Named Entity Recognition for Code-Mixed Kannada-English Social Media Data

Poojitha Nandigam, Appidi Abhinav Reddy, Manish Shrivastava
Language Technologies Research Centre (LTRC)
International Institute of Information Technology, Hyderabad, India
poojitha.nandigam@research.iiit.ac.in, abhinav.appidi@research.iiit.ac.in,
m.shrivastava@iiit.ac.in

Abstract

Named Entity Recognition (NER) is a critical task in the field of Natural Language Processing (NLP) and is also a sub-task of Information Extraction. There has been a significant amount of work done in entity extraction and Named Entity Recognition for resource-rich languages. Entity extraction from code-mixed social media data like tweets from twitter complicates the problem due to its unstructured, informal, and incomplete information available in tweets. Here, we present work on NER in Kannada-English code-mixed social media corpus with corresponding named entity tags referring to Organisation (Org), Person (Pers), and Location (Loc). We experimented with machine learning classification models like Conditional Random Fields (CRF), Bi-LSTM, and Bi-LSTM-CRF models on our corpus.

1 Introduction

India has twenty-three significant languages with over seven hundred and twenty dialects. Kannada is one of the four major Dravidian languages and it is one of the top 30 most spoken languages of the world, with its own independent script and over fifty million speakers. The majority of people are multilingual and tend to mix words from different languages in speech and written text. This method of interchanging languages involves complex grammar and is commonly addressed by terms 'Code-switching' and 'Code-mixing' as described by Lipski (1978).

Code-mixing refers to the use of words, phrases, clauses or morphemes from different languages in the same sentence. Code-switching refers to the use of words or phrases or clauses from different languages within the same speech context. We can understand the difference between code-switching and code-mixing from the positions of altered elements. Code-mixing refers to the intra-sentential modification of codes, whereas code-switching refers to the inter-sentential modification of codes.

1.1 Characteristic of Code-Mixed Kannada-English Data

As explained above mixing happens at word level, phrase level, and morphological level too. Following are few more examples :

1. **Word level:** A complete word from English language is taken into Kannada language. This is language mixing occuring at word level. An example: 'Ee thara branch ideya' which means 'Is there a branch like this?'. Here 'branch' is an English word which got assimilated into Kannada.

2. **Phrase level:** This is a completely code-mixed sentence, that follows the structure of Kannada with English words embedded in it. One example is 'Kelsa bittu pitch reporter aagu olle future ide!' which means 'Leave your work and become pitch reporter, you have great future in that!', this follows the structure of Kannada with English words embedded in it. This is a completely code-mixed sentence.

3. **Morphological level:** The words that are borrowed from English language inflect Kannada suffixes that marks case or number. The word 'cinemagalu' in Kannada, the root word 'cinema' is borrowed from English and 'galu' is a Kannada morphene that marks plurality. Similarly 'caru' becomes 'car', this is nativization.

4. **Syntactic level:** All the examples above are instances of intra-sentential mixing. Here we discuss about intra-sentential and inter-sentential mixing. There are occurrences in Kannada-English CM data where inter-sentential mixing takes place. One such example is 'Born and brought up in bengaluru,

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 43 - 49
December 15-18, 2022 ©2022 Association for Computational Linguistics

Yaako nange mysoor thumba ista, mysoor alli kelsa sikdre ready to shift.’

We observe code-switching and code-mixing frequently on social media platforms like Facebook and Twitter. Here, we work only on the code-mixing aspect observed on Twitter data between Kannada and English languages.

Understanding the code-mixed Kannada-English complicates the problem due to its unstructured, informal, and incomplete information available in the data. Following are the challenges associated when dealing with them.

- **Ambiguous words**: Same word can have a different meaning in multiple languages. Like the word ‘Bali’ in English, which is a place in Indonesia, also used in Kannada with different meaning here as ‘Near’.

- **Variable Lexical Representations**: Some users prefer to use their own romanised form of native word. For example ‘hogilla’ is a Kannada word and it can be written as ‘hogila’, ‘hgilla’, ‘hogillla’ etc.

- **Word-level Code-mixing**: This is similar to language mixing at word-level. For example in the word ‘Kanglish’, its a fusion of two words Kannada and English at word level.

- **Reduplication**: This is common in Indian languages. People tend to use a second word which does not have a meaning on its own but with the first word it becomes a multi word expression. For example ‘postu geestu’, ‘desha gesha’, ‘man ban’. The first words in these examples are English which are followed by reduplicated words.

Here are some instances from a corpus of Kannada-English generated from Twitter data and also transliterated in English.

T1 : *“Haha ashtu idea illade gowdru bengaluru north bittu tumukur hogilla”*
Translation: *“Haha without having much idea gowda left bengaluru north and went to tumukur "*

T2: *“Eshwarappa avarey neevu petrol bunk ge hogilla ansuthe. me nimmannu karkondu hogthini"*
Translation: “Eshwarappa, it looks like you did not go to the petrol bunk. I will take you there."

2 Background and Related Work

There has been a plethora of research done on Named Entity Recognition (NER) from the early 2000s (Finkel et al., 2005). However, most of this is in resource-rich languages. The FIRE2 (Forum for Information Retrieval and Extraction) tasks have shed light on NER in Indian languages. Now, code-mixing has found its application in various areas after FIRE2, such as Query Labeling (Bhargava et al., 2015), Sentiment Analysis (Bhargava et al., 2016), Question Classification.

BR and Ramakanth Kumar (2012) has done the work on the Kannada POS tagger with probabilistic classifiers. Similar work has been done by Todi et al. (2018) in the Kannada POS tagger using machine learning models. Amarappa and Sathyanarayana (2013) worked on NER and classification in the Kannada language. Lakshmi and Shambhavi (2017) presented an automatic identification system for code-mixed Kannada-English Social media text. Shalini et al. (2018) worked on sentiment analysis for Code-Mixed Kannada-English Social Media Text. To the best of our knowledge, the corpus we created is the first Kannada-English code-mixed corpus with named entity tags.

3 Corpus and Annotation

This corpus consists of Kannada-English code mixed tweets gathered from twitter. The tweets were collected using twintproject[1]-an opensource twitter intelligence tool. The tweets are from the past 6 years based on various topics such as movies, sports, celebrities, politics, trending hashtags, social events.

We have retrieved a total of over 317,000 tweets using the twintproject, and after extensive cleaning and pre-processing, we were left with 6530 Kannada-English code mix tweets.

The pre-processing consists of the following steps.

- Removing noisy, useless tweets, i.e., tweets containing only URLs and hashtags.

- Tweets which were written in only Kannada, or only English were removed too.

- Tweets which contain linguistic units from both English and Kannada and having a minimum of five words are only considered, this

[1] https://github.com/twintproject/twint

way, we make sure the tweets adhere to the Kannada-English code mix standard.

- Tokenisation of tweets is done using Tweet Tokenizer.

The corpus will be made available for public use as soon as possible. The following explains the mapping of the tokens with their respective tags.

3.1 Annotation: Named Entity Tagging

We used three Named Entities (NE) tags “Person," “Location," and “Organisation" to tag the corpus. Two people manually did the annotations of the data for Named Entity tags. The annotators have a linguistic background, and are proficient in both Kannada and English. Each of three tags (“Person," “Location" and “Organisation") is divided into Bg-tag (Beginner tag) and It-tag (Intermediate tag) according to the BIO standard thus we have a total of six tags and an ‘Other’ tag to indicate it does not belong to any of the six tags. The Bg-tag is used to tag the beginning word of a Named Entity, whereas It-tag is used to tag a Named Entity, which is split into multiple continuous words. It-tag is assigned to the words which follow the words with a Bg-tag. The explanation of six tags is below.

The ‘Pers’ tag refers to the ‘Person’ entity, which is the name of the person, twitter handles and nicknames of people. The ‘Bg-Pers’ tag is given to the beginning word of a person’s name, and the ‘It-Pers’ tag follows ‘Bg-pers’ tag, if the person’s name is split into multiple continuous words.

The ‘Loc’ tag refers to the ‘Location’ entity, which is the name of the place like Bangalore, Hyderabad, India and others. The ‘Bg-Loc’ tag is assigned to the beginning word of the location name, and the ‘It-Loc’ tag follows ‘Bg-Loc’ tag, if the location name is split into multiple continuous words.

The ‘Org’ tag refers to the ‘Organisation’ entity, which is the name of the organization such as BJP, KFI, INC, Facebook, RBI, and others. The ‘Bg-Org’ tag is assigned to the beginning word of the organization name, and the ‘It-Org’ tag follows ‘Bg-Org’ tag, if the organization name is split into multiple continuous words. Following is an example that shows the application of principles described above.

T3 : *“Haha/other ashtu/other idea/other illade/other gowdru/Bg-Per bengaluru/Bg-Loc north/It-Loc bittu/other tumukur/Bg-Loc hogilla/other”*
Translation: “Haha without having much idea gowda left bengaluru north and went to tumukur."

Tag	Token Count	Cohen Kappa
Bg-Loc	1457	0.89
Bg-Org	3178	0.94
Bg-Pers	5899	0.88
It-Loc	188	0.84
It-Org	505	0.89
It-Pers	358	0.82
Total NE tokens	11585	

Table 1: Tags and their Count in Corpus and IAA.

T4 : *“@vs20012000/Bg-Per illa/other hogilla.../other Harish/Bg-Per ex/other Deputy/Bg-Org Mayor/It-Org organised/other volleyball/other tourney/other ge/other swalpa/other kelasa/other madidde...now/other in/other Bombay./Bg-Loc”*
Translation: “No, did not go.. i did a little bit of work for the volleyball tournament organized by Harish, ex-Deputy Mayor. Now i am in Bombay.."

3.2 Inter Annotator Agreement

The annotations of the data for Named Entity tags were manually done by two people with linguistic backgrounds, both proficient in Kannada and English. The quality of the annotation is validated using the inter-annotator agreement (IAA) between two annotation sets of 6,530 tweets and 152,987 tokens using Cohen’s Kappa coefficient (Hallgren, 2012) (refer Table 1 for Score). The agreement is significantly high. The agreement between the ‘Organisation’ tokens is high while that of ‘Location’ and ‘Person’ tokens is comparatively low due to unclear context and presence of an uncommon or confusing person and location names.

4 Corpus Statistics

We have collected more than 317,000 of tweets from Twitter using TwintProject. After extensive cleaning, we were left with 6,530 Kannada-English code mixed tweets, as part of annotation using six named entity tags along with ‘Other’ tag we tagged 152,987 tokens. We made sure that all the words in the corpus are in Roman script. We used hashtags related to politics, sports, social events and recent trends etc., in collecting the corpus.

5 Experiments

We present all experiments using a combination of features and systems. To understand the effect of different parameters and features of the model, we performed several experiments. Experiments were performed using some set of features at once and all at a time simultaneously changing the parameters of the model, like regularization parameters and algorithms of optimization like ‘L2 regularization’, ‘Average Perceptron’and ‘Passive Aggressive’ for CRF, optimization algorithms and loss functions in LSTM. We used five-fold cross-validation for CRF and three-fold for other experiments in order to validate our classification models. We used ‘scikit-learn,’ ‘Tensorflow,’ and ‘Keras’ libraries for the implementation of the above algorithms.

5.1 Conditional Random Field (CRF)

A Conditional Random Field (CRF) is an undirected probabilistic graphical model that is used for modeling sequential data. It is a model for predicting the most likely sequence of labels that correspond to a sequence of inputs. It has applications in POS tagging, NER, among others. It is a supervised learning method and most often used for structured prediction tasks. When it comes to NER, it has been proven to be better than the tree-based models. Whereas a discrete classifier predicts a label for a single sample without considering “neighboring" samples, a linear chain CRF can take context into account and predicts sequences of labels for sequences of input samples, which is popular in natural language processing.

5.2 LSTM

As our corpus is in sequential text data format, we use Bi-LSTM (combination of two LSTMs — where one runs forward, and one runs backward), which works best to tackle the NER problem as the context covers both past and future labels in a sequence because standard LSTM makes use of only past information in a sequence of text and not the future. Plain LSTM cells in a feedforward network which help us in getting better results by capturing the previous context while Bi-LSTMs also consider the opposite direction. Bi-LSTM considers a sequence of both tokens that are before and after a token of interest. Bi-LSTM network creates a context for each token in the text, which depends on both its past and future.

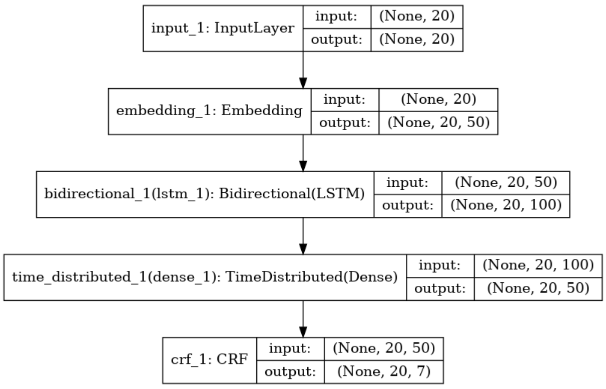

Figure 1: BiLSTM-CRF model architecture

5.3 LSTM-CRF

The Bi-LSTM-CRF is a combination of bidirectional LSTM and CRF (Huang et al., 2015; Lample et al., 2016). The Bi-LSTM model can be combined with CRF to enhance recognition accuracy. This combined model of Bi-LSTM-CRF inherits the ability to learn past and future context features from the Bi-LSTM model and use sentence-level tags to predict possible tags using the CRF layer. We processed the data in batches and used seven epochs.

5.4 Features

The features to our machine learning models consist of characters, lexical and word-level features such as char N-Grams of size 2 and 3 in order to capture the information from suffixes, emojis, mentions in social media like ‘#,’ ‘@,’ punctuation, numbers, numbers in the string. Features from adjacent tokens are used as contextual features.

1. **Character N-Grams:** N-gram is a contiguous sequence of n items from a given sample of text or speech, here the items are characters. Character N-Grams are language-independent (Majumder et al., 2002) and have proven to be efficient in the task of text classification. They are helpful when the text suffers from problems such as misspellings. (Cavnar et al., 1994; Huffman, 1995; Lodhi et al., 2002). Group of chars can help in capturing the semantic information and especially helpful in cases like code mixed language where there is free use of words, which vary significantly from the standard Kannada-English words.

2. **Word N-Grams:** Bag of words has been a staple in NER tasks for languages other than

English (Jahangir et al., 2012). Thus, we use word N-Grams, where we use adjacent words as a feature vector to train our model. These are also called contextual features.

3. **Capitalization:** In social media, people tend to use capital letters to refer to the names of locations, persons, and organizations; at times, they write the entire name in capitals (Von Däniken and Cieliebak, 2017) to give particular importance or to denote aggression. This gives rise to a couple of binary features. One feature is to indicate if the beginning letter of a word is capitalized, and the other is to indicate if the entire word is capitalized.

4. **Mentions and Hashtags:** In social media organizations, like Twitter and Facebook, people use '@' mentions to refer to persons or organizations, they use '#' hashtags in order to make something notable or to make a topic trending. Thus the presence of these two gives a reasonable probability for the word being a named entity.

5. **Numbers in String:** In social media, we see people using alphanumeric characters, generally to save the typing effort, shorten the message length or to showcase their style. When observed in our corpus, words containing alphanumeric are generally not named entities. Thus the presence of alphanumeric in words helps us in identifying negative samples.

6. **Common Symbols:** It is observed that currency symbols, brackets like '(,' '[,' etc. And other symbols are followed by numeric or some mention, not of much importance. Hence, the presence of these symbols is a good indicator of the words before or after them for not being a named entity.

6 Results and Discussion

Table 2 shows CRF results with 'l2-sgd' (Stochastic Gradient Descent with L2 regularization) algorithm for 100 iterations. The c2 value in the CRF model refers to the 'L2 regression,'. Experiments using the algorithms 'pa' (Passive-Aggressive) and 'ap' (Averaged Perceptron) resulted in similar F1-scores of 0.95.

Results for CRF without 'Other' tag are shown in Table 3 which resulted in F1-score of 0.54. We can observe from the results that the feature functions specified are able to capture information related to named entities in the CRF model. The table for feature specific results for the CRF model where results are calculated excluding the 'Other' tag shown in Table 4.

In the experiments with Bi-LSTM, we experimented with the optimizer, activation functions,and number of epochs. After several experiments, the best result we came through was using 'softmax' as activation function, 'rmsprop' as an optimizer,

Tag	Precision	Recall	F1-score
Bg-Loc	0.83	0.48	0.61
Bg-Org	0.83	0.52	0.64
Bg-Pers	0.85	0.55	0.67
It-Loc	0.68	0.27	0.38
It-Org	0.52	0.22	0.31
It-Pers	0.58	0.27	0.37
OTHER	0.96	0.99	0.98
weighted avg	0.95	0.96	0.95

Table 2: CRF Model with 'c2=0.1' and 'l2sgd' algo.

Tag	Precision	Recall	F1-score
Bg-Loc	0.73	0.28	0.40
Bg-Org	0.77	0.32	0.45
Bg-Pers	0.76	0.61	0.67
It-Loc	0.33	0.01	0.01
It-Org	0.15	0.03	0.05
It-Pers	0.51	0.06	0.10
weighted avg	0.72	0.44	0.54

Table 3: CRF Model without 'Other' tag, 'c2=0.1' and 'l2sgd' algo.

Feature	Precision	Recall	F1-score
Char N-Grams	0.68	0.38	0.49
Word N-Grams	0.55	0.08	0.14
Capitalization	0.85	0.44	0.58
Mentions, Hashtags	0.72	0.26	0.38
Numbers in String	0.01	0.01	0.01
Common Symbols	0.02	0.02	0.02

Table 4: Feature Specific Results for CRF.

Tag	Precision	Recall	F1-score
Bg-Loc	0.72	0.32	0.44
Bg-Org	0.69	0.55	0.61
Bg-Pers	0.74	0.72	0.73
It-Loc	0.60	0.06	0.11
It-Org	0.39	0.09	0.15
It-Pers	0.58	0.23	0.33
OTHER	0.98	0.99	0.99

Table 5: Bi-LSTM model with optimizer = 'rmsprop' and has a weighted f1-score of 0.96.

Tag	Precision	Recall	F1-score
Bg-Loc	0.76	0.41	0.54
Bg-Org	0.74	0.49	0.59
Bg-Pers	0.85	0.44	0.58
It-Loc	0.03	0.02	0.02
It-Org	0.24	0.06	0.10
It-Pers	0.26	0.03	0.05

Table 6: Bi-LSTM model without 'Other' tag, optimizer = 'rmsprop' and has a weighted f1-score of 0.54.

Tag	Precision	Recall	F1-score
Bg-Loc	0.65	0.40	0.49
Bg-Org	0.61	0.65	0.63
Bg-Pers	0.57	0.80	0.66
It-Loc	0.50	0.22	0.31
It-Org	0.30	0.25	0.27
It-Pers	0.54	0.41	0.47
OTHER	0.99	0.98	0.98

Table 7: Bi-LSTM-CRF model with optimizer = 'rmsprop' and has a weighted f1-score of 0.96.

Tag	Precision	Recall	F1-score
Bg-Loc	0.75	0.41	0.53
Bg-Org	0.72	0.50	0.59
Bg-Pers	0.85	0.44	0.58
It-Loc	0.25	0.01	0.02
It-Org	0.32	0.16	0.21
It-Pers	0.28	0.04	0.08

Table 8: Bi-LSTM-CRF model without 'Other' tag, optimizer = 'rmsprop' and has a weighted f1-score of 0.55.

Word	Truth	Predicted
amrita	Bg-Pers	Bg-Pers
went	OTHER	OTHER
to	OTHER	OTHER
bangalore	Bg-Loc	Bg-Loc
rama	Bg-Pers	Bg-Pers
na	OTHER	OTHER
imax	Bg-Org	Bg-Org
hattira	OTHER	OTHER
beti	OTHER	Bg-Pers
agalu	OTHER	OTHER

Table 9: An Example Prediction of our CRF Model.

and 'categorical cross-entropy' as our loss function. Table 5 shows the results of BiLSTM on our corpus using seven epochs, and random initialization of embedding vectors. The F1-score is 0.96.
Results for same experiment without including 'Other' tag are shown in Table 6.

In experiments with the Bi-LSTM-CRF model, after several trials, we got the best results with 'softmax' as activation function, 'rmsprop' as an optimizer, and 'crf-loss' as our loss function. Table 7 shows the results of Bi-LSTM-CRF on our corpus using seven epochs, and random initialization of embedding vectors. The F1-score is 0.96.
Results for same experiment without including 'Other' tag are shown in Table 8. Figure 1 shows the Bi-LSTM-CRF model architecture. The training, validation and testing are 70%, 10% and 20% of the total data respectively.

7 Conclusion and Future Work

Our contributions are as follows:

1. Presented an annotated code-mixed Kannada-English corpus for NER, which is, to the best of our knowledge is the first corpus. The corpus will be published online soon.

2. We have experimented with the machine learning models Conditional Random Fields (CRF), LSTM, and LSTM-CRF on our data, the F1-score for which is 0.95, 0.96, and 0.96 respectively, which is looking good considering the amount of research done in this new domain.

3. We are introducing and addressing named entity recognition of Kannada-English code-mixed data as a research problem.

For future work, the corpus can be enriched by also giving the respective POS tags for each token. The size of the corpus can be increased with more NE tags. The problem can be adapted for NER identification in code-mixed data containing more than two languages from multilingual societies.

References

S Amarappa and SV Sathyanarayana. 2013. Named entity recognition and classification in kannada language. *International Journal of Electronics and Computer Science Engineering*, 2(1):281–289.

Rupal Bhargava, Yashvardhan Sharma, and Shubham Sharma. 2016. Sentiment analysis for mixed script indic sentences. In *2016 ICACCI*, pages 524–529. IEEE.

Rupal Bhargava, Yashvardhan Sharma, Shubham Sharma, and Abhinav Baid. 2015. Query labelling for indic languages using a hybrid approach. In *FIRE Workshops*, pages 40–42.

Shambhavi BR and P Ramakanth Kumar. 2012. Kannada part-of-speech tagging with probabilistic classifiers. *international journal of computer applications*, 48(17):26–30.

William B Cavnar, John M Trenkle, et al. 1994. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, volume 161175. Citeseer.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics.

Kevin A Hallgren. 2012. Computing inter-rater reliability for observational data: an overview and tutorial. *Tutorials in quantitative methods for psychology*, 8(1):23.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Stephen Huffman. 1995. Acquaintance: Language-independent document categorization by N-grams. Technical report, DEPARTMENT OF DEFENSE FORT GEORGE G MEADE MD.

Faryal Jahangir, Waqas Anwar, Usama Ijaz Bajwa, and Xuan Wang. 2012. N-gram and gazetteer list based named entity recognition for Urdu: A scarce resourced language. In *Proceedings of the 10th Workshop on Asian Language Resources*, pages 95–104.

BS Sowmya Lakshmi and BR Shambhavi. 2017. An automatic language identification system for code-mixed english-kannada social media text. In *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, pages 1–5. IEEE.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

John Lipski. 1978. Code-switching and the problem of bilingual competence. *Aspects of bilingualism*, 250:264.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444.

P Majumder, M Mitra, and BB Chaudhuri. 2002. N-gram: a language independent approach to IR and NLP. In *International conference on universal knowledge and language*.

K Shalini, HB Barathi Ganesh, M Anand Kumar, and KP Soman. 2018. Sentiment analysis for code-mixed indian social media text with distributed representation. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1126–1131. IEEE.

Ketan Kumar Todi, Pruthwik Mishra, and Dipti Misra Sharma. 2018. Building a kannada pos tagger using machine learning and neural network models. *arXiv preprint arXiv:1808.03175*.

Pius Von Däniken and Mark Cieliebak. 2017. Transfer learning and sentence level features for named entity recognition on tweets. In *3rd Workshop on Noisy User-generated Text (W-NUT), Copenhagen, 7 September 2017*, volume 3, pages 166–171. ACL.

A Example Appendix

This is an appendix.