future internet

MDPI

Article

Addressing Syntax-Based Semantic Complementation: Incorporating Entity and Soft Dependency Constraints into Metonymy Resolution

Siyuan Du and Hao Wang *

School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China; dusiyuan233@shu.edu.cn

* Correspondence: wang-hao@shu.edu.cn

Abstract: State-of-the-art methods for metonymy resolution (MR) consider the sentential context by modeling the entire sentence. However, entity representation, or syntactic structure that are informative may be beneficial for identifying metonymy. Other approaches only using deep neural network fail to capture such information. To leverage both entity and syntax constraints, this paper proposes a robust model EBAGCN for metonymy resolution. First, this work extracts syntactic dependency relations under the guidance of syntactic knowledge. Then the work constructs a neural network to incorporate both entity representation and syntactic structure into better resolution representations. In this way, the proposed model alleviates the impact of noisy information from entire sentences and breaks the limit of performance on the complicated texts. Experiments on the SemEval and ReLocaR dataset show that the proposed model significantly outperforms the state-of-the-art method BERT by more than 4%. Ablation tests demonstrate that leveraging these two types of constraints benefits fine pre-trained language models in the MR task.

Keywords: metonymy resolution; entity representation; dependency integration

Citation: Du, S.; Wang, H. Addressing Syntax-Based Semantic Complementation: Incorporating Entity and Soft Dependency Constraints into Metonymy Resolution. *Future Internet* **2022**, *14*, 85. https://doi.org/10.3390/fi14030085

Academic Editor: Carlos Filipe Da Silva Portela

Received: 22 January 2022
Accepted: 11 March 2022
Published: 12 March 2022

1. Introduction

Metonymy is a common figurative language phenomenon that refers to substituting the name of a thing using one of its closely associated attributes (e.g., producer-for-product, place-for-event, place-for-inhabitant). This linguistic phenomenon [1,2] is pervasive in daily life and literature. For example, named entities in the text are often used in a metonymy manner to imply an irregular denotation.

Identifying ambiguities in metonymy is a fundamental process in many NLP applications such as relation extraction [3], machine comprehension [4], and other downstream tasks [5]. The following sentence shows an example of metonymy: "Last year, Ma acquired Eleme". In this example, the meaning of "Ma" under this particular circumstance has been changed. It is more appropriate to interpret "Ma" as "Ma's company Alibaba" instead of the literate concept of "a famous entrepreneur".

In literature, conventional methods for MR mainly rely on the features derived from lexical resources such as dictionaries, taggers, parsers, and WordNet or other handcrafted lexical resources. At present, more researchers are focusing on deep neural networks (DNN) [6,7], which is becoming the mainstream approach to handle various tasks in NLP, including metonymy resolution [8]. DNN models can effectively encode all words in a sentence in a sequential way to obtain the semantic representation of the text to easily capture contextual information throughout the whole sentence and achieve state-of-the-art performance. Moreover, pre-trained language representations demonstrated efficiency in improving many NLP tasks, e.g., text classification [9], event extraction [10], and relation extraction [11]. Benefiting from the contextual representations, these models significantly surpass competitive neural models in many NLP tasks [12,13], for example

ELMo [14,15] and BERT [16]. Among these representations, the pre-trained model BERT has an especially significant impact. It has proven improved results in 11 NLP tasks. Compared to conventional DNN methods, BERT provides a simple way to accurately and effectively obtain the sentence-level semantic representation, which greatly benefits the task of MR. However, there are still shortcomings of using pre-trained language representations such as BERT for metonymy resolution. From a linguistic point of view, BERT is relatively weak in two perspectives at least:

Facing the entity-targeted task MR, BERT does not find a proper way to specify explicit information about the entities. For the absence of entity information in BERT, the proposed entity-perceptive method can effectively improve the efficiency under the joint guidance of entity and context knowledge. Based on this, Entity BERT with entity awareness is constructed by us. The model applies BERT to train the neural network jointly by the entity and context and integrates them to be better word embedding. This way, it can effectively use the entity knowledge to improve the precision and recall rate.

Compared with sequence-based models, dependency-based models can capture non-local syntactic relations that are clues for the inference of the true meaning of entity nominals. Thus, selectively integrating the dependency relations to alleviate the lack of syntax in BERT is necessary. Figure 1 shows an example of the dependency parse tree with multiple dependency relations. Given a sentence, it is convenient to obtain the dependency relations between two words by launching a dependency parser using NLP toolkits on the shift. For example, in Figure 1, it is evident that the "nsubj" relation between "Ma" and "acquired" is a clue that indicates the role "Ma" played in the sentence intensely while the "nmod" relation between "Last year" and "acquired" has rare contributions. However, most MR models treat these relations as equivalent and have not distinguished these relations, leading to misunderstandings of the lexicon and output incorrect semantic representations. Dependency trees convey rich structural information that is proven helpful for extracting relations among entities in text. Therefore, in this paper, dependency relations are used as the prior knowledge to supervise the model to learn metonymies. Soft structural constraints based on dependency parse trees are imposed on state-of-the-art pre-trained language representations such as BERT. This work builds on a rich line of recent efforts on relation extraction models and graph convolutional networks (GCN). Differing from the previous works that either try to directly use dependency parse trees or concatenate the transition vector based on dependency parse trees, this work employs attention-guided GCN(AGCN) to integrate information. Additionally, the outputs of the BERT layer are passed to attention-guided GCN to extract the syntactic information through dependency parsing. Then both syntactic and semantic information will be fed into the attention-guided GCN integration component that learns how to selectively attend to the relevant structures valid for MR. Therefore, the model takes advantage of the relevant dependency ancestors and efficiently removes the noise from irrelevant chunks. Finally, sentence-level representations, as well as the outputs of attention-guided GCN, are combined as the input to a multi-layer neural network for classification.

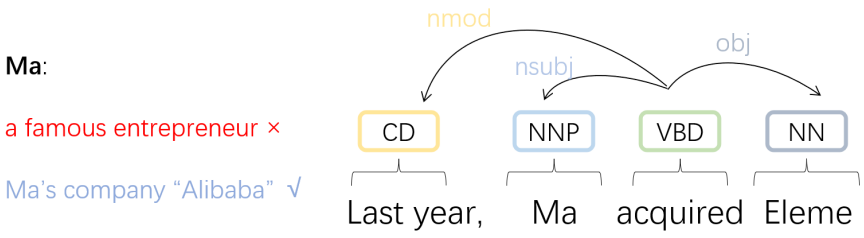


Figure 1. A metonymy example on the geographical parsing problem. According to linguistic valency in English grammar, the meaning of metonymic entity "Ma" is mainly decided by the root of the sentences, i.e., the verb "acquired". Dependency relations are the basis of understanding metonymy.

Considering entity awareness and dependency information both benefit MR, this paper presents EBAGCN (Entity BERT with Attention-guided GCN) to leverage entity constraints from sequence-based pre-trained language representations and soft dependency constraints from dependency trees at the same time.

As a result, the proposed approach is superior in capturing the semantics of the sentence and the long-distance dependency relations, which better benefits MR. To summarize, the main contributions of this paper are as follows:

- propose a novel method for metonymy resolution relying on recent advances on pre-trained language representations that integrate entity knowledge and significantly improve the accuracy.
- incorporate attention-guided GCN to MR with hard /soft dependency constraints, which imposes the pre-trained language representations with prior syntactic knowledge.
- experiments on two benchmark datasets show that the proposed model EBAGCN is significantly superior to previous works and improves the BERT-based baseline systems.

2. Related Work

Metonymy Resolution Analysis of metonymy as a linguistic phenomenon dates back to feature-based methods [17,18]. Ref. [19] use an SVM with a robust knowledge base built from Wikipedia to remain the best result in all feature-based methods. These methods inevitably suffer from error propagation because of their dependence on the manual feature extraction. Furthermore, the work takes much effort, while helpful information is hard to be caught. As a result, the performance is not satisfied.

Recently, the majority of works for MR have focused on the deep neural network (DNN). Ref. [8] propose PreWin based on Long Short Term Memory (LSTM) architecture. This work is the first to integrate structural knowledge into MR. In contrast, the effect is limited because of the setting of the predicate window, which only retains the words around the predicate while losing much important information in that process. Other approaches [20] leverage NER and POS features on LSTM to enrich the representation of tokens. However, these methods only improve slightly since they only pay attention to the independent tokens and ignore the relations between tokens in a sentence that is beneficial to MR.

Pre-trained language models have shown great success in many NLP tasks. In particular, BERT, proposed by [16], shows a significant impact. Intuitively, it is natural to introduce pre-trained models to MR. These models vastly outperform the conventional DNN models and reach the top of the leaderboard in MR. Nevertheless, pre-trained models encode the whole sentence to catch contextual features, leading to the ignorance of syntactic features in sentences. Ref. [21] fine-tunes BERT with target word masking and data augmentation to detect metonymy more accurately.

In addition to the context information provided by the sequence-based model, [22] pays attention to the entity word. They disambiguate every word in a sentence by reformulating metonymy detection as a sequence labeling task and investigate the impact of entity and context on metonymy detection.

Dependency Constraints Integration The research on MR so far has made limited application of dependency trees. However, research on other NLP classification tasks widely employ dependency information. Differing from traditional sequence-based models, dependency-based models integrate dependency information [23], taking advantage of seizing dependency relations that are obscure from the surface form alone.

As the effect of dependency information is widely recognized, more attention is paid to pruning strategies (i.e., how to distill syntactic information from dependency trees efficiently). Ref. [3] use the shortest dependency path between the entities in the full tree. Ref. [24] apply graph convolutional networks [25] model on a pruned tree and a novel pruning strategy to the input trees by retaining words immediately around the shortest path between entities among which a relation might hold. Although these hard-pruning

methods remove irrelevant relations efficiently based on predefined rules, they suffer from eliminating useful information wrongly at the same time.

More recently, ref. [26] proposed AGGCN and employed a soft-pruning strategy. The method enables the dependency relations to have weights to balance relevant and irrelevant information with multi-head attention mechanism. Ref. [27] proposed a dependency-driven approach for relation extraction with attentive graph convolutional networks (A-GCN). In this approach, an attention mechanism in graph convolutional networks is applied to different contextual words in the dependency tree obtained from an off-the-shelf dependency parser, to distinguish the importance of different word dependencies.

Pre-trained Model This is the idea of pre-training, originated in the field of computer vision, and then developed into NLP. A pre-trained word vector is the most common application of pre-training in NLP. The annotated corpus is very limited in many NLP tasks, which is not enough to train excellent word vectors. Therefore, large-scale unannotated corpus unrelated to the current task is usually used for pre-training word vectors. At present, many deep learning models tend to use pre-trained word vectors (such as Word2Vec [28] and GloVe [29], etc.) for initialization to accelerate the convergence speed of the network.

To consider contextual information when setting word vector, pre-trained models such as Context2Vec [30], ELMo [15] were developed and achieved good results. BERT is a model training directly on deep Transformer network. The best results were achieved in many downstream tasks of NLP through pre-training and fine-tuning [31]. Different from other deep learning models, BERT adjusts the context at all levels jointly before training to obtain bi-directional representation of each token. BERT solves the representation difficulty to a large extent by fine-tuning the output of a specific task. Compared with recurrent neural networks, BERT relying on Transformer can capture long-distance dependencies more effectively and have a more accurate semantic understanding of each token in the current context.

Graph Convolutional Network DNN models have achieved great success in both CV and NLP. As a representative model of deep learning, the convolutional neural network can solve regular spatial structures. While much data does not have structure, the graph convolutional (GCN) network arises at a historic moment. GCN is a widely used architecture to encode the information in a graph, where in each GCN layer, information in each node communicates to its neighbors through the connections between them. The effectiveness of GCN models to encode the contextual information over a graph of an input sentence has been demonstrated by many previous studies [32,33].

3. The Proposed Model

This section presents the basic components used for constructing the model. The overall architecture of the proposed model is shown in Figure 2.

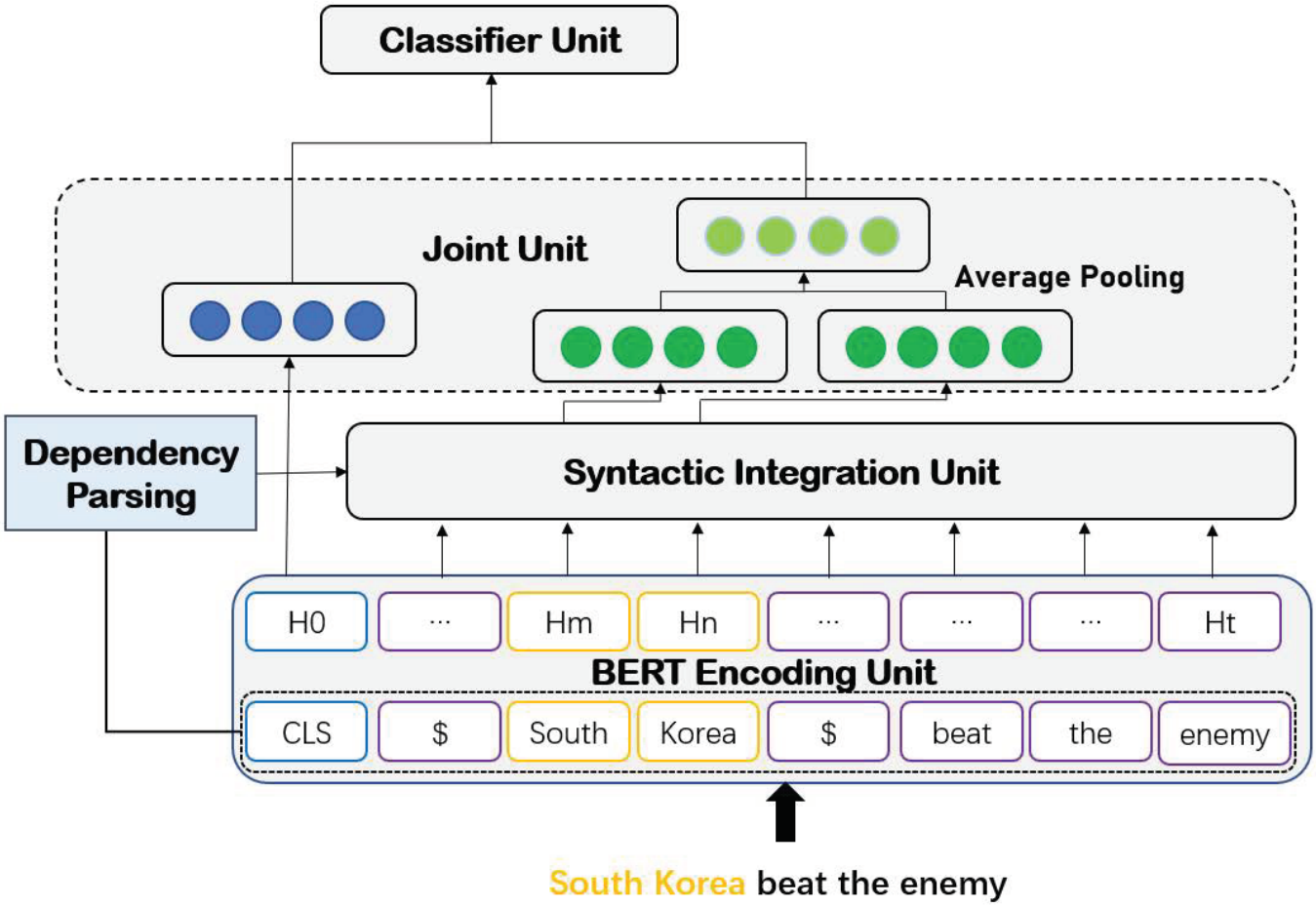


Figure 2. Model architecture. Better representation for the entity nominal "South Korea" is obtained based on the pre-trained language representation from the BERT unit by integrating the syntactic constraints using a soft dependency-based attention.

3.1. BERT Encoding Unit

The pre-trained language model BERT is a multi-layer bidirectional transformer encoder designed to pre-train deep bidirectional representations by conditioning both left and right context. This unit uses the BERT encoder to model sentences and output fine-tuned contextual representations. It takes as input sentence S, and computes for each token a context-aware representation. Concretely, the input packs as $[CLS, S_t, SEP]$, where CLS is a special token for classification; S_t is the token sequence of S generated by a WordPiece Tokenizer; SEP is the token indicating the end of a sentence. For each hidden representation h_i^0 at the index i, initial token embedding s_i^{tok} is concatenated with positional embedding s_i^{pos} as

$$h_i^0 = concat[s_i^{tok}; s_i^{pos}]. \quad (1)$$

After going through N successive Transformer encoder blocks, the encoder generates context-aware representations for each token to be the output of this unit, represented as h_i^N:

$$h_i^N = Transformers(h_i^0) \quad (2)$$

3.2. Syntactic Integration Unit

As is shown in Figure 3, the Syntactic Integration Unit is designed to integrate syntax into BERT and is the most crucial component of this approach. In a multi-layer GCN, the

node representation $h_i^{(l)}$ is produced by applying a graph convolution operation in layers from 1 to $l-1$, described as follows:

$$h_i^{(l)} = \rho(\sum_{j=1}^{n} A_{ij} W^{(l)} h_j^{(l-1)} + b^{(l)}) \tag{3}$$

where $W^{(l)}$ represents the weight matrix, $b^{(l)}$ stands for the bias vector, and ρ is an activation. $h^{(l-1)}$ and $h^{(l)}$ are the hidden state in prior and current layer, respectively.

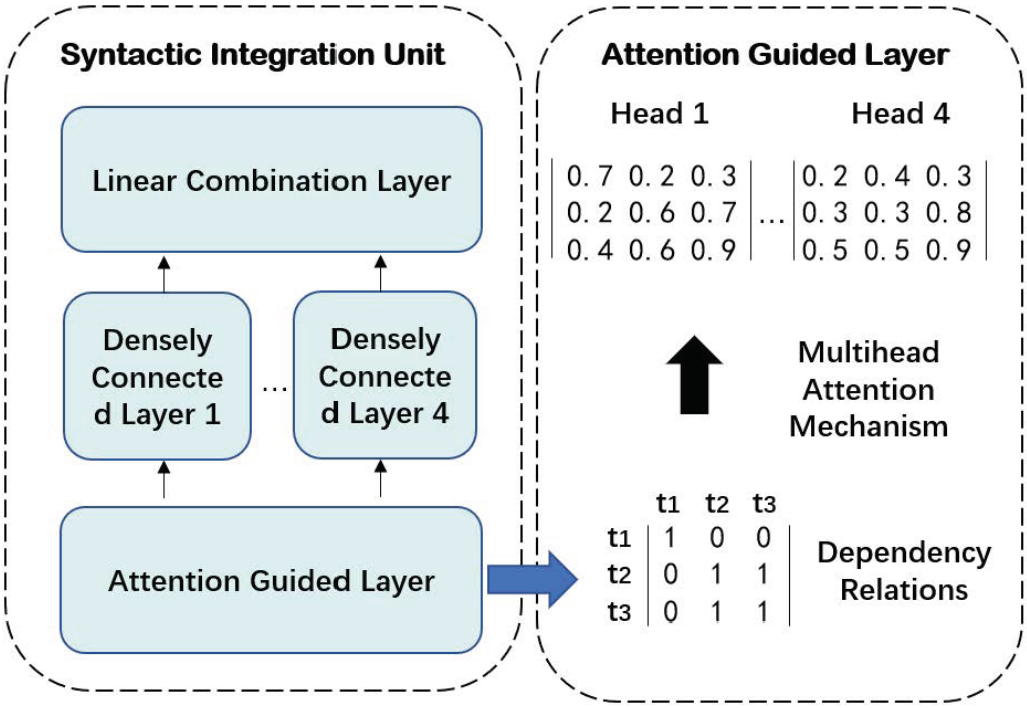


Figure 3. The structure in Syntactic Integration Unit.

Syntactic Integration Unit contains attention guided layer, densely connected layer, and linear combination layer.

Attention Guided Layer Most existing methods adopt hard dependency relations (i.e., 1, 0 denote relation exists or not) to impose syntactic constraints. However, these methods require the pre-defined pruning strategy based on expert experience and simply set the dependency relations considered "irrelevant" as zero-weight (not attended). These rules may bias representations, especially toward a larger dependency graph. Reversely, the attention guided layer helps to launch the "soft pruning" strategy. This layer generally generates the attention guided adjacency matrix $A^{(t)}$ whose weights range from 0 to 1 by multi-head attention [34]. The shape of $A^{(t)}$ is the same as the original adjacency matrix A for convenience. Precisely, $A^{(t)}$ is calculated as follows:

$$A^{(t)} = softmax(\frac{QW_i^Q \times (KW_i^K)^T}{\sqrt{d}})V \tag{4}$$

where Q, K, V are, respectively, query, key, value in multi-head attention, Q, K are both equal to the input representation $R^{(m-1)}$ (i.e., output of the prior module), d is the dimension of $R^{(m-1)}$, W_i^Q and W_i^K are both learnable parameters $\in \mathbb{R}^{d \times d}$, $A^{(t)}$ is the t-th attention guided adjacency matrix corresponding to the t-th head.

In this way, the attention guided layer outputs a large fully connected graph to reallocate the importance of each dependency relation rather than pruning the graph into a smaller structure as tradition.

Densely Connected Layer This layer helps to learn more local and non-local information and train a deeper model using densely connected operations. Each densely connected layer has L sub-layers. L is a hyper-parameter for each module. These sub-layers are placed in regular sequence, and each sub-layer takes all preceding sub-layers' output as input. The structure of Densely Connected Layer is shown in Figure 4. $g_j^{(l)}$ is calculated in this layer, which is defined as the concatenation of the initial representation and the representations produced in each preceding sub-layer:

$$g_j^{(l)} = [x_j, h_j^{(1)}, ...h_j^{(l-1)}], \tag{5}$$

where x_j is initial input representation, $h_j^{(1)}, ...h_j^{(l-1)}$ are the outputs of all preceding sub-layers. In addition, the dimension of representations in these sub-layers is shrunk to improve the parameter efficiency, i.e., $d_{hidden} = d/L$, where L is the number of sub-layers, d is the input dimension. For example, the number of sub-layer is 2 and input dimension is 1024, $d_{hidden} = d/L = 512$. Then a new representation whose dimension is $1024(512 \times 2)$ is formed by concatenating all these sub-layer outputs. N densely connected layers compute N adjacency matrixes produced by attention guided layer. The GCN computation for each sub-layer should be modified because of the application of multi-head attention:

$$h_{t_i}^{(l)} = \rho(\sum_{j=1}^{n} A_{ij}^{(t)} W_t^{(l)} g_j^{(l)} + b_t^{(l)}) \tag{6}$$

where t represents t-th head, $W_t^{(l)}$ and $b_t^{(l)}$ are learnable weights and bias, which are selected by t and associated with the attention guided adjacency matrix $A^{(t)}$.

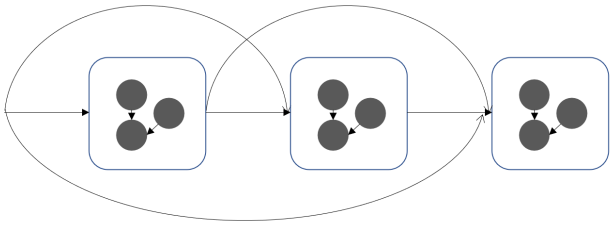


Figure 4. Dense connected architecture of Densely Connected Layer.

Linear Combination Layer In this layer, the final output is obtained by combining representations output by N Densely Connected Layer corresponding to N heads:

$$h_{out} = W_{out} h_{in} + b_{out}, \quad h_{in} = [h^{(1)}, ...h^{(N)}] \tag{7}$$

where $h_{out} \in \mathbb{R}^d$ is the combined representation of N heads as well as the output of the module. W_{out} and b_{out} are learnable weights and bias.

3.3. Joint Unit

In Joint Unit, the context representation and entity representation are united to form the final joint representation for MR.

Context Representation BERT encoder produces the final hidden state sequence H corresponding to the task-oriented embedding of each token. According to the BERT mechanism, the representation H_0 output by the special token "[CLS]" serves as the pooled representation of the whole sentence. Therefore, H_0 serves to represent the aggregate sequence as context representation.

Entity Representation To help the model capture the clues of entities and enhance the expression ability, entity indicator is inserted at the beginning and end of the entity.

The entity represents as follows: suppose that $H_m \ldots H_n$ are the hidden states of entity E output by Syntactic Integration Unit (m, n represent the start index and end index of the entity, respectively), an average operation is applied to obtain a final representation:

$$H_e = \frac{1}{n-m+1} \sum_{t=m}^{n} H_t \tag{8}$$

Representation Integration For classifying, model concatenate H_0 and H_e and consecutively apply two full connected layers with activation.

$$H_{final} = \rho(W^*[\rho(W^{'} concat[H_0; H_e] + b^{'}]) + b^*]) \tag{9}$$

3.4. Classifier Unit

A softmax layer is applied to produce a probability distribution $p(y|x, \theta)$:

$$p(y \mid x, \theta) = softmax(H_{final}) \tag{10}$$

θ refers all learnable parameters in the network $W^{'} \in \mathbb{R}^{d_h \times d_h * 2}$, $W^* \in \mathbb{R}^{r \times d_h}$, where r is the number of classification types, d_h is the dim of BERT representation.

4. Methodologies and Materials

4.1. Dataset

The experiments are conducted on two publicly available benchmarks: the SemEval2007 [18] and ReLocaR [8] datasets. Unlike WiMCor [35] and GWN [36] that contains huge amount of instances, SemEval2007 and ReLocaR are relatively smaller. The samples lay into two classes: literal and metonymic. SemEval contains 925 training and 908 test instances, while ReLocaR comprises a train (1026 samples) and a test (1000 samples) dataset. The class distribution of SemEval is approx 80% literal, 20% metonymic. To eliminate the high class bias of SemEval, the class distribution of ReLocaR sets to be 50% literal, 50% metonymic.

Since the MR task is still in its infancy, there is no available MR dataset for Chinese. English datasets SemEval and ReLocaR are employed to construct a Chinese MR dataset through text translation, manual adjustment, and labeling.

1. Text translation: examples of SemEval and ReLocaR are translated using API on the Internet and finally obtain independent Chinese samples;
2. Manual adjustment: Considering the poor quality of the dataset obtained from API, all examples are corrected and well selected to meet the Chinese expression norms;
3. Labeling: Inserting a pair of indicators to mark the entity.

After the above steps, an MR dataset called CMR in Chinese is constructed to verify the model performance on Chinese texts. Finally, the dataset contains 1986 entity-tagged instances, of which 1192 are randomly divided as a training set and 794 as a test set. Each instance contains a sentence with the entity tag and a classification tag of literal or metonymic.

4.2. Dependency Parsing

Given a sentence from the dataset, first the sentence is tokenized with the tokenization tool "jieba". Then, dependency parsing is launched for the tokenization list by the tool of Stanford CoreNLP [37]. After the dependency graph is output by dependency parsing, the dependency relations are first encoded into an adjacency matrix A. In particular, if there is a dependency edge existing between node i and j, then $A_{ij} = 1$ and $A_{ji} = 1$, otherwise $A_{ij} = 0$ and $A_{ji} = 0$.

4.3. Model Construction

The proposed models are encoded with Python 3.6 and deep learning framework PyTorch 1.1. They are trained on a Tesla v100-16GB GPU. EBAGCN requires approx. 1.5 times GPU memory compared with vanilla BERT-LARGE.

Given a sentence S with an entity E, MR aims to predict whether E is a metonymic entity nominal. The key idea of EBAGCN is to enhance BERT representation with structural knowledge from dependency trees and entities. Generally, the entire sentence will first go through the BERT Encoding Unit to obtain the deep bidirectional representation for each token. Then, launching dependency parsing to extract the dependency relations from each sentence. Subsequently, both deep bidirectional representations and dependency relations are fed into the Syntactic Integration Unit. The achieved vector representations are enriched by syntactic knowledge and integrated with context representation in Joint Unit. Finally, the fused embedding is served to produce a final prediction distribution in the Classifier Unit.

4.4. Model Setup

For both datasets, the batch size is set as 8 and number of training epochs as 20. The number of head for multi-head attention N is chosen from $\{1, 2, 4, 8\}$, max sequence length len from $\{128, 256\}$, initial learning rate for AdamW lr from $\{5 \times 10^{-6}, 1 \times 10^{-5}, 2 \times 10^{-5}\}$. The combination ($N = 4, len = 256, lr = 1 \times 10^{-5}$), ($N = 4, len = 128, lr = 2 \times 10^{-5}$) and ($N = 8, len = 128, lr = 2 \times 10^{-5}$) give the best results on SemEval, ReLocaR and CMR, respectively.

The evaluation for experiments are accuracy, precision, recall, and F1.

- **accuracy** The probability of the total sample that predicted correct results;
- **precision** The probability of actually being a positive sample of all the predicted positive samples;
- **recall** The probability of being predicted to be a positive sample in a sample that is actually positive;
- **F1** The F1 score is balanced by taking into account both accuracy and recall. The expression of F1 score is:

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{11}$$

5. Results

5.1. Models

The baseline models used in the experiment are listed below.

SVM+Wikipedia: SVM+Wikipedia is the previous SOTA statistical model. It applies SVM with Wikipedia's network of categories and articles to automatically discover new relations and their instances.

LSTM and BiLSTM: LSTM is one of the most potent dynamic classifiers publicly known [38]. Because of the featured memory function of remembering last hidden states, it achieves promising results and is widely used in various NLP tasks. Moreover, BiLSTM improves the token representation by being aware of the conditions from both directions [39], making contextual reasoning available. Additionally, two kinds of representations, GloVe [29] and ELMo [15] are performed separately to ensure a credible model result.

Paragraph, Immediate, and PreWin: These three models are primarily built upon BiLSTM. They simultaneously encode tokens as word embeddings and dependency tags as one-hot vectors (5–10 tokens in general). The difference between them is in the way of picking tokens. Immediate-y selects y number of words on the left and right side of the entity as input to the model [40,41]. The Paragraph model extends the Immediate model by taking the 50 words from the side of each entity as the input to the classifier. The PreWin model relies on a predicate window consisting of a direct vocabulary around the recognized predicate to eliminate noise over a long distance.

fastText: FastText is a tool that computes the word vector and classifies the text without great academic innovation. However, its advantages are obvious. In text classification, fastText can achieve performance similar to the deep network with little training cost.

CNN: The experiment applies the classical CNN model target for the text classification, which is composed of input layer, convolution layer, pooling layer, and softmax layer. Since the whole model adapts to the text (rather than CNN's traditional application: image), some adjustments are made to adapt the NLP task.

BiLSTM+Att: BiLSTM+Att applies an attention layer on the BiLSTM to increase the representation ability.

BERT: BERT is a language model trained on deep Transformer networks which performs NLP tasks well through pre-training and fine-tuning, and achieves the best results in many downstream tasks of NLP [31,42].

5.2. Main Result on SemEval and ReLocaR

On SemEval and ReLocaR, this approach compares with the feature-based model, deep neural network model and pre-trained language model. Table 1 reports the results.

Table 1. Result on English datasets. On both benchmarks, the F1-score for either class and the overall accuracy are given in this table. "L" and "M" denote literal and metonymic class, respectively. +NER+POS means integrating both NER and POS features with the baseline model. In general, due to the application of advanced pre-trained language representation along with soft dependency clues, EBAGCN obtains best results.

MODEL	SemEval			ReLocaR		
	F1-L	F1-M	Acc(std)	F1-L	F1-M	Acc(std)
SVM+Wikipedia [19]	91.6	59.1	86.2(N/A)	-	-	-
LSTM (GLoVE) [43]	85.2	28.7	72.6 (1.48)	78.4	78.4	78.4(0.91)
+NER+POS	87.5	27.3	77.4(1.34)	80.6	80.6	80.6(0.92)
BiLSTM (GLoVE) [43]	83.2	37.4	75.4(1.72)	82.9	83.0	82.9(0.85)
+NER+POS	88.8	37.7	82.0(1.36)	84.2	84.2	84.2(0.69)
Paragraph [8]	-	-	81.3 (0.88)	-	-	80.0(2.25)
Immediate 5 [8]	-	-	81.3 (1.11)	-	-	81.4(1.34)
Immediate 10 [8]	-	-	81.9 (0.89)	-	-	81.3(1.44)
PreWin [8]	90.6	57.3	83.1 (0.64)	84.4	84.8	83.6(0.71)
BiLSTM (ELMo) [43]	91.9	54.7	86.3(0.45)	90.0	90.1	90.0(0.40)
+NER+POS	91.6	55.6	86.1(0.47)	90.1	90.1	90.1(0.36)
BERT [16]	-	-	84.7(0.71)	-	-	91.3(0.54)
Entity BERT (this work)	93.2	66.0	88.8(0.63)	95.2	95.3	95.3(0.44)
EBGCN (this work)	93.5	67.5	89.1(0.60)	95.5	95.5	95.5(0.46)
EBAGCN (this work)	**94.0**	**68.3**	**89.8(0.85)**	**95.7**	**95.7**	**95.7(0.34)**

Three self-constructed models are compared in the experiment: Entity BERT (BERT model integrating the entity information by joint representation but discarding the syntactic constraints), EBGCN (Entity BERT with GCN, apply normal GCN without attention to impose hard syntactic constraints on Entity BERT), EBAGCN (Entity BERT with Attention-guided GCN, apply attention-guided GCN to impose soft syntactic constraints on Entity BERT).

The result in Table 1 shows that the models in this work significantly outperform previous SOTA model SVM+Wikipedia which is based on feature engineering. They also surpass all the DNN models including LSTM, BiLSTM, and PreWin, even if they incorporate POS and NER features to enrich the representation. This result illustrates that the conventional features cannot provide enough contextual information. Entity BERT and BERT are both pre-trained language models. However, there are significant differences in

effect due to the incorporation of entity constraint, which greatly improves the accuracy of the model.

Furthermore, the three models also perform differently. The experiment on EBGCN produces a decent accuracy that is 0.3% and 0.2% higher than Entity BERT on SemEval and ReLocaR, which illustrates that the application of GCN helps improve performance by catching the ignored information from syntax. Moreover, EBAGCN obtains an improvement of 0.7% and 0.2% compared with EBGCN in terms of accuracy. This fact provides ample proof that the introduction of the multi-head attention mechanism assists GCNs in learning better information aggregations by simultaneously pruning irrelevant information and emphasizing dominating relations concerning indicators such as verbs in a soft method.

The table also gives the F1-score for literal and metonymic results, respectively. The consequence shows that EBAGCN achieves the best F1-score on SemEval (metonymic accounts for 20%) and ReLocaR (metonymic accounts for 50%), which suggests that EBAGCN is adaptive in various class distributions.

To be specific, Entity BERT uses the BERT-based neural network to aggregate information about context and entity semantics to form better semantic vector representation. In this way, the model leverages entity words and enhances the interaction between entity words and context information, thus improving the accuracy and recall rate of MR. The improved result of Entity BERT verifies the importance of entity information and proves that Entity BERT can effectively solve the missing entity information in metonymy.

In the framework of cognitive linguistics, syntactic structures are considered to contain important information. Existing models based on DNN scan the information encoding of the whole sentence sequence and compress it into vector expression, which cannot capture the syntactic structure that plays an important role in the transmission of natural language information. In addition, all syntactic dependencies are added to the syntactic representation vector with the same weight, so it is impossible to distinguish the contribution of each dependency. Thus, EBAGCN steps further to leverage syntax knowledge selectively. The weight allocation system for syntax dependencies of EBAGCN do not just resist noise interference, but also improve the accuracy of MR. Finally, the proposed EBAGCN can effectively solve the low accuracy of long and difficult sentences as well as key word recognition.

5.3. Main Result on CMR

Unlike the English dataset ReLocaR and SemEval, Chinese text is harder to understand. Table 2 gives the result on CMR. As can be seen from the experimental results, fastText, CNN, and BiLSTM+Att have little gap in the Chinese MR task. BERT greatly improves the performance by relying on the powerful ability of the pre-trained model, and the Entity BERT proposed in this paper achieves a better result by reinforcing the entity information. Compared with Entity BERT, EBGCN is about 1.2% higher in accuracy, showing a huge performance improvement and proving the significance of dependency knowledge. However, taking advantage of syntactic noise elimination, EBAGCN obtains the SOTA result on CMR, proving the validity of this work.

Table 2. Result on self-constructed Chinese dataset CMR.

MODEL	Acc	Precision	Recall	F1-L	F1-M
fastText [44]	70.0	70.1	70.1	71.6	68.8
CNN [45]	73.1	73.1	73.1	73.3	73.3
BiLSTM+Att [43]	73.1	73.4	73.4	73.9	72.4
BERT [16]	81.7	81.6	81.6	75.4	87.7
Entity BERT (this work)	85.3	85.0	85.0	78.4	**91.9**
EBGCN (this work)	86.5	86.6	86.6	**90.8**	82.6
EBAGCN (this work)	**87.4**	**87.4**	**87.4**	88.3	86.5

6. Discussion

6.1. Entity Ablation Experiment

The validity of the Entity BERT was demonstrated in the main result above. This experiment mines further to understand the specific contributions of each module besides the pre-trained BERT component. Take Entity BERT as baseline, this work proposes three additional models:

- **Entity BERT NO-SEP-NO-ENT** discard both entity representation H_e and the entity indicators around the entity, i.e., only representation corresponding to "[CLS]" is used for classification;
- **Entity BERT NO-SEP** only discard the entity representation H_e but reserve entity indicators around the entity;
- **Entity BERT NO-ENT** only discard the entity indicators around the entity but reserve entity representation H_e.

Table 3 shows the results of the ablation experiment. From the table, all three methods perform worse than Entity BERT. Among them, Entity BERT NO-SEP-NO-ENT performs the worst, proving that both entity indicator and entity representation make great contributions to the model. The meaning of using the entity indicator is to integrate entity location information into the BERT pre-trained model. On the other hand, entity representation further enriches the information and helps the model achieve high accuracy.

Table 3. Ablation analysis on two datasets: ReLocaR and SemEval.

MODEL	SemEval (Acc)	ReLocaR (Acc)
Entity BERT NO-SEP-NO-ENT	84.7	84.7
Entity BERT NO-ENT	86.4	94.5
Entity BERT NO-SEP	87.0	94.6
Entity BERT	**89.2**	**95.5**

Entity BERT enhances the influence of entity information on discriminant results and provides a solid foundation for accurately representing the joint embedding of entity and context. In MR task, most of the key information is focused on entity. The integrity of entity representation largely determines the performance of the model. Making full use of the semantic, location and structural information of entity words can effectively reduce the influence of noise. Therefore, as entity information in metonymy text is difficult to be extracted and represented, a fusion perception method is applied in this paper, which can effectively improve the accuracy and efficiency of MR under the joint guidance of entity and context knowledge. Based on the above fashion, this paper puts forward Entity BERT that jointly training BERT with entity and context. In that case, Entity BERT makes use of the key information of the entity to reduce the influence of the noise in the sentence, Thus, the accuracy and recall rate are greatly improved.

6.2. Entity Contribution Verification

To further study the contribution of entity information on MR, this experiment inputs a single entity representation H_e into the BERT model without using contextual information representation H_0. This work maps semantic representation of several entities into Figure 5.

As shown in the left picture of Figure 5, before BERT is fine-tuned, the semantic representation of the entity is not far from the original, indicating that entity information is very sparse and weak without entity fine-tuning. However, as shown in the right figure, after BERT fine-tuning, metonymic and literal entities are divided into two clusters, which shows that the fine-tuned model can judge the metonymy.

The existing deep learning model depends on the context, whose representation is formed by inputting all the tokens in the whole sentence into the fully connected layer, inevitably containing a lot of noise. This experiment proves that integrating entity in-

formation into the language model helps the task of MR and verifies the rationality of Entity BERT.

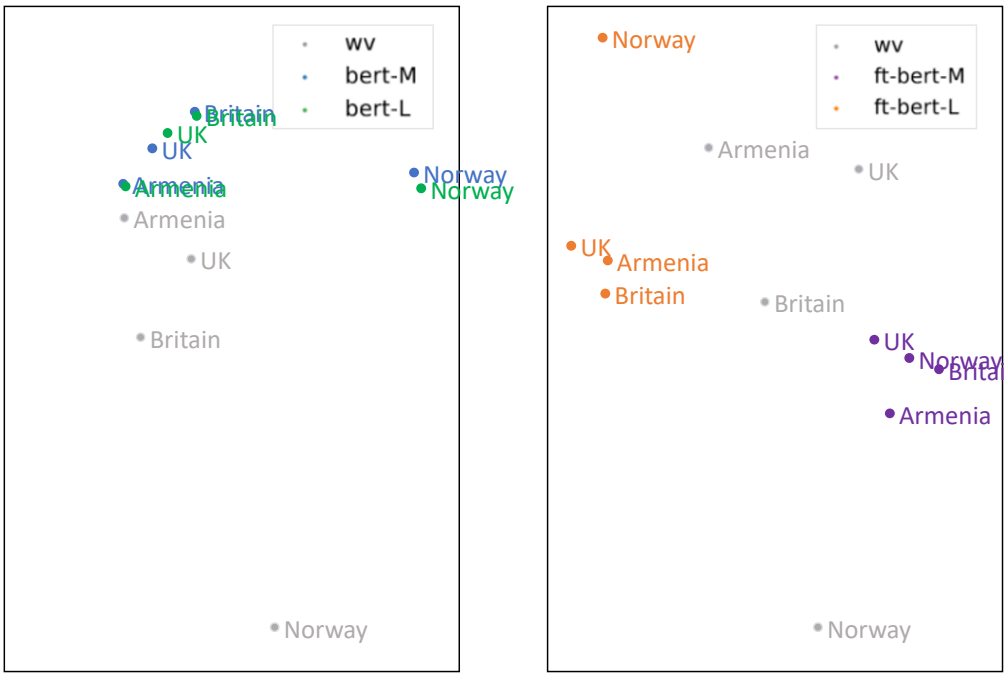


Figure 5. Semantic distribution maps of BERT before (**left**) and after fine-tuning (**right**). "WV" refers to the original semantics of entities, "bert-M" refers to the semantic representation of metonymic entities in BERT before fine-tuning, "bert-L" refers to the semantic representation of literal entities in BERT before fine-tuning, "ft-bert-M" refers to the semantic representation of metonymy entities in BERT after fine-tuning, "Ft-bert-l" refers to the semantic representation of literal entities in BERT after fine-tuning.

6.3. Comparison w.r.t Sentence Length

Figure 6 further compares the accuracy of EBAGCN and Entity BERT under different sentence lengths.

The causes of poor model performance may be more than one. For example, long sentences are likely to affect the accuracy of classification for the following reasons:

- Contextual meanings for long sentences are more difficult to capture and represent.
- The position of key tokens, such as a predicate, is noisy and, therefore, difficult to determine.

Intuitively, lacking model interpretability of non-local syntactic relations, sequence-based models such as Entity BERT cannot sufficiently capture long-distance dependence. Thus, the accuracy of Entity BERT drops fiercely as predicted as shown in Figure 6 when the sentence length grows. However, such a performance degradation can be alleviated using EBAGCN, which suggests that catching a mass of non-local syntactic relations helps the proposed model accurately infer the meaning of the entity, especially in longer sentences.

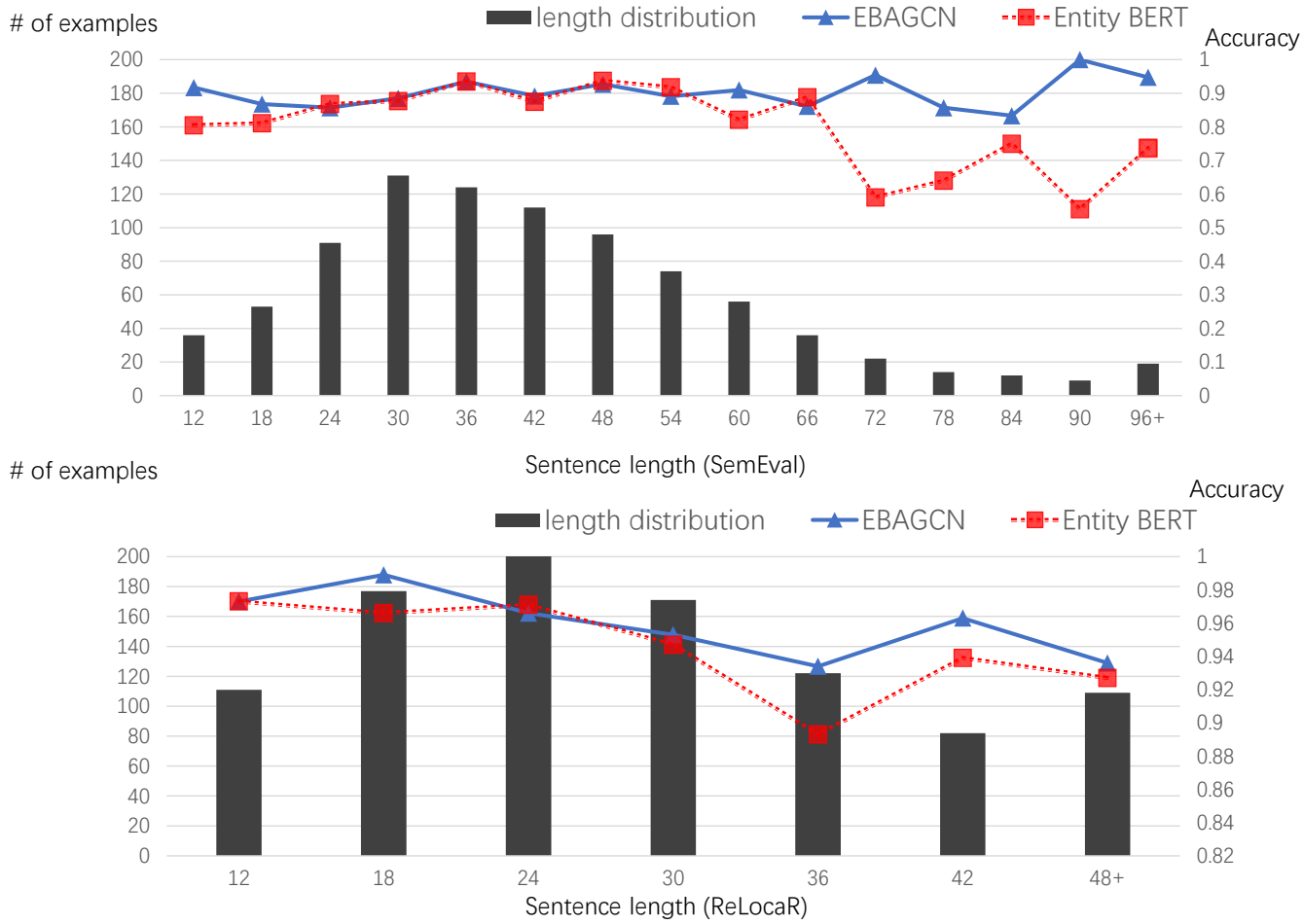


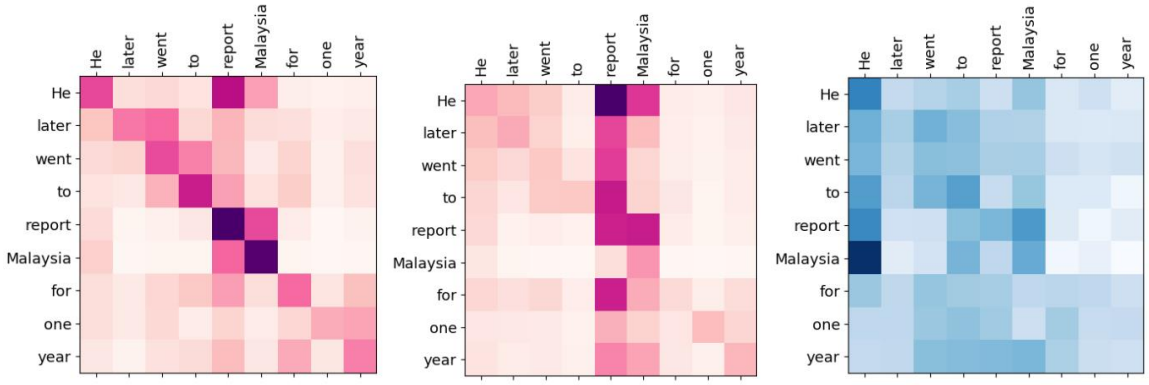
Figure 6. Experiment on ReLocaR and SemEval datasets w.r.t different sentence lengths.

The proposed EBAGCN solve the problem of complex sentence patterns and difficult syntactic understanding. By means of GCN based on the attention mechanism, the semantic and syntactic representation of the context is jointly trained. GCN effectively help integrate syntactic knowledge into vector representation, while attention mechanism highlights the expression of key information in dependency, which eliminates the syntactic noise to a certain extent.

6.4. Attention Visualization

This experiment provides a case study, using a motivating example that is correctly classified by EBAGCN but misclassified by Entity BERT, to vividly show the effectiveness of the proposed model. Given the sentence "He later went to report Malaysia for one year", people can easily distinguish "Malaysia" as a metonymic entity nominal by extending "Malaysia" as a concept of "a big event in Malaysia". Nevertheless, from the semantic perspective independently, the verb phrase "went to" is such a strong indicator that Entity BERT is prone to recognize "Malaysia" as a literal territory (for the regular usage of "went to someplace") falsely and overlooks the true predicate "report". How EBAGCN resolves the problems mentioned above is explained by visualizing the attention weights in the model.

First, the attention matrix of Transformer encoder blocks is compared in BERT Encoding Unit to display the syntactic integration's contribution to the whole model. Figure 7a,b shows that the weight for tokens in Entity BERT is more decentralized, while EBAGCN concentrates on "report" and "Malaysia" rather than "went to" thanks to the application of syntactic features, indicating that with the help of syntactic component, EBAGCN can better pick valid token and discard the irrelevant even misleading chunks.



(**a**) (**b**) (**c**)

Figure 7. Attention matrix visualization: (**a**) weights in BERT Encoding Unit (Entity BERT), (**b**) weights in BERT Encoding Unit (EBAGCN), (**c**) weights in Attention Guided Layer (EBAGCN).

The Proposed Model section shows that the Attention Guided Layer transfers the hard dependency matrix into an attention guided matrix, which enables the syntactic component to select relevant syntactic constraints efficiently. Thus, this work further displays the attention guided matrix to demonstrate the superiority of soft dependency relations. As shown in Figure 7c, after launching the multi-head mechanism, despite the existence of dependency relations for the prepositional phrase "for one year", the weights for these relations are pretty futile compared with the main clause that includes verb and other determining features for judging relations in prepositional phrase useless to the MR task. This approach sets the model free from pre-defined pruning strategy and automatically obtains high-quality relations.

7. Findings and Limitations

The main findings of this paper are as follows:

- integrating entity clues into the MR model better helps the model to represent an entity completely.
- MR depends on syntax to clarify the sentence structure. Under these circumstances, GCN inputs syntactic knowledge into the model completely and efficiently. Furthermore, soft pruning strategy helps to remove the noise in syntactic dependency relations.
- the proposed method can be generalized for different word-level NLP tasks, such as event extraction and relation extraction, indicating the great prospects of this research.

However, there are also some limitations need to be solved in the future:

- dependency directions and types (such as nsubj, nmod) should be added to the model to make the model more robust.
- the constraints mined in this paper all come from sentence. However, many metonymies are proper nouns that denote existing well-known works or events, suffering from the limitations of lacking "real world" and common sense knowledge to access the referents. External knowledge bases could be an appropriate choice to overcome this problem.

8. Conclusions

This paper proposed an entity- and syntax-aware approach to metonymy resolution based on pre-trained BERT, which substantially outperforms existing methods over a broad range of datasets. This work further demonstrated that the proposed end-to-end metonymy resolution model can improve the performance of more complicated and longer sentences. The experiments also evaluated the performance of different models in hard/soft dependency setting, and showed the proposed method to generalize the best. This constraint

integration method can be applied to tasks beyond metonymy resolution. Numerous word-level classification tasks such as relation classification lack high-quality, balanced datasets. Thus, the proposed approach can be applied to these tasks to contribute to the research community.

Author Contributions: Conceptualization, H.W. Methodology, S.D. Writing—original draft preparation, S.D. Writing—review and editing, S.D. and H.W. Supervision, S.D. and H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: SemEval: http://web.eecs.umich.edu/~mihalcea/affectivetext/#resources (accessed on 10 March 2022); ReLocaR: https://github.com/milangritta/Minimalist-Location-Metonymy-Resolution/tree/master/data (accessed on 10 March 2022).

Acknowledgments: We thank the anonymous reviewers for their comments. This work was supported in part by Shanghai Science and Technology Young Talents Sailing Program 21YF1413900, National Natural Science Foundation of China under grant 91746203, 61991410 and National Key R&D Program of China under grant 2018AAA0102804.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lakoff, G.; Johnson, M. Conceptual metaphor in everyday language. *J. Philos.* **1980**, *77*, 453–486.
2. Lakoff, G. Image metaphors. *Metaphor. Symb.* **1987**, *2*, 219–222.
3. Xu, Y.; Mou, L.; Li, G.; Chen, Y.; Peng, H.; Jin, Z. Classifying relations via long short term memory networks along shortest dependency paths. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1785–1794.
4. Seo, M.; Kembhavi, A.; Farhadi, A.; Hajishirzi, H. Bidirectional attention flow for machine comprehension. *arXiv* **2016**, arXiv:1611.01603.
5. Zhou, J.; Cao, Y.; Wang, X.; Li, P.; Xu, W. Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 371–383.
6. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation classification via convolutional deep neural network. In Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 2335–2344.
7. Zhang, D.; Wang, D. Relation classification via recurrent neural network. *arXiv* **2015**, arXiv:1508.01006.
8. Gritta, M.; Pilehvar, M.T.; Limsopatham, N.; Collier, N. Vancouver welcomes you! minimalist location metonymy resolution. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 1248–1259.
9. Sun, C.; Qiu, X.; Xu, Y.; Huang, X. How to fine-tune BERT for text classification? In Proceedings of the China National Conference on Chinese Computational Linguistics, Kunming, China, 18–20 October 2019; pp. 194–206.
10. Tian, C.; Zhao, Y.; Ren, L. A Chinese event relation extraction model based on BERT. In Proceedings of the 2nd International Conference on Artificial Intelligence and Big Data, Chengdu, China, 25–28 May 2019; pp. 271–276.
11. Wu, S.; He, Y. Enriching Pre-trained Language Model with Entity Information for Relation Classification. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2361–2364.
12. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642.
13. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv* **2016**, arXiv:1606.05250.
14. Peters, M.; Ammar, W.; Bhagavatula, C.; Power, R. Semi-supervised sequence tagging with bidirectional language models. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1756–1765.
15. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237.

16. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
17. Farkas, R.; Simon, E.; Szarvas, G.; Varga, D. GYDER: Maxent Metonymy Resolution. In Proceedings of the 4th International Workshop on Semantic Evaluations, Prague, Czech Republic, 23–24 June 2007; pp. 161–164.
18. Markert, K.; Nissim, M. SemEval-2007 Task 08: Metonymy Resolution at SemEval-2007. In Proceedings of the 4th International Workshop on Semantic Evaluations, Prague, Czech Republic, 23–24 June 2007; pp. 36–41.
19. Nastase, V.; Strube, M. Transforming Wikipedia into a large scale multilingual concept network. *Artif. Intell.* **2013**, *194*, 62–85.
20. Lee, J.; Seo, S.; Choi, Y.S. Semantic Relation Classification via Bidirectional LSTM Networks with Entity-Aware Attention Using Latent Entity Typing. *Symmetry* **2019**, *11*, 785.
21. Li, H.; Vasardani, M.; Tomko, M.; Baldwin, T. Target Word Masking for Location Metonymy Resolution. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 3696–3707.
22. Mathews, K.A.; Strube, M. Impact of Target Word and Context on End-to-End Metonymy Detection. *arXiv* **2021**, arXiv:2112.03256.
23. Bunescu, R.; Mooney, R.J. A shortest path dependency kernel for relation extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Vancouver, BC, Canada, 6–8 October 2005; pp. 724–731.
24. Zhang, Y.; Qi, P.; Manning, C.D. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2205–2215.
25. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
26. Guo, Z.; Zhang, Y.; Lu, W. Attention Guided Graph Convolutional Networks for Relation Extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 241–251.
27. Tian, Y.; Chen, G.; Song, Y.; Wan, X. Dependency-driven Relation Extraction with Attentive Graph Convolutional Networks. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Virtual Event, 1–6 August 2021. pp. 4458–4471.
28. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3111–3119.
29. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
30. Melamud, O.; Goldberger, J.; Dagan, I. context2vec: Learning generic context embedding with bidirectional lstm. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016; pp. 51–61.
31. Lin, C.; Miller, T.; Dligach, D.; Bethard, S.; Savova, G. A BERT-based universal model for both within-and cross-sentence clinical temporal relation extraction. In Proceedings of the 2nd Clinical Natural Language Processing Workshop, Minneapolis, MN, USA, 7 June 2019; pp. 65–71.
32. Chen, G.; Tian, Y.; Song, Y.; Wan, X. Relation Extraction with Type-aware Map Memories of Word Dependencies. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online Event, 1–6 August 2021; pp. 2501–2512. Available Online: https://aclanthology.org/2021.findings-acl.221.pdf (accessed on 10 March 2022).
33. Sun, K.; Zhang, R.; Mao, Y.; Mensah, S.; Liu, X. Relation extraction with convolutional network over learnable syntax-transport graph. In Proceedings of the AAAI Conference on Artificial Intelligence, Hilton New York Midtown, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8928–8935.
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
35. Mathews, K.A.; Strube, M. A Large Harvested Corpus of Location Metonymy. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 5678–5687.
36. Gritta, M.; Pilehvar, M.T.; Collier, N. A pragmatic guide to geoparsing evaluation. *Lang. Resour. Eval.* **2020**, *54*, 683–712.
37. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.R.; Bethard, S.; McClosky, D. The Stanford CoreNLP natural language processing toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 23–24 June 2014; pp. 55–60.
38. Sundermeyer, M.; Schlüter, R.; Ney, H. LSTM neural networks for language modeling. In Proceedings of the 13th Annual Conference of the International Speech Communication Association, Portland, OR, USA, 9–13 September 2012.
39. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
40. Mesnil, G.; He, X.; Deng, L.; Bengio, Y. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In Proceedings of the 14th Annual Conference of the International Speech Communication Association, Lyon, France, 25–29 August 2013; pp. 3771–3775.
41. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P.P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
42. Qu, C.; Yang, L.; Qiu, M.; Croft, W.B.; Zhang, Y.; Iyyer, M. BERT with history answer embedding for conversational question answering. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 1133–1136.

43. Zhang, S.; Zheng, D.; Hu, X.; Yang, M. Bidirectional long short-term memory networks for relation classification. In Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, Shanghai, China, 30 October–1 November 2015; pp. 73–78.
44. Joulin, A.; Grave, E.; Mikolov, P.B.T. Bag of Tricks for Efficient Text Classification. *arXiv* **2016**, arXiv:1607.01759.
45. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1746–1751.