Twitter Named Entity Extraction and Linking Using Differential Evolution

Utpal Kumar Sikdar and **Björn Gambäck**
Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
{sikdar.utpal,gamback}@idi.ntnu.no

Abstract

Systems that simultaneously identify and classify named entities in Twitter typically show poor recall. To remedy this, the task is here divided into two parts: i) named entity *identification* using Conditional Random Fields in a multi-objective framework built on Differential Evolution, and ii) named entity *classification* using Vector Space Modelling and edit distance techniques. Differential Evolution is an evolutionary algorithm, which not only optimises the features, but also identifies the proper context window for each selected feature. The approach obtains F-scores of 70.7% for Twitter named entity extraction and 66.0% for entity linking to the DBpedia database.

1 Introduction

Twitter has established itself as one of the most popular social networks, with about 320 million active users daily generating almost 500 million short messages, *tweets*, with a maximum length of 140 characters (Twitter, 2016). The language used is very noisy, with tweets containing many grammatical and spelling mistakes, short form of the words, multiple words merged together, special symbols and characters inserted into the words, etc. Hence it is difficult to analyse and monitor all types of tweets, and the vast number of tweets: specific messages may need to be filtered out from millions of tweets. Named entity extraction plays a vital role when filtering out relevant tweets from a collection. It is also useful for pre-processing in many other language processing tasks, such as machine translation and question-answering.

The paper is organized as follows: Section 2 describes related work on Twitter named entity recognition and linking. The actual Twitter name identification methodology and different features used are presented in Section 3. Section 4 focuses on classification of the identified named entities and their linking to DBpedia. Experimental results and a discussion of those appear in Section 5 and Section 6, respectively, while Section 7 addresses future work and concludes.

2 Related Work

The noisiness of the texts makes Twitter named entity (NE) extraction a challenging task, but several approaches have been tried: Li et al. (2012) introduced an unsupervised strategy based on dynamic programming; Liu et al. (2011) proposed a semi-supervised framework using a k-Nearest Neighbors (kNN) approach to label the Twitter names and gave these labels as an input feature to a Conditional Random Fields, CRF (Lafferty et al., 2001) classifier, achieving almost 80% accuracy on their own annotated data. Supervised models have been applied by several authors, e.g., Ritter et al. (2011) who applied Labeled LDA (Ramage et al., 2009) to recognise possible types of the Twitter names, and also showed that part-of-speech and chunk information are important components in Twitter NE identification.

A shared task challenge was organized at the ACL 2015 workshop on noisy user-generated text (W-NUT) (Baldwin et al., 2015), with two subtasks: Twitter named entity identification and classification of those named entities into ten different types. Of the eight systems participating, the best (Yamada et al., 2015) achieved an F_1 score of 70.63% for Twitter name identification and 56.41% for classification, by combining supervised machine learning with high quality knowledge obtained from several open knowledge bases such as Wikipedia. Akhtar et al. (2015) used a strategy based on differential evolution, getting F_1 scores of 56.81% for identification and 39.84% for classification.

D S Sharma, R Sangal and A K Singh. Proc. of the 13th Intl. Conference on Natural Language Processing, pages 198–207,
Varanasi, India. December 2016.

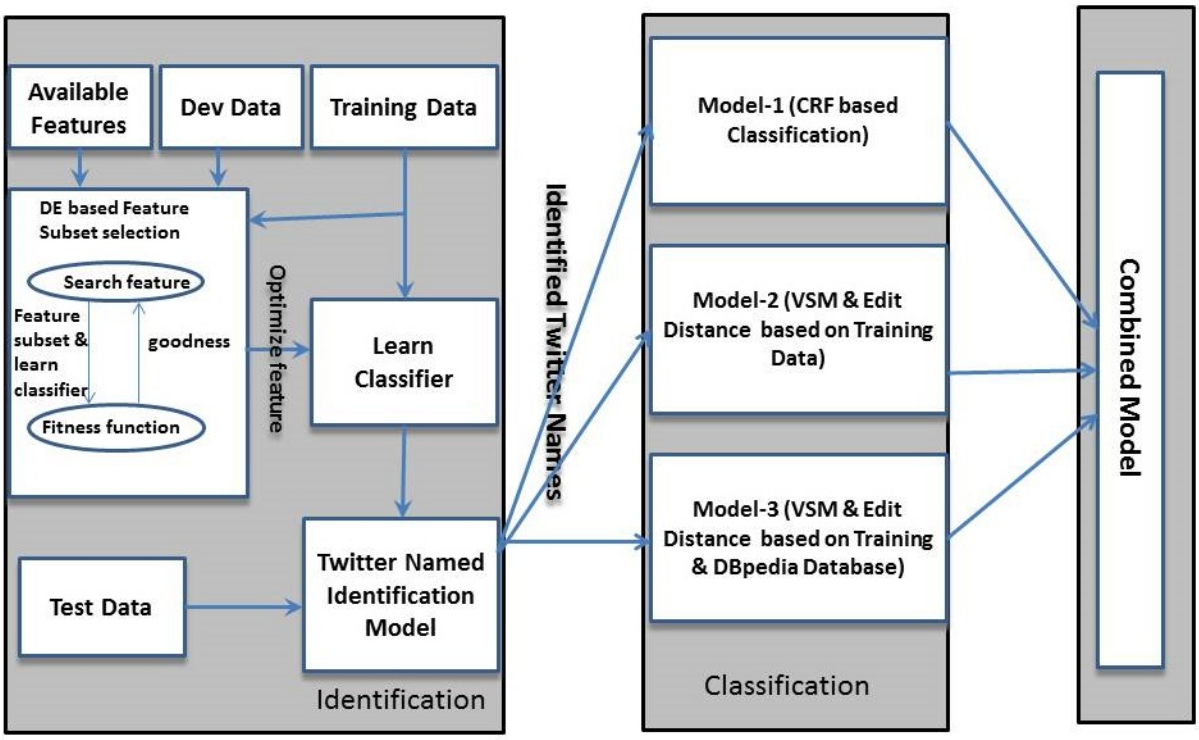

Figure 1: Twitter Named Identification and Classification system.

The challenge was repeated at W-NUT 2016 (Strauss et al., 2016), with ten participating teams. The best system (Limsopatham and Collier, 2016) used a bidirectional Long Short-Term Memory (LSTM) recurrent neural network model, obtaining F-scores of 65.89% for Twitter name identification and 52.41% for the classification task. Another system (Sikdar and Gambäck, 2016) utilized a CRF classifier built on a large feature set to get the second highest F-score on identification: 63.22%, but lower (40.06%) on classification.

A specific shared task on Twitter named entity recognition and linking (NEEL) to DBpedia was held at the #Microposts2016 workshop (Rizzo et al., 2016), with the problem defined as to identify named entities from the tweets (called 'Strong_typed_mention_match') and to link them to the DBpedia database ('Strong_link_match'). DBpedia extracts structured information from Wikipedia and links different Web data sets to Wikipedia data, allowing for sophisticated queries against Wikipedia. The DBpedia knowledgebase is available in 125 languages, with the English version describing 4.58 million items, out of which 4.22 million are classified in a consistent ontology.

Five teams participated in the #Microposts2016 NEEL challenge. However, most of the systems suffered from very low recall values in the Twitter NE identification task and were actually unable to efficiently recognise Twitter names: Two knowledge-based approaches (Caliano et al., 2016; Greenfield et al., 2016) achieved F-scores of 26.7% and 31.9%, respectively, due to recall values of 18.8% and 24.0%. Two other systems (Ghosh et al., 2016; Torres-Tramón et al., 2016) produced recall values of 28.9% and 24.2%. The best system (Waitelonis and Sack, 2016) achieved recall, precision and F-measure values of 49.4%, 45.3% and 47.3%. In this system, each token is mapped to gazetteers that are developed from the DBpedia database. Tokens are discarded if they match with stop words or are not nouns.

To increase recall and F-score, we take a two-step approach to identifying and classifying named entities in noisy user-generated texts. In the first step, Twitter names are identified using CRF within the framework of Differential Evolution (Storn and Price, 1997). In step two, the named entities are classified into seven categories and linked to DBpedia using a vector space model and edit distance techniques. The identified named entities are also classified using CRF, and the outputs of the classification models are later combined. Figure 1 shows the system architecture.

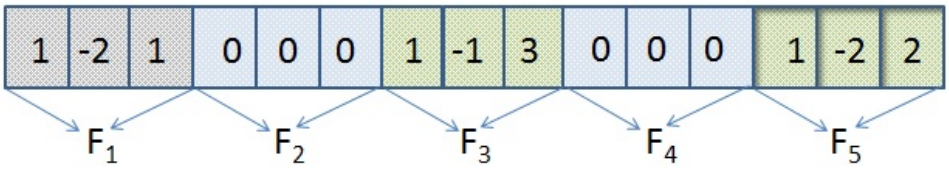

Figure 2: Chromosome representation of five available features; three are present and two absent.

3 Twitter Named Entity Identification

Twitter named entities are first extracted using a supervised machine learning approach, namely CRF in a Differential Evolution (DE) setting. However, Twitter names contain a lot of noise, making it difficult to identify them directly from the texts, so some words are segmented: words containing special characters (e.g., #, @, _), words containing a lower-case letter followed by a upper-case letter (e.g., 'realDonaldTrump' is split into real, Donald and Trump), letters followed by digits, etc. This section first briefly introduces Multi-objective Differential Evolution and then describes the DE-based Twitter name identification procedure (the left side of Figure 1).

3.1 Multi-objective Differential Evolution

Differential Evolution (Storn and Price, 1997) is a parallel direct search method over complex, large and multi-modal landscapes, and in general provides near-optimal solutions to an optimization problem. In DE, the parameters of the search space are encoded in the form of strings called chromosomes. A collection of N such strings is called a population, and is denoted by D-dimensional parameter vectors $X_{i,G} = [x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G}]$, $i = 1, 2, \ldots, N$ for each generation G. The initial vector population is chosen randomly, which covers different points in the search space.

For multi-objective optimization, more than one objective or fitness function is associated with each chromosome. These objective values represent the degrees of goodness of the string. DE generates new parameter vectors ("mutation") by adding the weighted difference between two population vectors to a third vector. The mutated vector's parameters are then mixed ("crossover") with the parameters of another predetermined vector, the target vector. For selection, these N trial vectors are merged with the current population, and the best N solutions are chosen from these $2 \times N$ candidate solutions based on domination, non-domination, and crowding distance. The processes of mutation, fitness computation, crossover and selection are executed for a pre-selected maximum number of generations.

3.2 DE-based Named Entity Extraction

When extracting named entities, suppose that there are a number of available features $F_1(-m, n), F_2(-m, n), \ldots F_K(-m, n)$, where K represents the total number of features, while m and n denote the preceding and succeeding context window lengths of each feature. Differential Evolution aims to find the relevant features along with proper context windows and learn a classifier using these features to optimize two objective functions: precision and recall.

Each DE chromosome represents the number of features along with their context windows, so the length of the chromosome is $D = K \times 3$. Each feature consists of one bit and two integer values: the bit represents presence or absence of a feature (1 or 0), two integer values denote the length of the preceding and succeeding context windows (0–5). The algorithm proceeds as follows:

Chromosome Initialization: All chromosomes in the first population generation are initialized with random values within the search space. The bit position feature value of '0' indicates that the particular feature is not participating in constructing the classifier and '1' indicates that the feature is present in constructing the classifier using the context features. The chromosome initialization is shown in Figure 2, where features F_1, F_3 and F_5 are present, and features F_2 and F_4 are absent.

Fitness Computation: More than one fitness function can be associated with each chromosome. Suppose that k features are present in the chromosome and that the context window for each feature is denoted by $F_p(-m_p, n_p)$ where $1 \leq p \leq k$. In Figure 2, the preceding and succeeding context window lengths of F_1, F_3 and F_5 are respectively (2, 1), (1, 3) and (2, 2). The CRF classifier is

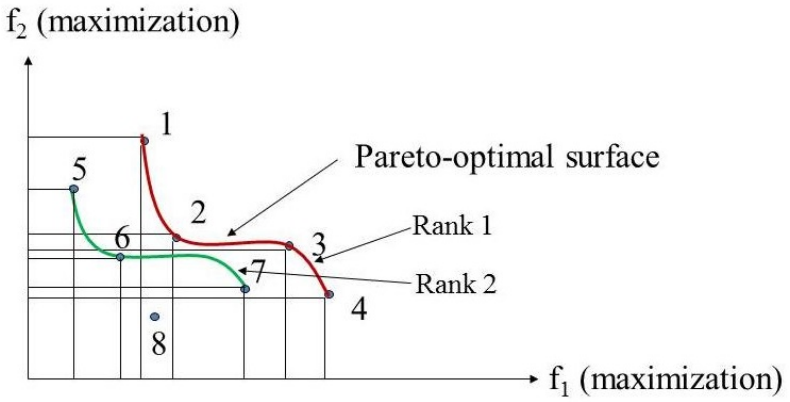

Figure 3: Dominated and non-dominated solutions. Here solutions 1, 2, 3 and 4 are non-dominating in relation to each other (rank 1). Solutions 5, 6 and 7 are also non-dominating to each other (rank 2), while solution 8 is dominated by at least one solution from rank 1 or rank 2.

trained on the k features along with their context features, and the DE search capability maximizes the two objective values precision and recall.

Mutation: A mutant vector $V_{i,G+1}$ is generated for each target vector X according to

$$V_{i,G+1} = X_{r1,G} + F_{mu} \times (X_{r2,G} - X_{r3,G})$$

where $\{r1, r2, r3\} \in \{1, 2, \ldots, N\}$) are randomly generated indices, $F_{mu} \in [0, 1]$ a mutation factor, and G resp. $G+1$ the current and next generations.

Crossover: Crossover is introduced to increase the diversity of the mutant parameter vector: the parameters of the mutant vector $V_{i,G+1}$ are mixed with the parameters of the target vector $X_{i,G}$ to generate a trail vector $U_{i,G+1}$:

$$U_{i,G+1} = (u_{i,1,G+1}, u_{i,2,G+1}, \ldots, u_{i,D,G+1})$$

where for $j = 1, 2, \ldots, D$,

$$u_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & : rd_j \leq C_r \ \ or \ \ j = j_{rd} \\ x_{i,j,G} & : rd_j > C_r \ \ and \ \ j \neq j_{rd} \end{cases}$$

where $rd_j \in [0, 1]$ is a randomly generated floating point value, $C_r \in [0, 1]$ is the crossover constant, and $j_{rd} \in \{1, 2, \ldots, D\}$ a randomly chosen index ensuring that the trail vector gets at least one parameter from the mutant vector.

Selection: To select the N best chromosomes for the next generation $G + 1$, the trial population is merged with the current population. These $2{\times}N$ solutions are ranked based on domination and non-domination relations in the objective function space. A solution non-dominated if at least one of the objective values is better than another solution. If all objective values of a solution are better than another solution, the latter solution is dominated. All the non-dominated solutions form a Pareto-optimal front (Deb, 2001), as exemplified in Figure 3. The top ranked solutions are added to the next generation population until the total number of solutions is equal to N. If the number of solutions of a particular rank is $> N$, the crowding distance sorting algorithm (Deb, 2001) is applied to discard the excess solutions.

Termination Condition: The processes of mutation, crossover, fitness computation and selection are executed for G_{max} generations. A set of non-dominated solutions is obtained on the final Pareto optimal front from the last generation population. None of these solutions is better compared to the others, and each of them represents a set of optimal feature combinations. The last generation population contains rank 1 solutions where some are good with respect to recall and others with respect to precision. Here, the solutions with highest F-scores are selected, but any criterion can be used based on user preferences.

3.3 Twitter Named Entity Features

A range of different features are utilised to extract named entities from the noisy text. All features (except part-of-speech) are domain independent, and most of them are binary (yes/no) flags:

- Local context (F_1): The preceding and succeeding words of the current token.
- Word prefix (F_2) and suffix (F_3): Up to 4 leftmost/rightmost characters.

- Special character followed by token (F_4): If a special character (e.g., @, #) follows the token, this binary flag is set.
- Gazetteer list (F_5): A named entity list is extracted from the training data and F_5 indicates if the current token is on that list.
- Last (F_6) and first word (F_7): The token is the last/first word of the sentence.
- Alphanumeric (F_8), all digit (F_9): The token contains only alphanumerics or digits.
- Single capital (F_{10}), capital inner (F_{11}), initial capital (F_{12}), and all capital (F_{13}): The token contains only one single capital letter, any capital letter in inner position, starts with a capital letter, or contains only capitals.
- Stop word match (F_{14}): The current word matches with a stop word.
- Word normalization (F_{15}): Tokens with similar word shapes may belong to the same class. Word mapped to its equivalent class: each capital letter in the token is mapped to A, small letters to a, and digits to 0. Other characters are kept unaltered.
- Word frequency (F_{16}): Whether the word's frequency is greater than a certain threshold (the threshold values were set to 10 for training data and to 5 for development/test data).
- Word previously occurred (F_{17}): If the current word occurred earlier in the corpus.
- Word length (F_{18}): The word length is greater than some threshold (e.g., ≥ 5).
- Part-of-speech (F_{19}): Part-of-speech tags from the TweeboParser[1].
- Dynamic (F_{20}): The output label(s) of the previous token(s).

4 Twitter NE Classification and Linking

The previous section described how Twitter named entities are identified using CRF and Differential Evolution. In this section, the identified Twitter named entities are classified into seven predefined categories (Thing, Event, Character, Location, Organization, Person and Product) and linked to the DBpedia database. Three models are built for classification of Twitter names. The outputs of these three models are later merged using a priority-based approach (precision value). Finally, the classified Twitter names are linked to DBpedia.

The three Twitter named entities classification models are composed as follows.

Model-1: A CRF-based supervised classification technique is simply used to predict the classes of the Twitter names. All the extracted Twitter names from the DE-based approach are used as a feature in this model. So the model is developed with this feature along with the previous and next two context words of the current word.

Model-2: Model-2 is developed based on training datasets. Each type of training instances (Twitter names) is stored in a unique document whose class is one of the seven predefined categories. After removing all stop words and special characters from the tweets, the remaining non-entity words are stored in a separate document. Each extracted Twitter name from the DE-based approach is considered as a query (search keyword). The following steps are used to classify Twitter names.

1. Retrieve the top-100 documents for each query using a Vector Space Model (Salton et al., 1975) based on cosine similarity.
2. Calculate edit distance (also called Levenshtein distance) between the query and each of the retrieved documents.
3. Assign the closest document's class to the query, where the closest document is the one with minimum edit distance to the query. The class of an entity is retrieved using Lucene.[2]

Model-3: Model-3 is based on the DBpedia and training datasets. From the DBpedia ontology classes,[3] the datasets are mapped to the seven predefined categories and the same technique as described in Model-2 is applied. We extracted rdf-schema#label items from the csv files (e.g., African Volleyball Championship) and stored each in a unique document whose class is one of the seven predefined categories. In Model-3, the training data is merged with the DBpedia datasets.[4]

Combined Model: The extracted Twitter named entity classes are merged based on the priority of the above models. The priorities are given to the models based on the precision value of the development data. Suppose, for example, that Model-3 has the highest priority followed by Model-2 and Model-1. Voting in the ensemble is then carried out as follows: If Model-3 is unable to identify the

[1] http://www.cs.cmu.edu/~ark/TweetNLP/

[2] https://lucene.apache.org/core/3_5_0/contributions.html

[3] http://mappings.dbpedia.org/server/ontology/classes/

[4] Using the csv format of the DBpedia database.

Dataset	Number of	
	Tweets	Entities
Training	4,073	8,665
Development	100	340
Test	296	1,022

Table 1: Statistics of the NEEL2016 datasets

class of an entity (i.e., if it identifies it as belonging to the 'not-Entity' document class), the entity is passed to Model-2. If Model-2 also fails to identity the class of the entity, the class of the entity is assigned by Model-1.

A knowledge-based technique is used to link the Twitter named entities to DBpedia database. In the training data, for each entity a link was provided. In DBpedia, for each rdf-schema#label, a link was assigned. When the class of an entity (Twitter name) is extracted using the above models, the corresponding link is maintained based on training or DBpedia datasets, and the link is assigned to that particular entity. If no link is found, a NIL link is assigned to the entity.

5 Experiments and Results

The approach was tested on the NEEL2016 datasets (Rizzo and van Erp, 2016). The statistics of the datasets are given in Table 1. This data was used to train a CRF-based classifier as baseline and then to build all the models of the full DE-based system, for Twitter named entity identification as well as for categorization and DBpedia linking, as described in turn below.

5.1 Baseline Model

To obtain a baseline model, the CRF-based machine learning approach was applied to identify and classify the Twitter named entities using the features described in Section 3.3 and the same evaluation scorer as in the NEEL2016 challenge.

After building a classifier on the training data, the recall (R), precision (P), and F_1-measure values of the development datasets were 36.20%, 69.9% and 47.7% (as also shown in the first row of Table 3 below). The low F-score is due to the poor recall. The baseline approach was also evaluated on the unseen test data, obtaining recall, precision and F_1 scores of 23.7%, 50.7% and 32.3%. The performance on the test data is similar to that on the development data because of the bad recall.

In order to increase the data for available for training the models, the development data was merged with the training data. This also increases the number of named entities in the gazetteer list (feature F_5) and the statistics that some of the other features are based on. When the same model is built by merging the training and development data, and evaluated on the test data, the results are improved considerably, with recall, precision and F-measure values of 55.6%, 69.7% and 61.9%.

5.2 Twitter NE Identification Experiments

To enhance recall and F-score, Twitter names are first identified using the multi-objective DE-based technique. In a second step, the identified Twitter named entities are classified using vector space modelling and edit distances.

A baseline model for identification of Twitter names was built using the features described in Section 3.3. When trained on the training data and evaluated on the development data, the model shows recall, precision and F_1-scores of 56.1%, 87.5% and 68.4%, respectively, as given in the 'Dev Data' column of Table 2.

To improve on these results, a Differential Evolution-based feature selection technique was used to identify named entities in noisy text. For Twitter named entity identification, the parameters of the DE were set as follows:

- N (population size) = 100,
- C_r (probability of crossover) = 0.5,
- G_{max} (number of generation) = 100, and
- F_{mu} (mutation factor) = 0.5.

The best feature set along with the context features was determined based on development data. The selected features along with context features are $F_1(-3,4)$, $F_5(-1,0)$, $F_6(0,1)$, $F_8(-1,0)$, $F_{11}(0,1)$, $F_{12}(-2,2)$, $F_{14}(-2,1)$, $F_{17}(-2,1)$, $F_{19}(-2,1)$ and $F_{20}(-1,0)$. This setup achieved recall, precision and F-measure values of 79.9%, 93.8% and 86.3% on the development data. The performance of the system increases almost 18 F-measure points over the baseline model.

To fairly evaluate the approach, it was applied to the unseen test data using the selected features along with the context features, giving the recall, precision and F_1-scores of 73.9%, 89.2% and 80.8%, respectively; also shown in Table 2 (the 'Test Data' column). The F-measure performance is increased by 20 points over the baseline.

Training set / Test set / Method	Training Data: Dev Data R	Dev Data P	Dev Data F_1	Training Data: Test Data R	Test Data P	Test Data F_1	Training + Dev Data: Test Data R	Test Data P	Test Data F_1
Baseline (CRF-based)	56.1	87.5	68.4	48.4	81.8	60.8	79.3	88.2	83.5
DE-based identification	79.9	93.8	86.3	73.9	89.2	80.8	81.3	90.8	85.7

Table 2: Twitter Named Entity Identification results

When merging the development data with the training data and building a model using selected features, the recall, precision and F-measure values are 81.3%, 90.8% and 85.7%, respectively, also improving on the scores obtained by the baseline model. The results show that the multi-objective DE-based approach efficiently identifies Twitter names from noisy text.

5.3 Twitter NE Classification Experiments

In the next step, the identified Twitter names extracted using the multi-objective DE-based approach were classified. Three models were built to classify the Twitter named entities into the seven categories described at the beginning of Section 4.

In the first model (Model-1), the identified Twitter named entities are passed through a CRF-based supervised classifier. The recall, precision and F-measure values given in Table 3 are 38.8%, 40.7% and 39.8%, respectively, showing that the recall is increased over the baseline model (Section 5.1), which simultaneously identifies and classifies the named entities.

Another Twitter named classification model (Model-2) was built using VSM and edit distance techniques based on training data, giving recall, precision and F-scores of 27.9%, 87.2% and 42.3%. The probable reason for the low performance is that sufficient examples (instances) may not be found in the training data.

When the Model-2 approach was applied to the DBpedia database (Model-3), the recall, precision and F-scores are 67.4%, 86.7% and 75.8%, respectively. Model-3 achieves one of the best results among all the models and also outperforms all existing approaches, as shown in Table 3.

All the models were blindly evaluated on the test data (the second set of R-P-F_1 scores given in Table 3). The Model-3 results show that its performance is far better than all other present state-of-the-art approaches. It achieves recall, precision and F-measure values of 50.8%, 76.1% and 60.9%, respectively. Even though the performance of Model-3 on the development data is only slightly better than the best existing system, KEA (Waitelonis and Sack, 2016), the gap in performance on the test data is a lot larger, since the KEA system tries to increase the recall value as much as possible without controlling the precision value. As a result, KEA's performance on the test data is fairly bad.

In the last model (Combined Model), the outputs of the three models are merged, increasing the F-score almost 5 points compared to Model-3 on the test data. The precision, recall and F_1-scores of the Combined Model are 65.3%, 67.1% and 66.2%, respectively (see Table 3). Hence the Combined Model produces the best results compared to all the other models as well as to all the #Microposts2016 systems.

When merging the development data with the training data (the third set of R-P-F_1 scores in Table 3), the performances of Model-1 and Model-2 are better than Model-3 because many examples in the test data are seen in the development data and these two models are developed based on training instances, while Model-3 classifies the Twitter names based on the DBpedia database. Here, the Combined Model also achieves the best results among all the approaches, with precision, recall and F-score of 69.7%, 71.7% and 70.7%.

5.4 Twitter NE Linking Experiments

To link the classified Twitter names, we use a knowledge-based approach (utilizing either training data or the DBpedia database). The DBpedia links for Models 1 and 2 reported in Table 4 are based on training data. These models achieve better precision values than the state-of-the-art systems. In Model-3, the links are retrieved from the DBpedia database. The performance of Model-3 is better than the first and second models. The recall, precision and F-measure values of Model-3 on the test data are 26.9%, 67.1% and 38.4%.

The Combined Model performs best among our models and produces an F-measure value of 42.1%. All these models outperform the NEEL 2016 systems in terms of precision, but fail get

Strong_typed_mention_match	Training data						Training + Dev Data		
	Dev Data			Test Data			Test Data		
	R	P	F_1	R	P	F_1	R	P	F_1
Baseline Model	36.2	69.9	47.7	23.7	50.7	32.3	55.6	69.7	61.9
Model-1	38.8	40.7	39.8	32.8	33.8	33.3	64.5	66.3	65.4
Model-2	27.9	**87.2**	42.3	27.8	**86.9**	42.2	49.3	**91.0**	64.0
Model-3	67.4	86.7	75.8	50.8	76.1	60.9	52.5	76.6	62.3
Combined Model	**75.3**	79.0	**77.1**	**65.3**	67.1	**66.2**	**69.7**	71.7	**70.7**
KEA (Waitelonis and Sack, 2016)	66.0	57.2	61.3	49.4	45.3	47.3	-	-	-
MIT Lincoln Lab (Greenfield et al., 2016)	28.7	58.7	38.6	24.0	47.4	31.9	-	-	-
JU (Ghosh et al., 2016)	35.3	41.1	38.0	28.9	33.8	31.2	-	-	-
UniMiB (Caliano et al., 2016)	17.8	54.5	26.8	18.8	46.2	26.7	-	-	-
Insight-centre@NUIG (Torres-Tramón et al., 2016)	35.5	33.4	34.4	24.2	24.9	24.6	-	-	-

Table 3: Twitter Named Entity Classification results ('Baseline Model': using all features)

Strong_link_match	Training Data						Training + Dev Data		
	Dev Data			Test Data			Test Data		
	R	P	F_1	R	P	F_1	R	P	F_1
Baseline Model	18.4	85.5	30.3	13.5	**75.0**	22.8	46.9	74.4	57.5
Model-1	35.3	**90.9**	50.8	26.1	67.8	37.7	54.2	**78.1**	64.0
Model-2	29.0	90.2	43.9	19.2	65.9	29.7	49.3	77.4	60.2
Model-3	43.1	87.3	57.7	26.9	67.1	38.4	52.4	77.5	62.6
Combined Model	44.7	87.0	59.1	30.4	68.5	42.1	**57.2**	**78.1**	**66.0**
KEA (Waitelonis and Sack, 2016)	**86.2**	66.7	**75.2**	**56.0**	45.4	**50.1**	-	-	-
MIT Lincoln Lab (Greenfield et al., 2016)	41.8	79.9	54.9	28.5	64.6	39.6	-	-	-
JU (Ghosh et al., 2016)	16.1	58.6	25.2	21.7	29.0	24.8	-	-	-
Insight-centre@NUIG (Torres-Tramón et al., 2016)	32.4	49.1	39.0	16.7	25.7	20.2	-	-	-
UniMiB (Caliano et al., 2016)	38.7	45.2	41.7	13.9	35.4	16.2	-	-	-

Table 4: Twitter Named Entity Linking results ('Baseline Model': using all features)

a higher F-score than the KEA system (Waitelonis and Sack, 2016) due to lower recall. However, when development data are merged with the training data for the model building, the Combined Model again gives the best test data results among all the systems, achieving recall, precision and F-measure values of 57.2%, 78.1% and 66.0%.

6 Discussion

The task is divided into two parts: identification of Twitter named entities, and classification and linking to DBpedia. For the NE identification, a Conditional Random Fields classifier is used in a Differential Evolution framework. DE selects features randomly and lets the CRF classifier train on these features in order to optimize the feature selection based on fitness values on the development data. Hence, the CRF classifier runs many times based on the size of the population and the number of generation given as parameters to the Differential Evolution. So the learning time depends on the training data size, number of features, number of class, etc. Here, the training time is reduced since the training data size, number of features, and number of classes are small. To increase the speed, a multi-threading approach was also used for CRF learning.

When running the CRF classifier for the identification task using all the features described in Section 3.3, the performance (baseline model) is low because the model fails to efficiently retrieve Twitter names. To enhance the performance on the identification task, the DE setting was used, since it identifies near optimal features based on its search capability.

The system output was analysed in order to understand the nature of the errors encountered: a significant number of entities were not correctly detected, resulting in low performance. A closer look reveals that many misclassifications are caused by common word(s) in both query and document. For example, 'LiberalPhenom' is annotated as a person name, but the system identifies it as a organization since the 'LiberalPhenom' (split as 'Liberal Phenom') query is closer to the document 'Liberal Forum'. In many cases, the system classifies an entity as a Twitter name even though it is not considered as an entity in the gold

standard annotation. For example, 'InAbuDhabi' (split as'In Abu Dhabi' which is closer to 'Abu Dhabi') is tagged as a location, but in gold annotation it is not considered to be a Twitter name.

The output of the entity linking to DBpedia was also analysed. The linking performance is not good because many entities with links are not in the DBpedia database. For example, 'TheForceAwakens' (split as 'The Force Awakens') is correctly identified as a Twitter named entity by the system, but that name along with a link is not in DBpedia (Release 2014). Another cause of errors is that many entities and their corresponding links simply are not identified at all by the system.

7 Conclusion

In this paper, we propose a system for Twitter named entity extraction and linking to DBpedia. Twitter names are first identified from the noisy texts using a multi-objective technique based on Differential Evolution and a range of different features. The most important features are identified along with their context features. In a second step, the identified Twitter named entities are classified and linked to the DBpedia database. For classification of Twitter names, a vector space model and edit distance techniques are used, achieving results outperforming both a tough baseline model and other state-of-the-art systems.

In the future, we will experiment with other datasets and try to collect more examples along with contextual information to reduce misclassifications. It would also be interesting to apply the approach described here to tweet named entity extraction for under-resourced languages, such as the languages of India.

Furthermore, we will develop other entity extraction models to perform ensemble classification using a multi-objective approach, and experiment with tree-based structured learning (Yang and Chang, 2015) for the task of linking Twitter named entities to DBpedia.

Acknowledgements

We are most grateful for some very useful comments from several anonymous reviewers, that helped to substantially improve the paper.

Many thanks also to the NEEL 2016 shared task organisers (Rizzo et al., 2016) for collecting, annotating and publishing the data.

References

Md Shad Akhtar, Utpal Kumar Sikdar, and Asif Ekbal. 2015. IITP: Multiobjective differential evolution based Twitter named entity recognition. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 106–110, Beijing, China, July. ACL. Workshop on Noisy User-generated Text.

Timothy Baldwin, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 126–135, Beijing, China, July. ACL. Workshop on Noisy User-generated Text.

Davide Caliano, Elisabetta Fersini, Pikakshi Manchanda, Matteo Palmonari, and Enza Messina. 2016. UniMiB: Entity linking in tweets using Jaro-Winkler distance, popularity and coherence. In *Proceedings of the 6th Workshop on Making Sense of Microposts (#Microposts2016)*, pages 60–72, Montréal, Canada, April.

Kalyanmoy Deb. 2001. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester, England.

Souvick Ghosh, Promita Maitra, and Dipankar Das. 2016. Feature based approach to named entity recognition and linking for tweets. In *Proceedings of the 6th Workshop on Making Sense of Microposts (#Microposts2016)*, pages 74–76, Montréal, Canada, April.

Kara Greenfield, Rajmonda Caceres, Michael Coury, Kelly Geyer, Youngjune Gwon, Jason Matterer, Alyssa Mensch, Cem Sahin, and Olga Simek. 2016. A reverse approach to named entity extraction and linking in microposts. In *Proceedings of the 6th Workshop on Making Sense of Microposts (#Microposts2016)*, pages 67–69, Montréal, Canada, April.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, Williamstown, Maryland, USA, June.

Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. 2012. TwiNER: Named entity recognition in targeted Twitter stream. In *Proceedings of the 35th International Conference on Research and Development in Information Retrieval*, pages 721–730, Portland, Oregon, August. ACM SIGIR.

Nut Limsopatham and Nigel Collier. 2016. Bidirectional LSTM for named entity recognition in Twitter messages. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 145–152, Osaka, Japan, December. ACL. 2nd Workshop on Noisy User-generated Text.

Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 359–367, Portland, Oregon, June. ACL.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 248–256, Singapore, August. ACL.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, August. ACL.

Giuseppe Rizzo and Marieke van Erp. 2016. Named Entity rEcognition and Linking (NEEL) challenge. `http://microposts2016.seas.upenn.edu/challenge.html`.

Giuseppe Rizzo, Marieke van Erp, Julien Plu, and Raphaël Troncy. 2016. Making sense of microposts (#Microposts2016) Named Entity rEcognition and Linking (NEEL) challenge. In *Proceedings of the 6th Workshop on Making Sense of Microposts (#Microposts2016)*, pages 50–59, Montréal, Canada, April.

Gerard Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, November.

Utpal Kumar Sikdar and Björn Gambäck. 2016. Feature-rich Twitter named entity recognition and classification. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 164–170, Osaka, Japan, December. ACL. 2nd Workshop on Noisy User-generated Text.

Rainer Storn and Kenneth Price. 1997. Differential Evolution — a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, December.

Benjamin Strauss, Bethany E. Toma, Alan Ritter, Marie Catherine de Marneffe, and Wei Xu. 2016. Results of the WNUT16 named entity recognition shared task: Twitter lexical normalization and named entity recognition. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 138–144, Osaka, Japan, December. ACL. 2nd Workshop on Noisy User-generated Text.

Pablo Torres-Tramón, Hugo Hromic, Brian Walsh, Bahareh R. Heravi, and Conor Haye. 2016. Kanopy4Tweets: Entity extraction and linking for Twitter. In *Proceedings of the 6th Workshop on Making Sense of Microposts (#Microposts2016)*, pages 64–66, Montréal, Canada, April.

Twitter. 2016. Company information, June. `https://about.twitter.com/company`.

Jörg Waitelonis and Harald Sack. 2016. Named entity linking in #tweets with KEA. In *Proceedings of the 6th Workshop on Making Sense of Microposts (#Microposts2016)*, pages 61–63, Montréal, Canada, April.

Ikuya Yamada, Hideaki Takeda, and Yoshiyasu Takefuji. 2015. Enhancing named entity recognition in Twitter messages using entity linking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 136–140, Beijing, China, July. ACL. Workshop on Noisy User-generated Text.

Yi Yang and Ming-Wei Chang. 2015. S-MART: Novel tree-based structured learning algorithms applied to tweet entity linking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 504–513, Beijing, China, July. ACL.