Commonsense Knowledge Enhanced Sentiment Dependency Graph for Sarcasm Detection

Zhe Yu[1], **Di Jin**[1,2], **Xiaobao Wang**[2*], **Yawen Li**[3], **Longbiao Wang**[2,4] and **Jianwu Dang**[5,2]
[1]School of New Media and Communication, Tianjin University, Tianjin, China
[2]Tianjin Key Laboratory of Cognitive Computing and Application,
College of Intelligence and Computing, Tianjin University, Tianjin, China
[3]School of Economics and Management,
Beijing University of Posts and Telecommunications, Beijing, China
[4]Huiyan Technology (Tianjin) Co., Ltd, Tianjin, China
[5]Peng Cheng Laboratory, Shenzhen, China
{yu_zhe, jindi, wangxiaobao}@tju.edu.cn, warmly0716@bupt.edu.cn,
longbiao_wang@tju.edu.cn, jdang@jaist.ac.jp

Abstract

Sarcasm is widely utilized on social media platforms such as Twitter and Reddit. Sarcasm detection is required for analyzing people's true feelings since sarcasm is commonly used to portray a reversed emotion opposing the literal meaning. The syntactic structure is the key to make better use of commonsense when detecting sarcasm. However, it is extremely challenging to effectively and explicitly explore the information implied in syntactic structure and commonsense simultaneously. In this paper, we apply the pre-trained COMET model to generate relevant commonsense knowledge, and explore a novel scenario of constructing a commonsense-augmented sentiment graph and a commonsense-replaced dependency graph for each text. Based on this, a Commonsense Sentiment Dependency Graph Convolutional Network (CS-DGCN) framework is proposed to explicitly depict the role of external commonsense and inconsistent expressions over the context for sarcasm detection by interactively modeling the sentiment and dependency information. Experimental results on several benchmark datasets reveal that our proposed method beats the state-of-the-art methods in sarcasm detection, and has a stronger interpretability.

1 Introduction

Sarcasm, in which the intention behind its literal meaning is typically reversed, is a demanding and practical Natural Language Processing (NLP) problem. As illustrated in the first given example of Figure 1, the decisive sentiment words "love" and "ignore" lead to a contradiction expression. That is, there are some incongruity expressions in sarcastic contexts. Due to the prevalence of sarcasm in tweets, product reviews, and online debate forums, sarcasm detection is vital to a variety of applications, including genuine sentiment classification [Wang *et al.*, 2020; Liu and others, 2010; Jiang *et al.*, 2020] and opinion mining [Miao *et al.*, 2020; Bakshi *et al.*, 2016].

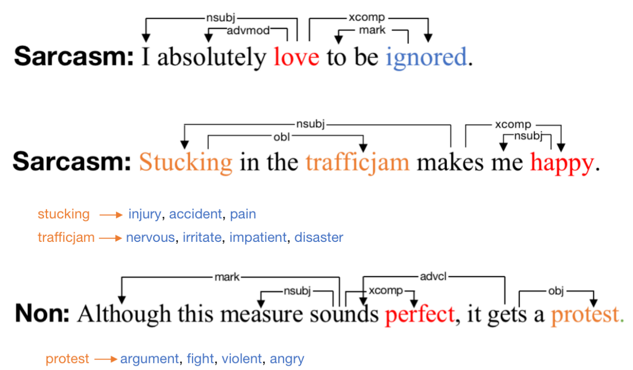

Figure 1: Examples of sarcasm and non-sarcasm expression. The words with incongruity are colored.

Early attempts at sarcasm detection extract incongruity expressions by searching a set of positive verbs and negative situations [Bamman and Smith, 2015; Riloff *et al.*, 2013; González-Ibánez *et al.*, 2011], or by employing lexical features [Lunando and Purwarianti, 2013]. Recent advances have been made using deep learning algorithms to capture incongruity, including text-based [Riloff *et al.*, 2013; Joshi *et al.*, 2015; Wang *et al.*, 2023] and multimodal-based methods [Liang *et al.*, 2022]. These modern sarcasm detection systems rely primarily on neural networks and attention mechanisms. Poria et al., [Poria *et al.*, 2016] apply pre-trained CNN models to detect sarcasm. Both Tay et al. [Tay *et al.*, 2018] and Xiong et al. [Xiong *et al.*, 2019] develop attention-based models to capture contradiction information between word pairs and RNN-based models to get compositional information.

Moreover, in many scenarios, it is challenging to identify

*Corresponding author

semantic contradictions based solely on the literal clue, which requires commonsense or knowledge. As shown in the middle example in Figure 1, the listener cannot understand the speaker's satirical purpose unless they realize that the phrase "trafficjam" is commonly associated with words such as "nervous", "irritate", "impatient", etc. Therefore, commonsense knowledge is essential for sarcasm detection. Commonsense is the information often possessed by individuals in order to comprehend daily circumstances [Ilievski *et al.*, 2021b; Ilievski *et al.*, 2021a; Storks *et al.*, 2019]. This information enables people to link knowledge fragments and develop new conclusions. VEALE and Hao [Veale and Hao, 2010] highlight that the comprehension of sarcasm often relied on outside-the-text commonsense knowledge. Although [Li *et al.*, 2021] developed a new deep learning framework for sarcasm detection utilizing commonsense information, their use of commonsense is somewhat superficial and they are unable to precisely illustrate how commonsense works. As the third example in Figure 1, [Li *et al.*, 2021] that coarsely introduces commonsense would inevitably associate the phrase "protest" with the words "argument", "fight", "violent" and "angry" according to commonsense knowledge. Thus, there would be a clear contrast between the commonsense of "protest" and "perfect". If we solely consider the semantic information, [Li *et al.*, 2021] would incorrectly determine that the expression is sarcastic. Once we explore the dependency relationship derived from the parsing tree, we can discover that "perfect" is an adjectival complement of "measure" and the commonsense of "protest" is a modifier of the result. Thus, it is crucial to employ internal syntactic information while exploiting external commonsense when detecting sarcasm.

Based on the above findings, we pose the problem of incorporating both commonsense knowledge and dependency information into sarcasm detection. This problem is non-trail, as dependency information is a type of structural information, while commonsense knowledge is a type of content information. They are located in their own spaces, which makes it very challenging for sarcasm detection models to effectively integrate these two forms of information. Simply splicing commonsense knowledge into the corresponding sentence would destroy the syntactic structure of the sentence. Or, solely concatenating the learned commonsense embeddings and the sentence embedding cannot successfully address the issue of mismatch between two different modalities [He *et al.*, 2017], i.e., there may be informational conflicts and redundancies, etc.

In this paper, we propose a novel framework, called Commonsense Sentiment Dependency Graph Convolutional Network (CSDGCN), for commonsense knowledge and syntactic structure co-enhanced sarcasm detection by constructing a commonsense-augmented sentiment graph and a commonsense-replaced dependency graph for each instance. Specifically, we use a pre-trained BERT [Devlin *et al.*, 2018] to encode each instance, use COMET [Bosselut *et al.*, 2019] to generate commonsense knowledge for each processed instance, then put selected commonsense knowledge candidates into BERT to encode commonsense knowledge. To manifest the role of commonsense in sentimental inconsistency modeling, we construct a commonsense-augmented sentiment graph for each instance. Besides, to embody the role of commonsense in syntactic structure, we construct a commonsense-replaced dependency graph for each instance. Thus, the information in two different modal spaces can be transferred to the same space, while the most vital information in commonsense knowledge for sarcastic judgment is preserved. Then, we separately feed the aforementioned two graphs into GCNs to leverage the role of commonsense knowledge and syntactic structure in incongruity expression for sarcasm detection and apply a retrieval-based attention mechanism to derive sentiment and dependency graph-oriented features from contextual representations for final prediction.

The main contributions of our work can be summarized as follows:

- To the best of our knowledge, we are the first to use GCNs to exploit the roles of both external commonsense and internal syntax structure in sarcasm detection.
- A novel scenario of commonsense enhanced sentiment and dependency graphs construction, which explicitly depicts the role of commonsense knowledge and syntax structure, is explored to extract the incongruity expressions in sarcasm detection.
- Experiment results on several benchmark datasets reveal that our proposed method beats the state-of-the-art methods with better interpretability in sarcasm detection.

2 Method

2.1 Task Definition

Sarcasm detection aims to identify if an utterance has a sarcastic meaning in a given scenario. Formally, given a text X with n words, $X = \{x_1, x_2, ..., x_n\}$, where x_i represents a single word in the text. Our model is supposed to classify the given text into sarcasm or non-sarcasm categories correctly.

2.2 Framework

Figure 2 gives an overview of our model which is composed of 6 modules. The input sentence is converted into vector representations via an encoding context representation module. Then, we use the reasoning commonsense module to acquire sentence commonsense knowledge based COMET [Bosselut *et al.*, 2019], select the suitable ones, and covert them into vector representations.

After that, in the deriving sentiment and dependency graph module, we build a commonsense-augmented sentiment graph for each sentence with its commonsense knowledge, which contributes to paying more attention to words with opposite sentiments. We also build a commonsense-replaced dependency module for each sentence to reserve syntactic structure information. Next, we separately feed the sentiment graph and dependency graph into GCNs in the learning graph representation module and employ a retrieval-based attention mechanism in the attention module. Finally, in the prediction module, we concatenate sentiment and dependency graph-oriented features of the attention module, perform a mean-pooling operation to obtain a single vector, and put it into the full connect layer to get the final sarcasm detection result.

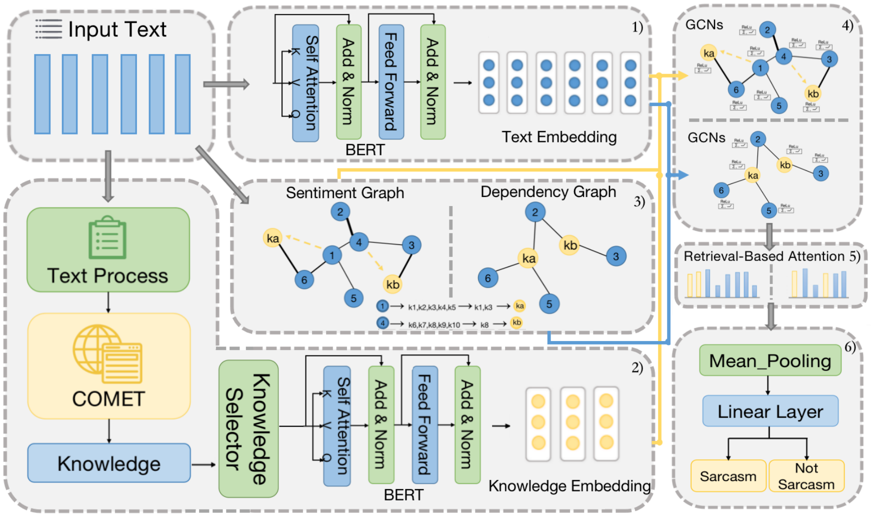

Figure 2: The architecture of our proposed model, which contains 6 modules: 1) Encoding Context Representation Module; 2) Reasoning Commonsense Module; 3) Deriving Sentiment and Dependency Graph Module; 4) Learning Graph Representation Module; 5) Attention Module; and 6) Prediction.

2.3 Encoding Context Representation Module

Given a sequence X containing n words, $X = \{x_1, x_2, ..., x_n\}$, n is the maximum length of the sequence. Then we feed X to BERT to encode the input sentence into vector representations: $\boldsymbol{X}_{enc} = BERT(X)$, where $\boldsymbol{X}_{enc} \in \mathbb{R}^{n\times d}$, which is the output of the last layer of BERT encoders and d is the embedding size. Here, BERT [Devlin *et al.*, 2018] is a pre-trained language model, which consists of multiple layers of bi-directional transformer encoders. The words' representations vary according to their contexts in BERT, thus resulting in outstanding representations of text.

2.4 Reasoning Commonsense Module

Most sarcasm detection work reckons that there are inconsistencies between words or segments in sarcastic statements, but it could be hard to recognize certain sentences as sarcastic without employing commonsense knowledge. For example, without commonsense knowledge, the sentence "Stucking in the trafficjam makes me happy" could be mistaken for a non-sarcastic sentence, leading to a misunderstanding. This example motivates us to develop the commonsense-assisted sarcasm detection method. We utilize **COM**monsE**nse** **T**ransformers (COMET)[Bosselut *et al.*, 2019] to generate commonsense knowledge bases automatically for each sentence. COMET is a framework to develop innovative and diversified commonsense knowledge tuples by adjusting the weights of language models. It comprises results from the ATOMIC[Sap *et al.*, 2019] and ConceptNet[Speer *et al.*, 2017] knowledge libraries. To generate candidates for commonsense knowledge, we adopt the same configuration as [Chakrabarty *et al.*, 2020] utilizing beam search with a size of 5. Our work utilizes the COMET model with ConceptNet tuples (subject-relation-object) to refine the model, and we only exploit the "causes" relation to generate knowledge candidates. Before feeding text into COMET, we conduct a series of text processing steps, including token lowercase, lemmatization, punctuation and stop words elimination.

After feeding the preprocessed sentence into COMET, five knowledge candidates are generated for each neutral word. There is a problem that simply using all the generated commonsense knowledge without making a choice would introduce a lot of noisy words to interfere with the classifier, so we need to select these commonsense. Obviously, sarcasm is always coupled with sentiment. As a result, sentiment polarity is considered in our work to select knowledge candidates. Given the knowledge set $K = \{k_1, k_2, ..., k_l\}$ (such as the knowledge set $\{k_1, k_2, ..., k_9, k_{10}\}$ of the input text in Figure 2), where k_i is the i^{th} knowledge candidate and l is the total number of knowledge candidates. The sentiment score of the knowledge candidate k_i is $s(k_i)$, which is calculated using SentiWordNet [Cambria *et al.*, 2020]. Here, we test three different sentiment polarity-based selection algorithms (i.e., Contrast Sentiment-Based Knowledge Selection, Majority Sentiment-Based Knowledge Selection, and Minority Sentiment-Based Knowledge Selection) to select the most useful commonsense as did in [Li *et al.*, 2021].

After performing the sentiment polarity-based knowledge selection algorithms, we can obtain $K_{selected} = \{k_1, k_2, ..., k_s\}$ for each text, such as the selected knowledge $\{k_1, k_3, k_8\}$ in Figure 2. Then, $K_{selected}$ is applied to sarcasm detection. The s selected commonsense knowledge is generated from m ($m \leq s$) words (such as the node 1 and node 4 in Figure 2) in X, and then put into BERT sequentially to obtain the commonsense knowledge embeddings. After that, for the commonsense knowledge embeddings from the same word, we exploit a mean-pooling to get the word's final commonsense knowledge embedding. Thus, the m words in X would have m commonsense vectors, i.e., $\boldsymbol{K}_{enc} \in \mathbb{R}^{m\times d}$.

2.5 Deriving Sentiment and Dependency Graph Module

As mentioned above, sentimental conflict is an important signal to judge a sentence as sarcastic. In order to model the sentimental conflict in sentences, and pay attention to the role of syntactic structure and commonsense knowledge, we build a commonsense-augmented sentiment graph and a commonsense-replaced dependency graph. Thus, we not only preserve the most vital sentiment information in commonsense knowledge and its role in the syntactic structure but also transform it into the same space as the syntactic structure.

Given the input sentence $X = \{x_1, x_2, ..., x_n\}$ and its corresponding selected knowledge set $K_{selected} = \{k_1, k_2, ..., k_s\}$, we could construct a commonsense-augmented sentiment graph based on the sentiment scores of words in X and $K_{selected}$ retrieved from SentiWordNet:

$$\boldsymbol{A}^S_{i,j} = \left| s(x_i) - s(x_j) \right|, \tag{1}$$

where $\boldsymbol{A}^S \in \mathbb{R}^{(n+m)\times(n+m)}$. When $i \in [1, n]$, $s(x_i)$ is the sentiment score of the word x_i in X. When $i \in (n, n+m]$, $s(x_i)$ is the average sentiment scores of all commonsense words generated by the same neutral word in X. In this way, words with opposite sentiments can be highly valued. Therefore, sentimental incongruity expressions in both sentences and commonsense knowledge can be propagated to distinguish contradictions between literal expressions and the true intention in sarcasm detection.

Furthermore, a sarcastic expression often depends on syntactic structure. Inspired by previous syntax-aware graph methods [Lou *et al.*, 2021; Tang *et al.*, 2020], we construct a dependency graph based on the parsing tree of the sentence $X = \{x_1, x_2, ..., x_n\}$:

$$\boldsymbol{A}^D_{i,j} = 1, \quad if\ \tau(x_i, x_j), \tag{2}$$

where $\boldsymbol{A}^D \in \mathbb{R}^{n\times n}$ and its initial remaining elements are 0, $\tau(x_i, x_j)$ indicates that there is a relationship between x_i and x_j in the dependency tree of the sentence. The syntax structure is obtained using the Stanford dependency parser. In order to illustrate how commonsense knowledge works in a dependency graph for sarcasm detection, we replace the word that can generate commonsense in $\boldsymbol{X}_{enc}$ to build the commonsense-replaced dependency graph, detailed steps are explained in the next module. It is exactly because of the commonsense-replaced dependency graph that we can clearly understand how the generated commonsense affects other nodes of words. We construct the undirected graph to enrich the sentiment and dependency information: $\boldsymbol{A}^D_{i,j} = \boldsymbol{A}^D_{j,i}$, $\boldsymbol{A}^S_{i,j} = \boldsymbol{A}^S_{j,i}$ as did in [Kipf and Welling, 2016], and also set a self-loop for each word in the dependency graph: $\boldsymbol{A}^D_{i,i} = 1$.

2.6 Learning Graph Representation Module

We now interactively use Sentiment Graph Convolutional Network and Dependency Graph Convolutional Network to learn the true intention behind the input text for sarcasm detection.

First, we feed the sentiment graph of the sentence into the multi-layers GCN architecture. Each node in the l^{th} GCN layer is updated according to the hidden representations of its neighborhoods according to the adjacency matrices of the sentiment graph. The process is defined as:

$$\boldsymbol{G}^l_S = RELU(\overline{\boldsymbol{A}^S}\boldsymbol{G}^{l-1}_S \boldsymbol{w}^l_S + \boldsymbol{b}^l_S), \tag{3}$$

where $\boldsymbol{G}^l_S \in \mathbb{R}^{(n+m)\times d}$ is the hidden graph representation evolved from the preceding GCN layer, $\boldsymbol{G}^l_S = \left\{\boldsymbol{g}^l_{S_1}, \boldsymbol{g}^l_{S_2}, ..., \boldsymbol{g}^l_{S_{n+m}}\right\}$and the original input nodes of the first GCN layer are the contexts and commonsense knowledge representation learned by BERT: $\boldsymbol{G}^0_S = \boldsymbol{H}^S_{enc} = [\boldsymbol{X}_{enc}; \boldsymbol{K}_{enc}] = \left\{\boldsymbol{h}^S_{enc_1}, \boldsymbol{h}^S_{enc_2}, ..., \boldsymbol{h}^S_{enc_{n+m}}\right\}$, $\boldsymbol{X}_{enc} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{K}_{enc} \in \mathbb{R}^{m\times d}$. $\overline{\boldsymbol{A}^S_i} = \boldsymbol{A}^S_i/(\boldsymbol{E}_i + 1), \boldsymbol{E}_i = \sum_{j=1}^{n} \boldsymbol{A}^S_{i,j}$. $\boldsymbol{w}^l_S \in \mathbb{R}^d, \boldsymbol{b}^l_S \in \mathbb{R}^d$ are the trainable parameters of the l^{th} GCN layer.

Then, we put the dependency graph into GCN:

$$\boldsymbol{G}^l_D = RELU(\overline{\boldsymbol{A}^D}\boldsymbol{G}^{l-1}_D \boldsymbol{w}^l_D + \boldsymbol{b}^l_D), \tag{4}$$

where $\boldsymbol{G}^l_D \in \mathbb{R}^{n\times d}$, $\boldsymbol{G}^l_D = \left\{\boldsymbol{g}^l_{D_1}, \boldsymbol{g}^l_{D_2}, ..., \boldsymbol{g}^l_{D_n}\right\}$, and $\boldsymbol{G}^0_D = \boldsymbol{H}^D_{enc} = \left\{\boldsymbol{h}^D_{enc_1}, \boldsymbol{h}^D_{enc_2}, ..., \boldsymbol{h}^D_{enc_n}\right\}$, $\boldsymbol{X}_{enc}$ is the initial embedding matrix of $\boldsymbol{H}^D_{enc}$. Assume that a word corresponding to the i^{th} embedding $\boldsymbol{h}^D_{enc_i}$ produces a final selected commonsense embedding $\boldsymbol{k}_{enc_j}$, then we replace the i^{th} embedding $\boldsymbol{h}^D_{enc_i}$ with $\boldsymbol{k}_{enc_j}$, so-called commonsense-replaced dependency graph, when performing GCN. Just like in Figure 2, node 1 and node 4 have generated commonsense k_a and k_b, we can get $\boldsymbol{K}_{enc} \in \mathbb{R}^{2\times d} = \{\boldsymbol{k}_{enc_a}, \boldsymbol{k}_{enc_b}\}$, and replace $\boldsymbol{h}^D_{enc_1}$ and $\boldsymbol{h}^D_{enc_4}$ with $\boldsymbol{k}_{enc_a}$ and $\boldsymbol{k}_{enc_b}$.

2.7 Attention Module

To obtain sentiment and dependency graph-oriented features from contextual representations respectively, we employ a retrieval-based attention mechanism:

$$\boldsymbol{z}_S = \sum_{t=1}^{n+m} y^S_t \boldsymbol{h}^S_{enc_t} \ , \ \boldsymbol{z}_D = \sum_{t=1}^{n} y^D_t \boldsymbol{h}^D_{enc_t}, \tag{5}$$

$$y^S_t = \frac{exp(x^S_t)}{\sum_{i=1}^{n+m} exp(x^S_i)} \ , \ y^D_t = \frac{exp(x^D_t)}{\sum_{i=1}^{n} exp(x^D_i)}, \tag{6}$$

$$x^S_t = \sum_{i=1}^{n+m} (\boldsymbol{h}^S_{enc_t})^\top \boldsymbol{g}^l_{S_i} \ , \ x^D_t = \sum_{i=1}^{n} (\boldsymbol{h}^D_{enc_t})^\top \boldsymbol{g}^l_{D_i}, \tag{7}$$

where $\boldsymbol{g}^l_D$ and $\boldsymbol{g}^l_S$ is the output of final GCN layer. We concatenate $\boldsymbol{z}_S$ and $\boldsymbol{z}_D$, and perform a mean-pooling operation to obtain a d-dimensional vector representing the final sarcastic representation:

$$\boldsymbol{z} = mean\_pooling([\boldsymbol{z}_S; \boldsymbol{z}_D]). \tag{8}$$

2.8 Prediction

Afterward, the final sarcastic representation $\boldsymbol{z}$ is introduced into a fully connected layer with softmax normalization to capture the probability distribution $\hat{\boldsymbol{y}}$ of sarcasm choice space:

Datasets	Train	Test
Twitter (Riloff)	1333	148
Twitter (Ptácek)	8497	1063
Twitter (Ghosh)	42717	10913

Table 1: Statistics of the experimental datasets.

$$\hat{\boldsymbol{y}} = softmax(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{b}), \tag{9}$$

where $\hat{\boldsymbol{y}} \in \mathbb{R}^{d_p}$ is the predicted sarcastic probability for the input sentence, d_p is the dimensionality of sarcasm labels. $\boldsymbol{W} \in \mathbb{R}^{d_p * d}$ and $\boldsymbol{b} \in \mathbb{R}^{d_p}$ are trainable parameters.

2.9 Learning Objective

We minimize the cross-entropy loss via the standard gradient descent algorithm to train the model:

$$\eta = -\sum_{i=1}^{N}\sum_{j=1}^{d_p} \boldsymbol{y}_i^j \log \hat{\boldsymbol{y}}_i^j + \lambda \left\|\theta\right\|^2, \tag{10}$$

where *N* is the training data size, $\boldsymbol{y}_i$ and $\hat{\boldsymbol{y}}_i$ respectively represent the ground-truth and estimated label distribution of instance i, θ denotes all trainable parameters of the model, λ represents the coefficient of L2-regularization.

3 Experiment

3.1 Datasets

We evaluate our model on three datasets, including Twitter datasets proposed by Ghosh et al. [Ghosh and Veale, 2017], Ptácek et al. [Ptáček *et al.*, 2014], and Riloff et al. [Riloff *et al.*, 2013]. We denote the three datasets as Twitter (Ghosh), Twitter (Ptácek), and Twitter (Riloff). In our work, each sample consists of a sequence of text with associated commonsense knowledge generated by COMET. Detailed statistics are summarized in Table 1.

3.2 Experimental Settings

In our experiments, the number of GCN layers is set to 3. The coefficient λ of L2 regularization is set to 0.01. Adam is utilized as the optimizer with the default learning rate of 0.001 to train the model, and the mini-batch size is 256 for Twitter (Ghosh), 64 for Twitter (Ptácek), and 8 for Twitter (Riloff). We use the pre-trained cased BERT-base [Devlin *et al.*, 2018] with 768-dimensional embedding. We perform Accuracy (Acc.), Macro F1-score (F1), Precision (P) and Recall (R) to measure the performance of the models.

3.3 Baseline Models

We compare our model, i.e. **CSDGCN_maj**, **CSDGCN_min** and **CSDGCN_con** (represent our model under majority, minority and contrast sentiment-based knowledge selection strategies respectively), with the following 8 baselines: 1) statistic technique: **NBOW** [Tay *et al.*, 2018]; 2) conventional neural networks: **TextCNN** [Kim, 2014], **ATT-LSTM** [Yang *et al.*, 2016]; 3) BERT-based models: **BERT** [Devlin *et al.*, 2018], **KnowBERT** [Peters *et al.*, 2019]; 4) sarcasm detection methods: **SAWS** [Pan *et al.*, 2020], **ADGCN** [Lou *et al.*, 2021], **SarDeCK** [Li *et al.*, 2021]. Note that KnowBERT and SarDeCK have employed a knowledge integration module.

3.4 Experimental Results

Table 2 reports the performance comparison of all models that do/do not incorporate external knowledge on three datasets. We observe that our model CSDGCN achieves the best performance on all three datasets in terms of Accuracy, Macro F1-score, Precision and Recall. Our method outperforms ADGCN on all three datasets, improving model Accuracy and F1-score by around 4.06% and 5.79% on Riloff dataset respectively. Compared with SarDeCK, our results have an improvement of 5.41%, 0.21% and 2.02% in terms of Accuracy on the three datasets respectively. Consistent with our motivation, incorporating both commonsense and syntax information contributes to the extraction of conflicting implications and incongruity expressions for sarcasm detection.

It is worth noting that the pre-trained BERT-based model outperforms the traditional deep learning models in most circumstances. We owe these results to the outstanding text representations of BERT. Although KnowBERT only concatenates the text representation and knowledge representation, the results of KnowBERT are superior to that of BERT. This demonstrates the significance of commonsense knowledge in sarcasm detection. In addition, SarDeCK achieves superior results than KnowBERT because its modeling of semantic inconsistencies and sentimental conflicts in sentences is more suited to the task of sarcasm detection. Our model CSDGCN utilizes the GCN model to exploit the roles of both external commonsense information and internal syntactic structure, hence achieving the best results. With the help of commonsense, we can get external information about the implied true meaning of words; with the help of syntactic information, we can truly understand the structure of the sentence rather than treat a sentence as a word sequence. As the node in the GCN layers could be updated according to the hidden representations of its neighborhoods, we feed the commonsense-augment sentiment graph and commonsense-replaced dependency graph to GCNs to capture the long-range sentiment semantic contradiction.

The commonsense knowledge selection strategies are also very important. In many cases, improper knowledge selection results in the wrong classification. Using all the generated commonsense knowledge without making a selection makes it tend to introduce a large number of noisy terms that interfere with the classification. As seen in the bottom three rows of Table 2, different selection strategies result in the best results on different datasets, indicating that the optimal knowledge selection strategy might depend on the data distributions.

3.5 Ablation Study

To analyze the impact of different components of the proposed CSDGCN bring to the performance, we conduct an ablation study and report the results in Table 3. The results without the sentiment graphs perform the worst, which shows the importance of capturing sentimental conflict in sarcasm detection. Furthermore, the model without dependency graphs also leads to considerably poorer performance. This implies

Methods	Twitter (Riloff)				Twitter (Ptácek)				Twitter (Ghosh)			
	Acc	F1	P	R	Acc	F1	P	R	Acc	F1	P	R
NBOW	75.80	74.78	75.23	74.39	71.24	71.20	71.22	71.34	77.10	77.86	78.21	77.76
ATT-LSTM	77.70	59.64	58.50	60.37	73.63	73.29	73.06	71.62	75.20	70.14	68.75	69.83
TextCNN	75.12	62.69	62.81	62.24	77.37	78.32	77.60	79.32	84.84	83.15	83.30	82.96
BERT	79.72	63.61	72.59	56.61	78.64	79.50	76.36	70.04	79.23	78.83	70.85	80.60
SAWS	77.70	55.10	50.53	60.81	73.44	72.13	72.46	71.92	82.28	81.36	81.11	81.68
ADGCN	83.10	75.88	76.77	75.13	82.12	81.55	81.37	81.80	82.90	82.81	82.94	82.76
KnowBERT	80.45	71.77	73.85	69.81	78.83	77.06	76.69	77.37	80.38	80.04	79.80	80.28
SarDeCK	81.75	63.26	70.73	61.27	84.74	84.36	84.09	84.84	83.91	83.33	82.90	84.35
CSDGCN_maj	86.14	79.42	79.42	79.42	**84.95**	**84.49**	**84.11**	**84.87**	83.20	80.90	80.44	81.37
CSDGCN_min	**87.16**	**81.67**	**82.75**	**80.74**	83.36	83.29	82.59	84.01	**85.93**	**84.61**	**84.25**	**84.96**
CSDGCN_con	84.46	76.81	79.46	75.03	82.90	81.40	81.76	81.03	84.06	83.62	83.58	83.60

Table 2: Experimental results (%) on three different datasets. The best scores are in bold.

Model	Riloff	Páteck	Ghosh
CSDGCN	**87.16**	**84.95**	**85.93**
w/o A	78.37	77.79	78.29
w/o R	85.13	81.93	83.00
w/o S	77.70	74.98	78.42
w/o D	80.40	81.74	80.47

Table 3: The Accuracy results of ablation study. A denotes commonsense augmentation, R is the commonsense replacement, S represents the sentiment graph, and D denotes the dependency graph.

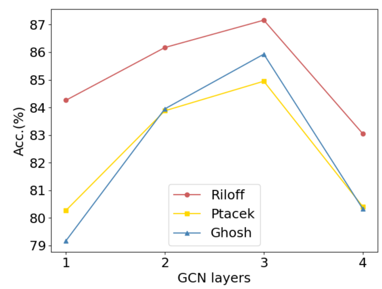

Figure 3: Impact of the number of GCN layers.

that syntax-aware information advances the model to learn the syntax structure of sentences rather than merely regarding a sentence as a word sequence. Besides, the removal of "commonsense augment" and "commonsense replacement" apparently degrades the performance, which indicates that incorporating commonsense in sentiment graphs and dependency graphs plays a critical role in understanding the implied semantics of sentences.

3.6 Impact of GCN Layers

To investigate the influence of the number of GCN layers in our model, we set the number of GCN layers to values ranging from 1 to 4. As seen in Figure 3, 3-layer GCN achieves the best experimental results, hence the number of GCN layers is fixed to 3 in CSDGCN. 1-layer GCN underperforms on all datasets, indicating that the network structure is insufficient to utilize satire-specific properties. Although the effect of 2-layer GCN is significantly better than that of 1-layer GCN, the learning capacity of the model is still insufficient. When a 4-layer GCN is incorporated into the model, the effect decreases dramatically, indicating overfitting due to the excessive number of layers.

3.7 Model Analysis

In this section, we provide some examples correctly identified by our model but misclassified by ADGCN or SarDeCK. Additionally, we extract the attention maps to investigate how commonsense knowledge and syntactic structure contribute to enhanced performance. Our method can obtain sentiment-oriented attention scores (i.e., y^S) and syntax-oriented attention scores (i.e., y^D). Figure 4 gives an attention visualization of the samples with their selected commonsense words. We also give an analysis of the wrongly predicted sample.

Case study: Sentiment-oriented attention scores (i.e., blue color) show that "perfect" and "bad" are important words in the first example. This sentimental inconsistency is easily identified as sarcasm without the aid of syntactic structure, such as the misclassification of SarDeCK. Due to the necessary syntax, our model can clearly learn that "perfect" is used to describe "measure", and "bad" is an adjective of "result", which cannot be simply regarded as a sentimental contradiction. Thus the syntax-oriented attention scores (i.e., green color) learned by our model give "measure" and "result" higher marks. Meanwhile, it can be seen that syntax-oriented attention gives "although" a higher attention score, indicating that this conjunction plays an important role in the syntax of the sentence. In fact, the different conjunctions may lead to different meanings, e.g, "This measure sounds perfect, so of course it gets a bad result", which is obviously a sarcastic sentence. Note that ADGCN also obtains the correct result for this example, since it also incorporates syntactic structure.

The second example, "Stucking in the trafficjam makes me happy" is difficult for ADGCN to classify as sarcastic because it lacks commonsense knowledge of the word "trafficjam". The word "trafficjam" is a neutral word in SentiWordNet. Actually, "trafficjam" implies "nervous", "irritate", and "accident" which are all negative phrases. Utilizing commonsense knowledge, both our model and SarDeCK correctly classify the sentence as sarcastic. Our model gives "happy"

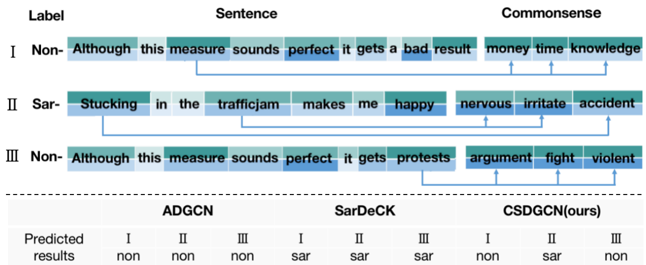

Figure 4: The learned attention scores of sentences with their commonsense words. In the top part of the figure, the bluer color in the word box represents the larger value of the sentiment-oriented attention score, the greener color in the word box represents the larger value of the syntax-oriented attention score, and the blue lines indicate which word generates commonsense words. The lower part of the figure depicts the classification results of models.

and the selected commonsense knowledge terms "nervous "and "irritate" higher sentiment-oriented attention scores, allowing sarcasm to be reliably detected using incongruity information. In addition, the overall attention scores enable us to identify which commonsense word plays a crucial role in sarcasm detection, e.g., "nervous" and "irritate" are more important than "accident".

Although SarDeCK can obtain the commonsense knowledge of the word "protest" in the third example, it incorrectly interprets this example as irony due to the absence of syntactic structural information. With the replacement of commonsense in the dependency graph, our model can learn that commonsense such as "violent", "argument" is generated by "protest", and the "perfect" is a description of "measure", they cannot be simply judged as sentimental inconsistency and recognized as irony. Thus, our model correctly classified this example as non-sarcastic due to the utilization of a commonsense-augmented sentiment graph and a commonsense-replaced dependency graph.

Error example: To understand the sarcastic sentence, "thanks to my wonderful mom she wrapped all the presents including me", the classifier must detect that the phrase "wrapped all gifts, including me" conveys a negative feeling, communicating dissatisfaction with the mother's behavior. In this case, however, the COMET used by our method just leverages the cause relationships to produce commonsense information such as "blessings", "happy", and "surprising". Clearly, the superficial commonsense lacks an understanding of the relationships between entities and is unable to recognize the irony in this circumstance. Therefore, our model lacks an understanding of the relationships between entities and cannot predict this sentence correctly.

4 Related Work

Since our work is text-based sarcasm detection, we highly introduce the related work of text-only sarcasm detection. The existing text-based sarcasm detection works can be divided into three categories: rule-based approaches, feature-based machine learning approaches, and deep learning approaches.

Rule-based approaches aim to identify sarcasm with fixed patterns. Veale et al.[Veale and Hao, 2010] present a nine-step method for separating ironic from non-ironic similes with the help of Google searches. Hashtags in Twitter messages, such as "sarcasm", "sarcastic", "not", "not true", "greatstart" were also considered in sarcasm detection. The hashtags were labeled by users to express their feelings. Maynard et al. [Maynard and Greenwood, 2014] developed a hashtag tokenizer, such that sentiment and sarcasm within hashtags can be detected. They also compiled a number of rules to improve the accuracy of sentiment classification when sarcasm is known to be present.

However, rule-based approaches only depend on fixed patterns, which makes it difficult to tackle complex sarcastic texts. The scholars began to enrich the feature set and use machine learning approaches. Ghosh et al. [Ghosh *et al.*, 2015] used an SVM classifier with a modified kernel and word embeddings. They treated the sarcasm detection task as a type of word sense disambiguation problem. Joshi et al. [Joshi *et al.*, 2015] developed a system that harnesses context incongruity to detect sarcasm, which incorporated both explicit incongruity features and implicit incongruity features.

Although feature-based algorithms achieved promising performance in sarcasm detection, the construction of discrete features is a time-consuming job. Researchers have recently considered deep learning-based methods due to its capability of extracting features automatically. Poria et al. [Poria *et al.*, 2016] used pre-trained CNNs to extract sentiment, emotion and personality features for sarcasm detection. Pan et al. [Pan *et al.*, 2020] presented a novel model by introducing snippet-level self-attention to model the incongruity between sentence snippets, which aims to address the problem that existing models are inefficient in identifying the sarcasm caused by sentence snippet incongruity. Li et al. [Li *et al.*, 2021] proposed a novel BERT-based model that can effectively process commonsense knowledge, aiming to address the issue that existing sarcasm detection approaches fail to involve commonsense knowledge to identify sarcasm. Lou et al. [Lou *et al.*, 2021] proposed a novel scenario of constructing an affective graph and a dependency graph for each sentence to learn the contradictory implications and incongruity expressions in sarcasm detection. However, it is indispensable to have both syntax information and commonsense knowledge in many situations.

5 Conclusion

In this paper, we present a novel framework for detecting sarcasm by explicitly integrating commonsense knowledge and syntactic structure. Concretely, we apply the COMET model to generate relevant commonsense for each sentence and explore a novel scenario of constructing a commonsense-augmented sentiment graph and a commonsense-replaced dependency graph for each text. Our CSDGCN depicts the role of external commonsense and inconsistent expressions over the context for sarcasm detection by interactively modeling the sentiment and dependency information. Experimental results on several benchmark datasets reveal that our proposed method outperforms the state-of-the-art methods for sarcasm detection, and also explicitly illustrates the role of commonsense knowledge.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (No. 62272340, 62276185, 62172056).

References

[Bakshi *et al.*, 2016] Rushlene Kaur Bakshi, Navneet Kaur, Ravneet Kaur, and Gurpreet Kaur. Opinion mining and sentiment analysis. In *Proceedings of 3rd international conference on computing for sustainable development*, pages 452–455, 2016.

[Bamman and Smith, 2015] David Bamman and Noah Smith. Contextualized sarcasm detection on twitter. In *Proceedings of the 29th international Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence*, pages 574–577, 2015.

[Bosselut *et al.*, 2019] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*, 2019.

[Cambria *et al.*, 2020] Erik Cambria, Yang Li, Frank Z Xing, Soujanya Poria, and Kenneth Kwok. Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 105–114, 2020.

[Chakrabarty *et al.*, 2020] Tuhin Chakrabarty, Debanjan Ghosh, Smaranda Muresan, and Nanyun Peng. r^3: Reverse, retrieve, and rank for sarcasm generation with commonsense knowledge. *arXiv preprint arXiv:2004.13248*, 2020.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Ghosh and Veale, 2017] Aniruddha Ghosh and Tony Veale. Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 482–491, 2017.

[Ghosh *et al.*, 2015] Debanjan Ghosh, Weiwei Guo, and Smaranda Muresan. Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *proceedings of the 2015 conference on Empirical Methods in Natural Language Processing*, pages 1003–1012, 2015.

[González-Ibánez *et al.*, 2011] Roberto González-Ibánez, Smaranda Muresan, and Nina Wacholder. Identifying sarcasm in twitter: a closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 581–586, 2011.

[He *et al.*, 2017] Dongxiao He, Zhiyong Feng, Di Jin, Xiaobao Wang, and Weixiong Zhang. Joint identification of network communities and semantics via integrative modeling of network topologies and node contents. In *Proceedings of 31th Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence*, 2017.

[Ilievski *et al.*, 2021a] Filip Ilievski, Alessandro Oltramari, Kaixin Ma, Bin Zhang, Deborah L. McGuinness, and Pedro Szekely. Dimensions of commonsense knowledge. *Knowledge-Based Systems*, 229:107347, 2021.

[Ilievski *et al.*, 2021b] Filip Ilievski, Pedro Szekely, and Bin Zhang. Cskg: The commonsense knowledge graph. In *Proceeding of 18th European Semantic Web Conference*, pages 680–696, 2021.

[Jiang *et al.*, 2020] Nan Jiang, Fang Tian, Jin Li, Xu Yuan, and Jiaqi Zheng. Man: mutual attention neural networks model for aspect-level sentiment classification in siot. *IEEE Internet of Things Journal*, 7:2901–2913, 2020.

[Joshi *et al.*, 2015] Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 757–762, 2015.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751, 2014.

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

[Li *et al.*, 2021] Jiangnan Li, Hongliang Pan, Zheng Lin, Peng Fu, and Weiping Wang. Sarcasm detection with commonsense knowledge. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3192–3201, 2021.

[Liang *et al.*, 2022] Bin Liang, Chenwei Lou, Xiang Li, Min Yang, Lin Gui, Yulan He, Wenjie Pei, and Ruifeng Xu. Multi-modal sarcasm detection via cross-modal graph convolutional network. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

[Liu and others, 2010] Bing Liu et al. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2:627–666, 2010.

[Lou *et al.*, 2021] Chenwei Lou, Bin Liang, Lin Gui, Yulan He, Yixue Dang, and Ruifeng Xu. Affective dependency graph for sarcasm detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1844–1849, 2021.

[Lunando and Purwarianti, 2013] Edwin Lunando and Ayu Purwarianti. Indonesian social media sentiment analysis with sarcasm detection. In *Proceedings of the 5th International Conference on Advanced Computer Science and Information Systems*, pages 195–198, 2013.

[Maynard and Greenwood, 2014] Diana G Maynard and Mark A Greenwood. Who cares about sarcastic tweets?

investigating the impact of sarcasm on sentiment analysis. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, 2014.

[Miao *et al.*, 2020] Zhengjie Miao, Yuliang Li, Xiaolan Wang, and Wang-Chiew Tan. Snippext: Semi-supervised opinion mining with augmented data. In *Proceedings of 29th The World Wide Web Conference*, pages 617–628, 2020.

[Pan *et al.*, 2020] Hongliang Pan, Zheng Lin, Peng Fu, and Weiping Wang. Modeling the incongruity between sentence snippets for sarcasm detection. In *Proceedings of the 24th European Conference on Artificial Intelligence*, pages 2132–2139. 2020.

[Peters *et al.*, 2019] Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*, 2019.

[Poria *et al.*, 2016] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. A deeper look into sarcastic tweets using deep convolutional neural networks. *arXiv preprint arXiv:1610.08815*, 2016.

[Ptáček *et al.*, 2014] Tomáš Ptáček, Ivan Habernal, and Jun Hong. Sarcasm detection on czech and english twitter. In *Proceedings of the 25th International Conference on Computational Linguistics*, pages 213–223, 2014.

[Riloff *et al.*, 2013] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 conference on Empirical Methods in Natural Language Processing*, pages 704–714, 2013.

[Sap *et al.*, 2019] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035, 2019.

[Speer *et al.*, 2017] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[Storks *et al.*, 2019] Shane Storks, Qiaozi Gao, and Joyce Y Chai. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*, 2019.

[Tang *et al.*, 2020] Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6578–6588, 2020.

[Tay *et al.*, 2018] Yi Tay, Luu Anh Tuan, Siu Cheung Hui, and Jian Su. Reasoning with sarcasm by reading in-between. *arXiv preprint arXiv:1805.02856*, 2018.

[Veale and Hao, 2010] Tony Veale and Yanfen Hao. Detecting ironic intent in creative comparisons. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 765–770. 2010.

[Wang *et al.*, 2020] Xiaobao Wang, Di Jin, Katarzyna Musial, and Jianwu Dang. Topic enhanced sentiment spreading model in social networks considering user interest. In *Proceedings of the 34th Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence*, pages 989–996, 2020.

[Wang *et al.*, 2023] Xiaobao Wang, Yiqi Dong, Di Jin, Yawen Li, Longbiao Wang, and Jianwu Dang. Augmenting affective dependency graph via iterative incongruity graph learning for sarcasm detection. In *Proceedings of the 37th Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence*, 2023.

[Xiong *et al.*, 2019] Tao Xiong, Peiran Zhang, Hongbo Zhu, and Yihui Yang. Sarcasm detection with self-matching networks and low-rank bilinear pooling. In *Proceeding of 28th The World Wide Web Conference*, pages 2115–2124, 2019.

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.