model, *CW*, (Collobert and Weston, 2008), which was trained for several months on the same Wikipedia 1bn word corpus. Note that increasing the volume of training data for the bilingual (and NMT) models was not possible because of the limited size of available sentence-aligned bitexts.

3.3.1 Similarity and relatedness

As in previous studies (Agirre et al., 2009a; Bruni et al., 2014; Baroni et al., 2014b), the 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 I used three different gold standards: WordSim-353 (Agirre et al., 2009a), MEN (Bruni et al., 2014) and SimLex-999 (Hill et al., 2015b). Recall that 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.

Table 3.1 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 Faruqui

[5] 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.

		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 3.1: 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 3.2: 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.

and Dyer (2014), 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, I ran three more evaluations designed to probe the sensitivity of models to similarity as distinct from relatedness or association. In the first, I measured performance on SimLex-Assoc-333 (Hill et al., 2015b). This evaluation comprises the 333 most related pairs in SimLex-999, according to an independent empirical measure of relatedness (free associate generation (Nelson et al., 2004)). Importantly, the pairs in SimLex-Assoc-333, while all strongly related, still span the full range of similarity scores.[6] 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 (Landauer and Dumais, 1997). I 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.[7] Note that, since TOEFL is a test of synonym recognition, it necessarily requires models to recognise similarity as opposed to relatedness.

Finally, I 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,[8] I presented a Gaussian SVM with the concatenation of the two word embeddings. I evaluated accuracy using 10-fold cross-validation.

As shown in Table 3.1, with these three additional similarity-focused tasks the same pattern of results is observed. 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 3.2, 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*.

[6] 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.

[7] To control for different vocabularies, I restricted the effective vocabulary of each model to the intersection of all model vocabularies, and excluded all questions that contained an answer outside of this intersection.

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

Effect of training data quantity In previous work, monolingual models were trained on corpora many times larger than the English half of the parallel translation corpus. Indeed, the ability to scale to large quantities of training data was one of the principal motivations behind the skipgram architecture (Mikolov et al., 2013c). To check if monolingual models simply need more training data to capture similarity as effectively as bilingual models, I therefore trained them on increasingly large subsets of Wikipedia.[9] As shown in Figure 3.1, 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 bilingual embeddings as the amount of training data increases.

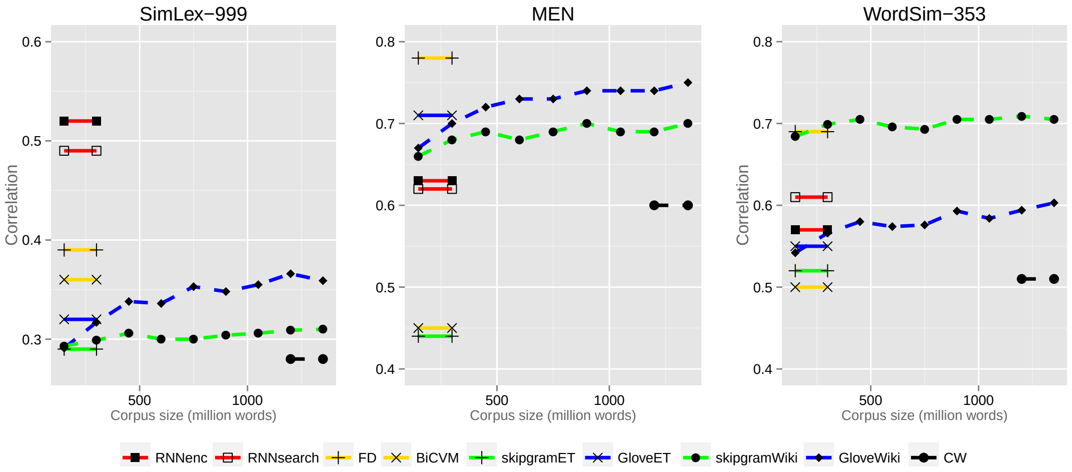


Figure 3.1: The effect of increasing the amount of training data on the quality of monolingual embeddings, based on similarity-based evaluations (SimLex-999) and two relatedness-based evaluations (MEN and WordSim-353). *ET* in the legend indicates models trained on the English half of the translation corpus. *Wiki* indicates models trained on Wikipedia.

3.3.2 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 (Mikolov et al., 2013c). This is because, if $\mathbf{m}, \mathbf{b}$ and $\mathbf{w}$ are skigram-style em-

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

beddings 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 Mikolov et al. (2013c). As in the previous section, for fair comparison I 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. 3.2, 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).[10] 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

[10] The performance of the FD embeddings on this task is higher than that reported by Faruqui and Dyer (2014) because I search for answers over a smaller total candidate vocabulary.

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 (Levy and Goldberg, 2014a), the lexical syntactic and conceptual properties of translation embeddings are in some sense independent of one another.

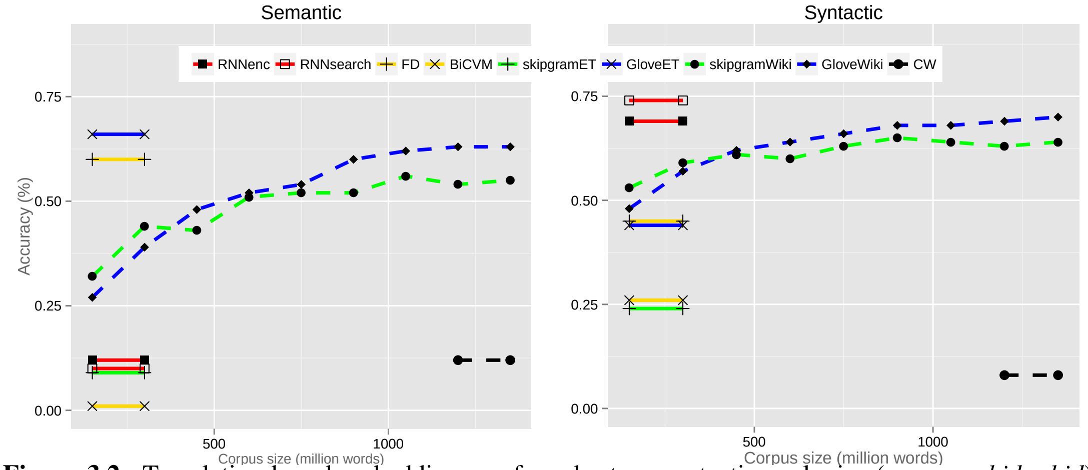


Figure 3.2: Translation-based embeddings perform best on syntactic analogies (*run,ran: hide, hid*). Monolingual skipgram/Glove models are better at semantic analogies (*father, man; mother, woman*)

3.4 Effect of target language

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

As shown in Table 3.3 (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.

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

Table 3.3: 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.

3.5 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 (Morin and Bengio, 2005) or approximate methods such as negative sampling (Mikolov et al., 2013c; Hermann and Blunsom, 2014), 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 (Jean et al., 2015). The algorithm uses (biased) importance sampling (Bengio and Sénécal, 2003) to approximate the probability distribution of words over a large target vocabulary

		RNN Search EN-FR	RNN Search EN-DE	RNN Search-LV EN-FR	RNN Search-LV 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 3.4: 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.

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, I tested how the quality of source embeddings is affected by scaling up the target language vocabulary in this way.

As shown in Table 3.4, there is no significant degradation of embedding quality when scaling to large vocabularies with using the approximate decoder. Note that for a fair comparison I 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 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/`.[11]

[11] A different solution to the rare-word problem was proposed by Luong et al. (2015b). I do not evaluate the effects on the resulting embeddings of this method because I lack access to the source code.

3.6 Discussion

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 3.3 (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 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, I 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 targThe horse raced past the barn fellet 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, I 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 Faruqui and Dyer (2014) that projecting monolingual embeddings into a bilingual space orientates them to better reflect the synonymy/antonymy distinction.

3.7 Conclusions

The analysis in this chapter makes it clear that not all distributed word representations learned by neural language models are alike. I 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 (Hill et al., 2015b). The semantic correspondence between the words in languages enables NMT embeddings to distinguish synonyms and antonyms, and also facilitates better performance on syntactic analogical reasoning. Just as being exposed to multiple languages can be an advantage for both the cognitive and cultural development of humans, training NLMs on bilingual texts can allow them to acquire a perspective on lexical semantics that may not otherwise emerge.

The language-specific effects I observed in Section 3.4 suggest a potential avenue for improving translation and multi-lingual embeddings in future work. As observed by both Hermann and Blunsom (2014) and Faruqui and Dyer (2014), 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 (Kočiský et al., 2014).

Together with a growing body of work, this chapter demonstrates that a focus on word representation can provide an instructive constrained environment for understanding how conceptual knowledge is encoded in distributed memories. Nevertheless, the majority of linguistic meaning is communicated not by individual words, but in the form of statements, utterances, propositions or observations that consist of multiple words, phrases or sentences. When interpreting such phrases, the interpretation of words always takes place in some wider context Marslen-Wilson and Tyler (1980), and it is based on such context-dependent interpretations that the utterance is understood. For technology to be capable of general language understanding, systems will undoubtedly require ways to interpret and represent the phrases and sentences as well as words. To this end, in the next chapter I develop a framework and various approaches for training neural language models to interpret phrases, which are then extended to full sentences in Chapter 5.

Chapter 4

Representing phrases with neural language models

Notwithstanding its clear importance to the overarching goal of general language understanding, the task of extending representation-learning techniques from words to larger linguistic chunks is extremely challenging. Many of the original approaches to this problem involved two (independent) stages. First, distributed representations of words were acquired using established techniques. Second, phrase representations were computed by applying fixed mathematical (vector) operations to the representations of words appearing in those phrases (Clark and Pulman, 2007; Mitchell and Lapata, 2008; Baroni et al., 2014a). An obvious limitation of such approaches is that they have no capacity to learn from data about how word meanings are influenced by their context, nor how word meanings in context combines to influence phrase meaning. Indeed, given that both phenomena are (in)famously complex, (Prinz, 2004; Fodor, 2004; Fauconnier, 1994) it seems unlikely that they could be modelled effectively by a single fixed mathematical operation.

Baroni and Zamparelli (2010) proposed a way to overcome this limitation in the context of adjective-noun (AN) combinations such as *red car*. The first stage of their method involved learning word-like representations (n_j and an_{ij}) for both nouns and AN combinations using traditional distributional methods (in this case using a model similar to the VSM described in Chapter 2, except in which AN bigrams are treated as single word-like entities). In the second stage, for each adjective in their study (indexed by i), a matrix A_i was learned via a least-squares regression algorithm such that, for each n_j, the (matrix-vector) product $A_i n_j$ was as close as possible to an_{ij}.

While the approach of Baroni and Zamparelli (2010) overcame certain shortcomings with existing methods, their approach still suffers from important limitations. First, while the strategy of using co-occurence counts (or indeed predictions) of neighbouring words to learn distributed representations of multi-word chunks is viable for two-word phrases, it is not known whether such a strategy would ever work for phrases or sentences (even assuming text corpora many orders of magnitude large than the largest available today). Second, in these models the process of acquiring representations of the noun and the AN unit is independent from that of learning the adjective matrix, whereas it may be desirable that, for instance, a noun representation encodes some information about how it behaves upon combination with other words. Third, the method requires an empirically valid mapping of words in a language to parts-of-speech (and, ultimately, the specification of analagous methods for other word-types), which is very challenging to achieve when considering the vast differences in information encoding across the languages of the world.

In this chapter, I propose a way of overcoming some of these limitations by using neural language models (NLMs) in conjunction with readily available data from dictionaries. The intuition is that 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 I experiment with two broad classes of 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, I learn target lexical representations by training the Word2Vec software (Mikolov et al., 2013c) on billions of words of raw text.

I demonstrate the utility of this 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 (Zock and Bilac, 2004). 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*. I show that a 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 greater task-specific engineering and access to much more dictionary data. Moreover, by exploiting models that learn bilingual word representations (Vulic et al., 2011; Klementiev et al., 2012; Hermann and Blunsom, 2013; Gouws et al., 2014), I show that the NLM approach can be easily extended to produce a potentially useful cross-lingual reverse dictionary.

The second application of the 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 for clues containing more than four 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 the code, training and evaluation sets (together with a system demo) are published online with this paper.[1]

While the success of these applications shows that the representations learned via the proposed approach can facilitate a useful degree of language understanding, this understanding relates to specific tasks. In Chapter 5, however, I demonstrate, via comparisons with many other NLMs on a wider selection of tasks and evaluations, that the same approach is equally promising for semantic representation and language understanding in general sense.

4.1 Neural language model architectures

The first model I apply to the dictionary-based learning task is a recurrent neural network (RNN). RNNs operate on variable-length sequences of inputs; in the case, natural language definitions, descriptions or sentences. RNNs (with LSTMs) have achieved state-of-the-art performance in language modelling (Mikolov et al., 2010), image caption generation (Kiros et al., 2015a) and approach state-of-the-art performance in machine translation (Bahdanau et al., 2015).

During training, the input to the RNN is a dictionary definition or sentence from an

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

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 (Mikolov et al., 2013c).

The set of all words in the training data constitutes the vocabulary of the RNN. For each word in this vocabulary I 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 the implementation, I 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.

4.1.1 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 (Bengio et al., 1994). 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 (Hochreiter and Schmidhuber, 1997) 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):

$$i_t^w = W_w v_t + U_w h_{t-1} + b_w$$

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 (I 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.

4.1.2 Bag-of-words NLMs

I 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:

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

4.1.3 Pre-trained input representations

I 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'.

4.1.4 Training objective

I 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. I 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, by Huang et al. (2012). In all experiments I apply a margin $m = 0.1$, which has been shown to work well on word-retrieval tasks (Bordes et al., 2015).

4.1.5 Implementation details

Since training on the dictionary data took 6-10 hours, I did not conduct a hyperparameter search on any validation sets over the space of possible model configurations such as embedding dimension, or size of hidden layers. Instead, I 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 the 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. (Faruqui et al., 2014)). The pre-trained embeddings used in the 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, I 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. (Cho et al., 2014a)). 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 (Bergstra et al., 2010) and trained with minibatch SGD on GPUs. The batch size was fixed at 16 and the learning rate was controlled by *adadelta* (Zeiler, 2012).

4.2 Reverse dictionaries

The most immediate application of the 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 def-

[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.

inition.[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*. the trained RNN model can perform a similar function, simply by mapping a phrase to a point in the target (Word2Vec) embedding 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 (Bilac et al., 2003, 2004; Zock and Bilac, 2004). Proximity is quantified differently in each case, but is generally a function of hand-engineered features of the two sentences. For instance, Shaw et al. (2013) 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.

4.2.1 Data collection and training

To compile a bank of dictionary definitions for training the model, I started with all words in the target embedding space. For each of these words, I extracted dictionary-style definitions from five electronic resources: *Wordnet, The American Heritage Dictionary, The Collaborative International Dictionary of English, Wiktionary* and *Webster's*. I chose these five dictionaries because they are freely-available via the WordNik

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

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

API,[5] but in theory any dictionary could be chosen. Most words in the training data had multiple definitions. For each word w with definitions $\{d_1 \dots d_n\}$ I 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, I supplemented this training data with information extracted from Simple Wikipedia. [6] For every word in the model's target embedding space that is also the title of a Wikipedia article, I 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, I 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, I 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, I therefore do not present these results in what follows.

4.2.2 Comparisons

As a baseline, I also implemented two entirely unsupervised methods using the neural (Word2Vec) word embeddings from the target word space. In the first (**W2V add**), I 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 (Mitchell and Lapata, 2010).

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

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

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

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

None of the models or evaluations from previous academic research on reverse dictionaries is publicly available, so direct comparison is not possible. However, I do compare performance with the commercial systems. The Dictionary.com system returned no candidates for over 96% of the input definitions. I 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.

4.2.3 Reverse dictionary evaluation

To the knowledge there are no established means of measuring reverse dictionary performance. In the only previous academic research on English reverse dictionaries that I are aware of, evaluation was conducted on 300 word-definition pairs written by lexicographers (Shaw et al., 2013). Since these are not publicly available I 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, I 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, I 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, I built a new dataset of **concept descriptions** that do not appear in the training data for any model. To do so, I randomly selected 200 adjectives, nouns or verbs from among the top 3000 most frequent tokens in the British National Corpus (Leech et al., 1994) (but outside the top 100). I 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 I asked two participants 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 the website for future comparisons.

Given a test description, definition, or question, all models produce a ranking of

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

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

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, I 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).

4.2.4 Results

Table 4.2 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 4.2 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]

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 cor-

[10] The trained neural language models are approximately half the size of the six training dictionaries stored as plain text, so would be hundreds of times smaller than the OneLook database of 1061 dictionaries if stored this way.

	Dictionary definitions								
Model	**Seen** (500 WN defs)			**Unseen** (500 WN defs)			**Concept descriptions** (200)		
OneLook	**0**	**.89/.91**	**67**	-	-	-	**18.5**	**.38**/.58	153
W2V add	-	-	-	923	.04/.16	163	339	.07/.30	150
W2V mult	-	-	-	1000	.00/.00	10*	1000	.00/.00	27*
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 4.2: Performance of different reverse dictionary models in different evaluation settings. *Low variance in *mult* models is due to consistently poor scores, so not highlighted.

rect 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 the I 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 (Iyyer et al., 2015).

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

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

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.

4.2.5 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 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.

4.2.6 Cross-lingual reverse dictionaries

The models developed thus far 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

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

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, I 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 (Hermann and Blunsom, 2013; Chandar et al., 2014; Gouws et al., 2014). For a test-of-concept experiment, I used English-French embeddings learned by the state-of-the-art BilBOWA model (Gouws et al., 2014) from the Wikipedia (monolingual) and Europarl (bilingual) corpora.[12] I trained the RNN model to map from English definitions to English words in the bilingual space. At test time, after reading an English definition, I then simply return the nearest French word neighbours to that definition.

Because no benchmarks exist for quantitative evaluation of bilingual reverse dictio-

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

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

naries, I 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, I 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-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).

4.2.7 Discussion

I 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. I also showed how the architecture can be easily extended to produce bilingual versions of the same model.

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

4.3 Answering crossword questions

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 (Ferrucci et al., 2010). 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 (Mollá and Vicedo, 2007), allowing systems access to pre-specified passages of text from which the answer can be inferred (Iyyer et al., 2014; Weston et al., 2015a), or centering both questions and answers on a particular knowledge base (Berant and Liang, 2014; Bordes et al., 2014).

In what follows, I 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 the architecture, I 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]

4.3.1 Evaluation

General Knowledge crossword questions come in different styles and forms. I 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. the **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, I 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. I again formed a list of 150 questions, beginning on 1 January 2015 and excluding any questions with multiple-word answers. For clear contrast, I 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, I only consider candidate words *whose length matches the length specified in the clue*.

Test set	Word	Description
Long (150)	*Baudelaire*	”French poet, 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.

[13] As the interest is in the language understanding, I do not address the question of fitting answers into a grid, which is the main concern of end-to-end automated crossword solvers (Littman et al., 2002).

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

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.

4.3.2 Benchmarks and comparisons

As with the reverse dictionary experiments, I compare RNN and BOW NLMs with a simple unsupervised baseline of elementwise addition of Word2Vec vectors in the embedding space (I discard the ineffective *W2V mult* baseline), again restricting candidates to words of the pre-specified length. I 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 (Littman et al., 2002). 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 (I focus only on the non-cryptic setting), and has also been featured in national media.[15] I are unable to compare against a third well-known automatic crossword solver, *Dr Fill* (Ginsberg, 2011), because code for Dr Fill's candidate-generation module is not readily available. As with the RNN and baseline models, when evaluating existing systems I discard candidates whose length does not match the length specified in the clue.

Certain principles connect the design of the existing commercial systems and dif-

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

ferentiate them from the 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 (Littman et al., 2002; Ginsberg, 2011).

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

Table 4.7: Responses from different models to example crossword clues. In each case the model output is filtered to include only candidates with the same number of letters as the correct answer (in brackets). BOW and RNN models are trained without Word2Vec input embeddings and cosine loss.

4.3.3 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, I 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, I 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 difference 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) (Mikolov et al., 2013c). 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.

4.3.4 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 (Bordes et al., 2014; Graves et al., 2014; Weston et al., 2015a).

4.4 Conclusion

Dictionaries exist in many of the world's languages. I have shown how these lexical resources can constitute valuable data for training the neural language models to interpret and represent the meaning of phrases. 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. I 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 exploit this idea explicitly. The code and training data for both models is available online, and trained models can be queried and compared with baselines on the web demo.[16]

On the reverse dictionary task that mirrors their 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. I also showed how knowledge encoded in this way into distributed representations can transfer to applications beyond the setting in which the model was originally trained. NLMs trained on the same data can answer general-knowledge crossword questions, and indeed outperform commercial systems on longer questions. While the QA experiments focused on crosswords, the results

[16] See `https://github.com/fh295/DefGen2` for the code, `http://www.cl.cam.ac.uk/~fh295/` for the data and `http://45.55.181.170/defgen/` for the demo

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. There are many avenues to develop and extend the approach introduced here in future work. One question of particular interest is the apparent success of BOW models that lack ‘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.

Moving beyond the two specific applications considered in this chapter, the plausibility of the output produced by the trained NLMs suggests that dictionary-based training may provide one way of inducing general-purpose (task-agnostic) distributed representations of phrases. In the next chapter, I test this hypothesis explicitly, using a more diverse set of benchmarks and tasks to compare the phrase representations of the dictionary-based NLMs with approaches covering a range of architectures, training data and objectives.

Chapter 5

Representing sentences with neural language models

As detailed in Chapters 2 and 3, there are established ways to acquire distributed word representations from naturally occurring (unlabelled) training data based on comparatively task-agnostic objectives (such as predicting adjacent words). Such methods are well understood empirically (Baroni et al., 2014b) and theoretically (Levy and Goldberg, 2014b). The best word representation spaces reflect consistently-observed aspects of human conceptual organisation (Hill et al., 2015b), and can be added as features to improve the performance of numerous language processing systems (Collobert et al., 2011).

However, as discussed in Chapter 4, the task of learning such representations for longer linguistic units such as phrases or sentences is far harder, and there is comparatively little consensus on the best ways to attack this problem.[1] Nevertheless, with the growth of available data and improvements in algorithms and infrastructure, NLMs have emerged that do indeed compute internal representations of whole sentences as continuous-valued vectors. Sch NLMs have been applied to machine translation (Sutskever et al., 2014), image captioning (Mao et al., 2015) and dialogue systems (Serban et al., 2015). While it has been observed informally that the internal sentence representations of these models can reflect semantic intuitions (Cho et al., 2014a), it is not known how to induce representations that are high quality in a general sense. Resolving this question could ultimately have a significant impact on language

[1] See the contrasting conclusions in (Mitchell and Lapata, 2008; Clark and Pulman, 2007; Baroni et al., 2014a; Milajevs et al., 2014) among others.

processing systems. Indeed, it is phrases and sentences, rather than individual words, that encode the human-like general world knowledge (or 'common sense') (Norman, 1972) that is a critical missing part of most current language understanding systems.

In this chapter, I address these questions by means of a systematic comparison of the distributed phrase and sentence representations acquired by cutting-edge NLMs. I focus on 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. I 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. I 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.

I 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 (Kiros et al., 2015b) are most effective for 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 (Hill et al., 2015a). Taken together, these findings constitute valuable guidelines for the application of phrasal or sentential representation-learning to language understanding systems.

5.1 Distributed sentence representations

To constrain the analysis, I compare neural language models that compute sentence representations from unlabelled, naturally-occurring data, as with the predominant methods for word representations.[2] Likewise, I do not focus on 'bottom up' models

[2] This excludes innovative supervised sentence-level architectures such as those of Socher et al. (2011),Kalchbrenner et al. (2014) and many others.

where phrase or sentence representations are built from fixed mathematical operations on word vectors (although I do consider a canonical case - see CBOW below); these were already compared by Milajevs et al. (2014). Most space is devoted to the novel approaches, and I refer the reader to the original papers for more details of existing models.

5.1.1 Existing models trained on text

SkipThought Vectors For consecutive sentences S_{i-1}, S_i, S_{i+1} in some document, the **SkipThought** model (Kiros et al., 2015b) 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 Units (Cho et al., 2014a)) 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 Le and Mikolov (2014) 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}.

I 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).

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

Bottom-Up Methods I train **CBOW** and **SkipGram** word embeddings (Mikolov et al., 2013c) on the Books corpus, and compose by elementwise addition as proposed by Mitchell and Lapata (2010).[4]

I also compare to **C-PHRASE** (Pham et al., 2015), an approach that exploits a (supervised) parser to infer distributed semantic representations based on a syntactic parse of sentences. C-PHRASE achieves state-of-the-art results for distributed representations on several evaluations used in this study.[5]

Non-Distributed Baseline I 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.

5.1.2 Models trained on structured resources

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

DictRep Hill et al. (2015a) 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. I implement their models using the available code and training data.[6]

CaptionRep Using the same overall architecture, I trained (**BOW** and **RNN**) models to map captions in the COCO dataset (Chen et al., 2015) to pre-trained vector representations of images. The image representations were encoded by a deep convolutional network (Szegedy et al., 2014) trained on the ILSVRC 2014 object recognition task (Russakovsky et al., 2014). Multi-modal distributed representations can be encoded by feeding test sentences forward through the trained model.

NMT I consider the sentence representations learned by neural MT models. These models have identical architecture to SkipThought, but are trained on sentence-aligned

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

[5] Since code for C-PHRASE is not publicly-available I use the available pre-trained model (`http://clic.cimec.unitn.it/composes/cphrase-vectors.html`). Note this model is trained on $3\times$ 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.

translated texts. I used a standard architecture (Cho et al., 2014a) on all available **En-Fr** and **En-De** data from the 2015 Workshop on Statistical MT (WMT).[7]

5.1.3 Novel text-based models

I 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, I 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 (Vincent et al., 2008). Transforming data into DAE representations (as a 'pre-training' or initialisation step) gives more robust (supervised) classification performance in deep feedforward networks (Vincent et al., 2010).

The original DAEs were feedforward nets applied to (image) data of fixed size. Here, I 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. I 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. I call this model the *Sequential Denoising Autoencoder* (**SDAE**). Note that, unlike SkipThought, SDAEs can be trained on sets of sentences in arbitrary order.

I 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 Dai and Le (2015). The 'word dropout' effect when $p_o \geq 0$ has also been used as a regulariser for deep nets in supervised language tasks (Iyyer et al., 2015), and for large p_x the objective is similar to word-level 'debagging' (Sutskever et al., 2011). For the SDAE, I tuned p_o, p_x on the

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

validation set (see Section 5.2.2).[8] I 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 semantics can be inferred from the content of adjacent sentences. The model could be said to exploit a type of *sentence-level Distributional Hypothesis* (Harris, 1954; Polajnar et al., 2015). 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.

I 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 $\mathbf{s} = \sum_{w \in S} u_w$.

5.1.4 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.

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

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

	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	$/5$
STS 2014	News	*Mexico wishes to guarantee citizens' safety.*	*Mexico wishes to avoid more violence.*	4
	Forum	*The problem is simpler than that.*	*The problem is simple.*	3.8
	WordNet	*A social set or clique of friends.*	*An unofficial association of people or groups.*	3.6
	Twitter	*Taking Aim #Stopgunviolence #Congress #NRA*	*Obama, Gun Policy and the N.R.A.*	1.6
	Images	*A woman riding a brown horse.*	*A young girl riding a brown horse.*	4.4
	Headlines	*Iranians Vote in Presidential Election.*	*Keita Wins Mali Presidential Election.*	0.4
SICK (test+train)		*A lone biker is jumping in the air.*	*A man is jumping into a full pool.*	1.7

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

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 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.

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

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

5.2 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* (Collobert et al., 2011; Mikolov et al., 2013a; Kiros et al., 2015b) - or by comparing with human relatedness judgements - *unspervised evaluation* (Hill et al., 2015a; Baroni et al., 2014b; Levy et al., 2015a). The former setting reflects a scenario in 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, I apply and compare both evaluation paradigms.

Data	Model	MSRP (Acc / F1)	MR	CR	SUBJ	MPQA	TREC
Unordered Sentences (Toronto Books: 70m sents, 0.9B words)	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
	SDAE	**76.4 / 83.4**	67.6	74.0	89.3	81.3	77.6
	SDAE+embs.	73.7 / 80.7	**74.6**	**78.0**	**90.8**	**86.9**	78.4
	ParagraphVec DBOW	72.9 / 81.1	60.2	66.9	76.3	70.7	59.4
	ParagraphVec DM	73.6 / 81.9	61.5	68.6	76.4	78.1	55.8
	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 Sentences (Toronto Books)	SkipThought	**73.0 / 82.0**	**76.5**	80.1	**93.6**	**87.1**	**92.2**
	FastSent	72.2 / 80.3	70.8	78.4	88.7	80.6	76.8
	FastSent+AE	71.2 / 79.1	71.8	76.7	88.8	81.5	80.4
Other structured data resource	NMT En to Fr	69.1 / 77.1	64.7	70.1	84.9	81.5	**82.8**
	NMT En to De	65.2 / 73.3	61.0	67.6	78.2	72.9	81.6
	CaptionRep BOW	73.6 / 81.9	61.9	69.3	77.4	70.8	72.2
	CaptionRep RNN	72.6 / 81.1	55.0	64.9	64.9	71.0	62.4
	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 5.2.1). Bold numbers indicate best performance in class. Underlined indicates best overall.

5.2.1 Supervised evaluations

Representations are applied to 6 sentence classification tasks: paraphrase identification (MSRP) (Dolan et al., 2004), movie review sentiment (MR) (Pang and Lee, 2005),

	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 5.2.2). Models are grouped according to training data as indicated in Table 5.3.

product reviews (CR) (Hu and Liu, 2004), subjectivity classification (SUBJ) (Pang and Lee, 2004), opinion polarity (MPQA) (Wiebe et al., 2005) and question type classification (TREC) (Voorhees, 2002). I follow the procedure (and code) of Kiros et al. (2015b): 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.

5.2.2 Unsupervised evaluations

I 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 (Marelli et al., 2014) consists of 10,000 pairs of sentences and relatedness judgements. The STS 2014 dataset (Agirre et al., 2014) 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.

5.3 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 (Ji and Eisenstein, 2013). SDAE is also consistently better than SAE, which aligns with other findings that adding noise to AEs produces richer representations (Vincent et al., 2008).

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 (Pham et al., 2015). 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.

5.4 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) (Voorhees, 2002). 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 (Bruni et al., 2014).

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 Almahairi et al. (2015), 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.

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, becoming 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 vision blurred at the edges, threatening my consciousness altogether.*
FastSent	*Isak's close enough to pick off 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 the sidewalk beneath my feet began to tremble even more.*
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 down the stairs.*
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 ears.*
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.

Knowledge transfer shows some promise It is notable that, with a few exceptions, the models with pre-trained word embeddings (+embs) outperform those with learned embeddings on both supervised and unsupervised evaluations. In the case of the DictRep models, whose training data is otherwise limited to dictionary definitions, this effect can be considered as a rudimentary form of knowledge transfer. The DictRep+embs model benefits both from the dictionary definition data and the enhanced lexical semantics acquired from a massive text corpus to build overall higher-quality sentence representations. In the context of this thesis, this is an important observation, as it justifies the pursuit of task-agnostic, general-purpose semantic representations as a useful means of injecting world knowledge into downstream language understanding systems.

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 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 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 (Wiebe et al., 2005)), 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 the knowledge, there are no existing benchmarks that overcome these limitations.

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

Overall consistency, $\alpha = 0.90$

MSRP	MR	CR	SUBJ	MPAQ	TREC
0.94 (6)	0.85 (1)	0.86 (4)	0.85 (1)	0.86 (3)	0.89 (5)

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

Overall consistency, $\alpha = 0.93$

News	Forum	WordNet	Twitter	Images	Headlines	All STS	SICK
0.92 (4)	0.92 (3)	0.92 (4)	0.93 (6)	0.95 (8)	0.92 (2)	0.91 (1)	0.93 (7)

Table 5.7: Internal consistency (Chronbach's α) among unsupervised evaluations when individual benchmarks are left out of the cohort.

5.5 Conclusion

Advances in deep learning algorithms, software and hardware mean that many architectures and objectives for learning distributed sentence representations from unlabelled data have recently become available to NLP researchers. This chapter is the first systematic comparison of these methods. I showed notable variation in the performance of approaches across a range of evaluations. Among other conclusions, I 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.

The second contribution of this chapter is two new objectives for learning distributed sentence representations: FastSent and Sequential Denoising Autoencoders. These models have certain practical advantages over the alternatives, and 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+embs: learning to compose pre-trained word embeddings into rich sentence representations via the word-phrase signal in dictionary definitions.

Such a study has only recently become viable thanks to recent progress in training deeper neural language models, and many interesting questions concerning sentential representation remain to be resolved. These include:

[13] Code for training and evaluating these new models is available at `https://github.com/fh295/SentenceRepresentation.git`. An online demo of the FastSent sentence space can be found at `http://45.55.60.98/`.

What is the optimal representation 'scope'? Evidence from cognitive science Fillmore (1985) and neuroscience Ding et al. (2016) suggests that humans compute relatively stable representations of 'frames' or 'scenes', which correspond to something like the action of a verb on one or more nouns. Sentences in general refer to many such scenes. It may ultimately be preferable to use methods similar to those studied in this chapter to acquire representations of scenes, and other machinery to inform the combination of these scenes, a conclusion further supported by experiments in Chapter 6. However, it is not clear how to prescribe such a strategy in practice, because the scope of the scenes in question is not encoded explicitly in text, whereas a full stop clearly delimits sentences.

What, if anything, is the correct model prior for sentences? A BOW model is blind to word order whereas an RNN encodes sequential structure. When training models on labelled data in a supervised setting, results show that either a hard (i.e. following pre-specified grammar rules) (Socher et al., 2013) or soft (i.e. learned, possibly convolutional) (Kalchbrenner et al., 2014) binary tree structure may be a more appropriate prior for capturing sentential meaning (although these effects are very marginal). We do not examine such networks here because it is not clear how to apply them in the context of unlabelled data.

As a cursory glance at Table 5.2 shows, the satisfactory resolution of these questions may require much more robust benchmarks and evaluations with which to compare models, and consequently they are beyond the scope of this thesis. Nonetheless, in the final study of this thesis, described in the next chapter, I apply an alternative way of studying sentence representations is applied that is less 'instrinsic' in nature, and therefore circumvents some of the problematic evaluation issues noted above. First, an overarching model architecture - with relatively weak prior structural assumptions - is constructed to exploit sentence representations in the resolution of a comparatively canonical and well-defined (extrinsic) downstream language prediction task. Then, the underlying nature of the sentence-representation algorithm is varied, and the ultimate effect on the downstream prediction task is measured.

Chapter 6

Representing and interpreting sentence semantics in context

In this chapter, I analyse the effect of different ways of computing distributed representations of words, phrases and sentences on the downstream performance of *memory networks* (Weston et al., 2015b). Memory networks are a relatively novel class of neural network that have been developed since 2014 and applied to language and reasoning problems (Weston et al., 2015a). These models are characterised not simply by their depth or quantity of internal representations, but by the explicit way in which they recruit information from those representations in order to complete their ultimate objective, weighting each representation in memory according to relevance. The component that computes the weights corresponding to each internal representation is typically referred to as an *attention* mechanism.

Neural networks with attention mechanisms are interesting for both scientific and engineering reasons. They can be understood as a rudimentary model of the poorly understood process of accessing and retrieving semantic memories in the human brain, a process that undoubtedly plays a critical part in human language understanding. Moreover, the attention mechanisms of trained networks can be interrogated to give a clear picture of which representations (and hence which information) is most relevant and useful to the network in the satisfaction of its objective, which can in turn facilitate a better understanding of the data and how networks can be modified to deliver improved performance.

Deep architectures with attention mechanisms were developed more or less simultaneously by different research groups for various language understanding tasks (Bahdanau et al., 2015; Hermann et al., 2015). The advantage of using memory networks for the present purpose is that, unlike the alternatives, the memory network setting makes a clear distinction between the various isolable components of the network, which themselves are defined very generally. It is therefore possible to focus on and and vary the component of interest (the module that represents text in semantic memory) while using neutral (weak prior) specifications of the other components. In this way, I hope to reach more robust conclusions about the effect of representational form on the network's objective task.

6.1 Testing representational forms 'in the wild'

Humans do not interpret language in isolation. The context in which words and sentences are understood, whether a conversation, book chapter or road sign, plays an important role in human comprehension (Altmann and Steedman, 1988; Binder and Desai, 2011). In this work, I test the ability of neural language models (NLMs) to use such wider contexts to help make predictions about natural language. As I show, the way in which such contexts are represented is a critical factor in whether they can be exploited by the network.

All of the analysis in this chapter is based on a new benchmark dataset (The Children's Book Test or CBT) designed to test the role of memory and context in language processing and understanding. The test requires models to make predictions about different types of missing words in children's books, given both nearby words and a wider context from the book. Humans taking the test predict all types of word with similar levels of accuracy. However, they rely on the wider context to make accurate predictions about named entities or nouns, whereas it is unimportant when predicting higher-frequency verbs or prepositions.

As I show, state-of-the-art classical NLM architectures, Recurrent Neural Networks (RNNs) with Long-Short Term Memory (LSTMs), perform differently to humans on this task. They are excellent predictors of prepositions (*on, at*) and verbs (*run, eat*), but lag far behind humans when predicting nouns (*ball, table*) or named entities (*Elvis, France*). This is because their predictions are based almost exclusively on local contexts.

In contrast, Memory Networks (Weston et al., 2015b) are one of a class of 'contextual models' that can interpret language at a given point in text conditioned directly on both local information and explicit representation of the wider context. As I show, on the CBT, Memory Networks can exploit contextual information to achieve markedly better prediction of named-entities and nouns than conventional language models. This is important for applications that require coherent semantic processing and/or language generation, since nouns and entities typically encode much of the important semantic information in language.

However, not all contextual models reach this level of performance. The way in which wider context is represented in memory turns out be a critical factor in the network's ability to perform the task. Thus, analysis of the CBT with memory networks provides a useful benchmark for comparing the efficacy of different ways of representing words phrases and sentences (in the context). Moreover, unlike many of the evaluations used to evaluate representations in Chapters 2-5, this benchmark tests representations 'in the wild': it measures the extent they enable a network trained on relatively downstream (or 'extrinsic') language task to improve its performance.

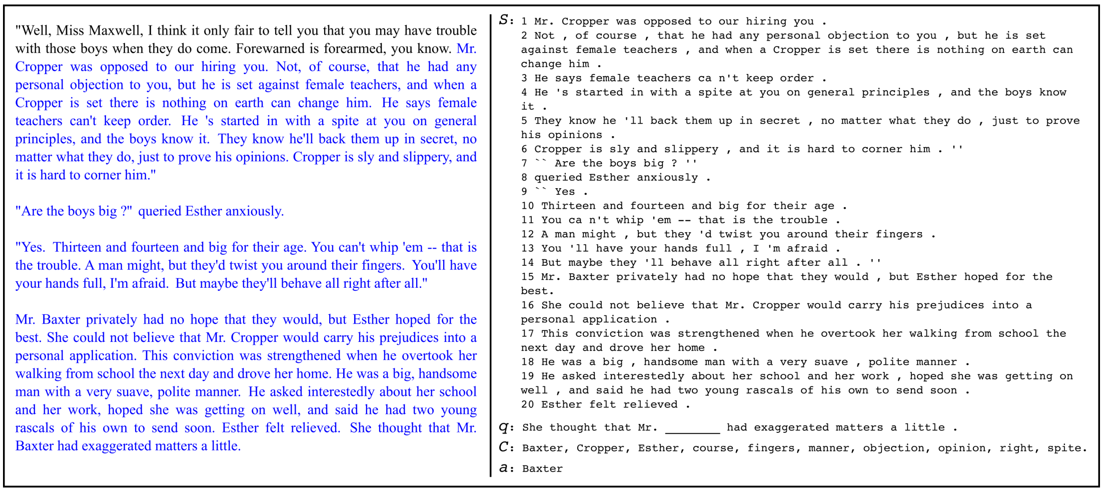
6.2 The Children's Book Test

The experiments in this paper are based on a new resource, the Children's Book Test, designed to measure directly how well language models can exploit wider linguistic context. The CBT is built from books that are freely available thanks to Project Gutenberg.[1] Using children's books guarantees a clear narrative structure, which can make the role of context more salient. After allocating books to either training, validation or test setS, example 'questions' (denoted x) are formed from chapters in the book by enumerating 21 consecutive sentences.

In each question, the first 20 sentences form the *context* (denoted S), and a word (denoted a) is removed from the 21^{st} sentence, which becomes the *query* (denoted q). Models must identify the *answer word* a among a selection of 10 candidate answers (denoted C) appearing in the context sentences and the query. Thus, for a question answer pair (x, a): $x = (q, S, C)$; S is an ordered list of sentences; q is a sentence (an ordered list $q = q_1, \ldots q_l$ of words) containing a missing word symbol; C is a bag of unique words such that $a \in C$, its cardinality $|C|$ is 10 and every candidate word $w \in C$ is such that $w \in q \cup S$. An example question is given in Figure 6.1.

For finer-grained analyses, I created four classes of question by removing distinct

[1] `https://www.gutenberg.org/`



"Well, Miss Maxwell, I think it only fair to tell you that you may have trouble with those boys when they do come. Forewarned is forearmed, you know. Mr. Cropper was opposed to our hiring you. Not, of course, that he had any personal objection to you, but he is set against female teachers, and when a Cropper is set there is nothing on earth can change him. He says female teachers can't keep order. He 's started in with a spite at you on general principles, and the boys know it. They know he'll back them up in secret, no matter what they do, just to prove his opinions. Cropper is sly and slippery, and it is hard to corner him."

"Are the boys big ?" queried Esther anxiously.

"Yes. Thirteen and fourteen and big for their age. You can't whip 'em -- that is the trouble. A man might, but they'd twist you around their fingers. You'll have your hands full, I'm afraid. But maybe they'll behave all right after all."

Mr. Baxter privately had no hope that they would, but Esther hoped for the best. She could not believe that Mr. Cropper would carry his prejudices into a personal application. This conviction was strengthened when he overtook her walking from school the next day and drove her home. He was a big, handsome man with a very suave, polite manner. He asked interestedly about her school and her work, hoped she was getting on well, and said he had two young rascals of his own to send soon. Esther felt relieved. She thought that Mr. Baxter had exaggerated matters a little.

```
S: 1 Mr. Cropper was opposed to our hiring you .
   2 Not , of course , that he had any personal objection to you , but he is set
   against female teachers , and when a Cropper is set there is nothing on earth can
   change him .
   3 He says female teachers ca n't keep order .
   4 He 's started in with a spite at you on general principles , and the boys know
   it .
   5 They know he 'll back them up in secret , no matter what they do , just to prove
   his opinions .
   6 Cropper is sly and slippery , and it is hard to corner him . ''
   7 `` Are the boys big ? ''
   8 queried Esther anxiously .
   9 `` Yes .
   10 Thirteen and fourteen and big for their age .
   11 You ca n't whip 'em -- that is the trouble .
   12 A man might , but they 'd twist you around their fingers .
   13 You 'll have your hands full , I 'm afraid .
   14 But maybe they 'll behave all right after all . ''
   15 Mr. Baxter privately had no hope that they would , but Esther hoped for the
   best.
   16 She could not believe that Mr. Cropper would carry his prejudices into a
   personal application .
   17 This conviction was strengthened when he overtook her walking from school the
   next day and drove her home .
   18 He was a big , handsome man with a very suave , polite manner .
   19 He asked interestedly about her school and her work , hoped she was getting on
   well , and said he had two young rascals of his own to send soon .
   20 Esther felt relieved .

q: She thought that Mr. ________ had exaggerated matters a little .

C: Baxter, Cropper, Esther, course, fingers, manner, objection, opinion, right, spite.

a: Baxter
```

Figure 6.1: A Named Entity question from the CBT (right), created from a book passage (left, in blue). In this case, the candidate answers C are both entities and common nouns, since fewer than ten named entities are found in the context.

types of word: Named Entities, (Common) Nouns, Verbs and Prepositions (based on output from the POS tagger and named-entity-recogniser in the Stanford Core NLP Toolkit (Manning et al., 2014)). For a given question class, the nine incorrect candidates are selected at random from words in the context having the same type as the answer. The exact number of questions in the training, validation and test sets is shown in Table 6.1. Full details of the candidate selection algorithm (e.g. how candidates are selected if there are insufficient words of a given type in the context) can be found with the dataset.[2]

Classical language modelling evaluations are based on average perplexity across all words in a text. They therefore place proportionally more emphasis on accurate prediction of frequent words such as prepositions and articles than the less frequent words that transmit the bulk of the meaning in language (Baayen and Lieber, 1996). In contrast, because the CBT allows focused analyses on semantic content-bearing words, it should be a better proxy for how well a language model can lend semantic coherence to applications including machine translation, dialogue and question-answering systems.

[2] The dataset can be downloaded from `http://fb.ai/babi/`.

	TRAINING	VALIDATION	TEST
NUMBER OF BOOKS	98	5	5
NUMBER OF QUESTIONS (CONTEXT+QUERY)	669,343	8,000	10,000
AVERAGE WORDS IN CONTEXTS	465	435	445
AVERAGE WORDS IN QUERIES	31	27	29
DISTINCT CANDIDATES	37,242	5,485	7,108
VOCABULARY SIZE	53,628		

Table 6.1: Statistics of the CBT. Breakdown by question class is provided with the data set files.

6.2.1 Related resources

There are clear parallels between the CBT and the Microsoft Research Sentence Completion Challenge (MSRCC) (Zweig and Burges, 2011), which is also based on Project Gutenberg (but not children's books, specifically). A fundamental difference is that, where examples in the MSRCC are made of a single sentence, each query in the CBT comes with a wider context. This tests the sensitivity of language models to semantic coherence beyond sentence boundaries. The CBT is also larger than the MRSCC (10,000 vs 1,040 test questions), requires models to select from more candidates on each question (10 vs 5), covers missing words of different (POS) types and contains large training and validation sets that match the form of the test set.

There are also similarities between the CBT and the CNN/Daily Mail (CNN QA) dataset recently released by Hermann et al. (2015). This task requires models to identify missing entities from bullet-point summaries of online news articles. The CNN QA task therefore focuses more on paraphrasing parts of a text, rather than making inferences and predictions from contexts as in the CBT. It also differs in that all named entities in both questions and articles are anonymised so that models cannot apply knowledge that is not apparent from the article. I do not anonymise entities in the CBT, as I hope to incentivise models that can apply background knowledge and information from immediate and wider contexts to the language understanding problem.[3] At the same time, the CBT can be used as a benchmark for general-purpose language models whose downstream application is semantically focused generation, prediction or correction. The CBT is also similar to the MCTest of machine comprehension (Richardson et al., 2013), in which children's stories written by annotators are accompanied by four multiple-choice questions. However, it is very difficult to train statistical models only on MCTest because its training set consists of only 300 examples.

[3] See Appendix A.4 for a sense of how anonymisation changes the CBT.

6.3 Memory representation in memory networks

Context sentences of S are encoded into memories, denoted m_i, using a feature-map $\phi(s)$ mapping sequences of words $s \in S$ from the context to one-hot representations in $[0,1]^d$, where d is typically the size of the word vocabulary. Memory networks are a comparatively new technology, and can be challenging to train as each constituent component must be optimised via a single error signal backpropagated from the output predictions. Indeed, this work constitutes their first application to naturally occurring language. We therefore constrain our analyses to three simple (word-order independent) forms for forming representations s of the context, detailed below. Ultimately, however, as the infrastructure challenges reduce, the prior structures and representational forms for word, phrase or sentence representation treated in Chapters 2-5 could also be used and might lead to improved results.

- **Lexical memory:** Each word occupies a separate slot in the memory (each phrase s is a single word and $\phi(s)$ has only one non-zero feature). To encode word order, time features are added as embeddings indicating the index of each memory, following Sukhbaatar et al. (2015).

- **Window memory:** Each phrase s corresponds to a window of text from the context S centred on an individual mention of a candidate c in S. Hence, memory slots are filled using windows of words $\{w_{i-(b-1)/2} \dots w_i \dots w_{i+(b-1)/2}\}$ where $w_i \in C$ is an instance of one of the candidate words in the question.[4] Note that the number of phrases s is typically greater than $|C|$ since candidates can occur multiple times in S. The window size b is tuned on the validation set. I experimented with encoding as a standard bag-of-words, or by having one dictionary per window position, where the latter performed best.

- **Sentential memory:** This setting follows the original implementation of Memory Networks for the bAbI tasks where the phrases s correspond to complete sentences of S. For the CBT, this means that each question yields exactly 20 memories. I also experiment with or without Positional Encoding (PE) as introduced by Sukhbaatar et al. (2015) to encode the word positions.

The order of occurrence of memories is less important for sentential and window formats than for lexical memory. So, instead of using a full embedding for each time

[4] See Appendix A.5 for discussion and analysis of using candidates in window representations and training.

index, I simply use a scalar value which indicates the position in the passage, ranging from 1 to the number of memories. An additional parameter (tuned on the validation set) scales the importance of this feature. As I show in Appendix A.3, time features only gave a marginal performance boost in those cases.

For sentential and window memory formats, queries are encoded in a similar way to the memories: as a bag-of-words representation of the whole sentence and a window of size b centred around the missing word position respectively. For the lexical memory, memories are made of the n words preceding the word to be predicted, whether these n words come from the context or from the query, and the query embedding is set to a constant vector 0.1.

6.3.1 End-to-end training

The MemN2N architecture, introduced by Sukhbaatar et al. (2015), allows for direct training of memory networks with backpropagation.

First, 'supporting memories', those useful to find the correct answer to the query q, are retrieved. This is done by embedding both the query and all memories into a single space of dimension p using an embedding matrix $\mathbf{A} \in \mathbb{R}^{p \times d}$ yielding the query embedding $\mathbf{q} = \mathbf{A}\phi(q)$ and memory embeddings $\{\mathbf{c}_i = \mathbf{A}\phi(s_i)\}_{i=1,\ldots n}$, with n the number of memories. The match between $\mathbf{q}$ and each memory $\mathbf{c}_i$ in the embedding space is fed through a softmax layer giving a distribution $\{\alpha_i\}_{i=1,\ldots n}$ of matching scores which are used as an *attention* mechanism over the memories to return the first supporting memory:

$$\mathbf{m}_{o1} = \sum_{i=1\ldots n} \alpha_i \mathbf{m}_i\,, \qquad \text{with} \quad \alpha_i = \frac{e^{\mathbf{c}_i^\top \mathbf{q}}}{\sum_j e^{\mathbf{c}_j^\top \mathbf{q}}},\; i = 1, \ldots n, \tag{6.1}$$

and where $\{\mathbf{m}_i\}_{i=1,\ldots n}$ is a set of memory embeddings obtained in the same way as the $\mathbf{c}_i$, but using another embedding matrix $\mathbf{B} \in \mathbb{R}^{p \times d}$. During training, optimization is carried out using stochastic gradient descent (SGD). Extra experimental details and hyperparameters are given in Appendix A.1.

6.3.2 Self-supervision for window memories

Training a memory network with multiple components by backpropagating a single error signal derived from its final predictions can constitute a challenging non-convex optimisation problem. After initial experiments, I found that it was beneficial to use a heuristic to provide a stronger signal for learning memory access. A related approach was successfully applied by Bordes et al. (2015) to question answering about knowledge bases.

Memory supervision (knowing which memories to attend to) is not provided at training time but is inferred automatically using a simple heuristic: during training, the correct supporting memory is assumed to be among the window memories whose corresponding candidate is the correct answer. In the common case where more than one memory contains the correct answer, the model picks the single memory $\tilde{m}$ that is already scored highest by itself, i.e. scored highest by the query in the embedding space defined by $\mathbf{A}$.[5]

Training is carried out by making gradient steps using SGD to force the model, for each example, to give a higher score to the supporting memory $\tilde{m}$ relative to any other memory from any other candidate. Instead of using eq (6.1), the model selects its top relevant memory using:

$$m_{o1} = \arg\max_{i=1,\ldots n} \mathbf{c}_i^\top \mathbf{q}\,. \tag{6.2}$$

If m_{o1} happens to be different from $\tilde{m}$, then the model is updated.

At test time, rather than use a hard selection as in eq (6.2) the model scores each candidate not only with its highest scoring memory but with the sum of the scores of all its corresponding windows after passing all scores through a softmax. That is, the score of a candidate is defined by the sum of the α_i (as used in eq (6.1)) of the windows it appears in. This relaxes the effects of the max operation and allows for all windows associated with a candidate to contribute some information about that candidate. As shown in the ablation study in Appendix A.3, this results in slightly better performance on the CNN QA benchmark compared to hard selection at test time.

Note that self-supervised Memory Networks do not exploit any new label information beyond the training data. The approach can be understood as a way of achieving *hard attention* over memories, to contrast with the *soft attention*-style selection described in Section 6.3.1. Hard attention yields significant improvements in image captioning (Xu et al., 2015). However, where Xu et al. (2015) use the REINFORCE algorithm (Williams, 1992) to train through the max of eq (6.2), the self-supervision heuristic permits direct backpropagation.

[5] TF-IDF similarity worked almost as well in the experiments, but a random choice over positives did not.

6.4 Baseline and comparison models

In addition to memory network variants, I also applied many different types of language modelling and machine reading architectures to the CBT.

6.4.1 Non-learning baselines

I implemented two simple baselines based on word frequencies. For the first, I selected the most frequent candidate in the entire training corpus. In the second, for a given question I selected the most frequent candidate in its context. In both cases I broke ties with a random choice.

I also tried two more sophisticated ways to rank the candidates that do not require any learning on the training data. The first is the ‘sliding window’ baseline applied to the MCTest by Richardson et al. (2013). In this method, ten ‘windows’ of the query concatenated with each possible candidate are slid across the context word-by-word, overlapping with a different subsequence at each position. The overlap score at a given position is simply word-overlap weighted TFIDF-style based on frequencies in the context (to emphasize less frequent words). The chosen candidate corresponds to the window that achieves the maximum single overlap score for any position. Ties are broken randomly.

The second method is the word distance benchmark applied by Hermann et al. (2015). For a given instance of a candidate w_i in the context, the query q is ‘superimposed’ on the context so that the missing word lines up with w_i, defining a subsequence s of the context. For each word q_i in q, an alignment penalty $P = \min(\min_{j=1...|s|}\{|i-j| : s_j = q_i\}, m)$ is incurred. The model predicts the candidate with the instance in the context that incurs the lowest alignment penalty. I tuned the maximum single penalty $m = 5$ on the validation data.

6.4.2 N-gram language models

I trained an n-gram language model using the KenLM toolkit (Heafield et al., 2013). I used Knesser-Ney smoothing, and a window size of 5, which performed best on the validation set. I also compare with a variant of language model with cache (Kuhn and De Mori, 1990), where I linearly interpolate the n-gram model probabilities with unigram probabilities computed on the context.

6.4.3 Supervised embedding models

To directly test how much of the CBT can be resolved by good quality dense representations of words (word embeddings), I implement a supervised embedding model similar to that of (Weston et al., 2010). In these models I learn both input and output embedding matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{p \times d}$ for each word in the vocabulary (p is still the embedding dimension and d the vocabulary size). For a given input passage q and possible answer word w, the score is computed as $S(q, w) = \phi(q)\mathbf{A}^\top \mathbf{B}\phi(w)$, with ϕ the feature function defined in Section 6.3. These models can be considered as lobotomised Memory Networks with zero hops, i.e. the attention over the memory component is removed.

I encode various parts of the question as the input passage: the entire **context + query**, just the **query**, a sub-sequence of the query defined by a **window** of maximum b words centred around the missing word, and a version (**window + position**) in which I use a different embedding matrix for encoding each position of the window. I tune the window-size $d = 5$ on the validation set.

6.4.4 Recurrent language models

I trained probabilistic RNN language models with LSTM activation units on the training stories (5.5M words of text) using minibatch SGD to maximise the negative log-likelihood of the next word. Hyper-parameters were tuned on the validation set. The best model had both hidden layer and word embeddings of dimension 512. When answering the questions in the CBT, I allow one variant of this model (**context + query**) to 'burn in' by reading the entire context followed by the query and another version to read only the **query** itself (and thus have no access to the context). Unlike the canonical language-modelling task, all models have access to the query words *after* the missing word (i.e if k is the position of the missing word, I rank candidate c based on $p(q_1 \dots q_{k-1}, c, q_{k+1} \dots q_l)$ rather than simply $p(q_1 \dots q_{k-1}, c)$).

Mikolov and Zweig (2012) previously observed performance boosts for recurrent language models by adding the capacity to jointly learn a document-level representation. I similarly apply a context-based recurrent model to the language-modelling tasks, but opt for the convolutional representation of the context applied by Rush et al. (2015) for summarisation. the Contextual LSTM (CLSTM) learns a convolutional attention over windows of the context given the objective of predicting all words in the query. I tuned the window size ($w = 5$) on the validation set. As with the standard LSTM, I trained the CLSTM on the running-text of the CBT training set (rather than the structured query and context format used with the Memory Networks) since this proved much more effective, and I report results in the best setting for each method.

6.4.5 Human performance

I recruited 15 native English speakers to attempt a randomly-selected 10% from each question type of the CBT, in two modes either with question only or with question+context (shown to different annotators), giving 2000 answers in total. To the knowledge, this is the first time human performance has been quantified on a language modelling task based on different word types and context lengths.

6.4.6 Other related approaches

The idea of conditioning language models on extra-sentential context is not new. Access to document-level features can improve both classical language models (Mikolov and Zweig, 2012) and word embeddings (Huang et al., 2012). Unlike the present work, these studies did not explore different representation strategies for the wider context or their effect on interpreting and predicting specific word types.

The original Memory Networks (Weston et al., 2015b) used hard memory selection with additional labeled supervision for the memory access component, and were applied to question-answering tasks over knowledge bases or simulated worlds. Sukhbaatar et al. (2015) and Kumar et al. (2015) trained Memory Networks with RNN components end-to-end with soft memory access, and applied them to additional language tasks. The attention-based reading models of Hermann et al. (2015) also have many commonalities with Memory Networks, differing in word representation choices and attention procedures. Both Kumar et al. (2015) and Hermann et al. (2015) propose bidirectional RNNs as a way of representing previously read text. the experiments in Section 6.5 provide a possible explanation for why this is an effective strategy for semantically-focused language processing: bidirectional RNNs naturally focus on small windows of text in similar way to window-based Memory Networks.

Other recent papers have proposed RNN-like architectures with new ways of reading, storing and updating information to improve their capacity to learn algorithmic or syntactic patterns (Joulin and Mikolov, 2015; Dyer et al., 2015; Grefenstette et al.,

Methods	Named Entities	Common Nouns	Verbs	Prepositions
Humans (query)$^{(*)}$	0.520	0.644	0.716	0.676
Humans (context+query)$^{(*)}$	***0.816***	***0.816***	***0.828***	0.708
Maximum frequency (corpus)	0.120	0.158	0.373	0.315
Maximum frequency (context)	0.335	0.281	0.285	0.275
Sliding window	0.168	0.196	0.182	0.101
Word distance model	0.398	0.364	0.380	0.237
Kneser-Ney language model	0.390	0.544	0.778	0.768
Kneser-Ney language model + cache	0.439	0.577	0.772	0.679
Embedding Model (context+query)	0.253	0.259	0.421	0.315
Embedding Model (query)	0.351	0.400	0.614	0.535
Embedding Model (window)	0.362	0.415	0.637	0.589
Embedding Model (window+position)	0.402	0.506	0.736	0.670
LSTMs (query)	0.408	0.541	0.813	0.802
LSTMs (context+query)	0.418	0.560	**0.818**	0.791
Contextual LSTMs (window context)	0.436	0.582	0.805	**0.806**
MemNNs (lexical memory)	0.431	0.562	0.798	0.764
MemNNs (window memory)	0.493	0.554	0.692	0.674
MemNNs (sentential memory + PE)	0.318	0.305	0.502	0.326
MemNNs (window memory + self-sup.)	**0.666**	**0.630**	0.690	0.703

Table 6.2: Results on CBT test set. $^{(*)}$Human results were collected on 10% of the test set.

2015). While I do not study these models in the present work, the CBT would be ideally suited for testing this class of model on semantically-focused language modelling.

6.5 Results

The form of memory representations Of central importance to this thesis is the fact that the particular form of the internal memory representations had a clear impact of the performance of memory networks.

When each sentence in the context is stored as an ordered sequence of word embeddings (*sentence mem + PE*), performance is generally poor. Encoding the context as an unbroken sequence of individual words (*lexical memory*) works particularly well for capturing prepositions and verbs, but is less effective with nouns and entities. In contrast, *window memories* centred around the candidate words are more useful than either word-level or sentence-level memories when predicting named entities and nouns.

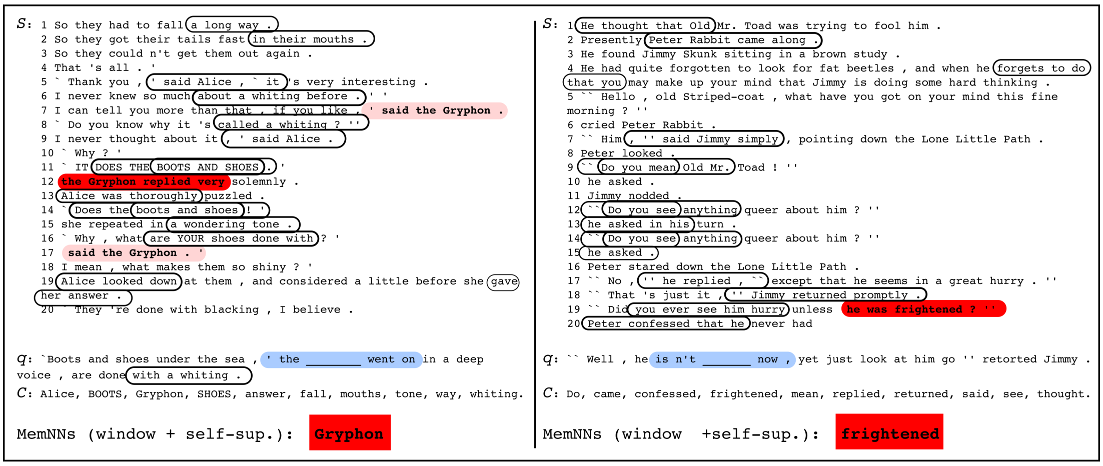
It is tempting to ascribe the poor performance of full-sentence embeddings to the limitations of bag-of-words encoding (and the blindness to word order). However, for nouns and named entity prediction, networks with memories that do not encode word order (window memories) ara capable of outperforming those that do (both lexical memories and the DeepMind bidirectional RNN reading models (Hermann et al., 2015) discussed in Section 6.5.1). This suggests that word order itself is of secondary importance compared with the ‘scope’ of the representation. In particular, it appears that full sentences in general contain too much disparate information for the pertinent parts to be easily accessed if encoded into a single representation.

Of course, it is important to note that these conclusions relat only to the present (word-prediction) task and the CNN QA task described below. The extent to which they generalise remains to be determined by future work.

Modelling syntactic flow In general, there is a clear difference in model performance according to the type of word to be predicted. the main results in Table 6.2 show conventional language models are very good at predicting prepositions and verbs, but less good at predicting named entities and nouns. Among these language models, and in keeping with established results, RNNs with LSTMs demonstrate a small gain on n-gram models across the board, except for named entities where the cache is beneficial. In fact, LSTM models are better than humans at predicting prepositions, which suggests that there are cases in which several of the candidate prepositions are 'correct', but annotators prefer the less frequent one. Even more surprisingly, when only local context (the query) is available, both LSTMs and n-gram models predict verbs more accurately than humans. This may be because the models are better attuned to the distribution of verbs in children's books, whereas humans are unhelpfully influenced by their wider knowledge of all language styles.[6] When access to the full context is available, humans do predict verbs with slightly greater accuracy than RNNs.

Capturing semantic coherence The best performing Memory Networks predict common nouns and named entities more accurately than conventional language models. Clearly, in doing so, these models rely on access to the wider context (the supervised EMBEDDING MODEL (QUERY), which is equivalent to the memory network but with no contextual memory, performs poorly in this regard). The fact that LSTMs without attention perform similarly on nouns and named entities whether or not the context is available confirms that they do not effectively exploit this context. This may be a symptom of the difficulty of storing and retaining information across large numbers of time steps that has been previously observed in recurrent networks (See e.g. Bengio et al. (1994)).

[6] I did not require the human annotators warm up by reading the 98 novels in the training data, but this might have led to a fairer comparison.



```
S: 1 So they had to fall a long way .
   2 So they got their tails fast in their mouths .
   3 So they could n't get them out again .
   4 That 's all . '
   5 ` Thank you , ' said Alice , ` it 's very interesting .
   6 I never knew so much about a whiting before . ' '
   7 I can tell you more than that , if you like , ' said the Gryphon .
   8 ` Do you know why it 's called a whiting ? ''
   9 I never thought about it , ' said Alice .
   10 ` Why ? '
   11 ` IT DOES THE BOOTS AND SHOES . '
   12 the Gryphon replied very solemnly .
   13 Alice was thoroughly puzzled .
   14 ` Does the boots and shoes ! '
   15 she repeated in a wondering tone .
   16 ` Why , what are YOUR shoes done with ? '
   17 said the Gryphon . '
   18 I mean , what makes them so shiny ? '
   19 Alice looked down at them , and considered a little before she gave
  her answer .
   20 ` They 're done with blacking , I believe .

q: `Boots and shoes under the sea , ' the _______ went on in a deep
voice , are done with a whiting .
C: Alice, BOOTS, Gryphon, SHOES, answer, fall, mouths, tone, way, whiting.

MemNNs (window + self-sup.): Gryphon
```

```
S: 1 He thought that Old Mr. Toad was trying to fool him .
   2 Presently Peter Rabbit came along .
   3 He found Jimmy Skunk sitting in a brown study .
   4 He had quite forgotten to look for fat beetles , and when he forgets to do
  that you may make up your mind that Jimmy is doing some hard thinking .
   5 `` Hello , old Striped-coat , what have you got on your mind this fine
   morning ? ''
   6 cried Peter Rabbit .
   7 `` Him , '' said Jimmy simply , pointing down the Lone Little Path .
   8 Peter looked .
   9 `` Do you mean Old Mr. Toad ! ''
   10 he asked .
   11 Jimmy nodded .
   12 `` Do you see anything queer about him ? ''
   13 he asked in his turn .
   14 `` Do you see anything queer about him ? ''
   15 he asked .
   16 Peter stared down the Lone Little Path .
   17 `` No , '' he replied , `` except that he seems in a great hurry . ''
   18 `` That 's just it , '' Jimmy returned promptly .
   19 `` Did you ever see him hurry unless he was frightened ? ''
   20 Peter confessed that he never had

q: `` Well , he is n't _______ now , yet just look at him go '' retorted Jimmy .

C: Do, came, confessed, frightened, mean, replied, returned, said, see, thought.

MemNNs (window  +self-sup.): frightened
```

Figure 6.2: Correct predictions of MemNNs (window memory + self-supervision) on CBT on Named Entity (left) and Verb (right). Circled phrases indicate all considered windows; red ones are the ones corresponding to the returned (correct) answer; the blue windows represent the queries.

Self-supervised memory retrieval The window-based Memory Network with self-supervision (in which a hard attention selection is made among window memories during training) outperforms all others at predicting named entities and common nouns. Examples of predictions made by this model for two CBT questions are shown in Figure 6.2. It is notable that this model is able to achieve the strongest performance with only a simple window-based strategy for representing questions.

METHODS	VALIDATION	TEST
MAXIMUM FREQUENCY (ARTICLE)$^{(*)}$	0.305	0.332
SLIDING WINDOW	0.005	0.006
WORD DISTANCE MODEL$^{(*)}$	0.505	0.509
DEEP LSTMS (ARTICLE+QUERY)$^{(*)}$	0.550	0.570
CONTEXTUAL LSTMS (“ATTENTIVE READER”)$^{(*)}$	0.616	0.630
CONTEXTUAL LSTMS (“IMPATIENT READER”)$^{(*)}$	0.618	0.638
MEMNNS (WINDOW MEMORY)	0.580	0.606
MEMNNS (WINDOW MEMORY + SELF-SUP.)	0.634	0.668
MEMNNS (WINDOW MEMORY + ENSEMBLE)	0.612	0.638
MEMNNS (WINDOW MEMORY + SELF-SUP. + ENSEMBLE)	0.649	0.684
MEMNNS (WINDOW + SELF-SUP. + ENSEMBLE + EXCLUD. COOCURRENCES)	**0.662**	**0.694**

Table 6.3: Results on CNN QA. $^{(*)}$Results taken from Hermann et al. (2015).

6.5.1 News article question answering

To examine how well the conclusions generalise to different machine reading tasks and language styles, I also tested the best-performing Memory Networks on the CNN

QA task (Hermann et al., 2015).[7] This dataset consists of 93k news articles from the CNN website, each coupled with a question derived from a bullet point summary accompanying the article, and a single-word answer. The answer is always a named entity, and all named entities in the article function as possible candidate answers.

As shown in Table 6.3, the window model without self-supervision achieves similar performance to the best approach proposed for the task by DeepMind (Hermann et al., 2015) when using an ensemble of MemNN models. The use of an ensemble is an alternative way of replicating the application of *dropout* (Hinton et al., 2012) in the previous best approaches (Hermann et al., 2015) as ensemble averaging has similar effects to dropout (Wan et al., 2013). When self-supervision is added, the Memory Network greatly surpasses the state-of-the-art on this task. Finally, the last line of Table 6.3 (*excluding co-occurrences*) shows how an additional heuristic, removing from the candidate list all named entities already appearing in the bullet point summary, boosts performance even further.

Some common principles pertinent to the key questions of this thesis may explain the strong performance of the best performing models on this task. The DeepMind attentive/impatient reading models encode the articles using bidirectional RNNs (Graves et al., 2008). For each word in the article, the combined hidden state of such an RNN naturally focuses on a window-like chunk of surrounding text, much like the window-based memory network or the CLSTM. Together, these results therefore support the principle that the most informative representations of text correspond to sub-sentential chunks. Indeed, the observation that the most informative representations for neural language models correspond to small chunks of text is also consistent with recent work on neural machine translation, in which Luong et al. (2015a) demonstrated improved performance by restricting their attention mechanism to small windows of the source sentence.

Given these commonalities in how the reading models and Memory Networks represent context, the advantage of the best-performing Memory Network instead seems to stem from how it accesses or retrieves this information; in particular, the hard attention and self-supervision. Jointly learning to access an

d use information is a difficult optimization. Self-supervision in particular makes effective Memory Network learning more tractable.

[7] The CNN QA dataset was released after the primary experiments were completed, hence I experiment only with one of the two large datasets released with that paper.

6.6 Conclusion

In this chapter, I have presented an alternative framework for analysing and evaluating different forms of linguistic representation. The approach differs from those taken in Chapters 2-5, in that it directly tests the effect of representations on more extrinsic language-prediction task whose application to language technology is clear and unambiguous. The development of neural language models such as memory networks, which compute multiple layers of internal representations in an otherwise end-to-end architecture, makes this form of analysis and evaluation increasingly viable.

The conclusions from this chapter were based largely on the Children's Book Test, a new semantic language modelling task. The CBT measures how well models can use both local and wider contextual information to make predictions about different types of words in children's stories. A particular strength of the CBT over similar existing benchmarks is the clear separating drawn between prediction of syntactic function words and more semantically informative terms. This distinction makes the CBT a robust proxy for the impact of language models on applications that require a focus on semantic coherence. It also facilitates finer grained analyses that permit more detailed conclusions about the effects of various modelling decisions, including, most pertinently, the form of memory representations.

The most consistent finding overall was that memories that encode sub-sentential chunks (windows) of informative text seem to be most useful to neural nets, particularly for tasks involving the prediction of the most semantically informative words in text. Indeed, this effect was observed on both the CBT and the CNN QA benchmark, an independent test of machine reading that focuses solely on the prediction of entities.

Since the experiments in this chapter were carried out, further evidence has emerged of the strength of this effect. Models that combine important aspects of the best memory networks (self-supervision) and the DeepMind reading models (context representation based on bidirectional RNNs) seems to outperform both models Kadlec et al. (2016). This provides further support for the utility of window-like memory representations, while highlighting the benefits of a soft, flexible or variable window length over a prescribed, fixed memory scope.

Chapter 7

Conclusion

Computational models that encode semantic knowledge in distributed representations are performing better and better on tasks that mirror human cognitive capacities. However, unlike these models, which are each trained to accomplish a specific AI task, humans are continuously learning without a particular skill or application in mind. The human learner is able to acquire general concepts and knowledge from experience via constant observation of the environment, such that it can later be applied quickly and effectively to multiple tasks and goals. The objective of this thesis has been to develop ways of effecting this human-like unsupervised learning in neural language models.

Thanks largely to the power of the distributional hypothesis of natural language, various established methods already existed for acquiring general-purpose word representations from linguistic input. For phrase or sentence semantics, however, this was not the case. Most general approaches to interpreting phrases or sentences involved mapping between text and symbolic or localist representations such as logical forms (see e.g. Poon and Domingos 2009). This thesis demonstrates that, in light of recently-developed neural language modelling architectures, the symbolic approach is no longer the only way to represent phrases and sentences in a generally-applicable way. In the long term, the potential impact of extending the distributional approach from words to phrases and sentences is tremendous, given the desirable computational and modelling properties of distributed representations and the fact that that most information transfer between humans involves these larger units rather than words in isolation.

7.1 Contributions of this thesis

A new resource for the evaluation of distributed word representations A clear understanding of how word concepts can be acquired from text can ultimately shed light on the same phenomenon in the case of larger linguistic units. However, without robust ways to evaluate the quality of word representations, it would be difficult to compare various approaches and detect improvements. Existing methods suffered from a range of limitations, such as low word coverage, poorly defined scores or low inter-rater agreement. In Chapter 2 I described SimLex-999, a resource designed to mitigate these limitations. SimLex covers a more representative set of word concepts than many alternative evaluation resources. It measures semantic similarity, a relation about which native English speakers seem to have clearer, more consistent intuitions. Since its release, SimLex-999 has been used to evaluate numerous new algorithms and approaches for word representation learning (see e.g. Lazaridou et al. 2015; Levy et al. 2015a; Lu et al. 2015). It has also been translated into German, Italian and Russian (Leviant, 2016).

A novel method for acquiring distributed word representations from bilingual texts The fact that SimLex-999 facilitates a better understanding of the differing representational geometry of word embedding spaces was underlined in Chapter 3. Here, I showed how the objective of translating between sentences in bilingual corpora, via neural machine translation models, yields word representation spaces that are qualitatively different from those learned via more conventional monolingual methods. Specifically, neural translation embeddings are more naturally orientated according to semantic similarity than conventional word embeddings. This observations enabled us to achieve what was at the time the best reported performance of a distributional model on SimLex-999.

A simple method for extending word representations to phrases In Chapter 4, I showed how NLMs could be trained effectively on the textual definitions or descriptions in dictionaries and encyclopedias. In these models, dictionaries provide a bridge between lexical meaning and phrase meaning, allowing the model's interpretation of phrases to be 'supervised' by the corresponding lexical representation (acquired by models like those described in Chapters 2 and 3). The combination of the representational power of NLMs and the principled semantic information in dictionaries proved to be very powerful. The trained models generalise well beyond the training data. They are capable of beating established dictionary-indexing software at retrieving concepts not defined in the training data, an effect that is magnified when the linguistic style of description of definition differs from that of the training set, and can even answer general-knowledge crossword questions. Moreover, a systematic comparison of various methods for acquiring phrase or sentence representations from unlabelled data (Chapter 5) revealed the representations encoded by models trained in this way to perform more consistently than alternative NLM architectures across a suite of supervised and unsupervised evaluations.

Two novel models for learning distributed sentence representations from text In Chapter 5, I also developed two further algorithms for acquiring distributed sentence representations, each with certain specific advantages over alternative approaches. The first, the sequential denoising autoencoder, can be trained on any collection of unordered sentences, and learns representations that are particularly applicable to paraphrasing applications. The second, FastSent, is a modification CBOW, a well-known log-linear model for lexical representation learning, in which word embeddings are optimised to form useful sentence representations under the addition operation. Like other shallow neural language models, FastSent performs best in unsupervised applications involving a linear decoding of its representation space. It outperforms alternatives at direct prediction of sentence relatedness, and qualitative analysis (e.g. via the web demo) suggests a more semantically plausible space of sentence representations than alternatives.

Interpreting language in context with memory networks Much like words, sentences are rarely interpreted in isolation, but as elements in some wider story, discourse or conversation. In Chapter 6, I described how *memory networks*, a novel neural network architecture, can be adapted as a context-aware model of sentence processing. The first memory networks were applied to question answering about simulated discussions, but here I applied them to natural language, and the task of interpreting a sentence in a story conditioned on explicit representations of multiple previous sentences. Using the Children's Book Test, our novel training and evaluation resource for sentence completion in context, I compared multiple ways of representing sentences in such a model. As well as a model of machine comprehension in context, the model framework and evaluation resource together provide an extrinsic (task-based) means to compare different strategies for encoding language in distributed memory. Using this strategy, I showed that models that effectively focus on small sub-sentential windows convey more useful information than those whose focus is both broader (entire sentences) or narrower (ordered sequences of words). This is one of the first studies to apply neural language models to sentence processing in context. The contributions can be understood in a general tendency of language processing research away from analysing individual sentences in isolation, towards models that can effectively interpret utterances as part of a broader stream of language.

7.2 Future work

The approaches to knowledge representation and language learning described in this thesis involve training models to make predictions from language corpora or structured text-based resources. The extension of the training environment beyond unstructured running text, in Chapters 4 and 5, was intended to mitigate the discrepancy between the information available to human language learners and the signal from text alone. Nevertheless, there remains a clear point of difference between these approaches and human language learning that may be critical to address if models are to exhibit truly human-like linguistic behaviours.

For the models considered in this thesis, the learning environment is *passive*, in that the training information source is pre-determined before learning begins, and does not alter at any stage as a consequence of the output of the model. As such, these experiments mimic the part of language learning in which children observe adult language by listening to the conversations of others or reading. Such passive observation must be how babies learn their first words, and may well be a critical part of language learning even after they can talk. Nevertheless, once they can talk, children actively influence the nature of their future linguistic experience (e.g by moving conversations in a particular direction). In such an interactive setting, learners can infer linguistic meaning not simply by observing correct language, or even determining what in the world that language refers to, but also by noticing the reaction it provokes in others and benefitting from the communicative acts that it facilitates. Indeed, to realise more complex communicative acts such as negotiation, humans must ultimately learn to produce multiple utterances and receive multiple responses with a only a delayed indication of whether their linguistic behaviour was ‘correct’ with respect to the goal in question. These aspects of learning, which are not part of the environments and algorithms studied in this thesis, may be critical for efficiently training models to replicate human language understanding and production in a robust way.

The next stage of this research programme is therefore to place the language models described in this thesis into more dynamic, interactive and goal-driven learning environments. In recent work, interactive learning frameworks such as reinforcement learning have proved very effective tools for training agents to resolve computer games, particularly when deep neural networks are used to represent the situations faced by the agent and thus effectively reduce the search space among state-action pairs (Mnih et al., 2015). Indeed, the same strategy has also been applied to language games, in which states are described by textual descriptions and the model agent must choose between three possible actions at each stage (Narasimhan et al., 2015).

A general approach to training language models as agents was recently sketched by Mikolov et al. (2015). This framework relies on a deterministic environment programmed to reflect and describe the world, a learner agent that interacts with that environment, and a *teacher* agent which interacts with the learner agent in light of the environment. In a typical interaction, the teacher issues a task to the learner (*find the rabbit*) and the learner has to work out how to accomplish the task by specifying a sequence of actions. Actions are communicated via language to the environment (*Learner to Environment: I move left*) and the consequences of the action are communicated via language back to the leaner (*Environment to Learner: You are now in the kitchen*). Aside from attempting and realising actions (*Leaner to Environment: I look behind me*), the leaner might also engage the teacher to get help (*Learner to Teacher: Can you remind me what to do again?*). The learner receives reward when the teacher's instructions are satisfied, and may receive penalties for undesirable behaviours such as taking too long. In principle, if learning agents have sufficient capacity for remembering, represent and generalising, they could be 'taught' a general form of language understanding via exposure to multiple task/environment combinations.

The framework of Mikolov et al. (2015), and specifically the constraint that information about the environment is encoded in language, yields a comparatively simple way to train language models in an interactive way. Nevertheless, there are many likely challenges that must be resolved before it can be applied as a general way of yielding agents or models with robust linguistic behaviour. First, the effect of constraining all information sharing to language is not known. While language in the Mikolov et al. (2015) framework is grounded in the responses of the environment, language as a channel of information transfer is very narrow bandwidth, and the information about the consequence of actions will be much more constrained than that received by humans. Second, unlike with game-playing scenarios, linguistic behaviour (possible utterances) cannot be reduced to a (small) finite number of possible actions and corresponding symbols. On the one hand, it is not clear how the environment would learn to interpret the possibly noisy output of the learner. On the other hand, requiring the learner to generate unique, unambiguous commands would surely result in unrealistic output. Finally, there is the problem of knowing how to teach. With such a large search-space of possible utterances available to the learner, teaching an uninitiated model to find actions that are at all relevant to the environment and objective may be difficult in the general case. As the authors suggest, such a process will likely involve a structured curriculum of tasks and environments, beginning with the very simple. Even once learning is up and running, it may not be trivial to formalise the vast and dynamic selection of behavioural goals that are characteristic of human activity and which combine to make general language understanding and production viable.

These challenges are significant, and their resolution may require expertise in many fields, from linguistics, machine-learning and neuroscience to multi-agent systems and even robotics. Nevertheless, recent developments in each of these disciplines are such that progress towards the overall goal no longer seems unattainable. And if the necessary collaborations can be realised, and progress in this direction is achieved, then for general linguistic utterances, much as is the case already for single words, the path from frequency to meaning may become a little less obscure.

List of publications

The work in this thesis has also been reported in the following publications.

Chapter 2 SimLex-999: Estimating Semantic Models with (Genuine) Similarity Estimation. Hill, F., Reichart, R. & Korhonen, A. *Computational Linguistics*. 2015.

Chapter 3 *Embedding Word Similarity with Neural Machine Translation*. Hill, F., Cho, K., Jean, S. & Bengio, Y. Proceedings of ICLR (Workshop). 2015

Chapter 4 Learning to Understand Phrases by Embedding the Dictionary. Hill, F., Cho, K., Korhonen, A. & Bengio, Y. *Transactions of the Association for Computational Linguistics*. 2016

- *Crossword Computer has All the Answers*. The Times. 8 March 2016
- *This Artificial Intelligence is Trained on Crosswords*. Wired. 7 March 2016

Chapter 5 *Learning Distributed Representations of Sentences from Unlabelled Data*. Hill, F., Cho,K. & Korhonen, A. Proceedings of NAACL. 2016

Chapter 6 *The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations*. Hill, F., Bordes, A., Chopra, S. & Weston, J. Proceedings of ICLR. 2016

- *Facebook's AI is Learning by Reading Load's of Children's Books*. New Scientist. 25 February 2016

Bibliography

Eneko Agirre and Aitor Soroa. Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 7–12. Association for Computational Linguistics, 2007.

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*, 2009a.

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, 2009b.

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.

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.

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.

Gerry Altmann and Mark Steedman. Interaction with context during human sentence processing. *Cognition*, 30(3):191–238, 1988.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

Jan Anward, Edith Moravcsik, and Leon Stassen. Parts of speech: A challenge for typology. *Linguistic typology*, 1(2):167–184, 1997.

R Harald Baayen and Rochelle Lieber. Word frequency distributions and lexical semantics. *Computers and the Humanities*, 30(4):281–291, 1996.

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

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

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

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

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, 2014b.

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.

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.

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

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

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.

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

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003b.

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

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.

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.

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.

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.

Jeffrey R Binder and Rutvik H Desai. The neurobiology of semantic memory. *Trends in cognitive sciences*, 15(11):527–536, 2011.

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.

David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

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

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

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, 2012a.

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, 2012b.

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

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

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.

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.

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*, 2014a.

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*, 2014b.

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

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.

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.

Brigitte Cordier. Factor analysis of correspondences. In *Proceedings of COLING*, 1965.

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

Sebastian J Crutch, Sarah Connell, and Elizabeth K Warrington. The different representational frameworks underpinning abstract and concrete knowledge: Evidence from odd-one-out judgements. *The Quarterly Journal of Experimental Psychology*, 62(7):1377–1390, 2009.

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

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

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.

Nai Ding, Lucia Melloni, Hang Zhang, Xing Tian, and David Poeppel. Cortical tracking of hierarchical linguistic structures in connected speech. *Nature neuroscience*, 19(1):158–164, 2016.

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.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2015.

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

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

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.

Gilles Fauconnier. *Mental spaces: Aspects of meaning construction in natural language*. Cambridge University Press, 1994.

Christiane Fellbaum. *WordNet*. Wiley Online Library, 1999.

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.

Charles J Fillmore. Frames and the semantics of understanding. *Quaderni di semantica*, 6(2):222–254, 1985.

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.

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

Jerry Fodor. Having concepts: A brief refutation of the twentieth century. *Mind & Language*, 19(1):29–47, 2004.

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

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

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

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

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

Alex Graves, Marcus Liwicki, Horst Bunke, Jürgen Schmidhuber, and Santiago Fernández. Unconstrained on-line handwriting recognition with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 577–584, 2008.

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

Edward Grefenstette, Karl Moritz Hermann, Mustafa Suleyman, and Phil Blunsom. Learning to transduce with unbounded memory. *NIPS*, 2015.

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

Russell Grigg. *Lacan, language, and philosophy*. SUNY Press, 2009.

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

Kenneth Harper. Measurement of similarity between nouns. In *Proceedings of COLING*, 1965.

Zellig S Harris. Distributional structure. *Word*, 1954.

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

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.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August 2013.

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

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

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. URL `http://arxiv.org/abs/1506.03340`.

Felix Hill and Anna Korhonen. Learning abstract concept embeddings from multimodal data: Since you probably can't see what i mean. In *EMNLP*, pages 255–265. Citeseer, 2014.

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

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

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.

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

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

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *Proceedings of ICLR*, 2016.

Geoffrey E Hinton. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

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

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.

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.

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.

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.

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

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

Armand Joulin and Tomas Mikolov. Inferring algorithmic patterns with stack-augmented recurrent nets. *NIPS*, 2015.

Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. Text understanding with the attention sum reader network. *arXiv preprint arXiv:1603.01547*, 2016.

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.

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

Markus Kiefer and Friedemann Pulvermüller. Conceptual representations in mind and brain: theoretical developments, current evidence and future directions. *Cortex*, 48 (7):805–825, 2012.

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.

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.

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

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, 2015b.

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

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

Roland Kuhn and Renato De Mori. A cache-based natural language model for speech recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12 (6):570–583, 1990.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. *http://arxiv.org/abs/1506.07285*, 2015.

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.

Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. Combining language and vision with a multimodal skip-gram model. *arXiv preprint arXiv:1501.02598*, 2015.

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

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.

Ira Leviant. *Separated by an Un-common Language: Towards Judgment Language Informed Vector Space Modeling*. PhD thesis, Technion, 2016.

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

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

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, 2015a.

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*, 2015b.

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.

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

Ang Lu, Weiran Wang, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Deep multilingual correlation for improved word embeddings. In *Proceedings of NAACL*, 2015.

Margery Lucas. Semantic priming without association: A meta-analytic review. *Psychonomic Bulletin & Review*, 7(4):618–630, 2000.

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

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *Proceedings of EMNLP*, 2015a.

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*, 2015b.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

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.

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.

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.

William Marslen-Wilson and Lorraine Komisarjevsky Tyler. The temporal structure of spoken language understanding. *Cognition*, 8(1):1–71, 1980.

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.

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.

Risto Miikkulainen and Michael G Dyer. Encoding input/output representations in connectionist cognitive systems. *Proceedings of the Connectionist Summer School*, 1989.

Risto Miikkulainen and Michael G Dyer. Natural language processing with modular pdp networks and distributed lexicon. *Cognitive Science*, 15(3):343–399, 1991.

Tomas Mikolov and Geoffrey Zweig. Context dependent recurrent neural network language model. In *SLT*, pages 234–239, 2012.

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

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, 2013a.

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

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, 2013c.

Tomas Mikolov, Armand Joulin, and Marco Baroni. A roadmap towards machine intelligence. *arXiv preprint arXiv:1511.08130*, 2015.

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

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

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

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

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.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language understanding for text-based games using deep reinforcement learning. *Proceedings of EMNLP*, 2015.

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

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.

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

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.

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

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.

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.

Karalyn Patterson, Peter J Nestor, and Timothy T Rogers. Where do you know what you know? the representation of semantic knowledge in the human brain. *Nature Reviews Neuroscience*, 8(12):976–987, 2007.

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.

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 2014a.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014b.

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.

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.

Steven Pinker. *Language learnability and language development, with new commentary by the author*, volume 7. Harvard University Press, 2009.

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

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 (LSDSem)*, page 1, 2015.

Hoifung Poon and Pedro Domingos. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 1–10. Association for Computational Linguistics, 2009.

Jesse J Prinz. *Furnishing the mind: Concepts and their perceptual basis*. MIT press, 2004.

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.

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

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, 2010b.

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

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

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, volume 1, page 2, 2013.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*, 2015.

Eleanor Rosch, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny Boyes-Braem. Basic objects in natural categories. *Cognitive psychology*, 8(3):382–439, 1976a.

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

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.

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

Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *Proceedings of EMNLP*, 2015.

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.

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.

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.

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

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.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer, 2013.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. *Proceedings of NIPS*, 2015.

Lin Sun and Anna Korhonen. Improving verb clustering with automatically acquired selectional preferences. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 638–647. Association for Computational Linguistics, 2009.

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.

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.

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.

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.

Michael Tomasello and Michael Tomasello. *Constructing a language: A usage-based theory of language acquisition*. Harvard University Press, 2009.

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.

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

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.

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

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.

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.

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

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.

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.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann L Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1058–1066, 2013.

Tong Wang, Abdel-rahman Mohamed, and Graeme Hirst. Learning lexical embeddings with syntactic and lexicographic knowledge. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 458–463, 2015.

Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 81(1):21–35, 2010.

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*, 2015a.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *Proceedings of ICLR*, 2015b.

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

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.

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

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

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

Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning (ICML'15)*, 2015. URL `http://arxiv.org/abs/1502.03044`.

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.

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

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer vision–ECCV 2014*, pages 818–833. Springer, 2014.

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.

Geoffrey Zweig and Christopher JC Burges. The microsoft research sentence completion challenge. Technical report, Technical Report MSR-TR-2011-129, Microsoft, 2011.

Appendix A

A.1 Chapter 6: Experimental details

Setting The text of questions is lowercased for all Memory Networks as well as for all non-learning baselines. LSTMs models use the raw text (although I also tried lowercasing, which made little difference). Hyperparameters of all learning models have been set using grid search on the validation set. The main hyperparameters are embedding dimension p, learning rate λ, window size b, number of hops K, maximum memory size n ($n = all$ means using all potential memories). All models were implemented using the Torch library (see `torch.ch`). For CBT, all models have been trained on all question types altogether. We did not try to experiment with word embeddings pre-trained on a bigger corpus.

Optimal hyper-parameter values on CBT:

- Embedding model (context+query): $p = 300$, $\lambda = 0.01$.
- Embedding model (query): $p = 300$, $\lambda = 0.01$.
- Embedding model (window): $p = 300$, $\lambda = 0.005$, $b = 5$.
- Embedding model (window+position): $p = 300$, $\lambda = 0.01$, $b = 5$.
- LSTMs (query & context+query): $p = 512$, $\lambda = 0.5$, 1 layer, gradient clipping factor: 5, learning rate shrinking factor: 2.
- Contextual LSTMs: $p = 256$, $\lambda = 0.5$, 1 layer, gradient clipping factor: 10, learning rate shrinking factor: 2.
- MemNNs (lexical memory): $n = 200$, $\lambda = 0.01$, $p = 200$, $K = 7$.

- MemNNs (window memory): $n = all$, $b = 5$, $\lambda = 0.005$, $p = 100$, $K = 1$.
- MemNNs (sentential memory + PE): $n = all$, $\lambda = 0.001$, $p = 100$, $K = 1$.
- MemNNs (window memory + self-sup.): $n = all$, $b = 5$, $\lambda = 0.01$, $p = 300$.

Optimal hyper-parameter values on CNN QA:

- MemNNs (window memory): $n = all$, $b = 5$, $\lambda = 0.005$, $p = 100$, $K = 1$.
- MemNNs (window memory + self-sup.): $n = all$, $b = 5$, $\lambda = 0.025$, $p = 300$, $K = 1$.
- MemNNs (window memory + ensemble): 7 models with $b = 5$.
- MemNNs (window memory + self-sup. + ensemble): 11 models with $b = 5$.

A.2 Results on CBT Validation Set

Methods	Named Entities	Common Nouns	Verbs	Prepositions
Maximum frequency (corpus)	0.052	0.192	0.301	0.346
Maximum frequency (context)	0.299	0.273	0.219	0.312
Sliding window	0.178	0.199	0.200	0.091
Word distance model	0.436	0.371	0.332	0.259
Kneser-Ney language model	0.481	0.577	0.762	0.791
Kneser-Ney language model + cache	0.500	0.612	0.755	0.693
Embedding Model (context+query)	0.235	0.297	0.368	0.356
Embedding Model (query)	0.418	0.462	0.575	0.560
Embedding Model (window)	0.457	0.486	0.622	0.619
Embedding Model (window+position)	0.488	0.555	0.722	0.683
LSTMs (query)	0.500	0.613	0.811	**0.819**
LSTMs (context+query)	0.512	0.626	**0.820**	0.812
Contextual LSTMs (window context)	0.535	0.628	0.803	0.798
MemNNs (lexical memory)	0.519	0.647	0.818	0.785
MemNNs (window memory)	0.542	0.591	0.693	0.704
MemNNs (sentential memory + PE)	0.297	0.342	0.451	0.360
MemNNs (window memory + self-sup.)	**0.704**	**0.642**	0.688	0.696

A.3 Ablation Study on CNN QA

METHODS	VALIDATION	TEST
MEMNNS (WINDOW MEMORY + SELF-SUP. + EXCLUD. COOCURRENCES)	0.635	0.684
MEMNNS (WINDOW MEMORY + SELF-SUP.)	0.634	0.668
MEMNNS (WINDOW MEM. + SELF-SUP.) -TIME	0.625	0.659
MEMNNS (WINDOW MEM. + SELF-SUP.) -SOFT MEMORY WEIGHTING	0.604	0.620
MEMNNS (WINDOW MEM. + SELF-SUP.) -TIME -SOFT MEMORY WEIGHTING	0.592	0.613
MEMNNS (WINDOW MEM. + SELF-SUP. + ENSEMBLE)	0.649	0.684
MEMNNS (WINDOW MEM. + SELF-SUP. + ENSEMBLE) -TIME	0.642	0.679
MEMNNS (WINDOW MEM. + SELF-SUP. + ENSEMBLE) -SOFT MEMORY WEIGHTING	0.612	0.641
MEMNNS (WINDOW MEM. + SELF-SUP. + ENSEMBLE) -TIME -SOFT MEMORY WEIGHTING	0.600	0.640

(*Soft memory weighting*: the softmax to select the best candidate in test as defined in Section 6.3.2)

A.4 Effects of Anonymising Entities in CBT

METHODS	NAMED ENTITIES	COMMON NOUNS	VERBS	PREPOSITIONS
MEMNNS (WORD MEM.)	0.431	0.562	0.798	0.764
MEMNNS (WINDOW MEM.)	0.493	0.554	0.692	0.674
MEMNNS (SENTENCE MEM.+PE)	0.318	0.305	0.502	0.326
MEMNNS (WINDOW MEM.+SELF-SUP.)	0.666	0.630	0.690	0.703
ANONYMIZED MEMNNS (WINDOW +SELF-SUP.)	0.581	0.473	0.474	0.522

To see the impact of the anonymisation of entities and words as done in CNN QA on the self-supervised Memory Networks on the CBT, we conducted an experiment where I replaced the mentions of the ten candidates in each question by anonymised placeholders in train, validation and test. The table above shows results on CBT test set in an anonymised setting (last row) compared to MemNNs in a non-anonymised setting (rows 2-5). Results indicate that this has a relatively low impact on named entities but a larger one on more syntactic tasks like prepositions or verbs.

A.5 Candidates and Window Memories in CBT

In the main results in Table 6.2 the window memory is constructed as the set of windows over the candidates being considered for a given question. Training of MEMNNS (WINDOW MEMORY) is performed by making gradient steps for questions, with the true answer word as the target compared against all words in the dictionary as described in Sec. 6.3.1. Training of MEMNNS (WINDOW MEMORY + SELF-SUP.) is performed by making gradient steps for questions, with the true answer word as the target compared against all other candidates as described in Sec. 6.3.2. As MEMNNS (window memory + self-sup.) is the best performing method for named entities and common nouns, to see the impact of these choices I conducted some further experiments with variants of it.

Firstly, window memories do not have to be restricted to candidates, we could consider all possible windows. Note that this does not make any difference at evaluation time on CBT as one would still evaluate by multiple choice using the candidates, and those extra windows would not contribute to the scores of the candidates. However, this may make a difference to the weights if used at training time. We call this "all windows" in the experiments to follow.

Secondly, the self-supervision process does not have to rely on there being known candidates: all that is required is a positive label, in that case I can perform gradient steps with the true answer word as the target compared against all words in the dictionary as described in Sec. 6.3.1, while still using hard attention supervision as described in 6.3.2. We call this "all candidates" in the experiments to follow.

Thirdly, one does not have to try to train on only the *questions* in CBT, but can treat the children's books as a standard language modeling task. In that case, *all candidates* and *all windows* must be used, as multiple choice questions have not been constructed for every single word (although indeed many of them are covered by the four word classes). I call this "LM" (for language modeling) in the experiments to follow.

Results with these two alternatives are presented in Table A.1, the new variants are the last three rows. Overall, the differing approaches have relatively little impact on the results, as all of them provide superior results on named entities and common nouns than without self-supervision. However, note that the use of all windows or LM rather than candidate windows does impact training and testing speed.

Methods	Named Entities	Common Nouns	Verbs	Prepositions
MemNNs (lexical memory)	0.431	0.562	0.798	0.764
MemNNs (window memory)	0.493	0.554	0.692	0.674
MemNNs (sentential memory + PE)	0.318	0.305	0.502	0.326
MemNNs (window memory + self-sup.)	**0.666**	**0.630**	0.690	0.703
MemNNs (all windows + self-sup.)	0.648	0.604	0.711	0.693
MemNNs (all windows + all candidates + self-sup.)	0.639	0.602	0.698	0.667
MemNNs (LM + self-sup.)	0.638	0.605	0.692	0.647

Table A.1: Results on CBT test set when considering all windows or candidates.