1 Annotation Guidelines

The task here is to find the salient named entities from a tweet.

By named entities, we mean all the phrases in the tweet that are uniquely identifiable with a name. For example, the name ‘Dravid’ is a named entity of category Person. We are interested in only four categories of named entities, Person, Place, Organization and Event. Some examples for each category are given below:

1. Person - Dravid, Brendon McCullum, Nolan
2. Place - Delhi, India, MCG
3. Organization - Google, Apple, Indian Cricket Board, Blackcaps (New Zealand team)
4. Event - World cup 2015, Sa vs Nz match

In the annotation interface, enter your name and collection (either training set or inter-annotator set). You can see a tweet with an associated image. The training set consists of 10938 tweets. In the inter-annotator set, there are 60 tweets, with 20 each from 3 topics, namely Movie (tweets posted for 62nd national film awards in India, whose results were announced on March 24, 2015), Cricket (tweets from Cricket World Cup 2015) and Apple (tweets from launch of Apple iWatch on April 24, 2015). Below the image, you get to see a list of named entities to select. These named entities may not be precise or clean as they are identified by programs. For example, the entry @amlahash denotes the cricketer ‘Hashim Amla’ but is embedded with other characters. Still it can be considered as a valid named entity. Often the named entities listed could overlap, like Amla and Hashim Amla. In such cases you should choose the longest correct named entity which is Hashim Amla in this case. Sometimes, you get to see more than one named entities falsely marked as one single entity (for instance, Apple AAPL MacBook AppleWatch AppleWatchEvent). In this case, you can mark them as **N**ot Annotatable.

Your task now is to select the salient named entities and mark those named entities if you can see it in the image. The following table list some tips to do that for each category of named entities.

1. Person (Sangakkara) - Mark it if you can find the person Sangakkara in the image. Even a piece of text ‘Sangakkara’ in the image should also be considered.
2. Place (MCG) - Mark it if you are confident that the place is Melbourne cricket ground. You can always google for similar images.
3. Organization (Blackcaps) - If you get to see any New Zealander in the picture or logo of the New Zealand team, you can mark it.
4. Event (World cup 2015) - If you get to see any clue for World cup 2015, you can mark it.

Every-time you select a named entity, you get to see another list of options that corresponds to it’s Wikipedia entry. Pick an entry which perfectly matches the Wikipedia page. For example, if the named entity is “CWC15”, it’s corresponding Wikipedia page is `http://en.wikipedia.org/wiki/2015_Cricket_World_Cup` and the option you choose is whatever that comes after `http://en.wikipedia.org/wiki/` which is `2015_Cricket_World_Cup`. Since this list is also computed by a program, there could be a situation where none of the options explains the selected named entity. If such is the case, you can mark it as ‘None’.

If you see a sarcastic tweet, do all the above and put **S** in the comment sections. For example a morphed image of the NE or tweet-text contradicting the NE (as in calling victory a defeat) should be marked S. If you see an advertisement or pointless tweet (tweet not related to cricket (for training set) or any of the three topics (for inter-annotator set)), you need not annotate named entities and mark **P** in the comment section. If a tweet appears like a retweet (repeating tweet text and/or tweet image), you don’t need to annotate, mark **D** in the comment section.

All the best!

2 Tweet Linker Details

Tweet Linker is a supervised binary classifier, trained using the CWC15 tweets that have a Wikipedia title. In this appendix we explain Wikipedia article sources and features used in it.

The Tweet Linker links the SNEs annotated in the dataset to their Wikipedia articles. While we used Wikipedia data dump as source of Wikipedia articles, we found that alternate sources of processed Wikipedia articles were also giving good results in our experiments. A brief explanation of the three different sources of Wikipedia articles (referred as KB heneforth) and results using them is presented in Table 1. For training Tweet Linker we use CWC15 tweets having relevant Wikipedia title as positive sample. We report the performance for positive samples in a 5-fold cross validation.

2.0.1 KBs

Wikipedia: The English Wikipedia dump dated 10 June, 2015 is indexed using Lucene[1]. Using Lucene's multi-field query, the SNE mention is searched in the Wikipedia title and tweet is searched in the description of the Wikipedia article. The Lucene score (an OR query, score is sum of field scores) gives the context similarity score.

DBpedia: [2] [**?**] The SNE mention is searched in DBpedia. For each entity retrieved from DBpedia, the abstract of the entity is analyzed for similarity with the tweet. This gives context similarity score.

GCD: Google Cross-Wiki Dictionary (GCD) [**?**] is a string to entity mapping, where strings are the anchor hyper-texts that refer to the Wikipedia page titles. A ranked list of probable Wikipedia entities are retrieved for an SNE. The ranking criterion is the Jaccard similarity between the anchor text and the tweet. So if the tweet is highly similar to the anchor text, then context similarity will have a high score.

The Tweet Linker classifies the linking of Wikipedia entities given by KB as correct or wrong. In Table 1, we compare the performance of three KBs in terms of P, R and F, using a Random Forest (RF) classifier, considering top 1 KB retrieval rank. Here, we find that sources of processed Wikipedia articles doing better than unprocessed Wikipedia articles. DBpedia gave better F than other KBs. This is probably due to better retrieval ranking in DBpedia and relative freshness of the data, since GCD was created in 2012, when number of articles in Wikipedia was 3.8M, whereas DBpedia accesses the current Wikipedia having 4.8M articles.

Table 1. Choice of KB

KB	**P**	**R**	**F**
Wikipedia	0.60	0.45	0.512
DBpedia	**0.81**	0.54	**0.64**
GCD	0.64	**0.58**	0.58

The Tweet Linker chooses the topmost Wikipedia entities given by KB and classifies the linking as correct or wrong. A RF classifier built using features described in 2.0.2 is used as the linker, giving F of 0.64.

2.0.2 Features

F1. Lexical position or Word Order of NE: When there are multiple NEs detected in a tweet, this feature captures the index of the SNE in the list of NEs.

F2. Presence in Hashtag: This binary feature indicates if the SNE occurs within a hash tag, i.e if the SNE is wholly or partly present in a hashtag.

F3. Preceded by @: This binary feature indicates if the SNE is preceded by a @ symbol in the tweet, which makes it a Twitter userid.

F4. Topic word: This feature indicates if the SNE contains the topic word. Here topic refers to topic of the tweet collection (discussed in feature *F8*).

F5. Retrieval Rank: The Wikipedia and GCD corpus are indexed using Lucene. The rank of the entity is normalized with result size. For DBpedia, this is the retrieval rank of DBpedia query results.

F6. Context Similarity: Context Similarity is calculated as the Jaccard similarity between the tweet and the context of the Wikipedia entity in the KB, as suggested by Dalvi *et al.* [**?**]. In case of GCD, context is the anchor text from web page. In Wikipedia and DBpedia, context is the abstract of the entity.

F7. Link Probability: Probability that the mention is a hyperlink in Wikipedia. A corpus of all hyperlinks in Wikipedia is created. Link Probability is the ratio of number of times a mention occurs in

[1] `https://lucene.apache.org/`

[2] `http://wiki.dbpedia.org/Ontology`

this corpus to the total number of times it occurs in Wikipedia.

F8. Salience Probability: We define salience probability as the probability that the NE is salient to tweets on the topic. To calculate this we first collect the tweets on given topic. In the case of CWC15, the topic is 'cricket' and in the case of Replab filtering task, the topic is the entity-of-interest (for example 'Porche' or 'MIT') . The list of experts in the topic is obtained from the Cognos API [**?**]. We collect tweets of these experts to create the topic-specific tweet collection. NERs (same as the one used in candidate SNE generation in Section **??**) identify the NEs in the tweets. For a new tweet in this topic, we argue that the NEs prominent to the topic are the salient NEs of the tweet. Let t be a tweet containing NE n and topic c. Let salience probability or probability that n is the SNE of a tweet, be referred to $P_s(n|t)$. We can write this as Equation 1.

$$P_s(n|t) \propto P(n|TopicTweetCollection) \qquad (1)$$

Assuming independence of the entities in a given tweet Equation 1 can be written as Equation 2.

$$P_s(n|t) = P(n|c,t) \cdot P(c|t) \qquad (2)$$

where

$$P(c|t) = \frac{N(c)}{\sum_{c\prime} N(c\prime)} \qquad (3)$$

$$P(n|c,t) = \frac{\sum_{n \in t} N(c)}{N(c)} \qquad (4)$$

and N(c) is the Number of tweets in topic c.

F9. Page Rank: The NEs prominent to the topic are ***N*** (created in *F8*). These NEs are used to create the Page rank vector. Two NEs are linked if the same expert talks about both of them and weight of the link is the Pointwise Mutual Information (PMI) [**?**] of the NEs.

$$PMI(N_1, N_2) = \log \frac{n(N_1, N_2)E}{n(N_1)n(N_2)} \qquad (5)$$

where $n(N_{(.)})$ is the number of experts who talked about the named entity $N_{(.)}$ and E is the total number of experts in this Cognos topic. The page rank vector of SNE (when present) is used as the feature for the learner. Creating the topic-specific tweet collection and page-rank calculation is done as pre-processing.