A FEATURE BASED SIMPLE MACHINE LEARNING APPROACH WITH WORD EMBEDDINGS TO NAMED ENTITY RECOGNITION ON TWEETS

(KAVRAM TANIMA ÜZERİNE ÖZELLİK TABANLI BİR MAKİNE ÖĞRENMESİ YAKLAŞIMI)

by

Mete TAŞPINAR, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

June 2017

This is to certify that the thesis entitled

A FEATURE BASED SIMPLE MACHINE LEARNING APPROACH WITH WORD EMBEDDINGS TO NAMED ENTITY RECOGNITION ON TWEETS

prepared by **Mete TAŞPINAR** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering** at the **Galatasaray University** is approved by the

Examining Committee:

Prof. Dr. Tankut ACARMAN (Supervisor)
Department of Computer Engineering
Galatasaray University ----------

Asst. Prof. Murat Can GANİZ
Department of Computer Engineering
Marmara University ----------

Asst. Prof. Murat AKIN
Department of Industrial Engineering
Galatasaray University ----------

Date: ----------

ACKNOWLEDGEMENTS

I owe my sincere gratitude to Prof. Dr. Tankut Acarman for the constant support along the whole master period, and continuous encouragement and supervision in writing my thesis. I am grateful to him for his patience in answering my questions on my research.

My sincere gratitude also goes to Asst. Prof. Murat Can Ganiz from Marmara University for his constant guidance in writing my thesis. I am very thankful to him for always being ready to talk about my research.

Finally, I would like to thank to my wife Zeren and to my 50-days old son Ekin for their love, care, unconditional support and everlasting patience.

June 2017

Mete TAŞPINAR

TABLE OF CONTENTS

LIST OF SYMBOLS

NER	**:** Named Entity Recognition
NLP	**:** Natural Language Processing
NEEL	**:** Named Entity rEcognition and Linking
CRF	**:** Conditional Random Fields
EMM	**:** European Media Monitor
POS	**:** Part-of-Speech
ML	**:** Machine Learning
P	**:** Precision
R	**:** Recall
F	**:** F-Measure
TP	**:** True Positive
TN	**:** True Negative
FP	**:** False Positive
FN	**:** False Negative
MITIE	**:** MIT Information Extraction Toolkit

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

With the widespread use of the internet and especially the mobile platforms, data is now being produced to a large extent. It is not possible to produce meaningful information on this quantity of data with human power. For this reason, knowledge of data mining has emerged. Machine learning algorithms are used in data mining.

The application areas of machine learning are quite extensive. These are estimation of unwanted (spam) mails, automatic grouping of mails (primary, social, updates etc.), identification of anomalies in credit card or account movements, voice recognition, face/age recognition in the given picture or video, product recommendation to customers and shape recognition.

Machine learning algorithms are generally divided into two as supervised and unsupervised learning. The information (label) to be extracted from the obligation to supervised learning is known in advance. We give the resultant algorithm that will output when certain inputs are given. The algorithm predicts the result with enough data to recognize the data set and then outputs with new entries. In the case of unsupervised learning, the information to be predicted is not known in advance, and the algorithm predicts this information too.

Named Entity Recognition (NER) is a well-studied domain in Natural Language Processing (NLP). Traditional NER systems, such as Stanford NER system, achieve high performance with formal and good grammatically well-structured texts. However, when these systems are applied to informal and noisy texts, which have mixed language with emoticons or abbreviations, there is a significant degradation in results.

Supervised learning algorithms are used in this study. The social media application Twitter data is used as the data set. The biggest challenges in such studies are data collection and data cleansing. It is said that these processes have received more than 80% of the study.

In this study, Named Entity Recognition process was performed on micro blog (Twitter) data. Person, organization, location, product, event and character information were tried to be predicted from the given 140 characters. The data set of the NEEL conference held in 2016 was used. The results of the proposed system are also compared with the results in the same conference. The conference data includes tweet ids. The texts of the tweets are reached via the ids. Since about a year has passed since this conference, the accounts in the given ids have been able to reach nearly half of the data set due to closure of the accounts, or because the tweets are confidential. Data requests from groups that we have compared conference owners and conference results have either never been answered or negative feedbacks have been received. When evaluating our system, especially our performance in the trained part of the algorithms, we have had a negative effect on our performance.

In addition, the evaluation criteria of other groups have not been explained in detail in the articles they published, leading to some assumptions. Our group's evaluation criteria and detailed information requests were either unanswered or not informed of their privacy policy requirements.

The difficulty of working in social media, i.e. non-official data, is a major factor that makes it difficult to work because any language rules are not respected and abbreviations are used in order to highlight, emphasize, or limit space. The dataset consists of English tweets. Algorithms have been developed to provide very high performance on newspaper and journal data, which have been working on named entity recognition for many years. However, these algorithms give poor performances for the above-mentioned type of data written in everyday diction. For this reason, it is necessary to develop new systems specific to these data types. This work focuses on this problem.

In the study, basically two main experiments were carried out. In the first stage; Algorithm input has many features such as word lengths, starting with capital letters, emoji, hashtag, mention use, consonant/vowel letters etc. No significant improvement has been observed in a large number of experiments with this feature set. In addition to the previous features in the second stage; The Word2Vec feature is used. Along with this feature, a high degree of healing is observed in the algorithm.

According to the results of the experiments, it was better than 2 out of 3 studies using this dataset by participating in the 2016 NEEL conference.

Keywords: Named Entity Recognition, Information Extraction, Word2Vec, Social Media, Informal Texts, Twitter

ÖZET

İnternetin ve özellikle de mobil platformların yaygınlaşmasıyla günümüzde çok büyük miktarda veri üretilmeye başlanmıştır. Bu miktardaki veriden anlamlı bilgi çıkarmak – özellikle anlık olarak- insan gücüyle mümkün olamamaktadır. Bu sebeple veri madenciliği bilimi ortaya çıkmıştır. Veri madenciliğinde makine öğrenmesi algoritmaları kullanılmaktadır.

Makine öğrenmesinin uygulama alanları oldukça geniştir. İstenmeyen (spam) maillerin tahmini, maillerin otomatik olarak gruplanması (birincil, sosyal, güncellemeler vb ..), kredi kartı veya hesap hareketlerinde anomali tespiti, ses tanıma, verilen resimde veya videoda yüz/yaş tanıma/anlama, müşterilere ürün önerme, hastalık teşhisi, şekil tanıma gibi popüler örnekler mevcuttur.

Makine öğrenmesi algoritmaları genel anlamda gözetimli ve gözetimsiz öğrenme olarak 2'ye ayrılır. Gözetimli öğrenmede çalışılan veriden çıkarılacak bilgi (etiket) önceden bellidir. Yani; belirli girdiler verilince çıkacak sonucu algoritmaya veririz. Algoritma yeterli miktarda veriyle veri kümesini tanıdıktan sonra yeni girdilerle çıkacak sonucu tahmin eder. Gözetimsiz öğrenmede ise tahmin edeceği bilgiler önceden belli değildir, algoritma bu bilgiyi de tahmin eder.

Kavram Tanıma doğal dil işlemede uzun süredir çalışılan bir alandır. Stanford NER gibi geleneksel yöntemler resmi ve gramer olarak düzgün verilerde çok iyi sonuçlar vermektedir. Fakat bu sistemler sosyal medya gibi kısaltmaların ve dil yanlışlarının çok olduğu verilerde iyi sonuçlar vermemektedir.

Bu çalışmada gözetimli öğrenme algoritmaları kullanılmıştır. Veri kümesi olarak sosyal medya uygulaması Twitter verileri kullanılmıştır. Bu tür çalışmalarda en büyük

zorluklar veri kümesi bulma ve bulunan veriyi temizleme işlemleridir. Bu işlemlerin çalışma zamanının %80'inden fazlasını aldığı söylenir.

Bu çalışmada mikro blog (Twitter) verilerinde kavram tanıma/çıkarma işlemi yapılmıştır. Verilen 140 karakter içinden kişi, organizasyon, lokasyon, ürün, olay ve karakter bilgileri tahmin edilmeye çalışılmıştır. Veri kümesi olarak 2016 yılında yapılan NEEL konferansı verileri kullanılmıştır. Önerdiğimiz sistemin sonuçları da aynı konferanstaki sonuçlarla karşılaştırılmıştır. Konferans datası tweet id'lerini kapsamaktadır. Id'ler üzerinden tweetlerin text kısmına ulaşılmıştır. Bu konferansın üstünden yaklaşık bir yıl geçtiği için, verilen id'lerdeki hesapların kapanması veya tweet'lerin gizli duruma geçmesi yüzünden veri kümesinin yaklaşık yarısına ulaşabilmiştir. Konferans sahiplerine ve konferansta sonuçlarını karşılaştırdığımız gruplardan data talebimiz de ya hiç cevaplanmamış ya da olumsuz dönüşler alınmıştır. Sistemimizin değerlendirilmesi yapılırken özellikle algoritmaların eğitilme kısmında diğer gruplara göre eksik veriyle çalışmamız performansımızı olumsuz yönde etkilemiştir.

Ayrıca diğer grupların değerlendirme kriterleri yayınladıkları makalelerde ayrıntılı anlatılmadığı için karşılaştırmalarımızı bazı varsayımlarda bulunup yapmamıza sebep olmuştur. Grupların değerlendirme kriterleriyle ayrıntılı bilgi taleplerimiz de ya hiç cevaplanmamış ya da gizlilik politikaları gereği bilgi verilmemiştir.

Sosyal medya yani resmi olmayan verilerde çalışma zorluğu herhangi bir dil kuralına uyulmaması ve verinin dikkat çekmesi, vurgulanması amacıyla veya yer kısıtı olmasından kısaltmalar kullanılması çalışmayı zorlaştıran başlıca etmenlerdir. Veri kümesi ingilizce tweet'lerden oluşmaktadır. Kavram tanıma için uzun yıllardır üstünde çalışılan gazete, dergi verilerinde oldukça yüksek performans sağlayan algoritmalar geliştirilmiştir. Fakat bu algoritmalar gündelik dille yazılan yukarıda bahsedilen tipte veriler için düşük performanslar vermektedir. Bu sebeple bu veri tiplerine özgü olarak yeni sistemler geliştirmek gerekmektedir. Bu çalışmada bu sorun üstüne yoğunlaşılmıştır.

Çalışmada temel olarak 2 etapta deneyler yapılmıştır. İlk etapta; algoritmalara input olarak kelime uzunlukları, büyük/küçük harf ile başlama, emoji, hashtag, mention kullanımı, sesli/sessiz harf oranı vb gibi çok sayıda özellik verilmiştir. Bu özellik kümeleriyle yapılan çok sayıda deneylerde kayda değer bir iyileşme görülmemiştir. İkinci etapta önceki özelliklere ek olarak; Word2Vec özelliği kullanılmıştır. Bu özellikle beraber algoritmada yüksek oranda iyileşme gözlemlenmiştir.

Yapılan deney sonuçlarına göre 2016 NEEL konferansına katılıp bu veri kümesini kullanan 3 çalışmadan 2'sinden daha iyi sonuç alınmıştır.

1. INTRODUCTION

Gathering meaningful data from social media platforms becomes more important with ever increasing amount of valuable data accumulated in these platforms. For mining microblog texts such as Tweets, several challenges of meaning lies along enormous amount of noisy data involving abbreviations, typing errors, and special characters to indicate special terms such as hashtags or mentions. Due to these unpredicted and informal data, existing NER systems do not perform well (Ek et al, 2011; Ritter et al, 2011; Celikkaya et al, 2013; Kucuk et al, 2014; Kucuk & Steinberger, 2014; Eken & Tantug, 2015; Ghosh et al, 2016; Greenfield et al, 2016; Okur et al, 2016; Torres-Tramon et al, 2016).

In this study, we present fast and scalable feature based machine learning approach using additional word embedding features for identifying different classes of named entities in tweets. In order to evaluate our approach we use the NEEL dataset (Rizzo et al, 2016) for our experimental study and we evaluate our results with respect to the studies (Ghosh et al, 2016; Greenfield et al, 2016; Torres-Tramon et al, 2016) published in NEEL 2016 Challenge.

We conduct several experiments with different subsets of the features. We illustrate that the addition of word embedding features considerably increases the accuracy, and a simple machine learning classifier with word embedding features can compete with more complicated methods such as Conditional Random Fields (CRF). The impact of this study and usability of the results is crucial for several reasons.

Firstly, more advanced classifiers and ensemble learning approaches can be applied to improve the achieved performance.

Secondly, use of machine learning algorithms and word embeddings algorithms such as Word2Vec (Mikolov et al, 2013a ; Mikolov et al, 2013b; Mikolov et al, 2013c) will allow to develop highly scalable and distributed models versus the traditional models used in this domain such as CRFs.

1.1 Machine Learning

Machine learning systems search through data to look for meaningful patterns. Machine learning algorithms are categorized as supervised or unsupervised.

Supervised learning is to learn or predict mapping function from the given input variables and the given output variable. Before training the model the correct answers are known. Supervised algorithms are categorized as classification and regression. When the output of the algorithm is a category such as color or yes/no choice we call it the classification, if the output is a real value such as money, weight, age or temperature we call it the regression.

For unsupervised learning, there is only input data and no output data for training. Unsupervised algorithms are categorized as clustering and association.

1.1. Applications of Machine Learning Problems

There are various applications of machine learning.

Spam mail detection is the identify the email messages that are spam or legitimate. Gmail's email tagging (Primary, Social, Updates, etc.) is a good example of a machine learning application. Credit card fraud detection is the identify the transactions that were made by the customer or not. Speech understanding is the identify the request from the

user. The iPhone Siri has this property. Face, age detection is the identify the faces and ages of the people from the given pictures.

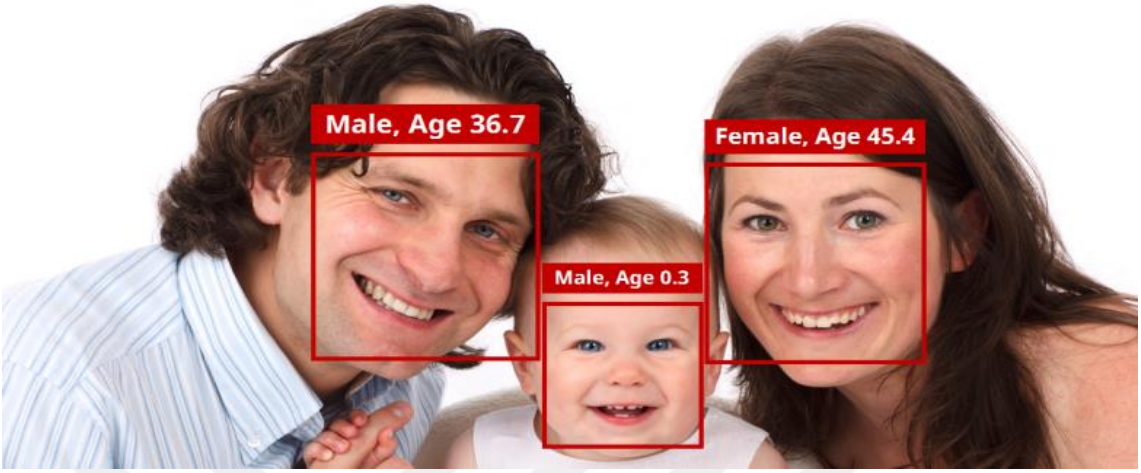

Figure 1.1: Face, age detection

Product Recommendation is the identify the products which the customer can be interested in. Amazon and Facebook have this ability. They recommend products to buy or the people to connect with. Medical diagnosis is the prediction of a person has an illness or not with the help of the given the symptoms of patient records. We know that using IBM Watson they could diagnose cancer. Stock trading is the determine whether the stock should be bought, held or sold according to its price movements. Shape detection is used to predict which shape the person is drawing. Various mobile applications do this.

1.2. Named Entity Recognition (NER)

Named Entity Recognition is an information extraction task which targets at the identifying proper names of people, organizations, locations, or other entities. There are several NER tools which are built for formal text types, especially for English. These are OpenNLP, Stanford NER (Finkel et al, 2005), AlchemyAPI, OpenCalais, spaCy1, Alias-i LingPipe, Natural Language Toolkit (Python-NLTK).

These existing tools for NER perform well on formal texts such as newspapers or articles, but they perform poorly when applied to informal texts such as social media texts.

In Table 1.1 there is a Named Entity Recognition example for the sentence “Albert Einstein was born in Ulm”.

I stands for Inside, O stands for Outside and B stands for Beginning.

Table 1.1: Named Entity Recognition example

Word	**Entity**
Albert	B-PERSON
Einstein	I-PERSON
was	O
born	O
in	O
Ulm	I-LOCATION

2. LITERATURE REVIEW

NER on microblog texts is a new research area and attracted a lot of attention in the past few years. A special Twitter implementation of GATE Natural Language Processing (NLP) framework abbreviated as TwitIE (Bontcheva et al, 2013) is presented in (Torres-Tramon et al, 2016). GATE NLP is based on Stanford NER (Finkel et al, 2005) classifier and uses CRF model. In (Ghosh et al, 2016), a feature based approach performing Stanford NER (Finkel et al, 2005) is described and ARK is used for part-of-speech (POS) tagging. Several features such as length of the mention and when the mention is capitalized are trained with the supervised classifiers such as Random Forest, Naive Bayes, k-nearest neighbour and support vector machines. The standard NER system implementations such as Stanford NER (Finkel et al, 2005), MIT Information Extraction Toolkit (MITIE), twitter_nlp (Ritter et al, 2011) and TwitIE (Bontcheva et al, 2013) are studied in (Greenfield et al, 2016). The dataset is trained for MITIE.

In (Caliano et al, 2016), a knowledge-base approach is described. They use T-NER (Ritter et al, 2011) a state-of-the-art NER system to segment the tweet into entities and non-entities. In (Geyer et al, 2016), they present exploratory analysis using MITIE with named entity recognition in the micropost genre.

For Turkish tweets the NER software of European Media Monitor (EMM) is used in (Kucuk et al, 2014) and (Celikkaya et al, 2013). In (Kucuk et al, 2014) for the informal short texts there are difficulties; not capitalizing the initial letters (PLO), not seperating names from suffixes with apostrophes, modifying names (repeating chars), utilizing non-accentuated chars instead of Turkish chars, lack of clues for NER (person titles/professions), due to the char limitation, referring to location and organization

names in contracted forms (face instead of Facebook, bogazici instead of Bogazici University). They first performed NER experiment using the NER software of EMM (European Media Monitor) then performed NER experiment using modified version of EMM: single token (appeared at least 30 times in Turkish news articles) and extend organization names (about 550 names).

In (Celikkaya et al, 2013) they studied with 3 Turkish datasets. They first tokenized the data, and then they mame a morphological analysis. The difficulties for the informal Turkish micro texts were slang words , repeated characters for ejaculation, hash tags, mentions and lack of capitalization. They used Conditional random fields (CRFs) as the classification algorithm. For future work they planned to add numex and timex entity types to their system.

Appearance of some NEs in hashtags. 3 different Turkish datasets including tweets, a speech-to-text interface and the data taken from a Turkish hardware forum are applied to a machine learning algorithm named CRFs (Seker & Eryiğit, 2012, Eken & Tantug, 2015). In (Seker & Eryiğit, 2012) they studied specifically for the recognition of person, location and organization. They used some gazetteers, a two-level morphological analyzer (Oflazer, 1994) and a morphological disambiguator (Sak et al, 2008). They also added morphological features like stemming, part of speech tag, noun case, proper noun and inflectional features, lexical features like case feature and a start of the sentence feature. They obtained highest results for Turkish named entity recognition.

In (Eken & Tantug, 2015) in order to not to miss the words contain spelling errors they applied distance based matching with Levenshtein distance algorithm (Levenshtein, 1966). This algorithm calculate distance between two strings, in this study they used this calculation to compare the input token and the token in the gazetteer. They improved performance on tweets and got %64 f-measure (Eken & Tantug, 2015).

Considering semi-supervised learning approaches, artificial neural networks are used to extract NER on Turkish tweets with word embeddings (Okur et al, 2016). In (Okur et al, 2016) they obtained better F-score performances then the previous NER systems on Turkish tweets. They could easily adapt their system which is in the other languages because they did not employ any language dependent features. They also added local features like context, capitalization, previous tags, word type information, token prefixes, token suffixes and word embeddings.

For Spanish formal documents, a rule-based approach is applied in (Moreno et al, 2015) and in (Moreno et al, 2016) an unsupervised feature generation is shown to improve the stand-alone performance of the process NER. In (Moreno et al, 2015) they proposed their system to extract mentions of medicinal products and active ingredients. The system evaluated with Spanish technical documents but their approach was language independent. They achieved %90 F-measure. For future work they planned to enhance their system to recognize other relevant entities like dosages forms. In (Moreno et al, 2016) they proposed an automatic feature extraction process without using a dictionary to build an active ingredient named entity recogniser. Their system was language and domain independent. They achieved %87.3 F1. For future work they planned to enhance their system with traditional named entities.

In (Ritter et al, 2011), the system exceeds with an increase of F1 by 25%, the Stanford NER (Finkel et al, 2005) system performance by re-building the part-of-speech, and chunking jobs. Their system is a supervised model based on Conditional Random Fields (CRF).

3. METHODOLOGIES

3.1. Platforms

We used python (3.5.2) for programming language and PyCharm and Sublimtext for IDE. For machine learning algorithms we used gensim and sci-kit libraries.

3.2. Our Approach

Our approach to NER in short and noisy texts, in particular tweets is a simple, fast and scalable feature based machine learning approach with additional word embedding features for identifying different classes of named entities in tweets. The features used in this study are given in Table 3.1.

We employed a three term running window structure while running the experiments. Therefore, features that are described in Table 3.1 (except the last one) are calculated for the term t, the term before t and the term after t. For the feature POS Tag, we used Stanford POS Tagger with 36-tag tagset.

Table 3.1: Features used in our approach

Feature Name	**Description**	**Type**
StartCapital	Whether the term is capitalized or not	Boolean
AllCapital	If the term is all uppercase	Boolean
HashTag	If the term starts with the letter '#'	Boolean
Mention	If the term starts with the letter '@'	Boolean
POS	POS Tag of the term	Nominal
Length	Number of characters in the word	Numeric
VowelRatio	The ratio of number consonant over the number of vowels in the word	Numeric
SimClassCentroid[i]	Cosine Similarity between term's word2vec vector to the centroid word2vec vector of class i	Numeric

Second, we adapted a Word2Vec (Siencnik, 2015) algorithm to our system to create vector space representations of each term in our training set. Word2vec is trained by a large corpus obtained from Twitter. Following this we calculated the class centroid vectors by averaging all the term vectors belonging to a particular class. For each term in our dataset, cosine similarity to each class centroid is calculated. These similarity values are used to develop feature based machine learning.

4. EVALUATION

4.1. Dataset

For training and testing purposes, we use the NEEL 2016 twitter dataset provided by (Rizzo et al, 2016). This dataset involves 6025 tweets, 8665 entities for training and 3164 tweets, 1022 entities for testing. The dataset is constituted by the tweet IDs and it does not provide the tweet texts.

Since some Twitter accounts were closed and not available due to restrictions, we could not obtain text information from Twitter API. Finally, we have 3450 of 8665 entities for training and 695 of 1022 entities for testing. Table 4.1 gives the number of annotated NER types in our train and test data.

We use Python (3.5.2) programming language with gensim library for executing word2vec as an underlying word embeddings algorithm and scikit-learn library for training and testing purposes. We use the following supervised algorithms in our experimental study: Logistic Regression, Support Vector Machine, KNeighborsClassifier, MultinomialNB, GaussianNB, BernoulliNB, ExtraTreeClassifier and DecisionTreeClassifier.

Table 4.1: Distribution of NER Types in the dataset

NER Type	Train Data	Test Data
Person	529	238
Thing	321	29
Organization	597	122
Location	511	25
Product	298	238
Event	85	16
Character	16	27

4.2. Performance Measures

We measured our system effectiveness with the measures called Precision, Recall and F-Measure.

Precision (P) is the fraction of retrieved documents that are relevant. In other words how many of the returned documents are correct.

$$\text{Precision} = \frac{\#\ (relevant\ items\ retrieved)}{\#\ (retrieved\ items)}$$

Recall (R) is the fraction of relevant documents that are retrieved.

$$\text{Recall} = \frac{\#\ (relevant\ items\ retrieved)}{\#\ (relevant\ items)}$$

F-Measure is the harmonic mean of precision and recall.

$$\text{F} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

These measures can also be explained with the following table.

Table 4.2: Contingency Table

	Relevant	Nonrelevant
Retrieved	True Positives (TP)	False Negatives (FN)
Non Retrieved	False Positives (FP)	True Negatives (TN)

Then

$$\text{Precision} = \frac{TP}{TP + FP}$$

And

$$\text{Recall} = \frac{TP}{TP + FN}$$

4.3. Evaluation Results

We conduct several experiments using different subsets of the feature set and the entity types. The dataset is annotated with seven entity types: Person, Thing, Organization, Location, Product, Event and Character (Table 4.1).

The 5 features that we selected for the results table are as the following ones, the other features that we try did not generate good scores.

- If the word is capitalized
- If the word is all capitalized
- If the word starts with the letter '#'
- If the letter starts with the letter '@'
- POS Tag

The other metric for our experiments is the Word2Vec feature. We calculated the average vector of the 7 NER types. Then we calculated the cosine distance with the word and the average vector of all 7 NER types. We used a Word2Vec (Kucuk & Steinberger, 2014) model trained on 400 million tweets for our model. This trained model did not cover all our tweet data.

Evaluation results with 7 NER types and different subsets of features are given through Table 4.3 to Table 4.5.

In Table 4.3, 5 features and 7 NER types are used. A precision of 0.55 was reached and F1 is reached at level 0.49 when we use ExtraTreeClassifier algorithm.

Table 4.3: Experiment Results with 5 features and 7 NER Types

5 features, 7 NER classes	**Precision**	**Recall**	**F1**	**F1 (Micro Average)**
Logistic Regression	0.27	0.25	0.24	0.246043
SVC	0.52	0.44	0.48	0.444604
SVC(kernel='linear')	0.24	0.27	0.24	0.271942
KNeighborsClassifier	0.35	0.30	0.29	0.296402
MultinomialNB	0.35	0.24	0.25	0.243165
GaussianNB	0.29	0.23	0.22	0.225899
BernoulliNB	0.35	0.24	0.25	0.237410
ExtraTreeClassifier	0.55	0.46	0.49	0.458992
DecisionTreeClassifier	0.55	0.46	0.49	0.457553

In table 4.4, 7 word2vec features and 7 NER types are used and KNeighboursClassifier reaches at 0.70 precision and 0.57 F1.

Table 4.4: Experiment Results with 7 word2vec features and 7 NER Types

7 word2vec features, 7 NER classes	**Precision**	**Recall**	**F1**	**F1 (Micro Average)**
Logistic Regression	0.65	0.55	0.52	0.551079
SVC	0.66	0.58	0.55	0.579856
SVC(kernel='linear')	0.68	0.57	0.53	0.569784
KNeighborsClassifier	0.70	0.58	0.57	0.582733
MultinomialNB	0.38	0.25	0.17	0.247482
GaussianNB	0.73	0.35	0.45	0.348201
BernoulliNB	0.00	0.04	0.00	0.038848
ExtraTreeClassifier	0.58	0.52	0.49	0.519424
DecisionTreeClassifier	0.52	0.45	0.41	0.453237

In table 4.5, combination of 5 features and 7 word2vec features slightly increases precision at 0.71 and F1 at 0.58 using Logistic Regression algorithm.

Table 4.5: Experiment Results with 5 features + 7 word2vec features and 7 NER Types

5 features + 7 word2vec features, 7 NER classes	**Precision**	**Recall**	**F1**	**F1 (Micro Average)**
Logistic Regression	0.71	0.56	0.58	0.556834
SVC	0.63	0.56	0.58	0.555395
SVC(kernel='linear')	0.69	0.55	0.57	0.546762
KNeighborsClassifier	0.56	0.50	0.50	0.499280
MultinomialNB	0.36	0.25	0.25	0.248920
GaussianNB	0.51	0.29	0.28	0.289208
BernoulliNB	0.28	0.26	0.25	0.258992
ExtraTreeClassifier	0.54	0.51	0.49	0.512230
DecisionTreeClassifier	0.52	0.46	0.46	0.463309

In Table 4.6, we also evaluate both of 5 features and 7 word2vec features with an additional class of ‘No Type’, which means that a term is not a named entity. Due to the nature of natural language, an overwhelming majority of the terms in tweets are not named entities. This leads to a highly skewed class distribution where No Type class dominates with 86%. On this dataset, two models, ExtreTreeClassifier with 5 features and Logistic Regression with 5 features + 7 word2vec features can reach 0.88 F1.

Table 4.6: Experiment Results with 7 NER Types + with NO Type

Models (with 7 types + No Type)	**Precision**	**Recall**	**F1**	**F1 (Micro Average)**
7 Cosine Similarity with Logistic Regression	0.83	0.88	0.84	0.876501
7 Cosine Similarity + 5 Features with Logistic Regression	0.88	0.91	0.88	0.907469
5 Features with Logistic Regression	0.85	0.89	0.86	0.891141
5 Features with SVC(kernel='linear')	0.83	0.89	0.86	0.891704
5 Features with SVC	0.88	0.90	0.87	0.897334
5 Features with SVC(C=1000000.0)	0.88	0.90	0.88	0.895457
5 Features with ExtraTreeClassifier	0.88	0.90	0.88	0.895270
5 Features with DecisionTreeClassifier	0.88	0.90	0.88	0.895270
5 Features with MultinomialNB	0.85	0.89	0.86	0.890578
5 Features with GaussianNB	0.90	0.48	0.62	0.478415

In order to get a detailed look of the results of our best performing model (Logistic Regression, 5 features + 7 word2vec features, 7 NER classes), we provide confusion matrix and class based evaluation metrics. As given in Table 4.7, majority of the instances belong to Person and Product class (238 entities for each), whose is calculated by summing the columns of the first row (Person) and the fifth row (Product), respectively.

At the meantime, organization is the most misclassified class by the machine learning algorithms. We see from the first column that majority of the Organization entities (57 out of 122) are misclassified as Person. Actually the majority of the misclassifications are accumulated at the first column.

Table 4.7: Confusion Matrix of Logistic Regression, 5 features + 7 word2vec features, 7 NER classes

NER Type	Person	Thing	Organization	Location	Product	Event	Character
Person	**209**	17	4	0	2	6	0
Thing	0	**21**	1	0	1	6	0
Org.	57	12	**37**	1	4	11	1
Location	7	2	0	**15**	0	1	0
Product	4	3	8	0	**94**	129	0
Event	0	3	2	0	0	**11**	0
Character	15	1	7	0	3	1	**0**

Table 4.8: Class based evaluation metrics for Logistic Regression, 5 features + 7 word2vec features, 7 NER classes

NER Type	Precision	Recall	F1	Support
Person	0.72	0.88	0.79	238
Thing	0.36	0.72	0.48	29
Organization	0.63	0.30	0.41	122
Location	0.94	0.60	0.73	25
Product	0.90	0.39	0.55	238
Event	0.07	0.69	0.12	16
Character	0.00	0.00	0.00	27

We compare our results with the three studies in the NEEL 2016 workshop (Rizzo et al, 2016) in Table 4.9. Our results outperform two of the three methods (Torres-Tramon et al, 2016; Greenfield et al, 2016) algorithm that uses TwitIE (Bontcheva et al, 2013), Stanford NER (Finkel et al, 2005), MITIE and twitter_nlp (Ritter et al, 2011), and they are close to the level in precision achieved by (Ghosh et al, 2016) using feature-based approach.

Table 4.9: Comparison of the performance with respect to the studies presented in NEEL 2016 workshop (Rizzo et al, 2016)

Study	**Precision**	**Recall**	**F1**
A feature based approach performing Stanford NER, (Ghosh et al, 2016)	0.729	0.626	0.674
Stanford NER, MITIE, twitter_nlp and TwitIE, (Greenfield et al, 2016)	0.587	0.287	0.386
TwitIE (CRF Model), (Torres-Tramon et al, 2016)	0.435	0.459	0.447
Our approach (Logistic Regression, 5 features + 7 word2vec features, 7 NER classes)	0.71	0.56	0.58

5. CONCLUSIONS

We have presented and evaluated a simple yet effective machine learning classification algorithms based approach to identify named entity types on noisy microblogging texts such as tweets. Our approach is based on extracting Tweet specific syntactic features along with word embeddings, in particular word2vec (Mikolov et al, 2013a; Mikolov et al, 2013b; Mikolov et al, 2013c) based semantic features and using them in traditional machine learning classifiers such as Logistic Regression, Support Vector Machines, and k-Nearest Neighborhood algorithms.

Experimental results show that our result can outperform two of the three studies in the NEEL 2016 workshop (Rizzo et al, 2016) (please see Table 5.8) in terms of F1 and get very close precision performance (0.71 vs. 0.729) to the best performing study (Ghosh et al, 2016). In the future, we are planning to employ more tweet specific features and use more complicated machine learning models such as ensembles and deep learning approaches.

In this thesis, we have explored some new ways to recognized named entities better than the existing approaches. Named entity recognition problem and similar problems that can be modeled as a sequential labeling problem such as POS tagging, dependency parsing etc... have the potential to leverage word representation learning more.

Conventional supervised learning methods, Maximum Entropy Markov Models (MEMM), Conditional Random Fields (CRF) still produce the top scores in various evaluation settings for sequential labeling tasks. Approaches that propose to use word

embeddings as part of CRFs have huge potential in the sense that they make it possible to leverage the power of word embeddings when there is the problem of data sparsity.

Regarding that we plan to expand our work by surveying new ways of incorporating word embeddings into a conventional supervised sequential labeling setting. C&W embeddings (Collobert et al, 2011; Tang et al, 2014) only incorporate contextual information in generating the new word representation vectors. However, for tasks such as sentiment classification or opinion mining, it is insufficient just to incorporate context information. For example, the two example phrases "this is a good Italian restaurant" and "this is a terrible Italian restaurant" have the same context, however, the sentiments are opposite each other.

As a result, it is necessary to extend existing C&W neural architecture in the sense that we can incorporate the sentiment information. This is the direction that we plan to focus in the future so that we will be able to devise more effective embeddings by incorporating different characteristics of the data such as the sentiment polarities into the procedure.

REFERENCES

Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., Aswani, N. (2013). TwitIE: An Open -Source Information Extraction Pipeline for Microblog Text. In Proceedings of the International Conference on Recent Advances in Natural Language Processing, ACL.

Caliano, D., Fersini, E., Manchanda, P., Palmonari, M., Messina, E.. (2016) UniMib: Entity Linking in Tweets using Jaro-Winkler Distance, Popularity and Coherence. In 6 th International Workshop on Making Sense of Microposts (#Microposts).

Celikkaya, G., Torunoglu, D., Eryigit, G. (2013). Named Entity Recognition on Real Data: A Preliminary Investigation for Turkish. In Proceedings of the 7th International Conference on Application of Information and Communication Technologies.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12:2493–2537.

Ek, T., Kirkegaard, C., Jonsson, H., Nugues, P. (2011) Named entity recognition for short text messages. Procedia-Social and Behavioral Sciences 27, 178–187.

Eken, B. and Tantug, A. C. (2015). *Recognizing named entities in Turkish tweets*. In Proceedings of the Fourth International Conference on Software Engineering and Applications, Dubai, UAE, January 2015.

Finkel, J. R., Grenager, T., Manning, C. (2005). Incorporating Non -local Information into Information Extraction Systems by Gibbs Sampling. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005).

Geyer, K., Greenfield, K., Mensch, A., Simek, O. (2016) Named Entity Recognition in 140 Characters or Less. In 6 th International Workshop on Making Sense of Microposts (#Microposts).

Ghosh, S., Maitra, P., Das, D. (2016). Feature Based Approach to Named Entity Recognition and Linking for Tweets. In 6 th International Workshop on Making Sense of Microposts (#Microposts).

Greenfield, K., Caceres, R., Coury, M., Geyer, K., Gwon, Y., Matterer, J., Mensch, A., Sahin, C., Simek, O. (2016). A Reverse Approach to Named Entity Extraction and Linking in Microposts. In 6 th International Workshop on Making Sense of Microposts (#Microposts).

Kucuk, D., Jacquet, G., Steinberger, R. (2014). Named Entity Recognition on Turkish Tweets. In Proceedings of the Language Resources and Evaluation Conference.

Kucuk, D. and Steinberger, R. (2014). Experiments to Improve Named Entity Recognition on Turkish Tweets. In Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM) @ EACL 2014, pages 71–78,Gothenburg, Sweden, April 26-30 2014.

Levenshtein, V. (1966). Binar Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics Doklady, vol. 10, pp, 707-710.

Mikolov, T., Yih, W., Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. In Proceedings of NAACL HLT.

Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient estimation of word representations in vector space. ICLR Workshop.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013). Distributed representations of phrases and their compositionality. In NIPS.

Moreno, I., Moreda, P., Rom´a-Ferri, M. T. (2015). MaNER: a MedicAl Named Entity Recogniser for Spanish. In Proceedings of the 20th International Conference on Applications of Natural Language to Information Systems, NLDB 2015.

Moreno, I., Moreda, P., Rom´a-Ferri, M. T. (2016). An Active Ingredients Entity Recogniser System Based on Profiles. In Proceedings of 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016, Salford, UK.

Oflazer, K. (1994). Two-level description of Turkish morphology. Literary and Linguistic Computing, 9(2):137-148.

Okur, E., Demir, H., Özgür, A. (2016) Named Entity Recognition on Twitter for Turkish using Semi-supervised Learning with Word Embeddings. Proceedings of the

Tenth International Conference on Language Resources and Evaluation, LREC 2016, Portorož, Slovenia.

Ritter, A., Clark, S., Etzioni M., Etzioni, O. (2011). Named Entity Recognition in Tweets: An Experimental Study. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. 2011 Edinburgh, Scotland, UK, July 27–31, 2011.

Rizzo, G., Erp, M.V., Plu, J., Troncy, R. (2016). *Making Sense of Microposts (#Microposts2016) Named Entity rEcognition and Linking (NEEL) Challenge*. In 6th Workshop on Making Sense of Microposts (#Microposts2016), pp 50–59.

Sak, H., Güngör, T., Saraçlar, M. (2008). Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In GoTAL 2008, volume 5221 of LNCS, pages 417–427.

Seker, G. A. and Eryigit, G. (2012) Initial explorations on using CRFs for Turkish Named Entity Recognition. In Proceedings of the 24th International Conference on Computational Linguistics, COLING 2012, Mumbai, India.

Siencnik, S. (2015). Adapting word2vec to Named Entity Recognition. In Proceedings of the 20th Nordic Conference of Computational Linguistics NODALIDA 2015.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In Proceedings of ACL, pages 1555–1565.

Torres-Tramon, P., Hromic, H., Walsh B., Heravi, B., Hayes, C. (2016). Kanopy4Tweets: Entity Extraction and Linking for Twitter. In 6 th International Workshop on Making Sense of Microposts (#Microposts).

APPENDICES

Appendix A.

```
from twython import Twython, TwythonError
import time
import csv

CONSUMER_KEY = "XXXXXqkBd538vOhNehrXXXX"
CONSUMER_SECRET = "XXXX6e5vKxt5MLc3KaGVRz0NhS7yusVRZj19XXXX"
OAUTH_TOKEN = "XXX026-mQH1ORAmXXXrltk1eN5iTbLoF1MlXXXX"
OAUTH_TOKEN_SECRET = "XXXMqbko54T9dRcIsb5vc1Q6lWCMXXX"
twitter = Twython(CONSUMER_KEY, CONSUMER_SECRET, OAUTH_TOKEN,
OAUTH_TOKEN_SECRET)
twitter = Twython(app_key=CONSUMER_KEY,
      app_secret=CONSUMER_SECRET,
      oauth_token=OAUTH_TOKEN,
      oauth_token_secret=OAUTH_TOKEN_SECRET)
twitter.verify_credentials()
with open("microposts2016-neel-test_neel.gs") as file:
   lines = []
   index = 0
   index_error = 0
   lines_with_texts = []
   for line in file:
      # The rstrip method gets rid of the "\n" at the end of each line
      lines.append(line.rstrip().split('\t'))
```

```
try:
          tweet = twitter.show_status(id=lines[index][0])
           lines_with_texts.append(
              lines[index][0] + '\t' + tweet['text'] + '\t' + lines[index][1] + '\t' +
lines[index][2] + '\t' +
              temp_NER + '\t' +  str(tweet['retweet_count'])  + '\t' +
str(tweet['favorite_count']) + '\t' +
              tweet['text'] [int(lines[index][1]):int(lines[index][2])])
           index = index + 1
       except TwythonError as e:
           #print("Not valid ID")
           print (e)
           index = index + 1
           index_error = index_error + 1


   thefile = open('microposts2016-neel-test_neel.txt', 'w')
   for item in lines_with_texts:
      thefile.write("%s\n" % item)
```

Appendix B.

```
from nltk.tag import StanfordPOSTagger
import nltk
import numpy as np
from sklearn import preprocessing
import gensim
from scipy import spatial
def expFile(filename, POS_File, model):
   with open(filename) as file:
 for line in file:
          lines.append(line.rstrip().split('\t'))
          index = index + 1
      for line in lines:
          train_labels_dict[line[0] + '_-_' + line[7]] = line[4]
          set_tweets.add(line[0] + '_-_' + line[1])
……..
      average4Thing = np.average(value4Thing, axis=0)
      average4Organization = np.average(value4Organization, axis=0)
      average4Person = np.average(value4Person, axis=0)
      average4Location = np.average(value4Location, axis=0)
      average4Product = np.average(value4Product, axis=0)
      average4Event = np.average(value4Event, axis=0)
      average4Character = np.average(value4Character, axis=0)
   ……
      words.append('& Start Doc &')
      for letter in set_tweets:
          words.append('& Start Tweet &')
          index_set = 0
          tweet_id = ''
```

```
 for word in nltk.word_tokenize(letter):
            if(index_set == 0):
               tweet_word = word.split('_-_')
               tweet_id = tweet_word[0]
               words.append(tweet_id + '_-_' + tweet_word[1])
            else:
               words.append(tweet_id + '_-_' + word)
            index_set = index_set + 1
         words.append('& End Tweet &')
      words.append('& End Doc &')
      …….
      df = pd.DataFrame(data, index=words)
      for j in range(df.index.size):
         wordbefore=str(df.index[j - 1])
         if  (df.index[j - 1] == '& End Doc &' or df.index[j - 1] == '& Start Doc &' or
df.index[j - 1] == '& Start Tweet &' or df.index[j - 1] == '& End Tweet &' or df.index[j -
1] == '& End Doc &' or df.index[j - 1] is None):
         else:
           temp_index = df.index[j - 1]
            if "_-_"  in wordbefore:
               temp_index = wordbefore.split('_-_')[1]
            df.iloc[j]['1 WB Letter'] = len(temp_index)
            df.iloc[j]['1 WB Is Capital'] = temp_index.istitle()
            df.iloc[j]['1 WB Is All Capital'] = temp_index.isupper()
            if (countVowels(temp_index) > 0):
               df.iloc[j]['1 WB Cons Vow Ratio'] = countCons(temp_index) /
countVowels(temp_index)


……
from sklearn import svm
import configparser
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score
from sklearn.tree import ExtraTreeClassifier
from sklearn.tree import DecisionTreeClassifier
def main():
   model_path = "./word2vec_twitter_model/word2vec_twitter_model.bin"
model = gensim.models.Word2Vec.load_word2vec_format(model_path, binary=True,
unicode_errors='ignore')
df_train = expFile('microposts2016-neel-training_neel.txt', 'POS_Train.txt', model)
   df_Xtrain = df_train[0]
   df_Xtrain_array = df_Xtrain.values
   X = df_Xtrain_array[:, 0:12]
   Y = df_train[1]
tree = DecisionTreeClassifier()
   tree.fit(X, Y)
……..
   predictions = tree.predict(X)
   print(accuracy_score(Y, predictions))
   print(confusion_matrix(Y, predictions))
print(classification_report(Y, predictions))
print(f1_score(Y, predictions, average='micro'))
```

BIOGRAPHICAL SKETCH

Mete Taşpınar was born in Mersin on August 29th, 1980. In 1999, he graduated from İçel Anatolian High School in Mersin. He began his undergraduate studies in 1999 at Galatasaray University Computer Engineering Department. In 2005, he received his BSc degree in Computer Engineering. He is currently pursuing his MSc degree in the same university. Presently, he is working as a Software Developer at Yapi Kredi Bank. Before, joining to Yapı Kredi Bank, he worked in NCR, TUBITAK, Vistek Isra Vision and Akbank.

PUBLICATIONS

Mete Taşpınar is the first author of the paper entitled "A Feature Based Simple Machine Learning Approach With Word Embeddings to Named Entity Recognition on Tweets" which was published in the Proceedings of the 22nd International Conference on Natural Language & Information Systems (NLDB 2017). The conference was held in University of Liège, Liège, Belgium on 21 – 23 June 2017.