Paraphrase-Sense-Tagged Sentences

Anne Cocos and Chris Callison-Burch
Department of Computer and Information Science
University of Pennsylvania
{acocos|ccb}@seas.upenn.edu

Abstract

Many natural language processing tasks require discriminating the particular meaning of a word in context, but building corpora for developing sense-aware models can be a challenge. We present a large resource of example usages for words having a particular meaning, called Paraphrase-Sense-Tagged Sentences (PSTS). Built upon the premise that a word's paraphrases instantiate its fine-grained meanings – i.e. *bug* has different meanings corresponding to its paraphrases *fly* and *microbe* – the resource contains up to 10,000 sentences for each of 3 million target-paraphrase pairs where the target word takes on the meaning of the paraphrase. We describe an automatic method based on bilingual pivoting used to enumerate sentences for PSTS, and present two models for ranking PSTS sentences based on their quality. Finally, we demonstrate the utility of PSTS by using it to build a dataset for the task of hypernym prediction in context. Training a model on this automatically-generated dataset produces accuracy that is competitive with a model trained on smaller datasets crafted with some manual effort.

1 Introduction

Word meaning is context-dependent. While lexical semantic tasks like relation prediction have been studied extensively in a non-contextual setting, applying such models to a downstream task like textual inference or question answering requires taking the full context into account. For example, it may be true that *rotavirus* is a type of *bug*, but *rotavirus* is not within the realm of possible answers to the question *"Which **bug** caused the server outage?"*

Many tasks in natural language processing require discerning the meaning of polysemous words within a particular context. It can be a challenge to develop corpora for training or evaluating sense-aware models, since particular attention must be paid to making sure the distribution of instances for a given word reflects its various meanings. This paper introduces Paraphrase-Sense-Tagged Sentences (PSTS)[1], a large resource of example usages of English words having a particular meaning. Rather than assume a rigid inventory of possible senses for each word, PSTS is grounded in the idea that the many fine-grained meanings of a word are instantiated by its paraphrases. For example, the word *bug* has different meanings corresponding to its paraphrases *fly*, *error*, and *microbe*, and PSTS includes sentences where *bug* takes on each of these meanings (Figure 1). Overall, the resource contains up to 10,000 sentences for each of roughly 3 million English lexical and phrasal paraphrases from the Paraphrase Database (PPDB) (Bannard and Callison-Burch, 2005; Ganitkevitch et al., 2013; Pavlick et al., 2015).

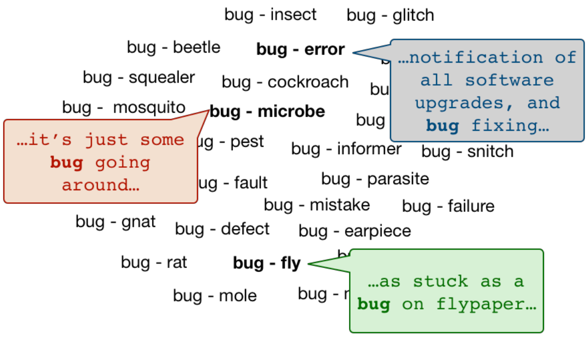


Figure 1: We assume that the fine-grained meanings of the noun *bug* are instantiated by its paraphrases. Example usages of *bug* pertaining to each paraphrase are extracted automatically via a method inspired by bilingual pivoting (Bannard and Callison-Burch, 2005).

PSTS was compiled by automatically extracting

[1] http://psts.io

sentences from the English side of bilingual parallel corpora using a technique inspired by bilingual pivoting (Bannard and Callison-Burch, 2005). For instance, to find a sentence containing *bug* where it means *fly*, we select English sentences where *bug* is translated to the French *mouche*, Spanish *mosca*, or one of the other foreign words that *bug* shares as a translation with *fly*. Qualitative analysis of the sentences in PSTS indicates that this is a noisy process, so we implement and compare two methods for ranking sentences by the degree to which they are 'characteristic' of their associated paraphrase meaning. When used to rank PSTS sentences, a supervised regression model trained to correlate with human judgments of sentence quality, and an unsupervised lexical substitution model (Melamud et al., 2016) lead to respectively 89% and 96% precision within the top-10 sentences.

In Section 5 we demonstrate a use of PSTS by automatically constructing a training set for the task of hypernym prediction in context (Shwartz and Dagan, 2016; Vyas and Carpuat, 2017). In this task, a system is presented with a pair of words and sentence-level contexts for each, and must predict whether a hypernym relation holds for that word pair in the given contexts. We automatically generate training data for this task from PSTS, creating a training set with 5 and 30 times more training instances than the two existing datasets for this task – both of which rely on manually-generated resources. We train a contextual hypernym prediction model on the PSTS-derived dataset, and show that it leads to prediction accuracy that is competitive with or better than than the same model trained on the smaller training sets.

2 Related Work

In general, there are three basic categories of techniques for generating sense-tagged corpora: manual annotation, application of supervised models for word sense disambiguation, and unsupervised methods. Manual annotation asks humans to hand-label word instances with a sense tag, assuming that the word's senses are enumerated in an underlying sense inventory (typically WordNet (Miller, 1995)) (Edmonds and Cotton, 2001; Mihalcea et al., 2004; Petrolito and Bond, 2014). Manually sense-tagged corpora, such as SemCor (Miller et al., 1994) or OntoNotes (Weischedel et al., 2013), can then be used to train supervised word sense disambiguation (WSD) classifiers to predict sense labels on untagged text (Ando, 2006; Zhong and Ng, 2010; Rothe and Schütze, 2015). Top-performing supervised WSD systems achieve roughly 74% accuracy in assigning WordNet sense labels to word instances (Ando, 2006; Rothe and Schütze, 2015). In shared task settings, supervised classifiers typically out-perform unsupervised WSD systems (Mihalcea et al., 2004).

Within the set of unsupervised methods, one long-standing idea is to use foreign translations as proxies for sense labels of polysemous words (Brown et al., 1991; Dagan, 1991). This is based on the assumption that a polysemous English word e will often have different translations into a target language, depending on the sense of e that is used. To borrow an example from Gale et al. (1992), if the English word *sentence* is translated to the French *peine* (judicial sentence) in one context and the French *phrase* (syntactic sentence) in another, then the two instances in English can be tagged with appropriate sense labels based on a mapping from the French translations to the English sense inventory. This technique has been frequently applied to automatically generate sense-tagged corpora, in order to overcome the costliness of manual sense annotation (Gale et al., 1992; Dagan and Itai, 1994; Diab and Resnik, 2002; Ng et al., 2003; Chan and Ng, 2005; Apidianaki, 2009; Lefever et al., 2011). Our approach to unsupervised sense tagging in this paper is related, but different. Like the translation proxy approach, our method relies on having bilingual parallel corpora. But in our case, the sense labels are grounded in English paraphrases, rather than in foreign translations. This means that our method does not require any manual mapping from foreign translations to an English sense inventory. It also enables us to generate sense-tagged examples using bitext over multiple pivot languages, without having to resolve sense mapping between languages.

There is a close relationship between sense tagging and paraphrasing. Some research efforts assume that words have a discrete sense inventory, and they represent each word sense as a set or cluster of paraphrases (Miller, 1995; Cocos and Callison-Burch, 2016). Other work (Melamud et al., 2015a), including in lexical substitution (McCarthy and Navigli, 2007, 2009), represents the contextualized meaning of a word instance by the set of paraphrases that could be substituted for it. This paper takes the view that assuming a discrete underlying sense inventory can be too rigid for many applications; humans have notoriously low agreement in manual sense-tagging tasks (Cinková et al., 2012), and the appropriate sense granularity varies by setting. Instead, we assume a "one paraphrase per fine-grained meaning" model in this paper as a generalizable approach to word sense modeling. In PSTS, a word type has as many meanings as it has paraphrases, but its paraphrase-sense-tagged instances can be grouped based on a coarser sense inventory if so desired.

3 Constructing PSTS

For a paraphrase pair like *coach*↔*trainer*, PSTS includes a set of sentences $S^{\overline{coach},trainer}$ containing *coach* in its *trainer* sense (e.g. *My* ***coach*** *cancelled the workout*), and a set of sentences $S^{coach,\overline{trainer}}$ containing *trainer* in its *coach* sense (e.g. *It's just a sprain, according to her* ***trainer***). This section describes the method for enumerating sentences corresponding to a particular paraphrase pair for inclusion in PSTS.

3.1 Sentence Extraction

Our method for extracting sentences for PSTS is inspired by bilingual pivoting (Bannard and Callison-Burch, 2005), which discovers same-language paraphrases by 'pivoting' over bilingual parallel corpora. Specifically, if the English phrases *coach* and *trainer* are each translated to the same Slovenian phrase *trener* in some contexts, this is taken as evidence that *coach* and *trainer* have approximately similar meaning. We apply this idea in reverse: to find English sentences where *coach* means *trainer* (as opposed to *bus* or *railcar*), we extract sentences from English-Slovenian parallel corpora where *coach* has been aligned to their shared translation *trener*.

The starting point for extracting PSTS is the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013; Pavlick et al., 2015), a collection of over 80M lexical (one-word) and phrasal English paraphrase pairs.[2] Because PPDB was built using the pivot method, it follows that each paraphrase pair *x*↔*y* in PPDB has at least one shared foreign translation. The paraphrases for a target word x are used as proxy labels for x's fine-grained senses.

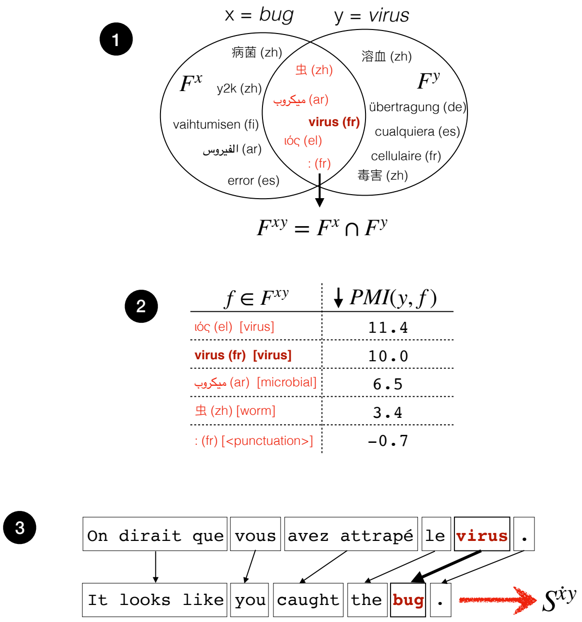


Figure 2: Extracting sentences containing the noun $x = bug$ in its $y = virus$ sense for PSTS set $S^{\overline{x}y}$. In Step (1), the set F^{xy} of translations shared by *bug* and *virus* is enumerated. In Step (2), the translations $f \in F^{xy}$ are ranked by $PMI(y, f)$, in order to prioritize *bug*'s translations most 'characteristic' of its meaning in the *virus* sense. In Step (3), sentences where *bug* has been aligned to the French translation $f = virus$ are extracted from bitext corpora and added to the set $S^{\overline{x}y}$.

The process for extracting PSTS sentences $S^{\overline{x},y}$ for *x*↔*y* consists of three steps: (1) finding a set F^{xy} of shared translations for *x* and *y*, (2) prioritizing translations that are most 'characteristic' of *x*'s shared meaning with *y*, and (3) extracting sentences from bilingual parallel corpora. The process is illustrated in Figure 2, and described in further detail below.

Step 1: Finding Shared Translations. In order to find sentences containing the English term x where it takes on its meaning as a paraphrase of y, we begin by finding the sets of foreign translations for x and y, F^x and F^y respectively. These translations are enumerated by processing the phrase-based alignments induced between English sentences and their translations within a large, amalgamated set of English-to-foreign bitext corpora. Once the translation sets F^x and F^y are extracted for the individual terms, we take their intersection

[2] Note that while the term *paraphrase* is generally used to denote different words or phrases with approximately the same meaning, the noisy bilingual pivoting process can produce paraphrase pairs that are more loosely semantically related (i.e. meronyms, holonyms, or even antonyms). Here we take a broader definition of *paraphrase* to mean any pair derived from bilingual pivoting.

$(x \leftrightarrow y)$	f	$\log p(f\|y)$	$\log p(f)$	PMI(y,f)	Sentence segment
hot ↔ warm	cálida (es)	-1.96	-12.75	10.79	With the end of the hot season last year, ...
	ciepłego (pl)	-3.92	-14.34	10.42	I think that a hot cup of milk...would be welcome.
	chaudes (fr)	-3.30	-12.63	9.33	Avoid getting your feet too close to hot surfaces...
hot ↔ spicy	吃辛辣 (zh)	-4.41	-17.75	13.34	People with digestion issues should shun hot dishes.
	épicé (fr)	-1.61	-14.32	12.72	Hot jambalaya!
	辣 (zh)	-1.92	-12.98	11.06	...a manufacturer of soy sauce, hot pepper paste...
hot ↔ popular	très vogue (fr)	-8.19	-17.40	9.21	...skin aging - a hot topic in the cosmetic industry.
	très demande (fr)	-9.11	-17.47	8.36	This area of technology is hot.
	热门 (zh)	-3.61	-11.77	8.17	Now the town is a hot spot for weekend outings.

Table 1: Example PSTS sentence segments for the adjective x=*hot* as a paraphrase of $y \in \{\text{warm}, \text{spicy}, \text{popular}\}$. For each example, the pivot translation f is given along with its translation probability $p(f|y)$, foreign word probability $p(f)$, and PMI(y, f).

as the set of shared translations, F^{xy}.

Step 2: Prioritizing Characteristic Translations. Our goal is to build $S^{\overline{x}y}$ such that its sentences containing x are "highly characteristic" of x's shared meaning with y, and vice versa. However, not all pivot translations $f \in F^{xy}$ produce equally characteristic sentences. For example, consider the paraphrase pair *bug* ↔ *worm*. Their shared translation set, $F^{bug,worm}$, includes the French terms *ver* (*worm*) and *espèce* (*species*), and the Chinese term 虫 (*bug*). In selecting sentences for $S^{\overline{bug},worm}$, PSTS should prioritize English sentences where *bug* has been translated to the most characteristic translation for *worm* – *ver* – over the more general 虫 or *espèce*.

We propose using pointwise mutual information (PMI) as a measure to quantify the degree to which a foreign translation is "characteristic" of an English term. To avoid unwanted biases that might arise from the uneven distribution of languages present in our bitext corpora, we treat PMI as language-specific and use shorthand notation f_l to indicate that f comes from language l. The PMI of English term e with foreign word f_l can be computed based on the statistics of their alignment in bitext corpora:

$$\text{PMI}(e, f_l) = \frac{p(e, f_l)}{p(e) \cdot p(f_l)} = \frac{p(f_l|e)}{p(f_l)} \qquad (1)$$

The term in the numerator of the rightmost expression is the translation probability $p(f_l|e)$, which indicates the likelihood that English word e is aligned to foreign term f_l in an English-l parallel corpus. Maximizing this term promotes the most frequent foreign translations for e. The term in the denominator is the likelihood of the foreign word, $p(f_l)$. Dividing by this term downweights the emphasis on frequent foreign words. This is especially helpful for mitigating errors due to mis-alignments of English words with foreign stop words or punctuation. Both $p(f_l|e)$ and $p(f_l)$ are estimated using maximum likelihood estimates from an automatically aligned English-l parallel corpus.

Step 3: Extracting Sentences. To extract $S^{\overline{x}y}$, we first order the shared translations for paraphrase pair x↔y, $f \in F^{xy}$, by decreasing $PMI(y, f)$. Then, for each translation f in order, we extract up to 2500 sentences from the bitext corpora where x is translated to f. This process continues until $S^{\overline{x}y}$ reaches a maximum size of 10k sentences. Table 1 gives examples of sentences extracted for various paraphrases of the adjective *hot*, ordered by decreasing PMI.

PSTS is extracted from the same English-to-foreign bitext corpora used to generate English PPDB (Ganitkevitch et al., 2013), consisting of over 106 million sentence pairs, and spanning 22 pivot languages. Sentences are extracted for all paraphrases with a minimum PPDBSCORE[3] threshold of at least 2.0. The threshold value serves to produce a resource corresponding to the highest-quality paraphrases in PPDB, and eliminates considerable noise. In total, sentences were extracted for over 3.3M paraphrase pairs covering nouns, verbs, adverbs, and adjectives (21 part-of-speech tags total). Table 2 gives the total number of paraphrase pairs covered and average number of sentences per pair in each direction. Results are

[3] The PPDBSCORE is a supervised metric trained to correlate with human judgments of paraphrase quality (Pavlick et al., 2015).

POS	Paraphrase pairs	Mean $\lvert S^{\overline{xy}}\rvert$	Median $\lvert S^{\overline{xy}}\rvert$
N*	1.8M	856	75
V*	1.1M	972	54
R*	0.1M	1385	115
J*	0.3M	972	72
Total	3.3M	918	68

Table 2: Number of paraphrase pairs and sentences in PSTS by macro-level part of speech (POS). The number of sentences per pair is capped at 10k in each direction.

given by macro-level part-of-speech, where, for example, N* covers part-of-speech tags NN, NNS, NNP, and NNPS, and phrasal constituent tag NP.

4 PSTS Validation and Ranking

Bilingual pivoting is a noisy process (Bannard and Callison-Burch, 2005; Chan et al., 2011; Pavlick et al., 2015). Although shared translations for each paraphrase pair were carefully selected using PMI in an attempt to mitigate noise in PSTS, the analysis of PSTS sentences that follows in this section indicates that their quality varies. Therefore, we follow the qualitative analysis by proposing and evaluating two metrics for ranking target word instances to promote those most characteristic of the associated paraphrase meaning.

4.1 Qualitative Evaluation of PSTS

Our primary question is whether automatically-extracted PSTS sentences for a paraphrase pair truly reflect the paraphrase meaning. Specifically, for sentences like s_{bug} where $s_{bug} \in S^{\overline{bug,virus}}$, does the meaning of the word *bug* in s_{bug} actually reflect its shared meaning with *virus*?

We used human judgments to investigate this question. For a pair like *bug*↔*insect*, annotators were presented with a sentence containing *bug* from $S^{\overline{bug,insect}}$, and asked whether *bug* means roughly the same thing as *insect* in the sentence. The annotators chose from responses *yes* (the meanings are roughly similar), *no* (the meanings are different), *unclear* (there is not enough contextual information to tell), or *never* (these phrases never have similar meaning). We instructed annotators to ignore grammaticality in their responses, and concentrate specifically on the semantics of the paraphrase pair.

Human annotation was run in two rounds, with the first round of annotation completed by NLP researchers, and the second (much larger) round completed by crowd workers via Amazon Mechanical Turk (MTurk). In the first round (done by NLP researchers), a batch of 240 sentence-paraphrase instances (covering lexical and phrasal noun, verb, adjective, and adverb paraphrases) corresponding to 40 hand-selected polysemous target words was presented to a group of 10 annotators, split into 5 teams of 2. To encourage consistency, each pair of annotators worked together to annotate each instance. For redundancy, we also ensured that each instance was annotated separately by two pairs of researchers. In this first round, the annotators had inter-pair agreement of 0.41 Fleiss' kappa (after mapping all *never* and *unclear* answers to *no*), indicating weak agreement (Fleiss, 1971).

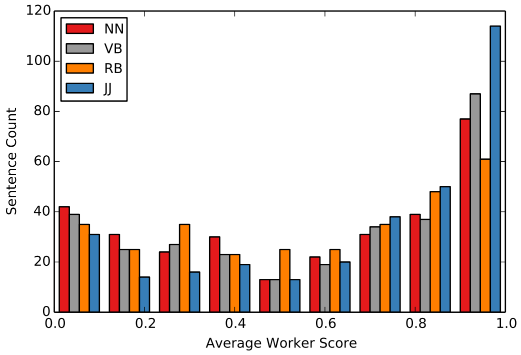
In the second round we generated 1000 sentence-paraphrase instances, and each instance was evaluated individually by 7 workers on MTurk. In each MTurk assignment, we also included an instance from the first round that was annotated as unanimously *yes* or unanimously *no* by the NLP researchers in order to gauge agreement between rounds. The crowd annotators had inter-annotator agreement of 0.34 Fleiss' kappa (after mapping all *never* and *unclear* answers to *no*) – slightly lower than that of the NLP researchers in round one. The crowd workers had 75% absolute agreement with the 'control' instances inserted from the previous round.

There was weak inter-annotator agreement in both annotation rounds. To determine why, we manually examined 100 randomly selected instances that received an even or nearly even split of *yes* and *no* responses. Most of the time (71%), annotators disagreed on the boundary between "roughly similar" and "different" meanings. For example, in *"An American can not rent a car in Canada,* ***drive*** *it to the USA and then return it to Canada."*, annotators were closely split on whether the target word *drive* had roughly similar meaning to its paraphrase *guide*. Another common reason for disagreement was ambiguity of the target word within the given context (13%), as in the instance *"I think some* ***bug*** *may have gotten in the clean room."* (paraphrase *virus*). Further disagreements occurred when the target word and paraphrase were morphologically different forms of the same lemma (6%) (*"...a matter which is very*

***close** to our hearts..."* with paraphrase *closely*). The remaining 10% of closely split instances are generally cases where annotators did not consider all possible senses of the target word and paraphrase. For example, in *"It does not look good for the intelligence **agency** chief"*, only 4 of 7 crowd workers said that *service* was an appropriate paraphrase for its synonym *agency*.

4.1.1 Human annotation results

To quantify the overall quality of sentences in PSTS, we calculate the average human rating for each annotated instance, where *no* (32.1% of all annotations), *never* (3.9%), and *unclear* (2.8%) answers are mapped to the value 0, and *yes* answers are mapped to the value 1. The combined results of this calculation from both rounds are given in Figure 3. Overall, the average rating is 0.61, indicating that more sentence-paraphrase instances from PSTS are judged by humans to have similar meaning than dissimilar meaning. In general, adjectives produce higher-quality PSTS sentences than the other parts of speech. For nouns and adjectives, phrasal paraphrase pairs are judged to have higher quality than lexical paraphrase pairs. For verbs and adverbs, the results are reversed.



(a) Histogram of sentence ratings by part of speech

POS	Lexical	Phrasal
NN	0.55	0.64
VB	0.63	0.49
JJ	0.68	0.73
RB	0.64	0.36
Combined	0.62	0.55

(b) Mean sentence ratings by paraphrase type

Figure 3: Human evaluation of the degree to which a PSTS sentence from $S^{\overline{x}y}$ containing term x reflects x's shared meaning with its paraphrase y (range 0 to 1; higher scores are better).

To understand why some sentences are of poor quality, we manually examine 100 randomly selected instances with average human rating below 0.3. On close inspection, we disagreed with the low rating for 25% of the sentences (which mirrors the finding of 75% absolute agreement between expert- and crowd-annotated control instances in the second round of annotation). In those cases, either the meaning of the target in context is a rare sense of the target or paraphrase (e.g. *"the appropriation is intended to **cover** expenses"* with paraphrase *capture*), or the target word is ambiguous in its context but could be construed to match the paraphrase meaning (e.g. *"We're going to **treat** you as a victim in the field."* with paraphrase *discuss*).

For the truly poor-quality sentences, in roughly one third of cases the suggested PPDB paraphrase for the target word is of poor quality due to misspellings (e.g. *manage*↔*mange*) or other noise in the bilingual pivoting process. One common source of noise was mis-tagging of the target word in context, leading to a suggested paraphrase pertaining to the wrong part of speech. For example, in the sentence *"Increase in volume was accompanied by a change to an ovaloid or **elongate** shape"*, the target *elongate*, which appears as an adjective, was mis-tagged as a verb, yielding the suggested but erroneous paraphrase *lie*.

The remaining poor-quality sentences (roughly 50 of the 100 examined) were cases where the target word simply did not take on its shared meaning with the suggested paraphrase. Most of these occurred due to polysemous foreign translations. For example, PSTS wrongly suggests the sentence *"...to become a part of Zimbabwe's **bright** and positive history"* as an example of *bright* taking on the meaning of *high-gloss*. This error happens because the shared Spanish translation, *brillante*, can be used with both the literal and figurative senses of *bright*, but *high-gloss* only matches the literal sense.

4.2 Sentence Quality Ranking

Given the amount of variation in PSTS sentence quality, it would be useful to have a numeric quality estimate. In the formation of PSTS (Section 3) we used point-wise mutual information $PMI(y, f)$ of the English paraphrase y with the shared foreign translation f to estimate how characteristic a sentence containing English target word x is of its shared sense with y. But the Spearman correlation between PMI and the average human ratings for the annotated sentence-paraphrase instances is 0.23 ($p < 0.01$), indicating only weak positive correlation. Therefore, in order to enable selection within PSTS of the most characteristic sentences for each paraphrase pair for downstream tasks, we propose and evaluate two models to rerank PSTS sentences in a way that better corresponds to human quality judgements.

4.2.1 Supervised Regression Model

The first ranking model is a supervised regression, trained to correlate with human quality judgments. Concretely, given a target word x, its paraphrase y, and a sentence $s_x \in S^{\overline{x},y}$, the model predicts a score whose magnitude indicates how characteristic s_x is of x's shared meaning with y. This task is formulated as ordinary least squares linear regression, where the dependent variable is the average human quality rating for a sentence-paraphrase instance, and the features are computed based on the input sentence and paraphrase pair. There are four groups, or types, of features used in the model that are computed for each paraphrase-sentence instance, ($x \leftrightarrow y$, $s_x \in S^{\overline{x},y}$):

PPDB Features. Seven features from PPDB 2.0 for paraphrase pair $x \leftrightarrow y$ are used as input to the model. These include the pair's PPDBSCORE, and translation and paraphrase probabilities.

Contextual Features. Three contextual features are designed to measure the distributional similarity between the target x and paraphrase y, as well as the substitutability of paraphrase y for the target x in the given sentence. They include the mean cosine similarity between word embeddings[4] for paraphrase y and tokens within a two-word context window of x in sentence s_x; the cosine similarity between context-masked embeddings for x and y in s_x (Vyas and Carpuat, 2017), and the AddCos lexical substitution metric where y is the substitute, x is the target, and the context is extracted from s_x (Melamud et al., 2015b) (Table 3).

Syntactic Features. Five binary features indicate the coarse part-of-speech label assigned to paraphrase $x \leftrightarrow y$ (NN, VB, RB, or JJ), and whether $x \leftrightarrow y$ is a lexical or phrasal paraphrase.

Mean contextual similarity $f(y, s_x) = \frac{\sum_{w \in W} cos(v_y, v_w)}{\lvert W \rvert}$
AddCos (Melamud et al., 2015b) $f(x, y, s_x) = \frac{\lvert W \rvert \cdot cos(v_x, v_y) + \sum_{w \in W} cos(v_y, v_w)}{2 \cdot \lvert W \rvert}$
Context-masked embedding similarity (Vyas and Carpuat, 2017) $f(x, y, s_x) = cos(v_{x,mask}, v_{y,mask})$ $v_{x,mask} = [v_x \odot v_{W_{min}}; v_x \odot v_{W_{max}}; v_x \odot v_{W_{mean}}]$

Table 3: Contextual features used for sentence quality prediction, given paraphrase pair $x \leftrightarrow y$ and sentence $s_x \in S^{\overline{x},y}$. W contains words within a two-token context window of x in s_x. v_x is the word embedding for x. $v_{W_\star}$ are vectors composed of the column-wise min/max/mean of embeddings for $w \in W$. The $\odot$ symbol denotes element-wise multiplication.

PMI. The final feature is simply $PMI(y, f)$.

The features used as input to the model training process are the sixteen listed above, as well as their interactions as modeled by degree-2 polynomial combinations (153 features total). During training and validation, we apply feature selection using recursive feature elimination in cross-validation (RFECV) (Guyon et al., 2002).

We train the model on the 1227 sentence-paraphrase instances that were annotated in one or both rounds of human evaluation, after ignoring instances marked as 'unclear' by two or more workers. The quality rating for each instance is taken as the average annotator score, where *no*, *never*, and *unclear* answers are mapped to the value 0, and *yes* answers are mapped to the value 1. We refer to the predicted quality scores produced by this model as the REG(ression) score.

4.2.2 Unsupervised LexSub Model

Lexical substitution (hereafter LexSub) is the task of identifying meaning-preserving substitutes for target words in context (McCarthy and Navigli, 2007, 2009). For example, valid substitutes for *bug* in *There are plenty of places to plant a **bug** in her office* might include *microphone* or *listening device* but not *glitch*. The tasks of sense tagging and LexSub are closely related, since valid substitutes for a polysemous word must adhere to the correct meaning in each instance. Indeed, early

[4] For computing all contextual features, we used 300-dimensional *skip-gram* embeddings (Mikolov et al., 2013) trained on the Annotated Gigaword corpus (Napoles et al., 2012)).

LexSub systems explicitly included sense disambiguation as part of their pipeline (McCarthy and Navigli, 2007), and later studies have shown that performing sense disambiguation can improve the results of LexSub models and vice versa (Cocos et al., 2017; Alagić et al., 2018).

We adopt an off-the-shelf LexSub model called CONTEXT2VEC (Melamud et al., 2016) as an unsupervised sentence ranking model. CONTEXT2VEC learns word and context embeddings using a bi-directional LSTM such that words and their appropriate contexts have high cosine similarity. In order to apply CONTEXT2VEC to ranking sentence-paraphrase instances, we calculate the cosine similarity between the paraphrase's CONTEXT2VEC word embedding and the context of the target word in the sentence, using a pre-trained model.[5] The resulting score is hereafter referred to as the C2V score.

4.2.3 Ranking Model Comparison

We compare the PSTS REG and C2V scoring models under two evaluation settings. First, we measure the correlation between predicted sentence scores under each model, and the average human rating for annotated sentences. Second, we compare the precision of the top-10 ranked sentences under each model based on human judgments. In the latter experiment, we also compare with a baseline LexSub-based sentence selection and ranking model in order to validate bilingual pivoting as a worthwhile sentence selection approach.

To calculate correlation between C2V model rankings and human judgments, we simply generate a C2V score for each of the 1227 human-annotated sentence-paraphrase instances. For the REG model, since the same instances were used for training, we use 5-fold cross-validation to estimate model correlation. In each fold, we first run RFECV on the training portion, then train a regression model on the selected features and predict ratings for the test portion. The predicted ratings on held-out portions from each fold are compared to the mean annotator ratings, and Spearman correlation is calculated on the combined set of all instances.

We calculate precision under each model by soliciting human judgments, via the same crowdsourcing interface used to gather sentence annotations in Section 4.1. Specifically, for each of 40 hand-picked polysemous target words t (10 each nouns, verbs, adjectives, and adverbs), we select two paraphrases p and ask workers to judge whether t takes on the meaning of p in the top-10 PSTS sentences from $S^{\bar{t},p}$ as ranked by REG or C2V.

	LexSub (baseline)	PSTS+REG	PSTS+C2V
ρ	–	**0.40**	0.34
P@1	0.91	0.85	**0.98**
P@5	0.93	0.88	**0.97**
P@10	0.92	0.89	**0.96**

Table 4: Correlation (ρ) of REG and C2V scores with human ratings for 1227 PSTS sentence-paraphrase instances, and precision of top-1/5/10 ranked sentences as evaluated by humans.

We also use top-10 precision to see how our bilingual pivoting approach for enumerating meaning-specific sentences compares to a system that enumerates sentences using a LexSub model alone, without bilingual pivoting. The baseline LexSub model selects sentences containing *coach* in its *trainer* sense by scoring *trainer* as a substitute for *coach* in a large set of candidate sentences using CONTEXT2VEC, and ranking them. We consider the union of all PSTS sentence sets containing *coach*, $S^{\overline{coach},*}$, as candidates. The top-10 scoring sentences are evaluated by humans for precision, and compared to the ranked sets of top-10 PSTS sentences under the REG and C2V models. Results are given in Table 4.

The supervised REG model produces a higher correlation (0.40) between model scores and human ratings than does the unsupervised C2V model (0.34) or the PMI metric (0.23), indicating that REG may be preferable to use in cases where sentence quality estimation for a wide quality range is needed. While a correlation of 0.40 is not very high, it is important to note that the correlation between each individual annotator and the mean of other annotators over all target sentence-paraphrase instances was only 0.36. Thus the model predicts the mean annotator rating with roughly the same reliability as individual annotators.

For applications where it is necessary to choose only the highest-quality examples of target words with a specific paraphrase-aligned meaning, the

[5] http://u.cs.biu.ac.il/~nlp/resources/downloads/context2vec/

C2V ranking of PSTS sentences is best. 96% of top-10 ranked sentences under this model were evaluated by humans to be good examples of target words with the specified meaning, versus 89% for the REG model and 92% for the LexSub baseline. This indicates that the different methods for enumerating example sentences – bilingual pivoting (PSTS) and LexSub score – are complementary, and that combining the two produces the best results.

5 Hypernym Prediction in Context

Finally, we aim to demonstrate that PSTS can be used to automatically construct a training dataset for the task of predicting hypernymy in context, without relying on manually-annotated resources or a pre-trained word sense disambiguation model.

Most work on hypernym prediction has been done out of context: the input to the task is a pair of terms like (*table*, *furniture*), and the model predicts whether the second term is a hypernym of the first (in this case, it is). However, both Shwartz and Dagan (2016) and Vyas and Carpuat (2017) point out that hypernymy between two terms depends on their context. For example, the *table* mentioned in *"He set the glass down on the* ***table****"* is indeed a type of furniture, but in *"Results are reported in* ***table*** *3.1"* it is not. This is the motivation for studying the task of predicting hypernymy within a given context, where the input to the problem is a pair of sentences each containing a target word, and the task is to predict whether a hypernym relationship holds between the two targets. Example task instances are in Table 5.

Previous work on this task has relied on either human annotation, or the existence of a manually-constructed lexical semantic resource (i.e. WordNet), to generate training data. In the case of Shwartz and Dagan (2016), who examined fine-grained semantic relations in context, a dataset of 3,750 sentence pairs was compiled by automatically extracting sentences from Wikipedia containing target words of interest, and asking crowd workers to manually label sentence pairs with the appropriate fine-grained semantic relation.[6] Subsequently, Vyas and Carpuat (2017) studied hypernym prediction in context. They generated a larger dataset of 22k sentence pairs which used example sentences from WordNet as contexts, and WordNet's ontological structure to find sentence pairs where the presence or absence of a hypernym relationship could be inferred. This section builds on both previous works, in that we generate an even larger dataset of over 84k sentence pairs for studying hypernymy in context, and use the existing test sets for evaluation. However, unlike the previous methods, our dataset is constructed without any manual annotation or reliance on WordNet for contextual examples. Instead, we leverage the sense-specific contexts in PSTS to generate training instances automatically.

5.1 Producing a Training Set

Because PSTS can be used to query sentences containing target words with a particular fine-grained sense, our hypothesis is that, given a set of term pairs having known potential semantic relations, we can use PSTS to automatically produce a large training set of sentence pairs for contextual hypernym prediction. More specifically, our goal is to generate training instances of the form:

$$(t, w, c_t, c_w, l)$$

where t is a target term, w is a possibly related term, c_t and c_w are contexts, or sentences, containing t and w respectively, and l is a binary label indicating whether t and w are a hyponym-hypernym pair in the senses as they are expressed in contexts c_t and c_w. The proposed method for generating such instances from PSTS relies on WordNet (or another lexical semantic resource) only insofar as we use it to enumerate term pairs (t, w) with known semantic relation; the contexts (c_t, c_w) in which these relations hold or do not are generated automatically from PSTS.

The training set is deliberately constructed to include instances of the following types:

(a) Positive instances, where (t, w) hold a hypernym relationship in contexts c_t and c_w $(l = 1)$ (Table 5, example *a*).

(b) Negative instances, where (t, w) hold some semantic relation other than hypernymy (such as meronymy or antonymy) in contexts c_t and c_w $(l = 0)$. This will encourage the model to discriminate true hypernym pairs from other semantically related pairs (Table 5, example *b* shows an antonym pair in context).

[6] In this study, which included the relations *equivalence*, *forward* and *reverse entailment*, *negation/alternation*, *other-related*, and *independence*, hyponym-hypernym pairs were labeled as *forward entailment* and hypernym-hyponym pairs labeled as *reverse entailment* instances.

Ex.	Target Word (t)	Related Word (w)	Contexts	Hypernym (l)
(a)	tuxedo	dress	c_t: People believe my folderol because I wear a black **tuxedo**. c_w: The back is crudely constructed and is probably an addition for fancy **dress**.	Yes
(b)	defendant	plaintiff	c_t: The plaintiff had sued the **defendant** for defamation. c_w: The court found that the **plaintiff** had made sufficiently full disclosure.	No
(c)	bug	micro-phone	c_t: An address error usually indicates a software **bug**. c_w: You have to bring the **microphone** to my apartment.	No

Table 5: Example instances for contextual hypernym prediction, selected from the PSTS-derived dataset.

(c) Negative instances, where (t, w) hold a known semantic relation, including possibly hypernymy, in some sense, but the contexts c_t and c_w are not indicative of this relation ($l = 0$). This will encourage the model to take context into account when making a prediction (Table 5, example c).

Beginning with a target word t, the procedure for generating training instances of each type from PSTS is as follows:

Find related terms. The first step is to find related terms w such that the pair (t, w) are related in WordNet with relation type r (which could be one of synonym, antonym, hypernym, hyponym, meronym, or holonym), and $t \leftrightarrow w$ is a paraphrase pair present in PSTS. The related terms are not constrained to be hypernyms, in order to enable generation of instances of type (b) above.

Generate contextually related instances (types (a) and (b) above). Given term pair (t, w) with known relation r, generate sentence pairs where this relation is assumed to hold as follows. First, order PSTS sentences in $S^{\bar{t}w}$ (containing target t) and $S^{t\overline{w}}$ (containing related term w in its sense as a paraphrase of t) by decreasing quality score. Next, choose the top-k sentences from each ordered list, and select sentence pairs $(c_t, c_w) \in S^{\bar{t}w} \times S^{t\overline{w}}$ where both sentences are in their respective top-k lists. Add each sentence pair to the dataset as a positive instance ($l = 1$) if r = hypernym, or as a negative instance ($l = 0$) if r is something other than the hypernym relation.

Generate contextually unrelated instances (type (c) above). Given term pair (t, w) with known relation r, generate sentence pairs where this relation is assumed *not* to hold as follows. First, pick a confounding term t' that is a paraphrase of t (i.e. $t \leftrightarrow t'$ is in PPDB), but unrelated to w in PPDB. This confounding term is designed to represent an alternative sense of t. For example, a confounding term corresponding to the term pair (t, w) =(*bug*, *microphone*) could be *glitch* because it represents a sense of *bug* that is different from *bug*'s shared meaning with *microphone*. Next, select the top-$k/2$ sentences containing related term w in its sense as w' from $S^{\overline{w},w'}$ in terms of quality score. Choose sentence pairs $(c_t, c_w) \in S^{\bar{t},w} \times S^{\overline{w},w'}$ to form negative instances.

To form the PSTS-derived contextual hypernym prediction dataset, this process is carried out for a set of 3,558 target nouns drawn from the Shwartz and Dagan (2016) and Vyas and Carpuat (2017) datasets. For each target noun, all PPDB paraphrases that are hypernyms, hyponyms, synonyms, antonyms, co-hyponyms, or meronyms from WordNet were selected as related terms. There were $k = 3$ sentences selected for each target/related term pair, where the PSTS sentences were ranked by the C2V model. This process resulted in a dataset of over 84k instances, of which 32% are positive contextual hypernym pairs (type (a)). The 68% of negative pairs are made up of 38% instances where t and w hold some relation other than hypernymy in context (type (b)), and 30% instances where t and w are unrelated in the given context (type (c)).

5.2 Baseline IMS Training Set

In order to compare the quality of the PSTS-derived contextual hypernym dataset to one produced using sentences sense-tagged by a supervised WSD model, we generate a baseline training set using word instances with senses tagged by the English all-words WSD model It Makes Sense (IMS) (Zhong and Ng, 2010). IMS is a supervised sense tagger that uses a SVM classifier operating over syntactic and contextual features.

We begin by extracting an inventory of sentences pertaining to WordNet senses using IMS. Specifically, a pre-trained, off-the-shelf version of IMS[7] is used to predict WordNet 3.0 sense labels for instances of the same target nouns present in the PSTS-derived training set. The instances are drawn from the English side of the same English-foreign bitext used to extract PSTS, so the source corpora for the PSTS-derived and IMS contextual hypernym datasets are the same. We select the top sentences for each sense of each target noun, as ranked by IMS model confidence, as a sentence inventory for each sense.

Next, we extract training instances (t, w, c_t, c_w, l) using the same procedure outlined in Section 5.1. Term pairs (t, w) are selected such that t and w have related senses in WordNet, and both t and w are within the set of target nouns. Related instances are generated from the top-3 IMS-ranked sentences for the related senses of t and w, and unrelated sentences are chosen by selecting an un-related WordNet sense of t to pair with the original sense of w, and vice versa. Finally, we truncate the resulting set of training instances to match the PSTS-derived dataset in size and instance type distribution: 84k instances total, with 32% positive (contextual hypernym) pairs, 38% contextually related non-hypernym pairs, and 30% contextually unrelated pairs.

5.3 Contextual Hypernym Prediction Model

Having automatically generated a dataset from PSTS for studying hypernymy in context, the next steps are to adopt a contextual hypernym prediction model to train on the dataset, and then to evaluate its performance on existing hypernym prediction test sets.

The model adopted for predicting hypernymy in context is a fine-tuned version of the BERT pre-trained transformer model (Devlin et al., 2019) (Figure 4). Specifically, we use BERT in its configuration for sentence pair classification tasks, where the input consists of two tokenized sentences (c_t and c_w), preceded by a '`[CLS]`' token and separated by a '`[SEP]`' token. In order to highlight the target t and related term w in each respective sentence, we surround them with left and right bracket tokens "<" and ">". The model predicts whether the sentence pair contains contextualized hypernyms or not by processing the input through a transformer encoder, and feeding the output representation of the '`[CLS]`' token through fully connected and softmax layers.

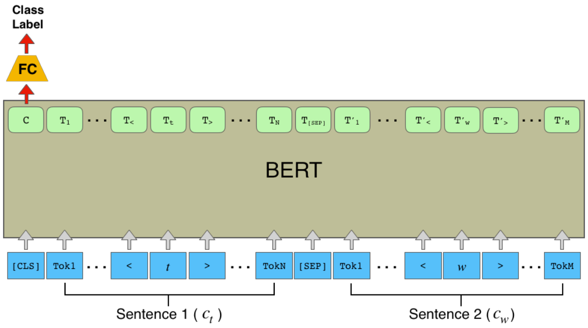


Figure 4: The contextual hypernym prediction model is based on BERT (Devlin et al., 2019). Input sentences c_t and c_w are tokenized, prepended with a `[CLS]` token, and separated by a `[SEP]` token. The target word t in the first sentence, c_t, and the related word w in the second sentence, c_w, are surrounded by < and > tokens. The class label (*hypernym* or *not*) is predicted by feeding the output representation of the `[CLS]` token through fully-connected and softmax layers.

5.4 Experiments

To test our hypothesis that PSTS can be used to generate a large, high-quality dataset for training a contextualized hypernym prediction model, we perform experiments that compare the performance of the BERT hypernym prediction model on existing test sets after training on our PSTS dataset, versus training on on datasets built using manual resources or a supervised WSD model.

We use two existing test sets for contextual hypernym prediction in our experiments. The first, abbreviated S&D-binary, is a binarized version of the fine-grained semantic relation dataset from Shwartz and Dagan (2016). While the original dataset contained five relation types, we convert all *forward entailment* and flipped *reverse entailment* instances to positive (hypernym) instances, and the rest to negative instances. The resulting dataset has 3750 instances (18% positive and 82% negative), split into train/dev/test portions of 2630/190/930 instances respectively. The second dataset used in our experiments is "WordNet Hypernyms in Context" (WHiC) from Vyas and Carpuat (2017). It contains 22,781 instances (23%

[7] `https://www.comp.nus.edu.sg/~nlp`

positive and 77% negative), split into train/dev/test portions of 15716/1704/5361 instances respectively. There are two primary differences between the WHiC and S&D-binary datasets. First, S&D-binary contains negative instances where the word pair has a semantic relation other than hypernymy in the given contexts – i.e. type (b) from Table 5 – whereas WHiC does not. Second, because its sentences are extracted from Wikipedia, S&D-binary contains some instances where the meaning of a word in context is ambiguous; WHiC sentences selected from WordNet are unambiguous. Our PSTS-derived contextual hypernym prediction dataset, which contains semantically related negative instances and has some ambiguous contexts (as noted in Section 4.1.1) is more similar in nature to S&D-binary.

For both the S&D-binary and WHiC datasets, we compare results of the BERT sentence pair classification model on the test portions after fine-tuning on the PSTS dataset, the supervised IMS baseline dataset, the original training set, or a combination of the PSTS dataset with the original training set. In order to gauge how different the datasets are from one another, we also experiment with training on S&D-binary and testing on WHiC, and vice versa. In each case we use the dataset's original dev portion for tuning the BERT model parameters (batch size, number of epochs, and learning rate). Results are reported in terms of weighted average F-Score over the positive and negative classes, and given in Table 6.

Training Set	**Test Set**	
	WHiC	S&D-binary
S&D-binary	68.6	79.2
WHiC	**78.7**	71.7
IMS	69.8	81.4
PSTS	73.4	79.7
PSTS+WHiC	78.5	
PSTS+S&D-binary		**82.5**

Table 6: Performance of the BERT fine-tuned contextual hypernym prediction model on two existing test sets, segmented by training set. All results are reported in terms of weighted average F1.

In the case of S&D-binary, we find that training on the 85k-instance PSTS dataset leads to a modest improvement in test set performance of 0.6% over training on the original 2.6k-instance manually-annotated training set. Combining the PSTS and original training sets leads to a 4.2% relative performance improvement over training on the original dataset alone, and out-performs the IMS baseline built using a supervised WSD system. However, on the WHiC dataset, it turns out that training on the PSTS dataset as opposed to the original 15.7k-instance WHiC training set leads to a relative 6.7% drop in performance. But training the model on the PSTS training data leads to better performance on WHiC than training on instances produced using the output of the supervised IMS WSD system, or from training on S&D-binary. It is not surprising that the PSTS-derived training set performs better on the S&D-binary test set than it does on the WHiC test set, given the more similar composition between PSTS and S&D-binary.

6 Conclusion

We present PSTS, a resource of up to 10k English sentence-level contexts for each of over 3M paraphrase pairs. The sentences were enumerated using a variation of bilingual pivoting (Bannard and Callison-Burch, 2005), which assumes that an English word like *bug* takes on the meaning of its paraphrase *fly* in sentences where it is translated to a shared foreign translation like *mouche* (fr). Human assessment of the resource shows that sentences produced by this automated process have varying quality, so we propose two methods to rank sentences by how well they reflect the meaning of the associated paraphrase pair. A supervised regression model has higher overall correlation (0.4) with human sentence quality judgments, while an unsupervised ranking method based on lexical substitution produces highest precision (96%) for the top-10 ranked sentences.

We leveraged PSTS to automatically produce a contextualized hypernym prediction training set, without the need for a supervised sense tagging model or existing hand-crafted lexical semantic resources. To evaluate this training set, we adopted a hypernym prediction model based on the BERT transformer (Devlin et al., 2019). We showed that this model, when trained on the large PSTS training set, achieves a slight gain of 0.6% accuracy relative to training on a smaller, manually-annotated training set, without the need for manual annotations. This suggests that it is worth exploring the use of PSTS to generate sense-specific datasets for other contextualized tasks.

Acknowledgements

We are grateful for support from the Allen Institute for Artificial Intelligence (AI2) Key Scientific Challenges program and the Google Ph.D. Fellowship program. This work was also supported by DARPA under the LORELEI program (HR0011-15-C-0115). The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA and the U.S. Government.

We especially thank our anonymous reviewers for their thoughtful, substantive, and constructive comments.

References

Domagoj Alagić, Jan Šnajder, and Sebastian Padó. 2018. Leveraging lexical substitutes for unsupervised word sense induction. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, pages 5004–5011, New Orleans, Louisiana. Association for the Advancement of Artificial Intelligence.

Rie Kubota Ando. 2006. Applying alternating structure optimization to word sense disambiguation. In Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL), pages 77–84, New York, New York. Association for Computational Linguistics.

Marianna Apidianaki. 2009. Data-driven semantic analysis for multilingual WSD and lexical selection in translation. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL), pages 77–85, Athens, Greece. Association for Computational Linguistics.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL), pages 597–604, Ann Arbor, Michigan. Association for Computational Linguistics.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. 1991. Word-sense disambiguation using statistical methods. In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL), pages 264–270, Berkeley, California. Association for Computational Linguistics.

Tsz Ping Chan, Chris Callison-Burch, and Benjamin Van Durme. 2011. Reranking bilingually extracted paraphrases using monolingual distributional similarity. In Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics, pages 33–42, Edinburgh, UK. Association for Computational Linguistics.

Yee Seng Chan and Hwee Tou Ng. 2005. Scaling up word sense disambiguation via parallel texts. In Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI), pages 1037–1042, Pittsburgh, Pennsylvania. Association for the Advancement of Artificial Intelligence.

Silvie Cinková, Martin Holub, and Vincent Kríž. 2012. Managing uncertainty in semantic tagging. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL), pages 840–850, Avignon, France. Association for Computational Linguistics.

Anne Cocos, Marianna Apidianaki, and Chris Callison-Burch. 2017. Word sense filtering improves embedding-based lexical substitution. In Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications, pages 110–119, Valencia, Spain. Association for Computational Linguistics.

Anne Cocos and Chris Callison-Burch. 2016. Clustering paraphrases by word sense. In Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pages 1463–1472, San Diego, California. Association for Computational Linguistics.

Ido Dagan. 1991. Lexical disambiguation: sources of information and their statistical realization. In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL), pages 341–342, Berkeley, California. Association for Computational Linguistics.

Ido Dagan and Alon Itai. 1994. Word sense disambiguation using a second language monolingual corpus. Computational Linguistics, 20(4):563–596.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Minneapolis, Minnesota. Association for Computational Linguistics.

Mona Diab and Philip Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. In Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL), pages 255–262, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Philip Edmonds and Scott Cotton. 2001. SENSEVAL-2: overview. In Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems, pages 1–5, Toulouse, France. Association for Computational Linguistics.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. Psychological Bulletin, 76(5):378.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. Using bilingual materials to develop word sense disambiguation methods. In Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation, pages 101–112, Montréal, Canada.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. Machine Learning, 46(1-3):389–422.

Els Lefever, Véronique Hoste, and Martine De Cock. 2011. ParaSense or how to use parallel corpora for word sense disambiguation. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL): Short Papers-Volume 2, pages 317–322, Portland, Oregon. Association for Computational Linguistics.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 Task 10: English lexical substitution task. In Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 48–53, Prague, Czech Republic. Association for Computational Linguistics.

Diana McCarthy and Roberto Navigli. 2009. The English lexical substitution task. Language Resources and Evaluation Special Issue on Computational Semantic Analysis of Language: SemEval-2007 and Beyond, 43(2):139–159.

Oren Melamud, Ido Dagan, and Jacob Goldberger. 2015a. Modeling word meaning in context with substitute vectors. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pages 472–482.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning (CONLL), pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Oren Melamud, Omer Levy, and Ido Dagan. 2015b. A simple word embedding model for lexical substitution. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, pages 1–7, Denver, Colorado. Association for Computational Linguistics.

Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. 2004. The SENSEVAL-3 english lexical sample task. In Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, pages 25–28, Barcelona, Spain. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26, Lake Tahoe.

George A. Miller. 1995. WordNet: A lexical database for English. Commun. ACM, 38(11):39–41.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994, pages 240–243, Plainsboro, New Jersey. Association for Computational Linguistics.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX), pages 95–100, Montréal, Canada. Association for Computational Linguistics.

Hwee Tou Ng, Bin Wang, and Yee Seng Chan. 2003. Exploiting parallel texts for word sense disambiguation: An empirical study. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), pages 455–462, Sapporo, Japan. Association for Computational Linguistics.

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. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL) (Volume 2: Short Papers), pages 425–430, Beijing, China. Association for Computational Linguistics.

Tommaso Petrolito and Francis Bond. 2014. A survey of WordNet annotated corpora. In Proceedings of the Seventh Global WordNet Conference, pages 236–245, Tartu, Estonia. University of Tartu Press.

Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL) - Volume 1: Long Papers, pages 1793–1803, Beijing, China. Association for Computational Linguistics.

Vered Shwartz and Ido Dagan. 2016. Adding context to semantic data-driven paraphrasing. In Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics, pages 108–113, Berlin, Germany. Association for Computational Linguistics.

Yogarshi Vyas and Marine Carpuat. 2017. Detecting asymmetric semantic relations in context: A case study on hypernymy detection. In Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM), pages 33–43, Vancouver, Canada. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. OntoNotes release 5.0 LDC2013T19. Linguistic Data Consortium, Philadelphia, PA.

Zhi Zhong and Hwee Tou Ng. 2010. It Makes Sense: A wide-coverage word sense disambiguation system for free text. In Proceedings of the ACL 2010 System Demonstrations, pages 78–83, Uppsala, Sweden. Association for Computational Linguistics.