Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 5978–5985
Marseille, 11–16 May 2020

One Classifier for All Ambiguous Words: Overcoming Data Sparsity by Utilizing Sense Correlations Across Words

Prafulla Kumar Choubey, Ruihong Huang
Department of Computer Science and Engineering
Texas A&M University, College Station, TX 77840
prafulla.choubey@tamu.edu, huangrh@cse.tamu.edu

Abstract

Most supervised word sense disambiguation (WSD) systems build word-specific classifiers by leveraging labeled data. However, when using word-specific classifiers, the sparseness of annotations leads to inferior sense disambiguation performance on less frequently seen words. To combat data sparsity, we propose to learn a single model that derives sense representations and meanwhile enforces congruence between a word instance and its right sense by using both sense-annotated data and lexical resources. The model is shared across words that allows utilizing sense correlations across words, and therefore helps to transfer common disambiguation rules from annotation-rich words to annotation-lean words. Empirical evaluation on benchmark datasets shows that the proposed shared model outperforms the equivalent classifier-based models by 1.7%, 2.5% and 3.8% in F1-score when using GloVe, ELMo and BERT word embeddings respectively.

Keywords: Word Sense Disambiguation, Systematic Polysemy, Lexical Semantics

1. Introduction

Word sense disambiguation (WSD) aims to automatically identify the correct meaning of a word in a particular context and is essential for many downstream natural language processing tasks, including information extraction, text classification, information retrieval, and machine translation.

Most existing data-driven approaches for Word sense disambiguation (WSD) build word-specific classifiers to predict the right sense of a word instance, which lack the capability to generalize across words and therefore require sufficient sense-annotated data for every word (de Lacalle and Agirre, 2015) in order to disambiguate them well. Consequently, the model's performance decreases significantly when there is a lack of training data for a word or some of its senses. As shown in Table 1, the state-of-the-art word-specific classifier model (Luo et al., 2018b) is able to achieve over 90% F1 score on senses with more than 200 training instances. But the performance of the model drops quickly on annotations-lean senses, especially on senses with less than 10 training instances.

No.	1-2	3-5	6-10	11-40	41-70	71-200	≥ 200
F1	**13.8**	**39.7**	**51**	63.9	86.1	89.1	93

Table 1: Performance of the state-of-the-art word-specific classifier model (Luo et al., 2018b) on word senses with a different number of training instances. The model was trained and evaluated on standard WSD training and test datasets, described further in the Evaluation Section.

Essentially, the design of word-specific classifiers overlooks the correlations among senses of different words. Studies on systematic polysemy (Apresjan, 1974; Rumshisky and Batiukova, 2008; Boleda et al., 2012) have shown that similar senses exist widely between words denoting objects of the same category, e.g., as shown in Table 2, the words "bank" and "school" can both refer to a building and an institution; similarly, the verbs "digest" and "swallow" share two related senses depending on whether the argument refers to concrete objects or abstract concepts. Modeling systematic polysemy (Pustejovsky, 1995; Utt and Padó, 2011) and accurately identifying words with similar senses is challenging though, which is *not* our focus here. Instead, we aim to directly address the limitation of word-specific classifiers for WSD that completely isolate a word from others and build a single classifier for WSD that is shared across words and senses. The shared model allows utilizing sense correlations across words and therefore allows to transfer common disambiguation rules learned from disambiguating annotation-rich words and applies the rules for improving the disambiguation of annotation-lean words that share a sense alternation pattern.

Specifically, we build a single neural network model for WSD that derives sense representations and meanwhile enforces congruence between a word instance and its right sense, by using both lexical resources and sense-annotated data. Using this shared model to measure the resemblance between each word sense and a word context, WSD becomes a ranking task that selects the word sense having the maximum similarity score with a word context. The shared model, agnostic to distinct words and word senses, can be trained using the entire sense-annotated corpus, which allows capturing correlations between senses across words and key attributes (e.g., concrete vs. abstract arguments) separating related word senses.

In principle, this approach is similar to knowledge-based approaches for word sense disambiguation, especially the classic algorithm Lesk and its extensions (Lesk, 1986; Agirre et al., 2014; Basile et al., 2014), that apply a common strategy to disambiguate any word by referring to sense representations and measuring overlaps between a word context and sense representations. However, different from the previous knowledge-based approaches that

Sense Group	Lexemes	Glosses	Examples
building	bank (n)	a **building** in which the business of banking transacted	the bank is on the corner of Nassau and Witherspoon
	school (n)	a **building** where young people receive education	the school was built in 1932
institution	bank (n)	a financial **institution** that accepts deposits and channels the money into lending activities	he cashed a check at the bank
	school (n)	an educational **institution**	the school was founded in 1900
concrete arguments	digest (v)	**convert food** into absorbable substances	I cannot digest milk products
	swallow (v)	pass through the esophagus as part of **eating or drinking**	swallow the raw fish–it won't kill you!
abstract arguments	digest (v)	arrange and integrate **in the mind**	I cannot digest all this information
	swallow (v)	**believe or accept** without questioning or challenge	am I supposed to swallow that story?

Table 2: Examples illustrating correlation among word-senses, both Nouns and Verbs. The bolded parts of glosses indicate the sense group and the underlined texts of example sentences illustrate key syntactic and semantic constraints of a sense.

directly use sense representations provided by lexical-semantic resources, we leverage sense-annotated corpus as well in a data-driven manner to *learn* to build sense representations and *learn* to measure the overlap between a word context and a sense representation.

In addition, the sense inventory WordNet (Miller et al., 1990) contains an example for many word senses and the well-composed example (Table 2) can precisely illustrate essential semantic or syntactic constraints in adopting a word sense. Therefore, we use the example of a word sense, if available, as a prototype in a regularization component of the shared model for guiding the WSD system to concentrate on most relevant segments of a word context. Empirical evaluation shows that our approach outperforms previous models on the benchmark English all-words WSD data-sets, and improves WSD performance on annotation-lean words.

2. Related Work

Knowledge-based word sense disambiguation approaches (Lesk, 1986; Agirre et al., 2014; Basile et al., 2014) rely on sense definitions and lexico-semantic resources to measure overlap between sense representations and a word context, which is flexible to handle infrequent words. However, in the absence of any supervision, knowledge-based approaches may suffer from a mismatch between sense representations and a word context and their performance consistently falls behind the data-driven approaches.

Data-driven methods (Zhong and Ng, 2010; Shen et al., 2013; Iacobacci et al., 2016) mostly build word specific classifiers (word-experts). Recent neural network models (Raganato et al., 2017a; Kågebäck and Salomonsson, 2016; Vial et al., 2018) use common first layers for all words and learn generalized low-level context representations. However, while using common context encoding layers, neural network models build word specific sense prediction layers and none of them completely pull out from the word expert notion. The crucial limitation of word expert models is their lack of capability of generalizing across words and their senses.

In line with the recent neural network-based models, Luo et al. (2018b) and Luo et al. (2018a) use both sense definitions and sense-annotated data in a neural network classifier-based models. Further, Vial et al. (2019) merges senses across words guided by the hypernym-hyponym relation in WordNet to ease the problem of sparsity, but the fundamental generalization issue that results from word specific sense prediction layers remains.

The proposed shared model for word sense disambiguation combines the best of both types of approaches. Specifically, by utilizing both the available sense-annotated data and knowledge resources, the sense-context resemblance model learns to generate and attend to word sense representations for disambiguating a word context. Similar to our approach, a concurrent work by Kumar et al. (2019) has also shown the benefit of learning sense embeddings in a shared model with common parameters for zero-shot WSD. In addition, we study how correlated and uncorrelated word senses, identified based on VerbNet, contribute to the improvement in the model's performance on annotation-lean word senses.

3. A Shared Neural Network Model for Word Sense Disambiguation

Our approach for word sense disambiguation measures the appropriateness of a word sense in a word context through a unified neural network that uses the same set of parameters for all the words and senses. Specifically, as shown in Figure 1, like the previous neural network-based methods (Raganato et al., 2017a; Luo et al., 2018b; Luo et al., 2018a), we use bidirectional LSTMs (bi-LSTMs) (Hochreiter and Schmidhuber, 1997) for encoding a word context, sense definitions and a prototype example sentence for each sense. Next, we calculate dot product similarities and absolute differences between a gloss encoding and the target word encoding. We also calculate dot product similarities and absolute differences between a sense prototype encoding and the word context encoding, with the prototype and context encodings attended by the gloss encoding. Then, we further apply a two layer fully connected feed-forward neural network over the four vectors recording similarities and differences, to predict the right sense for the target word. In all our discussions and experiments, we consider a word context as a sentence containing an ambiguous word.

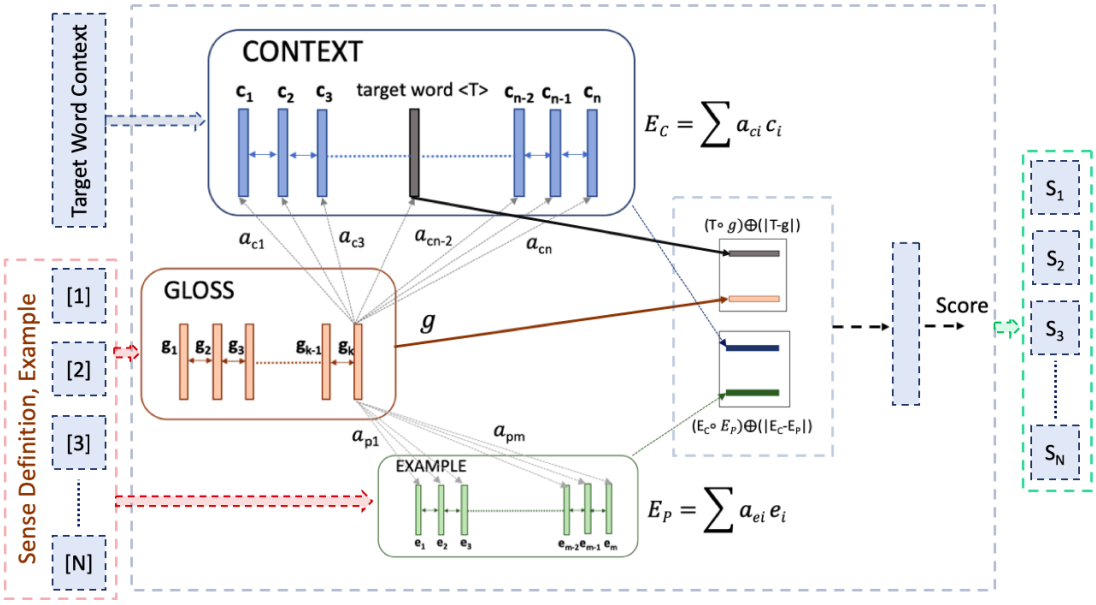

Figure 1: The Architecture of the Shared Neural Network Model. The model uses glosses and sense examples for measuring the correspondence between a word sense and a context.

3.1. The Gloss Encoding Module

The gloss encoding module represents each sense as a vector in a shared semantic space, distinguished from the previous data-driven methods that represent each sense as an orthogonal one-hot vector. It uses a simple bi-LSTM to encode a gloss and learn contiguous representations for word senses. We first transform the sequence of words in a gloss to pre-trained word embeddings in the embedding layer and then apply bi-LSTM ($biLSTM_{gloss}$) over the sequence of word embeddings to obtain their hidden states. Let n_g denote the number of words in a gloss G, we obtain the gloss representation g by using the last hidden state of bi-LSTM. Empirically, we found that the sense representation obtained by the simple bi-LSTM performs comparably to the ones obtained through complex operations such as pooling and self-attention.

$$\begin{aligned} H_G &= biLSTM_{gloss}(G) \in R^{n_g \times 2d_{rnn}} \\ g &= H_G[n_g] \in R^{2d_{rnn}} \end{aligned} \quad (1)$$

Note that this module is used to transform any sense to a vector representation in the same semantic space provided with its gloss.

3.2. The Target Word Encoding Module

Like the gloss encoding module, we use a separate bi-LSTM encoder ($biLSTM_{context}$) to obtain the target word encoding. Let C represent the sequence of words in a word context (a sentence) with n_c words and let n_w be the position of the ambiguous word in C. The $biLSTM_{context}$ transforms the sequence of words in a word context to their hidden states H_C. We obtain the target word representation, T, by using the hidden state of $biLSTM_{context}$ at the position n_w, which allows incorporating local contextual information into the target word representation.

$$\begin{aligned} H_C &= biLSTM_{context}(C) \in R^{n_c \times 2d_{rnn}} \\ T &= H_C[n_w] \in R^{2d_{rnn}} \end{aligned} \quad (2)$$

3.3. Regularization Using Sense Prototypes

The example provided by the sense inventory WordNet (Miller et al., 1990) for each word sense, if available, clearly illustrates essential semantic or syntactic constraints for adopting a word sense in a real sentence. Therefore, the shared model measures correspondence between a sense prototype and a word context as well, with both the prototype and context encodings attended by the gloss encoding.

First, we use the same bi-LSTM context encoder, $biLSTM_{context}$, to obtain the sense prototype encoding. Specifically, let P represent the sequence of words in a sense prototype sentence with n_p words. The $biLSTM_{context}$ transforms the sequence of words to their hidden states H_P.

$$H_P = biLSTM_{context}(P) \in R^{n_p \times 2d_{rnn}} \quad (3)$$

Then, we use the gloss encoding g to calculate attention scores over words in H_C and H_P, by calculating element-wise product and absolute difference (Mou et al., 2015) between g and each word of H_C and and H_P. The attention mechanism from the gloss embedding to H_C and and H_P allows the model to concentrate on context segments that are essential for recognizing the right word sense. The word context representation and sense prototype representation, E_C and E_P, are finally generated by summing over the products of attention weights A_C and A_P with hidden states H_C and H_P respectively.

$$\begin{aligned} h_C[i] &= [H_C[i] \cdot g; |H_C[i] - g| \in R^{4d_{rnn}}] \\ \alpha_C[i] &= W_{s1}(tanh(W_{s2} h_C[i] + b_{s2})) + b_{s1} \in R \\ A_C &= softmax(\alpha_C) \in R^n \\ E_C &= \sum_i A_C[i].H_C[i] \in R^{2d_{rnn}} \end{aligned} \quad (4)$$

$$
\begin{aligned}
h_P[i] &= [H_P[i] \cdot g; |H_P[i] - g| \in R^{4d_{rnn}}] \\
\alpha_P[i] &= W_{s1}(tanh(W_{s2}h_P[i] + b_{s2})) + b_{s1} \in R \\
A_P &= softmax(\alpha_P) \in R^n \\
E_P &= \sum_i A_P[i] \cdot H_P[i] \in R^{2d_{rnn}}
\end{aligned} \tag{5}
$$

Enforcing correspondence between the word context representation and the sense prototype representation helps in identifying specific syntactic or semantic constraints useful for disambiguating a word.

3.4. The Correspondence Calibrating Module

To measure the correspondence between a word sense and a word context, we first calculate element wise product scores and absolute differences between a gloss encoding and the target word encoding as well as between a sense prototype encoding and the word context encoding.

$$
I = [T \cdot g; |g - T|; E_P \cdot E_C; |E_P - E_C|] \in R^{8d_{rnn}} \tag{6}
$$

Next, we apply a two layer fully connected feed-forward neural network over the four vectors.

$$
\begin{aligned}
F &= tanh(W_L I + b_L) \in R^{2d_{rnn}} \\
score &= sigmoid(WF + b) \in R
\end{aligned} \tag{7}
$$

It is critical for the correspondence calculation module to effectively learn discriminative features and assign a higher correspondence score to the true word sense over all the other senses for a given word. By using element wise product scores and absolute differences to directly measure the similarity and differences between two vectors, the following feed-forward neural network can effectively distill essential features for calibrating the relevance of a word sense wrt. a word context.

Note that the neural network parameters W and b are shared across all words and senses. This is in contrast to the previous models that create separate W and b for each of the words and their senses. In addition, unlike previous approaches, the correspondence calculation module does not learn to directly label the target word with one of its senses. Rather, the module learns to associate the target word with its senses, considering one sense at a time. The correspondence score calculated for each sense, essentially, represents the extent to which the sense adheres to the target word in a particular context.

3.5. Learning and Inference

The optimization objective requires the shared model to assign a higher score to the true sense over all the other senses for a given word. Let T_s and F_s represent the score of the true sense and the scores of false senses for the target word respectively. We estimate the parameters of the shared model, Θ, by minimizing the following loss function:

$$
loss_\Theta = -(log(T_s) + \sum_{f_s \in F_s} (1 - log(f_s))) \tag{8}
$$

This loss function aims to maximize the predicted score for the true sense while minimizing the scores to zero for false senses.

3.6. The Single Classifier for All Ambiguous Words

Once we have optimized the shared model (Θ) on the training dataset, we use it to calculate a score for each of the senses of a given word that measures the correspondence between a word sense and the target word. Suppose we are given an ambiguous word T and its context sentence C, as well as all its senses S, each with a gloss G and a sense prototype P, we infer the most probable sense as p_s having the maximum predicted correspondence score.

$$
p_s = argmax_{<G,P> \in S} \Theta(C, T, <G, P>) \tag{9}
$$

The single classifier approach uses all the words and senses to train a shared neural network model, which can better capture structural regularities across correlated senses and sense alternation patterns across words. Meanwhile, the shared model uses many less parameters in a single neural net, unlike the word expert approach that uses word-specific classifiers and the parameters grow linearly with the number of ambiguous words.

4. Evaluation

4.1. Dataset

Training: We use the SemCor (Miller et al., 1993) dataset for training our neural network. SemCor is the largest manually annotated English corpus for word sense disambiguation. It consists of 352 documents from the Brown corpus with 226,036 sense annotations based on WordNet 1.6 (Miller et al., 1990) which was later mapped to WordNet 3.0 (Raganato et al., 2017b).

Validation and Evaluation: We evaluate our models on the benchmark fine-grained English all-words WSD dataset that includes test datasets from Senseval-2 (Edmonds and Cotton, 2001), Senseval-3 (Mihalcea et al., 2004), SemEval-2007 (Pradhan et al., 2007), SemEval 2013 (Navigli et al., 2013) and SemEval-2015 (Moro and Navigli, 2015). All these test sets were originally annotated with different versions of WordNet senses which were later standardized to WordNet 3.0 by Raganato et al. (2017b). Since both the training and test datasets are mapped to WordNet 3.0, we use WordNet 3.0 for extracting sense definitions and an example sentence for each sense, if available. Also, following the previous work on supervised WSD (Luo et al., 2018a; Luo et al., 2018b; Raganato et al., 2017a), we use the SemEval-2007 dataset, the smallest among all, for validation and parameter tuning.

4.2. Model Settings and Model Training

We determine model hyper-parameters based on the SemEval 2007 dataset. In all our neural net models[1], we use single layer BiLSTM with 512 hidden dimensions for encoding gloss and context. Both BiLSTMs, $biLSTM_{gloss}$ and $biLSTM_{context}$, use orthogonal initialization, and all linear layers (W_{s1}, W_{s2}, W_L, W) use uniform initializations. All the models are trained with mini-batch size of 8

[1]Models based on GloVe and ELMo embeddings are implemented using Pytorch 0.4.1 and AllenNLP. BERT embedding based models are implemented using Pytorch 1.2.0 and pytorch-transformers (Wolf et al., 2019).

using ADAM (Kingma and Ba, 2014) optimizer with learning rate set to 0.0001. For regularization, we use dropout rate (Srivastava et al., 2014) of 0.5 on the output activations of all encoders and neural layers. We use pre-trained word embeddings which are kept fixed during training. To ensure fair comparison with previous works, we evaluate our system using GloVe (Pennington et al., 2014), ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) word embeddings. Training runs for 20 epochs for models with GloVe embeddings and 6 epochs for models with ELMo and BERT embeddings.

4.3. Baseline Systems

We compare our proposed model with a heuristic baseline, recent knowledge based methods and data-driven methods.

1. **Heuristic MFS** predicts the most frequent sense of a word in the training dataset.
2. **Lesk**$_{ext+emb}$ uses a word similarity function defined in distributional semantics space to score gloss-context overlap and uses that to identify the most appropriate sense (Basile et al., 2014). Our models are directly comparable to this system, except that we leverage sense-annotated corpora to *learn* to attend to gloss.
3. **Babelfy** creates semantic interpretations of the input text and uses a densest subgraph heuristic to jointly perform WSD and entity linking (Moro et al., 2014).
4. **IMS** uses linear support vector machines on lexical and syntactic features defined on the context of target word (Zhong and Ng, 2010).
5. **BiLSTM**$_{+att.+LEX+POS}$ combines BiLSTM model with self-attention and uses multi-task learning framework for WSD, parts-of-speech tagging and semantic labeling (Raganato et al., 2017a).
6. **GAS** models semantic relationship between the context, gloss and hyper- and hypo-nyms of target word using memory modules in a word-specific classifier framework (Luo et al., 2018b).
7. **HCAN** extends the GAS model and uses the sophisticated hierarchical co-attention mechanism to generate gloss and context representations that attend to each other at both word and sentence embedding levels (Luo et al., 2018a).
8. **Sense-Compression** clusters different word-senses using hyper- and hypo-nym relations to reduce the number of word-senses in WordNet and improve coverage of supervised WSD systems. This is still a word-specific classifier model, and also the previous best performing model on the all-words WSD English benchmark datasets (Vial et al., 2019).

4.4. Proposed Single Classifier-based Models

1. **Single Classifier**$_{BiLSTM+Gloss+Example}$: the proposed shared neural network model for WSD that uses glosses and sense examples for measuring the correspondence between a word sense and a context.
2. **Single Classifier**$_{BiLSTM+Gloss}$: a variation of the shared model, with no regularization based on sense examples.

In addition, we implemented our own word-specific model, **Word-Specific Classifier**$_{BiLSTM+Gloss}$, which is equivalent to the **Single Classifier**$_{BiLSTM+Gloss}$, except that the word-specific model uses word-specific sense prediction layers. This model can be seen as a simplified implementation of the previous system **GAS** (Luo et al., 2018b) after ignoring hyper- and hypo-nyms and memory modules.

4.5. Results

The first section of Table 3 shows results of the Heuristic baseline and previous classic WSD models that rely on either a knowledge base or sense annotated corpus. The second section shows results of neural network models that leverage both knowledge base and annotations and use GloVe word embeddings. Using a much simpler architecture for encoding the gloss and context, the **Word-Specific Classifier**$_{BiLSTM+Gloss}$ approach catches up with the complex memory augmented **GAS** model that additionally exploits hyper- and hypo-nym relations. The **Single Classifier**$_{BiLSTM+Gloss}$ approach outperforms its equivalent word expert model **Word-Specific Classifier**$_{BiLSTM+Gloss}$ across all the four test datasets and with an improvement of 1.1% in the combined test set. The Single Classifier$_{BiLSTM+Gloss}$ approach can better disambiguate all types of words including nouns, verbs, adjectives and adverbs. Further, by using an example sentence for regularization, the complete model **Single Classifier**$_{BiLSTM+Gloss+Example}$ generalizes more easily and better captures essential syntactic and semantic constraints for recognizing a word sense, which improves the overall F1 scores by an additional 0.6%.

When using more powerful contextualized ELMo and BERT word embeddings (the third and fourth section of Table 3 respectively), we observe similar trends but with larger performance gains overall. Specifically, compared to its equivalent word expert model **Word-Specific Classifier**$_{BiLSTM+Gloss}$, the **Single Classifier**$_{BiLSTM+Gloss}$ achieved clear performance gains of 2.1% and 2.0% in F1-score on the combined test set, when using ELMo embeddings and BERT embeddings respectively. In addition, adding regularization based on sense examples further boosts the WSD performance when using either ELMo or BERT word embeddings. For instance, with BERT embeddings, the regularization yields a performance gain of 1.8% in F1-score on the combined test set and noticeable improvements across all the individual test sets and across several word types. Overall, the complete model **Single Classifier**$_{BiLSTM+Gloss+Example}$ consistently outperforms the previous state-of-the art word-specific model **Sense-Compression** by 1.3%, 1.5% and 1.2% in F1 score, when using GloVe, ELMo and BERT embeddings respectively.

4.6. Analysis

In order to evaluate the capabilities of single-classifier models in utilizing sense correlations and transferring common sense disambiguation rules from annotation-rich to

Models	Test Datasets				Combined Test Datasets				
	SE2	SE3	SE13	SE15	All	Noun	Verb	Adj	Adv
Heuristic MFS	65.6	66.0	63.8	67.1	65.5	67.7	49.8	73.1	80.5
Lesk$_{ext+emb}$ (Basile et al., 2014)	63.0	63.7	66.2	64.6	64.2	70.0	51.1	51.7	80.6
Babelfy (Moro et al., 2014)	67.0	63.5	66.4	70.3	66.4	68.9	50.7	73.2	79.8
IMS (Zhong and Ng, 2010)	70.9	69.3	65.3	69.5	68.9	70.5	55.8	75.6	82.9
BiLSTM$_{+att.+LEX+POS}$ (Raganato et al., 2017a)	72.0	69.1	66.9	71.5	69.9	71.5	57.5	75.0	83.8
GloVe Word Emeddings									
GAS (Luo et al., 2018b)	72.2	70.5	67.2	72.6	70.6	72.2	57.7	76.6	**85.0**
HCAN (Luo et al., 2018a)	72.8	70.3	68.5	72.8	71.1	72.7	**58.2**	77.4	84.1
Sense-Compression (Vial et al., 2019)	-	-	-	-	70.8	-	-	-	-
Word-Specific Classifier$_{BiLSTM+Gloss}$	72.6	70.3	66.8	71.3	70.4	72.0	58.0	75.2	84.0
Single Classifier$_{BiLSTM+Gloss}$	73.3	**71.0**	69.3	71.7	71.5	73.6	58.1	76.4	84.1
Single Classifier$_{BiLSTM+Gloss+Example}$	**74.0**	70.8	**70.3**	**73.0**	**72.1**	**74.2**	58.0	**78.1**	84.4
ELMo Word Embeddings									
Word-Specific Classifier$_{BiLSTM+Gloss}$	74.1	71.3	68.8	72.5	71.8	72.6	61.8	76.9	**85.1**
Single Classifier$_{BiLSTM+Gloss}$	**75.3**	73.1	**71.8**	75.6	73.9	75.5	63.3	77.7	84.4
Single Classifier$_{BiLSTM+Gloss+Example}$	75.2	**74.1**	**71.8**	**76.9**	**74.3**	**75.8**	**64.9**	**77.8**	83.4
BERT Word Embeddings									
Word-Specific Classifier$_{BiLSTM+Gloss}$	74.5	72.4	70.6	74.5	73.0	74.8	60.6	77.8	84.8
Single Classifier$_{BiLSTM+Gloss}$	76.7	74.2	74.1	74.1	75.0	77.1	63.0	78.5	85.8
Single Classifier$_{BiLSTM+Gloss+Example}$	**77.7**	**75.8**	**75.1**	**79.2**	**76.8**	**78.7**	**66.5**	**79.5**	**86.5**

Table 3: F1-scores for different baselines and proposed models on benchmark datasets for fine-grained English all-words WSD. SE2, SE3, SE13 and SE15 denote senseval-2, senseval-3, semeval 2013 and semeval 2015 datasets respectively. Most frequent sense is assigned to words for which none of its sense has been observed during the training. For direct comparison against Luo et al. (2018b) and Luo et al. (2018a), we have adopted their data filtering and evaluation settings exclusively for GloVe word-embedding based models.

# of Training Instances	1-2	3-5	6-10	>10
Percentage	10.9	10.7	12.7	65.7

Table 4: Segments of senses based on the number of training instances, and the size (percentage) of each segment of senses in the test set.

annotation-lean senses, we compare their performance against the word-specific classifier model on senses with different levels of annotation support. First, we remove test instances which use MFS-based back-off strategy and rank the remaining word senses based on the number of training instances each sense has. Then, we segment word senses into four portions with a reasonable density ($> 10\%$ of the test set) and ensure that senses with the same number of training instances are in one segment. Table 4 shows the four segments of senses and the percentage of each segment of senses in the test set.

As seen in Table 5, the performance gains for the **Single Classifier**$_{BiLSTM+Gloss}$ model mostly come from recognizing annotation-lean word senses (senses having 10 or less training instances). Through shared parameters, the single classifier model exploits systematic correlations across word-senses and thus effectively ease data-sparsity issues. The model **Single Classifier**$_{BiLSTM+Gloss+Example}$ further constrains the context encoder to focus on relevant contextual cues through the prototype sentence-based regularization. The improvements are consistent when using all the three word-embeddings and are strengthened by utilizing better contextualized word embeddings.

In addition, we further conduct experiments to study how additional training instances from other words help in disambiguating sparsely annotated word-senses. The hypothesis is that while the single classifier shared across words can enable learning from any additional training instances, training instances from words with well correlated word senses are more useful than instances from other words. Since recognizing and evaluating systematic polysemy is beyond the scope of this work, we use VerbNet[2] (Kipper et al., 2006; Loper and Bird, 2002) to approximately identify verbs that have correlated word senses. Specifically, we first identify verbs that have five or less training instances for each of its senses. In total, we identified 114 such annotation-lean verbs and they have 138 instances in the test set. Then, we use VerbNet to group the other verbs found in the training set into *in-class* and *out-of-the-class* verbs, depending on whether a verb is in the same verb class with one of the identified annotation-lean verbs. Coincidentally, the number of training instances of *in-class* verbs and *out-of-the-class* verbs are comparable[3].

Table 6 compares the performance of three models trained on in-class and out-of-the-class verb instances. We observe that the word-specific classifier model obtains comparable performance when trained with either in-class or out-of-the-class verb instances. However, the single-classifier models are able to effectively transfer common syntactic

[2]VerbNet is a broad-coverage verb lexicon that groups verbs into fine-grained verb classes extending Levin classes to achieve syntactic and semantic coherence among members of a class.

[3]Specifically, there are 28,640 and 26,923 training instances for *in-class* and *out-of-the-class* verbs respectively.

Models	GloVe				ELMo				BERT			
	1-2	3-5	6-10	>10	1-2	3-5	6-10	>10	1-2	3-5	6-10	>10
Word-Specific Classifier$_{BiLSTM+Gloss}$	13.8	39.7	51.0	76.5	16.6	40.4	54.0	79.1	24.1	41.8	57.1	80.5
Single Classifier$_{BiLSTM+Gloss}$	24.3	40.2	56.9	**77.2**	28.2	48.7	**62.3**	**79.9**	36.9	49.6	64.3	80.6
Single Classifier$_{BiLSTM+Gloss+Example}$	**25.8**	**43.3**	**60.6**	**77.2**	**31.1**	**53.1**	61.9	**79.9**	**38.7**	**59.3**	**68.6**	**82.2**

Table 5: F1-scores of the word-specific classifier and the two proposed single-classifier systems, evaluated separately on senses with different numbers of training instances.

Models	In	Out
Word-Specific Classifier$_{BiLSTM+Gloss}$	**34.8**	34.1
Single Classifier$_{BiLSTM+Gloss}$	**42.0**	37.7
Single Classifier$_{BiLSTM+Gloss+Example}$	**44.9**	40.6

Table 6: The performance (F1 scores) of BERT word embedding based models on the identified annotation-lean verbs, when only using training instances of In-class (In) verbs or Out-of-the-class (Out) verbs.

(and semantic) alternation patterns across verbs in the same VerbNet class and, thus, significantly improve the performance when trained on in-class verb instances. In addition, the single classifier models are able to leverage out-of-the-class verb instances to learn to better measure the correspondence between word sense and word context representations, outperforming the word-specific classifier as well, but the improvements of performance are less than the single classifier models trained using in-class verb instances.

5. Conclusions

Improving the word sense disambiguation performance on resource-low senses is necessary to build practically useful systems. We have presented a novel single classifier approach for word sense disambiguation that is distinguished from word-specific classification approaches and allows to capture sense correlations across words. In the future, we will continue to improve the performance of word sense disambiguation, especially on rare senses, by exploring semi-supervised learning. We are keen to incorporate word sense disambiguation into real applications as well.

6. Acknowledgements

We gratefully acknowledge support from National Science Foundation via the awards IIS-1942918, IIS-1755943 and IIS-1909252.

7. Bibliographical References

Agirre, E., López de Lacalle, O., and Soroa, A. (2014). Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.

Apresjan, J. D. (1974). Regular polysemy. *Linguistics*, 12(142):5–32.

Basile, P., Caputo, A., and Semeraro, G. (2014). An enhanced lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600.

Boleda, G., Schulte im Walde, S., and Badia, T. (2012). Modeling regular polysemy: A study on the semantic classification of catalan adjectives. *Computational Linguistics*, 38(3):575–616.

de Lacalle, O. L. and Agirre, E. (2015). A methodology for word sense disambiguation at 90% based on large-scale crowdsourcing. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 61–70.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Edmonds, P. and Cotton, S. (2001). Senseval-2: overview. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5. Association for Computational Linguistics.

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

Iacobacci, I., Pilehvar, M. T., and Navigli, R. (2016). Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 897–907.

Kågebäck, M. and Salomonsson, H. (2016). Word sense disambiguation using a bidirectional lstm. *arXiv preprint arXiv:1606.03568*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipper, K., Korhonen, A., Ryant, N., and Palmer, M. (2006). Extending verbnet with novel verb classes. In *LREC*, pages 1027–1032. Citeseer.

Kumar, S., Jat, S., Saxena, K., and Talukdar, P. (2019). Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681, Florence, Italy, July. Association for Computational Linguistics.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM.

Loper, E. and Bird, S. (2002). Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

Luo, F., Liu, T., He, Z., Xia, Q., Sui, Z., and Chang, B. (2018a). Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1402–1411. Association for Computational Linguistics.

Luo, F., Liu, T., Xia, Q., Chang, B., and Sui, Z. (2018b). Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2473–2482. Association for Computational Linguistics.

Mihalcea, R., Chklovski, T., and Kilgarriff, A. (2004). The senseval-3 english lexical sample task. In *Proceedings of SENSEVAL-3, the third international workshop on the evaluation of systems for the semantic analysis of text*.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Miller, G. A., Leacock, C., Tengi, R., and Bunker, R. T. (1993). A semantic concordance. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, pages 303–308, Stroudsburg, PA, USA. Association for Computational Linguistics.

Moro, A. and Navigli, R. (2015). Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 288–297.

Moro, A., Raganato, A., and Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Mou, L., Men, R., Li, G., Xu, Y., Zhang, L., Yan, R., and Jin, Z. (2015). Natural language inference by tree-based convolution and heuristic matching. *arXiv preprint arXiv:1512.08422*.

Navigli, R., Jurgens, D., and Vannella, D. (2013). Semeval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 222–231.

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

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

Pradhan, S. S., Loper, E., Dligach, D., and Palmer, M. (2007). Semeval-2007 task 17: English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 87–92. Association for Computational Linguistics.

Pustejovsky, J. (1995). The generative lexicon: A theory of computational lexical semantics.

Raganato, A., Bovi, C. D., and Navigli, R. (2017a). Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167.

Raganato, A., Camacho-Collados, J., and Navigli, R. (2017b). Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 99–110.

Rumshisky, A. and Batiukova, O. (2008). Polysemy in verbs: systematic relations between senses and their effect on annotation. In *Proceedings of the Workshop on Human Judgements in Computational Linguistics*, pages 33–41. Association for Computational Linguistics.

Shen, H., Bunescu, R., and Mihalcea, R. (2013). Coarse to fine grained sense disambiguation in wikipedia. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, volume 1, pages 22–31.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Utt, J. and Padó, S. (2011). Ontology-based distinction between polysemy and homonymy. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 265–274. Association for Computational Linguistics.

Vial, L., Lecouteux, B., and Schwab, D. (2018). Improving the coverage and the generalization ability of neural word sense disambiguation through hypernymy and hyponymy relationships. *CoRR, abs/1811.00960*.

Vial, L., Lecouteux, B., and Schwab, D. (2019). Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation. *arXiv preprint arXiv:1905.05677*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhong, Z. and Ng, H. T. (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. Association for Computational Linguistics.