Natural Language Engineering (2023), **29**, pp. 669–692
doi:10.1017/S135132492200033X

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

An empirical study of incorporating syntactic constraints into BERT-based location metonymy resolution

Hao Wang[1], Siyuan Du[1], Xiangyu Zheng[1] and Lingyi Meng[2,*]

[1]School of Computer Engineering and Science, Shanghai University, Shanghai, China and [2]School of Foreign Languages, East China Normal University, Shanghai, China
*Corresponding author. E-mail: lymeng@fl.ecnu.edu.cn

(Received 22 November 2020; revised 11 June 2022; accepted 13 June 2022; first published online 1 August 2022)

Abstract

Metonymy resolution (MR) is a challenging task in the field of natural language processing. The task of MR aims to identify the metonymic usage of a word that employs an entity name to refer to another target entity. Recent BERT-based methods yield state-of-the-art performances. However, they neither make full use of the entity information nor explicitly consider syntactic structure. In contrast, in this paper, we argue that the metonymic process should be completed in a collaborative manner, relying on both lexical semantics and syntactic structure (syntax). This paper proposes a novel approach to enhancing BERT-based MR models with hard and soft syntactic constraints by using different types of convolutional neural networks to model dependency parse trees. Experimental results on benchmark datasets (e.g., RELOCAR, SEMEVAL 2007 and WIMCOR) confirm that leveraging syntactic information into fine pre-trained language models benefits MR tasks.

Keywords: Syntax; Anaphora resolution; Information retrieval

1. Introduction

Metonymy is a type of figurative language that is pervasive in literature and in our daily conversation. It is commonly used to refer to an entity by using another entity closely associated with that entity (Lakoff and Johnson 1980; Lakoff 1987; Fass 1988; Lakoff 1991, 1993; Pustejovsky 1991). For example, the following two text snippets show a word with literal usage and metonymic usage:

(1) *President Vladimir Putin arrived in* ***Spain*** *for a two-day visit*.
(2) ***Spain*** *recaptured the city in 1732*.

In the first sentence, the word '*Spain*' refers to the geographical location or a country located in extreme southwestern Europe. However, in the second sentence, the meaning of the same word has been redirected to an irregular denotation, where '*Spain*' is a metonymy for '*the Spanish Army*' instead of its literal reading of the location name.

In natural language processing (NLP), *metonymy resolution* (MR) (Markert and Nissim 2002; Nissim and Markert 2003; Nastase and Strube 2009; Gritta *et al*. 2017; Li *et al*. 2020) is a task aimed at resolving metonymy for named entities. MR attempts to distinguish the word with metonymic usage from literal usage given that word in an input sentence, typically location or organisation names. MR has been shown to be potentially helpful for various NLP applications such as machine

translation (Kamei and Wakao 1992), relation extraction (RE) (Chan and Roth 2011) and geographical parsing (Monteiro, Davis, and Fonseca 2016; Gritta *et al.* 2017; Li *et al.* 2020). While other types of metonymies exist, in this paper, we are only interested in a specific type of conventional (regular) metonymy, namely, *location metonymy*. The task of *location metonymy resolution* (Markert and Nissim 2002; Gritta *et al.* 2017; Li *et al.* 2020) constitutes classifying a location name within the given sentence into *metonymic* or *literal* class.

Although many named entity recognition (NER) systems and word sense disambiguation (WSD) systems exist, these systems generally do not explicitly handle metonymies. NER systems only identify entity names from a sentence, but they are not able to recognise whether a word is used metonymically. Existing WSD systems only determine which fixed 'sense' (interpretation) of a word is activated from a close set of interpretations, whereas metonymy interpretation is an open problem. They cannot infer the metonymic reading of a word out of the dictionary. Lakoff and Johnson (1980) and Fass (1988) found that metonymic expressions mainly fell into several fixed patterns, most of which were quite regular. Therefore, recent methods for MR are mainly structured into two phases (Markert and Nissim 2002): *metonymy detection*[a] and *metonymy interpretation* (Nissim and Markert 2003). Metonymy detection attempts first to distinguish the usage of entity names between metonymic and literal. Then, metonymy interpretation determines which fine-grained metonymic pattern it involves such as *place-for-people* or *place-for-event*. The difference between metonymy detection and metonymy interpretation can be seen as from a coarse-grained (binary, metonymic or literal) to fine-grained (a particular type of metonymic expression) classification (Mathews and Strube 2020).

In computational linguistics, conventional feature-based methods for location MR (Nissim and Markert 2003; Farkas *et al.* 2007; Markert and Nissim 2007, 2009; Brun, Ehrmann, and Jacquet 2007; Nastase and Strube 2009; Nastase *et al.* 2012) rely heavily on handcrafted features delivered from either linguistic resources or off-the-shelf taggers and dependency parsers. These methods struggle with the problem of data sparsity and heavy feature engineering. Later, deep neural network (DNN) models (Mikolov *et al.* 2013; Gritta *et al.* 2017; Mathews and Strube 2020) become mainstream in handling various NLP tasks, including MR. These models have better performances since they take more contextual information into account. Although DNN models provide a giant leap forward compared to feature-based methods, training high-performance DNN models requires large-scale and high-quality datasets. However, existing datasets for MR are rather small because the cost of collecting and annotating datasets is very expensive and unaffordable. This situation raises a need to transfer the knowledge from existing large-scale datasets. Recently, pre-trained language models (PLMs), especially BERT (Devlin *et al.* 2019), have shown superior performance on various NLP downstream applications (Sun, Huang, and Qiu 2019; Qu *et al.* 2019; Lin *et al.* 2019b). The main advantage of PLMs is that they do not need to be trained from scratch. When applying PLMs to a specific dataset, only some additional fine-tuning is required, which is much cheaper. Benefiting from being pre-trained on a large-scale dataset with efficient self-supervised learning objectives, PLMs can efficiently capture the syntax and semantics in the text (Tang *et al.* 2018; Jawahar, Sagot, and Seddah 2019). Therefore, it is natural to adopt BERT to generate entity representations for MR tasks.

However, directly adopting BERT into MR tasks might encounter problems. While BERT has a strong advantage in modelling lexical semantics and generates informative token embeddings, BERT has difficulty in fully modelling completed syntactic structures as it might need deeper layers to capture long-distance dependencies (Tang *et al.* 2018; Zhang, Qi, and Manning 2018; Jawahar *et al.* 2019). Given the sentence '*He later went to manage* ***Malaysia*** *for one year*', BERT tends to focus more on the former verb '*went*' and ignore the latter verb '*manage*', which might

[a]Metonymy detection is also called *metonymy recognition* by Nissim and Markert (2003).

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

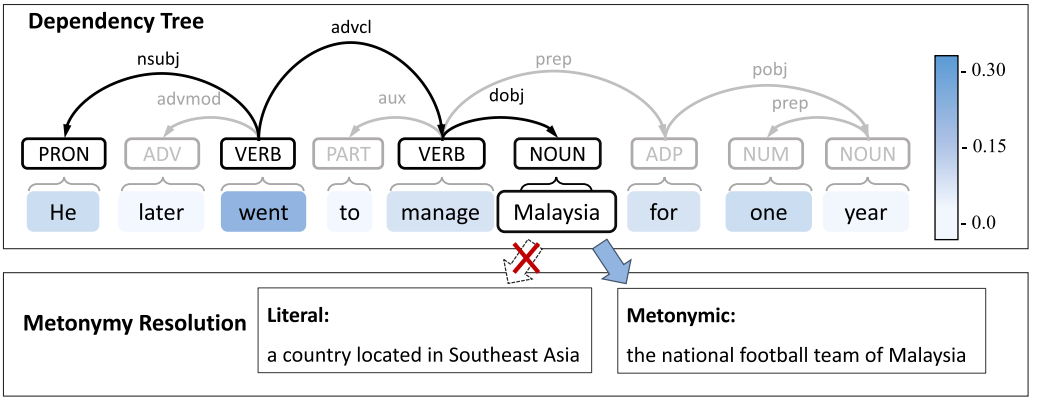

Figure 1. An example illustrates that syntactic information helps metonymy resolution. '*Malaysia*' is metonymically used. The words in deeper blue colour have higher weights in the BERT's attention. Since the sentence above contains two verbs, it is confusing to infer metonymy. While '*manage*' strongly suggests a metonymic reading, BERT currently has not addressed that verb. The attention weight for '*went*' is higher than that of '*manage*'.

lead to incorrect prediction of the MR label for '*Malaysia*'.[b] As shown in Figure 1, dependency parse trees that convey rich structural information might help to recognise the metonymic usage. Therefore, syntactic knowledge is necessary for improving BERT-based MR models.

Previous studies (Nissim and Markert 2003; Nastase and Strube 2009; Nastase *et al.* 2012) suggested that syntax was a strong hint in constructing metonymy routes. Both the lexical semantics and the syntactic structure (specifically, dependency relations) jointly assisted in recognising novel readings of a word. In a metonymic sentence, the target entity is artificially violated its fixed usage in order to introduce a novel metonymic reading, which was traditionally treated as syntactico-semantic violation (Hobbs and Martin 1987; Pustejovsky 1991; Chan and Roth 2011). Generally, an entity is an argument to at least one predicate, there exist explicit syntactic restrictions on the entity and the predicate. In other words, the inference of metonymic reading primarily relies on the selectional preferences of verbs (Fass 1988). As shown in Figure 1, '*Malaysia*' refers to the national football team of Malaysia. The verbs and dependency arcs among verbs (coloured in a dark colour) were a strong clue to that metonymy, while other words (coloured in grey) had less contribution. This motivated us to explore an interesting question: Can jointly leveraging lexical semantics and syntactic information for MR can bring benefits?

As a part of ongoing interest in introducing prior syntactic knowledge into DNNs and PLMs, this paper investigates different ways to incorporate hard and soft syntactic constraints into BERT-based location MR models, following the idea that lexical semantics are potentially helpful for MR. Firstly, we employ an entity-aware BERT encoder to obtain entity representations. To force the model to focus on the target entity for prediction, we leverage explicit entity location information by inserting special entity markers before and after the target entity of the input sentence. Then, to take advantage of relevant dependencies and eliminate the noise of irrelevant chunks, we adopt two kinds of graph convolutional neural networks to impose hard and soft syntactic constraints on BERT representations in appropriate ways. Finally, the model selectively aggregates syntactic and semantic features to be helpful for MR inference. As a result, the proposed approach shows state-of-the-art (SOTA) performances on several MR benchmark datasets. To the best of our knowledge, this work is the first attempt to integrate syntactic knowledge and contextualised embeddings (BERT) for MR in an end-to-end deep learning framework.

[b] Although we know that '*Malaysia*' is metonymically used, the resolution of the metonymy here is unclear without further contextual information. Possible resolutions include as the national football team of Malaysia, as a department of a multi-national business.

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

2. Background: Metonymy resolution

Previous research in cognitive linguistics (Fundel, Küffner, and Zimmer 2007; Janda 2011; Pinango *et al.* 2017) revealed that metonymic expressions are based on actual, well-established transfer relations between the source entity and the target referent, while those relations were not expected to be lexically encoded. Metonymy consists of a natural transfer of the meaning of concepts (Kvecses and Radden 1998) which evokes in the reader or listener a deep 'contiguity' process. For instance, in a restaurant a waitress says to another, 'Table 4 *asked for more beer*', which involves a lesser-conventionalised *circumstantial metonymy*.[c] A large amount of knowledge is necessary to interpret this kind of metonymy (Zarcone, Utt, and Padó 2012), for example, Table 4 cannot ask for beer, but the guests occupying Table 4 can. In contrast to circumstantial metonymy, systematic metonymy (also called conventional metonymy) is more regular (Nissim and Markert 2003), for example, producer-for-product, place-for-event and place-for-inhabitant. Such reference shifts occur systematically with a wide variety of location names. As '*Malaysia*' refers to the national football team, as shown in Figure 1, the name of a location often refers to one of its national sports teams. It is easy to apply supervised learning approaches to resolve systematic metonymies by distinguishing between literal readings and a pre-defined set of metonymic patterns (e.g., place-for-event, place-for-people and place-for-product). Since the metonymic patterns place-for-event and place-for-product are rather rare in the natural language, the majority of MR works focus on the prediction of place-for-people.

Empirical methods for MR mainly fall into feature-based and neural-based approaches. Most feature-based methods (Brun *et al.* 2007; Nastase and Strube 2009; Nastase *et al.* 2012) based on statistical models are commonly evaluated on the SemEval 2007 Shared Task 8 benchmark (Markert and Nissim 2007). Markert and Nissim (2002) were the first to treat MR as a classification task. Nissim and Markert (2003) extracted more syntactic features and showed that syntactic head-modifier relations were important clues to metonymy recognition. Brun *et al.* (2007) presented a hybrid system combining a symbolic and an unsupervised distributional approach to MR, relying on syntactic relations extracted by the syntactic parsers. Farkas *et al.* (2007) applied a maximum entropy classier to improve the MR system on the extracted feature set. Nastase and Strube (2009) expanded the feature set used in Markert and Nissim (2007) with more sophisticated features such as WordNet 3.0 and WikiNet (Wikipedia's category network). Nastase *et al.* (2012) explored the usage of local and global contexts for the task of MR in a probabilistic framework. Nastase and Strube (2013) used a support vector machine with a large-scale knowledge base built from Wikipedia. These feature-based methods suffer from error propagation due to their high dependence on the extraction process of handcrafted features. Furthermore, constructing the feature set requires external NLP tools and extra pre-processing costs.

Recently, the majority of MR models incorporate DNNs. Gritta *et al.* (2017) first applied a BiLSTM neural network, called PreWin, to extract useful features from the predicate windows. During encoding, PreWin retained only the words and corresponding dependency labels within the predicate to eliminate noise in context. Rather than using BERT as a classifier after fine-tuning, Mathews and Strube (2020) proposed to leverage BERT as an encoder to initialise word embeddings lightly; then, they fed the BERT's embeddings into the PreWin system to perform MR.

PLMs have shown great success in many NLP tasks. Those models can produce context-aware and tokenwise pre-trained representations on a large-scale unlabelled dataset. They are fine-tuned on a downstream task and do not need to learn parameters from scratch. These models, such as ELMo (Peters *et al.* 2017; Peters *et al.* 2018) and BERT (Devlin *et al.* 2019), considerably surpass competitive neural models in many NLP tasks (Socher *et al.* 2013; Rajpurkar *et al.* 2016). PLMs obtained new SOTA results on language understanding tasks and are outstanding in capturing contextual or structural features (Glavaš and Vulić 2021). Intuitively, introducing pre-trained models to MR tasks is a natural step. Li *et al.* (2020) first attempted to use the BERT framework

[c] *Circumstantial metonymy* is also known as *unconventional metonymy* (Nissim and Markert 2003).

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

for MR tasks directly. Given the vast range of entities in the world, it is impossible to learn all entity mentions. To address data sparsity and force the model to make predictions based only on context, Li *et al*. (2020) proposed a word masking approach based on BERT by replacing all target entity names with an [*X*] token during training and inference. The masking approach substantially outperformed existing methods over a broad range of datasets.

Despite their successes, they did not investigate the role of syntax and how syntax affects MR. However, identifying the metonymic usage of an entity should collaboratively rely on both the entity and the syntax. The above issue motivated us to concentrate on modelling dependency associations among words that may be potentially helpful for MR to enrich BERT representations.

3. Related works

Since entity names are often used in a metonymic manner, MR has a strong connection with other NLP tasks such as WSD and RE. These tasks share similar pre-processing techniques and neural network architectures in utilising syntactic information (Joshi and Penstein-Rosé 2009; Li *et al*. 2014; Peng *et al*. 2017; Zhang *et al*. 2018; Fu, Li, and Ma 2019). Integrating dependency relations with DNN models has shown promising results for various NLP tasks (Joshi and Penstein-Rosé 2009; Li *et al*. 2014; Peng *et al*. 2017; Zhang *et al*. 2018; Fu *et al*. 2019). However, the effect of dependency integration for neural-based MR models is still not recognised and has made limited progress so far.

With recent advances in RE (Zhang *et al*. 2018; Wu and He 2019; Guo, Zhang, and Lu 2019), we investigate the use of dependency integration for MR. Our first concern is the integration approach, whether directly concatenating dependency embeddings with token embeddings or imposing dependency relations using a graph model is more appropriate. Extensive works have discussed this issue, and most of them treated dependency relations as features. For example, Kambhatla (2004) trained a statistical classifier for RE by combining various lexical, syntactic and semantic features derived from the text in the early data pre-processing stage. Zhang, Zhang, and Su (2006) studied embedding syntactic structure features in a parse tree to help RE. As a result, those models were sensitive to linguistic variations, which prevented further applying the dependency integration approach.

Recent research employs graph-based models to integrate DNNs and dependency parse trees. A variety of hard pruning strategies relying on pre-defined rules have been proposed to distil dependency information that improves the performance of RE. For example, Xu *et al*. (2015) used the shortest dependency path between the entities in the entire tree. Liu *et al*. (2015) combined the shortest dependency path between the target entities using a recursive neural network and attached the subtrees to the shortest path with a convolutional neural network. To leverage hierarchy information in dependency parse trees, Miwa and Bansal (2016) performed bottom-up or top-down computations along the parse tree or the subtree below the lowest common ancestor (LCA) of the entities. Zhang *et al*. (2018) pruned words except for the immediate ones around the shortest path, given that those words might hold vital information to hint at the relation between two target entities. They applied graph convolutional network (GCN) to model the dominating dependency tree structures. Although these hard pruning methods remove irrelevant relations efficiently, some useful information may also be eliminated. To resolve the above conflicts, Guo *et al*. (2019) proposed a soft pruning method called AGGCN (attention-guided graph convolutional network), a model that pools information over dependency trees by using GCN. They transform original dependency trees into fully connected edge-weighted graphs, balancing the weights of dependency relations between including and excluding information. Note that dependency-guided approaches, such as Zhang *et al*. (2018) and Guo *et al*. (2019), worked on the RE task. To the best of our knowledge, we are the first to incorporate syntactic constraints into BERT-based models for MR.

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

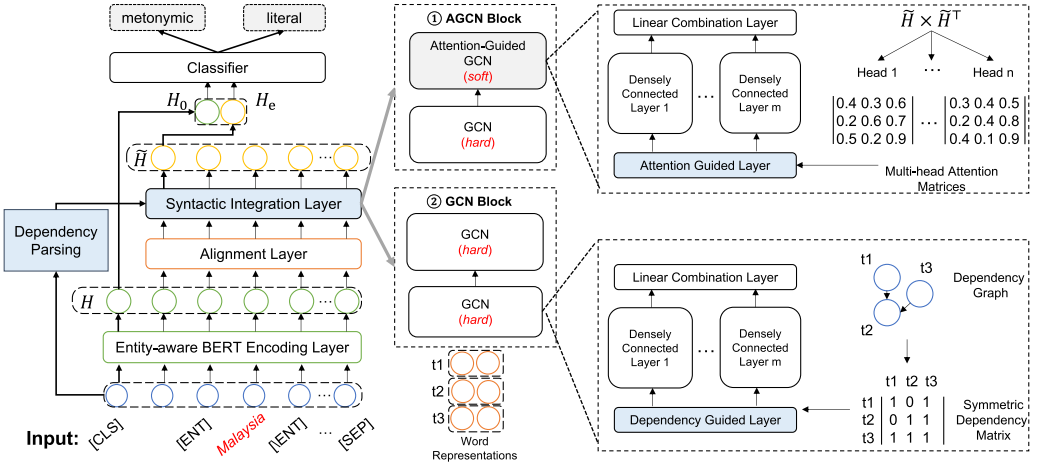

Figure 2. Illustration of the architecture of the proposed model with syntactic integration. It can distinguish metonymic usage of the entity name '*Malaysia*' given the enriched representation by incorporating hard and soft syntactic constraints using GCN and AGCN blocks. In this model, both the context and entity semantics are considered to resolving metonymies.

4. Proposal

The task addressed in this paper is MR. Given an entity name E within a sentence S, MR predicts whether E involves a metonymic or literal usage. The critical insight of this paper is that incorporating syntactic constraints may help BERT-based MR. As shown in Figure 1, the closest governing verb in the dependency parse tree plays a dominant role in resolving metonymies. Therefore, we consider that lexical semantics and syntactic structure essential for identifying metonymies.

Figure 2 illustrates the overall architecture of the proposed model. We propose an end-to-end neural-based approach for MR tasks and train the model based on recent advances in PLMs. Since BERT has shown superior performance on various NLP tasks, we employ BERT as an input encoder to produce tokenwise semantic representations by passing the input sentences through the BERT encoder. To enrich these tokenwise representations with syntactic knowledge given dependency parse trees, we propose two ways to incorporate syntactic constraints using different types of GCNs, for example, non-attentive GCN and attentive GCN (AGCN). We first perform dependency parsing for input sentences to extract corresponding dependency parse trees and then convert those parse trees into dependency adjacency matrices. Then, we use the GCN to encode dependency adjacency matrices explicitly. However, vanilla GCNs represent the adjacency edges among nodes using hard 0 and 1 labels. To learn these weights, following Guo *et al.* (2019), we adopt the self-attention mechanism (Vaswani *et al.* 2017) upon GCNs to tune the weights. As a result, the final representations contain rich syntactic knowledge, and lexical semantics serve to make predictions.

4.1 Entity-aware encoding with BERT

BERT, consisting of multi-layer bidirectional transformers, is designed to produce deep bidirectional representations. The BERT encoding layer uses a multi-layer transformer encoder to generate a sentence-level representation and fine-tuned contextual token representations for each token. We omit a detailed architecture description of BERT and only introduce the primary part of the entity-aware BERT encoder. Concretely, we pack the input as $[CLS, S_t, SEP]$, where $[CLS]$ is a unique token for classification, S_t is the token sequence of S generated by a WordPiece Tokenizer and $[SEP]$ is the token indicating the end of the sentence. Our model takes the packed sentence S as input and computes context-aware representations. Following Wu and He (2019), which enriches

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

BERT with entity information for relation classification, we insert special [*ENT*] indicators before and after the entity nominal. This simple approach lets BERT easily locate the MR entity position. For each h_x^0 at the index x, we concatenate initial token embeddings with positional embeddings and segment embeddings as follows:

$$h_x^0 = concat[S_x^{tok}; S_x^{pos}; S_x^{seg}]. \tag{1}$$

After going through N successive transformer encoder blocks, the encoder generates entity-aware BERT representation at the x-th position represented by h_x^N as follows:

$$h_x^N = BERT(h_x^0) \tag{2}$$

4.2 Alignment

BERT applies WordPiece Tokenizer (a particular type of subword tokenizer) to further segment words into word pieces, for example, from '*played*' to ['*play*', '*##ed*']. However, dependency parsing relies on words and hence does not execute further segmentation. Thus, we need to align BERT's tokens against the input words and restore word representations by adopting the average pooling operation on BERT's token representations. Assume $h_x, \ldots, h_y$ are BERT representations of tokens (x and y represent the start and end indices of the token sequence), we obtain the embedding $\tilde{h}_i$ of the i-th word by[d]:

$$\tilde{h}_i = \frac{1}{y-x+1} \sum_{t=x}^{y} h_t \tag{3}$$

4.3 Syntactic integration

Our model requires both the GCN and AGCN layers for data processing purposes. The nonattention GCNs are inserted before the AGCNs to impose dependency graphs explicitly. Then, the AGCNs learn the soft weights of edges in the dependency graph. The syntactic integration layer enriches the final representations with dependency relation information, making them both context- and syntax-aware.

Although the GCN layer is similar to the AGCN layer in architecture, the main difference between the former and the latter is whether the attention matrix A is initialised by directly using the dependency adjacency matrix A or computing multi-head self-attention scores given $\tilde{H}$. The choice of A is dependent on the present application of the GCN or AGCN, which can be expressed as follows:

$$\tilde{A} = \begin{cases} A, & \text{if GCN} \\ \varphi(\tilde{H} \times \tilde{H}^{\mathsf{T}}), & \text{if AGCN} \end{cases} \tag{4}$$

φ is a soft attention function, such as additive (Bahdanau, Cho, and Bengio 2015), general dot-product (Luong, Pham, and Manning 2015) or scaled dot-product (Vaswani *et al.* 2017) attention. Therefore, the attention-guided layer composes both the attentive and nonattentive modules. We use the scaled dot-product attention in our model for efficiency.

[d]Let $H = [h_1, \ldots, h_n]$ denotes a sentence. A more effective approach is to construct a mapping matrix M to project BERT's token representations H into the full sentence representations, that is, word representations $\tilde{H}$. The projection matrix M records the transformation from the original words to the subwords, which can be served as a router to restore word-wise representations as $\tilde{H} = HM^{\mathsf{T}}$, where M is the projection matrix where $M \in \mathbb{R}^{m \times n}$. m and n denote the length of the input sentence in word and in token after tokenisation.

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

4.3.1 Dependency-guided layer

Many existing methods have adopted hard dependency relations (i.e., 1 or 0 indicates whether the association exists or not, respectively) to impose syntactic constraints. These methods require handcrafted rules based on expert experience. Moreover, hard rules set dependency relations considered as *irrelevant* as zero weights (not attended), which may develop biased representations, especially towards sparser dependency graphs.

We adapt the graph convolution operation to model dependency trees by converting each tree into its corresponding adjacency matrix A. In particular, $A_{ij} = A_{ji} = 1$ if there exists a dependency edge between word i and word j; otherwise, $A_{ij} = A_{ji} = 0$. Empirically, a self-loop is necessary for an addition to edges. Then, in a multi-layer GCNs, the node (word) representation $\tilde{h}_i^{(l)}$ is produced by applying a graph convolution operation in layers from 1 to $l-1$. The convolutional operation can be described as follows:

$$\tilde{h}_i^{(l)} = \rho\left(\sum_{j=1}^{n} A_{ij} W^{(l)} \tilde{h}_j^{(l-1)} + b^{(l)}\right) \tag{5}$$

where $W^{(l)}$ represents the weight matrix, $b^{(l)}$ denotes the bias vector and ρ is an activation function. $\tilde{h}^{(l-1)}$ and $\tilde{h}^{(l)}$ are the hidden states in the prior and current layers, respectively. Each node gathers and aggregates information from its neighbouring nodes during graph convolution.

4.3.2 Attention-guided layer

The incorporation of dependency information is more challenging than just imposing dependency edges. How to use relevant information while ignoring irrelevant information from the dependency trees remains a problem. Hard pruning methods (Zhang *et al.* 2018) are likely to prune one of the long sentences containing two verbs, causing information loss. Guo *et al.* (2019) proposed adopting multi-head self-attention mechanism (Vaswani *et al.* 2017) as a *soft pruning* strategy to extract relevant features from dependency graphs. Guo *et al.* (2019) introduced attention-guided GCN (called AGCN) to represent graphs as an alternative to previous GCNs (Kipf and Welling 2017). AGCN relies on a large, fully connected graph to reallocate the importance of each dependency relation rather than hard-pruning the graph into a smaller or simpler structure. The soft pruning strategy of distributing weight to each word can partly avoid this problem.

Generally, the AGCN layer models the dependency tree with the soft attention $\tilde{A}$, in which each cell weight ranges from 0 to 1. The shape of $\tilde{A}$ is the same as the original dependency adjacency matrix A for convenience. We compute attention weights in $\tilde{A}$ by using the multi-head attention (Vaswani *et al.* 2017). For the k-th head, $\tilde{A}^{(k)}$ is computed as follows:

$$\tilde{A}^{(k)} = softmax\left(\frac{QW_i^Q \times (KW_i^K)^T}{\sqrt{d}}\right) \tag{6}$$

where Q and K are the query and the key in multi-head attention, respectively, Q and K are both equal to the input representation $\tilde{H}$ (i.e., the output of the last module), d denotes the dimension of $\tilde{H}$, W_i^Q and W_i^K are both learnable parameters $\in \mathbb{R}^{d \times d}$, and $A^{(k)}$ is the k-th attention-guided adjacency matrix corresponding to the k-th head. Thus, we can replace the hard matrix A in the previous equation with the soft attention matrix $A^{(k)}$. The dependency relations, especially the indirect, multi-hop ones, are modelled by the multi-head mechanism.

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

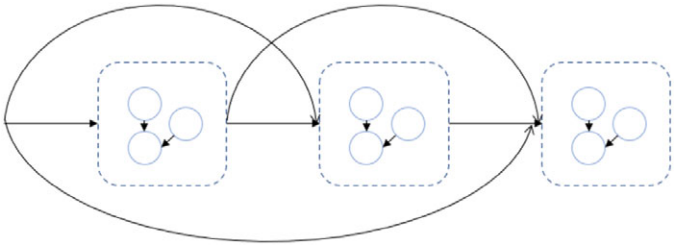

Figure 3. Densely connected structure. The sublayers are densely connected to fuse structures.

4.3.3 Densely connected structure

Previous work (Huang *et al.* 2017) has proven that dense connections across GCN layers helps to capture structural information. Deploying the densely connected structure forces the model to learn more non-local information and train a deeper GCN model. In our model, each densely connected structure has L sublayers placed in regular sequence. Each sublayer takes the outputs of all preceding sublayers as input, as shown in Figure 3. We define the output of the densely connected structure $g_j^{(l)}$ as follows:

$$g_j^{(l)} = \left[\tilde{h}_j^{(0)}, \tilde{h}_j^{(1)}, \ldots, \tilde{h}_j^{(l-1)}\right] \tag{7}$$

where x_j is the initial representation outputted by the alignment layer. $\tilde{h}_j^{(1)}, \ldots, \tilde{h}_j^{(l-1)}$ denote a concatenation of the representations produced by preceding sublayers. In addition, the dimension of representations in these sublayers shrinks to improve parameter efficiency, that is, $d_{hidden} = d/L$, where L is the number of sublayers, and d is the input dimension, with three sublayers and an input dimension of 768, $d_{hidden} = d/L = 256$. It outputs a fresh representation whose dimension is $768(256 \times 3)$ by concatenating all these sublayer outputs. Thus, the layer conserves considerable information at a low computational cost. This layer helps the weight gradually flow to the determining token. N densely connected layers are constructed to compute N adjacency matrices produced by attention-guided layers, where N denotes the numbers of the head. The GCN computation for each sublayer should be modified to adapt the multi-head attention as follows:

$$\tilde{h}_{k_i}^{(l)} = \rho\left(\sum_{j=1}^{n} \tilde{A}_{ij}^{(k)} W_k^{(l)} g_j^{(l)} + b_k^{(l)}\right) \tag{8}$$

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

4.3.4 Multi-layer combination

In this layer, we combine the representations outputted by N densely connected layers corresponding to N heads to generate the final latent representations:

$$\tilde{h}_{out} = W_{out}\tilde{h}_{in} + b_{out} \tag{9}$$

$$\tilde{h}_{in} = [\tilde{h}^{(1)}, \ldots, \tilde{h}^{(N)}] \tag{10}$$

where $\tilde{h}_{out} \in \mathbb{R}^d$ is the aggregated representation of N heads. W_{out} and b_{out} are the weights and biases learned during training.

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

4.4 Classifier

This layer maps the final hidden state sequence H to the class *metonymic* or *literal*. The representation H_i corresponds to the token t_i. Specifically, H_0 denotes '*[CLS]*' at the head of the subword sequence after tokenisation, which serves as the pooled embedding to represent the aggregate sequence.

Suppose that $\tilde{h}_{x'}, \ldots, \tilde{h}_{y'}$ are the word representations against the entity span E outputted by the syntactic integration layer. x' and y' represent the start and end index of the words in the entity span, respectively. We apply an operation of average pooling to obtain the final entity encoding:

$$H_e = MeanPooling(\tilde{h}_{x'}, \ldots, \tilde{h}_{y'}) \tag{11}$$

For classification, we concatenate H_0 and H_e consecutively, applying two fully connected layers with activation. Then, we apply a softmax layer to make the final prediction. The learning objective is to predict metonymic and literal classes for an entity within a given sentence:

$$H_{final} = \rho(W^*[\rho(W^{'} concat[H_0; H_e] + b^{'}]) + b^*]) \tag{12a}$$

$$\hat{\gamma} = \arg\max \frac{\exp(H_{final})}{\exp \sum_{0=r}^{|\Gamma|}(H_r)} \tag{12b}$$

where $\hat{\gamma}$ refers to a class type in the metonymy type set Γ. $W^{'} \in \mathbb{R}^{d \times 2d}$, $W^* \in \mathbb{R}^{r \times d}$, $|\Gamma|$ is the number of classification types, and d is the dimension of the hidden representation vector. While there are only two classes in this task, this approach can generalise to multiple classes.

5. Experiments

5.1 Datasets

We conducted our experiments mainly on three publicly available benchmarks: two small size location metonymy datasets, SemEval (Markert and Nissim 2007) and ReLocaR (Gritta *et al.* 2017), and a large size dataset, WiMCor (Mathews and Strube 2020). SemEval and ReLocaR are created to evaluate the capability of a classifier to distinguish literal (*geographical territories* and *political entities*), metonymic (*place-for-people*, *place-for-product*, *place-for-event*, *capital-for-government* or *place-for-organisation*) and mixed (metonymic and literal frames invoked simultaneously or are unable to distinguish) location mentions.

SemEval: The SemEval dataset[e] focuses on locations retrieved from the British National Corpus. The distribution of categories in the SemEval dataset is approximately 80% literal, 18% metonymic and 2% mixed to simulate the natural distribution of location metonymy. Therefore, a literal default tag already provides 80% precision. Although it contains finer-grained labels of metonymic patterns, such as *place-for-people*, *place-for-event* or *place-for-product*, we use only coarse-level labels of metonymy or literal in the experiment. Our experiment excluded the mixed class since it accounts for only 2% of the data. Finally, the dataset comprises training (910 samples) and testing (888 samples) partitions.

ReLocaR: The ReLocaR dataset[f] was collected using the sample data from Wikipedia's Random Article API. The data distribution of ReLocaR classes (literal, metonymic and mixed) is approximately 49%, 49% and 2%, respectively. We excluded mixed class instances. The processed dataset contains 1026 training and 982 testing instances and has a better label balance to eliminate the bias due to sub-sampling of the majority class to balance the classes.

[e] http://web.eecs.umich.edu/ mihalcea/affectivetext/#resources.
[f] https://github.com/milangritta/Minimalist-Location-Metonymy-Resolution/tree/master/data.

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

WIMCOR: The above datasets are limited in size. We also conduct our experiments on a large harvested corpus of location metonymy called WIMCOR.[g] WimCor is composed of a variety of location names, such as names of towns (e.g., '*Bath*'), cities (e.g., '*Freiburg*') and states (e.g., '*Texas*'). The average sentence length in WIMCOR is 80 tokens per sentence. While the samples in WIMCOR are annotated with coarse-grained, medium-grained and fine-grained labels, only the coarse labels (binary, i.e., metonymic or literal) are used in our experiments. The training set contains 92,563 literal instances and 31,037 metonymic instances.

5.2 Setup

5.2.1 Data pre-processing

This section introduces the way to obtain dependency relation matrices. We performed dependency parsing using the spaCy parser[h] and transformed all dependency trees (one parse tree per sentence) into symmetric adjacency matrices, ignoring the dependency directions and types for simplicity. In preliminary work, we conducted experiments using asymmetric matrices, but we did not observe any improvements.

For BERT variants, we followed Devlin *et al.* (2019) and used the tokenizer in BERT to segment words into word pieces as discussed in Section 4.1. We inserted the special *[ENT]* indicator before and after the entity spans as Wu and He (2019) did for E-BERT experiments. To adapt the sequence length distribution corresponding to each dataset, we set the max sequence length to 256 for SEMEVAL, and 128 for RELOCAR and WIMCOR.

5.2.2 Comparison setting

To evaluate our approach, we compared our model with different previous models: SVM, BiLSTM and PLMs, for example, BERT and ELMo. We took the current SOTA MR system BERT+MASK (Li *et al.* 2020) as the baseline, which did not include additional entity information. Following the best practices in Devlin *et al.* (2019) and Guo *et al.* (2019), we constructed the baseline and GCN models and set up the hyperparameters.

5.2.3 Training details

For all BERT-based models, we initialised the parameters of the BERT encoder using the pre-trained models released by Huggingface.[i] By launching two unsupervised tasks, namely, masked language model and the next sentence prediction (Devlin *et al.* 2019) on the large pre-training corpus, the sentence tokens are well represented. We empirically found that the large-cased-BERT model, which is case-sensitive and contains 24 transformer encoding blocks, each with 16 self-attention heads and 1024 hidden units, provides the best performance on the experimental datasets. The number of sublayers L_p for GCN block and L_n for ACGN block are 2 and 4, respectively.

For the SEMEVAL and RELOCAR datasets, we set the batch size to 8 and the number of training epochs to 20. For the WIMCOR dataset, we trained for 1 epoch and then reach convergence. We chose the number of heads for the multi-head attention N from the set $\{1, 2, 4, 6, 8\}$, and the initial learning rate for AdamW lr from the set $\{5 \times 10^{-6}, 1 \times 10^{-5}, 2 \times 10^{-5}\}$. The small learning rate yields more stable convergence and optimal results during model training but underfitting during training. A proper value for learning rate should be 1×10^{-5} or 2×10^{-5}. We chose these hyperparameters based on our experience obtained from extensive preliminary experiments, given the trade-off between time cost and performance depending on datasets. The combinations

[g] https://kevinalexmathews.github.io/software/.
[h] https://spacy.io/.
[i] https://github.com/huggingface/transformers.

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

of $(N = 8, lr = 1 \times 10^{-5})$, $(N = 4, lr = 2 \times 10^{-5})$ and $(N = 4, lr = 2 \times 10^{-5})$ provided the best results on SEMEVAL, RELOCAR and WIMCOR datasets, respectively. E-BERT+AGCN requires approximately 1.5 times the GPU memory compared with BERT when training the models on a Tesla V100-16GB GPU.

5.3 Results

5.3.1 Models

We compared our proposed method with different MR methods to evaluate it. The task of location MR is to detect the locations with literal reading only and ignore all other possible readings. Following Gritta, Pilehvar and Collier (2020), we classify the entity phrase as either literal or metonymic. The baseline models used in our experiments are listed below.

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

LSTM and BiLSTM: LSTM is one of the most powerful dynamic classifiers publicly known (Sundermeyer, Schlüter, and Ney 2012). Thanks to the featured memory function of remembering the last hidden states, it achieves decent results and is widely used on various NLP tasks (Gao *et al*. 2018; Si *et al*. 2019). Moreover, BiLSTM improves the token representation by being aware of the conditions from both directions (Hochreiter and Schmidhuber 1997), making ture contextual reasoning available. Additionally, two kinds of representations, GloVe (Pennington, Socher and Manning 2014) and ELMo, are tested separately to ensure model reliability.

Paragraph, Immediate and PreWin: Three models, Paragraph, Immediate and PreWin, are built upon BiLSTM models. They simultaneously encode tokens into word vectors and dependency relation labels into one-hot vectors (generally 5–10 tokens selected from the left and right of the entity work best). The three models differ in the manner of token picking. Immediate x chooses the x number of words to the immediate right and left of the entity as input to the model (Collobert *et al*. 2011; Mesnil *et al*. 2013; Mikolov *et al*. 2013; Baroni, Dinu, and Kruszewski 2014), for example, Immediate-5/10 takes the 5/10 words to the immediate right and left of the entity as input to a model. The Paragraph model extends the Immediate model that takes more words (50 words) from each entity's side as the input. PreWin selects the words near the local predicate to eliminate long-distance noise in the input.

PreWin (BERT) is the reimplementation of the PreWin system with BERT embeddings as the input. Instead of deploying BERT as a classifier, we replace the original GloVe embeddings with BERT embeddings used in the PreWin model and initialise word embeddings using BERT embeddings. Word embeddings are combined by summing subword embeddings to generate GloVe-like word embeddings.

BERT, +AUG, +MASK: Three BERT-based MR models are described in Li *et al*. (2020). The vanilla BERT model (Devlin *et al*. 2019) can be directly used to detect metonymies by performing sentential classification. BERT encodes the input tokens into distributed vector representations after fine-tuning over datasets. BERT+AUG is fine-tuned with data augmentation (Li *et al*. 2020). This method generates new samples by randomly substituting the target entity nominal with one from all the extracted target words. BERT+MASK fine-tunes the BERT model with target word masking that replaces the input target word with the single token [*ENT*] during training and evaluation.

E-BERT (sent) and **E-BERT (sent+ent)**: **E**ntity-aware **BERT**, namely, E-BERT, enriches the semantic representations by incorporating the entity information. The input to the E-BERT (sent) model is slightly different from the original dataset, where we inserted *[ENT]* markers before and

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

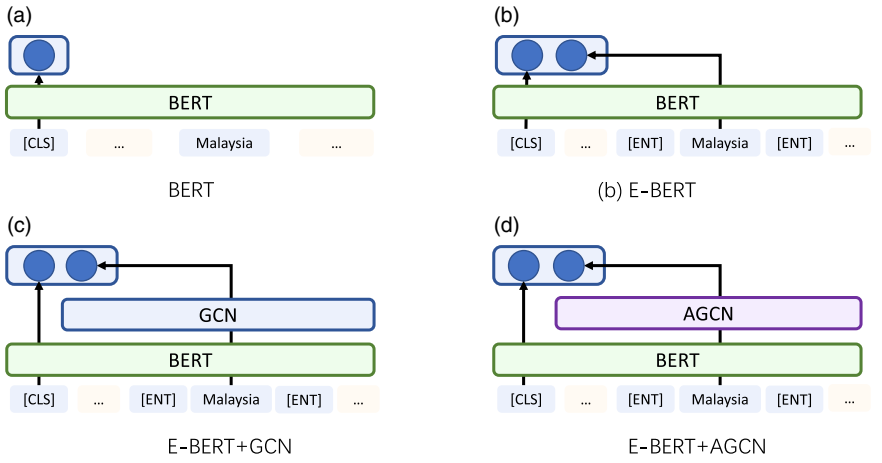

Figure 4. Variants of the architecture for extracting entity and sentence representations from the deep Transformers network. (a) A model with the standard input and with sentence output at the position of [*CLS*]; (b) a model using the entity-marked input and with the sentence (i.e., [*CLS*]) and the entity outputs; (c) and (d) two models with the entity-marked input and with the sentence (i.e., [*CLS*]) and the entity outputs using GCN and AGCN, respectively.

after the entity spans, making BERT aware of the entity position. The E-BERT (sent) model represents the sentence using the encoding at the [*CLS*] position. The E-BERT (sent+ent) model shares the same network structure as the R-BERT model (Wu and He 2019) for RE, but it depends on a sole entity. Concretely, this variation concatenates the target entity's sentential encoding and corresponding encoding.

E-BERT+GCN: This model applies a hard pruning strategy using GCN computation to integrate syntactic information into BERT representations. The input sentences are inserted with the *[ENT]* label before and after the metonymic and literal entity span.

E-BERT+AGCN: We build the fully attentive system E-BERT+AGCN based on E-BERT+GCN. The attention-guided layer in E-BERT+AGCN employs a soft attention mechanism to assign proper weights to all dependencies. Figure 4 illustrates all BERT variants used in this paper, including BERT, E-BERT, E-BERT+GCN and E-BERT+AGCN.

5.3.2 Overall evaluation

We compared the averaged F1 and accuracy scores by running each model 10 times (see Table 2). In the accuracy comparison, the performance of the feature-based model SVM+Wiki is still superior to most of the recent DNN models. LSTMs yielded better results due to operations, such as structure variation modelling, word representation improvement and feature integration. Of note, NER and part-of-speech (POS) features have less of an effect on BiLSTM (ELMo). The semantics provided by POS/NER features might be redundant for ELMo representations. PreWin surpasses the baseline LSTM (GloVe) by a large margin on both RELOCAR and SEMEVAL datasets. The result indicates the significance of syntactic information. In addition, the process of choosing and retaining related tokens is also a major contributor to PreWin, resulting in at least 1.2 and 2.2 points higher than Paragraph and Immediate on SEMEVAL and RELOCAR, respectively.

The results of E-BERT+GCN and E-BERT+AGCN show that our model is able to leverage syntactic information that is useful for MR and demonstrates its advantages over previous works with SOTA results. Specifically, E-BERT+AGCN considerably outperforms the previous model based on heavy feature engineering, that is, SVM+Wiki. Our model also surpassed previous DNN models, including LSTM, BiLSTM and PreWin, even when enriched with POS and

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

Table 1. Statistics of the datasets used in this paper. The table describes the number of identical entities and the number of overlapping entities in the training and test sets. The table includes sentence length, entity position and the number of verbs per sentence

		ReLocaR	SemEval	WiMCor
Train	# of sent	1026	910	123,600
	sent_len	22.0±9.7	30.1±17.7	84.5±55.0
	ent_pos	9.9±7.5	15.4±13.5	41.2±39.1
	distinct ent.	336	183	1010
	# of verbs	2.0±1.4	2.9±2.2	8.0±6.6
Test	# of sent.	982	888	41,200
	length	23.4±9.9	29.7±16.9	84.8±55.8
	ent_pos	11.8±9.2	14.7±12.1	41.1±39.0
	distinct	346	174	1010
	# of verbs	2.1±1.5	2.9±2.1	8.0±6.8
Overlapped		140	114	1010

NER features. Furthermore, we compared E-BERT+AGCN with two baseline models: E-BERT (the entity-aware BERT model without syntactic integration) and E-BERT+GCN (imposing hard syntactic constraints with GCN).

Moreover, the experiment on E-BERT+GCN shows an accuracy increase that is 0.3% and 0.2% higher than E-BERT (sent+ent) on the SemEval and ReLocaR datasets, respectively. GCN improves performance by catching useful information from syntax. The hard pruning behaviour of Immediate 5 exerted on E-BERT+GCN has little effect, which shows that pruning graphs crudely may be counterproductive. E-BERT+AGCN obtains improvements of 0.7% and 0.2% on the SemEval and ReLocaR datasets, respectively, compared with E-BERT+GCN. Therefore, introducing a multi-head attention mechanism that assists GCNs in information aggregation seems successful. The standard deviation of E-BERT+AGCN is also lower than E-BERT+GCN, indicating a more robust model performance. Our approach effectively incorporates soft-dependency constraints into MR models by pruning irrelevant information and emphasising dominant relations concerning indicators.

We also report F1 scores for literal class and metonymic class separately. ReLocaR is a class-balanced dataset with literal and metonymic independently accounting for 50% of all examples in the training dataset. The F1 score of ReLocaR is relatively higher than that of the SemEval dataset due to the shorter sentence length. In the ReLocaR rows, the F1 of both classes indicates a slight upgrade compared with baseline E-BERT and E-BERT+GCN, since the standard deviations are relatively higher. Conversely, SemEval serves as a benchmark with literal and metonymic accounting for 80% and 20%. The imbalance causes a lack of metonymic evidence, making the model learning process insufficient. As reflected in Table 2, earlier models, such as LSTM, have an inferior F1 performance on the metonymic class compared with the literal class. The considerable performance gap of 3.4% and 4.0% in F1-M between BERT and E-BERT+AGCN shows that E-BERT+AGCN is more powerful in capturing syntactic clues to solve the sample limitation. To summarise, E-BERT+AGCN achieves the highest F1 scores for both SemEval and ReLocaR and is able to adapt to various class distributions in the dataset.

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

Table 2. The overall F1 and accuracy scores on the SemEval and ReLocaR datasets. 'L' and 'M' denote literal and metonymic classes. +NER+POS means integrating both NER and POS features with the baseline model. In general, E-BERT+AGCN obtains the best results. The boldface denotes the best results and '↑' means statistically significant improvement over the baseline (BERT+MASK, Li *et al*. 2020) with p-value ≤ 0.05. † and ‡ are the results reported in the previous works of Gritta *et al*. (2017) and Li *et al*. (2020), respectively. Since the datasets are slightly different, we re-implement systems of Li *et al*. (2020) and report the new results labelled by ∗

MODEL	SemEval			ReLocaR		
	F1-L	F1-M	Acc. (std.)	F1-L	F1-M	Acc. (std.)
SVM+Wiki	91.6	59.1	86.2 (N/A)	–	–	–
LSTM (GloVe)	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)	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)
BiLSTM (ELMo)	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)
Paragraph†	–	–	81.3 (0.88)	–	–	80.0 (2.25)
Immediate-5†	–	–	81.3 (1.11)	–	–	81.4 (1.34)
Immediate-10†	–	–	81.9 (0.89)	–	–	81.3 (1.44)
PreWin (GloVe)†	90.6	57.3	83.1 (0.64)	84.4	84.8	83.6 (0.71)
PreWin (BERT)‡	–	–	87.1 (0.54)	–	–	92.2 (0.48)
BERT*	91.6	59.7	86.2 (0.32)	91.8	91.8	91.8 (0.81)
+AUG*	91.9	56.9	86.4 (0.55)	91.4	91.4	91.4 (0.08)
+MASK* (***SOTA***)	93.0	63.3	88.2 (0.61)	95.3	95.4	95.3 (0.41)
E-BERT (sent)	93.5	60.0	87.6 (0.55)	94.0	94.0	94.0 (0.58)
E-BERT (sent+ent)	93.2	66.0	88.8 (0.63)	95.2	95.3	95.3 (0.44)
+GCN	93.5↑	67.5↑	89.1 (0.60)↑	95.5	95.5	95.5 (0.46)
+GCN (Immediate-5)	93.6↑	65.7↑	89.0 (0.50)↑	95.3	95.4	95.4 (0.44)
+AGCN	**94.0**↑	**68.3**↑	**89.6 (0.85)**↑	**95.7**↑	**95.8**↑	**95.8 (0.34)** ↑

In addition, to verify the effectiveness of our model on a larger dataset, we launch the experiment using the WiMCor dataset. Table 3 also gives the results on the WiMCor dataset. Though the increase is not substantial in terms of accuracy or F1 scores, our model leads to a 0.2 percentage point improvement compared to E-BERT, given the fact that the WiMCor testing set contains 41,200 instances.

5.3.3 Cross-domain evaluation

Since the metonymy datasets were created using different annotation guidelines, it is informative to study the generalisation ability of the developed models. Thus, we launched across-domain experiments to evaluate the model's performance under cross-domain configurations as follows:

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

Table 3. Results of the WIMCOR dataset

	MODEL	Acc	Precision	Recall	F1-L	F1-M
WIMCOR→WIMCOR	E-BERT	96.8	95.1	92.5	97.8	93.8
	E-BERT+GCN	96.9	95.2	**92.6**	97.9	93.9
	E-BERT+AGCN	**97.0**	**96.4**	91.7	**98.0**	**94.0**

Table 4. Cross-domain accuracy, precision, recall and F1 scores. The best results are indicated in boldface

	Models	Acc	Precision	Recall	F1-L	F1-M
WIMCOR→RELOCAR	E-BERT	59.1	80.1	25.2	69.3	38.3
	E-BERT+GCN	58.8	81.8	23.6	69.4	36.6
	E-BERT+AGCN	**65.8**	**83.9**	**39.9**	**72.7**	**54.0**
WIMCOR→SEMEVAL	E-BERT	80.2	42.4	15.0	88.6	22.1
	E-BERT+GCN	80.0	35.1	7.8	88.7	12.7
	E-BERT+AGCN	**81.6**	**52.7**	**23.4**	**89.4**	**32.4**
SEMEVAL→RELOCAR	E-BERT	66.4	**99.4**	33.7	74.6	50.3
	E-BERT+GCN	**70.9**	96.1	**44.4**	**76.9**	**60.6**
	E-BERT+AGCN	70.1	99.0	41.1	76.7	58.1
RELOCAR→SEMEVAL	E-BERT	72.5	40.1	**93.4**	80.0	56.1
	E-BERT+GCN	**74.1**	**41.5**	91.6	**81.5**	**57.1**
	E-BERT+AGCN	69.8	37.7	92.2	77.7	53.7

WIMCOR→RELOCAR, WIMCOR→SEMEVAL, SEMEVAL→RELOCAR, RELOCAR→SEMEVAL. We trained models on one dataset for all configurations and tested them on another dataset and then compared the three models, E-BERT, E-BERT+GCN and E-BERT+AGCN. As shown in Table 4, the result on the WIMCOR dataset indicates the robustness of E-BERT+AGCN on a large benchmark. Although incorporating hard syntactic constraints improves the MR results slightly, the soft constraint is more efficient than the hard in the experiments in terms of accuracy.

5.3.4 Sentence length

We further compared the accuracy of E-BERT+AGCN and E-BERT with different sentence lengths, as shown in Figure 5. The experiment is conducted on both SEMEVAL and the RELOCAR datasets. Note that the average sentence length of RELOCAR is shorter than that of the SEMEVAL.

To highlight the issues, we primarily discuss the SEMEVAL dataset. Long sentences are likely to affect the classification accuracy and cause poor performance for two reasons: 1. contextual meanings for long sentences are more difficult to capture and represent; 2. the position of key tokens, like the predicate, can be far from the entity in a sentence, therefore, difficult to determine. Thus, it is challenging for sequence-based models to retain adequate performance when tested with long sequences. BERT fails to utilise structural information such as dependency trees that have been proven to benefit NLP tasks. Previous studies (Tang *et al.* 2018) have shown that BERT lacks model interpretability for non-local syntactic relations, for example, long-distance

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

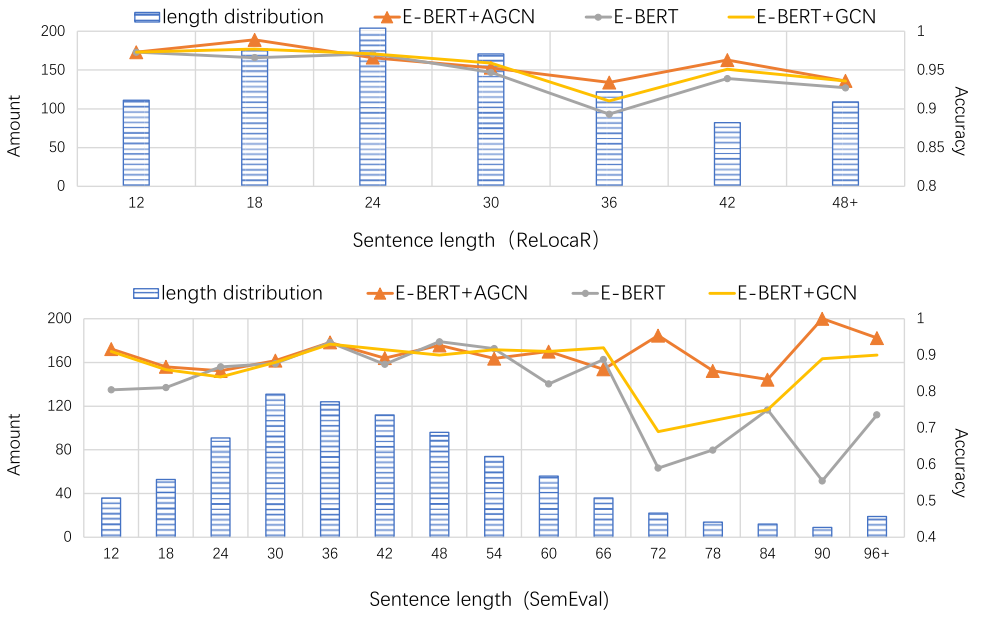

Figure 5. Comparison on the ReLocaR and the SemEval datasets w.r.t. different sentence lengths. E-BERT+AGCN significantly outperforms E-BERT and E-BERT+GCN on the SemEval dataset when sentence length is longer than 70.

syntactic relations. The accuracy drops as in Figure 5 for BERT when the sentence length grows. In this case, a dependency-based model is more suitable for handling long-distance relations while reducing computational complexity. E-BERT+AGCN alleviated such performance degradation and outperformed the two baselines in all buckets, and the improvement becomes more significant when the sentence length increases (≥ 30 on ReLocaR and ≥ 66 on SemEval). The results in Figure 5 confirm that E-BERT+AGCN produces better entity and contextual representations for MR, especially for longer sentences.

5.3.5 Case study

This section describes is a case study using a motivating example correctly classified by E-BERT+AGCN but misclassified by E-BERT to show the effectiveness of our model. Given the sentence, '*He later went to manage Malaysia for one year*', a native speaker can easily identify '*Malaysia*' as a metonymic term by linking '*Malaysia*' to the concept of '*the national football team of Malaysia*'. In the above sentence, since the verb phrase, '*went to*' is a strong indicator, E-BERT is prone to overlook the second predicate '*manage*'. As a result, E-BERT would falsely recognises '*Malaysia*' as a literal territory (due to customary usage of '*went to somewhere*'). We can explain how the problem mentioned above is resolved in E-BERT+AGCN by visualising the attention weights of the model. We first compare the attention matrix of the transformer encoder blocks in E-BERT to check the contribution of syntactic integration to the whole model. Figure 6(a) shows that the weights of tokens in BERT are decentralised. Intuitively, E-BERT provides insufficient semantic knowledge to MR, resulting in the lose of useful information as to which target words should be considered. In Figure 6(b), the attention in E-BERT+AGCN concentrates on the predicate '*manage*' and the entity '*Malaysia*' rather than on '*went to*' and other tokens. With the integration of syntactic components, the dependency information assisted the model in being aware of the sentence structure. As a result, our model selects relevant tokens and discards the irrelevant and misleading tokens.

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

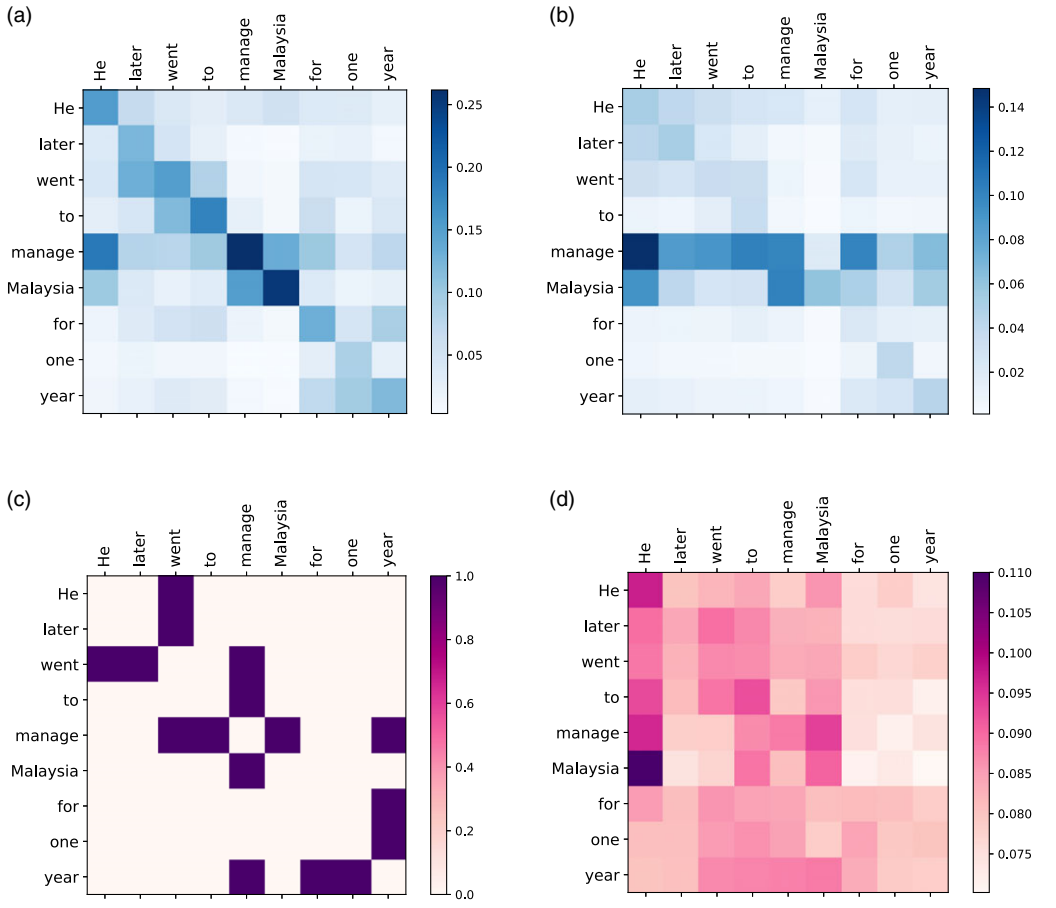

Figure 6. Visualisation of attention matrices (better viewed in colour). (a) averaged attention weights in the E-BERT encoder (E-BERT), (b) averaged attention weights in the E-BERT encoder (E-BERT+AGCN); (c) attention weights in the non-attention modules (E-BERT+GCN and E-BERT+AGCN); (d) averaged attention weights in the attention modules (E-BERT+AGCN). (a) with (b) show the effect of incorporating syntactic constraints on BERT. (c) and (d) illustrate soft attention weights compared to hard ones. (a)∼(d) illustrate that incorporating syntactic information forces the model to focus on the neighbours of the target word in the dependency graph, compared to Figure 1.

Furthermore, the sentence in the example can be divided into the main clause '*He later went to manage Malaysia*' and the prepositional phrase '*for one year*'. The main clause contains the predicate and the entity that dominates the MR inference. However, conventional methods consider the modified relation between '*one*' and '*year*' as well as other irrelevant connections to have the same weight. This process introduces massive noise in feature extraction.

As shown in Figure 6(b), the prepositional phrase '*for one year*' is irrelevant to the MR task. Despite the existence of dependency relations for the prepositional phrase, the weights of those relations are relatively lower compared with the main clause, which includes the verb and its dependent words. After launching the multi-head attention mechanism, the model is free from fixed pruning rules and flexibly learns the connections among tokens.

The syntactic component (GCN Block) first selects relevant syntactic features efficiently given the hard dependency adjacency matrix (see Figure 6(c)). Then, the attention-guided layer learns the soft attention matrix. To demonstrate the superiority of soft dependency relations, we use Figure 6(d) to visualise the attention weights of the attention-guided layer. Unlike the attention in the BERT encoding layer, the attention-guided layer's attention matrix reflects more information

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

Table 5. Samples for error analysis. Bold denotes the target entities for MR. 'Label' refers to the class label of the target entity, followed by the correctness of the predictions of E-BERT and E-BERT+AGCN models

#	Sentence	Label	E-BERT	E-BERT+AGCN
*S*1	Her personal bests in the event are 1.92 metres outdoors (**Marseille** 2015) and 1.93 metres indoors (Budapest 2015)	MET	×	×
*S*2	Engaged in very long range strategic bombing missions to enemy military, industrial and transportation, were Italy, France, Germany, Austria, Hungary, **Romania**, and Yugoslavia	MET	×	✓
*S*3	This led to an open uprising by **Schleswig-Holstein**'s large German majority in support of independence from Denmark and of close association with the German Confederation	LIT	×	×
*S*4	The LP had advance orders of a half million and sold another half million by September 1965, making it the second album to sell a million copies in the United Kingdom, after the soundtrack to the 1958 film **South Pacific**	MET	×	✓
*S*5	After spending three years in London on board the prison hulk Newgate, Hutchinson was transported to Australia on the **Hillsborough**, sometimes referred to as the 'Fever Ship' since some ninety-five of the three hundred convicts aboard died from typhoid fever brought aboard from the prison hulks	MET	×	×

about dependency relationships. GCN is prone to trust information for all its one-hop neighbours in dependency graphs while overlooking other neighbours. In contrast, AGCN uses multi-head attention to attend to different representation subspaces to reduce the information loss jointly.

6. Error analysis

In most cases shown in Table 5, E-BERT+AGCN makes correct predictions. However, typical issues caused by various reasons remain unsolved. We will discuss three types of such unsolved errors here.

6.1 Error propagation

E-BERT+AGCN predicts the entity type given dependency relations and contextualised embeddings. Concretely, *S*1 shows an example that presents isolation of the term '*Marseille 2015*' in the dependency parse tree. The subtree of '*Marseille 2015*' and the remaining parts are split by parentheses. In an extreme case, E-BERT+AGCN fails to recognise '*Marseille*' as '*a sport event in Marseille*'. We found that the connection between '*beats*' and '*Marseille*' is missing in the parse tree.

6.2 Multiple predicates

The predicate plays a crucial role in understanding a sentence (Shibata, Kawahara, and Kurohashi 2016). In MR tasks, if an entity exerts an action on others, it is probably a metonym. *S*2 is a typical example of long distance between the predicate and the entity. In this situation, BERT fails to classify and cannot catch non-local features due to long distances. Meanwhile, E-BERT+AGCN correctly predicts metonymy by relying on the related entity and the predicate, '*engaged*'. In other words, E-BERT+AGCN can mine strong connections in sentences with a relatively straightforward and smart effort. The observation proves again that the syntactic component is efficient in searching keywords.

In more complex cases, our method might fail to detect metonymy. For example, in *S*3, conventional models might easily find the predicate '*uprising*'. The event participant is '*Schleswig-Holstein's large German majority*' rather than '*Schleswig-Holstein*'. E-BERT+AGCN could not

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

trace the predicate and made an incorrect prediction even though it was aware of syntactic structural knowledge.

6.3 Knowledge deficiency

Many metonymies are proper nouns that refer to existing well-known works or events. Previous models struggle with the limitations of lacking real-world and common-sense knowledge to access the referents, which results in poor interpretability. In contrast, in sentence *S*4, '*South Pacific*' refers to a film in 1958. E-BERT fails to recognise such a metonym, while E-BERT+AGCN successfully extends the implication of '*South Pacific*' to the film called *South Pacific* due to the dependency between '*film*' and '*South Pacific*'. *S*5 is one of the failed cases. In sentence *S*5, E-BERT+AGCN fails to detect metonymy, notwithstanding the explanation of '*Hillsborough*', referring to '*Fever Ship*', which has been mentioned in the discourse. In fact, '*Hillsborough*' is a ship name involved in unconventional metonymy. As discussed in Section 2, identifying such a logical metonymy is difficult since the interpretation requires additional, inferred information.

7. Conclusion and future work

This paper shows the success of a neural architecture deploying a dependency-guided network to capture non-local clues for MR. This approach incorporates hard and soft dependency constraints into MR, enabling context- and syntax-aware representations. Experimental results and analyses on MR benchmark datasets showed that our proposed architecture surpasses previous approaches. Our work also demonstrates the importance of syntax for NLP applications.

There are several potential directions for future research. For example, further introducing dependency types (Tian *et al*. 2021) into the GCN variations and using external knowledge bases (Mihaylov and Frank 2018; Lin *et al*. 2019a; Yang *et al*. 2019) to mine latent relations appear to be of interest. To make full use of the prior knowledge for MR, we also plan to replace BERT with knowledge-enhanced PLMs (Zhang *et al*. 2019).

Acknowledgement. We thank the anonymous reviewers for their valuable comments. This work was supported by Shanghai Science and Technology Young Talents Sailing Program 21YF1413900, Fundamental Research Funds for the Central Universities (43800-20101-222340) and in part by the National Natural Science Foundation of China under grants 91746203, 61991410 and the National Key R&D Program of China under grant 2018AAA0102804.

References

Bahdanau D., **Cho K. and Bengio Y.** (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Baroni M., **Dinu G. and Kruszewski G.** (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 238–247.

Brun C., **Ehrmann M. and Jacquet G.** (2007). XRCE-M: A hybrid system for named entity metonymy resolution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Association for Computational Linguistics, pp. 488–491.

Chan Y.S. and Roth D. (2011). Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA. Association for Computational Linguistics, pp. 551–560.

Collobert R., **Weston J.**, **Bottou L.**, **Karlen M.**, **Kavukcuoglu K. and Kuksa P.P.** (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research* **12**, 2493–2537.

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

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

Farkas R., **Simon E.**, **Szarvas G. and Varga D.** (2007). Gyder: Maxent metonymy resolution. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 161–164.

Fass D. (1988). Metonymy and metaphor: What's the difference? In *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*. International Committee on Computational Linguistics, pp. 177–181.

Fu T.-J., **Li P.-H. and Ma W.-Y.** (2019). GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 1409–1418.

Fundel K., **Küffner R. and Zimmer R.** (2007). RelEx—relation extraction using dependency parse trees. *Bioinformatics* **23**(3), 365–371.

Gao G., **Choi E.**, **Choi Y. and Zettlemoyer L.** (2018). Neural metaphor detection in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 607–613.

Glavaš G. and Vulić I. (2021). Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online. Association for Computational Linguistics, pp. 3090–3104.

Gritta M., **Pilehvar M.T. and Collier N.** (2020). A pragmatic guide to geoparsing evaluation toponyms, named entity recognition and pragmatics. *Lang Resources & Evaluation* **54**, 683–712.

Gritta M., **Pilehvar M.T.**, **Limsopatham N. and Collier N.** (2017). Vancouver welcomes you! minimalist location metonymy resolution. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, vol. 1. Association for Computational Linguistics, pp. 1248–1259.

Guo Z., **Zhang Y. and Lu W.** (2019). Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 241–251.

Hobbs J.R. and Martin P. 1987. Local pragmatics. Technical report. SRI International Menlo Park CA Artificial Intelligence Center, pp. 520–523.

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

Huang G., **Liu Z.**, **Van Der Maaten L. and Weinberger K.Q.** (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708.

Janda L.A. (2011). Metonymy in word-formation. *Cognitive Linguistics* **22**(2), 359–392.

Jawahar G., **Sagot B. and Seddah D.** (2019). What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 3651–3657.

Joshi M. and Penstein-Rosé C. (2009). Generalizing dependency features for opinion mining. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. Association for Computational Linguistics, pp. 313–316.

Kambhatla N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, vol. 22. Association for Computational Linguistics.

Kamei S.-i. and Wakao T. (1992). Metonymy: Reassessment, survey of acceptability, and its treatment in a machine translation system. In *Proceedings of the 30th Annual Meeting on Association for Computational Linguistics (ACL'92)*. Association for Computational Linguistics, pp. 309–311.

Kipf T.N. and Welling M. (2017). Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*.

Kvecses Z. and Radden G. (1998). Metonymy: Developing a cognitive linguistic view. *Cognitive Linguistics* **9**(1), 37–78.

Lakoff G. (1987). Image metaphors. *Metaphor and Symbol* **2**(3), 219–222.

Lakoff G. (1991). Metaphor and war: The metaphor system used to justify war in the gulf. *Peace Research*, **23**(2/3), 25–32.

Lakoff G. (1993). The Contemporary Theory of Metaphor. In Ortony A (ed), *Metaphor and Thought*. Cambridge, UK: Cambridge University Press, pp. 202–251.

Lakoff G. and Johnson M. (1980). *Conceptual Metaphor in Everyday Language*, vol. **77**. JSTOR, pp. 453–486.

Li D., **Wei F.**, **Tan C.**, **Tang D. and Ke X.** (2014). Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 49–54.

Li H., **Vasardani M.**, **Tomko M. and Baldwin T.** (2020). Target word masking for location metonymy resolution. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, pp. 3696–3707.

Lin B.Y., **Chen X.**, **Chen J. and Ren X.** (2019a). KagNet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, pp. 2822–2832.

Lin C., **Miller T.**, **Dligach D.**, **Bethard S. and Savova G.** (2019b). 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*, pp. 65–71.

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

Liu Y., Wei F., Li S., Ji H., Zhou M. and Wang H. (2015). A dependency-based neural network for relation classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, pp. 285–290.

Luong T., Pham H. and Manning C.D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal. Association for Computational Linguistics, pp. 1412–1421.

Markert K. and Nissim M. (2002). Metonymy resolution as a classification task. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*. Association for Computational Linguistics, pp. 204–213.

Markert K. and Nissim M. (2007). Semeval-2007 task 08: Metonymy resolution at semeval-2007. In *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, pp. 36–41.

Markert K. and Nissim M. (2009). Data and models for metonymy resolution. *Lang Resources & Evaluation* **43**, 123–138.

Mathews K.A. and Strube M. (2020). A large harvested corpus of location metonymy. In *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association, pp. 5678–5687.

Mesnil G., He X., Deng L. and Bengio Y. (2013). Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25–29, 2013*. International Speech Communication Association, pp. 3771–3775.

Mihaylov T. and Frank A. (2018). Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 821–832.

Mikolov T., Sutskever I., Chen K., Corrado G.S. and Dean J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pp. 3111–3119.

Miwa M. and Bansal M. (2016). End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 1105–1116.

Monteiro B.R., Davis C.A. and Fonseca F. (2016). A survey on the geographic scope of textual documents. *Computers & Geosciences* **96**, 23–34.

Nastase V., Judea A., Markert K. and Strube M. (2012). Local and global context for supervised and unsupervised metonymy resolution. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 183–193.

Nastase V. and Strube M. (2009). Combining collocations, lexical and encyclopedic knowledge for metonymy resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 910–918.

Nastase V. and Strube M. (2013). Transforming wikipedia into a large scale multilingual concept network. *Artificial Intelligence* **194**, 62–85.

Nissim M. and Markert K. (2003). Syntactic features and word similarity for supervised metonymy resolution. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 56–63.

Peng N., Poon H., Quirk C., Toutanova K. and Yih W.-t. (2017). Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics* **5**, 101–115.

Pennington J., Socher R. and Manning C.D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1532–1543.

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

Peters M.E., Ammar W., Bhagavatula C. and Power R. (2017). Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017 Volume 1: Long Papers*. Association for Computational Linguistics, pp. 1756–1765.

Pinango M.M., Zhang M., Foster-Hanson E., Negishi M., Lacadie C. and Constable R.T. (2017). Metonymy as referential dependency: Psycholinguistic and neurolinguistic arguments for a unified linguistic treatment. *Cognitive Science* **41**(2SUPPL.S2), 351–378.

Pustejovsky J. (1991). The generative lexicon. *Computational Linguistics* **17**(4), 409–441.

Qu C., Yang L., Qiu M., Croft W.B., Zhang Y. and Iyyer, M. (2019). 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*, pp. 1133–1136.

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

Rajpurkar P., Zhang J., Lopyrev K. and Liang P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 2383–2392.

Shibata T., Kawahara D. and Kurohashi S. (2016). Neural network-based model for Japanese predicate argument structure analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 1235–1244.

Si C., Chen W., Wang W., Wang L. and Tan T. (2019). An attention enhanced graph convolutional LSTM network for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1227–1236.

Socher R., Perelygin A., Wu J., Chuang J., Manning C.D., Ng A.Y. and Potts C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1631–1642.

Sun C., Huang L. and Qiu X. (2019). Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. 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)*. Association for Computational Linguistics, pp. 380–385.

Sundermeyer M., Schlüter R. and Ney H. (2012). LSTM neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*. International Speech Communication Association, pp. 194–198.

Tang G., Muller M., Gonzales A.R. and Sennrich R. (2018). Why self-attention? a targeted evaluation of neural machine translation architectures. In *Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 4263–4272.

Tian Y., Chen G., Song Y. and Wan X. (2021). 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)*, Online. Association for Computational Linguistics, pp. 4458–4471.

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł. and Polosukhin I. (2017). Attention is all you need. *In Advances in Neural Information Processing Systems*, pp. 5998–6008.

Wu S. and He Y. (2019). Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2361–2364.

Xu Y., Mou L., Li G., Chen Y., Peng H. and Jin Z. (2015). 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*. Association for Computational Linguistics, pp. 1785–1794.

Yang A., Wang Q., Liu J., Liu K., Lyu Y., Wu H., She Q. and Li S. (2019). Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 2346–2357.

Zarcone A., Utt J. and Padó S. (2012). Modeling covert event retrieval in logical metonymy: Probabilistic and distributional accounts. In *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2012)*, Montréal, Canada. Association for Computational Linguistics, pp. 70–79.

Zhang M., Zhang J., and Su J. (2006). Exploring syntactic features for relation extraction using a convolution tree kernel. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. Association for Computational Linguistics, pp. 288–295.

Zhang Y., Qi P. and Manning C.D. (2018). Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 2205–2215.

Zhang Z., Han X., Liu Z., Jiang X., Sun M. and Liu Q. (2019). ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 1441–1451.

A. Comparison of data size

Figure A.1. shows the performance of E-BERT, E-BERT+GCN and E-BERT+AGCN against different settings of data size. Given the SemEval dataset has fewer metonymic instances, we conduct the experiment on ReLocaR only. Using only 20% of training data, two models achieve desirable F1 scores near 90. The result demonstrates the robustness of E-BERT+GCN and E-BERT+AGCN models. When comparing them under the same data size setting, E-BERT+AGCN

https://doi.org/10.1017/S135132492200033X Published online by Cambridge University Press

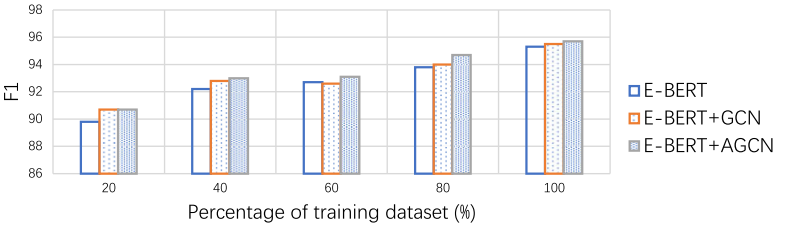

Figure A.1. Comparison of F1 scores w.r.t training data size on the RELOCAR dataset. We train the model with different percentages {20, 40, 60, 80, 100} of the training dataset.

substantially outperforms E-BERT, and the performance gap between E-BERT+AGCN and E-BERT is always larger than 0.4%. The observation suggests that the E-BERT+AGCN model has better generalisation than E-BERT, especially for small datasets.

Cite this article: Wang H, Du S, Zheng X and Meng L (2023). An empirical study of incorporating syntactic constraints into BERT-based location metonymy resolution. *Natural Language Engineering* **29**, 669–692. https://doi.org/10.1017/S135132492200033X