How Much Competence Is There in Performance? Assessing the Distributional Hypothesis in Word Bigrams

Johann Seltmann
University of Potsdam[◇]
jseltmann@uni-potsdam.de

Luca Ducceschi
University of Trento[♣]
luca.ducceschi@unitn.it

Aurélie Herbelot
University of Trento[♠]
aurelie.herbelot@unitn.it

[◇]Department of Linguistics, [♣] Dept. of Psychology and Cognitive Science,[♠]Center for Mind/Brain Sciences,
Dept. of Information Engineering and Computer Science

Abstract

The field of Distributional Semantics (DS) is built on the 'distributional hypothesis', which states that meaning can be recovered from statistical information in observable language. It is however notable that the computations necessary to obtain 'good' DS representations are often very involved, implying that if meaning is derivable from linguistic data, it is not directly encoded in it. This prompts questions related to fundamental questions about language acquisition: if we regard text data as linguistic *performance*, what kind of 'innate' mechanisms must operate over that data to reach *competence*? In other words, how much of semantic acquisition is truly data-driven, and what must be hard-encoded in a system's architecture? In this paper, we introduce a new methodology to pull those questions apart. We use state-of-the-art computational models to investigate the amount and nature of transformations required to perform particular semantic tasks. We apply that methodology to one of the simplest structures in language: the word bigram, giving insights into the specific contribution of that linguistic component.[1]

1 Introduction

The traditional notions of *performance* and *competence* come from Chomsky's work on syntax (Chomsky, 1965), where much emphasis is put on the mental processes underpinning language acquisition. Chomsky posits the existence of a Universal Grammar, *innate* in the human species, which gets specialised to the particular language of a speaker. By exposure to the imperfect utterances of their community (referred to as *performance* data), an individual configures their UG to reach some ideal knowledge of that community's language, thereby reaching *competence*.

The present paper borrows the notions of 'performance', 'competence' and 'innateness' to critically analyse the semantic 'acquisition' processes simulated by Distributional Semantics models (DSMs). Our goal is to tease apart how much of their observed competence is due to the performance data they are exposed to, and how much is contributed by 'innate' properties of those systems, i.e. by their specific architectures.

DSMs come in many shapes. Traditional unsupervised architectures rely on counting co-occurrences of words with other words or documents (Turney and Pantel, 2010; Erk, 2012; Clark, 2012). Their neural counterparts, usually referred to as 'predictive models' (Baroni et al., 2014) learn from a language modelling task over raw linguistic data (e.g. Word2Vec, Mikolov et al., 2013, GloVE Pennington et al., 2014). The most recent language embedding models (Vaswani et al., 2017; Radford et al., 2018), ELMo (Peters et al., 2018), or BERT (Devlin et al., 2018) compute *contextualised* word representations and sentence representations, yielding state-of-the-art results on sentence-related tasks, including translation. In spite of their differences, all models claim to rely on the *Distributional Hypothesis* (Harris, 1954; Firth, 1957), that is, the idea that distributional patterns of occurrences in language correlate with specific aspects of meaning.

The Distributional Hypothesis, as stated in the DSM literature, makes semantic acquisition sound like an extremely data-driven procedure. But we should ask to what extent meaning indeed is to be found in statistical patterns. The question is motivated by the observation that the success of the latest DSMs relies on complex mechanisms being applied to the underlying linguistic data or the task at hand (e.g. attention, self-attention, negative sampling, particular objective functions). Such mechanisms have been shown to apply very significant transformations to the original input data: for instance, the Word2Vec objective function introduces parallelisms in the space that make it perform particularly well on analogy tasks (Gittens et al., 2017). Models such as BERT apply extensive processing to the input through stacks of encoders. So while meaning can be *derived* from training regimes involving raw data, it is not directly encoded in it.

Interestingly, Harris himself (Harris, 1954) points out that a) distributional structure is in no simple relation to the structure of meaning; b) different distributions in language encode different phenomena with various levels of complexity. We take both points as highlighting the complex relation between *linguistic* structure and the *cognitive* mechanisms that are necessary to apply to the raw input to retrieve semantic information. The point of our paper is to understand better what is encoded in observable linguistic structures (at the level of raw performance data), and how much distortion of the input needs to be done to acquire meaning (i.e. what cognitive mechanisms are involved in learning semantic competence).

In the spirit of Harris, we think it is worth investigating the behaviour of specific components of language and understand which aspects of meaning they encode, and to what extent. The present work illustrates our claim by presenting an exploratory analysis of one of the simplest recoverable structure in corpora: the word bigram. Our methodology is simple: we test the raw distributional behaviour of the constituent over different tasks, comparing it to a state-of-the-art model. We posit that each task embodies a specific aspect of competence. By inspecting the difference in performance between the simplest and more complex models, we get some insight into the way a particular structure (here, the bigram) contributes to the acquisition of specific linguistic faculties. The failures of raw linguistic data to encode a particular competence points at some necessary, 'innate' constraint of the acquisition process, which might be encoded in a model's architecture as well as the specific task that it is required to solve.

In what follows, we propose to investigate the behaviour of the bigram with respect to three different levels of semantic competence, corresponding to specific tasks from the DS literature: a) word relatedness; b) sentence relatedness; c) sentence autoencoding (Turney, 2014; Bowman et al., 2016). The first two tasks test to which extent the linguistic structure under consideration encodes topicality: if it does, it should prove able to cluster together similar lexical items, both in isolation and as the constituents of sentences. The third task evaluates the ability of a system to build a sentence representation and from that representation alone, recover the original utterance. That is, it tests *distinguishability* of representations. Importantly, distinguishability is at odds with the relatedness tasks which favour *clusterability*. The type of space learned from the raw data will necessarily favour one or the other. Our choice of tasks thus allows us to understand which type of space can be learned from the bigram: we will expand on this in our discussion (§6).[2]

2 Related work

The Distributional Hypothesis is naturally encoded in *count-based* models of Distributional Semantics (DS), which build lexical representations by gathering statistics over word co-occurrences. Over the years, however, these simple models have been superseded by so-called *predictive* models such as Word2Vec (Mikolov et al., 2013) or FastText (Bojanowski et al., 2017), which operate via language modeling tasks. These neural models involve sets of more or less complex procedures, from subsampling to negative sampling and subword chunking, which give them a clear advantage over methods that stick more closely to distributions in corpora. At the level of higher constituents, the assumption is that a) additional composition functions must be learned over the word representations to generate meaning 'bottom-up' (Clark, 2012; Erk, 2012); b) the semantics of a sentence influences the meaning of its parts 'top-down', leading to a notion of contextualised word semantics, retrievable by yet another class of distributional models (Erk and Padó, 2008; Erk et al., 2010; Thater et al., 2011; Peters et al., 2018). Bypassing the word level, some research investigates the meaning of sentences directly. Following from classic work on seq2seq architectures and attention, various models have been proposed to generate sentence embeddings through highly param-

[2] Our code for this investigation can be found under https://github.com/sejo95/DSGeneration.git.

eterised stacks of encoders (Vaswani et al., 2017; Radford et al., 2018; Devlin et al., 2018).

This very brief overview of work in DS shows the variety of models that have been proposed to encode meaning at different levels of constituency, building on more and more complex mechanisms. Aside from those efforts, much research has also focused on finding ideal hyperparameters for the developed architectures (Bullinaria and Levy, 2007; Baroni et al., 2014), ranging from the amount of context taken into account by the model to the type of task it should be trained on. Overall, it is fair to say that if meaning can be retrieved from raw language data, the process requires knowing the right transformations to apply to that data, and the right parametrisation for those transformations, including the type of linguistic structure the model should focus on. One important question remains for the linguist to answer: how much semantics was actually contained in corpus statistics, and where? We attempt to set up a methodology to answer this question, and use two different types of tasks (relatedness and autoencoding) to support our investigation.

While good progress has been made in the DS community on modelling relatedness, distinguishability has received less attention. Some approaches to autoencoding suggest using syntactic elements (such as syntax trees) for decomposition of an embedding vector into a sentence (Dinu and Baroni, 2014; Iyyer et al., 2014). However, some research suggests that this may not be necessary and that continuous bag-of-words representations and n-gram models contain enough word order information to reconstruct sentences (Schmaltz et al., 2016; Adi et al., 2017). Our own methodology is inspired by White et al. (2016b), who decode a sentence vector into a bag of words using a greedy search over the vocabulary. In order to also recover word order, those authors expand their original system in White et al. (2016a) by combining it with a traditional trigram model, which they use to reconstruct the original sentence from the bag of words.

3 Methodology

3.1 A bigram model of Distributional Semantics

We construct a count-based DS model by taking bigrams as our context windows. Specifically, for a word w_i, we construct an embedding vector $\vec{v_i}$ which has one entry for each word w_j in the model. The entry $\vec{v}_{ij}$ then contains the bigram probability $p(w_j|w_i)$.

We talked in our introduction of 'raw' linguistic structure without specifying at which level it is to be found. Following Church and Hanks (1990), we consider the joint probability of two events, relative to their probability of occurring independently, to be a good correlate of the fundamental psycholinguistic notion of *association*. As per previous work, we thus assume that a PMI-weighted DS space gives the most basic representation of the information contained in the structure of interest. For our bigram model, the numerator and denominator of the PMI calculation exactly correspond to elements in our bigram matrix B weighted by elements of our unigram vector U:

$$pmi(w_i, w_j) \equiv \log \frac{p(w_j|w_i)}{p(w_j)} \tag{1}$$

In practice, we use PPMI weighting and map every negative PMI value to 0.

Word relatedness: following standard practice, we compute relatedness scores as the cosine similarity of two PPMI-weighted word vectors, $cos(\vec{w_i}, \vec{w_j})$. For evaluation, we use the MEN test collection (Bruni et al., 2014), which contains 3000 word pairs annotated for relatedness; we compute the spearman ρ correlation between system and human scores.

Sentence relatedness: we follow the proof given by Paperno and Baroni (2016), indicating that the meaning of a phrase ab in a count-based model with PMI weighting is roughly equivalent to the addition of the PMI-weighted vectors of a and b (shifted by some usually minor correction). Thus, we can compute the similarity of two sentences $S1$ and $S2$ as:

$$cos(\sum_{w_i \in S_1} \vec{w_i}, \sum_{w_j \in S_2} \vec{w_j}) \tag{2}$$

We report sentence relatedness scores on the SICK dataset (Marelli et al., 2014), which contains 10,000 utterance pairs annotated for relatedness. We calculate the relatedness for each pair in the dataset and order the pairs according to the results. We then report the spearman correlation between the results of the model and the ordering of the dataset.

Autoencoding of sentences: White et al. (2016b) encode a sentence as the sum of the word embedding vectors of the words of that sentence. They *decode* that vector (the *target*) back into a bag of words in two steps. The first step, *greedy addition* begins with an empty bag of words. In each step a word is selected, such that the sum of the word vectors in the bag and the vector of the candidate item is closest to the target (using Euclidian distance as similarity measure). This is repeated until no new word could bring the sum closer to the target than it already is. The second step, *n-Substitution* begins with the bag of n words found in the greedy addition. For each subbag of size $m \leq n$ it considers replacing it with another possible subbag of size $\leq m$. The replacement that brings the sum closest to the target vector is chosen. We follow the same procedure, except that we only consider subbags of size 1, i.e. substitution of single words, for computational efficiency. In addition, the bigram component of our model B lets us turn the bags of words back into an ordered sequence.[3] We use a beam search to perform this step, following Schmaltz et al. (2016).

We evaluate sentence autoencoding in two ways. First, we test the bag-of-words reconstruction on its own, by feeding the system the encoded sentence embedding and evaluating whether it can retrieve *all* single words contained in the original utterance. We report the proportion of perfectly reconstructed bags-of-words across all test instances. Second, we test the entire autoencoding process, including word re-ordering. We use two different metrics: a) **the BLEU score:** (Papineni et al., 2002), which computes how many n-grams of a decoded sentence are shared with several reference sentences, giving a precision score; b) **the CIDEr-D score:** (Vedantam et al., 2015) which accounts for both precision and recall and is computed using the average cosine similarity between the vector of a candidate sentence and a set of reference vectors. For this evaluation, we use the PASCAL-50S dataset (included in CIDEr-D), a caption generation dataset, that contains 1000 images with 50 reference captions each. We encode and decode the first reference caption for each image and use the remaining 49 as reference for the CIDEr and BLEU calculations.

For the actual implementation of the model, we build B and U from 90% of the BNC ($\approx$ 5.4 million sentences), retaining 10% for development purposes. We limit our vocabulary to the 50000 most common words in the corpus, therefore the matrix is of the size 50002×50002, including tokens for sentence beginning and end.

3.2 Comparison

In what follows, we compare our model to two Word2Vec models, which provide an upper bound for what a DS model may be to achieve. One model, W2V-BNC, is trained from scratch on our BNC background corpus, using gensim (Řehůřek and Sojka, 2010) with 300 dimensions, window size ±5, and ignoring words that occur less than five times in the corpus. The other model, W2V-LARGE, is given by out-of-the-box vectors released by Baroni et al. (2014): that model is trained on 2.5B words, giving an idea of the system's performance on larger data. In all cases, we limit the vocabulary to the same 50,000 words included in the bigram model.

Note that given space restrictions, we do not disentangle the contribution of the models themselves and the particular type of linguistic structure they are trained on. Our results should thus be taken as indication of the amount of information encoded in a raw bigram model compared to what can be obtained by a state-of-the-art model using the best linguistic structure at its disposal (here, a window of ±5 words around the target).

4 Results

Word relatedness: the bigram model obtains an acceptable $\rho = 0.48$ on the MEN dataset. W2V-BNC and W2V-LARGE perform very well, reaching $\rho = 0.72$ and $\rho = 0.80$. Note that whilst the bigram model lags well behind W2V, it achieves its score with what is in essence a unidirectional model with window of size 1 – that is, with as minimal input as it can get, seeing 10 times less co-occurrences than W2V-BNC.

Sentence relatedness: the bigram model obtains $\rho = 0.40$ on the sentence relatedness task. Interestingly, that score increases by 10 points, to $\rho = 0.50$, when filtering away frequent words with probability over 0.005. W2V-BNC and W2V-LARGE give respectively $\rho = 0.59$ and $\rho = 0.61$.

Sentence autoencoding: we evaluate sentence autoencoding on sentences from the Brown corpus (Kučera and Francis, 1967), using seven bins

[3] Note that although a bigram language model would normally perform rather poorly on sentence generation, having a constrained bag-of-words to reorder makes the task considerably simpler.

	original sents.		in matrix	
sent. length	W2V	CB	W2V	CB
3-5	0.556	0.792	0.686	0.988
6-8	0.380	0.62	0.646	0.988
9-11	0.279	0.586	0.548	1.0
12-14	0.210	0.578	0.402	1.0
15-17	0.178	0.338	0.366	0.978
18-20	0.366	0.404	0.984	0.974
21-23	0.306	0.392	0.982	0.968

Table 1: Fraction of exact matches in bag-of-word reconstruction (W2V refers to W2V-LARGE)

	all	2-10	11-23
CIDEr-D bigram	**1.940**	**1.875**	**2.047**
BLEU bigram	**0.193**	**0.209**	**0.176**
CIDEr-D random	1.113	1.1	1.134
BLEU random	0.053	0.059	0.045

Table 2: CIDEr-D and BLEU scores on reordering of bags-of-words using our bigram matrix and random reordering. Results are given for all sentences as well as sentences of lengths 2-10 and 11-23.

Original sentence	Reconstruction
They have to be.	they have to be .
Six of these were proposed by religious groups.	by these were six of religious groups proposed .
His reply, he said, was that he agreed to the need for unity in the country now.	the need for the country , in his reply , he said that he was now agreed to unity .

Table 3: Examples of decoded and reordered sentences. All words in the original sentences were retrieved by the model, but the ordering is only perfectly recovered in the first case.

for different sentence lengths (from 3-5 words to 21-23 words). Each bin contains 500 sentences. In some cases, the sentences contained words that aren't present in the matrix and which are therefore skipped for encoding. We thus look at two different values: a) in how many cases the reconstruction returns exactly the words in the sentence; b) in how many cases the reconstruction returns the words in the sentence which are contained in the matrix (results in Table 1).

The bigram model shines in this task: ignoring words not contained in the matrix leads to almost perfect reconstruction. In comparison, the W2V model has extremely erratic performance (Table 1), with scores decreasing as a function of sentence length (from 0.686 for length 3-5 to 0.366 for length 15-17), but increasing again for lengths over 18.

One interesting aspect of the bigram model is that it also affords a semantic competence that W2V does not naturally have: encoding a sequence and decoding it back into an ordered sequence. We inspect how well the model does at that task, compared to a random reordering baseline. Results are listed in Table 2. The bigram model clearly beats the baseline for all sentence lengths. But it is expectedly limited by the small n-gram size provided by the model. Table 3 contains examples of sentences from the brown corpus and their reconstructions. We see that local ordering is reasonably modeled, but the entire sentence structure fails to be captured.

5 Discussion

On the back of our results, we can start commenting on the particular contribution of bigrams to the semantic competences tested here. First, bigrams are moderately efficient at capturing relatedness: in spite of encoding extremely minimal co-occurrence information, they manage to make for two thirds of W2V's performance, trained on the same data with a much larger window and a complex algorithm (see $\rho = 0.48$ for the bigram model vs $\rho = 0.72$ for W2V-BNC). So relatedness, the flagship task of DS, seems to be present in the most basic structures of language use, although in moderate amount.

The result of the bigram model on sentence relatedness is consistent with its performance at the word level. The improved result obtained by filtering out frequent words, though, reminds us that logical terms are perhaps not so amenable to the distributional hypothesis, despite indications to the contrary (Abrusán et al., 2018).

As for sentence autoencoding, the excellent results of the bigram model might at first be considered trivial and due to the dimensionality of the space, much larger for the bigram model than for W2V. Indeed, at the bag-of-words level, sentence reconstruction can in principle be perfectly achieved by having a space of the dimensionality of the vocabulary, with each word symbolically expressed as a one-hot vector.[4] However,

[4] To make this clear, if we have a vocabulary $V =$

as noted in §2, the ability to encode relatedness is at odds with the ability to distinguish between meanings. There is a trade-off between having a high-dimensionality space (which allows for more discrimination between vectors and thus easier reconstruction – see White et al., 2016b) and capturing latent features between concepts (which is typically better achieved with lower dimensionality). Interestingly, bigrams seem to be biased towards more symbolic representations, generating representations that distinguish very well between word meanings, but they do also encapsulate a reasonable amount of lexical information. This makes them somewhat of a hybrid constituent, between proper symbols and continuous vectors.

6 Conclusion

So what can be said about bigrams as distributional structure? They encode a very high level of lexical discrimination while accounting for some basic semantic similarity. They of course also encode minimal sequential information which can be used to retrieve local sentence ordering. Essentially, they result in representations that are perhaps more 'symbolic' than continuous. It is important to note that the reasonable correlations obtained on relatedness tasks were achieved *after* application of PMI weighting, implying that the raw structure requires some minimal preprocessing to generate lexical information.

On the back of our results, we can draw a few conclusions with respect to the relation of performance and competence at the level of bigrams. Performance data alone produces very distinct word representations without any further processing. Some traces of lexical semantics are present, but require some hard-encoded preprocessing step in the shape of the PMI function. We conclude from this that as a constituent involved in acquisition, the bigram is mostly a marker of the uniqueness of word meaning. Interestingly, we note that the notion of contrast (words that differ in form differ in meaning) is an early feature of children's language acquisition (Clark, 1988). The fact that it is encoded in one of the most simple structures in language is perhaps no coincidence.

In future work, we plan a more encompassing study of other linguistic components. Crucially, we will also investigate which aspects of state-of-the-art models such as W2V contribute to score improvement on lexical aspects of semantics. We hope to thus gain insights into the specific cognitive processes required to bridge the gap between raw distributional structure as it is found in corpora, and actual speaker competence.

$\{cat, dog, run\}$ and we define $cat = [100]$, $dog = [010]$ and $run = [001]$, then, trivially, $[011]$ corresponds to the bag-of-word $\{dog, run\}$.

References

Márta Abrusán, Nicholas Asher, and Tim Van de Cruys. 2018. Content vs. function words: The view from distributional semantics. In *Proceedings of Sinn und Bedeutung 22*.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. International Conference on Learning Representations (ICLR), Toulon, France.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)*, pages 238–247.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21. Association for Computational Linguistics.

E. Bruni, N. K. Tran, and M. Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

John A Bullinaria and Joseph P Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study: A computational study. *Behavior Research Methods*, 39(3):510–526.

Noam Chomsky. 1965. *Aspects of the theory of syntax*. MIT Press.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Eve V Clark. 1988. On the logic of contrast. *Journal of Child Language*, 15(2):317–335.

Stephen Clark. 2012. Vector space models of lexical meaning. In Shalom Lappin and Chris Fox, editors, *Handbook of Contemporary Semantics – second edition*. Wiley-Blackwell.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Georgiana Dinu and Marco Baroni. 2014. How to make words with vectors: Phrase generation in distributional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 624–633.

Katrin Erk. 2012. Vector space models of word meaning and phrase meaning: a survey. *Language and Linguistics Compass*, 6:635–653.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP2008)*, pages 897–906, Honolulu, HI.

Katrin Erk, Sebastian Padó, and Ulrike Padó. 2010. A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4):723–763.

John Rupert Firth. 1957. *A synopsis of linguistic theory, 1930–1955*. Philological Society, Oxford.

Alex Gittens, Dimitris Achlioptas, and Michael W Mahoney. 2017. Skip-gram- zipf+ uniform= vector additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–76.

Zelig Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Mohit Iyyer, Jordan Boyd-Graber, and Hal Daumé III. 2014. Generating sentences from semantic vector space representations. In *NIPS Workshop on Learning Semantics*.

Henry Kučera and Winthrop Nelson Francis. 1967. *Computational analysis of present-day American English*. Dartmouth Publishing Group.

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

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Denis Paperno and Marco Baroni. 2016. When the whole is less than the sum of its parts: How composition affects pmi values in distributional semantic vectors. *Computational Linguistics*, 42(2):345–350.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. `http://is.muni.cz/publication/884893/en`.

Allen Schmaltz, Alexander M. Rush, and Stuart Shieber. 2016. Word ordering without syntax. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2319–2324, Austin, Texas. Association for Computational Linguistics.

S. Thater, H. Fürstenau, and M. Pinkal. 2011. Word meaning in context: A simple and effective vector model. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, Chiang Mai, Thailand.

Peter D. Turney. 2014. Semantic composition and decomposition: From recognition to generation. *CoRR*, abs/1405.7908.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

R. Vedantam, C. L. Zitnick, and D. Parikh. 2015. Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

L. White, R. Togneri, W. Liu, and M. Bennamoun. 2016a. Modelling sentence generation from sum of word embedding vectors as a mixed integer programming problem. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 770–777.

Lyndon White, Roberto Togneri, Wei Liu, and Mohammed Bennamoun. 2016b. Generating bags of words from the sums of their word embeddings. In *17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*.