Representing lexical ambiguity in prototype models of lexical semantics

Barend Beekhuizen
Department of Language Studies
University of Toronto, Mississauga
Depts. of Linguistics and Computer Science
University of Toronto
barend@cs.toronto.edu

Chen Xuan Cui
Department of Computer Science
University of Toronto
bobcui@cs.toronto.edu

Suzanne Stevenson
Department of Computer Science
University of Toronto
suzanne@cs.toronto.edu

Abstract

We show, contrary to some recent claims in the literature, that prototype distributional semantic models (DSMs) are capable of representing multiple senses of ambiguous words, including infrequent meanings. We propose that word2vec contains a natural, model-internal way of operationalizing the disambiguation process by leveraging the two sets of representations word2vec learns, instead of just one as most work on this model does. We evaluate our approach on artificial language simulations where other prototype DSMs have been shown to fail. We furthermore assess whether these results scale to the disambiguation of naturalistic corpus examples. We do so by replacing all instances of sampled pairs of words in a corpus with pseudo-homonym tokens, and testing whether models, after being trained on one half of the corpus, were able to disambiguate pseudo-homonyms on the basis of their linguistic contexts in the second half of the corpus. We observe that word2vec well surpasses the baseline of always guessing the most frequent meaning to be the right one. Moreover, it degrades gracefully: As words are more unbalanced, the baseline is higher, and it is harder to surpass it; nonetheless, Word2vec succeeds at surpassing the baseline, even for pseudo-homonyms whose most frequent meaning is much more frequent than the other.

Keywords: distributed semantic models; word meaning; ambiguity; prototype models; exemplar models; word2vec

Introduction

A central question for the cognitive science of language is how word meanings are represented in the minds of language users. Distributional semantic models (DSMs) represent word meanings as vectors in a high-dimensional space (Landauer & Dumais, 1997; Erk, 2012). The location of these points is based on the words in the neighbouring linguistic context (e.g., a window of words around the target word, or the document the word occurs in). DSMs have been successful in simulating diverse facets of human cognition, such as similarity judgments and analogy completion (e.g., McNamara, 2011; Pereira, Gershman, Ritter, & Botvinick, 2016).

Given that a vast majority of the words in English (and presumably most languages) are ambiguous (Klein & Murphy, 2001), the question arises whether a single vector, which functions as a 'prototype' of the word's meaning, can adequately represent the multiple meanings of an ambiguous word. Several researchers have argued that this is indeed the case. Schütze (1998), Burgess (2001), and Kintsch (2001) each show, using different models and set-ups, how aggregate representations of the context words can disambiguate ambiguous words. Arora, Li, Liang, Ma, and Risteski (2018) propose that word vectors are combinations of the vectors of the component meanings, and that these meaning vectors can be recovered from the 'compact' representation. Further circumstantial evidence for the adequacy of prototype representations comes from the fact that the DSMs successfully model various aspects of cognition even when representing a massively ambiguous vocabulary (Pereira et al., 2016).

Other work, however, suggests that single vector representations are inadequate for the representation of word meaning ambiguity. In the computational linguistics literature, this consideration has led to approaches in which multiple vector representations are learned for a word, each serving as the prototype of *one* of its senses (Reisinger & Mooney, 2010; Li & Jurafsky, 2015). In cognitive science, this assumption has led to the proposal of exemplar-based models, in which a word meaning is represented not as one or more prototype vectors, but as a weighted trace of the memorized contexts that a word occurred in. Jamieson, Avery, Johns, and Jones (2018), for instance, demonstrate that their exemplar-based model of word meaning representation succeeds where two widely-used DSMs (LSA; Landauer & Dumais, 1997 and BEAGLE; Jones & Mewhort, 2007) fail: While the prototype DSMs are able to represent the dominant (most frequent) meaning of a word, subordinate meanings are poorly captured by a single vector, suggesting that these models cannot reliably identify the intended meaning of an ambiguous word in context.

Given the general success of prototype DSMs, such a failure to simulate a key cognitive behaviour would indeed be worrisome if it applied to the entire class of approaches. However, Beekhuizen, Milić, Armstrong, and Stevenson (2018) show in a series of corpus experiments that not all prototype DSMs behave alike in representing ambiguous meanings. In this paper, we will argue that claims concerning the inadequacy of prototype DSMs are not justified. We will do so by showing that another prototype DSM, the CBOW algorithm of word2vec, has model-internal properties that enable it to disambiguate word meaning, and to succeed at accurately representing the infrequent meaning of ambiguous words. Crucially, we believe that this success in disambiguating infrequent meanings is driven by the fact that word meaning interpretation is distributed over two sets of representations in word2vec.

Our Approach

Word2vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) is a word embedding model that learns a distributed semantic space that enables it to best predict words from their contexts. Figure 1 illustrates the process graphically, when using the continuous bag-of-words (CBOW) algorithm of word2vec. Each word in the vocabulary is represented as a row vector in the context matrix C and as a column vector in the target word matrix T. At every training step, the model is given a target word t as well as a window of k context words on either side of the target word. The model then learns to best predict the target word from the context words.

To determine its prediction, word2vec first takes the vector representations in C of the k context words and averages them, forming the aggregate context vector a. The context vector a is then compared with the current word representations in T to predict which word is most likely the target in that context. Intuitively, the more similar the context a is to the current vector representation of a word in T, the higher the predicted probability of observing that word. In training, after making a prediction for an example context, the model checks how far it is off from the desired probability distribution – that is, a probability of 1 for observing the given target word t and 0 for all other words – and proportionally updates the vectors in both C and T to minimize this error.

Although word2vec trains both a context matrix C and target matrix T, researchers typically just use one set of the trained representations (those of the context matrix C) as the resulting DSM of word meaning. Then, for disambiguating a word, a natural approach is to combine the vector representations (from that matrix) for the ambiguous word and its (presumably disambiguating) context words, and then to compare the resulting vector to other representations – for instance, synonyms of the two possible meaning of the ambiguous word – from the same matrix, under the assumption that the aggregate vector will be closest to the appropriate synonym (i.e., the one corresponding to the intended meaning of the ambiguous word). This approach has been explored in computational linguistics by Iacobacci, Pilehvar, and Navigli (2016).

In contrast, we propose a novel approach to using word2vec representations in modeling the disambiguation process, by drawing on its *training procedure* to derive the contextual interpretation of a word. Our insight is that both the context matrix C and the target matrix T contain learned knowledge that is important in disambiguation, just as they work together in the training process to form compatible representations of the context and target words (cf. Mitra, Nalisnick, Craswell, & Caruana, 2016). Rather than throwing away this important information and using representations from just one of the matrices, we use both the C and T matrices: We form an aggregate context vector a using C as a representation of the context of a word to be disambiguated, and compare that aggregate vector to representations of synonyms of its possible meanings embedded in the target word matrix T.

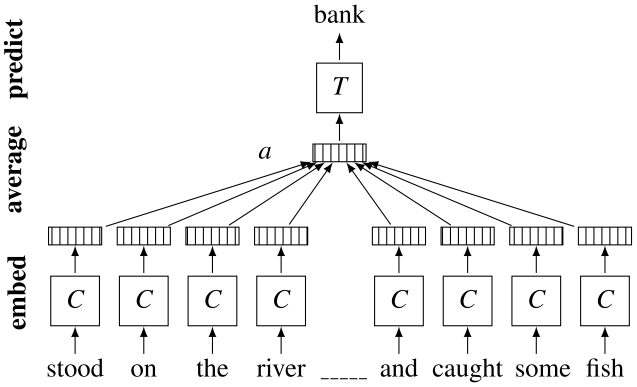

Figure 1: Word2vec model, using the CBOW algorithm

The use of a part of the training procedure is desirable, as it addresses an issue Jamieson et al. (2018) raise, namely that the prototype DSMs that have been shown to work use adhoc patches that are added to the models in order to represent word meanings in context. For word2vec, the aggregate context vector a is a representation of the context that is native to the model, as is the process of comparing a to representations in the T embedding space.

On a conceptual level, we believe that word2vec reflects an important property of word meaning interpretations, namely that they are not completely represented 'in' the word itself (cf. Elman, 2009). The word can be thought to provide a 'sketch' of the meaning (Levinson, 2000) that is completed through inferential processes by the linguistic and extralinguistic context in which it is embedded (e.g. Sperber & Wilson, 1986). This consideration is in fact one of the motivations of an exemplar-based approach. However, in word2vec too, ambiguous meanings are similarly not fully 'represented' in the word vectors of C or T. Rather, C and T, along with the algorithm that compares them, share the responsibility for predicting the target words from the context.

With regard to interpretation of infrequent meanings of a word, this approach gives word2vec an advantage. Given that word2vec's objective is to predict the target word, it suffices to optimize the representations in T so that the vector of the ambiguous target word represents just enough of the infrequent meaning to enable the appropriate context words to predict it (cf. the notion of 'good enough semantic processing' in Ferreira, Bailey, & Ferraro, 2002; Frisson, 2009). In the experiments below, we will illustrate how using the context and target matrix together allows word2vec to represent infrequent word meanings and identify them in context.

Artificial Language Simulations

As a first proof of concept, we replicate the artificial language simulation of Jamieson et al. (2018), which compared disambiguation in an exemplar-based model of word meaning to two prototype DSMs, and found the latter less successful.

An artificial corpus was generated in which the homophone sound form /breɪk/ (i.e., the sound of *break* or *brake*) was used in three contexts corresponding to three different meanings (to brake a car, to break the news, or to break a plate; henceforth all referred to as *break*). The models were tested to see whether they could identify each of the three meanings of *break* used in various disambiguating contexts (e.g., *man break car*, *woman break news*, *woman break plate*). Aside from sentences containing *break*, sentences with verbs that are synonymous to one of the three meanings were generated as well (e.g., *woman stop car*, *man report news*, *man smash glass*). These unambiguous verbs enabled evaluation of whether disambiguation models were able to identify the correct meaning of *break* in the context. Crucially, the corpus was generated either so that all meanings of *break* were equally frequent (balanced), or so that one meaning was 4 times as frequent as the other two meanings (unbalanced). For further details, see Jamieson et al. (2018).

We replicate this experiment for word2vec by generating the corpus in the same way as outlined above and training word2vec on it.[1] We then apply our approach using word2vec (described in the previous section) to see if it can correctly disambiguate the different meanings of *break*. To do so, we see whether the prediction of *break* in a sample context (e.g., *woman+car*) is as strong as the prediction of the appropriate unambiguous word (in this case, *stop*), and much stronger than the inappropriate unambiguous words (those corresponding to the other meanings of *break*). Importantly, the approach follows the flow of the learning procedure of word2vec: we average the representations of the context words in C to create an aggregate context vector a, and then compare a to the representation in T of each of the four different words (*break*, *stop*, *smash*, *report*) to determine the strength of prediction.

As Figure 2 shows, word2vec successfully predicts both *break* and its contextually appropriate synonym, both for the balanced corpus (where the three meanings of *break* are equally frequent) and the unbalanced corpus (where one meaning is more frequent than the others). Note that in all cases, the aggregate context vector is about as similar to the correct unambiguous verb as it is to *break*. For example, the model has learned that in the context of *woman* and *news*, both *report* and *break* are similarly predicted, and thus are similar to each other *in this context*.

Interestingly, we found that this behaviour is only present when both C and T are used; when aggregating context word vectors in C and then comparing them to the vectors of the unambiguous words in C again, the appropriate disambiguation behaviour was not achieved.[2] This means that word2vec is able to represent the contextually disambiguated meaning of a verb through the interaction of its context matrix C with its target matrix T. This behaviour can be expected, as the training algorithm of word2vec optimizes the similarity of the aggregate representations in C (i.e., the vector a) to that of the target word in T. That is: a and the vector of the target word in T are (by design) embedded in the same space, whereas an *aggregate* representation of the context words in T (as opposed to the individual words' representations in T) and the vector of the target word in T are not.

Our successful results contrast with those in Jamieson et al. (2018), who found that, while their exemplar-based word meaning model (Instance Theory of Semantics, henceforth ITS) performed well in this task, the two prototype DSMs – LSA and BEAGLE – were not as successful. In particular, in the balanced condition, all three models show the desirable disambiguation behaviour, but in the unbalanced condition, ITS can successfully disambiguate, but LSA and BEAGLE cannot. For these prototype models, only the most frequent meaning (the *stop* sense of *break*) is activated correctly, whereas the contexts of infrequent meanings (the *report* and *smash* senses) also activate (incorrectly) the most frequent meaning.

While our approach using word2vec demonstrates that a prototype DSM *can* successfully disambiguate infrequent meanings, a potential point of criticism is that our approach may work in an artificial setting like this, but not when the model is trained on a corpus with a realistic vocabulary size and many more unique contexts. After all, a realistic set-up necessitates a far greater degree of compression to allow for a maximally accurate prediction given only 200 dimensions to store all information in — and thereby a greater chance of having infrequent meanings being pushed out by the more frequent ones. Furthermore, the artificial language set-up tests the disambiguation on the data it was trained on, and so we are not directly addressing whether the model can carry out disambiguation in a generalizable way. These issues led to the design of the next experiment.

Disambiguation in a Naturalistic Setting

While the artificial language experiment provides a proof-of-concept of contextual disambiguation, it cannot test whether models have *generalizable* knowledge that scales to *naturalistic* contexts. The obstacle to larger-scale, more realistic scenarios is that testing disambiguation requires knowing the "correct" answer – that is, for any given instance of an ambiguous word in context, we need to know which meaning was intended in order to judge whether a model is performing appropriately. This requires a natural corpus that has the instances of homonyms annotated with the correct meaning in each case.

Since no such corpora of substantial size exist, we follow Arora et al. (2018) in adopting a method of using "pseudo-

[1] In all experiments reported, we used the implementation of word2vec in gensim (Řehůřek & Sojka, 2010), using CBOW with 200 vector dimensions, a window size of 5, a minimum frequency of 1, and otherwise default parameter settings. All software used is available as supplementary material at `https://tinyurl.com/w2vcogsci`.

[2] We also tried other ways to use word2vec, including its Skip-Gram variant, but CBOW with both C and T matrices was the most robust with unbalanced homonyms.

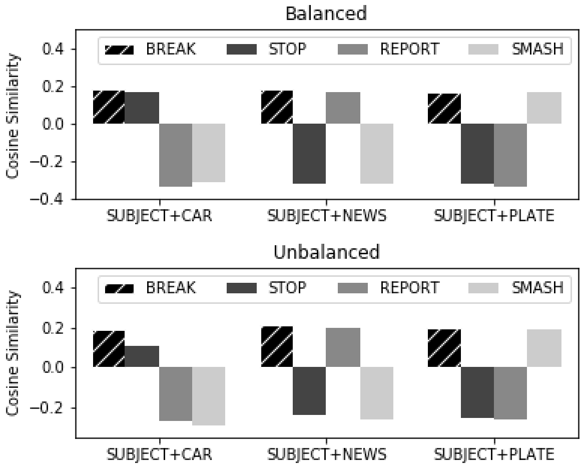

Figure 2: Cosine similarities between the word2vec representations of the context words (on *x*-axis) and the representations of the target words (in legend), for the balanced corpus and the unbalanced corpus.

homonyms” – pairs of words that are considered as if they were a single word. For example, if we consider the set of usages of *pizza* and *water* as if they were a single word with meanings PIZZA and WATER, then we would have a corpus in which all the instances in context of `pizza_water` are known to be disambiguated as either PIZZA or WATER (corresponding to the original word in that instance).

This set-up allows us to present our word2vec-based disambiguation approach with test cases (contexts containing a pseudo-homonym), and see whether it can identify which component meaning of the pseudo-homonym was intended in that context. We similarly evaluate the performance of ITS (Jamieson et al., 2018) on the same data, to see how our approach, based on a prototype DSM, compares to an exemplar-based approach to word meaning.

We use the TASA corpus of Landauer, Foltz, and Laham (1998), with the first half of the corpus as training data, and the second half as test data. Using a training-test split of the data, we made sure the models were actually tested on their capacity to disambiguate target words in novel, unseen contexts. We sampled 100 pairs of non-homonymous words that were similar to one of the real homonyms listed in Armstrong, Tokowicz, and Plaut (2012) in their length, frequency, and relative frequency of the two component meanings. This was done to make sure the pseudo-homonyms displayed similar relevant properties as real homonyms (Piantadosi, Tily, & Gibson, 2012).[3] We next explain how we can test each model under this approach.

Pseudo-homonym set-up for word2vec. For word2vec, we need to modify the corpus to enable training on a set of pseudo-homonyms, which were created by merging two non-homonymous words – e.g., replacing all instances of the words *pizza* and *water* with the single token `pizza_water`. The context and target matrices of word2vec were trained once on the original version of the training data, yielding C and T, and again on the version with pseudo-homonyms, yielding C' and T'. In this way, we have representations both for the pseudo-homonyms and for their component words individually. Then, for each instance of a pseudo-homonym in the test data, say `pizza_water`, we tested whether its aggregate context vector a from C' (based on the pseudo-homonym version of the corpus) was more similar to the correct or incorrect component meaning representation in T – *pizza* or *water* – whichever occurred in the original corpus).[4]

Pseudo-homonym set-up for ITS. ITS (Jamieson et al., 2018) follows the intuition that an accurate representation of word meaning is derived from all previously encountered instances of the word. Starting with words represented as high-dimensional random vectors, ITS represents the *memory trace* of each document in a corpus as the sum of the random vectors of all the words in that document. Word meanings in context are then derived from the matrix of memory traces by presenting the model with a *probe* in the form of a set of words, and retrieving its *echo*: an aggregate of all memory traces, weighted by how similar they are to the probe. Figure 3 presents a graphical representation of the echo retrieval process.

In our ITS set-up, we constructed a matrix of 20K-dimensional memory traces for the training portion of the original TASA corpus. Then, for each instance of either of the component words of a pseudo-homonym in the test data, a context probe was constructed out of the five words to the left and to the right of the word (excluding stopwords and punctuation), plus the two component words of the pseudo-homonym themselves. The echo of this aggregate probe was retrieved and compared to the echo of each component word individually. The component word whose echo had the highest cosine similarity to the echo of the aggregate context probe was selected as the disambiguated meaning.[5]

Results This approach gives us $91,703$ ambiguous pseudo-homonym tokens in the test data, aggregated over the 3 simulations (on average 306 per pseudo-homonym). We find that word2vec scores an overall accuracy (proportion of correctly disambiguated test items) of .85 versus .69 for ITS. This means that overall, word2vec is better able to disambiguate words in their naturalistic contexts.

It is important to also consider how these accuracies compare to a chance baseline – is either model doing better than random guessing? Assuming there is some way to know which is the most frequent (dominant) meaning, a model that always guessed the dominant meaning would achieve a score

[3] Due to the random sampling, we ran three simulations, each with a new set of 100 pseudo-homonyms, and report aggregate findings of the three simulations.

[4] To compare vectors from C' to those from T, we use Orthogonal Procrustes, a standard method, to rotate T to T' so the vectors are all in a compatible vector space.

[5] This set-up was found to yield the best results for ITS compared to other set-ups we tried.

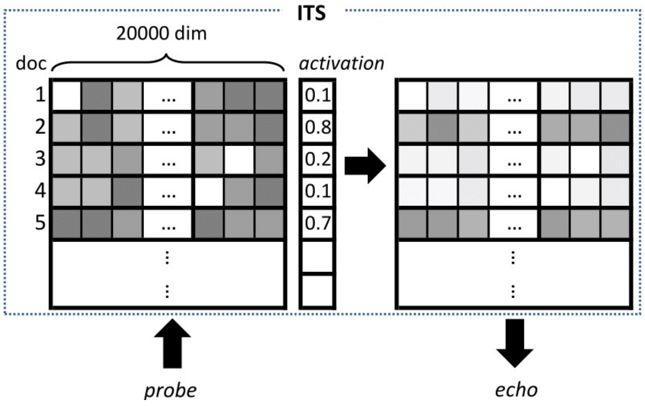

Figure 3: A visual example of the retrieval of an echo in ITS through the selective activation of memory traces when presented with a probe.

of .73 for our pseudo-homonyms – i.e., the average relative frequency of the dominant meaning. This seems like a reasonable baseline to assume, since we are interested in whether a model can learn the non-dominant meanings of ambiguous words. For each simulation, a two-tailed paired-samples t-test compared the accuracy per pseudohomonym for the model predictions and the dominant meaning baseline. In all simulations, word2vec did significantly better than the baseline (Sim. 1: $T = 9.71, p < 0.001$ / Sim. 2: $T = 11.96, p < 0.001$ / Sim. 3: $T = 10.01, p < 0.001$). ITS, however, performed significantly worse than the baseline (Sim. 1: $T = 3.32, p < 0.01$ / Sim. 2: $T = 2.34, p < 0.05$ / Sim. 3: $T = 2.74, p < 0.01$).[6]

Critical for our purposes is whether each model not just performed accurately for words with balanced meanings, but also was able to accurately disambiguate cases where one of the meanings is much more dominant. To assess this, we look at each pseudo-homonym individually. To compare fairly across pseudo-homonyms with different baselines (different degrees of dominance of meanings), we need a measure which looks at the amount by which each model surpasses (or falls short of) that baseline. A common measure to do so is the so-called reduction in error rate over the baseline (RER), defined as the amount by which the model improves over the baseline, divided by the error rate of the baseline.[7]

Figure 4 plots the RER for each pseudo-homonym as a function of its baseline (the relative frequency of its dominant meaning). The lines indicate the best linear fit between the two per simulation (all linear fits with Pearson's r are significant at $p < .001$). Both models display a downward slope across all simulations. This is unsurprising, since we would expect for any model that it is more difficult to disambiguate a very unbalanced homonym toward the infrequent meaning.

However, as can be gleaned from Figure 4, the slopes for word2vec are less negative than those of ITS, a difference that is significant across all three simulations (Sim. 1: $T = 3.03, p < .01$ / Sim. 2: $T = 4.83, p < .001$ / Sim. 3: $T = 2.90, p < .01$). This means that word2vec degrades more gracefully as homonyms become more unbalanced than ITS. Indeed, ITS only surpasses the baseline for relatively balanced items, and is unable to do better than the baseline for items whose most frequent meaning has a relative meaning frequency of around .66 or more. By contrast, the regression lines for word2vec only touch the null line (meaning always guessing the most frequent meaning) for the most unbalanced pseudo-homonyms (right end of the x-axis).

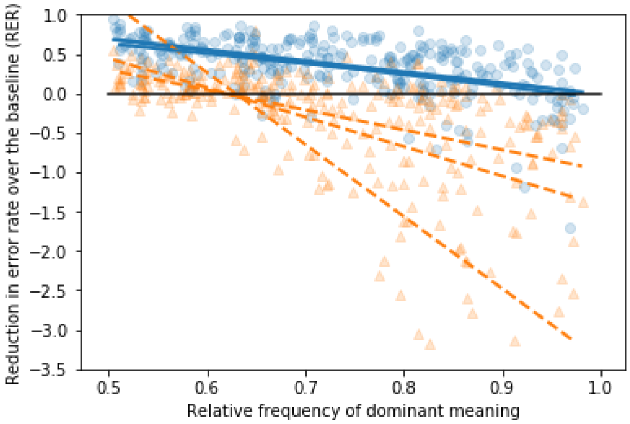

Figure 4: Reduction in error rate over the baseline (RER), aggregated over the three simulations. Dots (orange triangles for ITS, blue circles for word2vec) represent pseudo-homonyms. Regression lines are given for each simulation (orange dashed lines for ITS, blue solid lines (all overlapping) for word2vec). The black line represents zero error rate reduction; values below 0 are error rate increase, above 0 error rate reduction.

This means that, contrary to the predictions of Jamieson et al. (2018), and arguments raised in the computational linguistics literature (Reisinger & Mooney, 2010; Li & Jurafsky, 2015), not all prototype DSMs are unable to represent a contextually-resolved meaning of an unbalanced ambiguous word: word2vec performs adequately on such disambiguation tasks. Scaling up the disambiguation experiment to a more naturalistic corpus size and set of contexts, our approach using word2vec consistently surpasses the most-frequent sense baseline, and can thus be said to robustly resolve lexical ambiguities on the basis of the context words. Furthermore, word2vec degrades gracefully: it is harder to do better than chance for very unbalanced items than it is for balanced ones, but word2vec nonetheless on average surpasses the baseline even for very unbalanced pseudo-homonyms.

General Discussion

In this paper, we set out to show that, contrary to claims in the literature (Griffiths, Steyvers, & Tenenbaum, 2007; Reisinger & Mooney, 2010; Jamieson et al., 2018), proto-

[6] By virtue of transitivity, this also means that word2vec performs better than ITS (Sim. 1: $T = 12.13, p < 0.001$ / Sim. 2: $T = 11.82, p < 0.001$ / Sim. 3: $T = 11.23, p < 0.001$).

[7] That is, $RER = (model\_acc - baseline\_acc)/(1 - baseline\_acc)$

type distributed semantic models are capable of representing infrequent meanings of ambiguous words. We proposed that word2vec contains a natural, model-internal way of operationalizing the disambiguation process, and tested this approach successfully on the artificial language simulations for which Jamieson et al. (2018) showed that other prototype DSMs failed.

Importantly, we further assessed whether these results scaled to the disambiguation of naturalistic corpus examples. We generated a pseudo-homonym corpus by replacing all instances of sampled pairs of words in a corpus with pseudo-homonym tokens. We then trained word2vec on one half of the corpus, and assessed if the model was able to disambiguate pseudo-homonyms on the basis of their linguistic contexts in the second half of the corpus. We observed that our disambiguation approach using word2vec well surpasses the baseline of always guessing the most frequent meaning to be the right one, in contrast to an exemplar-based model (Jamieson et al., 2018). Word2vec moreover degrades gracefully: as words are more unbalanced (i.e., as the most frequent meaning has a higher relative frequency), the baseline is higher, and it is harder to surpass it. Word2vec nonetheless succeeds at surpassing the baseline, even for very unbalanced pseudo-homonyms.

A follow-up question is why Word2vec can represent infrequent meanings while LSA and BEAGLE cannot. It is tempting to speculate that this is due to the fact that word2vec vectors are trained to *predict* words, whereas LSA and BEAGLE vectors reflect the *counting* of words, and prediction-based DSMs have been found to outperform count-based DSMs (Baroni, Dinu, & Kruszewski, 2014). However, Levy and Goldberg (2014) argue the skipgram variant of word2vec performs implicit factorization of a count-based matrix in its objective function, so the actual differences between count-based and prediction-based models are not completely clear. This is an open area of research to which our findings contribute an important data point – i.e., that our approach to using the prediction mechanism of word2vec in semantic disambiguation outperforms a non-predictive approach using count-based DSMs (BEAGLE and LSA, as shown in Jamieson et al., 2018). A relevant future step is the comparison of our approach using the CBOW algorithm of word2vec to other prediction-based models or variants such as skipgram (Mikolov et al., 2013), as well as other contemporary approaches such as GloVe (Pennington, Socher, & Manning, 2014) and ELMo (Peters et al., 2018).

Another option is that it is the use of both the context word and target word matrices that allows us to achieve these results. Whereas off-the-shelf vectors have been used extensively in cognitive modeling experiments, our paper proposes to use a model-internal approach that leverages the fact that word2vec represents meaning as context word vectors *and* as target word vectors. This approach addresses the concern of Jamieson et al. (2018) that many prototype models only have ad hoc ways of carrying out the disambiguation procedure. It furthermore instantiates two critical points of the perspective on lexical semantics put forward by Elman (2009), namely: (1) that the drive to predict upcoming (linguistic) behaviour has sizable impact on the kinds of representations learned, and (2) that the interpretation of a word is always a function of some prior knowledge of the word as well as its context. It is effectively this idea that, combined with high-parametric representations and an abundance of data to train on, has led to the success of contemporary NLP word-meaning models such as ELMo (Peters et al., 2018).

We would like to argue that because of this distributed way in which word2vec learns to predict words, its representations reflect the important point that not all of a word meaning representation needs to be stored 'inside of' the word itself, but also by how word meanings relate to other word meanings (i.e., the 'oppositions' with other lexical items they have; Trubetzkoy, 1969 (1939)), as well as by rich pragmatic interpretive processes (Sperber & Wilson, 1986; Levinson, 2000). An important goal for the cognitive sciences of word meaning is to develop computationally precise models of how these processes work and interact. The present paper constitutes a stepping stone towards that goal.

Acknowledgments

SS and CXC are supported by an NSERC Discovery Grant RGPIN-2017-06506 to SS. We would like to thank the anonymous reviewers for their thoughtful comments.

References

Armstrong, B. C., Tokowicz, N., & Plaut, D. C. (2012). eDom: Norming software and relative meaning frequencies for 544 English homonyms. *Behavior Research Methods*, *44*(4), 1015–1027.

Arora, S., Li, Y., Liang, Y., Ma, T., & Risteski, A. (2018). Linear algebraic structure of word senses, with applications to polysemy. *TACL*, *6*, 483–495.

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings ACL.*

Beekhuizen, B., Milić, S., Armstrong, B., & Stevenson, S. (2018). What company do semantically ambiguous words keep? Insights from distributional word vectors. In *Proceedings CogSci.*

Burgess, C. (2001). Representing and resolving semantic ambiguity: A contribution from high-dimensional memory modeling. In D. S. Gorfein (Ed.), *On the consequences of meaning selection: Perspectives on resolving lexical ambiguity* (pp. 233–261). American Psychological Association.

Elman, J. L. (2009). On the meaning of words and dinosaur bones: Lexical knowledge without a lexicon. *Cognitive science*, *33*(4), 547–582.

Erk, K. (2012). Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, *6*(10), 635-653.

Ferreira, F., Bailey, K. G., & Ferraro, V. (2002). Good-enough representations in language comprehension. *Current directions in psychological science*, *11*(1), 11–15.

Frisson, S. (2009). Semantic underspecification in language processing. *Language and Linguistics Compass*, *3*(1), 111–127.

Griffiths, T. L., Steyvers, M., & Tenenbaum, J. B. (2007). Topics in semantic representation. *Psychological Review*, *114*(2), 211.

Iacobacci, I., Pilehvar, M. T., & Navigli, R. (2016). Embeddings for word sense disambiguation: An evaluation study. In *Proceedings ACL*.

Jamieson, R. K., Avery, J. E., Johns, B. T., & Jones, M. N. (2018). An instance theory of semantic memory. *Computational Brain & Behavior*, *1*(2), 119–136.

Jones, M. N., & Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, *114*(1), 1–37.

Kintsch, W. (2001). Predication. *Cognitive science*, *25*(2), 173–202.

Klein, D. E., & Murphy, G. L. (2001). The representation of polysemous words. *Journal of Memory and Language*, *45*(2), 259–282.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*(2), 211.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), 259–284.

Levinson, S. C. (2000). *Presumptive meanings: The theory of generalized conversational implicature*. MIT Press.

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Proceedings NeurIPS*.

Li, J., & Jurafsky, D. (2015). Do multi-sense embeddings improve natural language understanding? In *Proceedings EMNLP*.

McNamara, D. S. (2011). Computational methods to extract meaning from text and advance theories of human cognition. *Topics in Cognitive Science*, *3*(1), 3–17.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings NeurIPS*.

Mitra, B., Nalisnick, E. T., Craswell, N., & Caruana, R. (2016). A dual embedding space model for document ranking. *CoRR*, *abs/1602.01137*.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings EMNLP*.

Pereira, F., Gershman, S., Ritter, S., & Botvinick, M. (2016). A comparative evaluation of off-the-shelf distributed semantic representations for modelling behavioural data. *Cognitive Neuropsychology*, *33*(3-4), 175–190.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings NAACL* (pp. 2227–2237).

Piantadosi, S. T., Tily, H., & Gibson, E. (2012). The communicative function of ambiguity in language. *Cognition*, *122*(3), 280–291.

Řehůřek, R., & Sojka, P. (2010, May 22). Software Framework for Topic Modelling with Large Corpora. In *Proceedings LREC*.

Reisinger, J., & Mooney, R. J. (2010). Multi-prototype vector-space models of word meaning. In *Human Language Technologies: NAACL* (pp. 109–117).

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

Sperber, D., & Wilson, D. (1986). *Relevance: communication and cognition*. Harvard University Press.

Trubetzkoy, N. S. (1969 (1939)). *Principles of phonology*. University of California Press.