Unsupervised Deep Structured Semantic Models for Commonsense Reasoning

Shuohang Wang[1]*, **Sheng Zhang**[2], **Yelong Shen**[4], **Xiaodong Liu**[3],
Jingjing Liu[3], **Jianfeng Gao**[3], **Jing Jiang**[1]
[1]Singapore Management University,[2]Johns Hopkins University, [3]Microsoft, [4]Tencent AI Lab
{shwang.2014,jingjiang}@smu.edu.sg, zsheng2@jhu.edu
{xiaodl,jingjl,jfgao}@microsoft.com, yelongshen@tencent.com

Abstract

Commonsense reasoning is fundamental to natural language understanding. While traditional methods rely heavily on human-crafted features and knowledge bases, we explore learning commonsense knowledge from a large amount of raw text via unsupervised learning. We propose two neural network models based on the Deep Structured Semantic Models (DSSM) framework to tackle two classic commonsense reasoning tasks, Winograd Schema challenges (WSC) and Pronoun Disambiguation (PDP). Evaluation shows that the proposed models effectively capture contextual information in the sentence and co-reference information between pronouns and nouns, and achieve significant improvement over previous state-of-the-art approaches.

1 Introduction

Commonsense reasoning is concerned with simulating the human ability to make presumptions about the type and essence of ordinary situations they encounter every day (Davis and Marcus, 2015). It is one of the key challenges in natural language understanding, and has drawn increasing attention in recent years (Levesque et al., 2011; Roemmele et al., 2011; Zhang et al., 2017; Rashkin et al., 2018a,b; Zellers et al., 2018; Trinh and Le, 2018). However, due to the lack of labeled training data or comprehensive hand-crafted knowledge bases, commonsense reasoning tasks such as Winograd Schema Challenge (Levesque et al., 2011) are still far from being solved.

In this work, we propose two effective unsupervised models for commonsense reasoning, and evaluate them on two classic commonsense reasoning tasks: Winograd Schema Challenge (WSC) and Pronoun Disambiguation Problems (PDP). Compared to other commonsense reasoning tasks, WSC and PDP better approximate real human reasoning, and can be more easily solved by native English-speaking adults (Levesque et al., 2011). In addition, they are also technically challenging. For example, the best reported result on WSC is only 20 percentage points better than random guess in accuracy (Radford et al., 2019).

1. *The city councilmen refused the demonstrators a permit because* ***they*** *feared violence.*

   Who feared violence?

   A. **The city councilmen** B. The demonstrators

2. *The city councilmen refused the demonstrators a permit because* ***they*** *advocated violence.*

   Who advocated violence?

   A. The city councilmen B. **The demonstrators**

Table 1: Examples from Winograd Schema Challenge (WSC). The task is to identify the reference of the pronoun in bold.

Table 1 shows two examples from WSC. In order to resolve the co-reference in these two examples, one cannot predict what "***they***" refers to unless she is equipped with the commonsense knowledge that "*demonstrators usually cause violence and city councilmen usually fear violence*".

As no labeled training data is available for these tasks, previous approaches are based on either hand-crafted knowledge bases or large-scale language models. For example, Liu et al. (2017) used existing knowledge bases such as ConceptNet (Liu and Singh, 2004) and WordNet (Miller, 1995) for external supervision to train word embeddings and solve the WSC challenge. Recently, Trinh and Le (2018) first used raw text from books/news to train a neural Language Model (LM), and then em-

[*] Work done when the author was at Microsoft

Proceedings of NAACL-HLT 2019, pages 882–891
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

ployed the trained model to compare the probabilities of the sequences, where the pronouns are replaced by each of the candidate references, and to pick the candidate that leads to the highest probability as the answer.

Because none of the existing hand-crafted knowledge bases is comprehensive enough to cover all the world knowledge[1], we focus on building unsupervised models that can learn commonsense knowledge directly from unlimited raw text. Different from the neural language models, our models are optimized for co-reference resolution and achieve much better results on both the PDP and WSC tasks.

In this work we formulate the two commonsense reasoning tasks in WSC and PDP as a pairwise ranking problem. As the first example in Table 1, we want to develop a pair-wise scoring model $\text{Score}_\theta(x_i, y)$ that scores the correct antecedent-pronoun pair ("*councilmen*", "*they*") higher than the incorrect one ("*demonstrators*", "*they*"). These scores depend to a large degree upon the contextual information of the pronoun (e.g., they) and the candidate antecedent (e.g., councilmen). In other words, it requires to capture the semantic meaning of the pronoun and the candidate antecedent based on the sentences where they occur, respectively.

To tackle this issue, we propose two models based on the framework of Deep Structured Similarity Model (DSSM) (Huang et al., 2013), as shown in Figure 1(a). Formally, let S^x be the sentence containing the candidate antecedent x_i and S^y the sentence containing the pronoun y which we're interested in. DSSM measures the semantic similarity of a pair of inputs (x_i, y) by 1) mapping x_i and y, together with their context information, into two vectors in a semantic space using deep neural networks f_1 and f_2, parameterized by θ; and 2) computing cosine similarity[2] between them. In our case, we need to learn a task-specific semantic space where the distance between two vectors measures how likely they co-refer. Commonsense knowledge such as "demonstrators usually cause violence" can be implicitly captured in the semantic space through DSSM, which is trained on a large amount of raw text.

DSSM requires labeled pairs for training.Since there is no labeled data for our tasks, we propose two unsupervised DSSMs, or UDSSMs. As shown in Figure 1(b) and 1(c), $(\mathbf{S}^x, \mathbf{S}^y)$ are encoded into contextual representations by deep neural networks f_1 and f_2; then we compute pair-wise their co-reference scores.

In what follows, we will describe two assumptions we propose to harvest training data from raw text. **Assumption I: A pronoun refers to one of its preceding nouns in the same sentence.** The sentences generated by this assumption will be used for training UDSSM-I. Some examples will be shown in the "data generation" section. **Assumption II: In a sentence, pronouns of the same gender and plurality are more likely to refer to the same antecedent than other pronouns.** Similarly, the sentences following the assumption will be used for training UDSSM-II.

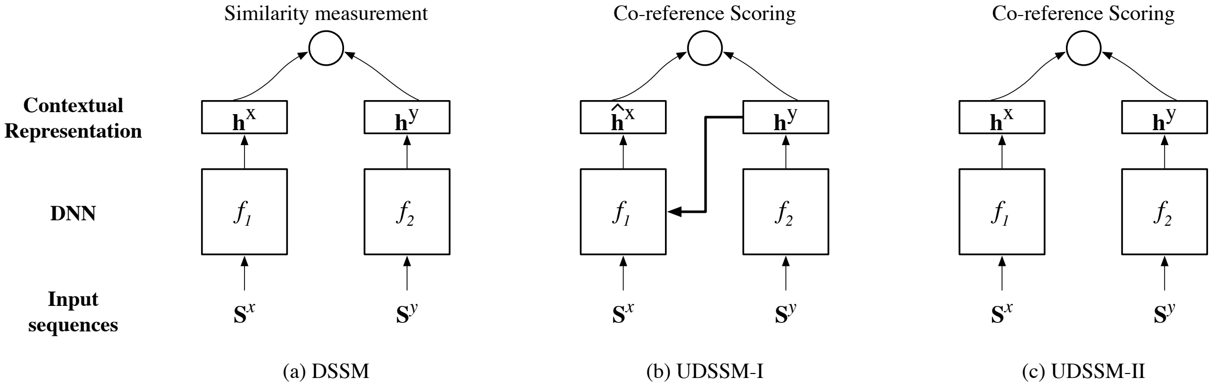
Note that the two models, UDSSM-I and UDSSM-II are trained on different types of pairwise training data, thus the model structures are different, as illustrated in Figure 1(b) and 1(c), respectively. Experiments demonstrated that our methods outperform stat-of-the-art performance on the tasks of WSC and PDP.

2 Related Work

As a key component of natural language understanding, commonsense reasoning has been included in an increasing number of tasks for evaluation: COPA (Roemmele et al., 2011) assesses commonsense causal reasoning by selecting an alternative, which has a more plausible causal relation with the given premise. Story Cloze Test (ROCStories, Mostafazadeh et al. 2016) evaluates story understanding, story generation, and script learning by choosing the most sensible ending to a short story. JOCI (Zhang et al., 2017) generalizes the natural language inference (NLI) framework (Cooper et al., 1996; Dagan et al., 2006; Bowman et al., 2015; Williams et al., 2018) and evaluates commonsense inference by predicting the ordinal likelihood of a hypothesis given a context. Event2Mind (Rashkin et al., 2018b) models stereotypical intents and reactions of people, described in short free-form text. SWAG (Zellers et al., 2018) frames commonsense inference as multiple-choice questions for follow-up events given some context. ReCoRD (Zhang et al., 2018)

[1] We don't believe it is possible to construct such a knowledge base given that the world is changing constantly.

[2] DSSMs can be applied to a wide range of tasks depending on the definition of (x, y). For example, (x, y) is a query-document pair for Web search ranking, a document pair in recommendation, a question-answer pair in QA, and so on. See Chapter 2 of (Gao et al., 2018) for a survey.



(a) DSSM (b) UDSSM-I (c) UDSSM-II

Figure 1: An overview of (a) the general framework of Deep Structured Semantic Model (DSSM) and our two unsupervised models based on DSSM: (b) UDSSM-I and (c) UDSSM-II. Compared with DSSM, both UDSSM-I and UDSSM-II compute co-reference scores instead of similarity.

evaluates a machine's ability of commonsense reasoning in reading comprehension.

Among all these commonsense reasoning tasks, the Winograd Schema Challenge (WSC) and Pronoun Disambiguation Problems (PDP) (Levesque et al., 2011) are known as the most challenging tasks for commonsense reasoning. Although both tasks are based on pronoun disambiguation, a subtask of coreference resolution (Soon et al., 2001; Ng and Cardie, 2002; Peng et al., 2016), PDP and WSC differ from normal pronoun disambiguation due to their unique properties, which are based on commonsense, selecting the most likely antecedent from both candidates in the directly preceding context.

Previous efforts on solving the Winograd Schema Challenge and Pronoun Disambiguation Problems mostly rely on human-labeled data, sophisticated rules, hand-crafted features, or external knowledge bases (Peng et al., 2015; Bailey et al., 2015; Schüller, 2014). Rahman and Ng (2012) hired workers to annotate supervised training data and designed 70K hand-crafted features. Sharma et al. (2015); Schüller (2014); Bailey et al. (2015); Liu et al. (2017) utilized expensive knowledge bases in their reasoning processes. Recently, Trinh and Le 2018 applied neural language models trained with a massive amount of unlabeled data to the Winograd Schema Challenge and improved the performance by a large margin. In contrast, our unsupervised method based on DSSM significantly outperforms the previous state-of-the-art method, with the advantage of capturing more contextual information in the data.

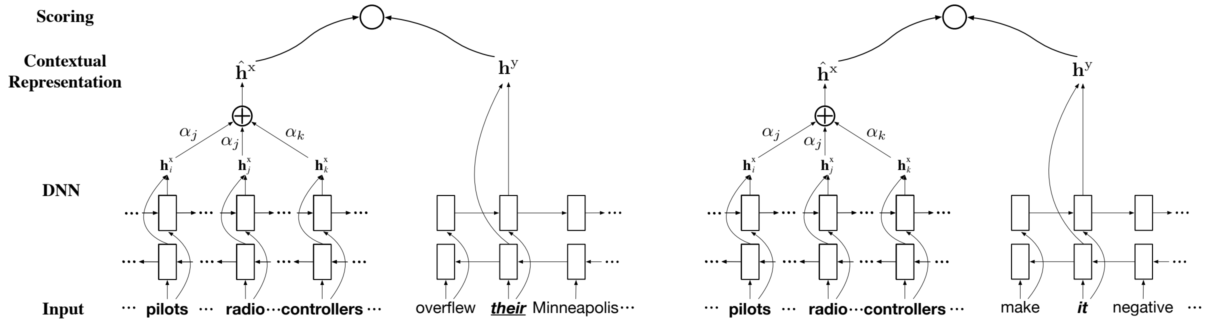
3 Approach

As shown in Figure 1, we propose two unsupervised deep structured semantic models (**UDSSM-I** and **UDSSM-II**), which consist of two components: DNN encoding and co-reference scoring. For the model UDSSM-I, the co-referred word pairs are automatically learned through an attention mechanism, where the attention weights are the co-reference scores for word pairs. For the second model UDSSM-II, we will directly optimize the co-reference score during training. After all, we will get the co-reference scoring function, $\text{Score}_\theta(x_i, y)$, to compare the candidate answers in the tasks of PDP/WSC. Next, we will show the details of our models trained in an unsupervised way.

In the following sections, we will use uppercase symbols in bold, e.g., $\mathbf{S}^x$, to represent matrices. Lowercase symbols in bold, e.g., $\mathbf{h}^x$, represent vectors. A regular uppercase symbol, e.g., S^x, represents a lexical sequence. A regular lowercase symbol, e.g., x_i or y, represents a token.

3.1 UDSSM-I Model

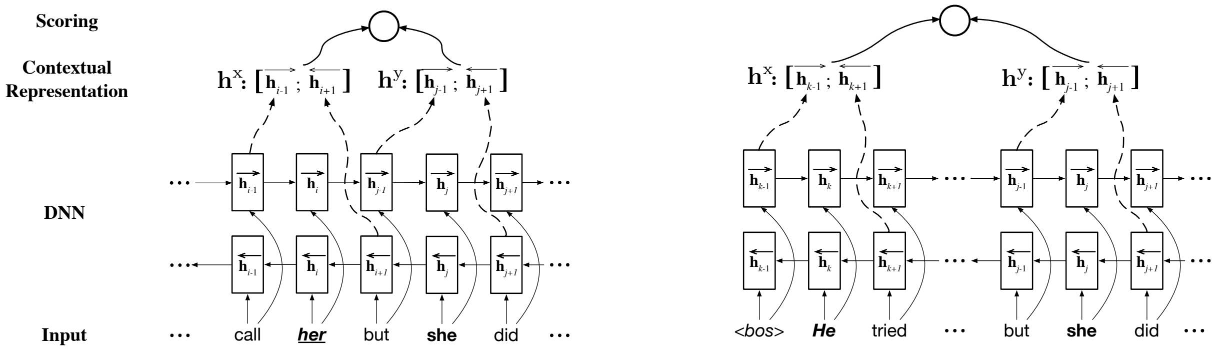
This model is developed based on Assumption I. Its architecture is shown in Figure 2. The sentences generated based on this assumption contain a pronoun y and a set of its preceding nouns $\{x_i, x_j ...\}$, which includes the referred word by pronoun. For example, the sentence in Figure 2. As there is no clear label for the co-referred word pairs under this assumption, our model will rank the set of nouns $\{x_i, x_j ...\}$ which contains the noun that the pronoun y refers to higher than the set which does not. And the co-reference score between words will not be optimized directly during


Scoring
Contextual Representation
DNN
Input
ĥx
hy
αj
αk
hxi
hxj
hxk
… pilots … radio…controllers …
overflew their Minneapolis…
make it negative …

(a) Positive example

(b) Negative example

Figure 2: The procedure of using UDSSM-I to compute the co-reference scores of a positive example and a negative example respectively. The positive example is generated from the sentence '*Two **Northwest Airlines pilots** failed to make **radio contact** with **ground controllers** for more than an hour and overflew **their** Minneapolis destination by 150 miles before discovering the mistake and turning around.*". The negative one replaces the second sequence with one sequence from different sentence.



(a) Positive example

(b) Negative example

Figure 3: The procedure of using UDSSM-II to compute the co-reference scores of a positive example and a negative example respectively. Both examples are generated from the sentence "*He tried twice to call her but she did not answer the phone*".

training, but is learned indirectly through the attention mechanism. We will describe in turn how the training data is generated from raw text, the model architecture, and the co-reference scoring function for the final prediction on the tasks of PDP/WSC.

3.1.1 Data Generation

The main challenge of PDP/WSC tasks is that it has no labeled training data. Here we introduce a simple method to collect unsupervised training data by leveraging some linguistic patterns. Following Assumption 1, the first hypothesis we make is that "the pronoun refers to one of the preceding nouns", which is a common phenomenon in well-written stories or news. In this way, we generate (S^{x}, S^{y}) pairs from raw text as follows:

- Parse the sentences in the raw text to obtain entity names, nouns and pronouns.
- Pick sentences that contain at least one pronoun and multiple nouns preceding it.
- Split each sentence into two sub-sentences to form a positive pair (S^{x}, S^{y}), where S^{x} is the first sub-sentence with identified nouns and entity names while S^{y} is the second sub-sentence with a pronoun.
- One or more negative pairs are generated from (S^{x}, S^{y}) by replacing S^{y} with $S^{\mathrm{y}_{neg}}$ randomly sampled from other positive pairs.

We split the sentence with pronouns and nouns into two sub-sequences separated by the previous word of the pronoun. Therefore, the example sentence in the Figure 2 can be split into two sub-sentences as shown below:

- S^{x}: " **Two Northwest Airlines pilots** failed to make **radio contact** with **ground controllers** for more than an hour and"

- S^{y}: "overflew **their** Minneapolis destination by 150 miles before discovering the mistake and turning around".

As the sentences are collected from raw text, the co-reference words are not given. Our proposed UDSSM-I model will learn the co-reference scoring function through attention mechanism based on the generated sequence pairs. Next, we will introduce the details of this model.

3.1.2 Model Architecture

This method takes the pair of sequences, (S^{x}, S^{y}), as inputs, and computes similarity between the sequences collected from the same sentence. As we hypothesize that one of the nouns in the first sequence and the pronoun in the second are co-referred, we only use the contextual representations of nouns and pronoun to represent the sequences. To obtain the contextual representation, we first use a bi-directional LSTM to process these sequences [3]:

$$\mathbf{H}^{\mathrm{x}} = \text{Bi-LSTM}(\mathbf{S}^{\mathrm{x}}), \mathbf{H}^{\mathrm{y}} = \text{Bi-LSTM}(\mathbf{S}^{\mathrm{y}}), \quad (1)$$

where $\mathbf{S}^{\mathrm{x}} \in \mathbb{R}^{d \times X}$, $\mathbf{S}^{\mathrm{y}} \in \mathbb{R}^{d \times Y}$ are the word embeddings of the two sequences. d is the dimension of the word embeddings. X, Y are the lengths of the two sequences. $\mathbf{H}^{\mathrm{x}} \in \mathbb{R}^{l \times X}$ and $\mathbf{H}^{\mathrm{y}} \in \mathbb{R}^{l \times Y}$ are the hidden states of bi-directional LSTM. Our model is task-specifically constructed, so we directly use the hidden state of the first pronoun in the second sequence as its representation:

$$f_2(S^y) \;=\; \mathbf{h}^{\mathrm{y}} \;=\; \mathbf{h}^{\mathrm{y}}_2, \qquad (2)$$

where $\mathbf{h}^{\mathrm{y}}_2 \in \mathbb{R}^{\mathrm{l}}$ is the second[4] vector from $\mathbf{H}^{\mathrm{y}}$ and it represents the contextual information of the pronoun. Next, we will get the representation of the first sequence. As there are multiple nouns in the first sequence and the pronoun usually refers to only one of them, we use the weighted sum of all the LSTM hidden states of the nouns to represent the sequence, $\hat{\mathbf{h}}^{\mathrm{x}} \in \mathbf{R}^{l}$, as follows:

$$\begin{aligned} \mathbf{H}^{\mathrm{n}} &= [\mathbf{h}^{\mathrm{x}}_i; \mathbf{h}^{\mathrm{x}}_j; ...] \\ \alpha &= \text{SoftMax}\left((\mathbf{W}^{\mathrm{g}}\mathbf{H}^{\mathrm{n}} + \mathbf{b}^{\mathrm{g}} \otimes \mathbf{e}_N)^{\mathrm{T}}\mathbf{h}^{\mathrm{y}}\right), \\ f_1(S^x) &= \hat{\mathbf{h}}^{\mathrm{x}} = \mathbf{H}^{\mathrm{n}}\alpha, \end{aligned} \qquad (3)$$

where $i, j...$ are the positions of the nouns in the sequence S^x and $[;]$ is the concatenation of two vectors. $\mathbf{H}^{\mathrm{n}} \in \mathbb{R}^{l \times N}$ are all the hidden states of the nouns[5] in $\mathbf{H}^{\mathrm{x}}$ in the sequence. N is the number of nouns in the sequence. $\alpha \in \mathbb{R}^{N}$ is the weights assigned for the different nouns and $\hat{\mathbf{h}}^{\mathrm{x}} \in \mathbb{R}^{l}$ is the weighted sum of all the hidden states of the nouns. $\mathbf{W}^{\mathrm{g}} \in \mathbb{R}^{l \times l}$ and $\mathbf{b}^{\mathrm{g}} \in \mathbb{R}^{l}$ are the parameters to learn; $\mathbf{e}_N \in \mathbb{R}^{\mathrm{N}}$ is a vector of all 1s and it is used to repeat the bias vector N times into the matrix. Then we will maximize the similarity of the contextual representations of ($\hat{\mathbf{h}}^{\mathrm{x}}, \mathbf{h}^{\mathrm{y}}$). Meanwhile, we also need some negative samples $\mathbf{h}^{\mathrm{y}_{neg}}_k$ for $\hat{\mathbf{h}}^{\mathrm{x}}$. Then our loss function for this method is constructed:

$$L = -\log\left(\frac{\exp\left(\hat{\mathbf{h}}^{\mathrm{x}}\mathbf{h}^{\mathrm{y}}\right)}{\exp\left(\hat{\mathbf{h}}^{\mathrm{x}}\mathbf{h}^{\mathrm{y}}\right) + \sum_k^K \exp\left(\hat{\mathbf{h}}^{\mathrm{x}}\mathbf{h}^{\mathrm{y}_{neg}}_k\right)}\right), \qquad (4)$$

where $\mathbf{h}^{\mathrm{y}_{neg}}_k \in \mathbb{R}^{l}$ is the randomly sampled hidden state of pronoun from the sequences not in the same sentence with S^y.

3.1.3 Co-reference Scoring Function

Overall, the model tries to make the co-reference states similar to each other. The co-reference scoring function is defined:

$$\text{Score}_\theta(x_i, y) = g(\mathbf{h}^{\mathrm{x}}_i, \mathbf{h}^{\mathrm{y}}) = (\mathbf{W}^{\mathrm{g}}\mathbf{h}^{\mathrm{x}}_i + \mathbf{b}^{\mathrm{g}})^{\mathrm{T}}\mathbf{h}^{\mathrm{y}}, \qquad (5)$$

where the candidate located at the i-th position is represented by its LSTM hidden state $\mathbf{h}^{\mathrm{x}}_i$ and the pronoun in the snippet is represented by $\mathbf{h}^{\mathrm{y}}$. And the output value of this function for each candidate will be used for the final prediction. Next, we will introduce the other unsupervised method.

3.2 UDSSM-II Model

This model is developed based on Assumption II. Its architecture is shown in Figure 3. As the model is similar to the previous one, we will introduce the details in a similar way.

3.2.1 Data Generation

The second assumption is that "the pronoun pairs in a single sentence are co-reference words if they are of the same gender and plurality; otherwise they are not." Based on this assumption, we can directly construct the co-reference training pairs as follows:

[3] We use two different LSTMs to process the sequences S^{x} and S^{Y} here. This is to make the negative sampling in Eqn. (4) more efficient, so that we can directly use the other representations in the same batch as negative ones.

[4] We assign the word just before the pronoun to the second sequence, so the pronoun always appears in the second position of the sequence.

[5] We use the toolkit of spaCy in Python for POS and NER, and we will remove the sequences that contain less than 2 nouns.

- Parse the raw sentences to identify pronouns.
- Pick sentences that contain at least two pronouns.
- The sub-sequence pair with pronouns of the same gender and plurality is labeled as a positive pair; otherwise it is labeled as negative.
- Replace the corresponding pronoun pairs with a special token "**@Ponoun**".

Take the following sentence as an example: "**He** tried twice to call **her** but **she** did not answer the phone." There are three pronouns detected in the sentence, and we assume that the words **her** and **she** are co-reference words, while pairs ($\mathbf{she}, \mathbf{He}$) and ($\mathbf{her}, \mathbf{He}$) are not. Thus we can obtain three training examples from the given sentence. However, in the PDP and WSC tasks, models are asked to compute the co-reference scores between pronoun and candidate nouns, instead of two pronouns. Therefore, we replace the first pronoun in the sentence with a place holder; i.e., a negative training pair is generated by splitting the raw sentence into the following two sub-sequences:

- S^{x}: " **@Ponoun** tried twice to call her"
- S^{y}: "but **she** did not answer the phone."
- label: Negative

and the positive training pair can be generated by the same way:

- S^{x}: " He tried twice to call **@Ponoun**"
- S^{y}: "but **she** did not answer the phone."
- label: Positive

Thus, we could directly train the encoder and co-reference scoring components through the generated training pairs.

3.2.2 Model Architecture

The previous method, UDSSM-I, follows the task setting of PDP/WSC, and builds the model based on the similarity of the representations between nouns and the pronoun. As there is no signal indicating the exact alignment between co-reference words, the model tries to learn it based on the co-occurrence information from large scale unlabelled corpus. For the method of UDSSM-II, each representation pair ($\mathbf{h}^{\mathrm{x}}, \mathbf{h}^{\mathrm{y}}$) has a clear signal, r, indicating whether they are co-referred or not. For simplicity, we do not have to split the sentence into two parts. We first use LSTM to process the sentence as follows:

$$\overrightarrow{\mathbf{H}} = \overrightarrow{\mathrm{LSTM}}([\mathbf{S}^{\mathrm{x}}; \mathbf{S}^{\mathrm{y}}]), \overleftarrow{\mathbf{H}} = \overleftarrow{\mathrm{LSTM}}([\mathbf{S}^{\mathrm{x}}; \mathbf{S}^{\mathrm{y}}]), \quad (6)$$

where we can concatenate the word embeddings, $[\mathbf{S}^{\mathrm{x}}; \mathbf{S}^{\mathrm{y}}]$, of two sequences collected under Assumption II. $\overrightarrow{\mathrm{LSTM}}$ and $\overleftarrow{\mathrm{LSTM}}$ are built in different directions, and $\overrightarrow{\mathbf{H}}$, $\overleftarrow{\mathbf{H}}$ are the hidden states of the corresponding LSTM. Suppose that the pronoun pair in the sentence are located at the i-th and j-th positions as shown in the bottom part of Figure 3(a). We use the hidden states around the pronouns as their contextual representations as follows:

$$f_1(S^x) = \mathbf{h}^{\mathrm{x}} = \begin{bmatrix} \overrightarrow{\mathbf{h}_{i-1}} \\ \overleftarrow{\mathbf{h}_{i+1}} \end{bmatrix}, f_2(S^y) = \mathbf{h}^{\mathrm{y}} = \begin{bmatrix} \overrightarrow{\mathbf{h}_{j-1}} \\ \overleftarrow{\mathbf{h}_{j+1}} \end{bmatrix}, \quad (7)$$

where $\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$ is the concatenation of all the vectors inside it. Then we further concatenate these representation pair:

$$\mathbf{h}^c = \begin{bmatrix} \mathbf{h}^{\mathrm{x}} \\ \mathbf{h}^{\mathrm{y}} \end{bmatrix}, \quad (8)$$

where $\mathbf{h}^c \in \mathbb{R}^{4l}$, and it will be the input of loss function with cross entropy as follows:

$$\begin{aligned} L &= -r \log \left(\frac{\exp(\mathbf{w}^p \mathbf{h}^c)}{\exp(\mathbf{w}^p \mathbf{h}^c) + \exp(\mathbf{w}^n \mathbf{h}^c)} \right) \\ &- (1-r) \log \left(\frac{\exp(\mathbf{w}^n \mathbf{h}^c)}{\exp(\mathbf{w}^p \mathbf{h}^c) + \exp(\mathbf{w}^n \mathbf{h}^c)} \right), \end{aligned}$$

where $r \in \{0, 1\}$ indicates whether the pronouns at the m-th and n-th positions should be considered co-reference or not. $\mathbf{w}^p \in \mathbb{R}^{4l}$ and $\mathbf{w}^n \in \mathbb{R}^{4l}$ are the parameters to learn.

3.2.3 Co-reference Scoring Function

Similar to the Eqn.(5), for each candidate, we use co-reference scoring function $\mathrm{Score}_\theta(x_i, y)$ for the answer selection:

$$\mathrm{Score}_\theta(x_i, y) = g(\mathbf{h}_i^{\mathrm{x}}, \mathbf{h}^{\mathrm{y}}) = \mathbf{w}^p \begin{bmatrix} \overrightarrow{\mathbf{h}_{i-1}} \\ \overleftarrow{\mathbf{h}_{i+1}} \\ \overrightarrow{\mathbf{h}_{j-1}} \\ \overleftarrow{\mathbf{h}_{j+1}} \end{bmatrix}, \quad (9)$$

where i is the position of the candidate in the sentence and j is the position of the pronoun.

	PDP	WSC
Co-reference Resolution Tool	41.7%	50.5
Patric Dhondt (WS Challenge 2016)	45.0%	-
Nicos Issak (WS Challenge 2016)	48.3%	-
Quan Liu (WS Challenge 2016 - winner)	58.3%	-
Unsupervised Semantic Similarity Method (USSM)	48.3%	-
Neural Knowledge Activated Method (NKAM)	51.7%	-
USSM + Cause-Effect Knowledge Base	55.0%	52.0%
USSM + Cause-Effect + WordNet + ConceptNet Knowledge Bases	56.7%	52.8%
USSM + NKAM	53.3%	
USSM + NKAM + 3 Knowledge Bases	66.7%	52.8%
ELMo	56.7%	51.5%
Google Language Model (Trinh and Le, 2018)	60.0%	56.4%
UDSSM-I	75.0%	54.5%
UDSSM-II	**75.0%**	**59.2%**
Google Language Model (ensemble)	70.0%	61.5%
UDSSM-I (ensemble)	76.7%	57.1%
UDSSM-II (ensemble)	**78.3%**	**62.4%**

Table 2: The experiment results on PDP and WSC datasets. We compare our models to Goolge LM trained on the same corpus [6].

4 Experiments

In this section, we will introduce the datasets to train and evaluate our models for commonsense reasoning, the hyper-parameters of our model, and the analysis of our results.

4.1 Datasets

Training Corpus We make use of the raw text from Gutenberg [7], a corpus offerring over 57,000 free eBooks, and 1 Billion Word [8], a corpus of news, to train our model. We first ignore the sentences that contain less than 10 tokens or longer than 50 tokens. Then, for the model UDSSM-I, we collect all the sentences with the pronoun before which there're at least two nouns. For UDSSM-II, we collect all the sentences with at least 2 pronouns. In total, we collect around 4 million training pairs from each corpus for our proposed method respectively, and we split 5% as validation set.

Evaluation Dataset We evaluate our model on the commonsense reasoning datasets, Pronoun Disambiguation Problems (PDP) [9] and Winograd Schema challenges (WSC) [10], which include 60 and 285 questions respectively. Both of the tasks are constructed for testing commonsense reasoning and all the questions from these challenges are obvious for human beings to solve with commonsense knowledge, but hard for machines to solve with statistical techniques.

4.2 Experimental Setting

We use the same setting for both our models. The hidden state dimension of a single-directional LSTM is set to be 300. We use 300 dimensional GloVe embeddings [11] for initialization. We use Adamax to optimise the model, set learning rate to be 0.002, dropout rate on all layers are tuned from [0, 0.1, 0.2] and the batch size from [30, 50, 100, 200]. For the model UDSSM-I, in one batch, we treat all sequence pairs not from the same sentence as negative cases. And it takes around 30 hours on a single K40 GPU to train our models, which are much faster than training a large LM (Jozefowicz et al., 2016) taking weeks on multiple GPUs.

[6] The best models reported in the works of Radford et al. (2019) and Trinh and Le (2018) are trained on a much larger corpus from Common Crawl.

[7] http://www.gutenberg.org

[8] https://github.com/ciprian-chelba/1-billion-word-language-modeling-benchmark

[9] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/PDPChallenge2016.xml

[10] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.xml

[11] https://github.com/stanfordnlp/GloVe

4.3 Experimental Results

The experiment results are shown in Table 2. Most of the performance in the top of the Table 2 are the models trained with external knowledge bases, such as Cause-Effect (Liu et al., 2016), WordNet (Miller, 1995), ConceptNet (Liu and Singh, 2004) knowledge bases. Unsupervised Semantic Similarity Method (USSM) (Liu et al., 2017) is based on the skip-gram model (Mikolov et al., 2013) to train word embeddings and the embeddings of all the words connected by knowledge bases are optimized to be closer. Neural Knowledge Activated Method (NKAM) (Liu et al., 2017) trained a binary classification model based on whether the word pairs appear in the knowledge base. One limitation of these methods is that they rely heavily on the external knowledge bases. Another limitation is that they just linearly aggregate the embeddings of the words in the context, and that's hard to integrate the word order information. Instead, our model with LSTM can better represent the contextual information. Besides, our model don't need any external knowledge bases, and achieve a significant improvement on both of the datasets.

We further compare our models with the unsupervised baselines, ELMo (Peters et al., 2018) which selects the candidate based on the cosine similarity of the hidden states of noun and pronoun. Another unsupervised baseline, Google Language Model for commonsense reasoning (Trinh and Le, 2018), which compares the perplexities of the new sentences by replacing the pronoun with candidates. To make a fair comparison to Trinh and Le (2018)'s work, we also train our single model on the corpus of Gutenberg only. We can see that both of our methods get significant improvement on the PDP dataset, and our UDSSM-II can achieve much better performance on the WSC dataset. We also report our ensemble model (nine models with different hyperparameters) trained with both corpus of Gutenberg and 1 Billion Word, and it also achieve better performance than Google Language Model trained with the same corpus.

Finally, we also compare to the pre-trained Coreference Resolution Tool (Clark and Manning, 2016a,b)[12], and we can see that it doesn't adapt to our commonsense reasoning tasks and can't tell the difference between each pair of sentences from WSC. In this way, our model can get much better performance.

4.4 Analysis

WSC 1:	**Paul** tried to call George on the phone, but **he** wasn't successful.
Ours 1:	**He** tried to call 911 using her cell phone but that **he** could n't get the phone to work.
WSC 2:	Paul tried to call **George** on the phone, but **he** was n't available .
Ours 2:	He tried twice to call **her** but **she** did not answer the phone .

Table 3: Comparison of the data from WSC and our training data. Our sentences are retrieved from the UDSSM-II training dataset based on the BM25 value for analysis. The pseudo labels in our training data can help identify the co-references in WSC.

In this subsection, we will conduct further analysis on the reason that our models work, the benefit of our models comparing to a baseline, and the limitation of our proposed models.

We have a further analysis on the pair-wise sentences, which we collected for training, to check how our model can work. We find that some reasoning problems can somehow be converted to the paraphrase problem. For example, in Table 3, we make use of Lucene Index[13] with BM25 to retrieve the similar sentences to the WSC sentences from our training dataset, and make a comparison. We can see that these pairs are somehow paraphrased each other respectively. For the first pair, the contextual representations of "Paul" and "he" in WSC could be similar to the contextual representations of "he" in our training sentence. As these representations are used to compute the co-reference score, the final scores would be similar. The pseudo label "positive" for our first sentence will make the positive probability of the golden co-references "Paul" and "he" in WSC higher. And for the second pair in Table 3, the pseudo label of positive in our second sentence will make the positive probability of the golden co-references "George" and "he" in WSC 2 higher. In this way, these kinds of co-reference patterns from training data can be directly mapped to solve the Winograd Schema Challenges.

[12] https://github.com/huggingface/neuralcoref

[13] http://lucene.apache.org/pylucene/

Here's another example from PDP demonstrating the benefit of our method: "Always before, Larry had helped Dad with his work. But *he* could not help him now, for Dad said that ". Trinh and Le (2018) failed on this one, probably because language models are not good at solving long distance dependence, and tends to predict that "he " refers to "his" in the near context rather the correct answer "Larry". And our model can give the correct prediction.

We further analysis the predictions of our model. We find that some specific commonsense knowledge are still hard to learn, such as the following pairs:

- The trophy doesn't fit into the brown **suitcase** because **it** is too small.
- The **trophy** doesn't fit into the brown suitcase because **it** is too large.

To solve this problem, the model should learn the knowledge to compare the size of the objects. However, all of our models trained with different hyper-parameters select the same candidate as the co-referred word for "it" in both sentences. To solve the problem, broader data need to collect for learning more commonsense knowledge.

5 Conclusion

In conclusion, to overcome the lack of human labeled data, we proposed two unsupervised deep structured semantic models (UDSSM) for commonsense reasoning. We evaluated our models on the commonsense reasoning tasks of Pronoun Disambiguation Problems (PDP) and Winograd Schema Challenge (Levesque et al., 2011), where the questions are quite easy for human to answer, but quite challenging for the machine. Without using any hand-craft knowledge base, our model achieved stat-of-the-art performance on the two tasks.

In the future work, we will use Transformer, which is proved to be more powerful than LSTM, as the encoder of our unsupervised deep structured semantic models, and we will collect a larger corpus from Common Crawl to train our model.

References

Daniel Bailey, Amelia Harrison, Yuliya Lierler, Vladimir Lifschitz, and Julian Michael. 2015. The winograd schema challenge and reasoning about correlation. In *Knowledge Representation; Coreference Resolution; Reasoning*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Kevin Clark and Christopher D Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Kevin Clark and Christopher D Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. *Proceedings of the Association for Computational Linguistics*.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, and Steve Pulman. 1996. Using the framework. Technical report, Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*.

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai. *arXiv preprint arXiv:1809.08267*.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2011. The winograd schema challenge. In *AAAI spring symposium: Logical formalizations of commonsense reasoning*.

Hugo Liu and Push Singh. 2004. Conceptneta practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Quan Liu, Hui Jiang, Andrew Evdokimov, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, and Yu Hu. 2016. Probabilistic reasoning via deep learning: Neural association models. *arXiv preprint arXiv:1603.07704*.

Quan Liu, Hui Jiang, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, and Yu Hu. 2017. Combing context and commonsense knowledge through neural networks for solving winograd schema problems. In *AAAI Spring Symposium Series*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.

Haoruo Peng, Daniel Khashabi, and Dan Roth. 2015. Solving hard coreference problems. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of the conference on empirical methods in natural language processing*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The winograd schema challenge. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018a. Modeling naive psychology of characters in simple commonsense stories. *arXiv preprint arXiv:1805.06533*.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A. Smith, and Yejin Choi. 2018b. Event2mind: Commonsense inference on events, intents, and reactions. In *Proceedings of the Association for Computational Linguistics*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

Peter Schüller. 2014. Tackling winograd schemas by formalizing relevance theory in knowledge graphs. In *Knowledge Representation and Reasoning Conference*.

Arpit Sharma, Nguyen H. Vo, Somak Aditya, and Chitta Baral. 2015. Towards addressing the winograd schema challenge: Building and using a semantic parser and a knowledge hunting module. In *Proceedings of the International Conference on Artificial Intelligence*.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies)*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. ReCoRD: Bridging the Gap between Human and Machine Commonsense Reading Comprehension. *arXiv preprint arXiv:1810.12885*.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *Transactions of the Association for Computational Linguistics*.