Antonym vs Synonym Distinction using InterlaCed Encoder NETworks (ICE-NET)

Muhammad Asif Ali,[1] **Yan Hu,**[1] **Jianbin Qin,**[2] **Di Wang**[1]
[1] King Abdullah University of Science and Technology, KSA
[2] Shenzhen University, China
{muhammadasif.ali, yan.hu, di.wang}@kaust.edu.sa, qinjianbin@szu.edu.cn

Abstract

Antonyms vs synonyms distinction is a core challenge in lexico-semantic analysis and automated lexical resource construction. These pairs share a similar distributional context which makes it harder to distinguish them. Leading research in this regard attempts to capture the properties of the relation pairs, i.e., symmetry, transitivity, and trans-transitivity. However, the inability of existing research to appropriately model the relation-specific properties limits their end performance. In this paper, we propose InterlaCed Encoder NETworks (i.e., ICE-NET) for antonym vs synonym distinction, that aim to capture and model the relation-specific properties of the antonyms and synonyms pairs in order to perform the classification task in a performance-enhanced manner. Experimental evaluation using the benchmark datasets shows that ICE-NET outperforms the existing research by a relative score of upto 1.8% in F1-measure. We release the codes for ICE-NET at https://github.com/asif6827/ICENET.

1 Introduction

Antonyms vs synonyms distinction is a core challenge in natural language processing applications, including but not limited to: sentiment analysis, machine translation, named entity typing etc. Synonyms are defined as semantically related words, whereas antonyms are defined as semantically opposite words. For example “disperse" and “scatter" are synonyms, while “disperse" and “garner" are antonyms (Ono et al., 2015).

Existing research on the antonym-synonym distinction is primarily categorized into pattern-based and embedding-based approaches. Pattern-based approaches attempt to curate distinguishing lexico-syntactic patterns for the word pairs (Schwartz et al., 2015; Nguyen et al., 2017). A major limitation of the pattern-based approaches is the sparsity of the feature space. Despite using massive data sets, the generalization attempts result in highly overlapping and noisy features, which further deteriorate the model’s performance.

Embedding based methods rely on the distributional hypothesis, i.e., *“words that occur in the same contexts tend to have similar meanings"* (Harris, 1954). These methods use widely available embedding resources to capture/compute the semantic relatedness of synonym and antonym pairs (Nguyen et al., 2016; Etcheverry and Wonsever, 2019). Ali et al. (2019) proposed Distiller that uses non-linear projections to project the embedding vectors in task-specific dense sub-spaces.

The key challenge faced by existing embedding-based approaches is their inability to correctly model the inherent relation-specific properties among different relation pairs. These models mix different lexico-semantic relations and perform poorly when applied to a specific task (Ali et al., 2019). Existing approaches, moreover, model each relation pair independently, which is not adequate for antonym and synonym relation pairs as these relation pairs exhibit unique properties that may be exploited by modeling the relation pair in correlation with other instances (discussed in detail in Section 4).

Keeping in view the above-mentioned challenges, in this paper, we propose InterlaCed Encoder NETworks (ICE-NET) for antonym vs synonym distinction. ICE-NET uses multiple different encoders to capture relation-specific properties of antonym and synonym pairs from pre-trained embeddings in order to augment the end-performance of the antonyms vs synonyms distinction task. Specifically, it uses: (i) an encoder (ENC-1) to capture the symmetry of synonyms; (ii) an encoder (ENC-2) to model the symmetry for antonyms; and (iii) an encoder (ENC-3) to preserve the transitivity of the synonyms and trans-transitivity of antonym and synonym relation pairs by employing attentive graph convolutions. These relation-specific

Findings of the Association for Computational Linguistics: EACL 2024, pages 1462–1473
March 17-22, 2024

properties of antonym and synonym relation pairs are illustrated in Figure 1(a) and explained in Section 3.1.

We are the first to make an attempt to use attentive graph convolutions for modeling the underlying characteristics of antonym and synonym relation pairs. Note, this work is different from existing works using graph convolutional networks for relational data, e.g., (Schlichtkrull et al., 2018), as antonyms and synonyms possess unique properties which makes them different from relation pairs in the Knowledge Graphs (KG), e.g., FB15K, WN18 (Bordes et al., 2013).

ICE-NET is a shift from the existing instance-based modeling approaches to graph-based framing which allows effective information sharing across multiple instances at a time to perform the end classification in a performance-enhanced fashion. ICE-NET can be used with any available pre-trained embedding resources, which makes it more flexible than the existing approaches relying on huge text corpora. We summarize the major contributions of this paper as follows:

- We propose ICE-NET, i.e., a combination of interlaced encoder networks to refine relation-specific information from the pre-trained embeddings.
- ICE-NET is the first to use attentive graph convolutions for antonym vs synonym distinction that provide a provision to analyze/classify a word pair in correlation with multiple neighboring pairs/words, rather than independent instant-level modeling.
- We demonstrate the effectiveness of the proposed model using benchmark data sets. ICE-NET outperforms the existing models by a margin of upto 1.8% in terms of F1-measure.

2 Related Work

Earlier research on antonym synonym distinction attempts at capturing lexico-syntactic patterns between the word pairs co-occurring within the same sentence.

Lin et al. (2003) considered pharasal patterns: *"from X to Y"*, and *"either X or Y"* to identify synonyms amongst distributionally similar words. Baroni and Bisi (2004) used co-occurrence statistics to discover synonyms and distinguish them from unrelated terms. Van der Plas and Tiedemann (2006) used word alignment measures using parallel corpora from multiple different languages to capture synonyms. Lobanova et al. (2010) used a set of seed pairs to capture patterns in the data and later used these patterns to extract new antonym pairs from text corpora. Roth and Im Walde (2014) proposed discourse markers as features alternate to the lexico-syntactic patterns. Schwartz et al. (2015) proposed automated routines to acquire a set of symmetric patterns for word similarity prediction. Nguyen et al. (2017) proposed AntSynNET that uses a set of lexico-syntactic patterns between the word pairs within the same sentence captured over huge text corpora.

In the recent past, embedding models have received considerable research attention for antonyms vs synonyms distinction. These models are based on distributional hypotheses, i.e., words with similar meanings co-occur in a similar context (Goldberg and Levy, 2014; Pennington et al., 2014; Grave et al., 2018). A major advantage offered by the embedding-based approaches is the freedom to curate and train embedding vectors for features extracted from text corpora. Adel and Schütze (2014) used skip-gram modeling to train embedding vectors using coreference chains. Nguyen et al. (2016) used lexical contrast information in the skip-gram model for antonym and synonym distinction. Ono et al. (2015) uses dictionaries along with distributional information to detect probable antonyms. Ali et al. (2019) used a set of encoder functions to project the word embeddings in constrained subspaces in order to capture the relation-specific properties of the data. Xie and Zeng (2021) employed a mixture-of-experts framework based on a divide-and-conquer strategy. They used a number of localized experts focused on different subspaces and a gating mechanism to formulate the expert mixture.

We observe some of the limitations of the existing work as follows. The pattern-based approaches are limited owing to the noisy and overlapping nature of the patterns. The embedding models are limited by the challenges posed by the distributional nature of the word embeddings, e.g., in Glove embeddings top similar words for the word *"small"* yields a combination of synonyms, antonyms, and irrelevant words (Ali et al., 2019).

3 Background

3.1 Preliminaries

Antonyms and synonyms are a special kind of relation pairs (denoted by r_A and r_S) with unique properties, i.e., (a) antonyms possess symmetry,

(b) synonyms exhibit symmetry and transitivity, (c) antonyms and synonyms when analyzed in combination demonstrate trans-transitivity.

These properties are depicted in Figure 1 (a). For ease of interpretation, we use (h, r, t) to represent a relation tuple, where h corresponds to the "head" and t is the "tail" of relation r. For word pair (h, t) and relation r, symmetry implies (h, r, t) iff (t, r, h). The transitivity between the relation implies: *if* (h, r, t) and $(t, r, t^{'})$ hold then $(h, r, t^{'})$ also holds, as shown by the words "nasty" and "horrible" in Figure 1(a). Trans-transitivity implies: *if* (h, r_A, t) and $(t, r_S, t^{'})$ hold then $(h, r_A, t^{'})$ also holds, also illustrated between the words "nasty" and "pleasing".

3.2 KG Embeddings Methods

Ali et al. (2019) pointed out a key limitation of the translational embedding methods (commonly used for KG embeddings) in modeling symmetric relations. For instance, for a symmetric relation $\mathbf{r}$, it is not possible for translational embeddings to preserve both vector operations: $\mathbf{h} + \mathbf{r} = \mathbf{t}$ and $\mathbf{t} + \mathbf{r} = \mathbf{h}$ at the same time. This is also illustrated in Figure 1(b), where we show $\mathbf{t}^{'} \neq \mathbf{t}$. For details refer to the original article by Ali et al. (2019). Likewise, some of the key difference of our work from existing work, i.e., R-GCN by Schlichtkrull et al. (2018) are explained in Appendix A.2.

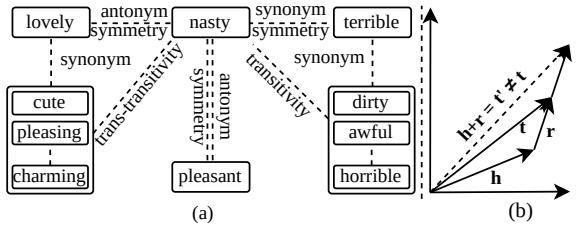

Figure 1: (a) Properties of the antonym and synonym relation pairs, i.e., symmetry, transitivity, and trans-transitivity; (b) Limitation of translational embeddings in capturing the antonym and synonym relations (Ali et al., 2019).

4 Proposed Approach

Given that existing KG embeddings are not able to model the relation-specific properties of the antonym and synonym pairs, we propose ICE-NET that takes pre-trained word embeddings as inputs and projects them to low-dimensional space. In order to ensure that low-dimensional space captures the relation-specific properties of the data to the best possible extent ICE-NET uses three different encoder networks. We call overall architecture as interlaced structure, because these networks are interconnected, i.e., (a) loss function of ENC-2 also depends upon ENC-1, (b) output of encoders (ENC-1, and ENC-2) is used as input to the ENC-3. Details about each encoder are as follows:

4.1 ENC-1

The goal of this encoder is to capture the symmetry of the synonym relation pairs. For this, we use a two-layered feed-forward function: $f_1(X) = \sigma W_{12} * \sigma(W_{11} * X + b_{11}) + b_{12}$ to project d-dimensional embeddings ($X \in \mathbf{R}^d$) to p-dimensions ($\mathbf{R}^p$). Here W_{11} and W_{12} are the weight matrices; b_{11} and b_{12} are the bias terms. For encoded word pairs to preserve symmetry among relation pairs, we employ negative sampling techniques. Specifically, we use a margin-based loss (shown in Equation 1) to project a word close to its true synonyms, while at the same time push it from irrelevant words. This formulation preserves the symmetry of the relation pair owing to the commutative nature of the inner product. It is also justified by the fact: if $\mathbf{x}_h$ is embedded close to $\mathbf{x}_t$, then $\mathbf{x}_t$ is also embedded close to $\mathbf{x}_h$.

$$\begin{aligned} \mathrm{L}_1 = & \sum_{(h,t)\in T_1} \max(0, \gamma_1 - \tanh(\langle f_1(\mathbf{x}_h), f_1(\mathbf{x}_t)\rangle)) \\ & + \sum_{(h',t')\in T_1'} \max(0, \gamma_1 + \tanh(\langle f_1(\mathbf{x}_h^{'}), f_1(\mathbf{x}_t^{'})\rangle)) \end{aligned} \quad (1)$$

Here γ_1 is the margin; T_1 corresponds to the synonym pairs; $\mathbf{x}_h$, $\mathbf{x}_t$ are the embedding vectors for head and tail words. T_1' is acquired by randomly replacing one of the words from the pairs in T_1 and/or using antonyms as negative samples.

4.2 ENC-2

This encoder aims to capture the symmetry for the antonym relation pairs. For this we use a two layered feed-forward function: $f_2(X) = \sigma W_{22} * \sigma(W_{21} * X + b_{21}) + b_{22}$ to project d-dimensional embeddings ($X \in \mathbf{R}^d$) to p-dimensions ($\mathbf{R}^p$). Here $X \in \mathbf{R}^d$ corresponds to the pre-trained word embeddings; W_{21} and W_{22} are the weight matrices; b_{21} and b_{22} are the bias terms. In order to preserve the symmetry of the antonym relations, we use another margin-based loss function (shown in Equation 2) to project a word close to its true antonyms, while at the same time push it from irrelevant words.

$$\begin{aligned} \mathrm{L}_2 = & \sum_{(h,t)\in T_2} \max(0, \gamma_2 - \tanh(\langle f_2(\mathbf{x}_h), f_1(\mathbf{x}_t)\rangle)) \\ & + \sum_{(h',t')\in T_2'} \max(0, \gamma_2 + \tanh(\langle f_2(\mathbf{x}_h^{'}), f_1(\mathbf{x}_t^{'})\rangle)) \end{aligned} \quad (2)$$

Note, for L_2 we use both functions, i.e., $f_1(X), f_2(X)$, that allows us to project $\mathbf{x}_h$ close

to its antonym $\mathbf{x}_t$ as well as synonyms of $\mathbf{x}_t$. Here again the symmetry of the relation is preserved by the commutative nature of the inner product. γ_2 is the margin term, T_2 corresponds to the antonym pairs; $\mathbf{x}_h$, $\mathbf{x}_t$ are the embedding vectors for head and tail words. T_2' is acquired by randomly replacing one of the words from the pairs in T_2 and/or using synonyms as negative samples.

Given that the encoders (ENC-1, ENC-2) use two different non-linear functions to project the pre-trained embeddings, it allows us to learn two projections for each word. Later, we use all possible projection scores as indicators for the word pair to be probable antonym and/or synonym pair. This setting is different from the previous research that embeds synonyms close to each other, while antonyms are projected at an angle of 180° (Ono et al., 2015) as it is hard to preserve the relation-specific properties for the resultant embeddings.

4.3 ENC-3

Finally, in order to preserve the transitivity of the synonym pairs and the trans-transitivity of antonym and synonym relation pairs in combination we propose an attentive graph convolutional encoder under transductive setting. We exploit the fact that the a word may be represented as a node in the graph, and each word may be surrounded by an arbitrary number of semantically related words as neighbouring nodes in the graph. We argue that this setting is more flexible in capturing the relation-specific properties involving arbitrary number of words, as it allows modeling the relation pairs in complete correlation with each other, which is more practical than modeling these pairs independent of each other. It also provides the provision for effective information sharing across the neighboring nodes using attention weights. Similar ideas has already been applied to capture the semantic-relatedness for embeddings trained for different languages (Ali et al., 2023a,b).

In our case, we use two different graphs, namely: G_h, and G_t, for preserving the relations amongst the head and tail words respectively. We outline the graph construction process in Algorithm 1. It is explained as follows:

Graph Construction. The graph construction process uses data set D and 300-d pre-trained Fasttext embeddings (Grave et al., 2018) as inputs and returns two graphs G_h and G_t as output. The details are as follows.

Algorithm 1 Graph Construction

Inputs: Embedding; $D = D_{tr} + D_{dev} + D_{test}$
Outputs: Graphs: $\mathrm{G}_h, \mathrm{G}_t$

1: $\{\mathrm{Syn}_h, \mathrm{Ant}_h\}_{h=1}^{V} \leftarrow \emptyset$; $\mathrm{G}_t \leftarrow \emptyset$
2: $\{\mathrm{Syn}_t, \mathrm{Ant}_t\}_{t=1}^{V} \leftarrow \emptyset$; $\mathrm{G}_h \leftarrow \emptyset$
3: Train $\mathrm{M}_{init}(D_{tr}; \mathrm{L}_1, \mathrm{L}_2)$
4: **for** inst$(h, t) \leftarrow 1$ to D **do**
5: $\quad y^*$ = score$(\mathrm{M}_{init}, \mathrm{inst})$
6: $\quad$ **if** $y^* \geq \mathrm{ANT}_{thr}$ **then**
7: $\quad\quad$ Update$\{\mathrm{Ant}_h; \mathrm{Ant}_t\}$
8: $\quad$ **else if** $y^* \leq \mathrm{SYN}_{thr}$ **then**
9: $\quad\quad$ Update$\{\mathrm{Syn}_h; \mathrm{Syn}_t\}$
10: $\quad$ **end if**
11: **end for**
12: **for** $pair \in \{\mathrm{Syn}_t, \mathrm{Ant}_t\}$ **do**
13: $\quad \mathrm{G}_h \leftarrow \mathrm{G}_h \cup \{edge_h(pair)\}$
14: **end for**
15: **for** $pair \in \{\mathrm{Syn}_h, \mathrm{Ant}_h\}$ **do**
16: $\quad \mathrm{G}_t \leftarrow \mathrm{G}_t \cup \{edge_t(pair)\}$
17: **end for**
18: **return** G_h; G_t

Firstly, we initialize dictionaries $\{\mathrm{Syn}_h, \mathrm{Ant}_h\}$ and $\{\mathrm{Syn}_t, \mathrm{Syn}_t\}$ to store probable synonym and antonym pairs with head word h and tail word t respectively (lines 1-2). We train a basic model (M_{init}) using the encoders (ENC-1 and ENC-2) and available training data (line 3) in an end-to-end fashion. Later, M_{init} is used to assign a score (y^*) to each pair in the data D (line 6). We use y^* compared against the thresholds $\{\mathrm{ANT}_{thr}$,$\mathrm{SYN}_{thr},\}$ to update the data structures $\{\mathrm{Ant}_h, \mathrm{Ant}_t\}$, and $\{\mathrm{Syn}_h, \mathrm{Syn}_t\}$ respectively (lines 6-9). The core logic is: we add inst(h, t) to Ant_h, if (a) head word (h) corresponds to a key in Ant_h, (b) it is a probable antonym pair with ($y^* \geq \mathrm{ANT}_{thr}$). Later, we use the information in the dictionaries to construct the graphs (lines 12-16).

We explain the construction of G_h using the information in Syn_t, Ant_t, as follows. Given that Syn_t contains the information about the list of probable synonym pairs with the tail word *"t"*. In order to preserve the transitivity for the synonym pairs with tail *"t"*, we formulate pairwise edges between the head terms in Syn_t. It is based on the assumption that head words of the relation pairs with the same tail, i.e., *"t"* are likely to be synonyms of each other. Likewise, Ant_t contains the information about the list of probable antonym pairs with the tail word *"t"*. In order to preserve the trans-transitivity of relation pairs with tail *"t"*, we formulate pairwise edges between the head terms in Ant_t.

It is based on the assumption that the head words of the antonym relation pairs with same tail *"t"* are likely to be synonyms of each other. Eventually, we combine these edges to formulate the graph G_h.

We follow a similar procedure to construct the graph G_t using information in Syn_h, Ant_h. Finally, we return graphs G_h and G_t as the output of the graph construction process.

Attentive Aggregation. The graph construction process surrounds each word in the graphs G_h and G_t by a set of probable synonyms. Later, it recomputes the representation of each word as an attentive aggregation of the neighbors. For this, it uses the following layer-wise information propagation mechanism:

$$L^{(i+1)} = \rho(\tilde{\xi_G} L^{(i)} W_i) \tag{3}$$

where $\tilde{\xi_G} = \bar{D}^{-1/2}(\xi_G + I)\bar{D}^{-1/2}$ is the normalized symmetric marix, $\bar{D}$ is the degree matrix of ξ_G, ξ_G is the weighted adjacency matrix containing attention weights for G, $L^{(i)}$ is the input representation from the previous layer, W_i is the learn-able weight matrix. We also add identity matrix I to ξ_G in order to allow self-connections for each word in the graphs. It allows the encoder to analyze each word as a weighted combination of itself and its semantic neighbors. Our formulation for attentive graph convolutions is inspired by Ali et al. (2020), and its non-euclidean variant Ali et al. (2021). Intuitive explanations in this regard are provided in Appendix A.1.

For ICE-NET, we use a two-layered attentive graph convolution encoder with ReLU nonlinearity to generate the final representations of each word. Specifically, for the relation tuples (h, r, t) in data D, the output of the encoders (ENC-1 and ENC-2) is separately processed by the attentive graph convolution networks to generate the final representations, as follows:

$$\begin{aligned}
\mathbf{X}_{hh} &= \tilde{\xi_{G_h}}(\text{ReLU}(\tilde{\xi_{G_h}}\ f_1(X_h)W_{hh_1})W_{hh_2} \\
\mathbf{X}_{ht} &= \tilde{\xi_{G_t}}(\text{ReLU}(\tilde{\xi_{G_t}}\ f_1(X_t)W_{ht_1})W_{ht_2} \\
\mathbf{X}_{th} &= \tilde{\xi_{G_h}}(\text{ReLU}(\tilde{\xi_{G_h}}\ f_2(X_h)W_{th_1})W_{th_2} \\
\mathbf{X}_{tt} &= \tilde{\xi_{G_t}}(\text{ReLU}(\tilde{\xi_{G_t}}\ f_2(X_t)W_{tt_1})W_{tt_2}
\end{aligned} \tag{4}$$

Here $f_1(X), f_2(X) \in \mathbf{R}^p$ are the outputs of the encoders (ENC-1, and ENC-2) used as inputs for ENC-3. W_i are learn-able weights, $\mathbf{X}_i \in \mathbf{R}^q$ are the outputs of attentive graph convolution. In order to train the attentive graph convolution network (ENC-3), we compute the score vectors $\{\mathbf{x}_1, \mathbf{x}_2\}$ and $\{\mathbf{x}_3, \mathbf{x}_4\}$ as indicative of synonymy and antonymy respectively.

word class	(a) Random			(b) Lexical		
	train	dev	test	train	dev	test
Adjective	5562	398	1986	4227	303	1498
Noun	2836	206	1020	2667	191	954
Verb	2534	182	908	2034	146	712

Table 1: Antonym/Synonym distinction datasets

$$\begin{aligned}
\mathbf{x}_1 &= \cos(\mathbf{X}_{th}, \mathbf{X}_{tt});\ \mathbf{x}_2 = \cos(\mathbf{X}_{hh}, \mathbf{X}_{ht}) \\
\mathbf{x}_3 &= \cos(\mathbf{X}_{hh}, \mathbf{X}_{tt});\ \mathbf{x}_4 = \cos(\mathbf{X}_{ht}, \mathbf{X}_{th})
\end{aligned} \tag{5}$$

where $\cos(\mathbf{X}, \mathbf{Y})$ is the element-wise cosine of the vector pairs in $\mathbf{X}$ and $\mathbf{Y}$. We concatenate these scores to get the feature matrix: $X_F = [\mathbf{x}_1; \mathbf{x}_2; \mathbf{x}_3; \mathbf{x}_4]$, and use cross-entropy loss to train the encoder, shown in Equation 6:

$$\mathrm{L}_3 = -\frac{1}{N}\sum_{i=1}^{N} \log(p(y_i|h_i, t_i)) \tag{6}$$

where $p(\mathbf{y} \mid \mathbf{x}_h, \mathbf{x}_t) = softmax(\mathbf{w}\mathbf{x}_F + \mathbf{b})$ with $\hat{y} = \text{argmax}_y\, p(\mathbf{y}|\mathbf{x}_h, \mathbf{x}_t)$, $\mathbf{w}$ is the weight matrix and $\mathbf{b}$ is the bias term.

4.4 The Complete Model.

Finally, we combine the loss functions of the individual encoder networks, i.e., $\mathrm{L}_1 + \mathrm{L}_2 + \mathrm{L}_3$ as the loss function of ICE-NET. We train the model in an end-to-end fashion.

5 Experiments and Results

5.1 Datasets

We evaluate the proposed approach on two different data sets: (a) A benchmark data set by (Nguyen et al., 2017) manually curated from WordNet and Wordnik[1]. It encompasses randomly split synonyms and antonym pairs corresponding to three word classes (adjective, noun, and verb). (b) A lexical split curated by (Xie and Zeng, 2021). For both data sets, the ratio between the antonyms and synonym pairs within each word class is approximately 1:1. The statistics of each data set are shown in Table 1.

5.2 Experimental Settings

Similar to the baseline methods, for main experimentation we report the results using random split and 300-d Fasttext embeddings (Grave et al., 2018) trained on wiki-news corpus. Results using the dLCE embeddings and lexical split of the data are discussed in Section 6. The embedding vectors

[1]http://www.wordnik.com

for the OOV tokens are randomly initialized. For model training, we use Adam optimizer (Kingma and Ba, 2014) with learning rate=0.001. The values for SYN_{thr} and ANT_{thr} are set to 0.15 and 0.10 respectively. For L_1 and L_2 the values for the margin terms are: $\gamma_1 = \gamma_2 = 0.9$. Output dimensionality of ENC-1 and ENC-2 is 80d and for ENC-3 is 60d. We used TensorFlow toolkit (version 2.12) to run the experiments. We report mean and standard deviation of the scores computed over five runs of the experiments. All experiments were performed using Intel Core-i9-10900X CPU, and Nvidia 3090Ti GPU. On this GPU, a single run of the experiments takes approximately thirty minutes.

5.3 Baseline Models

In order to test the effectiveness of ICE-NET, we design two baseline models. Baseline-1 aims to analyze the ability of ICE-NET to encode the information in the pre-trained embeddings. For this, we use random vectors in place of pre-trained embeddings. Baseline-2 aims to analyze the ability of graph convolutions to preserve relation-specific properties. For this, we use a basic variant of ICE-NET relying only on the ENC-1 and ENC-2.

We also compare ICE-NET with existing state-of-the-art research on the antonym-synonym distinction task, i.e., (i) AntSynNET by Nguyen et al. (2017), (ii) Parasiam by Etcheverry and Wonsever (2019), (iii) Distiller by Ali et al. (2019), and (iv) MoE-ASD by Xie and Zeng (2021). For all these models, we report the scores reported in the original papers, as they are computed using the same data settings as that of ours.

5.4 Main Results

The performance comparison of ICE-NET is reported in Table 2. For these results, we use the random split of the data and 300-d Fasttext embeddings. We boldface overall best scores with previous state-of-the-art underlined. A low variance of the results shows that ICE-NET yields a stable performance across multiple runs.

Comparing the performance of ICE-NET against the previous state-of-the-art, we observe, for the adjective data sets, the ICE-NET outperforms existing best by 2.1%, 0.2% and 1.8% for precision, recall, and F1 scores respectively. For the verbs data set, it outweighs the precision, recall and F1 score by 0.45%, 1.19%, and 0.77% respectively. For the nouns data set the improvement in performance for the precision and F1-scores is 6.42%, and 1.61%.

Analyzing the performance of ICE-NET against the baseline models, a significant decline in the performance for the baseline-1 shows that pre-trained embeddings carry a significant amount of relation-specific information which is refined by ICE-NET in a performance-enhanced fashion. Likewise, the performance comparison against the baseline-2 shows that attentive graph convolutions help the ICE-NET in capturing probable relation pairs by using the relation-specific properties, i.e., symmetry, transitivity, and trans-transitivity to the best possible extent, which in turn boosts the end performance of the model.

These results show the impact of using attentive graph convolutions for the distinction task. It affirms our hypothesis that graph convolutions offer an optimal setting to model the relation-specific data because it provides the provision for information sharing across semantically related words, rather than modeling data instances completely independently of each other.

6 Analyses

In this section, we perform a detailed analyses of the ICE-NET under different settings, namely: (i) dLCE embeddings (Nguyen et al., 2016), (ii) Lexical split, (iii) Ablation analysis, and (iv) Error analyses.

6.1 dLCE Embeddings

Results for ICE-NET using random split and dLCE embeddings are shown in Table 3. We also report the scores for the previous research using the same test settings (i.e., data split and embeddings). These results show ICE-NET outperforms the existing research yielding a higher value of F1-score across all three data categories (adjective, verb and noun). These results compared to the results in Table 2 (using fasttext embeddings) show that dLCE embeddings being trained on lexical contrast information carry more distinctive information for the distinction task compared to generalized pre-trained word embeddings.

6.2 Lexical Split

In this subsection, we analyze the results of ICE-NET corresponding to the lexical split of the antonym synonym distinction task (Xie and Zeng, 2021). Note, the lexical split assumes no overlap across train, dev, and test splits in order to avoid lexical memorization (Shwartz et al., 2016). Generally, the lexical split is considered a much tough evaluation setting compared to the random split, as

Methodology	Adjective			Verb			Noun		
	P	R	F1	P	R	F1	P	R	F1
Baseline-1 (Random vectors)	0.657	0.665	0.661	0.782	0.819	0.800	0.783	0.751	0.767
Baseline-2 (w/o Graph conv.)	0.828	0.909	0.867	0.837	0.915	0.879	0.818	0.818	0.818
AntSynNet (Nguyen et al., 2017)	0.750	0.798	0.773	0.717	0.826	0.768	0.807	0.827	0.817
Parasiam (Etcheverry and Wonsever, 2019)	0.855	0.857	0.856	0.864	0.921	0.891	0.837	0.859	0.848
Distiller (Ali et al., 2019)	0.854	0.917	0.884	0.871	0.912	0.891	0.823	0.866	0.844
MoE-ASD (Xie and Zeng, 2021)	0.878	0.907	0.892	0.895	0.920	0.908	0.841	**0.900**	0.869
ICE-NET	**0.896**± 0.0005	**0.919**± 0.0005	**0.908**± 0.0005	**0.899**± 0.001	**0.932**± 0.001	**0.915**± 0.001	**0.895**± 0.001	0.871± 0.001	**0.883**± 0.001

Table 2: ICE-NET performance comparison using random split

Methodology	Adjective			Verb			Noun		
	P	R	F1	P	R	F1	P	R	F1
AntSynNet (Nguyen et al., 2017)	0.763	0.807	0.784	0.743	0.815	0.777	0.816	0.898	0.855
Parasiam (Etcheverry and Wonsever, 2019)	0.874	**0.950**	0.910	0.837	**0.953**	0.891	0.847	0.939	0.891
Distiller (Ali et al., 2019)	0.912	0.944	0.928	0.899	0.944	0.921	0.905	0.918	0.911
MoE-ASD (Xie and Zeng, 2021)	0.935	0.941	0.938	**0.914**	0.944	0.929	0.920	0.950	0.935
ICE-NET	**0.936**±0.0002	0.945±0.0002	**0.940**±0.0002	0.913±0.001	**0.953**±0.001	**0.933**±0.001	**0.925**±0.001	**0.953**±0.001	**0.939**±0.001

Table 3: ICE-NET performance comparison using random split and dLCE Embeddings

it doesn't allow information sharing across different data splits based on overlapping vocabulary.

For the lexical split, the results for both dLCE embeddings and Fasttext embeddings are shown in Table 5. Comparing the performance of our model against existing research, it is evident for both embeddings, i.e., Fasttext and dLCE, ICE-NET yields a higher F1 measure compared to the existing models.

6.3 Ablation Analyses

The core focus of ICE-NET is to employ attentive graph convolutions in order to capture the relation-specific properties of antonym and synonym pairs in order to perform the distinction task in a robust way. In order to simplify things, we deliberately don't include any hand-crafted features, e.g., negation prefixes etc., as a part of ICE-NET.

For the ablation analyses of ICE-NET, we: (a) compare the performance of ICE-NET with and without attentive graph convolutions, (b) analyze the impact of different attention weights.

(a) Impact of attentive convolutions. In order to analyze the impact of attentive graph convolutions, we train a variant of ICE-NET encompassing only the encoder networks. Note, we also used a similar model in Section 5.4 (shown as baseline-2 in Table 2), however, the end goal of this analysis is to dig out a few example pairs that benefited especially from the attentive graph convolutions.

Some of the synonym and antonym word pairs that were corrected by attentive convolutions include: {(lecture, reprimand), (single,retire)} and {(tender,demand), (file, rank)} respectively. These word pairs were not easy to categorize otherwise by the variant of ICE-NET without graph convolutions. This shows the significance of the attentive convolutions in acquiring relation-specific information from semantically related neighbors that was helpful to reinforce the classification decision.

(b) Varying attention weights. We also analyze the impact of different attention weights on the end performance of the model. Corresponding results are shown in Table 4. For these experiments, we use five different types of attention weights, yielding adjacency matrices: A1, A2, A3, A4, and A5 in Table 4. We use hard attention weights that are not fine-tuned during the model training. The graphs (G_h and G_t) used in these experiments correspond to the best performing variant of ICE-NET.

For A1, we use random values as attention weights, i.e., we randomly assign a value to each word pair from the range ($0.1 \sim 0.9$). For A2, we use the identity as the adjacency matrix for the word pairs in the graphs, i.e., we completely ignore the effect of graph convolutions. For A3, we use the embedding similarity scores of the fasttext embeddings as the attention scores. This setting is based on the distributional hypothesis, i.e., distributionally similar words get higher scores. For A4, we use the embedding similarity scores from the output of ENC-1 network for the model M_{init}, trained entirely using two encoder networks. The motivation for using these scores as attention weights is the fact that ENC-1 is responsible for capturing the synonym pairs, so it will assign a higher score to probable synonyms, and a relatively lower score to probable antonyms.

For A5, we use attention weights similar to the setting of A4 with the difference that we downscale the weights for probably erroneous edges in the graph. For less confident relation pairs with scores closer to the thresholds, i.e., ANT_{thr}, SYN_{thr}, we simply downscale the attention weight by half. This setting in turn limits the error propagation in the end-model caused by the erroneous edges in the graphs.

Results in Table 4 show that ICE-NET (A5), outperforms other variants of attention weights. A similar performance is observed by the model ICE-

Adjacency	Adjective			Verb			Noun		
	P	R	F1	P	R	F1	P	R	F1
ICE-NET (A1 = Random)	0.862	0.863	0.863	0.799	0.894	0.844	0.816	0.863	0.839
ICE-NET (A2 = Identity)	0.849	0.886	0.867	0.761	0.896	0.823	0.830	0.861	0.845
ICE-NET (A3 = Fasttext)	0.880	0.873	0.877	0.867	0.930	0.897	0.851	**0.873**	0.862
ICE-NET (A4 = M_{init})	0.881	0.909	0.895	**0.899**	0.925	0.912	0.874	0.867	0.870
ICE-NET (A5 = Weighted-M_{init})	**0.896**±0.0005	**0.919**±0.0005	**0.908**±0.0005	0.898±0.001	**0.932**±0.001	**0.915**±0.001	**0.895**±0.001	0.871±0.001	**0.883**±0.001

Table 4: ICE-NET performance comparison using different adjacency matrices and random data split

Embedding	Model	Adjective			Verb			Noun		
		P	R	F1	P	R	F1	P	R	F1
FastText	Parasiam (Etcheverry and Wonsever, 2019)	0.694	0.866	0.769	0.642	**0.824**	0.719	0.740	0.759	0.748
	MoE-ASD (Xie and Zeng, 2021)	0.808	0.810	0.809	**0.830**	0.693	0.753	**0.846**	0.722	0.776
	ICE-NET	0.760±0.0005	**0.870**±0.0005	**0.815**±0.0005	0.740±0.001	0.777±0.001	**0.758**±0.001	0.763±0.002	**0.826**±0.002	**0.793**±0.002
dLCE	Parasiam (Etcheverry and Wonsever, 2019)	0.768	0.952	0.850	0.769	0.877	0.819	0.843	0.914	0.876
	MoE-ASD (Xie and Zeng, 2021)	**0.877**	0.908	0.892	**0.860**	0.835	0.847	**0.912**	0.869	0.890
	ICE-NET	0.835±0.0004	**0.971**±0.0004	**0.898**±0.0004	0.793±0.002	**0.938**±0.002	**0.859**±0.002	0.886±0.001	**0.915**±0.001	**0.900**±0.001

Table 5: Antonym/Synonym distinction performance for the lexical split

NET (A4). Relatively lower scores for the models using the random values and identity matrices as attention weights show the significance of sharing information amongst semantically related neighbors in an appropriate proportion in order to perform the end task in a performance-enhanced way. Likewise, the score for ICE-NET (A3) show that by default the distributional scores of the pre-trained embeddings are not suitable for the end task. These analyses clearly indicate that the choice of attention weight plays a vital role in capturing the properties of the data.

6.4 Error Analyses

For the variant of ICE-NET using random split and Fasttext embeddings, we collect a sample of approximately fifty error cases for each word class (adjectives, verbs, and nouns) to analyze the most probable reasons for the errors. We broadly categorize the errors into the following different categories: (a) the inability of input embeddings to cater to multiple senses, (b) the distributional embeddings for out-of-vocabulary (OOV) and/or rare words, and (c) other cases, e.g., negation prefix, errors with unknown reasons etc.

We separately report the number of erroneous edges/neighbours in the graphs: G_h and G_t. Information propagation over these erroneous edges may also lead to the classification errors, however, it is hard to quantify such errors.

For adjectives, almost 25% errors correspond to the sense category, 20% errors are caused by rare words and/or OOV tokens, and the rest errors are attributed to negation prefixes and unknown reasons. For nouns, 30% errors belong to the sense category, 12.5% errors result due to rare words and/or OOV tokens, with the rest of the errors assigned to the negation prefixes and unknown reasons. For verbs, 13% errors correspond to the sense category, 15% errors are caused by rare words/OOV tokens and the rest of the errors may be attributed to negation prefixes and unexplained reasons. Regarding erroneous neighborhoods in the graphs, almost 11%, 12% and 5% neighbors of the graphs for adjectives, nouns and verbs respectively are erroneous, which deteriorate the end-performance of ICE-NET by error propagation through attentive convolutions.

Considering the impact of different error categories on the end performance of ICE-NET. For multi-sense tokens the distributional embedding vectors are primarily oriented in the direction of the most prevalent sense of the underlying training corpora, which may be different from the sense in the word pair resulting in misclassifications. For example, “clean" and “blue" are two synonym words in the adjective dataset. Looking at the most similar words in the fasttext embeddings, we can see that the embedding vector for the word “blue" is more related to the colors, which makes it sense-wise different from the word “clean" which is more related to cleanliness. If we use these words to explain the properties of water, then these words are synonyms, however, it is not evident unless we explicitly consider the context along with word pair. Note, the phenomenon of multiple senses of a given word is more dominant among nouns compared with that of verbs and adjectives. This is also evident by a relatively lower performance of nouns relative to other word classes. For rare and OOV words, the embedding vectors are not adequately trained and their role in the end model is no better than the random vectors. This in turn limits the encoder networks of ICE-NET to encode relation-specific information.

7 Conclusion & Future Work

In this work we propose ICE-NET, which uses a set of interlaced encoder networks to capture the relation-specific properties of antonym and synonym pairs, i.e., symmetry, transitivity, and trans-transitivity, in order to perform antonyms vs synonyms distinction task. Results show that ICE-

NET outperforms the existing research by a relative score of up to 1.8% for F1-measure. Some promising future directions include: (i) using domain-specific text corpora along with training seeds, (ii) strategy to cut down the attention weights for the erroneous edges.

8 Limitations

Some of the core limitations of the ICE-NET are as follows:

1. Nouns and adjectives exhibit multiple different senses, which requires the need for the contextual information along with the word pair in order to model them. However, owing to unavailability of multi-sense data sets for the antonym vs synonym distinction task, current formulation of ICE-NET does not support multi-sense settings.
2. Erroneous edges in the adjacency graphs produced by M_{init} lead to error propagation. There is a need for an appropriate attention mechanism based on the semantics of the data.
3. The embeddings corresponding to the rare words and OOV tokens need to be initialized as a weighted average of semantically related tokens rather than random initialization.

Acknowledgements. Di Wang, Yan Hu and Muhammad Asif Ali are supported in part by the baseline funding BAS/1/1689-01-01, funding from the CRG grand URF/1/4663-01-01, FCC/1/1976-49-01 from CBRC and funding from the AI Initiative REI/1/4811-10-01 of King Abdullah University of Science and Technology (KAUST). Di Wang is also supported by the funding of the SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence (SDAIA-KAUST AI).

References

Heike Adel and Hinrich Schütze. 2014. Using mined coreference chains as a resource for a semantic task. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1447–1452.

Muhammad Ali, Maha Alshmrani, Jianbin Qin, Yan Hu, and Di Wang. 2023a. Gari: Graph attention for relative isomorphism of arabic word embeddings. In *Proceedings of ArabicNLP 2023*, pages 181–190.

Muhammad Ali, Yan Hu, Jianbin Qin, and Di Wang. 2023b. Gri: Graph-based relative isomorphism of word embedding spaces. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11304–11313.

Muhammad Asif Ali, Yifang Sun, Bing Li, and Wei Wang. 2020. Fine-grained named entity typing over distantly supervised data based on refined representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7391–7398.

Muhammad Asif Ali, Yifang Sun, Bing Li, and Wei Wang. 2021. Fine-grained named entity typing over distantly supervised data via refinement in hyperbolic space. *arXiv preprint arXiv:2101.11212*.

Muhammad Asif Ali, Yifang Sun, Xiaoling Zhou, Wei Wang, and Xiang Zhao. 2019. Antonym-synonym classification based on new sub-space embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6204–6211.

Marco Baroni and Sabrina Bisi. 2004. Using cooccurrence statistics and the web to discover synonyms in a technical language. In *LREC*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Mathias Etcheverry and Dina Wonsever. 2019. Unraveling antonym's word vectors through a siamese-like network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3297–3307.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *IJCAI*, volume 3, pages 1492–1493.

Anna Lobanova, Tom Van der Kleij, and Jennifer Spenader. 2010. Defining antonymy: A corpus-based study of opposites by lexico-syntactic patterns. *International Journal of Lexicography*, 23(1):19–53.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. *arXiv preprint arXiv:1605.07766*.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Distinguishing antonyms and synonyms in a pattern-based neural network. *arXiv preprint arXiv:1701.02962*.

Masataka Ono, Makoto Miwa, and Yutaka Sasaki. 2015. Word embedding-based antonym detection using thesauri and distributional information. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 984–989.

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

Michael Roth and Sabine Schulte Im Walde. 2014. Combining word patterns and discourse markers for paradigmatic relation classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 524–530.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pages 593–607. Springer.

Roy Schwartz, Roi Reichart, and Ari Rappoport. 2015. Symmetric pattern based word embeddings for improved word similarity prediction. In *CoNLL*, volume 2015, pages 258–267.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. *arXiv preprint arXiv:1603.06076*.

Lonneke Van der Plas and Jörg Tiedemann. 2006. Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 866–873.

Zhipeng Xie and Nan Zeng. 2021. A mixture-of-experts model for antonym-synonym discrimination. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 558–564.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

A Appendix

A.1 Justification for Attentive Convolutions

In this section, we provide intuitive explanations for: (a) the limitations posed by the distributional pre-trained word embeddings, and (b) why attentive graph convolutions are a better choice for capturing the relation-specific properties of the data, (c) computational efficency.

(a) Word Embeddings. We observe the nearest neighbours in the pre-trained word embeddings yield a blend of multiple different lexico-semantic relations and perform poorly on a specific task.

Underlying reason is the fact that the pre-trained word embeddings primarily rely on the distributional hypotheses, i.e., words sharing a similar context have similar meanings. From linguistic perspective, multiple words with varying relations (i.e., the antonyms and synonyms, hypernyms etc.,) may be used interchangeably within a fixed context. This in turn results these contextually similar words to be embedded close to each other. For example, nearest neighbours for the word “large" in the Glove embeddings are a combination of synonyms {“larger", “huge"}, antonyms {“small", “smaller"} and irrelevant words {“sized"} (Ali et al., 2019).

We argue that in order to refine information from the pre-trained embeddings for a specific task, the graphs provide a better alternative to analyze the words in combination with semantically related neighbours rather than instant-level modeling, as explained below.

(b) Attentive Graph Convolutions. The intuitive explanation for the attentive graph convolution network is to re-commute the representation of the word via attentive aggregation over the neighbouring words. The core idea is to surround each word by a set of semantically related neighbours during the graph construction process in order to smoothen the representation of the word.

It is based on the assumption that within the graphs, i.e., G_t and G_h, the neighbourhood of each word is dominated by its semantically relevant words compared to the antonyms and/or irrelevant words. And, recomputing the representation of each word by aggregating information from the neighbours will result in the final representation to more semantically coherent compared to the distributional embeddings, as the contribution of antonyms and other irrelevant words will be down-weighted. This is illustrated in Figure 2, where the

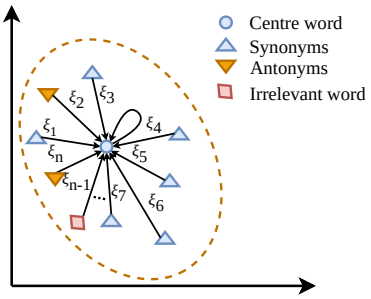

Figure 2: Illustration of attentive Graph Convolution Networks

representation of the centre word is recomputed using a combination of itself and its nearest neighbours (including synonyms, antonyms and irrelevant words). We use ξ_i as the attentive weight to control its degree of association for the i-th neighbor. The final representation of the word, i.e., the output of the attentive graph convolution network is later used for end-task, i.e., antonyms vs synonyms distinction.

(c) Computational Efficiency. Another noteworthy aspect is the computational efficiency of the attentive graph convolutions. Theoretically, for each layer the convolutions need to be computed between every word pair in the graphs which poses the following limitations: (a) it is time consuming and computationally inefficient, (b) accumulating information between all possible word pairs may incorporate noise in the model training and deteriorate the performance.

To circumvent that we use appropriate thresholds, i.e., ANT_{thr} and SYN_{thr}, to select only highly confident candidates for the graph construction. The values for these thresholds are computed empirically.

These thresholds are helpful in cutting down the un-necessary computations over the graphs (G_h and G_t) by limiting them to the neighbourhood $\mathcal{N}_i$ of each word i. Likewise the attention weights between word pairs (ξ_i) help in cutting down the noise by appropriately defining the contribution of the neighboring words. This setting is different from the graph convolution by Kipf and Welling (2016) that equally consider the contribution of the neighboring nodes in the graph.

A.2 Difference from R-GCN (Schlichtkrull et al., 2018)

Schlichtkrull et al. (2018) is proposed R-GCN, i.e. modeling the Relational data using the Graph Convolutional Networks, and used it for entity classifi-

cation and the link prediction task. Although, their problem settings for the link prediction task looks similar to ICE-NET, however, we emphasize some key differences as follows:

1. R-GCN uses GCN as an encoder to learn the representations followed by DistMult decoder (Yang et al., 2014) for link prediction. Note, this problem setting is different from ours, as R-GCN primarily deals with asymmetric relations which can be modeled by linear and/or bilinear transformations. On the other hand, ICE-NET deals with symmetric relations that cannot be modeled by the existing KG embedding methods, also shown in Figure 1(b).

2. Another justification in favour of the above-mentioned argument is the fact that currently the performance of the R-GCN is evaluated on KG embedding data sets, i.e., WN18, FB15k, and these data sets do not include symmetric relation pairs similar to antonym, synonym pairs etc.

3. R-GCN proposes relation-specific feature aggregation for the neighbouring nodes via a normalization sum. In contrast, we use attention weights to incorporate the impact of the degree of association of neighboring words/nodes.

4. ICE-NET is the first work that uses multiple encoders to capture the relation-specific properties of the antonym and synonym pairs (i.e., symmetry, transitivity and trans-transitivity), to eventually perform the distinction task in a performance-enhanced way.

A.3 Additional Data Sets

We also test the performance of ICE-NET on data sets other than the English. For this, we used antonym synonym pairs for the Urdu language also used by Ali et al. (2019). We acquired this data set from the authors of the Distiller (Ali et al., 2019). It is a relatively smaller data set encompassing approximately 750 instances, priorly splitted into 70% train, 25% test and 5% validation sets. For this data set, we used Fasttext embeddings (Grave et al., 2018) for Urdu as the pre-trained embeddings.

The experimental results in Table 6 show that ICE-NET outperforms the baseline models and Distiller by Ali et al. (2019) by significant margin.

Model	P	R	F1
Baseline-1 (Random Vectors)	0.687	0.653	0.670
Baseline-1 (w/o Graph conv.)	0.825	0.795	0.810
Distiller (Ali et al., 2019)	0.897	0.867	0.881
ICE-NET	**0.905**	**0.915**	**0.910**

Table 6: ICE-NET performance evaluation using

Specifically, it improve the F1-score by approximately 3.2% compared to the existing state-of-the art. These results also showcase the language-agnostic nature of ICE-NET. The same settings can be applied to the antonyms vs synonyms distinction task for multiple different languages provided with the availability of distributional embeddings and supervised training data.