Detecting Asymmetric Semantic Relations in Context: A Case-Study on Hypernymy Detection

Yogarshi Vyas and **Marine Carpuat**
Department of Computer Science
University of Maryland
yogarshi@cs.umd.edu and marine@cs.umd.edu

Abstract

We introduce WHIC[1], a challenging testbed for detecting hypernymy, an asymmetric relation between words. While previous work has focused on detecting hypernymy between word types, we ground the meaning of words in specific contexts drawn from WordNet examples, and require predictions to be sensitive to changes in contexts. WHIC lets us analyze complementary properties of two approaches of inducing vector representations of word meaning in context. We show that such contextualized word representations also improve detection of a wider range of semantic relations in context.

1 Introduction

Language understanding applications like question answering (Harabagiu and Hickl, 2006) and textual entailment (Dagan et al., 2013) benefit from identifying semantic relations between words beyond synonymy and paraphrasing. For instance, given *"Anand plays chess."*, and the question *"Which game does Anand play?"*, successfully answering the question requires knowing that *chess* is a kind of *game*, i.e. *chess* entails *game*. Such lexical entailment relations are asymmetric (*chess* $\implies$ *game*, but *game* $\not\Longrightarrow$ *chess*), and detecting their direction accurately is a challenge.

While prior work has defined lexical entailment as a relation between word types, we argue that it is better defined between word meanings illustrated by examples of usage in context. Ignoring context is problematic since entailment might hold between some senses of the words, but not others. Consider the word *game* in the following contexts:

1. The championship *game* was played in NYC.
2. The hunters were interested in the big *game*.

Given the sentence, *Anand is the world* chess *champion*, chess $\implies$ game in the first context, while chess $\not\Longrightarrow$ game in the second context.

Lexical entailment encompasses several semantic relations, with one important relation being *hypernymy* (Roller et al., 2014; Shwartz et al., 2016). In this work, we focus on hypernymy detection in context, and show that existing resources can be leveraged to automatically create test beds for evaluation. We introduce "Wordnet Hypernyms in Context" (WHIC, pronounced *which)*, a large dataset, automatically extracted from WordNet (Fellbaum, 1998) using examples provided with synsets. Crucially, WHIC includes challenging negative examples that assess the ability of models to detect the direction of hypernymy.

We use WHIC to determine the effectiveness of existing supervised models for hypernymy detection (Roller and Erk, 2016) applied to representations, not only of word types, but of words in context. Such contextualized representations are induced in two ways: the first is based on Context2Vec, a BiLSTM model that embeds contexts and words in the same space (Melamud et al., 2016); the second aims to capture geometric properties of the context in a standard word embedding space built using GloVe (Pennington et al., 2014).

We show that the two contextualized representations improve performance over context-agnostic baselines. The structure of WHIC lets us show that they have complementary properties: Context2Vec-based models have higher recall and tend to identify directionality much better than Glove-based models. We also show that the context-aware representations improve performance on identifying a broader range of semantic relations (Shwartz and Dagan, 2016).

[1] https://github.com/yogarshi/whic

*Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 33–43,
Vancouver, Canada, August 3-4, 2017. ©2017 Association for Computational Linguistics

Words (w_l, w_r)	Exemplars (c_l,c_r)	Does $w_l \implies w_r$?
staff, ***stick***	c_l = He walked with the help of a wooden ***staff***. c_r = The kid had a candied apple on a ***stick***.	Yes
staff, ***body***	c_l = The hospital has an excellent nursing ***staff***. c_r = The whole ***body*** filed out of the auditorium.	Yes
staff, ***stick***	c_l = The hospital has an excellent nursing ***staff***. c_r = The kid had a candied apple on a ***stick***.	No

Table 1: Examples of the context-aware hypernymy detection task

2 Detecting Hypernymy in Context

2.1 Task Definition

We frame hypernymy detection in context as a binary classification task. Each example consists of a 4-tuple (w_l, w_r, c_l, c_r), where w_l and w_r are word types, and c_l and c_r are sentences which illustrate each word usage. The example is treated as positive if $w_l \implies w_r$, given the meaning of each word exemplified by the contexts, and negative otherwise, as can be seen in Table 1.

As mentioned in Section 1, hypernymy is only one specific case of lexical entailment. The nature of entailment relations captured out-of-context can be broader depending on the test beds considered[2]. These relations can include synonymy, hypernymy, some meronymy relations, and also cause-effect relations.

2.2 Motivation

The need to study hypernymy detection in context is important due to several reasons. First, many downstream tasks which might benefit from detecting hypernyms will have words appearing in specific contexts. Second, existing definitions (and, by extension, annotations) of lexical entailment do not explicitly or consistently address polysemy. For instance, the substitutional definition for entailment by Zhitomirsky-Geffet and Dagan (2009) asks the reader to think of a natural sentence that provides the missing context to the two words being considered, thus constraining the possible senses of the two words. On the other hand, Turney and Mohammad (2013) propose a relational definition, inviting the reader to imagine a semantic relation that connects the two words and constrains their possible senses. In contrast, we propose to detect hypernymy between word meanings described by specific contexts.

Lexical entailment or hypernymy in context is also different from recognizing textual entailment (RTE). RTE (Dagan et al., 2006, 2013) involves detecting entailment relations between sentences, while hypernymy is a relation between words. Additionally, the two contexts c_l and c_r in our task can be very different, unlike in textual entailment, where the premise and hypothesis are usually related. For instance, the first example in Table 1 illustrates a scenario where the hypernymy relation holds between ***staff*** and ***stick***, but there is no entailment relationship between the two sentences. On the other hand, the sentence ”*Children smile and wave at the camera.*” entails ”*There are children present.*”, but there is no meaningful hypernymy relationship between words in the two sentences.

Finally, the proposed task is also related to, but different from word sense disambiguation (WSD). Unlike WSD, this task eschews an explicit sense inventory, instead relying on the provided contexts to decide the specific relation between the words. This might provide a more natural way to think about word senses for (untrained) human annotators (Erk et al., 2013). WSD can in principle be used as a preprocessing step to address hypernymy detection in context, but it is not required. Also, WSD remains a challenging task (Moro and Navigli, 2015) and it might introduce errors early in the preprocessing pipeline.

2.3 WHiC : A Dataset for Lexical Entailment in Context

We require a dataset to study hypernymy detection in context to satisfy the following desiderata: (1) the dataset should make it possible to assess the sensitivity of context-aware models to contexts that signal different word senses, and (2) the dataset should help quantify the extent to which models detect the asymmetric direction of hypernymy, rather than symmetric semantic similarity.

[2] We refer the reader to Turney and Mohammad (2013) and Shwartz et al. (2017) for comprehensive surveys of supervised and unsupervised methods for the out-of-context task.

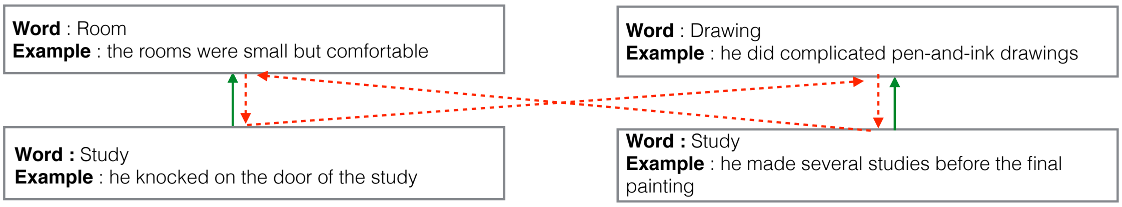
Word : Room
Example : the rooms were small but comfortable
Word : Drawing
Example : he did complicated pen-and-ink drawings
Word : Study
Example : he knocked on the door of the study
Word : Study
Example : he made several studies before the final painting

Figure 1: Sample dataset creation process based on two synsets of the word *study*. The green/solid lines indicate positive examples, while the red/dashed lines indicate negative examples

Existing datasets for lexical entailment (Baroni and Lenci, 2011; Baroni et al., 2012; Kotlerman et al., 2010) have driven progress on the **out of context** task only, and are therefore insensitive to context changes. In addition, they include a variety of negative examples without controlling for entailment direction. For instance, Baroni and Lenci (2011) use cohyponyms and random words as negative examples. Since cohyponyms are words that share a common hypernym (for example, *salsa* and *tango* are cohyponymys with respect to *dance*), hypernymy does not hold between them in any direction. On the other hand, random examples (also used by Baroni et al. (2012)) are likely to be detected using symmetric semantic similarity rather than asymmetric hypernymy detection.

Shwartz and Dagan (2016) recently introduced CONTEXT-PPDB, a dataset for fine-grained lexical inference in context. This dataset consists of word pairs along with a pair of sentential contexts, with a label indicating the semantic relation between the two words in the given contexts. However, since CONTEXT-PPDB only consists of ~3700 sentence pairs, it provides only a smaller number of annotated examples per relation, making it difficult to train large supervised models on (we return to this dataset in Section 5).

We address these gaps by introducing, WHIC, a large dataset automatically derived from WordNet (Fellbaum, 1998). WordNet groups synonyms into *synsets* and defines semantic relations such as hypernymy and meronymy between these synsets. Most synsets are further accompanied by one or more short sentences illustrating the use of the members of the synset. WHIC uses these example sentences as context for the words, and the hypernymy relations to draw candidate word pairs. The process starts from a seed list of words W and proceeds as follows (see Figure 1) :

1. For all word types $w \in W$ obtain synsets S_w.
2. For each synset $i \in S_w$, pick a hypernym synset s_h^i, with a corresponding word form w_h^i. Also obtain c^i and c_h^i which are example sentences corresponding to w^i and w_h^i respectively - (w^i, w_h^i, c^i, c_h^i) serves as a positive example. Repeat this process for all hypernyms (solid/green arrows in Figure 1).
3. **Permute** the positive examples to get negative examples. From (w^i, w_h^i, c^i, c_h^i) and (w^j, w_h^j, c^j, c_h^j), generate negative examples (w^i, w_h^j, c^i, c_h^j) and (w^j, w_h^i, c^j, c_h^i) (longer dashed/red arrows in Figure 1).
4. **Flip** the positive examples to generate negative examples. From (w^i, w_h^i, c^i, c_h^i) generate the negative example (w_h^i, w^i, c_h^i, c^i) (shorter dashed/red arrows in Figure 1).

We run this process using the 9000 most frequent words from Wikipedia as W (after filtering the top 1000 as stopwords). This yields a total of 5239 positive examples, 12303 negative examples from Step 3, and 5239 negative examples from Step 4.

WHIC satisfies the desiderata outlined above. The dataset has a well-defined focus, since we only pick hypernym-hyponym pairs. The negative examples generated in Steps 3 and 4 require discriminating between different word senses and entailment directions. Finally, with over 22000 examples distributed over 6000 word pairs, the dataset is large enough to train large supervised models. We define a 70/5/25 train/dev/test split, and ensure that each set contains different word pairs, to avoid memorization and overfitting (Levy et al., 2015).

3 Representing Words and their Contexts for Entailment

How can we construct representations of the meaning of target words w_l and w_r, and their respective exemplar contexts c_l and c_r?

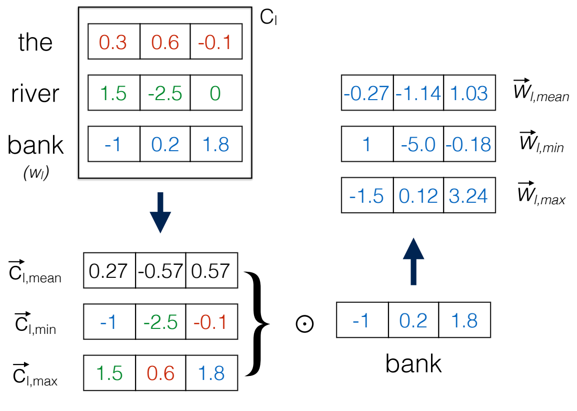

Figure 2: Constructing word-in-context representations for "bank", in the context "the river bank". ⊙ indicates element-wise multiplication.

We will construct representations for c_l, and c_r, and create context-aware representations for w_l and w_r by "masking" their word embeddings with the embeddings for c_l and c_r (Section 3.3). We compare two approaches to representing c_l and c_r. The first (Section 3.1) builds on standard representations for word types, which have proven useful for detecting lexical entailment and other semantic relations out of context (Baroni et al., 2012; Kruszewski and Baroni, 2015; Vylomova et al., 2016; Turney and Mohammad, 2013). The second approach (Section 3.2) uses a recurrent neural model to embed words and contexts in the same space, allowing direct comparisons between them.

3.1 Creating Context Representations from Word Type Representations

Given an example (w_l, w_r, c_l, c_r), let $\vec{w}_l$ and $\vec{w}_r$ refer to the context-agnostic representations of w_l and w_r, and let C_l and C_r represent the matrices obtained by row-wise stacking of the context-agnostic representations of words in c_l and c_r respectively.

Following Thater et al. (2011); Erk and Padó (2008), we apply a filter to word type representations to highlight the salient dimensions of the exemplar context, emphasizing relevant dimensions and downplaying unimportant ones. However, while prior work represents context by averaging word vectors, we propose richer representations that better capture the salient geometrical properties of the exemplar context that might get lost by averaging.

We construct fixed length representations for the contexts c_l and c_r by running convolutional filters over C_l and C_r. Specifically, we calculate the column-wise maximum, minimum and the mean over the matrices C_l and C_r, as done by Tang et al. (2014) for supervised sentiment classification. This yields three d-dimensional vectors for c_l ($\vec{c}_{l,max}$, $\vec{c}_{l,min}$, $\vec{c}_{l,mean}$), and three d-dimensional vectors for c_r ($\vec{c}_{r,max}$, $\vec{c}_{r,min}$, $\vec{c}_{r,mean}$). Computing the maximum and minimum across all vector dimensions captures the exterior surface of the "instance manifold" (the volume in embedding space within which all words in the instance reside), while the mean summarizes the density per-dimension within the manifold (Hovy, 2015).

3.2 LSTM-based Context Representations: Context2Vec

An alternative approach to contextualizing word representations is to directly compare the representations of words with representations of contexts. This can be done using Context2Vec (Melamud et al., 2016), a neural model that, given a target word and its sentential context, embeds both the word and the context in the same low-dimensional space using a BiLSTM, with the objective of having the context predict the target word via a log-linear model. This model approaches the state-of-the-art on lexical substitution, sentence completion, and supervised word sense disambiguation. For each example (w_l, w_r, c_l, c_r), we extract the word type representations $\vec{w}_{l,c2v}$ and $\vec{w}_{l,c2v}$ from Context2Vec, as well as the context representations $\vec{c}_{l,c2v}$, and $\vec{c}_{r,c2v}$.

3.3 Context-aware Masked Representations

Given these two methods to learn representations for words and their contexts, we also learn context aware word representations for the target words. We transform initial context-agnostic representations for target word types by taking an element-wise product of the word type vectors with vectors representing the context.

Specifically, for the context representations learned in Section 3.1, we take an element-wise product of the word type vectors ($\vec{w}_*$) with ($\vec{c}_{*,max}$, $\vec{c}_{*,min}$, $\vec{c}_{*,mean}$) where $* \in \{l, r\}$. This yields three d-dimensional vectors for w_l ($\vec{w}_{l,max}$, $\vec{w}_{l,min}$, $\vec{w}_{l,mean}$), and three for w_r ($\vec{w}_{r,max}$, $\vec{w}_{r,min}$, $\vec{w}_{r,mean}$). We refer to our final word-in-context representations for w_l and w_r as $\vec{w}_{l,mask}$ and $\vec{w}_{r,mask}$ respectively, where $\vec{w}_{l,mask}$ is the concatenation of $\vec{w}_{l,max}$, $\vec{w}_{l,min}$, $\vec{w}_{l,mean}$, and $\vec{w}_{r,mask}$ is also similarly constructed.

For the word and context representations obtained from Context2Vec (Section 3.2), we create the context-aware representations $\vec{w}_{l,c2v,mask}$ by vector multiplication between $\vec{w}_{l,c2v}$ and $\vec{c}_{l,c2v}$. We also obtain $\vec{w}_{r,c2v,mask}$ similarly.

4 Comparing Words and Contexts for Entailment

Given the word, context, and word-in-context representations described above, we predict entailment via supervised classification.

Our classifier is the *Hypernymy-Feature detector* (Roller and Erk, 2016), which is the current state-of-the-art supervised model for detecting hypernymy on several datasets. This model aims to overcome the shortcomings of previous supervised hypernymy detection models, which used linear classifiers on top of concatenation of the two vectors representing the target words. These models only captured notions of *prototypicality* without modeling the interactions between the two words; that is, they guessed that (*animal, sofa*) is a positive example because *animal* looks like a hypernym (Levy et al., 2015).

Instead, the *H-Feature detector* model trains a linear classifier using concatenation, as described above, and then removes this prototypical information from the word vectors by projecting them on a hyperplane orthogonal to the separating hyperplane learned by the linear classifier. By repeating this process, one can learn multiple classifiers, each of which increases the models representational power. In each iteration i, four features are extracted to represent the word pair, based on the current representations of the word pair $(\vec{x}, \vec{y})$ and the hyperplane $\vec{p_i}$ learned in the current iteration :

1. The similarity between $\vec{x}$ and the hyperplane, $\vec{x}.\vec{p_i}$
2. The similarity between $\vec{y}$ and the hyperplane, $\vec{y}.\vec{p_i}$
3. The similarity between the two words, $\vec{x}.\vec{y}$
4. The similarity between the difference of the two words, and the hyperplane, $(\vec{y} - \vec{x}).\vec{p_i}$

Features 1 and 2 capture similarities like the one included in the concatenation classifier. The third feature aims to overcome the shortcomings of the concatenation model by directly modeling the similarities between the two target words. Finally, the fourth feature captures the distributional inclusion hypothesis (Geffet and Dagan, 2005) – if word v is a hypernym of u, then the set of features of u are included in the set of features of v – by intuitively capturing whether y includes x (Roller et al., 2014).

5 Experimental Set-up

Tasks In addition to WHiC, we evaluate our context-aware representations on CONTEXT-PPDB. As mentioned in Section 2.3, CONTEXT-PPDB is a dataset for fine-grained lexical inference in context that captures other semantic relations beyond hypernymy. It has been created using 375 word pairs from a subset of the English Paraphrase Database (Ganitkevitch et al., 2013; Pavlick et al., 2015). These word pairs are semi-automatically labeled with semantic relations out-of-context. Shwartz and Dagan (2016) augmented them with examples of word usage in context, and re-annotated the word pairs given the extra contextual information. The final dataset consists of 3750 words/contexts tuples with a corresponding semantic label, one of which is entailment.

All our experiments are with the default train/dev/test splits on both datasets.

Contextualized Word Representations To obtain the Context2Vec representations, we use an existing 600-dimensional model trained on ukWaC (Ferraresi et al., 2006). We use 600 dimensional GloVe embeddings trained on the same corpus to create $\vec{w}_l$, $\vec{w}_r$, C_l, and C_r, and allow for a controlled comparison with Context2Vec. Context2Vec representations are significantly more expensive to train: Melamud et al. (2016) indicate that training requires ~30 hours on a Tesla K80 GPU, while the GloVe embeddings can be trained on the exact same amount of data in less than 7 hours on a CPU.

Supervised Lexical Entailment Classifier We use an SVM with an RBF kernel for WHiC and Logistic Regression for CONTEXT-PPDB as implemented in Scikit-Learn [3] as our classifiers, to allow for exact comparisons with past work on CONTEXT-PPDB. We use default parameters, except for adding class weights in the WHiC experiments to account for the unbalanced data. For WHiC we use features derived from the H-Feature

[3] http://scikit-learn.org

model described in Section 4. For CONTEXT-PPDB we simply concatenate the representations and use them directly as the features. We evaluate the predictions using F1 score.

6 Experiments on WHIC

In our first set of experiments, we evaluate the two models described in Section 3 on WHIC under a variety of combinations.

6.1 Overall Results

Results are summarized in Table 2. Supervised models[4] outperform the baseline that always predict that hypernymy holds ("All True Baseline") by up to 16 F-score points. Context-aware models outperform context-agnostic models by up to 3 points[5]. GloVe and Context2Vec models yield similar F1, both when used as word type representations alone, and when combined with masked representations. However, GloVe and Context2Vec representations capture complementary information: GloVe yields slightly better precision while Context2Vec models yield significantly better recall. The best performance overall is obtained by a hybrid model that uses word-type representations from Context2Vec and masked context-aware representations derived from GloVe.

Additionally using Context2Vec vectors directly ($\vec{c}_{l,c2v}$,$\vec{c}_{r,c2v}$) performs much worse than using them as masks ($\vec{w}_{l,c2v,mask}$,$\vec{c}_{r,c2v,mask}$). This highlights the benefit of using context to influence the word type representation rather than to directly compare word and context representations.

Finally, there is no benefit in using the context-aware masked representations without the word type representations: using just the masked representations by themselves does worse than using them in combination with the word type representations.

Overall, the scores in Table 2 highlight the challenging nature of WHIC, and leave scope for improvement with potentially better models for context-aware representations.

Supervised Model Config.				
Word-type	Context-aware	P	R	F
GloVe	None	44	60	51
GloVe	GloVe Masks	42	73	53
None	GloVe Masks	32	64	43
C2V	None	40	73	52
C2V	C2V Masks	41	73	52
None	C2V Masks	30	94	45
C2V	C2V Contexts	23	10	14
None	C2V Contexts	8	2	3
C2V Words	GloVe Masks	41	**78**	**54**
GloVe Words	C2V Masks	**44**	64	52
All True Baseline		24	100	38

Table 2: Results on WHIC. a) Word type indicates (GloVe or Context2Vec (C2V)) H-Features extracted from context-agnostic representations. b) Context aware indicates H-Features extracted from the context-aware representations described in Section 3.

6.2 Sensitivity to context

To determine the sensitivity of our models to context changes, we evaluate on the balanced subset of WHIC comprised of positive examples and negative examples created by permuting contexts in Step 3 of the dataset creation process. We analyze the predictions using a modified version of precision, recall and F-score, defined as the precision, recall, and F1-score calculated over each (w_l,w_r) word pair, and then averaged over all word pairs. We call these measures the Macro-P/R/F1.

Table 3 shows that context-aware representations generally improve performance on all three metrics, but the gain is larger on recall. Again we observe that models using Context2Vec word types and masks have a better Macro-R than the corresponding GloVe models. Overall, the masked representations obtained from Context2Vec perform the best on these metrics, closely followed by the overall best model that uses the Context2Vec word type representations and the masked representations from GloVe.

Finally, note that the all-true baseline surprisingly does as well as the best context-aware model on this metric. However, it cannot detect the direction of hypernymy (Section 6.3), and the structure of WHIC allows us to distinguish these two factors.

[4] We also tried two unsupervised context-agnostic baselines using cosine similarity and balAPinc (Kotlerman et al., 2010) but they trivially predicted all pairs as entailing

[5] A statistically significant difference with $p < 0.01$ under the McNemar's test (McNemar, 1947)

Supervised Model Config.		Context sensitivity			Directionality
Word Type rep.	Context-aware rep.	Macro-P	Macro-R	Macro-F	Pairwise Acc.
GloVe	None	13	28	17	59
GloVe	GloVe Masks	17	35	22	71
None	GloVe Masks	13	30	18	59
C2V	None	15	35	21	71
C2V	C2V Masks	16	35	21	72
None	C2V Masks	**18**	**45**	**25**	62
C2V	C2V Contexts	5	5	4	9
None	C2V Contexts	1	1	1	1
C2V	GloVe Masks	17	37	23	**76**
GloVe	C2V Masks	14	29	19	63
All True Baseline		18	50	25	0

Table 3: Macro-P/R/F1 and Pairwise accuracy, are intended to capture context-awareness (Section 6.2) and directionality-discrimination abilities (Section 6.3) of the models, respectively.

6.3 Sensitivity to Entailment Direction

Next, we evaluate to what extent the models capture the direction of hypernymy using the balanced subset of WHIC that consists of all positive examples and flipped negative examples generated in Step 4 in the dataset creation process. We measure directionality by looking at the fraction of pairs $((w_l, w_r, c_l, c_r), (w_r, w_l, c_r, c_l))$ where both examples are correctly labeled, i.e. the former is labeled as $\implies$ and the latter as $\not\implies$. We call this metric the pairwise accuracy.

As seen in Table 3, the best pairwise accuracy is again obtained by the hybrid model using word type representations from Context2Vec and the masked representations from GloVe. Overall Context2Vec models do a better job at capturing directionality than GloVe.

6.4 Nature of Contextualized Masks

We also hypothesized that masked contextualized representations based on the full volume of the context using min and max operations (Section 3.1) better capture salient context dimensions than the more usual vector averaging approach. We test this hypothesis empirically by replacing masked word-in-context representations $\vec{w}_{l,mask}$ and $\vec{w}_{r,mask}$ by two other ways to capture context. In the first method, we use the mean of the contexts $(\vec{c}_{l,mean}, \vec{c}_{r,mean})$. In the second method, we use $(\vec{w}_{l,mean}, \vec{w}_{r,mean})$, i.e. the masked representations calculated by using only the mean of the context, and not the max and min.

Table 4 shows that our preferred method outperforms the two alternatives on WHIC, with our proposed representations outperforming the other methods by 3 F1 points. Additionally, this increase in performance also comes with significant improvement in detection of asymmetric relations.

6.5 Summary

Overall, both Context2Vec and Glove representations improve performance over context-agnostic baselines. Using masking to contextualize word type representations works better than just using the context representations as is. The best performing model is a hybrid model that uses word type representations from Context2Vec and masked representations from GloVe. Analysis enabled by the structure of the dataset shows that all masked representations are sensitive to changes in meaning indicated by glosses from distinct WordNet synsets. However, the more expensive Context2Vec representations do a better job at recall and direction of hypernymy.

7 CONTEXT-PPDB

We now experiment on CONTEXT-PPDB to test the ability of contextualized representations to capture semantic relations beyond hypernymy, to aid future work on recognizing other contextualized relationships.

Shwartz and Dagan (2016) establish a baseline of 67 F1 on this dataset using rich features characterizing word pairs drawn from PPDB as

Dataset	Representations	P	R	F	Context sensitivity	Directionality
WHIC	$\vec{c}_{l,mean}$,$\vec{c}_{r,mean}$	**45**	59	51	17	58
	$\vec{w}_{l,mean}$,$\vec{w}_{r,mean}$	43	62	51	18	61
	$\vec{w}_{l,mask}$,$\vec{w}_{r,mask}$	42	**73**	**53**	**22**	**71**

Table 4: Impact of masks on WHIC measured by Precision (P), Recall (R), F-Measure (F), context sensitivity (Macro-F1) and directionality (Pairwise accuracy). Replacing our contextualized representations by a mean representation of the context, or a contextualized representation based only on the mean, leads to drops in performance.

Word Type	P	R	F
Baseline	68	70	67
++ context-aware rep.s	72	72	**72**

Table 5: Results on CONTEXT-PPDB. Baseline indicates the previous state of the art result on this dataset (Shwartz and Dagan, 2016)

well as similarity scores between words and contexts. The PPDB features notably include scores for likelihood of context-agnostic entailment labels, distributional similarities, and probabilities of the word pair being paraphrases, among other scores. Additionally, word representation features are used: given two word/context pairs (w_x, c_x, w_y, c_y), GloVe vectors are used to represent w_x and w_y, as well as words in c_x and c_y, and are used to extract the following feature, which capture the most salient word/context similarities between the two pairs :

$$\{\max_{w\in c_y} \vec{w_x}\cdot\vec{w}, \max_{w\in c_x} \vec{w_y}\cdot\vec{w}, \max_{w\in c_x, w'\in c_y} \vec{w}\cdot\vec{w'}\}$$

We augment this system with contextualized word representations. We use the GloVe based masked representations, as they can be obtained with a negligible computation cost in addition the features already included in the baseline, and as the labels denote a mix of directional and non-directional relations. This remarkably yields an improvement ~5 F1 points compared to the previous state-of-the-art (Table 5). Breaking down results per label (Table 6) shows an increase of 8 F1 points for the entailment class. This improvement again stems from a large increase in recall, mirroring the behavior observed on WHIC. The diverse "other-related" category also benefits from context-aware representations.

Label	Baseline	++ Context-aware Rep.s
Equivalence	76	76
Entailment	79	**87**
Alternation	55	55
Other-related	12	**28**
Independent	77	**78**

Table 6: Performance of the baseline and augmented model on all semantic relations in CONTEXT-PPDB measured using per-class F1

8 Related Work

WordNet and lexical entailment The "is-a" hierarchy of WordNet (Fellbaum, 1998) is a prominent source of information for unsupervised detection of hypernymy and entailment (Harabagiu and Moldovan, 1998; Shwartz et al., 2015), as well as a source of various datasets (Baroni and Lenci, 2011; Baroni et al., 2012). WHIC is inspired by the latter line of work, except that we extract exemplar contexts from WordNet in addition to relations between words.

Modeling word meaning in context Prior models for the meaning of a word in a given context aimed to capture semantic equivalence in tasks such as lexical substitution, word sense disambiguation or paraphrase ranking, rather than asymmetric relations such as entailment. One line of work (Dinu and Lapata, 2010; Reisinger and Mooney, 2010) views each word as a set of latent word senses. These models rely on token representations for individual occurrences of a word and then choose a set of token vectors based on the current context. An alternate set of models (Erk and Padó, 2008; Thater et al., 2011; Dinu et al., 2012) avoids defining a fixed set of word senses, and instead contextualizes word type vectors as we do here. These models share the idea of using an element-wise multiplication to apply a context mask to word type representations. The nature of the context representation varies: Erk and Padó (2008) use inverse selectional preferences; Thater et al. (2010) combine a first order co-occurrence based representation for the context with a second order representation for the target, Thater et al. (2011) rely on syntactic dependencies to define context. Apidianaki (2016) shows that bag-of-word context representation within a small context window works as well as syntactic definitions of context for ranking paraphrases in context.

Our use of convolution is motivated by success of similar models on sentence classification tasks. Tang et al. (2014) uses convolution over embedding matrices for unigrams, bigrams, and trigrams, while Hovy (2015) uses just unigrams. However, all these works use the resulting representations to predict properties of the sentence (e.g., sentiment), rather than to contextualize target word representations.

In-context lexical semantic tasks Besides entailment, other lexical semantic tasks studied in context include lexical substitution (McCarthy and Navigli, 2007) and cross-lingual lexical substitution (Mihalcea et al., 2010). The focus of these tasks and their related datasets is on synonymy and translation equivalence, since they require one to predict substitutes for a target word instance, which preserve its meaning in a given sentential context. On the other hand, the focus of this work and WHIC is on detecting more fine-grained relations via lexical entailment. Another related task is that of paraphrase ranking (Apidianaki, 2016). The work by Apidianaki (2016) is also notable because of their successful use of models of word-meaning in context from Thater et al. (2011), which is closely related to our work.

9 Conclusion

We introduced WHIC, a dataset to evaluate lexical entailment in context, providing exemplar sentences to ground the meaning of words being considered for entailment, and challenging examples designed to capture entailment direction accurately.

We showed that supervised models developed for context-agnostic lexical entailment can address the context-aware task to some extent, when replacing word representations with a contextualized version. We compared two contextualized representations including (1) a simple context-aware representation based on the geometry of word embeddings, and (2) Context2Vec, a more expensive BiLSTM-based model that yields representations of words and their context in the same space. Both improve performance over context-agnostic models, and have complementary properties: models using Context2Vec are more accurate at discriminating the direction of entailment. They also have a better recall when measured using metrics designed to test sensitivity to context. Finally, we also showed that contextualized representations can improve detection of other semantic relations in context.

While encouraging, the performance of models considered leave substantial room for improvement. For instance, it remains to be seen whether richer features for the supervised models and richer context representations can improve sensitivity to context, and whether the nuances of the task can be better captured with annotations on a graded scale, following previous work on word meaning in context (Erk et al., 2013).

Acknowledgements

The authors thank the anonymous reviewers for their comments, as well as the members of the CLIP lab at UMD and Mona Diab for many conversations which helped shape this paper. We also thank Vered Shwartz for help with data and code for CONTEXT-PPDB, and Stephen Roller for help with the H-Feature detector code.

References

Marianna Apidianaki. 2016. Vector-space models for PPDB paraphrase ranking in context. In *Proceedings of EMNLP 2016*. Austin, TX, USA, pages 2028–2034.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of EACL 2012*. pages 23–32. http://dl.acm.org/citation.cfm?id=2380822.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*. pages 1–10. http://dl.acm.org/citation.cfm?id=2140490.2140491.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the First*

International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment. Springer-Verlag, Southampton, UK, MLCW'05, pages 177–190.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Morgan & Claypool Publishers.

Georgiana Dinu and Mirella Lapata. 2010. Measuring Distributional Similarity in Context. In *Proceedings of EMNLP 2010*. Cambridge, MA, USA, pages 1162–1172. http://eprints.pascal-network.org/archive/00008156/.

Georgiana Dinu, Stefan Thater, and Sören Laue. 2012. A comparison of models of word meaning in context. In *Proceedings of NAACL-HLT 2012*. pages 611–615.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2013. Measuring Word Meaning in Context. *Computational Linguistics* 39(3).

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of EMNLP 2010*. Honolulu, HA, USA, October, pages 897–906. https://doi.org/10.3115/1613715.1613831.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2006. Introducing and evaluating ukWaC , a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop*.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB : The Paraphrase Database. *Proceedings of NAACL-HLT 2013* (June):758—-764. http://cs.jhu.edu/ ccb/publications/ppdb.pdf.

Maayan Geffet and Ido Dagan. 2005. The Distributional Inclusion Hypotheses and Lexical Entailment. In *Proceedings of ACL 2005*. Ann Arbor, MI, June, pages 107–114.

Sanda Harabagiu and Andrew Hickl. 2006. Methods for Using Textual Entailment in Open-Domain Question Answering. *Proceedings of ACL* (July):905–912. https://doi.org/10.3115/1220175.1220289.

Sanda Harabagiu and Dan Moldovan. 1998. Knowledge processing on an extended wordnet. *WordNet: An electronic lexical database* 305:381–405.

Dirk Hovy. 2015. Demographic Factors Improve Classification Performance. In *Proceedings of ACL-IJCNLP 2015*. Beijing, China, pages 752–762.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional Distributional Similarity for Lexical Inference. *Natural Language Engineering* 16(4):359–389. https://doi.org/10.1017/S1351324910000124.

German Kruszewski and Marco Baroni. 2015. Deriving Boolean structures from distributional vectors. *Transactions of ACL* 3:375–388.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do Supervised Distributional Methods Really Learn Lexical Inference Relations? In *NAACL HLT 2015*. pages 970–976.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of SEMEVAL 2007*. pages 48–53. https://doi.org/10.1007/s10579-009-9084-1.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* 12(2):153–157.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In *Proceedings of CoNLL 2016*. Berlin, Germany, pages 51–61.

Rada Mihalcea, Ravi Sinha, and Diana McCarthy. 2010. SemEval-2010 Task 2: Cross-Lingual Lexical Substitution. In *Proceedings of SemEval 2010 (ACL 2010)*. Uppsala, Sweden, July, pages 9–14.

Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. *Proc. of SemEval-2015* .

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. *Proceedings of ACL-IJCNLP 2015* pages 425–430.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of EMNLP 2014*. Doha, Qatar, pages 1532–1543. https://doi.org/10.3115/v1/D14-1162.

Joseph Reisinger and Raymond J Mooney. 2010. Multi-Prototype Vector-Space Models of Word Meaning. In *Proceedings of NAACL 2010*. Los Angeles, CA, June, pages 109–117.

Stephen Roller and Katrin Erk. 2016. Relations such as Hypernymy: Identifying and Exploiting Hearst Patterns in Distributional Vectors for Lexical Entailment. *Proceedings of EMNLP 2016* http://arxiv.org/abs/1605.05433.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet Selective: Supervised Distributional Hypernymy Detection. *Proceedings of COLING 2014* pages 1025–1036.

Vered Shwartz and Ido Dagan. 2016. Adding Context to Semantic Data-Driven Paraphrasing. In *Proceedings of *SEM 2016*. Berlin, Germany, pages 108–113.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving Hypernymy Detection with an Integrated Pattern-based and Distributional Method. In *Proceedings of ACL 2016*.

Vered Shwartz, Omer Levy, Ido Dagan, and Jacob Goldberger. 2015. Learning to Exploit Structured Resources for Lexical Inference. In *Proceedings of CoNLL 2015*. Beijing, China, pages 175–184. http://www.aclweb.org/anthology/K15-1018.

Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2017. Hypernyms under Siege: Linguistically-motivated Artillery for Hypernymy Detection. In *Proceedings of EACL 2017*. Valencia, Spain.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning Sentiment-Specific Word Embedding. In *Proceedings of ACL 2014*. Baltimore, MD, USA, pages 1555–1565. https://doi.org/10.3115/1220575.1220648.

Stefan Thater, Hagen Fuerstenau, and Manfred Pinkal. 2010. Contextualizing Semantic Representations Using Syntactically Enriched Vector Models. In *Proceedings of ACL 2010*. Uppsala, Sweden, July, pages 948–957. http://eprints.pascal-network.org/archive/00008090/.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2011. Word Meaning in Context : A Simple and Effective Vector Model. In *Proceedings of IJCNLP 2011*. Chiang Mai, Thailand, pages 1134–1143.

Peter Turney and Saif Mohammad. 2013. Experiments with three approaches to recognizing lexical entailment. *Natural Language Engineering* 1(1):1–42. https://doi.org/10.1017/S1351324913000387.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and Took, Gaggle and Goose, Book and Read: Evaluating the Utility of Vector Differences for Lexical Relation Learning. In *Proceedings of ACL 2016*. Berlin, Germany, pages 1671–1682.

Maayan Zhitomirsky-Geffet and Ido Dagan. 2009. Bootstrapping Distributional Feature Vector Quality. *Computational Linguistics* (November 2008).