EAI Endorsed Transactions
on Scalable Information Systems

Research Article **EAI.EU**

Word Embedding and String-Matching Techniques for Automobile Entity Name Identification from Web Reviews

Satanu Maity[1], Nilanjana Das[2], Mukta Majumder[3,*] and Dibya Ranjan Dasadhikary[4]

[1]Department of Computer Application, Bengal School of Technology and Management, Hooghly, India
[2]Midnapore Zone, WBSEDCL, Midnapore, India
[3]Department of Computer Science and Application, University of North Bengal, Siliguri, India
[4]Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India

Abstract

With the huge popularity of Internet, various types of information on a wide range of domains are floating over different social media platforms. To extract this information for using in diverse natural language processing applications, identifying the names is prerequisite. A study is presented here, to identify automobile names from noisy web reviews by exploring two widely used machine learning algorithms, Conditional Random Field and Support Vector Machine. The accuracy of machine learning classifiers radically rely on size and quality of training data which has been prepared manually by extracting discussion forum corpus; the task is time consuming and laborious; hence to leverage this word embedding is adopted. Though it enhances the system's performance but is unable to spot noisy names which occur in web reviews. Next, a gazetteer based string matching technique is proposed, it recognizes a new set of noisy automobile entities, resulting considerable improvement in accuracy.

Keywords: Noisy Name Identification, Automobile Discussion Forum, Machine Learning, Support Vector Machine, Conditional Random Field, Word Embedding, String Matching.

Received on 05 January 2021, accepted on 28 April 2021, published on 14 May 2021

doi: 10.4108/eai.14-5-2021.169918

[*] Corresponding author. Email: mukta_jgec_it_4@yahoo.co.in

1. Introduction

The smallest token in a corpus that conveys the most information, is an entity name. Naturally identifying this name becomes the ultimate choice for mining and extracting information from text. Named entity recognition (NER) system finds usability in a number of natural language processing (NLP) tasks like text classification, opinion mining, information retrieval, sentiment analysis, machine translation etc.

The literature has demonstrated that newsgroup, discussion forums and social media data have been used as sources of information bank in a number of NLP systems [1-6]. We have also considered the Internet as our data source for developing the NER system. There are plenty of discussion and review forums available in the Internet, like automobile discussion forum[2], diagnostic discussion forum[3], reviews on movie[4], mobile[5], any newly launch software or product[6], life insurance policy[7], and discussion forum on any recent news or trends[8] etc.

[2] https://www.zigwheels.com/
[3] https://abchomeopathy.com/
[4] https://www.imdb.com/
[5] https://www.gsmarena.com/
[6] https://www.capterra.com/
[7] https://www.bankbazaar.com/
[8] https://twitter.com/

1.1 Observation and Motivation

We have observed that to most of the common web users, these types of Social Media (web reviews or discussion forums) have become centre of attraction. Exploring the web reviews and discussion articles has emerged as a regular practice to these people before watching a movie or buying a car, bike, mobile, laptop or any other product. These people are accustomed to share information and their experiences in various aspects of daily life on the Internet through these types of Social Media. As a consequence, such corpus is full of customer to customer (C2C) message, business to business (B2B) conversation, business to customer (B2C) communication and vice-versa. Corporate giants and large business firms have identified that, the customer not only delivers profit by business transaction but also enrich product reviews in these Social Media platforms. By analysing these reviews, demand supply study, market trend estimation can be done if the targeted products and their companies (names of product and company etc.) can be identified. Hence, this requires a named entity identification system for the target domain.

Purchasing a family car is quite common these days. While purchasing the car, knowing its features is essential for the buyer. On the Internet there are also several web reviews and discussion forums available on automobiles; for example, CarWale[9], CareDekho[10], ZigWheels, CarTrade[11] etc. from where customer can get information about the features and other details of a car. One can also write reviews or post any query about a particular model of a car in those forums. Hence these web reviews are the valuable sources of experiences shared by existing users of automobiles. An automatic car recommender system based on buyer requirement and query can be developed if these discussions corpora available on the Internet can be used skilfully. To use these corpora in a variety of tasks like demand supply study, market trend estimation and other information extraction (IE) tasks, recognizing the named entities is prerequisite as these are the pivotal element of the corpus. And these corpora contain the primary named entities like: company, product and model names of a car. These observations motivate us to design a NER system in automobile domain.

1.2 Problem Formulation and Solution Strategy

But these corpora available in these web reviews and discussion forums are posted by common web users; hence these often contain a high amount of noisy text. NLP tools which are developed for grammatical and standard corpus often fall short to produce a fair performance on these corpora due to the noises. The intricate and uncertain naming style (lxi, vxi, zxi, etc.) of these car names and product names are also a major setback of this task. The misspellings and abbreviations are the other ambiguities to identify these automobile names. A specific name of a car can be spelled differently by various users; for example, 'Hyundai' is written as 'Hyndai', 'Hundayi', 'Hundai' etc. Sometimes these names also contain numeric values (I10, I20, V2, eV2, etc); which specify product and model of the car. Moreover, the uneven capitalization and punctuation of discussion forum text raise the difficulties; hence dedicated special system is required to handle the automobile noisy names.

For developing a NER system three types of techniques can be adopted: Linguistic, Machine Learning (ML) and Hybrid. The first one relies on the principle of handcrafted rules and requires domain expert [7]; hence difficult to follow for complex named entities [8-10]. For this reason, supervised ML based NER system is a preferable option. A ML classifier uses labelled training data where names are annotated manually [11]. A combination of these two techniques, called Hybrid approach is also used for named entity identification [12].

This paper explores multiple machine learning classifiers, conditional random field and support vector machine for automobile name identification from online discussion or user review corpus. The two classifiers are trained using a manually annotated data set (~105K Words) taken from "http://www.carwale.com/" user reviews. The authors have considered three named entity (NE) categories; company, product and model. To assess the robustness of the system, the testing experiment is performed on a data set taken from a different source CarDekho user review ("https://www.cardekho.com/"). Here the baseline Conditional Random Field (CRF) system obtains an F-Measure of 92.20 and the baseline Support Vector Machine (SVM) system achieves an F-Measure of 93.24.

The ML algorithm uses the training samples from annotated corpus to build a statistical classifier. Thus, the system majorly relies on the quantity and quality of training corpus. The manual preparation of quantitatively large labelled training corpus with full of enrich feature samples, is laborious and burdensome. On the other hand, a large collection of external corpora (raw), containing certain valuable information is well available that can leverage the labelled training corpus. Though efficient processing of raw data is a prerequisite; so that, only the informative parts have sufficient impact on the outcomes [13]. Here comes the importance of word embedding. Word embedding is a technique for generating word vectors using skip-gram and continuous bag-of-words (CBOW) model [14]. Word2vec[12] has been used for creating word vectors to enhance the performance of baseline system. The word embedding modification of CRF based system achieves an F-Measure of 94.09. And SVM system with word embedding based enhancement reaches an F-Measure of 94.93.

During the analysis, it has been observed that a few misspelled or abbreviated (noisy) names remain undetected

[9] https://www.carwale.com/
[10] https://www.cardekho.com/
[11] https://www.cartrade.com/
[12] https://code.google.com/archive/p/word2vec/

by the word embedding based enhancements of CRF and SVM models. To recognize these misspelled or abbreviated automobiles NEs, an automobile gazetteer list has been prepared and a novel gazetteer-based string-matching approach has been proposed. Finally, the CRF system by incorporating word embedding and string-matching technique achieved an F-Measure of 95.87 and the SVM system with word embedding and string matching accomplishes an F-Measure of 96.40.

The salient features of this article which contributes to the literature in several ways are as follows:

- The proposed NER system explores multiple machine learning classifiers and is an initiative (first NER system in automobile domain) for identifying entity names from automobile web reviews.
- The article proposes a word embedding based semi-supervised learning framework.
- It introduces a novel gazetteer-based string-matching technique to achieve higher accuracy by identifying the misspelled or abbreviated (noisy) entity names from web discussion corpus.

Rest part of the paper is subdivided into following five sections. Related previous works are presented in Section 2. The baseline NER system is discussed in Section 3. Word Embedding enhancement of the system is presented in Section 4. Section 5 represents the novel gazetteer-based string-matching approach for noisy name detection. And the conclusion is drawn in Section 6.

2. Related Work

It has been observed in the literature that most of the NER tasks which are available, majorly concentrated on identifying general domain names, like person, organization and location etc. Some specific systems are also available which work in identifying NEs like chemical, historical and biomedical etc.

Here, we have mentioned some of the NER systems which used supervised learning techniques. In the literature it was observed that for identifying names, different machine learning classifiers were applied by the researchers, like Hidden Markov Model [15-20], Maximum Entropy [21-25], Conditional Random Field [26-29] and Support Vector Machine [30-34]. We also found some NER systems which used multiple machine learning classifiers for identifying and classifying named entities [35-37].

To boost up the performance of a baseline system (model), various types of integrations or add-on modules have been incorporated [38]. Various NER systems which incorporated deep domain information likes Part-of-speech (POS), word pattern, out of domain POS, morphological pattern, semantic trigger, etc. for identifying names were [39-40]. Lin et al. proposed a Maximum Entropy based hybrid NER system which was combined with a rule-based technique [22]. Ekbal and Bandyopadhyay developed a NER system for Bengali corpus using three different classifiers Maximum Entropy, Conditional Random Field and Support Vector Machine [35]. Their system also used semi-supervised learning and weighted voting approaches as post-processing techniques. Poibeau proposed a hybrid NER system and incorporated multiple criteria-based techniques for boosting the robustness of the named entity recognizer [41].

Use of machine learning-based approach with word embedding features for the Disease named entity reorganization and normalization subtask of the BioCreative-V Chemical Disease relation (CDR) challenge task was found in [42]. The result of Russian named entity classification and equivalent NE retrieval using word-phrase representation was described by Ivanitskiy et al. [43]. They described that a word or an expression's context vector came out as an effective attribute to be used for guessing the type of a NE class. Seok et al. used CRF as a learning algorithm and applied word embedding feature for NE extraction purpose [44]. A few other NER tasks which used word embedding for identifying names were [45-47]. There are multiple embedding techniques like 'Word2Vev', and 'GloVe[13]' etc.

A good number of research works have been found for extracting names from informal web text. We have found a NER system that used a web search engine and a NE list for collecting web documents containing NE instances [48]. These web documents were further filtered out by sentence separation and text refinement procedures for finally classifying the NEs in appropriate classes. A novel n-gram based lexical system was proposed by Downey et al. for identifying complex NEs from the online corpus [49]. Their system incorporated Point wise Mutual Information (PMI) and Symmetric Conditional Probability (SCP) scores in the lexical method. Ritter et al. presented a T-NER for extracting NEs from the Twitter text [50]. They applied topic modelling, Labelled LDA to enhance the performance of their system. Another NER system was introduced by Karaa for extracting and categorizing NEs from web corpus [51]. To extract the context of the NEs Karaa used tfidf (term frequency and inverse document frequency). Majumder et al. used CRF as ML classifier along with active learning based semi-supervised approach for drug and disease names identification from discussion forum corpus [3]. But their NER system unable to recognize noisy named entities. In another attempted Majumder and Saha applied CRF as classification algorithm along with global context-based framework to track misspelled and abbreviated disease and drug names from healthcare web review corpus [4]. Aguilar et al. proposed a multi-task approach for named entity recognition from social media data using Part-of-Speech tags and gazetteer information [52]. They used CRF as classifier for their NER task. Singh et al. presented a named entity recognition system for Hindi-English code-mixed social media text (twitter) using word, character and lexical features [53]. Sabty et al. proposed a NER system for identifying NEs from Arabic-English Code-Mixed Data

[13] https://nlp.stanford.edu/projects/glove/

[54]. Performance of this system was enhanced by using a pooling technique and word embedding.

In the literature, any NER system is hardly found which works in the automobile domain for identifying car company, product and model names. Most of the NER systems discussed above also found difficulty in handling noisy NEs. Our proposed system is an initiative for identifying and classifying automobile names from noisy online user review or social media corpus.

3. NER system using CRF and SVM

Two ML classifiers conditional random field and support vector machine have been explored for developing the baseline NER system.

3.1. Conditional Random Field (CRF)

CRF is a conditional probabilistic model for annotating or labelling and classifying sequential data like natural language corpus [55]. This is an undirected graphical framework which illustrates a single log-linear allocation on the annotated sequence while given a particular observation sequence [56].

For this NER system development, the toolkit CRF++[14] has been used. It is a simple and customizable open-source executable model of CRF for segmenting or classifying sequential data such as natural language text. CRF++ is developed for general purpose and useful to a wide range of NLP applications, like Part-of-speech tagging, NER and opinion mining, etc.

3.2. Support Vector Machine (SVM)

SVM is a widely used supervised ML classifier introduced by Vapnik [57]. The classification problem generally engages in training and testing of data set that involves some data samples.

As the SVM is a binary classifier; the pair-wise or one-vs-rest approach is used for multi-class classification. To develop the proposed NER system SVM is used as one of the machine learning classifiers which carries out classification by building an N-dimensional hyperplane which best possibly splits data into two categories, positive class and negative class. This NER task involves two main phases: training and classification, which have been performed by YamCha[15], an open source and customizable implementation of SVM, widely used in a variety of NLP tasks, like sentiment analysis, NER and other sequential labelling problems.

The SVM toolkit, YamCha supports kernel function. For the proposed NER system development, 2nd degree of the polynomial kernel has been deployed for SVM.

3.3. The Set of Features Used to Train the Baseline Model

The feature value plays a key role in developing a machine learning based NER system. Various types of potential features have been explored and best suited feature values have been chosen for developing the baseline system. The system is trained with the labelled dataset by using several combinations of following candidate features.

Word Features

Word feature is most important to develop the NER system. The current word along with proceeding and following words have been used with a window size of three, five and seven, where the target word is at the middle.

Affix Feature

The Morphological information, like prefix and suffix, are regarded as important cues for terminology classification. We have used prefix and suffix of length three.

Digit Feature

In the corpora, it is found that often a current token/word is containing a numeric value (digit). Hence the feature like Is_Numerical or not is included.

Capitalization Feature

The NEs are often capitalized in a standard corpus. Binary feature like current token starts with a capital letter or not is used here.

Parts-of-speech (POS) Feature

Part-of-speech (POS) information is a key attribute for NER task. POS information of target word along with the surrounding words has been extracted using NLTK (Natural Language Processing Toolkit) POS tagger for this purpose.

Dictionary Feature

Named entity normally does not belong to dictionary, so we use a binary feature that the current word is a dictionary word or not.

Symbolic Feature

The general rule is that a sentence ends with only one terminal punctuation symbol like the period (.), the question mark (?), and the exclamation point (!). There are some complex/compound words (Name Entity like “I-10”) consist many symbols. Naturally, this feature has a great impact on accuracy.

Word Shape Feature

In the system, this attribute stands for the brief word class of the current token. The consecutive upper-case letters, lower case letters, digits, symbols are mapped to ‘X’, ‘x’, ‘0’, and ‘#’ respectively. For example, the brief word shape of “I-10” is “X#00”. The feature has a wider effect in identifying the NEs from the noisy text of web reviews.

[14] https://taku910.github.io/crfpp/

[15] http://chasen.org/~taku/software/yamcha/

Table 1. Result of Baseline CRF Based NER System on Dataset

Features used in CRF Baseline System	Word Window 7			Word Window 5			Word Window 3		
	Precision	Recall	F-Score	Precision	Recall	F-Score	Precision	Recall	F-Score
Only Word Feature	95.17	75.75	84.36	95.20	77.21	85.27	94.37	78.68	85.81
AFFIX	92.86	90.40	91.61	92.14	90.40	91.26	92.52	90.77	91.64
AFFIX-DIGT	93.22	90.40	91.79	92.86	90.40	91.61	92.88	90.77	91.81
AFFIX-POS	92.13	90.04	91.07	91.77	90.04	90.90	92.89	91.13	92.00
AFFIX-POS-CAP	92.13	90.04	91.07	91.79	90.40	91.09	92.53	91.13	91.82
AFFIX-POS-CAP-DIC	91.79	90.40	91.09	92.13	90.04	91.07	92.50	90.40	91.44
AFFIX-POS-SYM-CAP	92.48	90.04	91.24	91.77	90.04	90.90	92.84	91.13	91.98
AFFIX-POS-SYM-DIGT	92.00	90.40	91.19	91.79	90.40	91.09	92.55	91.50	92.02
AFFIX-POS-SYM-DIGT-CAP-SHAP	92.50	90.40	91.44	92.18	91.13	91.65	**92.91**	**91.50**	**92.20**
AFFIX-SYM-DIGT-CAP-SHAP	92.50	90.40	91.44	92.13	90.04	91.07	92.52	90.77	91.64
AFFIX-SYM-SHAP-DIC	92.14	90.40	91.26	92.14	90.40	91.26	92.50	90.40	91.44

3.4. Data Set Used in the Baseline NER System

To train the baseline NER system the data set has been taken from “Carwale” (http://www.carwale.com/) user reviews. The training data set contain ~105K words having ~4436 annotated NEs. Three name classes: Company (C), Product (P) and Model (M) have been considered here. The data set is annotated as follows:

“I ‘#O’ am ‘#O’ the ‘#O’ proud ‘#O’ owner ‘#O’ of ‘#O’ the ‘#O’ first ‘#O’ Hyundai ‘#BCNE’ i10 ‘#BPNE’ automatic ‘#O’ transmission ‘#O’ in ‘#O’ the ‘#O’ city ‘#O’ of ‘#O’ Chandigarh ‘#O’ 1.2L ‘#BMNE’, kappa ‘#CMNE’ variant ‘#O’ with ‘#O’ Oyster ‘#O’ Grey ‘#O’ colour ‘#O’”.

Where BCNE, BPNE and BMNE refer to beginning of company, product and model names respectively and CMNE refers to continuation of model name.

To test the robustness of the system, a data set containing ~10K words has been taken as test data from a different source, "CarDekho" (https://www.cardekho.com/) user reviews and tested it with the baseline system.

3.5. Performance of Machine Learning Based NER System

Table 1 and Table 3 have presented the accuracies obtained in different stages of baseline classifier preparation. The accuracy is calculated as F-Measure or F-Score (F) which is harmonic mean of recall and precision. Recall is the ratio of the number of relevant named entities retrieved to the total number of relevant named entities in the corpus whereas Precision is the ratio of the number of relevant named entities retrieved to the total number of irrelevant and relevant named entities retrieved. Both recall and precision are usually expressed as percentage. The calculation of Precision, Recall and F-Measure are shown below along with the Confusion Matrix in Table 2.

Table 2. Confusion Matrix

		Actual	
		Yes	No
Predicted	Yes	TP (True Positive)	FP (False Positive)
	No	FN (False Negative)	TN (True Negative)

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP+FN}. \quad (2)$$

$$\text{F} = \frac{(1+\beta^2)(\text{precision} \times \text{recall})}{(\beta^2 \times \text{precision}+\text{recall})} \quad (3)$$

‘β’ is relative weight between recall and precision, whose value is considered as 1.

The baseline CRF system achieves an F-Measure of 92.20 with 92.91% as precision and 91.50% as recall in word windows 3 and the baseline SVM system reaches an F-Measure of 93.24 with 95.71% as precision and 90.89% as recall in word windows 3. The result of the baseline system implies that some NEs are not detected by the system. This may be due to the quantity (size of training data: 105K words) and the quality (noisy text) of training corpus. To leverage this and to get better accuracy from the system, Word Embedding based semi-supervised framework has been adopted.

Table 3. Result of Baseline SVM Based NER System on Dataset

Feature used in SVM Baseline System	Word Window 7			Word Window 5			Word Window 3		
	Precision	Recall	F-Score	Precision	Recall	F-Score	Precision	Recall	F-Score
Only Word Feature	95.21	77.55	85.48	94.66	83.78	88.89	96.80	88.54	92.49
AFFIX-DIC	94.94	80.12	86.90	95.05	83.41	88.85	95.27	88.90	91.97
AFFIX-CAP	95.68	78.65	86.33	95.46	83.78	89.24	95.66	89.27	92.35
AFFIX-SHAP	94.92	80.55	87.15	94.85	82.75	88.39	95.58	90.24	92.83
AFFIX-DIGT-CAP	95.41	82.31	88.38	94.35	82.75	88.17	96.25	89.28	92.63
AFFIX-SYM-DIGT	95.51	79.73	86.91	95.67	83.21	89.01	95.83	89.38	92.49
AFFIX-POS-DIGT-CAP	94.92	80.48	87.11	95.49	84.51	89.67	95.69	89.24	92.35
AFFIX-SYM-DIGT-CAP	95.93	80.35	87.45	94.92	85.31	89.86	96.07	90.37	93.13
AFFIX-POS-SYM-DIGT-SHAP	95.35	80.81	87.48	95.96	86.71	91.10	**95.71**	**90.89**	**93.24**
AFFIX-POS-SYM-DIGT-SHAP-DIC	95.68	78.65	86.33	94.66	83.78	88.89	95.58	90.24	92.83
AFFIX-POS-SYM-DIGT-CAP-SHAP	95.35	79.33	86.61	96.10	84.84	90.12	96.03	90.58	93.23

4. Word Embedding

Word Embedding is the mapping of words with the vectors of actual numbers, helping to obtain enhanced performance in different NLP tasks like NER; sentiment analysis etc. by grouping similar words [29, 45, 46, 58-59]. It can capture a variety of dimensions of meaning and phrase information related to the prospective attributes of words within a vector. The vectors of the words depend on vocabulary size. The words, which have similar vectors, must be semantically similar. The semantically analogous like synonyms, antonyms, on a scale (like hot, warm, cool), etc. It is used in semantic parsing for digging out the meaning from text to enable natural language understanding. For a language model to predict the meaning of a text, it is necessary to be conscious of the contextual similarity of words. These vectors are numerical representations of contextual similarities between words and can be operated mathematically/logically just like other vectors. Generally, this word similarity is calculated by cosine distance of embedded vectors.

Assume data is embedded in three-dimensional space, such as small bubble represents word entities; blue circle symbolizes 'company name' (Hyundai, Maruti, Ford, etc.), red circle indicates 'product name' (i10, Figo, Alto etc.), and yellow circle denotes 'model name' (LXI, Magna, etc.) as shown in Figure 1. Here embedding is formed with neighbouring words which are closer in term of cosine distance between them (like-Hyundai and Maruti: both are company name) as well as have similar distance from their co-related words of the other group (i10 and Alto respectively). For example, the distance between Hyundai and i10 is similar to the distance between Maruti and Alto, thus Hyundai - i10 and Maruti - Alto are co-related word pair. As Hyundai and Maruti are closer in term of cosine distance and have the same cosine distance with their co-related words (i10 and Alto respectively), so they are group with each other. In this approach, the word embedding has been established by using Word2Vec.

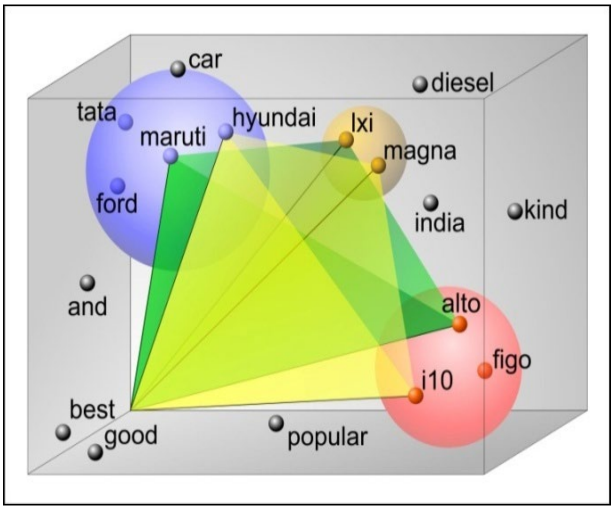

Figure 1. Example of word embedding

Word2Vec is a linguistic representation of a neural network that learns the embedding of every word in the corpus. Word2Vec offers skip-gram architecture along with continuous bag-of-words (CBOW) model, proposed by Mikolov et al. at Google [11]. Skip-gram forecasts the neighbouring words or context where a single word is given. The size of every word vector is 50 and the window dimension for context information has been set as 4 words preceding and 4 words following according to the current word. Another ~135K extra words have been collected from automobile discussion corpora and added to the original data set of ~105K words. These ~240K words have been used to create word vectors.

After using word embedding technique considerable improvements have been seen in the successive versions of CRF and SVM based NER systems. The results of these two NER systems are shown in Table 4 and Table 5. After word embedding enhancement, CRF based system reaches an F-Measure of 94.09 with 94.62% as precision and 93.57% as recall and SVM based system accomplish an F-Measure of 94.93 with 96.57% as precision and 93.34% as recall.

A comparative study has also been done between the baseline NER Systems with their word embedding based enhancements, shown in Figure 2. After analysing these results, it is found that the SVM and word embedding based NER system outperformed the CRF and word embedding based system. Although after using word embedding technique the performance of both versions of the system have increased, still a few NEs remain undetected as they are noisy (misspelled or abbreviated). To trace these misspelled or abbreviated automobile NEs, an automobile gazetteer list has been prepared and a novel gazetteer-based string-matching approach has been proposed, illustrated in the next section.

Table 4. Result of CRF Based NER System with Word Embedding

Features in CRF System with W2V	Word Window 7			Word Window 5			Word Window 3		
	Precision	Recall	F-Score	Precision	Recall	F-Score	Precision	Recall	F-Score
AFFIX-W2V	93.84	92.10	92.96	94.58	92.47	93.51	93.86	92.47	93.16
AFFIX-POS-W2V	93.45	91.37	92.40	93.49	92.10	92.79	94.25	93.20	93.72
AFFIX-POS-DIGT-CAP-SHAP-W2V	93.12	91.74	92.42	94.25	93.20	93.72	93.92	93.57	93.74
AFFIX-POS-SYM-DIGT-W2V	93.84	92.10	92.96	93.22	92.47	92.84	94.62	93.57	94.09
AFFIX-POS-SYM-DIGT-CAP-SHAP-W2V	93.12	91.74	92.42	94.59	92.83	93.70	93.92	93.57	93.74
AFFIX-SHAP-W2V	93.49	92.10	92.79	94.23	92.83	93.52	93.86	92.47	93.16
AFFIX-SYM-CAP-SHAP-W2V	94.20	92.10	93.14	94.59	92.83	93.70	93.88	92.83	93.35
AFFIX-SYM-DIGT-CAP-W2V	94.58	92.47	93.51	94.96	93.20	94.07	94.61	93.20	93.90
AFFIX-SYM-DIGT-CAP-SHAP-W2V	94.59	92.83	93.70	94.62	93.57	94.09	94.62	93.57	94.09
AFFIX-SYM-SHAP-DIC-W2V	93.84	92.10	92.96	94.59	92.83	93.70	93.86	92.47	93.16

Table 5. Result of SVM Based NER System with Word Embedding

Features in SVM System with W2V	Word Window 7			Word Window 5			Word Window 3		
	Precision	Recall	F-Score	Precision	Recall	F-Score	Precision	Recall	F-Score
AFFIX-DIC-W2V	96.18	76.85	85.44	95.06	81.62	87.83	96.15	91.14	93.58
AFFIX-POS-DIGT-W2V	94.59	80.15	86.77	95.21	85.64	90.17	95.80	92.24	93.99
AFFIX-CAP-W2V	95.46	81.25	87.78	94.88	87.11	90.83	96.55	92.24	94.35
AFFIX-DIGT-W2V	95.06	81.62	87.83	95.25	86.74	90.80	96.56	92.97	94.73
AFFIX-SYM-W2V	95.84	80.15	87.30	95.27	87.11	91.01	96.18	92.60	94.36
AFFIX-CAP-SHAP-W2V	94.65	81.62	87.65	94.84	86.01	90.21	96.56	92.60	94.54
AFFIX-SYM-CAP-SHAP-W2V	94.40	84.91	89.40	95.70	88.57	92.00	96.56	92.97	94.73
AFFIX-POS-SYM-DIGT-CAP-SHAP-W2V	95.17	84.55	89.55	95.71	88.94	92.20	96.57	93.34	94.93
AFFIX-SYM-DIGT-CAP-W2V	94.79	84.91	89.58	95.69	88.21	91.80	95.81	92.60	94.18
AFFIX-SYM-DIGT-CAP-SHAP-W2V	95.25	86.74	90.80	95.71	88.94	92.20	96.19	92.97	94.55

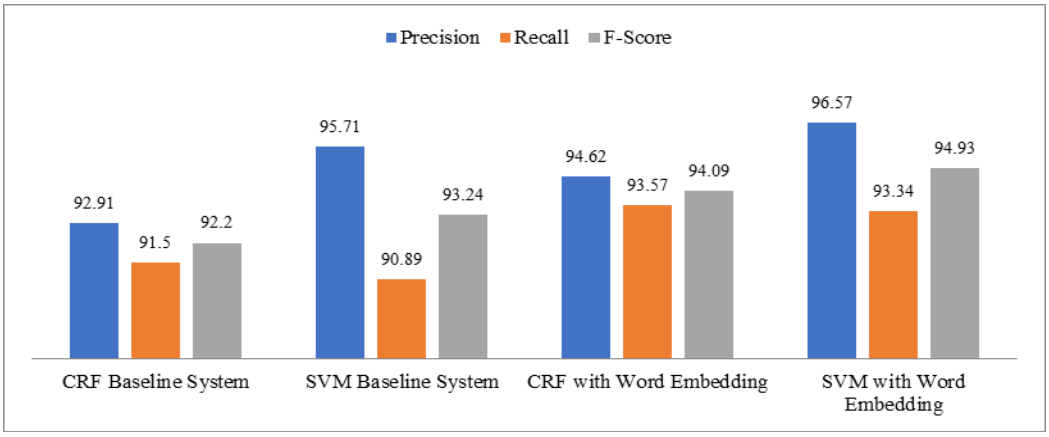

Figure 2. Comparison of CRF and SVM based NER System after Word Embedding

5. String-Matching Based Noisy Name Identification

Recognizing named entity from online user review or social media corpus faces the major setback from its noisy nature. The texts available in these social network platforms or discussion forums are posted by common web users; consequently, these often include a large amount of textual noises. A specific car NE can be spelled differently by different users; for example, ‘Hyundai’ is written as ‘Hyndai’, ‘Hundayi’, ‘Hundai’ etc. Moreover, the use of punctuation in web review corpus does not follow a standard grammatical rule. As the NEs of this corpus are often misspelled and abbreviated, traditional NLP techniques are unable to detect them; hence dedicated special system is required to handle the noisy names. A novel gazetteer-based string-matching algorithm (Algorithm 1) has been proposed to classify these noisy NEs.

First, a gazetteer list (G_LIST) of automobile named entities has been prepared from the training data. Then the classifier annotation on test data (output “OP” on test data) has been extracted. Next a not name word list (NN_List) has been prepared from “OP”. Each of these two lists contains a single word and followed by an annotation tag per line. Now compare every word of NN_LIST with each word of G_LIST and if these two matches more than or equal to 80% (string matching percentage) then change the annotation tag of the word of NN_LIST according to G_LIST. And finally, calculate the F-Measure on modified output data.

Algorithm 1. String Matching Based Noisy Name Identification

```
Begin
1. Prepare a gazetteer list (G_LIST) of Automobile Named Entity from training data;
2. On test data "T" find classifier annotation as output data "OP";
3. Prepare a list (NN_LIST) of words that are not classified as NE from "OP";
For ($P_1$=first line of NN_LIST; $P_1$<=last line of NN_LIST; Increment the pointer by $P_1$=$P_1$+1) {
        For ($P_2$=first line of G_LIST; $P_2$<=last line of G_LIST; Increment the pointer by $P_2$=$P_2$+1) {
                If (current word of $P_1$ match more than or equal to 80% with current word of $P_2$) {
                    Change the tag of the current word of $P_1$ according to the current word of $P_2$;
                Break
}
                Else {
                        Do not change the tag of the current word of $P_1$;
}
          }
        }
4. Calculate F-Measure on Modified Output Data;
End
```

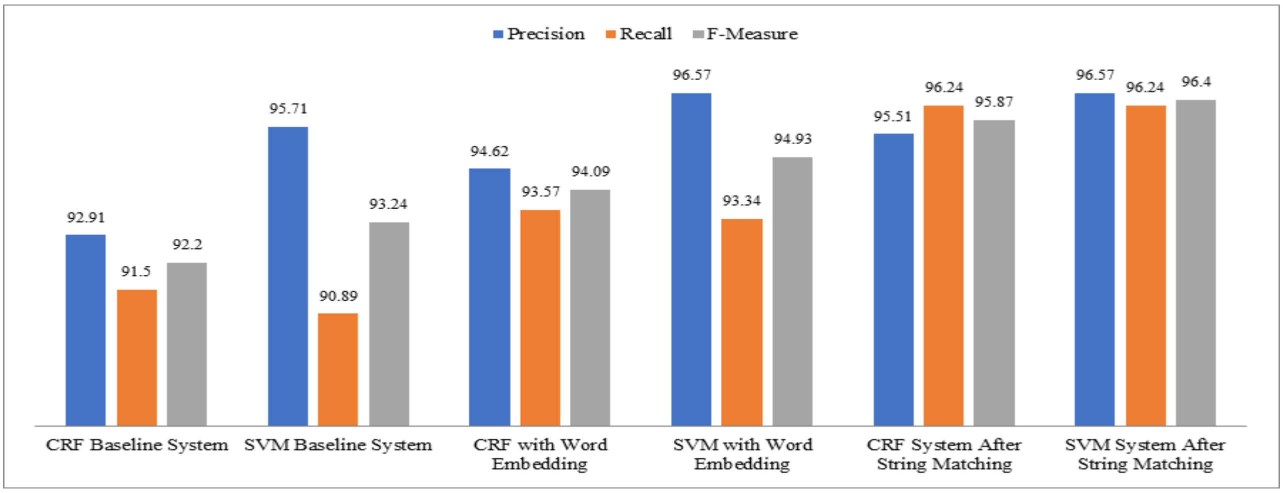

Figure 3. Comparing baseline NER result with word embedding and gazetteer-based string-matching NE identification

After executing the gazetteer-based string matching 'Algorithm 1', it has been observed that in the modified output data, a new set of names has appeared which were not identified previously. Hence after string matching CRF based system reaches an F-Measure of 95.87 with 95.51% as precision and 96.24% as recall and SVM based system achieves an F-Measure of 96.40 with 96.57% as precision and 96.24% as recall. Figure 3 demonstrates a comparative study among the results of baseline, word embedding based enhancement and string-matching based modification of the NER system in different stages of its development.

Gazetteer based string matching technique has been successfully incorporated to identify the noisy automobile names from web discussion corpus, but in very rare cases it misjudges not name words as named entities. For example, "Swift" is a named entity (product name: BPNE) and "Shift" is a not named word but it misclassifies "Shift" as a named entity (BPNE). It has also been found out that a few NEs remain undetected as the threshold matching percentage is considered here more than or equal to 80%. For example, quite a few occasions product name (BPNE) Hyundai 'i10' has been misspelled as Hyundai '100'. 'i10' matches 66.67% with '100'. Hence the system is unable to identify this as NE.

6. Conclusion

In the automobile discussion forums, user can write reviews or post some queries about any particular model of a car. Hence, these web reviews are the valuable sources of experiences shared by existing users of automobiles. To use these corpora in variety of tasks like demand supply study, market trend estimation, opinion mining and other information extraction tasks, recognizing the names is prerequisite. This article presents the study of developing NER system by exploring two machine learning algorithms, Conditional Random Field and Support Vector Machine to identify names from online automobile user review or discussion forum corpus. To learn machine learning classifiers with a set of identified candidate features, training corpus has been prepared under human supervision. To obtain better accuracy from the system, word embedding has been integrated. Though incorporation of word embedding improves system's performances but it has also been observed that a few NEs are not recognized by the system due to the noise in the corpus. To identify these noisy automobile NEs a gazetteer list has been prepared and a novel gazetteer-based string-matching approach has been proposed, it considerably improves the accuracy of the system up to F-Measure of 96.40, with recall 96.24% and precision 96.57%. Though the system is tested on automobile data set, the approach is generic and the similar procedure can also be prolific for other noisy domains.

References

[1] Liu X, Zhang S, Wei F, Zhou M. Recognizing Named Entities in tweets. In: ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics; 2011. p. 359–67.

[2] Li C, Weng J, He Q, Yao Y, Datta A, Sun A, et al. TwiNER: Named entity recognition in targeted twitter stream. In: SIGIR'12 - Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, New York, USA: ACM Press; 2012. p. 721–30.

[3] Majumder M, Barman U, Prasad R, Saurabh K, Saha SK. A Novel Technique for Name Identification from Homeopathy Diagnosis Discussion Forum. Procedia Technol. 2012 Jan 1;6:379–86.

[4] Majumder M, Saha SK. Use of global context for handling noisy names in discussion texts of a homeopathy

discussion forum. Knowl Manag E-Learning. 2014 Mar;6(1):18–29.

[5] Sharma SK, Hoque X. Sentiment predictions using support vector machines for odd-even formula in Delhi. Int J Intell Syst Appl. 2017;9(7):61–9.

[6] Jiang H, Zhou R, Zhang L, Wang H, Zhang Y. Sentence level topic models for associated topics extraction. World Wide Web. 2019 Nov;22(6):2545-60.

[7] Zheng H, He J, Huang G, Zhang Y, Wang H. Dynamic optimisation based fuzzy association rule mining method. International Journal of Machine Learning and Cybernetics. 2019 Aug;10(8):2187-98.

[8] Grishman R. The NYU system for MUC-6 or where's the syntax? In 1995. p. 167.

[9] Fukuda K, Tamura A, Tsunoda T, Takagi T. Toward information extraction: identifying protein names from biological papers. Pac Symp Biocomput. 1998;707–18.

[10] Baluja S, Mittal VO, Sukthankar R. Applying machine learning for high-performance named-entity extraction. Comput Intell. 2000 Nov 28;16(4):586–95.

[11] Zhou G, Su J. Named entity recognition using an HMM-based chunk tagger. In: aclweb.org. 2001. p. 473.

[12] Jamil Q, Zafar MR. Big Data and Named Entity Recognition Approaches for Urdu Language. EAI Endorsed Transactions on Scalable Information Systems. 2018 Apr 1; 5(16).

[13] Huang T, Gong YJ, Chen WN, Wang H, Zhang J. A probabilistic niching evolutionary computation framework based on binary space partitioning. IEEE transactions on cybernetics. 2020 Mar 11.

[14] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings. International Conference on Learning Representations, ICLR; 2013.

[15] Bikel DM, Miller S, Schwartz R, Weischedel R. Nymble: a High-Performance Learning Name-finder. In: IN PROCEEDINGS OF THE FIFTH CONFERENCE ON APPLIED NATURAL LANGUAGE PROCESSING. 1997. p. 194--201.

[16] Collier N, Nobata C, Tsujii J. Extracting the names of genes and gene products with a hidden Markov model. In: Proceedings of the 18th conference on Computational linguistics -. Morristown, NJ, USA: Association for Computational Linguistics (ACL); 2000. p. 201–7.

[17] Shen D, Zhang J, Zhou G, Su J, Tan C-L. Effective adaptation of a Hidden Markov Model-based named entity recognizer for biomedical domain. In: aclweb.org. 2003. p. 49–56.

[18] Morwal S, Jahan N. Named Entity Recognition Using Hidden Markov Model (HMM): An Experimental Result on Hindi, Urdu and Marathi Languages. Vol. 3, International Journal of Advanced Research in Computer Science and Software Engineering. 2013.

[19] Mao X, Li F, Wang H, Wang H. Named entity recognition of electronic medical record based on improved HMM algorithm. In: Proceedings - 2017 International Conference on Computer Technology, Electronics and Communication, ICCTEC 2017. Institute of Electrical and Electronics Engineers Inc.; 2017. p. 435–8.

[20] Patil N V., Patil AS, Pawar B V. HMM based Named Entity Recognition for inflectional language. In: 2017 International Conference on Computer, Communications and Electronics, COMPTELIX 2017. Institute of Electrical and Electronics Engineers Inc.; 2017. p. 565–72.

[21] Borthwick A. A maximum entropy approach to named entity recognition. (Doctoral dissertation, New York University, Graduate School of Arts and Science). 1999.

[22] Lin Y-F, Tsai T-H, Chou W-C, Wu K-P, Sung T-Y, Hsu W-L. A maximum entropy approach to biomedical named entity recognition. In: Proceedings of the 4th International Conference on Data Mining in Bioinformatics. 2004. p. 56–61.

[23] Saha SK, Sarkar S, Mitra P. Feature selection techniques for maximum entropy based biomedical named entity recognition. J Biomed Inform. 2009;42(5):905–11.

[24] Konkol M, Konopík M. Maximum entropy named entity recognition for Czech language. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2011. p. 203–10.

[25] Ahmed I, R S. Named Entity Recognition by Using Maximum Entropy. Int J Database Theory Appl. 2015;8(2):43–50.

[26] Settles B. Biomedical named entity recognition using conditional random fields and rich feature sets. In: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP). 2004. p. 104.

[27] Tsai TH, Chou WC, Wu SH, Sung TY, Hsiang J, Hsu WL. Integrating linguistic knowledge into a conditional random field framework to identify biomedical named entities. Expert Syst Appl. 2006 Jan 1;30(1):117–28.

[28] Leaman R, Gonzalez G. BANNER: An executable survey of advances in biomedical named entity recognition. In: Pacific Symposium on Biocomputing 2008, PSB 2008. 2008. p. 652–63.

[29] Das A, Ganguly D, Garain U. Named entity recognition with word embeddings and wikipedia categories for a low-resource language. ACM Trans Asian Low-Resource Lang Inf Process. 2017 Jan 1;16(3):1–19.

[30] Kazama J, Makino T, Ohta Y, Tsujii J. Tuning support vector machines for biomedical named entity recognition. In: Proceedings of the workshop on natural language processing in the bio-medical domain at ACL. 2002. p. 1–8.

[31] Ekbal A, Bandyopadhyay S. Bengali Named Entity Recognition using Support Vector Machine. In: Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages. 2008. p. 51–8.

[32] Ju Z, Wang J, Zhu F. Named entity recognition from biomedical text using SVM. In: 5th International Conference on Bioinformatics and Biomedical Engineering, iCBBE 2011. 2011.

[33] Patra R, Saha SK. A kernel-based approach for biomedical named entity recognition. Sci World J. 2013;2013.

[34] Debbarma A, Bhattacharya P, Purkayastha BS. Named entity recognition for a low resource language. Int J Recent Technol Eng. 2019;8(3):587–90.

[35] Ekbal A, Bandyopadhyay S. Named Entity Recognition using appropriate unlabeled data, post-processing and voting. Inform. 2010;34(1):55–76.

[36] Saha S, Ekbal A, Sikdar UK. Named entity recognition and classification in biomedical text using classifier ensemble. Int J Data Min Bioinform. 2015 Jan 1;11(4):365–91.

[37] Saha SK, Majumder M. Development of a hindi named entity recognition system without using manually annotated training corpus. Int Arab J Inf Technol. 2018;15(6):1088–98.

[38] Li JY, Zhan ZH, Wang H, Zhang J. Data-driven evolutionary algorithm with perturbation-based ensemble

surrogates. IEEE Transactions on Cybernetics. 2020 Aug 10.

[39] Ponomareva N, Pla F, Molina A, Rosso P. Biomedical named entity recognition: A poor knowledge HMM-based approach. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Verlag; 2007. p. 382–7.

[40] GuoDong Z, Jian S. Exploring deep knowledge resources in biomedical name recognition. In: Proceedings of joint workshop on natural language processing in biomedicine and its applications (JNLPBA). 2004. p. 96–9.

[41] Poibeau T. Boosting Robustness of a Named Entity Recognizer. Int J Semant Comput. 2009 Mar 1;3(1):91–104.

[42] Suárez-Paniagua V, Segura-Bedmar I, Mart\'inez P. Word Embedding Clustering for Disease Named Entity Recognition. Proc Fifth BioCreative Chall Eval Work. 2015;299–304.

[43] Ivanitskiy R, Shipilo A, Kovriguina L. Russian named entities recognition and classification using distributedword and phrase representations. In: CEUR Workshop Proceedings. 2016. p. 150–6.

[44] Seok M, Song H-J, Park C-Y, Kim J-D, Kim Y. Comparison of NER Performance Using Word Embedding. In: The 4th International Conference on Artificial Intelligence and Application. 2015. p. 784–8.

[45] Siencnik SK. Adapting word2vec to Named Entity Recognition. Proc 20th Nord Conf Comput Linguist (NODALIDA 2015). 2015;(Nodalida):239–43.

[46] Seok M, Song HJ, Park CY, Kim JD, Kim Y seop. Named entity recognition using word embedding as a feature. Int J Softw Eng its Appl. 2016;10(2):93–104.

[47] Zhao M, Masino AJ, Yang CC. A Framework for Developing and Evaluating Word Embeddings of Drug-named Entity. In Association for Computational Linguistics (ACL); 2019. p. 156–60.

[48] An J, Lee S, Lee GG. Automatic acquisition of named entity tagged corpus from world wide web. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics (ACL); 2003. p. 165–8.

[49] Downey D, Broadhead M, Etzioni O. Locating complex named entities in web text. In: IJCAI International Joint Conference on Artificial Intelligence. 2007. p. 2733–9.

[50] Ritter A, Sam C, Mausam, Etzioni O. Named entity recognition in tweets: An experimental study. In: EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. Association for Computational Linguistics; 2011. p. 1524–34.

[51] Abdessalem Karaa W Ben. Named Entity Recognition Using Web Document Corpus. Int J Manag Inf Technol. 2011 Feb 28;3(1):46–56.

[52] Aguilar G, Maharjan S, López-Monroy AP, Solorio T. A Multi-task Approach for Named Entity Recognition in Social Media Data. Proceedings of the 3rd Workshop on Noisy User-generated Text. Stroudsburg, PA, USA: Association for Computational Linguistics; 2019. p. 148–53.

[53] Singh V, Vijay D, Akhtar SS, Shrivastava M. Named Entity Recognition for Hindi-English Code-Mixed Social Media Text. In: Proceedings of the Seventh Named Entities Workshop. Stroudsburg, PA, USA: Association for Computational Linguistics; 2019. p. 27–35.

[54] Sabty C, Elmahdy M, Abdennadher S. Named Entity Recognition on Arabic-English Code-Mixed Data. In: Proceedings - 13th IEEE International Conference on Semantic Computing, ICSC 2019. Institute of Electrical and Electronics Engineers Inc.; 2019. p. 93–7.

[55] Lafferty J, Mccallum A, Pereira F. Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data Abstract. Dep Pap. 1999 Jun 28;2001(June):282–9.

[56] Blei DM, Ng AY, Jordan MI, Wallach HM, Hinton GE, Osindero S, et al. Conditional random fields: An introduction. Neural Comput. 2004;18(4–5):1–9.

[57] Sain SR, Vapnik VN. The Nature of Statistical Learning Theory. Technometrics. 1996;38(4):409.

[58] Razova E, Kotelnikov E. Concentration areas of sentiment lexica in the word embedding space. Int J Cogn Informatics Nat Intell. 2019;13(2):48–62.

[59] Nedjah N, Santos I, de Macedo Mourelle L. Sentiment analysis using convolutional neural network via word embeddings. Evol Intell. 2019 Apr 3;1–25.