```python
import bisect
import re

def tokenize_with_offsets(text):
  """Dummy tokenizer.
  Use any tokenizer you want as long it as the same API."""
  tokens, starts, ends = zip(*[
    (m.group(), m.start(), m.end())
    for m in re.finditer(r'\S+', text)
  ])
  return tokens, starts, ends

def get_labels(starts, ends, spans):
  """Convert offsets to sequence labels in BIO format."""
  labels = ["O"]*len(starts)
  spans = sorted(spans)
  for s,e,l in spans:
    li = bisect.bisect_left(starts, s)
    ri = bisect.bisect_left(starts, e)
    ni = len(labels[li:ri])
    labels[li] = f"B-{l}"
    labels[li+1:ri] = [f"I-{l}"]*(ni-1)
  return labels

text = "just setting up my twttr"
(tokens, starts, ends) = tokenize_with_offsets(text)

# tokens = ["just", "setting", "up", "my", "twttr"]
# starts = [0, 5, 13, 16, 19]
# ends = [4, 12, 15, 18, 24]

spans = [(19, 24, "ORG")]
labels = get_labels(starts, ends, spans)

# labels = ["O", "O", "O", "O", "B-ORG"]
```

Listing 1: Conversion of offset format to NER BIO format using one choice of tokenization.

A Converting data to BIO format for NER

In order to convert the dataset to NER format we suggest tokenizing Tweet text and utilizing the character offsets to identify mention tokens. E.g. `just setting up my twttr` with offsets `19` and `24`, and DBpedia category as `Organization`, can be converted to the NER BIO format as follows: `tokens, starts, ends = tokenize_with_offsets("just setting up my twttr")` and then assigning `O` labels to all tokens outside the phrase start and end offsets and `B-ORG` and `I-ORG` label to all tokens within the phrase offsets. This approach works as long as the tokenizer returned offsets correspond to the offset of the phrase in the original text, i.e. tokenization is non-destructive. See example code in listing 1.

B Metrics

Table A1: NERD Metrics

Metric	Description
`strong_mention_match`	`strong_mention_match` is a micro-averaged evaluation of entity mentions. A system span must match a gold span exactly to be counted as correct.
`strong_all_match`	`strong_all_match` is a micro-averaged link evaluation of all mentions. A mention is counted as correct if is either a link match or a nil match. A correct nil match must have the same span as a gold nil. For a correct link match a system link must have the same span and KB identifier as a gold link.
`entity_match`	`entity_match` is a micro-averaged tweet-level set-of-titles measure. It is the same as entity match reported by [Cornolti et al., 2013]

C Dataset details

NER types. See table 1.

Temporal distribution. See figure 3.

C.1 Academic Dataset Details

As explained in section 4.1 it is difficult to sample datasets for NERD tasks to ensure high number of Tweets containing diverse set of entities. Hence, we addressed this sampling issue by including a split based on Tweets already annotated for NERD or related tasks in existing academic benchmarks. This ensures high percentage of Tweets with named entities and linked entities. Please note not all the datasets we include in `TweetNERD-Academic` exist for NERD task. Some exist for NED, some for NER, and some for entity aspect extraction, and some for generic NLP tasks like part-of-speech tagging. We have included these datasets as they contain high density of entities and hence can warrant inclusion in a diverse entity linking test set.

Tgx [Dredze et al., 2016] This dataset is for cross domain co-reference resolution (CDCR). It contains Tweets around the 2013 Grammy music awards ceremony, therefore it mostly contains mentions of Grammy and Music Artists from 2013. Only tweets with person names have been annotated. Original spans detected via NER system and then annotators fixed mention detection issues, grouped similar mentions, and linked to English Wikipedia. Each Tweet annotated by two annotators. No information on annotator agreement provided in the paper. Contains person names who do not occur in Wikipedia.

Broad [Derczynski et al., 2016] This is an NER dataset and hence only contains mention detection annotations. Includes Person, Location, and Organization named entities. Annotations provided by experts and also via crowd-sourcing. They allow annotating username mentions as NE. The dataset has high temporal and geographical diversity with Tweets from 2009 to 2014. They find low agreement among crowd (35% F1) and gold annotations but high recall of named entities. The inter-annotator agreement is high.

Entity Profiling [Spina et al., 2012] Original dataset created for Entity level aspect extraction. Annotation process is non-traditional. We include this dataset for its high availability of named entities.

NEEL 2016 [Rizzo et al., 2016] Dataset created for the Making Sense of Microposts (#Microposts2016) Named Entity rEcognition and Linking (NEEL) Challenge. It cosists of NERD annotations. It includes annotation of Hashtags and user mentions. The dev and test set come from two events from December 2015 around the US primary elections and the Star Wars premiere.

NEEL v2 [Yang and Chang, 2015] This dataset is a combination of [Basave et al., 2014] and [Fang and Chang, 2014]. It includes Tweets annotated for NERD as well as for Information Retrieval (IR) given an entity as a query.

Fang and Chang [2014] Dataset of Tweets from December of 2012 from verified users containing location information. It contains Tweets annotated for NERD as well as IR task. Tweets only annotated for person, organization, location, event, and others NER type. For the IR task the authors take 10 query entities and sample 100 Tweets per query and assess if the Tweet contains a mention of the query entity. Entities come from Freebase [Google, 2022] which contains subset of entities of Wikipedia.

Twitter NEED [Locke, 2009] This dataset consists of Tweets annotated using CoNLL-2003 guidelines. The author allows marking of user mention as named entities. Tweets were collected on February 10 and March 15. It contained Tweets from February 10 about economic recession, Australian Bushfires, and gas explosion in Bozeman, MT on March 15. They found that Topic related Tweets had much higher rate of named entities.

Ark POS [Gimpel et al., 2011] This dataset was created for Part of Speech tagging for Tweets. It contains 6.4 tokens referring to proper nouns which make it likely to contain sufficient named entities and hence a likely candidate to be included for benchmarking NERD systems for Tweets.

WSDM2012 [Meij et al., 2012] It includes 20 Tweets each from a set of verified users. 562 Tweets are manually annotated by two annotators. Annotation was done at the Tweet level where relevant entities for a given Tweet were marked. The authors do not provide agreement rates. The annotated entities may or may not be mentioned explicitly in the text.

Yodie [Gorrell et al., 2015] It consists of Tweets annotated using DBPedia URI from financial institutions and news outlets and climate change discussions. The dataset period is 2013-2014. Tweets were tagged using Crowdflower interface using 10 NLP researchers with each Tweet tagged by three annotators. 89% of entities had unanimous agreement. Tweets were annotated for person, organization, and location entities, while linking included the NIL class.

D Evaluation system details

D.1 Named Entity Recognition (NER)

StanzaNLP [Qi et al., 2020]. Stanza is a collection of accurate and efficient tools for the linguistic analysis of many human languages based on the Universal Dependencies (UD) formalism and includes named entity recognition as a functionality. For each document stanza outputs entity mentions and their start and end character offsets which can be directly used for neleval evaluation.

Spacy[6] Spacy NLP library provides a transition-based named entity recognition component. The entity recognizer identifies non-overlapping labelled spans of tokens. The loss function optimizes for whole entity accuracy, which assumes a good inter-annotator agreement on boundary tokens for good performance. Spacy identified mentions are in the desired character offset format and hence can be directly used for evaluation.

AllenNLP [Peters et al., 2017]. The AllenNLP named entity recognizer uses a Gated Recurrent Unit (GRU) character encoder as well as a GRU phrase encoder, and it starts with pretrained GloVe vectors for its token embeddings. It was trained on the CoNLL-2003 NER dataset. AllenNLP outputs BIO labels. To extract mentions and their start and end character offsets we first extract the mentions from the BIO labels corresponding to the non-O tokens. We then perform a search for this phrase in the Tweet text to get the start and end offsets. This leads to some edge cases such as if there are two identical mentions correctly identified, we always count only the first match hence over-penalizing the model. On the other hand, if mention identified by the model was the latter one but only the former mention was part of the gold annotation we under-penalize the model.

Twitter NER [Mishra and Diesner, 2016]. Twitter NER is a conditional random field model trained specifically for Tweets using a combination of rules, gazetteers, and semi-supervised learning. It is a prominent non-neural baseline for NER on Tweets.

[6] https://spacy.io/api/entityrecognizer.

Social Media IE [Mishra, 2019]. SocialMediaIE is a multi-task model trained on a combination of tasks for social media information extraction. It uses a pre-trained language model along with multi-dataset multi-task learning setup and is jointly trained to perform NER, Part-of-Speech tagging, Chunking, and Supersense tagging.

BERTweet [Nguyen et al., 2020] fine-tuned on WNUT17 [Derczynski et al., 2017]. BERTweet[7] is a BERT style model specifically trained on 850M Tweets from 2012 to 2019 which has been found to perform best on multiple Tweet benchmarks.

D.2 Entity Linking given True Spans (EL)

Given true entity mentions from human annotated data, we compare linking only performance (also known as entity disambiguation) using `entity_match` and `strong_all_match` from neleval.

GENRE (Generative ENtity REtrieval) [Cao et al., 2021]. GENRE is a sequence-to-sequence model that links entities by generating their name in an autoregressive fashion. Its architechture is based on transformers and it fine-tunes BART [Lewis et al., 2019] for generating entity names, which in this case are corresponding Wikipedia article titles. We used the model that was trained on BLINK + AidaYago2.

REL (Radboud Entity Linker) [van Hulst et al., 2020][8]. REL is an open source toolkit for entity linking. It uses a modular architecture with mention detection and entity disambiguation components. We use REL *with* mentions to get *only* entity disambiguation results here.

Lookup Lookup is a simple heuristic based system. Given true mentions, we fetch the most likely entity based on popularity defined via mention candidate co-occurrence in wikipedia.

D.3 End to End Entity Linking (End2End)

To compare end to end entity linking systems we use `entity_match` and `strong_all_match` from neleval. Some of the models mentioned here have been introduced in Section D.2

GENRE. For end-to-end entity linking, a Markup annotation is used to indicate the span boundaries with special tokens, and the decoder decides to generate a mention span, a link to a mention, or continue to generate the input at each generation step. Therefore, the model is capable of both detecting and linking entities.

REL. We use REL *without* mentions to get complete End2End linking results in this case.

TagMe [Ferragina and Scaiella, 2012][9]. It is an end to end system and is based on a directory of links, pages and Wikipedia graph. We use TagME to get linking results.

DBPedia Spotlight [Daiber et al., 2013]. Spotlight first detects mentions in a two step process; in the first step, all possible mention candidates are generated using different methods, and the second step selects the best candidates based on a score which is a linear combination of selected features (such as annotation probability). The linking/disambiguation part uses cosine similarity and a vector representation which is based on a modification of TF-IDF weights.

Natural Language AI (NLAI) [10]. We use the `documents:analyzeEntities` endpoint of the API to get the entities in the Tweet. The system is black-box but is likely to use deep neural network based solutions for entity recognition and entity linking.

[7] https://huggingface.co/socialmediaie/bertweet-base_wnut17_ner
[8] https://github.com/informagi/REL
[9] https://github.com/gammaliu/tagme
[10] https://cloud.google.com/natural-language