PoKED: A Semi-Supervised System for Word Sense Disambiguation

Feng Wei [1]

Abstract

In this paper, we propose a semi-supervised neural system, named Position-wise Orthogonal Knowledge-Enhanced Disambiguator (PoKED), which effectively supports attention-driven, long-range dependency modeling for word sense disambiguation (WSD) tasks. The proposed PoKED system incorporates position-wise encoding into an orthogonal framework and applies a knowledge-based attentive neural model to solve the WSD problem. Our proposed unsupervised language model is trained over unlabeled corpora. Then the pre-trained language model is used to abstract the surrounding context of polysemous words in labeled corpora into context embeddings. We further use the semantic relations in the WordNet, by extracting semantic level inter-word connections from each document-sentence pair in the WSD dataset. Our experimental results from standard benchmarks show that our proposed system, PoKED, can achieve competitive performance compared with state-of-the-art knowledge-based WSD systems.

1. Introduction

Word Sense Disambiguation (WSD) is the task of mapping an ambiguous word in a given context to its correct meaning. For example, the word `bank` can either mean a `financial institution` or `sloping land`, based on different contexts. WSD is an important problem in natural language processing (NLP), both in its own right and as a stepping stone to more advanced tasks such as machine translation, information extraction and retrieval, and question answering. WSD, being "AI-complete" (Navigli, 2009), is still an open problem after over two decades of research. We can roughly distinguish between supervised and unsupervised approaches. Supervised methods require sense-annotated training data and are suitable for lexical sample WSD tasks when systems are required to disambiguate a restricted set of target words. The performance of supervised systems, however, is limited in WSD tasks as labeled data for the full lexicon is sparse and difficult to obtain. As WSD tasks are challenging and have practical applications, there has been interest in developing unsupervised systems. These systems only require an external knowledge source (e.g., WordNet) but no labeled training data.

Example 1: An example of the importance of human semantic knowledge to the word sense disambiguation. Our experiment shows without incorporating human semantic knowledge, the model would wrongly predict a sense of the target word "document". We believe the reason is due to the difficulty for the model to find a direct relationship through the given word and plain document. In the example, we can find the answer because we know "document" is a hypernym of "information", or "information" is a hyponym of "document". The example is selected from SemEval-15.

Document: *This document is a summary of the European Public Assessment Report (EPAR). It explains how the Committee for Medicinal Products for Human Use (CHMP) assessed the studies performed, to reach their recommendations on how to use the medicine. If you need more information about your medical condition or your treatment, read the Package Leaflet (also part of the EPAR) or contact your doctor or pharmacist. If you want more information on the basis of the CHMP recommendation, read the Scientific Discussion (also part of the EPAR).*
. . .

Sentence: *This document is a summary of the European Public Assessment Report (EPAR).*

Answer: *<noun.communication>[10] S: (n) document.01 (document%1:10:00::), written document.01 (written_document%1:10:00::), papers.01 (papers%1:10:00::) (writing that provides information (especially information of an official nature))*

In this paper, we propose a semi-supervised neural system for WSD tasks, which utilizes the whole document as the context for a word, rather than just the encompassing sentence used by most WSD systems. In order to model the whole document for WSD, we propose to incorporate recent position-wise encoding (Watcharawittayakul et al., 2018; Wei et al., 2019) into an orthogonal framework (Zhang et al., 2016; Wei et al., 2020). Our proposed model is used to train a pseudo-language model over unlabeled corpora. The pre-trained language model is then used to abstract the surrounding context of polysemous words in labeled corpora into context embeddings. Moreover, we propose a data enrichment method, which uses WordNet to extract inter-word

[1] Department of Electrical Engineering and Computer Science, York University, 4700 Keele St, Toronto, ON M3J 1P3, Canada. Correspondence to: Feng Wei <fwei@cse.yorku.ca>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

semantic connections as general knowledge from each given document. As shown in Example 1, such human semantic knowledge is essential to WSD tasks. In addition, we propose an end-to-end knowledge-based attentive neural WSD model, which explicitly uses the above extracted general knowledge to assist its attention mechanisms (Bahdanau et al., 2015). Our semi-supervised WSD system (PoKED), comprised of one unsupervised position-wise orthogonal network (PoNet) and one supervised knowledge-enhanced word sense disambiguator (KED) based on attentive networks. We evaluate our system, PoKED, on standard benchmarks (Raganato et al., 2017) and show that the proposed model, utilizing the whole document as the context for a word to be disambiguated, achieves better performance than the previous state-of-the-art knowledge-based models.

2. Related Work

Word sense disambiguation (WSD) approaches can be divided into two main categories: supervised, which require human intervention in the creation of sense-annotated datasets, and the so-called knowledge-based approach (Navigli, 2009), which requires the construction of a task-independent lexical-semantic knowledge resource. Once that work is available, it uses models that are completely autonomous.

Supervised. A popular system, *It makes sense* (Zhong & Ng, 2010), takes advantage of standard WSD features such as POS-tags, word co-occurrences, and collocations and creates individual support vector machine classifiers for each ambiguous word. Newer supervised models use deep neural networks and especially long short-term memory (LSTM) networks, a type of recurrent neural network particularly suitable for handling arbitrary-length sequences. (Yuan et al., 2016) proposed a deep neural model trained with large amounts of data obtained in a semi-supervised fashion. This model was re-implemented by (Le et al., 2018), reaching comparable results with a smaller training corpus. (Raganato et al., 2017) introduced two approaches for neural WSD using models developed for machine translation and substituting translated words with sense-annotated ones. (Luo et al., 2018) proposed combining labeled data and knowledge-based information in recent work. (Uslu et al., 2018) proposed *fastSense*, a model inspired by *fastText* (Joulin et al., 2017) which, rather than predicting context words, predicts word senses. More recently, (Huang et al., 2019) constructed context-gloss pairs and propose three BERT-based models for WSD. (Hadiwinoto et al., 2019) explored different strategies of integrating pre-trained contextualized word representations.

Knowledge-based. Knowledge-based models, instead, use the structural properties of a lexical-semantic knowledge base, and typically use the relational information between concepts in the semantic graph together with the lexical information contained therein (Navigli & Lapata, 2009). A popular algorithm used to select the sense of each word in this graph is PageRank (Page et al., 1999) that performs random walks over the network to identify the important nodes (Mihalcea et al., 2004). Another knowledge-based approach is Babelfy (Moro et al., 2014), which defines a semantic signature for a given context and compares it with all the candidate senses in order to perform the disambiguation task. (Chaplot & Salakhutdinov, 2018) proposed a method that uses the whole document as the context for the words to be disambiguated. It models word senses using a variant of the Latent Dirichlet Allocation framework (Blei et al., 2003), in which the topic distributions of the words are replaced with sense distributions modeled by a logistic normal distribution according to the frequencies obtained from WordNet. More recently, (Maru et al., 2019) introduced *SyntagNet*, a novel resource consisting of manually disambiguated lexical-semantic combinations. (Tripodi & Navigli, 2019) presented *WSDG*, a flexible game-theoretic model for WSD.

3. Position-wise Orthogonal Knowledge-Enhanced Disambiguator (PoKED)

In this section, we describe in detail the proposed semi-supervised neural system (PoKED) for WSD tasks. We aim to explore how position-wise embedding (unsupervised) could help the downstream WSD task, and how information from descriptive linguistic knowledge graphs (WordNet) can be incorporated into neural network architectures to improve the linguistic WSD task.

3.1. Unsupervised Language Model (PoNet)

First, we elaborate on the proposed unsupervised language model named PoNet.

The linguistic distribution hypothesis states that words that occur in close proximity should have a similar meaning. It implies that the particular sense of a polysemous word is highly related to its surrounding context. Moreover, humans decide the sense of a polysemous word by firstly understanding its occurring context (Harris, 1954). Following this theory, our proposed model has two stages: training a position-wise orthogonal network (PoNet) that abstracts context as embeddings as shown in Figure 1, and performing knowledge-based attentive WSD classification over pre-trained context embeddings as shown in Figure 4.

3.1.1. Position-wise Encoding

Recently an alternative method (Watcharawittayakul et al., 2018; Wei et al., 2019) of commonly used sequence embed-

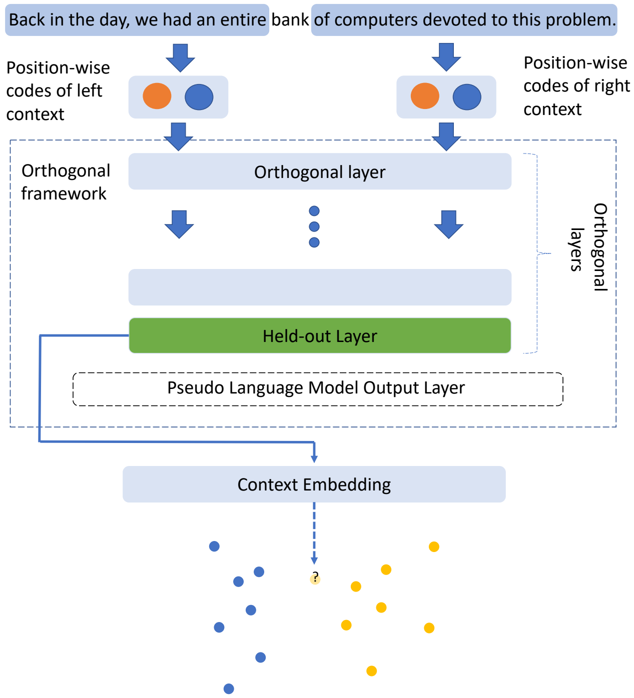
Back in the day, we had an entire bank of computers devoted to this problem.
Position-wise codes of left context
Position-wise codes of right context
Orthogonal framework
Orthogonal layer
Orthogonal layers
Held-out Layer
Pseudo Language Model Output Layer
Context Embedding
?

Figure 1. The diagram of the unsupervised position-wise orthogonal (PoNet) language module.

ding representations has been proposed, and it achieved competitive results in language modeling. The authors proved a nice theoretical property that guarantees that extracted codes can almost uniquely encode any variable-length sequence of words into a fixed-size representation without losing any information.

Given a vocabulary V, where each word can be represented by a 1-of-$|V|$ one-hot vector. Let $S = \{w_1, \cdots, w_N\}$ denote a sequence of N words from V, and e_n denote the one-hot vector of the n-th word in S, where $1 \leq n \leq N$. Assuming $z_0 = 0$, the extracted code z_n of the sequence from word w_1 to w_n is as follows:

$$z_n = \alpha \cdot z_{n-1} + e_n \tag{1}$$

where α is a constant forgetting factor. Thus, z_n can be viewed as a fixed-size representation of the subsequence $\{w_1, \cdots, w_n\}$. We can see that, according to the theoretical properties presented in (Zhang et al., 2015), any sequence of variable length can be uniquely and losslessly encoded into a fixed-size representation.

The main idea of position-wise encoding is to generate augmented encoding codes by concatenating two codes using two different forgetting factors. Each of these codes is still computed in the same way as the mathematical formulation shown in Equation (1). By using two different forgetting factors in the two codes, we can represent both short-term and long-term dependencies. Hence, our position-wise encoding can maintain the sensitivity to both nearby and faraway context.

3.1.2. Orthogonal Framework

More recently, a novel orthogonal framework (Zhang et al., 2016; Wei et al., 2020) has been proposed to learn neural networks in either supervised or unsupervised way. This framework introduces a linear orthogonal projection to reduce the dimensionality of the raw high-dimension data and then uses a finite mixture distribution to model the extracted features. By splitting the feature extraction and data modeling into two separate stages, it can derive a good feature extraction model that can generate better low-dimension features for the further learning process. More importantly, based on the analysis in (Zhang et al., 2016), the orthogonal framework has a tight relationship with neural networks since each hidden layer can also be viewed as an orthogonal model being composed of the feature extraction stage and data modeling stage. Therefore, the maximum likelihood-based unsupervised learning as well as the minimum cross-entropy error based supervised learning algorithms can be used to learn neural networks under the orthogonal framework for deep learning. In this case, the standard back-propagation method can be used to optimize the objective function to learn the models except that the orthogonal constraints are imposed for all projection layers during the training procedure.

Simply put, in practice in terms of the orthogonal formulation, (Zhang et al., 2016) proposed modeling z, which is heavily de-correlated but may still exist in a rather high dimension feature space, with a finite mixture model:

$$p(z) = \sum_{k=1}^{K} \pi_k \cdot f_k(z|\theta_k) \tag{2}$$

where K is the number of mixture components, π_k is the mixture weight of the k-th component ($\sum_{k=1}^{K} \pi_k = 1$), $f_k()$ denotes a selected distribution from the exponential family, and θ_k denotes all model parameters of $f_k()$.

An example of an orthogonal layer in deep feedforward networks is shown in Figure 2. For one hidden layer with input vector x ($x \in \mathbb{R}^D$) and output vector y ($y \in \mathbb{R}^G$), it is first split into two layers:

- The first layer is a linear orthogonal projection layer, which is used to project x to a feature vector z ($z \in \mathbb{R}^M, M < D$) and remove the noise signals by using an orthogonal projection matrix $U : z = Ux$.
- The second layer is a non-linear model layer, which converts z to the output vector y following the selected model $f_k()$ and a nonlinear log-likelihood pruning operation.

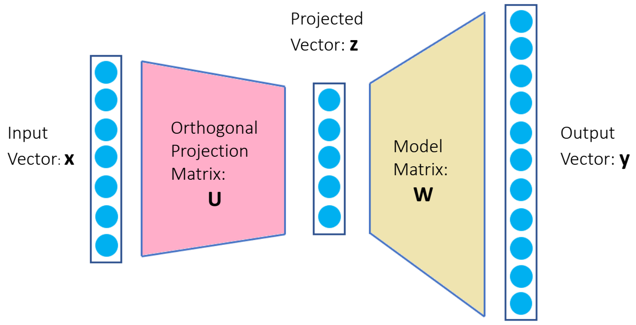

Figure 2. The orthogonal framework is viewed as a hidden layer in deep feedforward networks.

As the pseudo-language model (PoNet) is trained to predict the target word, the output layer is irrelevant to WSD tasks. The remaining layers, however, have learned the ability to generalize features from word to context during the training process. The held-out layer (the second last layer) are retained as context embeddings, which provides an effective representation of the surrounding context of a given target word.

3.2. Supervised Knowledge-based Attentive Model (KED)

In this section, we describe in detail the proposed supervised knowledge-enhanced attentive networks named KED for WSD tasks, along with the data enrichment method with WordNet.

3.2.1. Data Enrichment with WordNet

WordNet is a comprehensive lexical database for the English language (Miller, 1995), and is commonly used as the sense repository in WSD systems.

To provide our WSD model with explicit knowledge, we enrich the gloss information by extracting semantic level inter-word connections from each document-sentence pair in it; therefore we propose a WordNet-based data enrichment method.

Words in WordNet are organized into synsets, as shown in Figure 3, which in turn are related to each other through semantic relations, such as "hypernym" and "hyponym". In our data enrichment method, we use the semantic relations of WordNet to extract semantic level inter-word connections from each document-sentence pair in the WSD dataset. For each word w in a document-sentence pair, we need to obtain a set Z_w, which contains the positions of the document words that w is semantically connected to. Besides, when w itself is a document word, we also need to ensure that its position is excluded from Z_w.

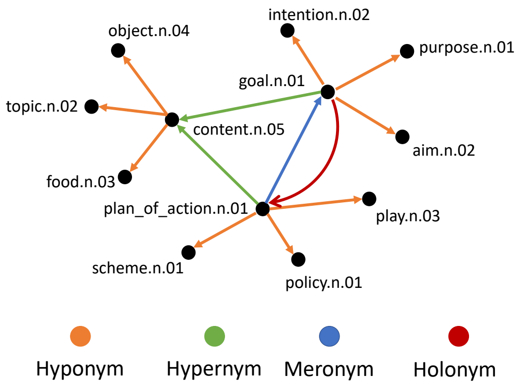

Figure 3. WordNet example showing several synsets and the relations between them.

Given a word w, its directly-involved synsets Φ_w represents the synsets that w belongs to, and its indirectly-involved synsets $\bar{\Phi}_w$ represents the synsets that are related to those in Φ_w through semantic relations. Based on the two concepts, we propose the following hypothesis: given a subject word w_s and an object word w_o, w_s is semantically connected to w_o if and only if $(\Phi_{w_s} \bigcup \bar{\Phi}_{w_s}) \bigcap \Phi_{w_o} \neq \emptyset$. According to the hypothesis, Algorithm 1 describes the process of extracting semantic level inter-word connections from each document-sentence pair.

Given a word w, we can easily obtain its directly-involved synsets Φ_w from WordNet, but obtaining its indirectly-involved synsets $\bar{\Phi}_w$ is much more complicated, because in WordNet, the way synsets are related to each other is flexible and extensible. In some cases, a synset is related to another synset through a single semantic relation. For example, the synset "cold.a.01" is related to the synset "temperature.n.01" through the semantic relation "attribute". In many other cases, however, a synset is related to another synset through a semantic relation chain. For example, first the synset "keratin.n.01" is related to the synset "feather.n.01" through the semantic relation "substance holonym"; then the synset "feather.n.01" is related to the synset "bird.n.01" through the semantic relation "part holonym"; and finally the synset "bird.n.01" is related to the synset "parrot.n.01" through the semantic relation "hyponym"; thus we can say that the synset "keratin.n.01" is related to the synset "parrot.n.01" through the semantic relation chain "substance holonym → part holonym → hyponym". We name each semantic relation in a semantic relation chain as a hop. Therefore, the above semantic relation chain is a 3-hop chain. Besides, each single semantic relation is a 1-hop semantic relation chain.

Let us use $\Gamma := \{\gamma_1, \gamma_2, ...\}$ to represent the semantic relations of WordNet, and use $\Omega_{\phi}^{\gamma_i}$ to represent the synsets that a synset ϕ is related to through a single semantic relation

Algorithm 1 Extract semantic level inter-word connections from each document-sentence pair

```
procedure EXTRACT($D$,$S$)
Input: Given a document $D$ and a relevant sentence $S$.
Output: Return the extraction results on $D$ and $S$
    for each document word $d_i$ in $D$ do
        $Z_{d_i} \leftarrow \{j \in \{1, ..., n\} \backslash \{i\} : (\Phi_{d_i} \bigcup \bar{\Phi}_{d_i}) \bigcap \Phi_{d_j} \neq \emptyset\}$    ▷ Obtain the extraction results $Z_{d_i}$
    for each sentence word $s_i$ in $S$ do
        $Z_{s_i} \leftarrow \{j \in \{1, ..., n\} \backslash \{i\} : (\Phi_{s_i} \bigcup \bar{\Phi}_{s_i}) \bigcap \Phi_{d_j} \neq \emptyset\}$    ▷ Obtain the extraction results $Z_{s_i}$
```

$\gamma_i \in \Gamma$. Since $\Omega_\phi^{\gamma_i}$ is easy to obtain from WordNet, we can further obtain the synsets that ϕ is related to through 1-hop semantic relation chains: $\Psi_\phi^1 = \bigcup_{\gamma_i \in \Gamma} \Omega_\phi^{\gamma_i}$, the synsets that ϕ is related to through 2-hop semantic relation chains: $\Psi_\phi^2 = \bigcup_{\hat{\phi} \in \Psi_\phi^1} \bigcup_{\gamma_i \in \Gamma} \Omega_{\hat{\phi}}^{\gamma_i}$, and by induction, the synsets that ϕ is related to through k-hop semantic relation chains: $\Psi_\phi^k = \bigcup_{\hat{\phi} \in \Psi_\phi^{k-1}} \bigcup_{\gamma_i \in \Gamma} \Omega_{\hat{\phi}}^{\gamma_i}$. In theory, if we do not limit the hop counts of semantic relation chains, ϕ can be related to all other synsets in WordNet, which is meaningless in many cases. Therefore, we use a hyper-parameter $\tau \in \mathbb{N}$ to represent the maximum hop count of semantic relation chains, and only consider the semantic relation chains that have no more than τ hops. Based on the above descriptions, given a word w and its directly-involved synsets Φ_w, we can obtain its indirectly-involved synsets: $\bar{\Phi}_w = \bigcup_{\phi \in \Phi_w} \bigcup_{k=1}^{\tau} \Psi_\phi^k$

3.2.2. Knowledge-based Attentive Neural Model

In this section, we describe in detail the proposed knowledge-based attentive model for WSD tasks. The key components of our model are the attention mechanisms, (i.e., knowledge-enhanced joint-attention and knowledge-enhanced self-attention). Knowledge-enhanced joint-attention aims to fuse the sentence representations into the document representations so as to obtain the sentence-aware document representations. Furthermore, Knowledge-enhanced self-attention aims to fuse the sentence-aware document representations into themselves so as to obtain the final document representations. More importantly, the most remarkable feature of our model is that it explicitly uses the general knowledge extracted by the data enrichment method to assist its attention mechanisms.

Given a document $D = \{d_1, ..., d_n\}$ and a relevant sentence $S = \{s_1, ..., s_m\}$, the task is to predict a sense among a list of K candidates $\mathscr{C} = \{\mathscr{c}_1, ..., \mathscr{c}_k\}$. As depicted in Figure 4, our proposed end-to-end supervised WSD model consists of five layers:

- Given a document-sentence pair, the *lexicon embedding layer* encodes the lexical features of each word to generate the document lexicon embeddings and the sentence lexicon embeddings. For each word, we use our pre-trained context embeddings based on position-wise encoding approach and orthogonal framework described in 3.1, and obtain its character embedding with a Convolutional Neural Network (CNN) (Kim, 2014). For both the document and the sentence, we pass the concatenation of the word embeddings and the character embeddings through a shared dense layer with ReLU activation, whose output dimensionality is d. Therefore we obtain the document lexicon embeddings $L_D \in \mathbb{R}^{d \times n}$ and the sentence lexicon embeddings $L_S \in \mathbb{R}^{d \times m}$.

- Based on the two lexicon embeddings, the *context embedding layer* encodes the contextual clues about each word to generate the document context embeddings and the sentence context embeddings. For both the document and the sentence, we process the lexicon embeddings (i.e., L_D for the document and L_S for the sentence) with a shared bidirectional LSTM (BiLSTM) (Hochreiter & Schmidhuber, 1997), whose hidden state dimensionality is $\frac{1}{2}d$. By concatenating the forward LSTM outputs and the backward LSTM outputs, we obtain the document context embeddings $C_D \in \mathbb{R}^{d \times n}$ and the sentence context embeddings $C_S \in \mathbb{R}^{d \times m}$.

- Based on the two context embeddings, the *coarse-grained memory layer* performs both document-to-sentence and sentence-to-document attention to generate the preliminary memories over the document-sentence pair. First we use *knowledge-enhanced joint-attention* (discussed in section 3.2.3) to fuse C_D into C_S, the outputs of which are represented as $\tilde{G} \in \mathbb{R}^{d \times n}$. Then we process $\tilde{G}$ with a BiLSTM, whose hidden state dimensionality is $\frac{1}{2}d$. By concatenating the forward LSTM outputs and the backward LSTM outputs, we obtain the coarse-grained memories $G \in \mathbb{R}^{d \times n}$, which are the sentence-aware document representations.

- Based on the coarse-grained memories, the *fine-grained memory layer* generates the refined memories over the document-sentence pair. First we use *knowledge-enhanced self-attention* (discussed in section 3.2.4) to fuse G into themselves, the outputs of which are represented as $\tilde{H} \in \mathbb{R}^{d \times n}$. Then we process $\tilde{H}$ with a BiLSTM, whose hidden state dimensionality

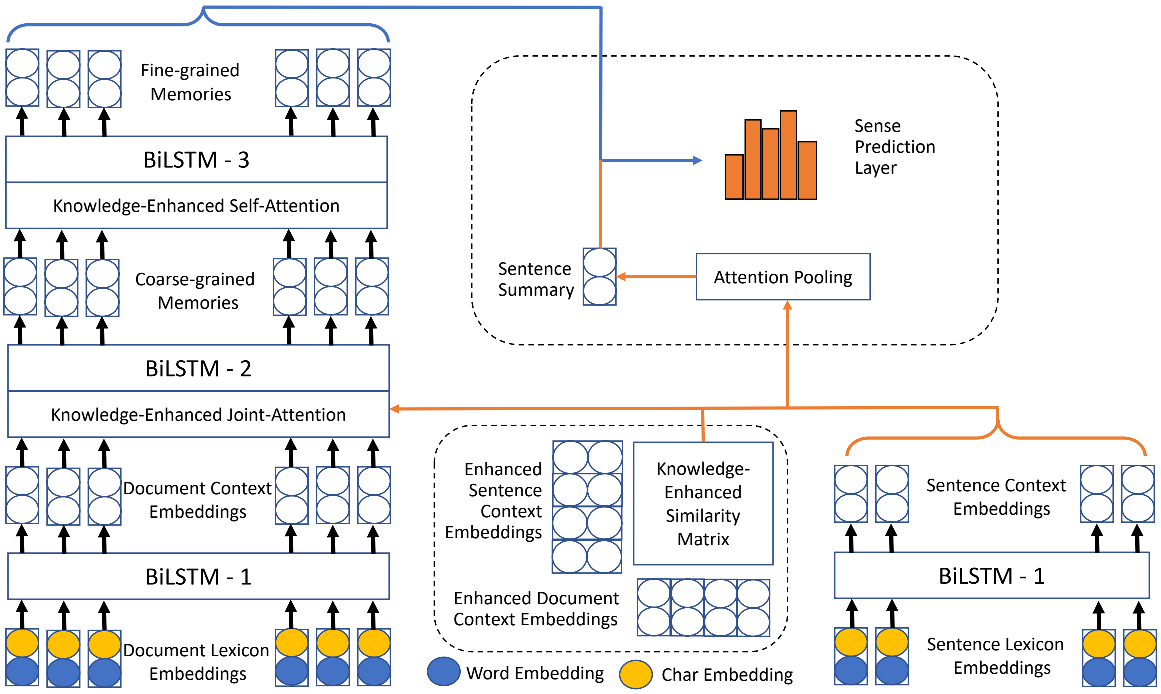
Fine-grained Memories
BiLSTM - 3
Knowledge-Enhanced Self-Attention
Coarse-grained Memories
BiLSTM - 2
Knowledge-Enhanced Joint-Attention
Document Context Embeddings
BiLSTM - 1
Document Lexicon Embeddings
Sense Prediction Layer
Sentence Summary
Attention Pooling
Enhanced Sentence Context Embeddings
Knowledge-Enhanced Similarity Matrix
Enhanced Document Context Embeddings
Sentence Context Embeddings
BiLSTM - 1
Sentence Lexicon Embeddings
● Word Embedding
● Char Embedding

Figure 4. Our proposed end-to-end supervised WSD model (KED).

is $\frac{1}{2}d$. By concatenating the forward LSTM outputs and the backward LSTM outputs, we obtain the fine-grained memories $H \in \mathbb{R}^{d \times n}$, which are the final document representations.

- Based on the fine-grained memories and the sentence context embeddings, the *sense prediction layer* generates the sense prediction. We first perform attention pooling on C_S to obtain a summary of the sentence:

$$\epsilon = C_S \mathrm{softmax}(\tanh(W_\epsilon C_S)^\top v_\epsilon) \in \mathbb{R}^{2d} \tag{3}$$

where W_ϵ and v_ϵ are trainable parameters. Then, with ϵ as a query, we compute a posterior distribution of all sense candidates in the list:

$$\mathscr{o} = \mathrm{softmax}(\tanh(W_\eta \epsilon H))^\top v_\eta) \in \mathbb{R}^n \tag{4}$$

where W_η and v_η are trainable parameters. Thus, for the training, we minimize $-\log(\mathscr{o}, \mathscr{t})$ on each training sample whose labeled sense is $\mathscr{t}$. For the inference, we take the index of the maximum element as $\mathscr{o}$.

3.2.3. Knowledge-Enhanced Joint-Attention

As a part of the coarse-grained memory layer, knowledge-enhanced joint-attention is aimed at fusing the sentence context embeddings C_S into the document context embeddings C_D, where the key problem is to calculate the similarity between each document context embedding c_{d_i} (i.e., the i-th column in C_D) and each sentence context embedding c_{s_j} (i.e., the j-th column in C_S). To solve this problem, we follow (Wang & Jiang, 2019) to incorporate a similarity function:

$$f(c_{d_i}, c_{s_j}) = v_f^\top [c_{d_i}; c_{s_j}; c_{d_i} \odot c_{s_j}] \in \mathbb{R}) \tag{5}$$

where v_f is a trainable parameter; $\odot$ represents element-wise multiplication. Since context embeddings contain high-level information, we believe that introducing the pre-extracted general knowledge into the calculation of such similarities will enhance the ability of the model to identify the boundaries of word senses and is helpful for the final disambiguation. Therefore, we use the pre-extracted general knowledge to construct the enhanced context embeddings. Specifically, for each word w, whose context embedding is c_w, to construct its enhanced context embedding $\tilde{c}_w$, first recall that we have extracted a set E_w, which includes the positions of the document words that w is semantically connected to, thus by gathering the columns in C_D whose indexes are given by E_w, we obtain the matching context embeddings $Z \in \mathbb{R}^{d \times |E_w|}$. Then by constructing a c_w-attentive summary of Z, we obtain the matching vector c_w^+ (if $E_w = \emptyset$, which makes $Z = \{\}$, we will set $c_w^+ = 0$):

$$t_i = v_c^\top \tanh(W_c z_i + U_c c_w) \in \mathbb{R} \tag{6}$$

$$c_w^+ = Z \;\mathrm{softmax}(\{t_1, ..., t_{|E_w|}\}) \in \mathbb{R}^d \tag{7}$$

where v_c, W_c, and U_c are trainable parameters; z_i represents the i-th column in Z. Finally we pass the concatenation of

c_w and c_w^+ through a dense layer with ReLU activation, whose output dimensionality is d. Therefore we obtain the enhanced context embedding $\tilde{c}_w \in \mathbb{R}^d$.

Based on this context embeddings, to perform knowledge-enhanced joint-attention, first we construct a knowledge-enhanced similarity matrix $A \in \mathbb{R}^{n\times m}$, where each element $A_{i,j} = f(\tilde{c}_{d_i}, \tilde{c}_{s_j})$. Then we construct the document-attentive sentence summaries R_S and the sentence-attentive document summaries R_D :

$$R_S = C_S \text{ softmax}_r^\top(A) \in \mathbb{R}^{d\times n} \tag{8}$$

$$R_D = C_D \text{ softmax}_c(A) \text{ softmax}_r^\top(A) \in \mathbb{R}^{d\times n} \tag{9}$$

where softmax_r represents softmax along the row dimension and softmax_c along the column dimension. Finally we pass the concatenation of C_D, R_S, $C_D \odot R_S$, and $R_D \odot R_S$ through a dense layer with ReLU activation, whose output dimensionality is d. Thus, we obtain the outputs $\tilde{G} \in \mathbb{R}^{d\times n}$.

3.2.4. Knowledge-Enhanced Self-Attention

As a part of the fine-grained memory layer, knowledge-enhanced self-attention is aimed at fusing the coarse-grained memories G into themselves. We use the pre-extracted general knowledge to guarantee that the fusion of coarse-grained memories for each document word will only involve a precise subset of the other document words. Specifically, for each document word d_i, whose coarse-grained memory is g_{d_i} (i.e., the i-th column in G), to perform the fusion of coarse-grained memories, first recall that we have extracted a set E_{d_i}, which includes the positions of the other document words that d_i is semantically connected to, thus by gathering the columns in G whose indexes are given by E_{d_i}, we obtain the matching coarse-grained memories $Z \in \mathbb{R}^{d\times |E_{d_i}|}$. Then by constructing a g_{d_i}-attentive summary of Z, we obtain the matching vector $g_{d_i}^+$ (if $E_{d_i} = \emptyset$, which makes $Z = \{\}$, we will set $g_{d_i}^+ = 0$):

$$t_i = v_g^\top \tanh(W_g z_i + U_g g_{d_i}) \in \mathbb{R} \tag{10}$$

$$g_{d_i}^+ = Z \text{ softmax}(\{t_1, ..., t_{|E_{d_i}|}\}) \in \mathbb{R}^d \tag{11}$$

where v_g, W_g, and U_g are trainable parameters. Finally we pass the concatenation of g_{d_i} and $g_{d_i}^+$ through a dense layer with ReLU activation, whose output dimensionality is d. Therefore we obtain the fusion result $\tilde{h}_{d_i} \in \mathbb{R}^d$, and further the outputs $\tilde{H} = \{\tilde{h}_{d_1}, ..., \tilde{h}_{d_n}\} \in \mathbb{R}^{d\times n}$.

Our proposed neural model is quite different from the existing WSD models in that it uses semantic level inter-word connections, which are pre-extracted from the WSD dataset using the WordNet-based data enrichment method, as explicit knowledge to assist the sense prediction of the target word. On one hand, the coarse-grained memory layer uses the explicit knowledge to assist both the document-to-sentence and sentence-to-document attentions. On the other hand, the fine-grained memory layer uses the explicit knowledge to assist the self-attention.

4. Experiments and Analysis

We conducted experiments on standard benchmark datasets introduced by (Raganato et al., 2017) (i.e., Senseval-2 (S2), Senseval-3 (S3), SemEval-2007 (SE07), SemEval-2013 (SE13) and SemEval-2015 (SE15)), in order to evaluate the performance of our proposed semi-supervised neural system (PoKED).

Implementation. To train the proposed unsupervised position-wise orthogonal (PoNet) pseudo-language model, we used BooksCorpus (Zhu et al., 2015) and English Wikipedia as part of our pretraining data. In addition, we included Giga5 (Parker et al., 2011), ClueWeb 2012-B (extended from (Callan et al., 2009)), and Common Crawl (Crawl) for pretraining. The vocabulary consists of the most frequent 1M words without lemmatization or case normalization. The dimension of word embedding was set to 128. The position-wise codes led to a dimension of 1024 for the input layer of the orthogonal framework. Then we appended three hidden layers of dimension 2048. Additionally, we chose constant forgetting factors $\alpha = (0.5, 0.9)$ for the position-wise codes.

To implement our proposed supervised neural model, we exploited the Stanford CoreNLP (Manning et al., 2014) to pre-process datasets. We used the WordNet interface provided by NLTK (BIRD & LOPER, 2004) to perform the WordNet-based data enrichment method. Additionally, we implemented a knowledge-based attentive neural model using Tensorflow (Abadi et al., 2016) and train it on SemCor (Miller et al., 1994) corpus. For each BiLSTM, we set its hidden state size to 256. For the training, we used ADAM (Kingma & Ba, 2014) as our optimizer, set the learning rate to 0.001, and set the mini-batch size to 32. To avoid overfitting, we applied Dropout (Srivastava et al., 2014) to dense layers and BiLSTMs with a value of 0.35. Besides, we applied an exponential moving average with a decay rate of 0.999.

Following (Raganato et al., 2017), (Luo et al., 2018) and (Hadiwinoto et al., 2019), we chose SE07, the smallest among these test sets, as the development set. When fine-tuning, we used the development set to find the optimal settings for our experiments. The reported WSD task results in F1-score are averaged over three runs.

Experimental results. We performed a comparison with three configurations of our model, namely: PoKED$_\alpha$, obtained using our pre-trained position-wise orthogonal (PoNet) context embeddings with general knowledge; PoKED$_\beta$, obtained using Roberta (Liu et al., 2019) and general knowledge; PoKED$_\gamma$, obtained using XLNet (Yang et al., 2019) and general knowledge.

As comparison systems we included four semi-supervised approaches mentioned above, namely: Babelfy (Moro et al., 2014), ppr$_{\text{w2w}}$, the best configuration of UKB (Agirre et al., 2018), WSD-TM (Chaplot & Salakhutdinov, 2018), and WSDG (Tripodi & Navigli, 2019). In addition, we also report the performances of relevant supervised models, namely: IMS (Zhong & Ng, 2010), IMS$_{\text{w2v}}$ (Iacobacci et al., 2016), Yuan$_{\text{LSTM}}$ (Yuan et al., 2016), Raganato$_{\text{BLSTM}}$ (Raganato et al., 2017), GAS (Luo et al., 2018), fastSense (Uslu et al., 2018), GLU-LW (Hadiwinoto et al., 2019) and GlossBERT (Huang et al., 2019).

The results of our evaluation are shown in Table 1. As we can see our models achieve state-of-the-art performances on four datasets, (i.e., S2, S3, SE07, SE15). It is worth noting that, on SE13 and SE15 datasets, ours perform better than most supervised systems. In general, the gap between supervised and semi-supervised systems is narrowed. This encourages new research in this direction. Our models perform particularly well on the disambiguation of *nouns* and *adjectives*.

Effect of general knowledge extraction. We obtain seven enriched WSD datasets by setting τ from zero to six separately, and train a different PoKED$_\alpha$ system on each enriched WSD dataset. As shown in Table 2, by increasing τ from zero to six in the data enrichment method, the amount of general knowledge rises monotonically, but the F1-score of our proposed PoKED$_\alpha$ first rises until τ reaches three for SE07 and SE15 datasets, and reaches four for S2, S3 and SE13 datasets, respectively, and then drops. We can conclude that the explicit knowledge provided by the WordNet-based data enrichment method plays an effective role in the training of our proposed PoKED$_\alpha$ system.

Effect of knowledge-enhancement. We conducted an ablation study and obtained a version without knowledge-enhancement based on PoKED$_\alpha$. Specifically, we replace knowledge-enhanced joint-attention and knowledge-enhanced self-attention with full bipartite attention and standard dot-scale attention, respectively. As shown in Table 3, we observe that the performance of the model without knowledge-enhancement drops dramatically compared with that of PoKED$_\alpha$, especially with the SE15 dataset. In this case, F1-score declined by 5.4%, 70.5 vs. 65.1. It indicates that human semantic knowledge is helping with WSD tasks remarkably, and plays an important role in our semi-supervised neural WSD system.

Quantitative analysis of the hunger for data. Specifically, instead of using all the training examples, we produce several training subsets (i.e., subsets of the training examples) so as to study the relationship between the proportion of the available training samples and the performance. We produce each training subset by sampling a specific number of sentences from all the sentences relevant to each document. By separately sampling 1, 2, 3, and 4 sentences on each document, we obtain four training subsets, which separately contain 20%, 40%, 60%, and 80% of the training samples. As shown in Figure 5, with PoKED$_\alpha$ trained on these training subsets, we evaluate its performance on the SE15, S3 and ALL, and find that PoKED$_\alpha$ performs much better than MFS baseline even when only 80% of the training samples are used. That is, when only a subset of the training examples is available, PoKED$_\alpha$ is still comparable to the state-of-the-art WSD models.

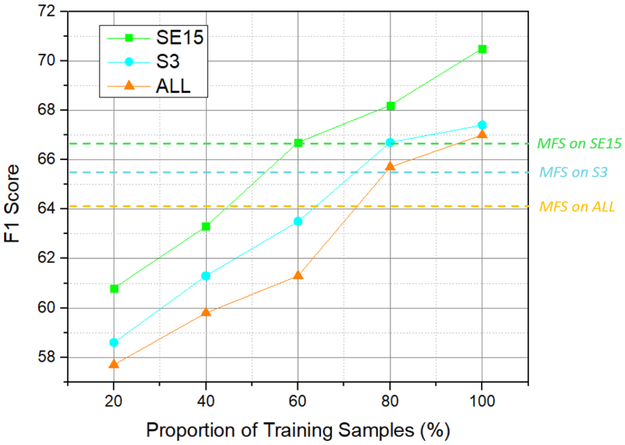

Figure 5. Quantitative Analysis of the Hunger for Data. With PoKED$_\alpha$ trained on the training subsets, we evaluate its performance on the SE15, S3 and ALL.

Discussion. According to the experimental results, PoKED not merely achieves state-of-the-art performances on most datasets, but performs better than most supervised systems. The reasons for these achievements, we believe, are as follows.

First, some inter-word semantic connections are distracting for the disambiguation of word sense. For example, the inter-word semantic connection between "ballpoint" and "pen" makes no sense given the context "Little John was looking for his toy box. Finally, he found it. The box was in the pen." It is the knowledge-enhanced attention mechanisms that enable PoKED to ignore such distracting inter-word semantic connections so that only the important ones are used.

Second, PoKED is designed to utilize the pre-extracted inter-word semantic connections extracted by the data enrichment method. Several approaches have been presented in recent years to make use of ontologies or types (such as (Dasigi et al., 2017)) to represent word tokens based on knowledge bases (e.g., WordNet). Nevertheless, the importance of inter-word semantic connections from document-sentence pairs

Table 1. Comparison with state-of-the-art algorithms: semi-supervised or knowledge-based (*semi-sup.*), and supervised (*sup.*). MFS refers to the Most Frequent Sense heuristic computed on SemCor (Miller et al., 1994) on each dataset. The results are provided as F1-score. **Bold** font indicates best systems. Results in the first and last blocks come from (Tripodi & Navigli, 2019; Hadiwinoto et al., 2019; Huang et al., 2019).

	Model	S2	S3	SE07	SE13	SE15	ALL	N	V	A	R
semi-sup.	*MFS*	64.7	65.4	53.9	62.9	66.6	64.1	68.1	49.5	74.1	80.6
	Babelfy	67.0	63.5	51.6	66.4	70.3	65.5	68.6	49.9	73.2	79.8
	ppr_{w2w}	68.8	66.1	53.0	**68.8**	70.3	67.3	-	-	-	-
	WSD-TM	69.0	66.9	55.6	65.3	69.6	66.9	69.7	51.2	76.0	**80.9**
	WSDG	68.9	65.5	54.5	67.0	**72.8**	67.2	70.4	51.3	75.7	80.6
	PoKED$_\alpha$ *(ours)*	68.8	**67.4**	54.6	65.2	70.5	67.0	70.1	50.8	75.5	79.7
	PoKED$_\beta$ *(ours)*	69.5	67.0	55.8	67.3	**72.8**	67.4	70.5	51.4	**76.2**	80.8
	PoKED$_\gamma$ *(ours)*	**69.8**	67.1	**56.0**	67.5	72.7	**67.7**	**70.8**	**51.6**	**76.2**	80.6
sup.	*IMS*	70.9	69.3	61.3	65.3	69.5	68.9	70.5	55.8	75.6	82.9
	IMS_{w2v}	72.2	70.4	62.6	65.9	71.5	70.1	71.9	56.6	75.9	84.7
	$Yuan_{LSTM}$	73.8	71.8	63.5	69.5	72.6	71.5	-	-	-	-
	$Raganato_{BLSTM}$	72.0	69.1	64.8	66.9	71.5	69.9	71.5	57.5	75.0	83.8
	GAS	72.2	70.5	-	67.2	72.6	-	-	-	-	-
	fastSense	73.5	73.5	62.4	66.2	73.2	-	-	-	-	-
	GLU-LW	75.5	73.4	68.5	71.0	76.2	-	-	-	-	-
	GlossBERT	76.5	73.4	69.2	75.1	79.5	-	-	-	-	-

Table 2. Effect analysis of general knowledge extraction. We report the F1-score performance of PoKED$_\alpha$ on standard benchmarks under each setting for τ. **#average** stands for average number of inter-word connections per word. **Bold** font indicates best performance.

τ	#average	S2	#average	S3	#average	SE07	#average	SE13	#average	SE15
0	0.51	67.1	0.47	65.6	0.30	52.7	0.29	63.2	0.36	67.1
1	0.92	67.7	0.84	66.2	0.51	53.4	0.41	63.6	0.68	68.4
2	1.47	68.1	1.32	66.5	1.93	53.9	0.93	64.1	2.03	69.6
3	2.35	68.4	3.06	66.9	**3.28**	**54.6**	1.58	64.7	**3.97**	**70.5**
4	**3.89**	**68.8**	**3.97**	**67.4**	3.86	54.0	**3.67**	**65.2**	4.87	69.9
5	5.79	68.3	4.28	66.8	4.77	53.2	4.22	64.5	5.52	69.3
6	6.22	68.0	5.45	66.4	6.02	52.6	5.19	63.4	6.45	68.2

Table 3. Effect of knowledge-enhancement. It shows human semantic knowledge plays an important role in our semi-supervised neural WSD system.

Model	S2	S3	SE07	SE13	SE15
PoKED$_\alpha$	68.8	67.4	54.6	65.2	70.5
- knowledge-enhancement	64.3 ▼-4.5	63.5 ▼-3.9	50.2 ▼-4.4	61.4 ▼-3.8	65.1 ▼-5.4

was overlooked. Therefore, in this paper, we explore an explicit (i.e., understandable and controllable) way to utilize general knowledge. Some inter-word semantic connections, especially those obtained through multi-hop semantic relation chains, enhance the ability of the model to identify the boundaries of word senses and is helpful for the final disambiguation.

Finally, an inter-word semantic connection extracted from a document-sentence pair usually also appears in many other document-sentence pairs; therefore it is very likely that the inter-word semantic connections extracted from a small number of training examples cover a larger amount of training examples. That is, we are using more training examples for model optimization than the available ones.

5. Conclusion

In this paper, we propose a semi-supervised neural system (PoKED), which incorporates human semantic knowledge into the neural network architectures for WSD tasks. Specifically, inter-word semantic connections are first extracted from each given document by a WordNet-based data enrichment method, and then provided as general knowledge to an end-to-end WSD model, which explicitly uses the general knowledge to assist its attention mechanisms. Experimental results show that PoKED achieves better performance than the previous state-of-the-art knowledge-based models.

Acknowledgements

I would like to thank my thesis advisor Professor Uyen Trang Nguyen for her tremendous help and many fruitful discussions. I also thank anonymous reviewers for their thorough constructive comments to improve the paper. Many of my past and present collaborators have made valuable contributions to this work.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 265–283, 2016.

Agirre, E., de Lacalle, O. L., and Soroa, A. The risk of sub-optimal use of open source nlp software: Ukb is inadvertently state-of-the-art in knowledge-based wsd. In *Proceedings of Workshop for NLP Open Source Software*, pp. 29–33, 2018.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

BIRD, S. and LOPER, E. Nltk: The natural language toolkit. Association for Computational Linguistics, 2004.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3: 993–1022, 2003.

Callan, J., Hoy, M., Yoo, C., and Zhao, L. Clueweb09 data set, 2009.

Chaplot, D. S. and Salakhutdinov, R. Knowledge-based word sense disambiguation using topic models. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Crawl, C. Common crawl. http://commoncrawl.org.

Dasigi, P., Ammar, W., Dyer, C., and Hovy, E. Ontology-aware token embeddings for prepositional phrase attachment. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2089–2098, 2017.

Hadiwinoto, C., Ng, H. T., and Gan, W. C. Improved word sense disambiguation using pre-trained contextualized word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5300–5309, 2019.

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

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

Huang, L., Sun, C., Qiu, X., and Huang, X. Glossbert: Bert for word sense disambiguation with gloss knowledge. *arXiv preprint arXiv:1908.07245*, 2019.

Iacobacci, I., Pilehvar, M. T., and Navigli, R. 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)*, pp. 897–907, 2016.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 427–431, 2017.

Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

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

Le, M., Postma, M., Urbani, J., and Vossen, P. Deep dive into word sense disambiguation with lstm. In *Proceedings of 27th International Conference on Computational Linguistics*, pp. 354–365, 2018.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

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

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, 2014.

Maru, M., Scozzafava, F., Martelli, F., and Navigli, R. Syntagnet: Challenging supervised word sense disambiguation with lexical-semantic combinations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3525–3531, 2019.

Mihalcea, R., Tarau, P., and Figa, E. Pagerank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20th International Conference on Computational Linguistics*, pp. 1126–1132, 2004.

Miller, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

Miller, G. A., Chodorow, M., Landes, S., Leacock, C., and Thomas, R. G. Using a semantic concordance for sense identification. In *Proceedings of the Workshop on Human Language Technology*, pp. 240–243, 1994.

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

Navigli, R. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10, 2009.

Navigli, R. and Lapata, M. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):678–692, 2009.

Page, L., Brin, S., Motwani, R., and Winograd, T. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Parker, R., Graff, D., Kong, J., Chen, K., and Maeda, K. English gigaword fifth edition. *Linguistic Data Consortium*, 2011.

Raganato, A., Camacho-Collados, J., and Navigli, R. 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*, pp. 99–110, 2017.

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

Tripodi, R. and Navigli, R. Game theory meets embeddings: a unified framework for word sense disambiguation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 88–99, 2019.

Uslu, T., Mehler, A., Baumartz, D., and Hemati, W. fastsense: An efficient word sense disambiguation classifier. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, 2018.

Wang, C. and Jiang, H. Explicit utilization of general knowledge in machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2263–2272, 2019.

Watcharawittayakul, S., Xu, M., and Jiang, H. Dual fixed-size ordinally forgetting encoding (fofe) for competitive neural language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4725–4730, 2018.

Wei, F., Nguyen, U. T., and Jiang, H. Dual-fofe-net neural models for entity linking with pagerank. In *International Conference on Artificial Neural Networks*, pp. 635–645. Springer, 2019.

Wei, F., Trang Nguyen, U., and Jiang, H. Fonet: A memory-efficient fourier-based orthogonal network for object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 688–689, 2020.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, pp. 5754–5764, 2019.

Yuan, D., Richardson, J., Doherty, R., Evans, C., and Altendorf, E. Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1374–1385, 2016.

Zhang, S., Jiang, H., Xu, M., Hou, J., and Dai, L. The fixed-size ordinally-forgetting encoding method for neural network language models. In *Proceedings of ACL*, 2015.

Zhang, S., Jiang, H., and Dai, L. Hybrid orthogonal projection and estimation (hope): a new framework to learn neural networks. *The Journal of Machine Learning Research*, 17(1):1286–1318, 2016.

Zhong, Z. and Ng, H. T. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010: System Demonstrations*, pp. 78–83, 2010.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 19–27, 2015.