Noname manuscript No.
(will be inserted by the editor)

CWI: A multimodal deep learning approach for named entity recognition from social media using character, word and image features

Meysam Asgari-Chenaghlu *[1] · M.Reza Feizi-Derakhshi [1] · Leili Farzinvash[1] · M. A. Balafar [1] · Cina Motamed[2]

Received: DD Month YEAR / Accepted: DD Month YEAR

Abstract Named Entity Recognition (NER) from social media posts is a challenging task. User generated content that forms the nature of social media, is noisy, and contains grammatical and linguistic errors. This noisy content makes tasks such as NER much harder. We propose two novel deep learning approaches utilizing multimodal deep learning and Transformers. Both of our approaches use image features from short social media posts to provide better results on the NER task. On the first approach, we extract image features using InceptionV3 and use fusion to combine textual and image features. This approach presents more reliable name entity recognition when the images related to the entities are provided by the user. On the second approach, we use image features combined with text and feed it into a BERT-like Transformer. The experimental results using precision, recall, and F1 score metrics show the superiority of our work compared to other state-of-the-art NER solutions.

Keywords Deep Learning · Named Entity Recognition · Multimodal Learning · Transformer

1 Introduction

A common social media delivery system such as Twitter supports various media types like video, image, and text. This media allows users to share their short posts called Tweets. Users can share their tweets with other users that are usually following the source user. However, there are rules to protect the privacy of users from unauthorized access to their timeline [1]. The very nature of user interactions on Twitter micro-blogging social media is oriented towards their

* Corresponding author
E-mail: m.asgari.c@tabrizu.ac.ir
[1] Department of Computer Engineering, University of Tabriz, Tabriz, Iran
[2] University of Orleans, Orleans, France

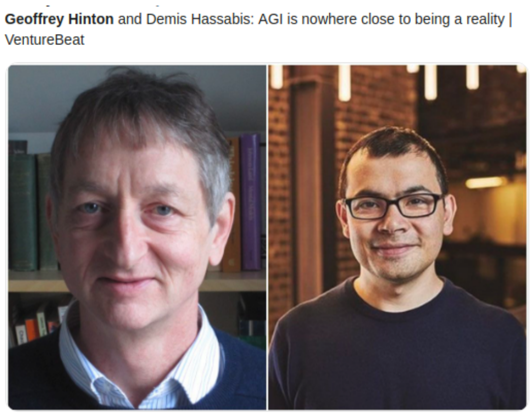
daily life, first witness news-reporting and engaging in various events (sports, political stands, etc.). According to studies, news on Twitter is propagated and reported faster than conventional news media [2].

Analysis and valuation of extracted information from such important media acts as a facilitator of many possibilities [3]. However, there are technical and scientific requirements to accomplish it. One of these requirements is NER. NER is the task of extracting important entities from textual data [4]. These important entities can vary from subject to subject but some entities are commonly known: person names, locations, and organizations. On the other hand, NER is also a widely investigated and researched subject in natural language processing [5, 6]. Most of the previous works only keep eye on the formal text. The formal text is a noise-free text written with minimum textual errors. Twitter is a social media that users are generating content most of the time and contains noise even in the texts written by experts. In some cases, the limitations of the Twitter platform also force users to write noisy text. But before investigating and finding a solution, it is mandatory to have a clear definition and categorization of the social media textual noise to address the problem properly.

Textual noise in social media is defined as **grammar mistakes**, **mistypes**, new words that are invented by users (**user invented words**), and internet abbreviations (**internet slang**) [7]. A *mistype* is recognized as an unintentional error in character sequences typed by a user. This type of noise can even have different patterns where users use various devices to input their text; for example, a mobile virtual keyboard has a different unintentional mistype pattern compared to a PC keyboard. *Grammar mistake* or error is an instance of faulty, unconventional, or controversial usage, such as a misplaced modifier or an inappropriate verb tense. Furthermore, there are also *abbreviated* and *informal sentences* that are intentional or unintentional. *Invented words* that are constantly growing is another major drawback for conventional NLP methods. The word "Selfie" is one of the famous ones in this category and after its invention by users, it is so commonly used that is familiar and treated like a real word from a dictionary. *Internet abbreviations* are another widely used category in social media; the word "Srsly" is an informal form of writing the word "Seriously". A major reason for such problems is the Twitter character limit for message length and users are not allowed to pass that. This limit was set to 140 to obey the SMS limit but after years it is updated to 280 Unicode characters [1]. Users in social media like Twitter are intentionally writing words in an abbreviated form to obey the character limit by dropping some characters. In cases, it is not a linguistic form of an abbreviation like the word "USA" (United States of America) [8, 9]. Some of such abbreviations are not present in formal corpus and accordingly, solutions based on formal corpus can not address them.

According to the noise definition, a NER model must address the noise problem to obtain acceptable results in Twitter-like social media. Dependency

[1] https://developer.twitter.com/en/docs/counting-characters



Geoffrey Hinton and Demis Hassabis: AGI is nowhere close to being a reality | VentureBeat

Fig. 1: A Tweet containing Image and Text: *Geoffrey Hinton* and *Demis Hassabis* are referred in text and respective images are provided with Tweet.

of many other NLP-based solutions and tasks directly to the NER results gives this task extra credit in social media. For example, detection of events, hot topics, or trending topics from social media can be done by many methods and systems [10] while a good NER can extract the underlying entities [4]. Having the entities and the events at hand, one can easily infer any related information about a person or an entity occurring inside an event.

Although Twitter has noisy content, tweets also carry extra information about the context users are sharing. For example, Figure 1 shows a tweet containing both the textual and visual content. This visual content is related to the textual content and is another description of the tweet. The form of data that is shared by this tweet is also referred to as multimodal data because it has more than one modality: text and image [11]. Utilization of image in the extraction of named entities can be very useful [12]. In cases where tweets are related to a person or an organization, the visual content is also relevant. For example, in Figure 1 text carries information about *Geoffrey Hinton* and *Demis Hassabis* and the image shows pictures of them. In cases where the NER model never saw these words or does not have any information about them in the trainset, visual content can guide it by creating a bias on human name rather than organization [12].

This intuition of utilizing multimodal data is used by researchers to provide a better solution for NER rather than conventional methods [12, 13]. However, still, some research questions remain unanswered. For example, in case of noise, only concatenated representations of word and characters with no feature extraction layers is not efficient enough. In the case of expelling visual content and just putting the text in mind, the model can not separate cases like shown in Figure 1 that both names can be person names or organizations at the same time. Also keeping the focus on the word embeddings or character embeddings disables the model from using shallow transfer learning provided by word embedding models such as GloVe [14] or fastText [15, 16]. On the other hand, only a character representation is not enough because it is hard to accomplish transfer learning on character-level. Only using token-based transfer learning also creates an out of vocabulary (OoV) problem for the model too. OoV usually happens on models where there is no embedding for the new token in the test phase. For example, if a model uses a transfer learning based on Word2Vec [17], if it sees a new word that was not present at the training corpus of Word2Vec, it will yield an error. This kind of error is prevented by using `[UNK]` (i.e., unknown token) which is a random vector.

In order to provide a better solution for NER task in social media and address the problems such as noise and OoV tokens, we present two different solutions based on multimodal learning. In our first solution, we use three deep learning-based feature extractors based on three different aspects: Character, Word, and Visual level. The character feature extractor only uses the character sequence of the tweet and provides a final feature vector accordingly. This feature extractor is noise rigid where there are OoV tokens and mistypes. Word feature extractor basically uses the shallow transfer learning provided from two pretrained word embedding models: GloVe and fastText. GloVe and fastText together provide a higher performance compared to cases where one is used. fastText on the other hand, can capture morphological changes in words and provide vectors in cases where the word is not seen before but its different morphological form has been seen. Visual content, is converted to top entity features using InceptionV3 [18]. The concatenated form of these three features guides the model in a noise-robust form. The second proposed model utilizes a BERT-like multimodal Transformer approach. This approach takes the both text and the visual entity features and provides the entity tags for the sequence. In this approach, we try to investigate the performance improvement made by a totally different architecture based on an autoencoder Transformer model [19, 20].

The novelty of our work lies in efficient utilization of character, word and image feature extractors to find a better solution for noise in social media. In order to address these issues, we designed WCI and multimodal Transformer; both of these models use subword based textual features. In case of WCI, it uses character features to address the any textual mistypes and the GloVe/fastText combination provides efficient understanding of words. However, the image features that are present in both of the models, creates a bias to make both models perform better in terms of evaluation metrics. We also show that separately using each one of these features helps to solve the problem but efficiently combining them with their best attributes works better.

The rest of the paper is organized as follows: Section 2 provides an insight view of previous methods; Section 3 describes the method we propose; Section 4 shows experimental evaluation and test results; discussion and future works are presented and discussed in Section 5; finally, Section 6 concludes the whole article.

2 Related Work

Many algorithms and methods have been proposed to classify or extract information from a single type of data such as audio, text, image, etc. However, in the case of social media, data comes in a variety of types such as text, image, video or audio in a bounded style. Most of the time, it is very common to caption a video or image with textual information. This information about the video or image can refer to a person, location and etc. From a multimodal learning perspective, jointly computing such data is considered to be more valuable in terms of representation and evaluation.

NER task, on the other hand, is the task of recognizing named entities from a sentence or group of sentences in a document format. Named entity is formally defined as a word or phrase that clearly identifies an item from set of other similar items [21, 22]. Equation 1 expresses a sequence of tokens.

$$ls = \langle w_1, w_2, \ldots, w_n \rangle, \tag{1}$$

$$o = \langle I_s, I_e, t \rangle, \tag{2}$$

$$o = \langle T_1, T_2, \ldots, T_n \rangle. \tag{3}$$

From this equation, the NER task is defined as the recognition of tokens that correspond to interesting items. These items from natural language processing perspective are known as named entity categories; BIO2 proposes four major categories, namely, organization, person, location and miscellaneous [23]. From the biomedical domain, gene, protein, drug and disease names are known as named entities [24, 25]. The output of NER task is formulated in 2. $I_s \in [1, N]$ and $I_e \in [1, N]$ is the start and end indices of each named entity and t is named entity type [26].

BIO2 tagging for NER is defined in equation 3. Table 1 shows BIO2 tags and their respective meanings; B and I indicate beginning and inside of the entity respectively, while O shows the outside of entity. Even though many tagging standards have been proposed for NER task, BIO is the foremost accepted by many real world applications [27].

A named entity recognizer gets s as input and provides entity-tags for each token. This sequential process requires information from the whole sentence rather than only tokens and for that reason, it is also considered to be a sequence tagging problem. Another analogous problem to this issue is part of speech tagging and some methods are capable of doing both [28]. However, in cases where noise is present and the input sequence has linguistic typos, many methods fail to overcome the problem. As an example, consider a sequence of tokens where a new token invented by social media users gets trended. This trending new word is misspelled and is used in a sequence along with other tokens in which the whole sequence does not follow known linguistic grammar. For this special case, classical methods and those which use engineered features do not perform well. Modern machine learning approaches such as deep learning and character or subword level models perfom better in such problems [29].

Table 1: BIO Tags and their respective meaning.

Begin	End	Description
B-PER	I-PER	Person
B-LOC	I-LOC	Location
B-ORG	I-ORG	Organization
B-MISC	I-MISC	Miscellaneous
O	O	Outside of entity

Using the sequence s itself or adding more information to it divides NER into two approaches: *unimodal* and *multimodal*. Although many approaches for NER have been proposed and reviewing them is not in the scope of this article, we focus on foremost analogues classical and deep learning approaches for NER in two subsections. In Section 2.1, unimodal approaches for NER are presented while in Section 2.2, emerging multimodal solutions are described.

2.1 Unimodal Named Entity Recognition

The recognition of named entities from only textual data (unimodal learning approach) is a well studied and explored research field. For a prominent example of this category, the Stanford NER is a widely used baseline for many applications [30]. The incorporation of non-local information in IE (information extraction) is proposed by the authors using Gibbs sampling. The conditional random field (CRF) approach used in this article, constructs a chain of cliques, where each clique represents the probabilistic relationship between two adjacent states. Also, the Viterbi algorithm has been used to infer the most likely state in the CRF output sequence. Equation 4 shows the proposed CRF method.

$$p(o|s) = \frac{\prod_{i=1}^{n} \phi_i(o_{i-1}, o_i, s)}{\sum_{o' \in o} \prod_{i=1}^{n} \phi_i(o'_{i-1}, o'_i, s)} \tag{4}$$

where ϕ is the potential function.

CRF finds the most probable likelihood by modeling the input sequence of tokens s as a normalized product of feature functions. In a simpler explanation, CRF outputs the most probable tags that follow each other. For example, it is more likely to have an *I-PER*, *O* or any other that that starts with *B-* after *B-PER* rather than encountering tags that start with *I-*.

T-NER is another approach that is specifically aimed to conduct NER task in Twitter [13]. A set of algorithms in their original work have been published to perform tasks such as POS (part of speech tagging), named entity segmentation and NER. Labeled LDA has been used by the authors in order to outperform baseline in [31] for NER task. Their approach strongly relies on the dictionary, contextual and orthographic features.

Deep learning techniques use distributed word or character representation rather than raw one-hot vectors. Most of this research in NLP field use pre-trained word embeddings such as *Word2Vec* [17], *GloVe* [14] or *fastText* [16]. These low dimensional real valued dense vectors have proved to provide better representation for words compared to one-hot vector or other space vector models.

The combination of word embedding along with bidirectional long-short term memory (LSTM) neural networks are examined in [28]. The authors also propose to add a CRF layer at the end of their neural network architecture in order to preserve output tag relativity. Utilization of recurrent neural networks (RNN) provides better sequential modeling over data. However, only using sequential information does not result in major improvements because these networks tend to rely on the most recent tokens. Instead of using RNN, authors used LSTM. The long and short term memory capability of these networks helps them to keep in memory what is important and forget what is not necessary to remember. Equation 5 formulates forget-gate of an LSTM neural network, eq. 6 shows input-gate, eq. 7 notes output-gate and eq. 8 presents memory-cell. Finally, eq. 9 shows the hidden part of an LSTM unit [32, 33].

$$lf_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f), \tag{5}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i), \tag{6}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o), \tag{7}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c), \tag{8}$$

$$h_t = o_t \circ \sigma_h(c_t). \tag{9}$$

For all these equations, σ is activation function (*sigmoid* or *tanh* are commonly used for LSTM) and $\circ$ is concatenation operation. W and U are weights and b is the bias which should be learned over training process.

LSTM is useful for capturing the relation of tokens in a forward sequential form; However, in natural language processing tasks, it is required to know the upcoming token. To overcome this problem, the authors have used a backward and forward LSTM combining the output of both.

In a different approach, character embedding followed by a convolution layer is proposed in [34] for sequence labeling. The utilized architecture is followed by a bidirectional LSTM layer that ends in a CRF layer. Character embedding is a useful technique that the authors tried to use it in a combination with word embedding. Character embedding with the use of convolution as feature extractor from character level, captures relations between characters that form a word and reduces spelling noise. It also helps the model to have an embedding when pretrained word embedding is empty or initialized as random for new words. These words are encountered when they were not present in the training set. Thus, in the test phase, the model fails to provide a useful embedding.

The NLP revolution of "Attention is all you need" was a game changer that eliminated need for any LSTM like sequential methods and replaced it with the scaled dot-product attention and positional encoding in **Transformer**

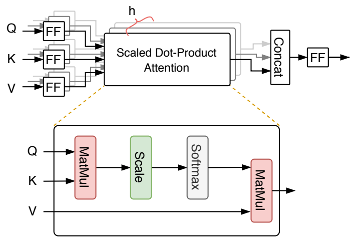


Fig. 2: Scaled dot-product attention mechanism.

stacks [19]. After this new research, many of researchers for various NLP tasks have used the Transformer paradigm; BERT, XLNet, ALBERT and T5 are examples of this new architecture [20, 35–37], however, there are many other related works too [38].

The foundation of these methods starts from tokenization and end at training on very huge data with a huge processing power. The tokenization part is done with Byte Pair Encoding (BPE) generally. The idea of utilizing BPE is novel itself in generating tokens even if it was proposed years ago for text compression [39]. The motivation behind using BPE is having better subword parts instead of words or characters [40]. Figure 2 shows the scaled dot-product attention and the multihead attention mechanism [19].

The attention mechanism has many forms and related studies in fields of machine translation reviewed its effects on the translation task [41]. The Transformer architecture proposed in [19] makes use of scaled dot-product attention that is computed using three vectors of Query, Key and Value (Q,K,V). Equation 10 shows this attention form.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{10}$$

The denominator part of this equation, $\sqrt{d_k}$ is the scale part, proposed in the original article based on the embedding size. The rest of the equation is identical to the Figure 2. Attention head on the other hand, is where scaled dot-product attention units are used in a multi-way, but before using this attention type, a feed-forward (FF as shown in the figure) is applied to each input. A Transformer, is simply a combination of multi-head attention units and feed-forward neural networks. Stacks of Transformer units in encoder and decoder parts make a Transformer based architecture. However, for many tasks, this architecture is useful. In the case of our study, a typical named entity recognizer architecture based on the Transformer is shown in Figure 3 [42]. The output embeddings of the last decoder or encoder part is used for generating final NER tags.

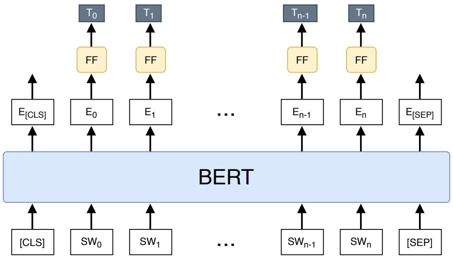


Fig. 3: Transformer based NER proposed in [42].

2.2 Multimodal Named Entity Recognition

Multimodal learning has become an emerging research interest and with the rise of deep learning techniques, it has become more visible in different research areas ranging from medical imaging to image segmentation and natural language processing [43–61]. On the other hand, very little research has been focused on the extraction of named entities with joint image and textual data concerning short and noisy content [12, 52, 62, 63] while several studies have been explored in textual NER using neural models [28, 34, 64–69].

State-of-the-art methods have shown acceptable evaluation on structured and well formatted short texts. Techniques based on deep learning such as utilization of convolutional neural networks [65, 69], recurrent neural networks [66] and long short term memory neural networks [28, 34] are aimed to solve NER problem.

The multimodal named entity recognizers can be categorized in two categories based on the tasks at hand, one tries to improve NER task with the utilization of visual data [12, 62, 63], and the other tries to give further information about the task at hand such as disambiguation of named entities [52]. We refer to both of these tasks as MNER[2]. To have a better understanding of MNER, equation 11 formulates the available multimodal data while equations 2 and 3 are true for this task.

$$s' = \langle i, w_1, w_2, \ldots, w_n \rangle \tag{11}$$

i refers to image and the rest goes same as equation 1 for word token sequence.

In [63], pioneering research was conducted using feature extraction from both image and textual data. The extracted features were fed to decision trees in order to output the named entity classes. Researchers have used multiple datasets ranging from buildings to human face images to train their image feature extractor (object detector and k-means clustering) and a text classifier has been trained on texts acquired from *DBPedia*.

Researchers in [62] proposed a MNER model with regards to triplet embedding of words, characters and image. Modality attention applied to this triplet

[2] Multimodal Named Entity Recognizer

indicates the importance of each embedding and their impact on the output while reducing the impact of irrelevant modals. Modality attention layer is applied to all embedding vectors for each modal, however the investigation of fine-grained attention mechanism is still unclear [70]. The proposed method with Inception feature extraction [18] and pretrained *GloVe* word vectors shows acceptable results on the dataset that the authors aggregated from Snapchat[3]. This method shows around 0.5 for precision and F-measure for four entity types (person, location, organization and misc) while for segmentation tasks (distinguishing between a named entity and a non-named entity) it shows around 0.7 for the metrics mentioned.

An adaptive co-attention neural network with four generations are proposed in [12]. The adaptive co-attention part is similar to the multimodal attention proposed in [62] that enabled the authors to have better results over the dataset they collected from Twitter. In their main proposal, convolutional layers are used for word representation, BiLSTM is utilized to combine word and character embeddings and an attention layer combines the best of the triplet (word, character and image features). VGG-Net16 [71] is used as a feature extractor for the image while the impact of other deep image feature extractors on the proposed solution is unclear, however the results show its superiority over related unimodal methods.

3 The Proposed Approach

In the present work, we propose two approaches for the NER problem. First we propose the CWI in Section 3.1 and in Section 3.2 we demonstrate our second approach, the multimodal Transformer. CWI is based on character-word-image features extracted using a deep neural network and the multimodal Transformer is utilizing a Transformer combined with image features. Both of these two use the same set of inputs, sentence, and related image from social media posts. For the Transformer approach, we use a BERT-like Transformer that gets the image features extracted using InceptionV3.

3.1 CWI: Character-Word-Image

CWI is able to handle noise by co-learning semantics from three modalities, character, word, and image. This model is composed of three parts, convolutional character embedding, joint word embedding (fastText-GloVe) and InceptionV3 image feature extraction [14, 16, 18]. Each part of this architecture is designed to address specific problems of NER in Twitter. Character feature extraction is designed to overcome the textual noise problem at the token level (mistypes and OoV). Word feature extraction using shallow transfer learning obtained from GloVe and fastText helps to improve semantic understanding of the text in presence of OoV tokens. On the other hand, the image feature

[3] A multimedia messaging application

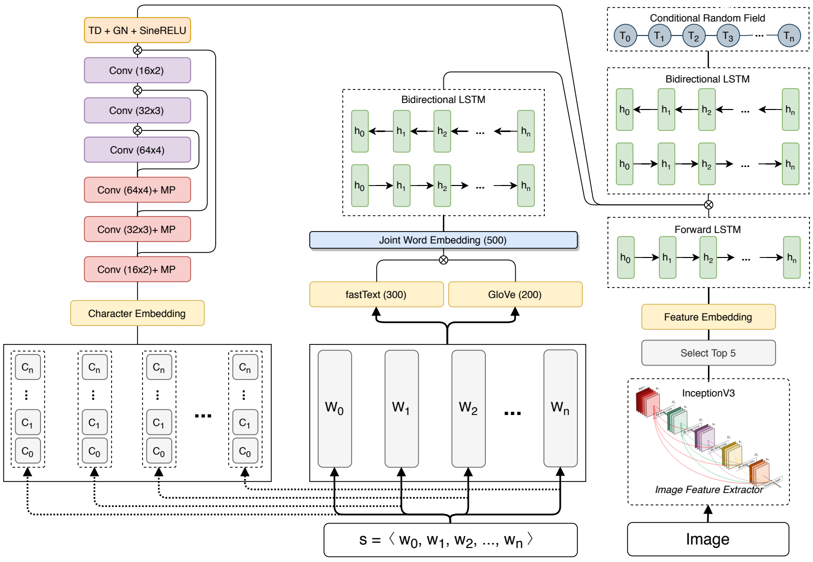


Fig. 4: Proposed CWI Model: Character (left), Word (middle) and Image (right) feature extractors combined by bidirectional long-short term memory and the conditional random field at the end.

extraction guides model where visual content has bias over specific entities such as buildings, humans, and related objects. The concatenated form of these features together form a more robust understanding of Tweet that is separable into token embeddings. Figure 4 shows the *CWI* architecture in more detail.

Character Feature Extraction shown in the left part of Figure 4 is a composition of six layers. Each sequence of words from a single tweet, $\langle w_1, w_2, \ldots, w_n \rangle$ is converted to a sequence of character representation defined by $\langle [c_{(0,0)}, c_{(0,1)}, \ldots, c_{(0,k)}], \ldots, [c_{(n,0)}, c_{(n,1)}, \ldots, c_{(n,k)}] \rangle$ and in order to apply one dimensional convolution, it is required to be in a fixed length. The parameter k shows the fixed length of the character sequence representing each word. Rather than using the one-hot representation of characters, a randomly initialized (uniform distribution) embedding layer is used. The first three convolution layers are followed by a one dimensional pooling layer. In each layer, kernel size is increased incrementally from 2 to 4 while the number of kernels are doubled starting from 16. Just like the first part, the second segment of this feature extractor uses three layers but with slight changes. Kernel size is reduced starting from 4 to 2 and the number of kernels is halved starting from 64. In this part, $\otimes$ sign shows concatenation operation. *TD + GN + SineRelu* note targeted dropout, group normalization and SineRelu [72–74]. These layers prevent the character feature extractor from overfitting. Equation 12 defines SineRelu activation function which is slightly different from Relu.

$$SineRelu(x) = \begin{cases} x & x > 0 \\ \epsilon(\sin x - \cos x) & x \leq 0 \end{cases} \tag{12}$$

Instead of using zero in the second part of this equation, $\epsilon(\sin x - \cos x)$ has been used for negative inputs, ϵ is a hyperparameter that controls the amplitude of $\sin x - \cos x$ wave. This slight change prevents the network from having dead-neurons and unlike Relu, it is differentiable everywhere. On the other hand, it has been proven that using GroupNormalization provides better results than BatchNormalization on various tasks [73].

However, the dropout has a major improvement on the neural network as an overfitting prevention technique [75], in our setup the TargtedDropout shows to provide better results. TargetedDropout randomly drops neurons whose output is over a threshold. On the other hand, skip connections presented in the model, provide better learning in the character feature extraction part and enables the model to learn in a better way in terms of evaluation metrics.

Word Feature Extraction is presented in the middle part of Figure 4. Joint embeddings from pretrained word vectors of GloVe[4] [14] and fastText[5] [16] by concatenation operation results in 500 dimensional word embedding. In order to have forward and backward information for each hidden layer, we used a bidirectional long-short term memory [32, 33]. For the words which were not in the pretrained tokens, we used a random initialization (uniform initialization) between -0.25 and 0.25 at each embedding. The result of this phase is extracted features for each word. fastText provides better embeddings when GloVe fails, and the reason behind it is the structure of fastText itself which is able to capture morphological semantics using subword embeddings.

Image Feature Extraction is shown in the right part of Figure 4. For this part, we use InceptionV3[6] pretrained on ImageNet [76]. Many models were available as the first part of image feature extraction, however the main reason we used InceptionV3 as feature extractor backbone is its better performance on ImageNet and the results obtained by this particular model were slightly better compared to others.

Instead of using the headless version of InceptionV3 for image feature extraction, we have used the full model which outputs the 1000 classes of ImageNet. Each of these classes resembles an item, the set of these items can present a person, location or anything that is identified as a whole. To have better features extracted from the image, we use an embedding layer. In other words, we looked at the top 5 extracted probabilities as words that is shown

[4] 6 billion tokens with 200 dimensional word vectors, available at: `http://nlp.stanford.edu/data/glove.6B.zip`

[5] 16 billion tokens with 300 dimensional word vectors, available at: `https://dl.fbaipublicfiles.com/fastText/vectors-english/wiki-news-300d-1M.vec.zip`

[6] InceptionV3 pretrained model on ImageNet, available at: `https://keras.io/applications/#inceptionv3`

in eq. 13; based on our assumption, these five words present textual keywords related to the image and combination of these words should provide useful information about the objects in visual data. An LSTM unit has been used to output the final image features. These combined embeddings from the most probable items in the image are essential to have extra information from a social media post.

$$IW = \underset{x}{\arg\operatorname{sort}}\{x|x = \text{Inception}(i)\}[1:5],\ x \in [0,1] \tag{13}$$

where IW is image-word vector, x is output of InceptionV3 and i is the image. x is in the range of [0,1] and $\sum\limits_{\forall k \in x} k = 1$ holds true, while $\sum\limits_{\forall k \in IW} k \leq 1$.

Multimodal Fusion in our work is presented as the concatenation of three feature sets extracted from words, characters and images. Unlike previous methods, our original work does not include an the attention layer to remove noisy features. Instead, we stacked LSTM units from word and image feature extractors to have better results. The last layer presented at the top right side of Figure 4 shows this part. In our second proposed variation, we apply attention layer to this triplet. Our proposed attention mechanism is able to detect on which modality to increase or decrease focus. Equations 14, 15 and 16 show attention mechanism related to the second variation of first model.

$$lu_{it} = \tanh(W_w h_{it} + b_w) \tag{14}$$

$$\alpha_{it} = \frac{exp(h_t^\top u_{it})}{\sum\limits_t \exp(h_t^\top u_{it})} \tag{15}$$

$$\beta_i = \sum_t \alpha_{it} h_{it} \tag{16}$$

Conditional Random Field is the last layer in our setup which forms the final output. The same implementation explained in eq. 4 is used for our method.

3.2 Multimodal Transformer

Transformer mechanism described in section 2.1 is used here with some modification on the hyper-parameters. Also, we changed the input format of the original BERT model that we describe in the current subsection. We call our modified BERT model as MSB (Multimodal Small BERT). The modified version is smaller than the BERT original model and is the same size as small BERT from original BERT released models. Encoder stack of the original Transformer model is also used here. BPE makes sure that the model does not encounter OoV tokens in any form of noise. This tokenization method uses byte pairs instead of white-space tokenization of the regular models.

Table 2: Transformer configuration for NER task: Tiny and Small versions.

Model	Hidden Size	# of Attention Heads	# of Transformer Layers
MSB-Tiny	128	2	2
MSB-Small	512	8	4

Byte Pair Encoding: For the tokenization part, we use BPE tokenizer [40]. The pretrained subotkens are released in [20][7]. Before tokenization, we used preprocessing operations such as URL removal. Removing URLs helps the model to skip the unnecessary operations on the input. The rest of text is given to the model with no changes. However, we further pretrained BERT to fit our task at hand, on the related corpus such as crawled Twitter corpus; on the tokenizer part, we used same as the released version in the original format.

Transformer Configuration: We used Transformers as our building block with getting motivation from BERT as our base and reduced the parameters using the main BERT-Tiny and BERT-Small configurations. The configurations we used are presented in Table 2. For both of these configurations, the vocabulary size is 30522. Pretrained version are released by google [8,9]. Figure 5 shows our approach and the utilization of image extracted features into BERT model. The `[SEP]` token has been used to separate the text and the outputs of image feature extractor (the labels). These two modalities in uniform structure are given to the Transformer to extract the final named entities. Another variation of the model is also introduced that has an extra CRF layer. The conditional random field helps the model to correct the mistakes by equation 4.

Combination of BPE and the Transformer gains much improvement in terms of evaluation metrics because it solves the OoV problem and uses a real bidirectional form of NLU instead of left-to-right or right-to-left. This uniform understanding of the splitted tokens with aid of more pretraining on the social media posts and other related details are provided in the next section.

4 Experimental Evaluation

The present section provides evaluation results of our model against baselines. Before diving into our results, a brief description of the dataset and its statistics are provided in Section 4.1. Experimental setups including model hyperparameter details are detailed in Section 4.2. Various aspects of proposed models are compared with regards to their hyperparameters in Section 4.3. Textual noise is also another major part of the experimental evaluation that is tested in different forms, Section 4.4 presents these analyses. These noise tests, regardless of the Twitter dataset itself which is a noisy dataset, is performed by

[7] `https://github.com/google-research/bert`

[8] BERT-Tiny: `https://storage.googleapis.com/bert_models/2020_02_20/uncased_L-2_H-128_A-2.zip`

[9] BERT-Small: `https://storage.googleapis.com/bert_models/2020_02_20/uncased_L-4_H-512_A-8.zip`

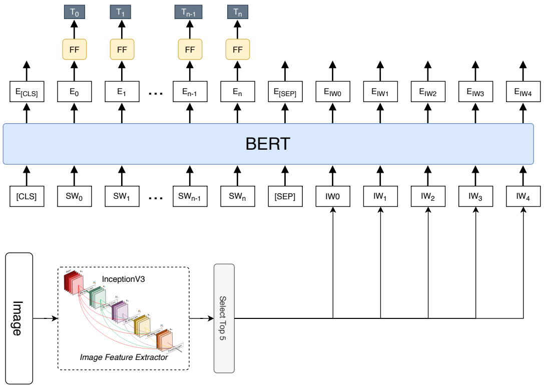


Fig. 5: Our proposed second approach: **M**ultimodal **S**mall **BE**RT.

random noise additions, and a comparison between the original model and the other state-of-the-art models is provided. We also provide a subsection (4.5) for evaluating the effect of different feature extractors. The effect of GloVe and fastText is also another major evaluation of the Section 4.6. This shallow transfer learning evaluation is provided to show how these two embeddings affect the output.

4.1 Dataset

In [12], a refined collection of tweets gathered from Twitter is presented. Their dataset, which is labeled for the NER task contains 8,257 tweets. There are 12,784 entities in total in this dataset. Table 3 shows statistics related to each named entity in the training, development, and test sets. Following CoNLL-2033 and the BIO2 tagging, this dataset is also tagged manually by experts. Short tweets that contain less than three words have been discarded by the annotators. Non-English tweets are also discarded. The overall dataset from 26.5 million tweets has been reduced to total of 8,257 tweets from 12,784 users. The training, development, and testing set is also split into 4,000, 1,000, and 3,257 tweets, respectively. All tweets contain images related to them. These images are posted by users and related samples from the dataset are presented in Figure 6.

4.2 Experimental Setup

In order to obtain the best results in Section 4.3, for our first model (CWI), we use the following setup in tables 4, 5, 6 and 7. For the second proposed method, the same parameter settings have been used with an additional attention layer. This additional layer has been added after layer 31 in Table 7 and before the final CRF layer, indexed as 32. *Adam* optimizer with 8×10^{-5} learning rate

Table 3: Statistics of named entity types in train, development and test sets [12].

Entity Type	Train	Dev.	Test	Total
Person	2217	552	1816	4583
Location	2091	522	1697	4308
Organization	928	247	839	2012
Miscellaneous	940	225	726	1881
Total Entities	6176	1546	5078	12784

Table 4: Implementation details of our model (CWI): Character Feature Extractor.
Con.: Connection; KS: Kernel Size; PS: Pooling Size; DR: Dropout Rate; TR: Target Rate; ↑: prior layer
⋆ MaxPooling has been applied to second dimension rather than channels

ID	Layer Name	Con.	Details
1	Input	–	35 × 40
2	Embedding	↑	Embedding vector size is set to 40 and initialized in range of [-0.25, 0.25] with uniform distribution
3	1D conv.	↑	KS: 2, # of Kernels: 16
4	1D MaxPooling	↑	PS: 2
5	1D conv.	↑	KS: 3, # of Kernels: 32
6	1D MaxPooling	↑	PS: 2
7	1D conv.	↑	KS: 4, # of Kernels: 64
8	1D MaxPooling	↑	PS: 2
9	1D conv.	↑	KS: 4, # of Kernels: 64
10	Concatenation	8,9	–
11	1D conv.	↑	KS: 3, # of Kernels: 32
12	Concatenation	6,11	–
13	1D conv.	↑	KS: 2, # of Kernels: 16
14	Concatenation	4,13	–
15	Targeted Dropout	↑	DR: 0.25, TR: 0.4
16	Sine Relu	↑	ϵ: 0.0025
17	Group Normalization	↑	Applied to 16 groups

is used in training phase with 10 epochs. The MSB model is also pretrained on Twitter data by using the Twitter API and gathered texts. This model has another variation that utilizes the CRF at the last layer for better performance.

For the MSB-Tiny and Small version, we used pretrained weights from BERT that google released. We also trained the model on two different datasets, Twitter-Multimodal-NER dataset (TMN) [12] and CoNLL-2003 [77]. The language model that has been used is also trained on the Twitter text data that gains more realistic texts to add to the modeling in the pretraining phase. The fine-tuning part has been done in two phases, we first fine-tuned masked language modeling on CoNLL-2003 NER dataset and in the second phase, we trained the whole model on NER task TMN dataset.

Figure 6 shows some visual samples of the dataset. Also, we present the result of our different approaches on these samples in fig. 7. In this figure, the Ground-Truth is highlighted with red color at the above line of each sentence and the results of our approaches are shown by different colors at the below

Table 5: Implementation details of our model (CWI): Word Feature Extractor.

ID	Layer Name	Con.	Details
18	Input	–	35
19	GloVe	18	GloVe Embedding vector, vector size: 200
20	fastText	18	fastText Embedding vector, vector size: 300
21	Concatenation	19,20	–
22	LSTM (Forward)	21	Size: 100
23	LSTM (Backward)	21	Size: 100
24	Concatenation	22,23	–

Table 6: Implementation details of our model (CWI): Image Feature Extractor.

ID	Layer Name	Con.	Details
25	Input	–	5 highest probability classes selected from InceptionV3
26	Embedding	↑	50
27	LSTM (Forward)	↑	Size: 50

Table 7: Implementation details of our model (CWI): Multimodal Fusion.

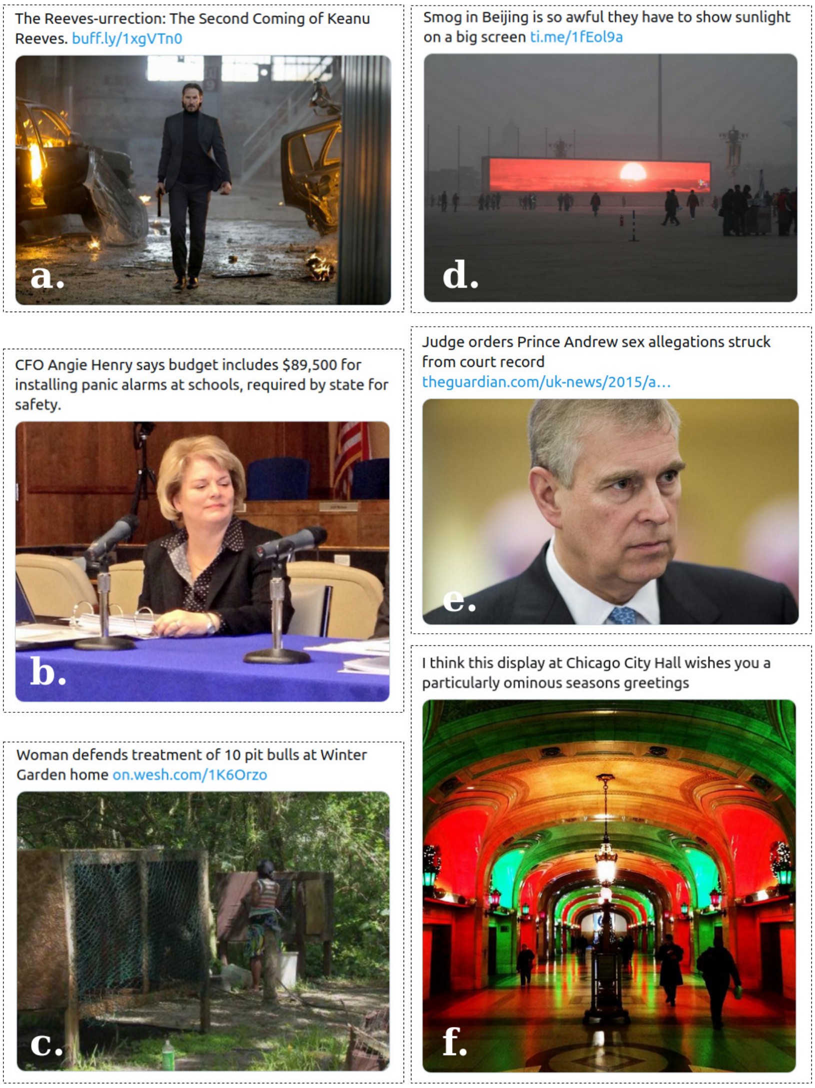
ID	Layer Name	Con.	Details
28	Concatenation	17,24,27	–
29	LSTM (Forward)	28	Size: 100
30	LSTM (Backward)	28	Size: 100
31	Concatenation	29,30	–
32	CRF	30	# of output Classes: 9, according to BIO2 1

lines of each sentence. Some samples such as the first one are not correctly labeled in the dataset, but our approach appropriately predicts the true labels.

4.3 Evaluation Results

Table 8 presents the evaluation results of our proposed models. Different variations of our proposed models are tested and reported in this table. CWI which is the original model with no attention mechanism shows 2% improvement on the Person class while it is also 1% ahead on the miscellaneous class. Overall, this model is 1% better than other state-of-the-art models on the F1 score. However, the different variations of this model report near the same results on the dataset, Transformer based variation of our proposed model shows 3% improvement compared to others. But the Transformer based model has more trainable parameters and also it uses transfer learning gained from autoencoder Transformer architecture.

The effect of TD+GN+SineRelu (Targeted dropout, Group Normalization, and Sine Relu) on the CWI model is investigated in Table 9. The phrase "No" in this table indicates that none of these are used and instead for SineRelu



The Reeves-urrection: The Second Coming of Keanu Reeves. buff.ly/1xgVTn0

CFO Angie Henry says budget includes $89,500 for installing panic alarms at schools, required by state for safety.

Woman defends treatment of 10 pit bulls at Winter Garden home on.wesh.com/1K6Orzo

Smog in Beijing is so awful they have to show sunlight on a big screen ti.me/1fFEol9a

Judge orders Prince Andrew sex allegations struck from court record theguardian.com/uk-news/2015/a…

I think this display at Chicago City Hall wishes you a particularly ominous seasons greetings

Fig. 6: Samples from Twitter dataset with text and related image [12].

normal Relu activation function is replaced. The utilization of these three (TD+GN+SineRelu) makes a huge impact on CWI and prevents it from overfitting. The overall improvement on F1 score is 7%.

The training time for the first model in a CoreI7 processor with Nvidia GeForce GTX 1650 is around 10 minutes while for the second approach it is 2 days for pretraining on Twitter data and fine tuning on the datasets. The training time for the InceptionV3 is not considered because we used the original released version with no changes.

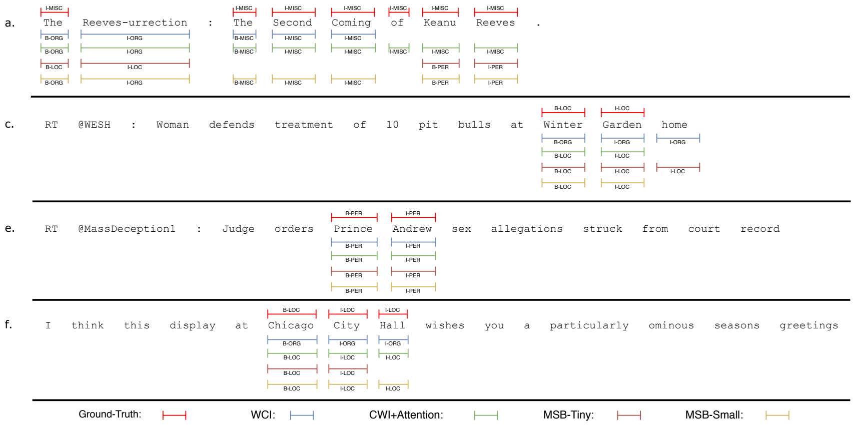

a.
I-MISC I-MISC I-MISC I-MISC I-MISC I-MISC I-MISC
The Reeves-urrection : The Second Coming of Keanu Reeves .
B-ORG I-ORG B-MISC I-MISC I-MISC
B-ORG I-ORG B-MISC I-MISC I-MISC I-MISC I-MISC I-MISC
B-LOC I-LOC B-PER I-PER
B-ORG I-ORG B-MISC I-MISC I-MISC B-PER I-PER
c.
B-LOC I-LOC
RT @WESH : Woman defends treatment of 10 pit bulls at Winter Garden home
B-ORG I-ORG I-ORG
B-LOC I-LOC
B-LOC I-LOC I-LOC
B-LOC I-LOC
e.
B-PER I-PER
RT @MassDeception1 : Judge orders Prince Andrew sex allegations struck from court record
B-PER I-PER
B-PER I-PER
B-PER I-PER
B-PER I-PER
f.
B-LOC I-LOC I-LOC
I think this display at Chicago City Hall wishes you a particularly ominous seasons greetings
B-ORG I-ORG I-ORG
B-LOC I-LOC I-LOC
B-LOC I-LOC
B-LOC I-LOC I-LOC
Ground-Truth:
WCI:
CWI+Attention:
MSB-Tiny:
MSB-Small:

Fig. 7: Results of our approaches on different samples from fig. 6, CRF variation of models has been used here.

Table 8: Evaluation results of different approaches compared to ours.

Method	Per.	Loc.	Org.	Misc.	Overall		
					Prec.	Recall	F1
Stanford NER [30]	73.85	69.35	41.81	21.80	60.98	62.00	61.48
BiLSTM+CRF [28]	76.77	72.56	41.33	26.80	68.14	61.09	64.42
LSTM+CNN+CRF [34]	80.86	75.39	47.77	32.61	66.24	68.09	67.15
T-NER [13]	83.64	76.18	50.26	34.56	69.54	68.65	69.09
BiLSTM+CNN+Co-Attention [12]	81.89	**78.95**	**53.07**	34.02	72.75	68.74	70.69
CWI (Ours)	85.81	76.68	50.18	35.65	73.64	69.68	71.61
CWI + Attention (Ours)	84.02	77.34	52.60	33.47	72.37	70.05	71.19
MSB-Tiny (Ours)	82.17	76.47	51.09	34.31	71.08	69.75	70.41
MSB-Small (Ours)	86.32	74.36	50.73	35.12	72.89	70.10	72.74
MSB-Tiny + CRF (Ours)	84.21	75.16	52.89	35.31	72.87	69.41	71.10
MSB-Small + CRF(Ours)	**86.44**	77.16	52.91	**36.05**	**74.97**	**72.04**	**73.47**

Table 9: Effect of *TD+GN+SineRelu* on our proposed model.

TD+GN+SineRelu	Overall	Per.	Loc.	Org.	Misc.
No	64.18	76.21	72.30	40.98	28.81
Yes	**71.61**	**85.81**	**76.68**	**50.18**	**35.65**

530 4.4 Noise Effect

The effect of noise on the dataset itself is undeniable because the whole dataset is gathered from Twitter with all noisy text attributes. To have a more sophisticated test on the model robustness on the textual noise we also prepared a test case by altering the actual Twitter dataset. For this test, we analyzed common mistakes and human typos by virtual and hardware keyboards in the category of unintentional errors. We also used internet slang to replace words that have slang in the dataset. We separated this test into two phases, full noisy text, and entity noise version. The full noise version alters any token while the entity noise just alters the tokens that are tagged as entities. The noise rate in character level for this results is set to 0.05 for both of the tests which means with a random chance of 0.05 each character is altered in a noisy and false form. For the slang replacement, we replaced all possible words with their respective slang. Table 10 presents the results of the test.

Table 10: Evaluation results of Noise test, reported results are overall F1 scores.

Method	Full Noise	Entity Noise
Stanford NER [30]	39.27	46.71
BiLSTM+CRF [28]	45.18	48.91
LSTM+CNN+CRF [34]	50.07	49.88
T-NER [13]	59.41	60.01
BiLSTM+CNN+Co-Attention [12]	60.23	59.64
CWI (Ours)	63.14	65.10
CWI + Attention (Ours)	64.97	65.71
MSB-Tiny (Ours)	64.26	64.09
MSB-Small (Ours)	65.99	63.87
MSB-Tiny + CRF (Ours)	65.67	66.06
MSB-Small + CRF(Ours)	66.28	66.54

4.5 Feature Extractor Effect

Each feature extractor in CWI model has its impact on the final results. In order to have a clear understanding and result from analysis on the impact of each feature extractor, we conducted a test with the base CWI model using different feature extractor combinations. For this test, we used **only character**, **only word**, **character and word**, **character and image**, and **word and image** variations of the base model. These variations are trained with the same setup explained in Section 4.2. Table 11 shows the results obtained for each of the variations with respect to the original dataset and the altered noisy version. For the second model, the impact is shown by just dropping visual content from the model.

From Table 11, it is seen that character and word features together form a more robust model in presence of noise. However, the image feature extractor also provides an additional improvement in terms of the F1 score. It is also clear from this table that MSB-Tiny and Small versions remain robust to noise in absence of image features. Comparing the results from this table to the ones in Table 10 shows that image feature extraction is extra information fore model to help improve its results in both CWI and MSB models.

Table 11: Evaluation results of different feature extractor variations using original dataset, full noise and entity noise versions; reported results are overall F1 scores.

Method	Orig. Dataset	Full Noise	Entity Noise
Only Character features	48.98	44.31	46.33
Only Word features	60.14	52.10	52.40
Character and Word features	66.71	63.91	64.02
Character and Image features	52.32	47.33	47.62
Word and Image features	64.45	51.12	50.10
MSB-Tiny + CRF (no Image)	64.41	60.01	59.16
MSB-Small + CRF (no Image)	65.21	60.87	61.70

4.6 Shallow Transfer Learning Effect

Another aspect of the proposed CWI model is its GloVe and fastText embedding usage. These two different forms of embeddings provide different vectors for each token in the word feature extractor part. Utilizing both of these embeddings provides a trainable and pretrained word embedding (GloVe) and a non-trainable but transferred one (fastText). The model architecture shown in Figure 4 presents a two-part embedding in the middle part (word feature extraction), namely fastText and GloVe. The GloVe part is embedded inside the model to be trainable parameters of the model but the fastText part is used to be a frozen part of the model (non-trainable parameters). The reason to use both not just one is that the fastText can give an embedding vector for any given word even if it is not seen before. GloVe vectors, on the other hand, are embedded inside the model to be part of the model trainable parameters. In order to show that this combination using both of them is useful, we conducted an experiment. In this experiment, we used different variations of the CWI model with only focusing on the embedding part of the word feature extractor. Two different models, one using only fastText, and the other only using GloVe are trained and tested on the original and noisy dataset. Figure 8 shows the result of different embedding sizes for only using GloVe or fastText. Table 12 shows the best values from this figure which are 300 and 300 for both GloVe and fastText.

Table 12: Evaluation results of word feature extractor variations on GloVe and fastText; reported results are overall F1 scores and embedding size for GloVe and fastText is 300 for both.

Method	Orig. Dataset	Full Noise	Entity Noise
Only fastText	67.30	62.70	63.69
Only GloVe	66.98	60.10	61.02

Third variation uses both of these embeddings but with different embedding sizes for each one. Table 13 shows these results. From this table it is clearly

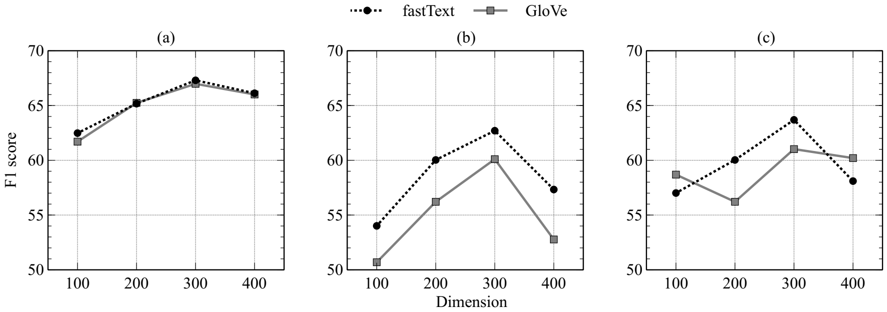


Fig. 8: Dimensionality effect on using only GloVe and fastText; Results are reported for: (a) Original Dataset, (b) Full Noise, and (c) Entity Noise.

Table 13: Evaluation results of different embedding sizes for GloVe and fastText; reported results are overall F1 scores.

Embedding	fastText				
	size	100	200	300	400
GloVe	100	69.49	70.05	68.40	67.33
	200	69.91	70.23	**71.61**	69.02
	300	68.52	69.83	70.98	68.35
	400	68.21	70.21	68.99	68.14

seen that best hyperparameter for embedding sizes is 300 for fastText and 200 for GloVe.

5 Discussion and Future Works

NER from noisy social media content is a mandatory task to be accomplished with acceptable results. The dependency of many other NLP and analytic tasks on NER is an undeniable fact. Regular and conventional methods fail to accomplish this task because they are mainly trained and evaluated on clean and noise-free textual data. However, there are methods that are trying to find acceptable solutions for it but the role of noise in the final results still keeps pushing towards better solutions. A combination of different aspects of social media data such as images is helpful towards better solutions. We utilized different forms of deep feature extractors and combined them using multimodal fusion to address the problem. From the analysis provided in section 4, it is clearly seen that our proposed solution works better in terms of evaluation metrics. The main reason that it works lies in the efficient utilization of different feature extractors.

The character feature extractor provides character level understanding and helps the final model to have an understanding of the character sequence of tokens while the image features are guiding the model to have a bias over entities extracted from visual content. Word feature extractor, on the other hand, uses both trainable and non-trainable parameters from two pretrained word vectors. In cases where a single one of these features is used the results are not acceptable and this fact shows the harmony behind the three of these feature extractors is a promising solution for the noisy social media entity extraction. However, investigation of newer architectures such as Transformers is another good solution for the problem which we did in our second proposed model.

This new era of social media on the Internet which users are the main contributors to the new and democratic internet requires modern NLP solutions. If a good understanding of the user generated content is required, then the models must be able to battle noise. Users on the internet are not professional linguists and their posts on social media are not edited by professionals. Recent advancements on multimodal models that can see posts from all sides seem a promising solution for harvesting information from unstructured content. However, there are still different aspects of multimodal data that are required to be analyzed; for example in the case of Twitter, users post video and audio with their text too. These modalities that are not explored in this article can be investigated and analyzed to make better models.

6 Conclusion

In this article, we proposed two NER approaches based on multimodal deep learning. In our first model, we used a new architecture in character feature extraction that has helped our model to handle the issue of noise. We used different features from character, word and image representation of the tweet. We also conducted different experiments to show the importance and effect of each feature extractor. Rather than CWI model, we also used Transformers as our building block to propose MSB. Instead of using direct image features from near last layers of image feature extractors such as Inception, we used the direct output of the last layer. The last layer is 1000 classes of diverse objects that are the result of InceptionV3 trained on ImageNet dataset. We used the top 5 classes out of these and converted them to one-hot vectors. The resulting image feature embedding out of these high probability one-hot vectors helped our model to overcome the issue of noise in images posted by social media users. Evaluation results of our proposed models compared to other state-of-the-art methods show their superiority to these methods overall while in two categories (Person and Miscellaneous) our model outperformed others.

Acknowledgments

The authors would like to thank Shervin Minaee for reviewing this work, and providing very useful comments to improve this work.

Conflict of Interest

The authors certify that there is no actual or potential conflict of interest in relation to this article.

References

1. Twitter. About Twitter, Inc, 2014. ISSN 01962892.
2. Miles Osborne, Victor Lavrenko, and Sasa Petrovic. Streaming First Story Detection with application to Twitter. *Computational Linguistics*, 2010. ISSN 1095-6859. doi:10.1016/j.ygyno.2008.10.024.
3. Sandeep Panem, Manish Gupta, and Vasudeva Varma. Structured information extraction from natural disaster events on twitter. In *Proceedings of the 5th International Workshop on Web-scale Knowledge Representation Retrieval & Reasoning*, pages 1–8, 2014.
4. Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. Twiner: named entity recognition in targeted twitter stream. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 721–730, 2012.
5. Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 2020.
6. David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.
7. Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, 2011.
8. Tajinder Singh and Madhu Kumari. Role of text pre-processing in twitter sentiment analysis. *Procedia Computer Science*, 89:549–554, 2016.
9. Eleanor Clark and Kenji Araki. Text normalization in social media: progress, problems and applications for a pre-processing system of casual english. *Procedia-Social and Behavioral Sciences*, 27:2–11, 2011.
10. Farzindar Atefeh and Wael Khreich. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164, 2015.
11. Firoj Alam, Ferda Ofli, and Muhammad Imran. Crisismmd: Multimodal twitter datasets from natural disasters. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12, 2018.
12. Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. Adaptive Co-attention Network for Named Entity Recognition in Tweets. *Aaai*, 2018. ISSN 0028-0836. doi:10.1001/jamapsychiatry.2014.1105.
13. Alan Ritter, Sam Clark, Mausam Etzioni, and Oren Etzioni. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing(EMNLP'11)*, 2011. ISBN 978-1-937284-11-4. doi:10.1075/li.30.1.03nad.

14. Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. ISBN 9781937284961. doi:10.3115/v1/D14-1162.
15. Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.
16. Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X.
17. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
18. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
19. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.
20. 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.
21. Rahul Sharnagat. Named Entity Recognition Literature Survey. In *11305R013*, 2014.
22. C. Li, A. Sun, J. Weng, and Q. He. Tweet segmentation and its application to named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):558–570, FEBRUARY 2015. ISSN 1041-4347. doi:10.1109/TKDE.2014.2327042.
23. Erik F Sang and Jorn Veenstra. Representing text chunks. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 173–179. Association for Computational Linguistics, 1999.
24. K. Li, W. Ai, Z. Tang, F. Zhang, L. Jiang, K. Li, and K. Hwang. Hadoop recognition of biomedical named entity using conditional random fields. *IEEE Transactions on Parallel and Distributed Systems*, 26(11):3040–3051, Nov 2015. ISSN 1045-9219. doi:10.1109/TPDS.2014.2368568.
25. C. Wei, R. Leaman, and Z. Lu. Simconcept: A hybrid approach for simplifying composite named entities in biomedical text. *IEEE Journal of Biomedical and Health Informatics*, 19(4):1385–1391, July 2015. ISSN 2168-2194. doi:10.1109/JBHI.2015.2422651.
26. Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *arXiv preprint arXiv:1812.09449*, 2018.

27. Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014. ISBN 9781941643006. doi:10.3115/v1/P14-5010.
28. Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.
29. Tomasz Stanislawek, Anna Wróblewska, Alicja Wójcika, Daniel Ziembicki, and Przemyslaw Biecek. Named entity recognition–is there a glass ceiling? *arXiv preprint arXiv:1910.02403*, 2019.
30. Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, 2005. ISBN 3-540-63438-X. doi:10.3115/1219840.1219885.
31. Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Proceedings of EMNLP/VLC-99*, 1999. doi:10.1.1.114.3629.
32. Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.
33. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
34. Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bidirectional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.
35. Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019.
36. Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.
37. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.
38. Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning based text classification: A comprehensive review. *arXiv preprint arXiv:2004.03705*, 2020.
39. Yusuxke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. Byte pair encoding: A text compression scheme that accelerates pattern matching. Technical report, Technical Report DOI-TR-161, Department of Informatics, Kyushu University, 1999.
40. Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

41. Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.
42. Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. Tuning multilingual transformers for named entity recognition on slavic languages. *BSNLP'2019*, page 89, 2019.
43. E. A. Bernal, X. Yang, Q. Li, J. Kumar, S. Madhvanath, P. Ramesh, and R. Bala. Deep temporal multimodal fusion for medical procedure monitoring using wearable sensors. *IEEE Transactions on Multimedia*, 20 (1):107–118, Jan 2018. ISSN 1520-9210. doi:10.1109/TMM.2017.2726187.
44. D. Wang, P. Cui, M. Ou, and W. Zhu. Learning compact hash codes for multimodal representations using orthogonal deep structure. *IEEE Transactions on Multimedia*, 17(9):1404–1416, Sep. 2015. ISSN 1520-9210. doi:10.1109/TMM.2015.2455415.
45. C. Ding and D. Tao. Robust face recognition via multimodal deep face representation. *IEEE Transactions on Multimedia*, 17(11):2049–2058, Nov 2015. ISSN 1520-9210. doi:10.1109/TMM.2015.2477042.
46. F. Chen, R. Ji, J. Su, D. Cao, and Y. Gao. Predicting microblog sentiments via weakly supervised multimodal deep learning. *IEEE Transactions on Multimedia*, 20(4):997–1007, April 2018. ISSN 1520-9210. doi:10.1109/TMM.2017.2757769.
47. H. Li, J. Sun, Z. Xu, and L. Chen. Multimodal 2d+3d facial expression recognition with deep fusion convolutional neural network. *IEEE Transactions on Multimedia*, 19(12):2816–2831, Dec 2017. ISSN 1520-9210. doi:10.1109/TMM.2017.2713408.
48. L. Pang, S. Zhu, and C. Ngo. Deep multimodal learning for affective analysis and retrieval. *IEEE Transactions on Multimedia*, 17(11):2008–2020, Nov 2015. ISSN 1520-9210. doi:10.1109/TMM.2015.2482228.
49. Y. Jiang, Z. Wu, J. Tang, Z. Li, X. Xue, and S. Chang. Modeling multimodal clues in a hybrid deep learning framework for video classification. *IEEE Transactions on Multimedia*, 20(11):3137–3147, Nov 2018. ISSN 1520-9210. doi:10.1109/TMM.2018.2823900.
50. J. Shi, X. Zheng, Y. Li, Q. Zhang, and S. Ying. Multimodal neuroimaging feature learning with multimodal stacked deep polynomial networks for diagnosis of alzheimer's disease. *IEEE Journal of Biomedical and Health Informatics*, 22(1):173–183, Jan 2018. ISSN 2168-2194. doi:10.1109/JBHI.2017.2655720.
51. D. Ramachandram and G. W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6): 96–108, Nov 2017. ISSN 1053-5888. doi:10.1109/MSP.2017.2738401.
52. Seungwhan Moon, Leonardo Neves, and Vitor Carvalho. Multimodal Named Entity Disambiguation for Noisy Social Media Posts. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018. doi:10.3322/caac.21166.
53. Kuan Liu, Yanen Li, Ning Xu, and Prem Natarajan. Learn to combine modalities in multimodal deep learning. *arXiv preprint arXiv:1805.11730*,

2018.

54. Lisa Beinborn, Teresa Botschen, and Iryna Gurevych. Multimodal grounding for language processing. *arXiv preprint arXiv:1806.06371*, 2018.
55. Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.
56. Samira Ebrahimi Kahou, Vincent Michalski, Kishore Konda, Roland Memisevic, and Christopher Pal. Recurrent Neural Networks for Emotion Recognition in Video. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction - ICMI '15*, 2015. ISBN 9781450339124. doi:10.1145/2818346.2830596.
57. Wei Liu, Wei Long Zheng, and Bao Liang Lu. Emotion recognition using multimodal deep learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016. ISBN 9783319466712. doi:10.1007/978-3-319-46672-9_58.
58. Samira Ebrahimi Kahou, Xavier Bouthillier, Pascal Lamblin, Caglar Gulcehre, Vincent Michalski, Kishore Konda, Sébastien Jean, Pierre Froumenty, Yann Dauphin, Nicolas Boulanger-Lewandowski, Raul Chandias Ferrari, Mehdi Mirza, David Warde-Farley, Aaron Courville, Pascal Vincent, Roland Memisevic, Christopher Pal, and Yoshua Bengio. EmoNets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, 2016. ISSN 17838738. doi:10.1007/s12193-015-0195-2.
59. Heung Il Suk, Seong Whan Lee, and Dinggang Shen. Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. *NeuroImage*, 2014. ISSN 10959572. doi:10.1016/j.neuroimage.2014.06.077.
60. Xi Cheng, Li Zhang, and Yefeng Zheng. Deep similarity learning for multimodal medical images. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization*, 2018. ISSN 21681171. doi:10.1080/21681163.2015.1135299.
61. Di Wu, Lionel Pigou, Pieter Jan Kindermans, Nam Do Hoang Le, Ling Shao, Joni Dambre, and Jean Marc Odobez. Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. ISSN 01628828. doi:10.1109/TPAMI.2016.2537340.
62. Seungwhan Moon, Leonardo Neves, and Vitor Carvalho. Multimodal named entity recognition for short social media posts. *arXiv preprint arXiv:1802.07862*, 2018.
63. Diego Esteves, Rafael Peres, Jens Lehmann, and Giulio Napolitano. Named Entity Recognition in Twitter Using Images and Text. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. ISBN 9783319744322. doi:10.1007/978-3-319-74433-9_17.

64. Alexandre Passos, Vineet Kumar, and Andrew McCallum. Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*, 2014.
65. Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.
66. Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.
67. Timothy Baldwin, Marie-Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135, 2015.
68. Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López Monroy, and Thamar Solorio. A multi-task approach for named entity recognition in social media data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 148–153, 2017.
69. Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. *arXiv preprint arXiv:1702.02098*, 2017.
70. Heeyoul Choi, Kyunghyun Cho, and Yoshua Bengio. Fine-grained attention mechanism for neural machine translation. *Neurocomputing*, 2018. ISSN 18728286. doi:10.1016/j.neucom.2018.01.007.
71. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
72. Aidan N. Gomez, Ivan Zhang, Kevin Swersky, Yarin Gal, and Geoffrey E. Hinton. Learning sparse networks using targeted dropout. *ArXiv*, abs/1905.13678, 2019.
73. Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.
74. Wilder Rodrigues. Sinerelu - an alternative to the relu activation function, Jul 2019. URL `https://medium.com/@wilder.rodrigues/sinerelu-an-alternative-to-the-relu-activation-function-e46a6199997d`.
75. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
76. J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.
77. Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.