TWEET RECOMMENDATION UNDER USER INTEREST MODELING WITH NAMED ENTITY RECOGNITION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DENİZ KARATAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

AUGUST 2014

Approval of the thesis:

TWEET RECOMMENDATION UNDER USER INTEREST MODELING WITH NAMED ENTITY RECOGNITION

submitted by **DENİZ KARATAY** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Pınar Karagöz _____
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Ahmet Coşar
Computer Engineering Department, METU _____

Assoc. Prof. Dr. Pınar Karagöz
Computer Engineering Department, METU _____

Assoc. Prof. Dr. Tolga Can
Computer Engineering Department, METU _____

Assoc. Prof. Dr. Osman Abul
Computer Engineering Department, TOBB UET _____

Dr. Ruket Çakıcı
Computer Engineering Department, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: DENİZ KARATAY

Signature :

ABSTRACT

TWEET RECOMMENDATION UNDER USER INTEREST MODELING WITH NAMED ENTITY RECOGNITION

Karatay, Deniz
M.S., Department of Computer Engineering
Supervisor : Assoc. Prof. Dr. Pınar Karagöz

August 2014, 62 pages

Twitter has become one of the most important communication channels with its ability of providing the most up-to-date and newsworthy information. Considering wide use of Twitter as the source of information, reaching an interesting tweet for a user among a bunch of tweets is challenging. As a result of huge amount of tweets sent per day by hundred millions of users, information overload is inevitable. In order for users to reach the information that they are interested easily, recommendation of tweets is an essential task. To extract information from this large volume of tweets, Named Entity Recognition (NER), is already being used by researchers. Commonly used NER methods on formal texts such as newspaper articles are built upon on linguistic features extracted locally. However, considering the short and noisy nature of tweets, performance of these methods is inadequate on tweets and new approaches have to be generated to deal with this type of data. Recently, tweet representation based on segments in order to extract named entities has proven its validity in NER field. Along with named entities extracted from tweets via tweet segmentation, user's retweet and mention history, and followed users are also considered as strong indicators of interest and a model representing user interest is generated. Reducing Twitter users' effort to access tweets carrying the information of interest is the main goal of the study, and a tweet recommendation approach under a user interest model generated via named entities is presented.

Keywords: Named Entity Recognition, Tweet Segmentation, Tweet Classification, Tweet Ranking, Tweet Recommendation

ÖZ

VARLIK İSMİ TANIMA İLE KULLANICI İLGİSİ MODELLEMEYE DAYALI TWEET TAVSİYE YÖNTEMİ

Karatay, Deniz
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Pınar Karagöz

Ağustos 2014 , 62 sayfa

Twitter, haber değeri taşıyan güncel bilgiyi sağlaması özelliğiyle en önemli iletişim kanallarından birisi olmuştur. Bilgi kaynağı olarak Twitter'ın yaygın kullanımı düşünüldüğünde, kullanıcının çok sayıda tweet içerisinden kendisi için ilginç olana erişmesi zordur. Milyonlarca kullanıcı tarafından gönderilen çok sayıdaki tweet sonucunda, aşırı bilgi yüklenmesi kaçınılmazdır. Kullanıcıların ilgilerini çeken bilgiye kolayca ulaşabilmesi için, tavsiye etme gerekli bir işlemdir. Varlık İsmi Tanıma, büyük hacimdeki verilerden bilgi çıkartmak için araştırmacılar tarafından kullanılmaktadır. Resmi makaleler üzerinde sık kullanılan Varlık İsmi Tanıma yöntemleri yerel bir şekilde çıkarılmış dilbilimsel özelliklere dayalıdır. Fakat tweet'lerin kısa ve kirli yapısı düşünüldüğünde, bu yöntemlerin performansları yetersizdir ve bu tip verileri ele almak için yeni yöntemler oluşturulmalıdır. Yakın geçmişte, varlık ismi çıkartmak için parçalara dayalı tweet simgeleme yöntemi Varlık İsmi Tanıma alanında geçerliliğini kanıtlamıştır. Tweet parçalama yöntemi ile tweet'lerden elde edilen varlık isimleri ile birlikte kullanıcının retweet ve mention tarihçesi ve takip ettiği kullanıcılar, kullanıcı ilgisinin güçlü göstergeleri olarak görülmüş, ve kullanıcı ilgisini temsil eden bir model oluşturulmuştur. Bu çalışmanın amacı, Twitter kullanıcılarının ilgilerini çeken bilgiyi taşıyan tweet'lere ulaşırken sarfettikleri çabayı azaltmaktır, ve varlık isimleri ile oluşturulmuş bir kullanıcı ilgisi modeline dayalı tweet tavsiye etme yöntemi sunulmuştur.

Anahtar Kelimeler: Varlık İsmi Tanıma, Tweet Parçalama, Tweet Sınıflandırma, Tweet Derecelendirme

Let there be free speech.

ACKNOWLEDGMENTS

First of all, I would like to thank to Pınar Karagöz for her supervision and guidance through the development of this thesis.

I am very lucky to have my beloved family always supporting me. I would like to thank my dear mother Semra Karatay and my dear father Kürşat Karatay for their love, sacrifices and care.

I would like to thank my darling Miraç Parlatan for his continuous support and never ending patience. He has been the absolute source of joy and light in my life.

I feel very privileged to have my friends Müge Ergene, Pınar Çetin, Başak Meral, and Esra Okumuş, and must acknowledge them for their support and encouragement during my thesis study. I also thank to my colleagues at work and the volunteers who has taken part in the experiment part of the study.

Finally, I would like to thank my company ASELSAN for supporting my thesis. In addition, I would like to acknowledge the financial support by TÜBİTAK 2210 Scholarship Program.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

API	Application Programming Interface
AU	Active Users
CG	Cumulative Gain
CoNLL	Conference on Computational Natural Language Learning
CRF	Conditional Random Field
DCG	Discounted Cumulative Gain
HMM	Hidden Markov Models
HTTP	Hyper-Text Transfer Protocol
IDCG	Ideal Discounted Cumulative Gain
IE	Information Extraction
IU	Inactive Users
JSON	JavaScript Object Notation
kNN	k-Nearest Neighbours
LN	Length Normalization
MUC	Message Understanding Conference
nDCG	Normalized Discounted Cumulative Gain
NER	Named Entity Recognition
NLP	Natural Language Processing
OAuth	Open Standard for Authorization
PMI-IR	Pointwise Mutual Information and Information Retrieval
PMI	Pointwise Mutual Information
POS	Part of Speech
REST	Representational State Transfer
SCP	Symmetrical Conditional Probability
SGML	Standard Generalized Markup Language
SL	Supervised Learning
SSL	Semi-supervised Learning
UL	Unsupervised Learning

URI	Uniform Resource Identifier
XML	Extensible Markup Language

CHAPTER 1

INTRODUCTION

In recent years, virtual communities and networks on Internet are adopted so well that even a term is derived for them: Social Media. Social media forms such as social networking, or microblogging, allow people to create, and share any kind of information and ideas. A service that embodies both social networking and microblogging, Twitter, has already settled into our lives, although it is a relatively new type of social media.

Twitter has become one of the most important communication channels with its ability of providing the most up-to-date and newsworthy information. Considering the 255 million monthly active users, and given the fact that 500 million tweets are sent per day [57], there lies a treasure for information extraction researchers and it attracts attention of not only academics but also organisations to extract user interests.

In this study, a system that recommends tweets according to the interests of the users is presented. A user interest model is generated where user interests are defined by means of relationship between the user and his friends as well as named entities extracted from tweets.

1.1 Motivation

Considering wide use of Twitter as the source of information, reaching an interesting tweet for a user among a bunch of tweets is challenging. As a result of huge amount of tweets sent per day, information overload is inevitable. In order for users to reach

the information that they are interested with ease, recommendation of tweets is an important task.

To extract information from this large volume of tweets generated by Twitter's millions of users, Named Entity Recognition (NER), which is the focus of this thesis, is already being used by researchers. Named Entity Recognition can be basically defined as identifying and categorising certain type of data (i.e. person, location, organisation names, date-time and numeric expressions) in a certain type of text. On the other hand, tweets are characteristically short and noisy. Given the limited length of a tweet, and restriction free writing style, named entity recognition on this type of data become challenging. Commonly used NER methods on documents written grammatically correct such as newspaper articles build upon linguistic features extracted locally such as capitalisation, POS tags of previous words. Considering the fact that tweets generally include grammar mistakes, misspellings, and illegal capitalisation, performance of the traditional methods is incompetent on tweets and new approaches have to be generated to deal with this type of data. Recently, tweet representation based on segments in order to extract named entities has proven its validity in NER field [34, 33].

In this thesis, it is aimed to reduce the Twitter user's effort to access to the tweet carrying the information of interest. By using tweet segmentation for named entity recognition on tweets, a tweet recommendation method under a user interest model generated via named entities is presented.

1.2 Contribution

The main contributions of this thesis are as follows:

- Named Entity Recognition studies on Turkish are very few, and all of the studies carry out supervised or semi-supervised methods based on morphological features [49, 7, 12, 55, 4, 44, 53]. In addition, only one NER study among them is evaluated on Twitter data [7]. This study focuses on an unsupervised NER method independent from morphological features. To the best of author's knowledge, this is the first work to study NER on Turkish tweets via tweet

segmentation.

- Although tweet recommendation problem is studied widely [60, 24, 19, 8], to the best of author's knowledge, tweet recommendation problem has not been studied on Turkish tweets via a named entity based method before. Tweet ranking problem has also not been studied with a named entity recognition based method.

- To achieve our goal, a graph based user interest model is generated via named entities extracted from user's friends' and user's own posts. In the user interest model, each included friend is ranked relatively based on their interactions with the user via retweets and mentions, and named entities are scored via ranking of the user posting them. Forming a NE representation of a Twitter user's profile, a novel tweet recommendation approach is presented on Turkish tweets.

1.3 Thesis Organisation

The rest of the thesis is organised as follows:

- Chapter 2 presents the related work on Named Entity Recognition, and Tweet Recommendation. In this chapter brief information about Twitter and used libraries in this study is also given.

- In Chapter 3, proposed method for tweet recommendation is explained. In this chapter, the overall design of the tweet recommendation system is presented and data gathering, knowledge base construction, named entity recognition, user interest model generation and finally recommendation phases, are explained in detail.

- In Chapter 4, experiments conducted to evaluate the performance of the system is presented. Experiments are conducted in an iterative manner: First, results of the experiment performed to evaluate the accuracy of the NER task is presented. Then the experiment conducted for user interest model generation to find the best values for parameters and create the most efficient user interest model is

given. Finally, recommendation performance of the system is evaluated, and results are presented in Chapter 4.

- Finally, conclusion is given by summarising the study. Powerful aspects and potential drawbacks of the proposed method is explained, and future work is discussed in Chapter 5.

CHAPTER 2

BACKGROUND AND RELATED WORK

In order to pursue this study, two main concepts, named entity recognition and tweet recommendation has to be explored. In addition, background information on used technologies has to be given. For this purpose, first in Section 2.1, general information on Twitter is given. In Section 2.2, overview of named entity recognition with its origin, factors that affect the process and learning methods with the studies they are followed are given. In Section 2.3, named entity recognition on tweets in non-Turkish languages are presented. In Section 2.4, named entity recognition on Turkish texts including tweets are presented. Finally, in Section 2.5, studies on tweet recommendation are given. Finally, in Section 2.6, used libraries and global context is explained.

2.1 General Information on Twitter

Twitter, created in 2006, is an online social networking and microblogging service which is worldwide popular with more than 255 million monthly active users who post 500 million tweets, handling 1.6 billion search queries per day [57]. Since this study focuses on *Twitter*, general information and important concepts to comprehend the study is given in this section.

Twitter enables its users to post and read text messages that are 140-character long, and these messages are called *tweets*. *Tweets* are the basic atomic building block of all things *Twitter* and are also known more generically as *status updates*. *Tweets* can be *favorited* by a user. Every user in *Twitter* has a unique *username*. Users may sub-

scribe to other users' tweets and this act is called as *following* where unsubscribing a subscribed user is called *unfollowing*. *Twitter* is centered on the concept of *following* which is taken into consideration in this study widely. Subscribers of a user are known as *followers* where users subscribed by a user are known as *friends*. *Friends*' posts appear in reverse chronological order on the users' *home timeline*. More than one *Friend* will result in a mix of *tweets* scrolling down the page, which gets confusing for the user as the number of *tweets* increases. *Retweeting* is another important concept which is the act of forwarding a tweet which enables users to share it with their own *followers*. There are two other concepts in Twitter which are activated via symbols. Users can group posts together by keywords by use of *hashtags* where words or phrases prefixed with "#" sign. Similarly, users can refer or reply to other users using "@" sign followed by a *username*, which is called *mentioning* [58].

Tweets are publicly visible to anyone by default, but users can lock their account and restrict message delivery to just their *followers*. In this study, only publicly visible tweets are included, content - username relationship is kept confidential and personal privacy is not violated.

2.2 Overview of Named Entity Recognition (NER)

2.2.1 NER Origins

Although it is currently recognized as one of the most significant sub-tasks of Information Extraction (IE), the term *named entity* was first derived for the Sixth Message Understanding Conference (MUC-6) which was focused on Information Extraction tasks [43]. Newspaper articles considered as unstructured text were the main source to extract structured information. During these studies, it is noticed that recognising units of information such as names, and numeric expressions is essential. Thus, MUC-6 conference committee developed a named entity task to fulfil the need as a short term subtask and defined it as identifying names of all the people, organisations, geographic locations, time, currency and percentage expressions [23]. To classify the names, SGML markup tags were used. Therefore, ENAMEX tag used for people, organization and location names, where NUMEX tag is used for currency, percentages

and time. Although there are studies related to the area, this conference is the first to define the concept on a solid basis with every aspect, and then it is started to be recognized as *Named Entity Recognition* [43].

The very first research paper in the area known is Lisa F. Rau's company name extraction study [43]. The study focuses on company names for their significance to the knowledge-based financial applications. Instead of using a list of all the names of companies in the world, the study tries to recognize and automatically extract company names from natural language text. This study recognizes company names on a handcrafted rule based manner by means of capital letters, stop words, company name related conjunctions and segmentation. It also tries to generate potential referring expressions for the company names. In this manner, this simple study laid the foundation of this specific area [46].

2.2.2 NER Factors

Named Entity Recognition strategies vary on basically three factors: Language, textual genre and domain, and entity type. **Language** is very important because language characteristics affect approaches. It is not difficult to guess that a major portion of the studies is on English language; however, languages such as German, Spanish, and Dutch are well studied in CoNLL conferences. In addition, Chinese, Japanese, French, Greek, Italian, Bulgarian, Polish, Romanian, Korean, Swedish, Portuguese, Arabic and Turkish are among the languages studied. However, porting a named entity extraction system into another language is challenging. Borthwick tested the same system on both English and Japanese texts, and faced challenges adapting the system. Even after adapting the system to the new language, the success rate decreased since a named entity extraction system's capabilities are highly dependent on the language [6]. Since a method based on a language is usually failed on another, a considerable portion of the studies are on language independence and multilingualism. Cucerzan and Yarowski noticed this gap at the early stages of NER studies, and studied on language independent named entity recognition. Their study is tested on Romanian, Greek, English, Turkish, and Hindi [12].

Textual genre is another concept whose effects cannot be neglected. NER system

designs are highly dependent on whether the text is journalistic, scientific, informal, formal, email etc. [43]. Very few studies are specifically dedicated to diverse genres. The study of D. Maynard et al. is one of the leading studies on diverse genres. In this study, a system with its capability for processing texts from widely differing domains and genres is designed for emails, scientific texts and religious text, aiming to eliminate the cost of adaptation of designed systems to different applications. Although the results were promising, the system was not automated and there needed to be made numerous alterations in order to support varying text genres [40]. When E. Minkov et al. created a system specifically designed for email documents; there were few studies on NER for informal documents. They finally needed to define a set of specialized structural features for email text genre to improve performance [42]. Also in a study conducted on Turkish, CRF based named entity approach is first conducted on news media data, and then real data such as twitter data. It is seen that same approach for different data sets is not effective [49, 7]. These experiments show that although different domains can be supported as well, still there is major challenge of migration of a system to a different textual genre [43]. Nevertheless, there are also studies on domain independent named entity recognition. Liao and Veeramachaneni's study is domain and data independent. The data is not required to be from the same domain as the one in which are interested to tag NEs [35].

The word *Named* in the expression *Named Entity*, aims to limit the task to only those entities for which one or many rigid designators, as defined by S.Kripke, stands for the referent [29]. In modal logic and the philosophy of language, a term is said to be a rigid designator when it refers to the same thing in all possible worlds in which that thing exists, and does not refer to anything else in those possible worlds in which that thing does not exist [21]. For instance, *the capital city of the Republic of Turkey* is only referred to *Ankara*. The most studied named **entity types** are types of proper names, names of persons, locations and organisations. These types are known as ENAMEX as mentioned above. In addition, temporal expressions (TIMEX) and numerical expressions (NUMEX) such as amounts of money, and percentage are also considered as named entities. Although some TIMEX and NUMEX instances types are good examples of rigid designators, there are also many invalid ones. For example, *the year 2014* is the 2014st year of the Gregorian calendar whereas *in June*

refers to the month of an undefined year. For practical reasons, NE definition is arguably loosened in such cases. Therefore the definition of the term named entity is not strict and often has to be explained in the context it is used [62]. From the beginning of the studies to the present day, one can observe that a major portion of the early studies tries to extract a variety of named entities [40, 5, 50, 6] within the same system. Over time, the effect of the entity type differences to a named entity extraction system is noticed, and studies focused on a monotype are also conducted. For example a study [14] composes training data for a specific entity type, and extracts named entities of this type using a specifically created filter.

2.2.3 NER Learning Methods

In the absence of training examples, generation of handcrafted rules was the preferred technique in early studies. Then using a set of training examples, the studies tend to ground on supervised machine learning (SL) methods in order for rule-based systems or sequence labelling algorithms to be induced automatically. Then the main drawback of SL is noticed: the need for a large annotated corpus. The lack of such resources and the cost of creating them give rise to two alternative learning methods: semi-supervised learning (SSL) and unsupervised learning (UL), and currently researches on NER tend to prefer these methods [43].

The commonly used technique to solve NER problem is **supervised learning**. In supervised learning techniques, the features of valid and invalid NE instances are studied by using annotated large volume of documents and rules which are designed to recognise instances of a given type. Typical SL approach system completes the following phases: a large annotated corpus is read, lists of entities are memorized, and disambiguation rules based on discriminative are created. Tagging words of a test corpus when they are annotated as entities in the training corpus is often accepted as a baseline SL method, and the performance of the system is evaluated on the basis of percentage of words that appear in both training and testing. Hidden Markov Model (HMM) as a supervised learning technique is widely used in the studies. A work based on HMM is an extraction of product names on Chinese free text [36].The study of Bikel et. al. makes use of HMM. In this study, person names, organization

names, location names, times, dates, percentages, and money amounts are extracted [5]. Sekine also extracted same named entity types, but this time by using decision trees [12]. Maximum Entropy approach is another supervised learning technique used for named entity extraction. Borthwick tried to extract ENAMEX along with TIMEX and NUMEX using maximum entropy on English and Japanese texts [6]. Another study tries to improve Borthwich's work, and presents a more effective extraction system again using maximum entropy [9]. Isozaki and Kazawa studies on Japanese texts using support vector machines and their study tries to extract person names, organisation names and date [3]. Asahara and Matsumoto also studies on Japanese texts, making use of character level information on a support vector machines based method on Japanese texts, extracting person, organisation, location names, date and times, money and percentage [26]. McCallum and Li tried to extract names of locations, organisations, and persons in English and German texts. In this study, Conditional Random Fields (CRF) based learning is used [41]. Eryigit and Seker also used CRF techniques in order to extract names of person, location and organisation, but they focused on Turkish news texts [49]. Eryigit et. al. also tested their CRF system for real data: twitter data, forum posts, and speech text [7]. The main challenge of using supervised techniques is that for a system to give good results, there needs to be a large number of rules, and large labelled data. In addition, training data should not include redundant data since it can mislead the recognition process.

Semi-supervised learning is actually another set of supervised learning tasks falling between unsupervised learning (without any labelled training data) and supervised learning (with completely labelled training data) and techniques. The difference is that semi supervised learning methods make use of both labelled and unlabelled data in conjunction for training and improvement in learning accuracy is obtained via this approach. Bootstrapping is the main technique for semi-supervised learning, and requires a small amount of supervision, in other words a set of seeds, to initiate the process of learning. Cucerzan and Yarowski made use of bootstrapping algorithm based on iterative learning that learns from unannotated text in their study. Their algorithm achieved competitive performance even when trained on a very short labelled name list, even without requiring other language-specific information, tokenizers or tools [12]. Collins and Singer also discussed the use of unlabelled data in their stud-

ies. They showed that unlabelled data usage can reduce the requirements (large number of rules, and labelled examples) for training a classifier to little number of seeds [10]. Liao and Veeramachaneni presents a simple semi-supervised learning algorithm using conditional random fields and they claim that any other model can be easily incorporated to their framework. Although their algorithm requires a small amount of labelled training data as the other works that makes use of semi supervised techniques do, the data is not required to be from the same domain [35]. The main challenge using semi-supervised techniques is unlabelled data selection. To obtain the best results, selection of documents rich in proper names are usually preferred [28].

Unsupervised learning is a set of methods trying to find hidden structure in unlabelled data and clustering is the typical approach of this technique. Generally, the techniques are based on a large unannotated corpus statistics. Alfonseca & Manandhar's study deals with labelling an input word with an appropriate NE type. Using a list of words that frequently co-occur with it in a large corpus, they try to assign a topic signature[1]. In Evans's study, the method for identification of hyponyms/hypernyms is applied in order to identify potential hypernyms of sequences of capitalized words appearing in a document [18]. Similarly, Cimiano and Völker adapted the same approach, but also number of occurrences is added to the feature set [61]. Shinyama and Sekine grounded their study on the observation that news articles consist of synchronously appearing named entities on the contrary of common nouns. Their study lead them to the fact that appearing in multiple news sources at the same time is a strong indication of being a named entity. Rare named entities can be identified via this technique in an unsupervised manner [52]. Etzioni et al. create features for each candidate entity and a large number of automatically generated discriminator phrases [17]. In Etzioni et al.'s work, Pointwise Mutual Information and Information Retrieval (PMI-IR) technique is used as a feature to determine that a named entity can be classified under a given type. PMI-IR measures the correlation between two expressions where high PMI-IR means that expressions lean towards to co-occur[56].

2.3 NER on Tweets

Although there are several studies with various methods in the literature, as mentioned above, textual genre is a main factor that affects NER progress. A successful study when adapted, usually does not result good on another type of data. Therefore, studies specifically made on Twitter has to be examined. Although in early studies formal texts such as news data is preferred, with the rise of social media, researches on real data, specially on Twitter, gained speed and explored below.

Ritter et al. adapted classic NLP pipeline to learn with tweets in English. As a typical supervised approach, for part-of-speech tagging, chunking and named entity recognition processes, they used a previously tagged out of domain text, tagged tweets, and unlabelled tweets [48]. Oliveira et al. adapts a supervised approach based on CRF, and twitter stream is controlled via number of filters. Each filter is responsible to recognize entities to some specific criteria [14]. Although the system extracts information in real-time, each filter has to be trained separately with convenient training data, which brings a huge labelling effort on number of training sets, and a complex feature selection phase. Although the distributed approach of using needed filters is good idea for specific purposes, in order to recognize all types as it is aimed in this study, all of the filters has to be used in conjunction, which requires a huge effort. To sum up, supervised learning approaches are strictly dependent on the used NLP technique, need a successful morphological analyzer as well as a large annotated tweet corpus. English is a well studied language in NLP, but Turkish is not. These are the reasons that suspend this study from this type of approaches, and head towards to the unsupervised methods.

Although unsupervised methods are more preferable for real data, there is a benefit in not skipping a remarkable semi-supervised based study. Liu et al. compose a semi-supervised system based on CRF and the k-Nearest Neighbours (kNN) algorithm. Tweets are labelled in a word level using kNN, linear CRFs are applied in order to compose a detailed classification [37]. The combination of two approaches, on the other hand, increases complexity, and feature selection becomes a major challenge since to maintain both systems function together, a satisfactory combination is needed. In addition, although the approach is not completely supervised, still suffi-

cient amount of labelled data is needed.

On the other hand, Li et al. seized upon unsupervised approaches and studied name entity recognition on targeted twitter stream. Targeted twitter stream is a set of tweets filtered according to user-defined selection criteria, and it is usually used to understand user opinions about a product, or an organization etc. In this study, a novel unsupervised NER system for a specific twitter stream is presented. The system makes use of global contexts in order to split tweets into meaningful and valid segments, and creates the set of candidate named entities. Then, named entities are ranked according to the local context of the stream in order to validate whether a candidate is a true named entity or not [34]. This system does not require any labelled data, or knowledge of linguistic features, as an advantage. Unfortunately, the study lacks of usage of local information in two ways: First, independent segmentation of tweets causes to lose dependent information because tweets published closely in time largely share the same segments. Secondly, this study ignores linguistic features, but these features are not always useless when recognizing named entities. Comparing to this study, local information in the tweets is needed because a reliable user model has to be constructed, therefore relationship between the tweets cannot be ignored. In addition, as language independent study, since it ignores all linguistic features completely, it fails on Turkish language, and all other agglutinating languages, because this study is on English. English language is more suitable for segmentation approaches since auxiliary verbs, or prepositions are written separately. Furthermore, to validate candidates as a named entity, a ranking approach is used. In a targeted twitter stream, it is very useful, since it is for sure that all the tweets mention some issue in common. However, this study is based on user, and the source tweets cannot be guaranteed to mention same concepts.

Li et. al furthered their studies in the light of their previous work, and improved it by combining global context from external knowledge bases and local context information hidden in the tweets. As an addition, tweets within a time space from different users are evaluated in conjunction [33]. However, in this study, it cannot be guaranteed that there will be tweets from different users within a same time scope as well as tweets cannot be guaranteed to consist of similar keywords since the data is not a targeted stream.

2.4 NER on Turkish Texts

So far, NER origins, factors, learning methods and NER approaches on tweets are examined. To pursue this study, previous NER studies on Turkish texts have to be examined.

Although some languages such as English, Spanish, Chinese and Japanese are studied well in the scope of named entity recognition [43], studies on Turkish is very few. First known study on Turkish is conducted by Cucerzan and Yarowski. They proposed a language independent named entity recognizer and it is evaluated on Turkish texts along with other texts in Romanian, English, Greek, and Hindi [12]. In Tür et al. [55], a statistical information extraction system on Turkish newspaper texts is presented. A person name extractor is proposed in Bayraktar et al. [4] for financial news articles, based on the determination of local patterns. In Tatar and Cicekli [53] an automatic rule learning method that exploits different features of the input text to identify the named entities located in the Turkish news articles is described. A rule based named entity recognizer for Turkish news texts is described in Kucuk and Yazici [30] which employs a set of lexical resources and sets of rules as information sources. Kucuk and Yazici furthered their study and presented a hybrid named entity recognizer for Turkish texts [31]. Eryigit et al. [49] proposed a successful CRF based named entity recognizer on Turkish news data. Although there are remarkable results, all of the studies on Turkish mentioned above is all on formal texts, mostly on news texts. However, this study focuses on Twitter data, tweets, which are highly short and informal texts.

In Ozkaya and Diri [44], person, organization and location names are extracted from Turkish e-mails. Although there are official e-mails as well as personal e-mails in the data set, this study can be evaluated as the first study on informal Turkish texts to the best of the author's knowledge. However, the study relies on rule-based methods and very dependent on the textual genre. Eryigit et al. furthered their study [49], and evaluated their system on informal Turkish texts such as forum posts, tweets, and speech texts. Although it underlines the challenges of named entity recognition in Turkish informal texts, this study could not go beyond an experimental study on real data since the same approach for formal texts is adapted [7].

2.5 Tweet Recommendation

In this study, the focus is suggesting a new approach on recommending tweets that users are interested in, in the pursue of letting the users to reach the tweets that they interest easily. Suggested tweet recommendation in this study is based on tweet ranking. Therefore studies on tweet recommendation making use of tweet ranking are examined, and given in this section.

The need for filtering mechanisms emerged as a result of the increasing volume of streaming data on microblogs such as Twitter. Users are exposed to large amount of texts in order to reach the information of interest. Tweet ranking is a task to address this problem, and a way of filtering vast amount of data. In Uysal and Croft [60], user's retweet behaviour is used to put more relevant tweets forward via classification of tweets as retweetable or not. For classification, there are author-based features such as user's follower count, statuses count, favourites count, tweet-based features such as containing hashtags or mentions, retweet status, length, content-based features such as novelty of the tweet compared to the other tweets that appear on the user's timeline, and user based features such as the relationship between user and the author of the tweet. Although this study is user oriented as this study, features used are all independent from each other, and the context of the tweet related with the relationship of tweet's author and the user is not considered. In Huang et al.[24], heterogeneous network based tweet ranking is studied by harnessing linkages between tweets and semantically-related web documents. In this study, non-informative tweets such as tweets with first personal nouns are eliminated and ranking is done according to informativeness of the tweet based on web documents. This study also ignores the context of the tweet and the user's interest. In Feng and Wang [19], again retweet behaviour is modelled and tweets are tried to be classified as retweetable or not. Term frequencies are used but named entities are not extracted. In the study of Chen et al. [8], collaborative personalised tweet recommendation is presented. The value of a tweet is estimated posted tweets by followees are ranked and the tweets that user is likely to be interested are brought forward. In this study, instead of analysing the user himself, the candidate tweet and the publisher of a candidate tweet to be recommended is analysed and tried to be ranked. Although these studies are remarkable,

none of them focuses on Turkish language.

2.6 External Libraries and Context

2.6.1 Twitter Data Crawling

In this study, user related data are the main input of the system. Therefore user related data, tweets posted and subscribed friends, are needed to be crawled from *Twitter*. In this section, technologies and concepts related to data crawling from *Twitter* are explained.

Twitter's application programming interface (API) is based on the REST (Representational State Transfer) architecture. REST is basically a procedure using simple HTTP calls to reach or manipulate information on a server by means of reading, creating, updating or deleting operations [20]. Web service APIs that are based on the REST constraints are called RESTful and they are basically defined with following aspects: base URI, an Internet media type for the data, and standard HTTP methods. Twitter's API is also a RESTful API, and implements a number of GET and POST methods and returns results in JSON format. In order to reach Twitter via APIs, authentication is necessary. For RESTful API of Twitter, requests sent are needed to be OAuth signed [58].

In this study, Twitter data is not manipulated but only monitored. Therefore only GET methods of Twitter RESTful API are used by means of a Java library called Twitter4j, which facilitates integration of a Java application with Twitter APIs [59].

2.6.2 TS Corpus

For NER task, tweet segmentation approach is adopted in this study. As it is given in Chapter 3, in order to segment a tweet, stickiness values of the possible segments have to be calculated so that the segmentation with maximum score can be found. Stickiness values are suggested to be calculated via their occurrence frequencies in a large corpus. For the corpus need, *TS Corpus* is used in this study.

TS Corpus is a tagged Turkish Corpus containing more than 491 million POSTagged tokens designed to be used for general purposes. *TS Corpus* serves a webpage aiming to combine computational linguistics studies and corpus linguistics studies on Turkish, and the corpus is still in progress [51].

TS Corpus serves a number of corpuses with different indexed documents. In this study, *TS Corpus TweetS* and *TS Corpus Wikipedia* are used. *TS Corpus Wikipedia* is a PosTagged Turkish corpus composed of Turkish Wikipedia Pages where *TS Corpus TweetS* is composed of tweets as it can be understood from their names [51].

In order to get frequencies, HTTP GET requests are sent to the corpus web pages and returned result in the form of XML is parsed.

2.6.3 Graph Databases and Neo4j

Most applications today handle deeply structured data such as networks or in more formal words graphs. The most obvious example of this is social networking sites, such as *Twitter*. Also in this study, two types of structural data is handled: Wikipedia dump consisting article titles with wikilinks, and twitter data consisting user friend relationship among with tweets posted.

Handling structured data in traditional relational databases that store data in tables is unnecessarily difficult and complex considering the relationships. On the other hand, a graph is a powerful way of representing structural data compared to tables. It also allows a more agile development by means of its flexible data structure and can be defined as a collection of nodes and edges that connect pairs of nodes. Nodes and edges can be anything that has a relation. From this point of view, graph databases are databases that use graph theory. In other words, instead of tables to represent information, graph databases use nodes, relationships and key-value properties [2]. Graph databases are ideal for ideal for analysing interconnections and they are considerably faster for associative data sets, which is the main reason why it is preferred in data mining especially on social networks.

Neo4j, implemented in Java, is an open source graph database designed to handle structured data. Although it is a relatively new project, it is currently the most popular

graph database with high-performance graph engine capable of all the features of a mature and strong database. It allows programmers to work with a flexible network structure rather than with strict and static tables in object-oriented manner, while providing fully transactional database. In this study, *Neo4j* library is used to build Wikipedia Graph Database and User Interest Model [11].

2.6.4 Zemberek

Zemberek, an open source Java library, provides morphological analysis and spell checking functions for Turkic languages, especially for Turkish. Along with these features, *Zemberek* provides a function for checking words against typos and suggesting a word instead if the word is not correctly typed [15]. Being the most popular NLP library for Turkish, Zemberek is officially used as spell checker in Open Office Turkish version and Pardus, Turkish national Linux distribution.

In this study, since NER task is tried to be carried out as a language independent process, morphological analysis feature of *Zemberek* is not used. Instead, in data preprocessing phase mentioned in Chapter 3, *Zemberek* is used for correcting purposes. Repeating characters that are used to express a feeling such as exaggerating, or yelling which is common in informal writing style, typos and asciification related problems reasoning from mobile usage are considered as correcting, and handled via *Zemberek* library using its spell checking and suggestion functions.

CHAPTER 3

PROPOSED METHOD

In this thesis, the main goal is to reduce Twitter users' effort to access to the tweet carrying the information of interest. To this aim, a tweet recommendation system under a user interest model generated via named entities is presented. The system mainly involves six phases; data gathering, knowledge base construction, data preprocessing, named entity recognition, user interest model generation based on named entities and finally recommendation. General information on the phases is as follows:

- **Data Gathering** is the process of collecting a Twitter user's data, including user's friends' posts as well as user's own posts. In this phase, user-friend relationship is also extracted and friends' relative ranking is generated as an output.
- **Knowledge Base Construction** is the process of generating a graph-based knowledge base of Turkish Wikipedia article titles and their links to each other, in order to validate named entity candidates generated as an output of Named Entity Recognition phase. Keeping this knowledge base up to date is also included in this phase. Although other phases iteratively follow each other and one's output is the other's input, this phase is independent and conducted in parallel.
- **Data Preprocessing** includes removing unnecessary parts of tweet texts such as mentions, hashtags, smileys, vocatives, links etc. Since informal writing style is commonly adopted in tweets, this phase is also responsible from normalising the tweet text such as getting rid of unnecessarily repeated characters, slang words, correcting asciification related problems.

- **Named Entity Recognition** is the next phase of data preprocessing phase. In this phase, tweet segmentation on preprocessed tweets is carried out by means of global context and segments as candidate named entities are generated. Then, these candidates are validated as named entities or ignored by usage of previously constructed knowledge base of Turkish Wikipedia article titles.

- **User Interest Model Generation** phase is a must. In this phase, using named entities extracted from user's and user's friends' tweets and user-friend relationships, a user interest model is generated. In other words, a Twitter user is represented via weighted named entities.

- **Tweet Recommendation** is the last phase, where two kinds of recommendation applications applied by comparing candidate tweets with the generated user interest model. Tweet classification which is the task of deciding whether a candidate tweet is interesting for the user or not, and tweet ranking which aims to sort tweets from the most recommendable to the least recommendable are performed in this phase.

The general overview of the system architecture can also be seen in Figure 3.1. Java programming language is used for implementation and Eclipse is chosen for the development environment. Neo4j is used for graph databases. Each phase is described in more detail in the following sections.

3.1 Data Gathering

In order to have an opinion about the user, his posts have to be examined. Therefore, using Twitter REST API, all tweets posted by user are crawled first. In this study, we tried to examine the user with not only his posts but also his friends' posts. However, crawling all friends' posts is a huge overload, and misleading since *Twitter* following mechanism does not show an actual interest every time. People sometimes tend to follow some users for a temporary occasion and then forget to unfollow. Sometimes they follow some users just to be informed of, although they are not actually interested in. There are also friends that do not post a tweet for a long time, but still followed by

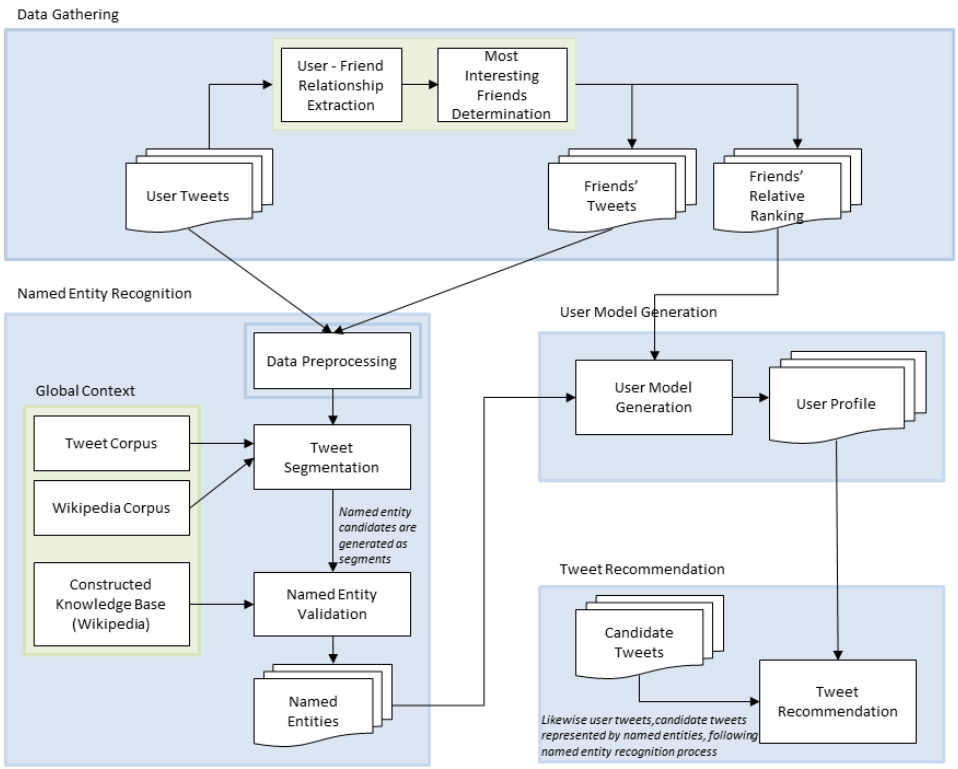

Figure 3.1: System Architecture

the user. Therefore, friends have to be ranked according to the relationship with the user. In this study, regarding retweets, mentions, and last tweet post time, user-friend relationship is tried to be extracted. Every mention and retweet of a friend's tweet makes the friend gain a score relative to the time the action happened. By means of the scores the friends gain, friends are ranked relatively, and only the most interesting friends' posts are crawled. At the end of this process, all of the needed data to generate a user profile in this study; user tweets, friends' tweets, and friends' relative ranking, is acquired.

After top friends' relative ranking is calculated, in order to obtain proportional results in following phases, rankings are normalised and mapped into the range of $[0.1, 0, 9]$ accepting the coefficient of the user himself is 1. The aim of mapping relative rankings to the minimum of $0, 1$ and not to 0 is not to lose any information, because ranking of a top friend may be 0. Normalisation is simply adjusting values measured in a range to a different range in this case. Feature scaling is used to bring all values into the range [0,1] as given in Equation 3.1, which is also called unity-based nor-

malisation, where x represents the value to be normalised in the set X and x' is the normalised value of x. This formula is generalised to normalise the range of values in the dataset X to another range R as given in Equation 3.2 [54].

$$x' = \frac{x - X_{min}}{X_{max} - X_{min}} \tag{3.1}$$

$$x' = R_{min} + \frac{(x - X_{min}) \cdot (R_{max} - R_{min})}{X_{max} - X_{min}} \tag{3.2}$$

Number of friends and tweets to be crawled after ranking the users, are defined via an experiment on choosing the best value for Number of Friends N_F and Number of Tweets N_T parameter, and results are given in detail in Section 4.2.

3.2 Knowledge Base Construction

The adopted method in this study segments the tweets and generates named entity candidates. These candidates have to be validated so that they can be used as an indicator of the user's interest. In this step, Wikipedia is chosen as a reference for a segment to be a named entity, or not. Although there are knowledge bases referencing Wikipedia in other languages such as DBpedia, there is not any up to date knowledge base containing structured information from Wikipedia in Turkish. Therefore, it is constructed from scratch. The latest Turkish Wikipedia dump published by Wikipedia is obtained. Unfortunately, Turkish Wikipedia articles are not proofread as well as English Wikipedia Articles. Therefore, the data is worked over, some wrong or duplicate titles and broken links are corrected. Then, using this dump, a graph-based knowledge base is constructed including the article titles, disambiguation and redirect pages along with wikilinks. Although other phases iteratively follow each other and one's output is the other's input, this phase is independent and conducted in parallel. Even universe is enlarging, so as the constructed knowledge base. The knowledge base is tried to be kept up to date in parallel in order to get actual results since agenda on Twitter is changing rapidly as new occasions emerge.

3.3 Data Preprocessing

For named entities to be extracted successfully, the informal writing style in tweets has to be handled. Before real data has entered our lives, studies on the area were being conducted on formal texts such as news articles. Generally named entities are assumed as words written in uppercase or mixed case phrases where uppercased letters are at the beginning and ending, and almost all of the studies bases on this assumption. However, capitalisation is not a strong indicator in tweet-like informal texts, sometimes even misleading. As the example of capitalisation shows, the approaches has to be changed. To extract named entities in tweets, the effect of the informality of the tweets has to be minimised as possible. To obtain this minimalism, following tasks are applied on the data:

- Links, hashtags, and mentions are removed since they cannot be a part of a named entity.
- Conjunctives, stop words, vocatives, and slang words etc. are removed.
- Although punctuation is not taken as an indicator since tweets are informal, still elimination of punctuation is needed. So, smileys are also removed.
- Repeating characters to express feelings are removed.
- Informal writing style related issues such as mistyping are corrected.
- Asciification related problems are solved since users connecting from mobile devices tend to ignore Turkish characters.

It can be seen that preprocessing tasks can be divided into two logical groups. Pre-segmenting, and Correcting. Removal of links, hashtags, mentions, conjunctives, stop words, vocatives, slang words and elimination of punctuation are considered as pre segmentation. It is accepted that parts in the texts before and after a redundant word, or a punctuation mark cannot form a named entity together, therefore every removal of words is behaved as it segments the tweet as well as punctuation does it naturally. Since tweets are pre-segmented before they are handled in tweet segmentation process, pre-segmentation tasks reduces the complexity of the text and increase

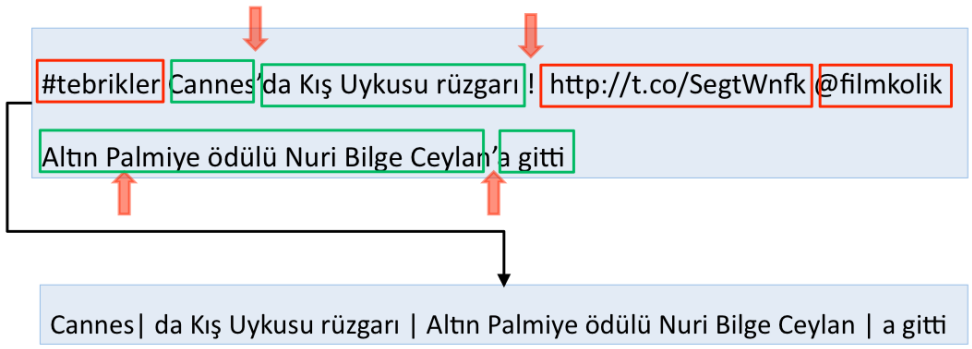

Figure 3.2: An example of tweet preprocessing

performance. On the other hand, removal of repeating characters that are used to express a feeling such as exaggerating, or yelling, handling mistyping and asciification related problems are considered as correcting and can be thought of conversion of tweets from informal to formal. Checking, Deasciification and Suggestion features of *Zemberek* is used for correcting purposes. An example of the result of a preprocessing phase is shown in Figure 3.2.

3.4 Finding NEs - Named Entity Recognition

As informal writing style is adopted in tweets, linguistic features commonly used in earlier methods such as capitalisation cannot be used efficiently as an indicator of a named entity in tweets. However, it is observed that the *correct collocation of a named entity* is still preserved in tweets and named entities can be detected by considering appearance statistics over a Web corpus. In this study, the idea of segmenting a tweet text into a set of phrases, each of which appears more than chance [16, 34] is adopted.

Although such corpus does not exist, computing the probability of being a valid phrase for a segment making use of the entire collection of tweets published in *Twitter* is the ideal case. From this point of view, a quick idea to compute the probability of being a valid phrase is to count a segment's appearance in a very large corpus. Therefore, a corpus serving this purpose in Turkish is needed. Although Li et. Al [34] used Microsoft Web N-Gram corpus, which is based on all the documents in the web to have a good estimation of the statistics of commonly used phrases, it is in the

EN-US market. *TS Corpus* presented in Section 2.6.2 indexes Wikipedia articles, and also Tweets [51]. With these features, TS Corpus fills the gap and fulfills the need of a corpus that gives statistics of commonly used phrases in Turkish.

In this study, these two ideas are combined in a way that statistics are collected; named entity candidates are generated, and finally validated. *TS Corpus* is used in to gather statistical information for various segmentation combinations by means of a dynamic programming algorithm. While collecting statistical information for segment combinations, tweet collection of *TS Corpus* is also used while computing probability of a segment to be a valid named entity, which is different from the previous studies. In this step, capitalisation like local linguistic features of a segment are not used. Instead, *TS Corpus* is used to derive segments, candidate named entities, and the knowledge base that is constructed earlier using Turkish *Wikipedia* dump is used to validate the candidate named entities.

Experiment to evaluate the performance of NER task adopted in this study is conducted and results are given in Section 4.1

3.4.1 Tweet Segmentation

In this section, the core part of named entity recognition method, segmentation, is explained in detail. The aim here is to split a tweet into consecutive segments where a word is not repeated in order to be able to represent a tweet as collocations of words. Each segment contains at least one word, but more than one word is also possible. For the optimal segmentation, the following objective function is used, where *F* is the *stickiness* function, *t* is an individual tweet, and *s* represents a segment.

$$\arg\max_{s_1 \ldots s_n} F(t) = \sum_{i=1}^{n} F(s_i) \tag{3.3}$$

Although the term *stickiness* is generally used for expressing tendency of a user to stay longer on a web page by a user, Li et. al defined it as the metric of a word group to be seen together in documents frequently, or not [34] and it is used in the same meaning in this study. The *stickiness* function basically measures the *stickiness* of a

segment or a tweet represented based on word collocations. A low *stickiness* value of a segment means that words are not used commonly together and can be further split to obtain a more suitable word collocation. On the other hand, a high *stickiness* value of a segment indicates that words in the segment are used together often and represent a word collocation, therefore cannot be further split. In order to determine the correct segmentation, the objective function above is used, where a tweet representation with the maximum *stickiness* acquired by summing the *stickiness* values of possible segmentations is chosen to be the correct segmentation. If a tweet consists of *l* words, then there exists $2^l - 1$ possible segmentations. A straightforward method would iterate all possible segmentations and compute their stickiness, however it is highly inefficient. Therefore a dynamic programming algorithm designed by Li et. al [34] is embraced and adapted to this study to compute stickiness values efficiently. Algorithm 1 explains the segmentation algorithm used in this study, and it is given below.

The algorithm basically segments the longer segment, which can be tweet itself, into two segments and evaluates the *stickiness* of the resultant segments recursively. More formally, given any segment $s = w_1 w_2 ... w_n$, adjacent binary segmentations $s_1 = w_1 ... w_j$ and $s_2 = w_j + 1 ... w_n$ is obtained by satisfying:

$$\arg\max_{s_1, s_2} F(s) = F(s_1) + F(s_2) \tag{3.4}$$

3.4.1.1 Stickiness Measurements

As it can be seen in Algorithm 1, the stickiness function is very significant when deciding the optimal segmentation. A high stickiness value of a segment indicates that continuing on splitting that segment results in a segmentation far from the correct word collocation. Although there are a number of collocation measurements [39, 45], they are all defined for two arguments and designed to measure the collocation of the bigram or the n-grams with the particular binary partition [34]. The framework proposed in [13] is used to define the stickiness functions and the generalised collocation measures of Point Mutual Information (PMI), Dice, and Symmetric Conditional Probability (SCP) are used in this study as explained below.

Algorithm 1: SegmentTweet

Tweet Segmentation in Recursive Manner

input : A tweet $t = w_1 \dots w_l$

output: Segment representation of the tweet $s_1 \dots s_n$

$L_1 \leftarrow null$;

/* L_1 stores the possible segmentations of the tweet */

for $i \leftarrow 1$ **to** l **do**

  $s_{i1} \leftarrow w_1 \dots w_i$;

  $s_{i2} \leftarrow w_{i+1} \dots w_l$;

  $s_i \leftarrow \{ s_{i1}, s_{i2} \}$;

  CalculateStickiness(s_i);

  add s_i to L_1 as a possible segmentation of tweet

  /* try to segment s_{i1} further */

  for $j \leftarrow 1$ **to** $i - 1$ **do**

    /* Form two shorter segments of s_{i1} */

    $s_{i1}^1 \leftarrow w_1 \dots w_j$;

    $s_{i1}^2 \leftarrow w_{j+1} \dots w_i$;

    $L_2 \leftarrow$ SegmentTweet(s_{i1}^1);

    foreach *element* e *of the list* L_2 **do**

      $S \leftarrow$ Concatenate e and s_{i1}^2;

      CalculateStickiness(S);

      add S to L_1 as a possible segmentation of tweet

    end

  end

end

sort L_1 and return $L \in L_1$ with highest score

- *PMI based Stickiness*

  PMI is a measure to calculate the degree of together occurrence of two words more often than by chance. PMI for bigram w1w2 definition is given in Equation 3.5.

$$PMI(w_1w_2) = log\frac{Pr(w_1|w_2)}{Pr(w_1)} = log\frac{Pr(w_1w_2)}{Pr(w_1)Pr(w_2)} \tag{3.5}$$

  Accordingly, PMI is defined by averaging all binary partitions as in Equation 3.7. where base case for segment s consisting only one word is given in Equation 3.6 where $s = w_1 \dots w_n$.

$$PMI(s) = logPr(w) \tag{3.6}$$

$$PMI(s) = log\frac{Pr(w_1 \dots w_n)}{\frac{1}{n-1}\sum_{i=1}^{n-1} Pr(w_1 \dots w_i)Pr(w_{i+1} \dots w_n)} \tag{3.7}$$

  Finally, result set of PMI function is normalised and mapped into [0,1] range and final form of stickiness function *F* is defined as in following Equation 3.8.

$$F(s) = \frac{1}{1 + e^{-PMI(s)}} \tag{3.8}$$

- *Dice based Stickiness*

  Dice is another measurement to calculate the degree of together occurrence of two words more often than by chance. Mathematically, Dice for bigram w1w2 definition is given in Equation 3.9.

$$Dice(w_1w_2) = \frac{2Pr(w_1w_2)}{Pr(w_1) + Pr(w_2)} \tag{3.9}$$

  Dice is defined by averaging all binary partitions as PMI as in Equation 3.11. where base case for segment s consisting only one word is given in Equation 3.10 where $s = w_1 \dots w_n$.

$$Dice(s) = 2logPr(w) \tag{3.10}$$

$$Dice(s) = log \frac{2Pr(w_1 \dots w_n)}{\frac{1}{n-1} \sum_{i=1}^{n-1} Pr(w_1 \dots w_i) + Pr(w_{i+1} \dots w_n)} \tag{3.11}$$

Finally, result set of Dice function is normalised and mapped into [0,1] range and final form of stickiness function *F* is defined as in following Equation 3.12.

$$F(s) = \frac{2}{1 + e^{-Dice(s)}} \tag{3.12}$$

- *SCP based Stickiness*

  SCP, proposed in [13], is designed to measure the *cohesiveness* of bigram w1w2 by considering both conditional probabilities for the bigram given each single term as given in 3.13.

$$SCP(w_1 w_2) = Pr(w_1 w_2 | w_1) Pr(w_1 w_2 | w_2) = \frac{Pr(w_1 w_2)^2}{Pr(w_1) Pr(w_2)} \tag{3.13}$$

  SCP function for segment s is defined as PMI by averaging all binary partitions. Smoothed SCP is given in 3.15 where base case for segment s consisting only one word is given in Equation 3.14 where $s = w_1 \dots w_n$.

$$SCP(s) = 2logPr(w) \tag{3.14}$$

$$SCP(s) = log \frac{Pr(s)^2}{\frac{1}{n-1} \sum_{i=1}^{n-1} Pr(w_1 \dots w_i) Pr(w_{i+1} \dots w_n)} \tag{3.15}$$

  Finally, result set of SCP function is normalised and mapped into [0,1] range and final form of stickiness function *F* is defined as in following Equation 3.16.

$$F(s) = \frac{2}{1 + e^{-SCP(s)}} \tag{3.16}$$

3.4.1.2 Length Normalization

Tweets by their nature contain low amount of long named entities; however, segments of various lengths are handled equally so far. Since they are less in number, longer segments have higher chances of being valid named entities than shorter ones, since global context results favour short named entities considering the stickiness measurements. Therefore, length normalisation given in Equation 3.17 defined empirically by designed by Li et. al [34] is used to favour relatively long segments in TS Corpus.

$$L(s) = \begin{cases} \frac{|s|-1}{|s|} & \text{if } |s| \geq 1 \\ 1 & \text{if } |s| = 1 \end{cases} \tag{3.17}$$

3.4.1.3 Stickiness Function

Finally, combining stickiness measurement function and length normalisation function, the following function to calculate stickiness is obtained:

$$F'(s) = L(s) \cdot F(s) \tag{3.18}$$

Experiment given in Section 4.1 includes the evaluation of performances of the measures explained above to calculate stickiness in terms of NER task as given in Section 4.1.3 and according to the results of the experiment, SCP measure gives better results than other stickiness measurements. The experiment given in Section 4.1 also evaluates Length Normalisation inclusion in terms of NER task success. The results are given in Section 4.1.5, and according to the experiment, LN inclusion gives better results in NER task. Therefore, SCP based stickiness function and, and length normalisation function is used to form stickiness function used in NER task of the system.

3.4.2 Candidate Validation

Thus far, tweets are segmented by means of *SegmentTweet* Algorithm given above, making use of the stickiness function explained in Section 3.4.1.3. In the result of this phase, tweet segments which are candidate named entities are obtained. These candidate named entities have to be validated whether they are real named entities or not, so that they can be used as an indicator of the user's interest. For this purpose, as explained in Section 3.2, Wikipedia is chosen as a reference for a segment to be a named entity, and a graph-based knowledge base based on Wikipedia is constructed.

The constructed knowledge base of Turkish Wikipedia Pages serves as a gazetteer. If the segment, candidate named entity, matches exactly with a Wikipedia title in the constructed knowledge base, then it is accepted to be a named entity easily. However, applied approach in order to validate a segment to be a named entity is not a straight forward string matching algorithm since exact matching of a gazetteer entity and a candidate named entity is not always the case because of two main reasons: First, Turkish is an agglutinative language, therefore there are generally affixes at the end of the word which causes the segments not to match exactly with gazetteer entities such as in the sentence of *Kemal Sunalın tüm filmlerini seviyorum*. In this sentence, there are only two named entities, the actor *Kemal Sunal*, and the name *film* which means *movie* in English. However, segmentation results in the set of *Kemal Sunalın*, *tüm*, *filmlerini*, *seviyorum*. Therefore, *Kemal Sunalın*, and *filmlerini* has to be validated without an affix. Secondly, the writing style adopted in *Twitter* may ignore a word at the beginning or at the end. For example, people generally refer to *Mustafa Kemal Atatürk*, the founder and the first president of Turkish Republic, as *Mustafa Kemal* by ignoring his last name. Since the gazetteer includes the full name, *Mustafa Kemal* cannot be validated with exact string matching.

In order to handle the cases mentioned above, an edit distance is needed to quantify how dissimilar two strings are. Although there are several definitions of edit distance using different sets of string operations such as insertion, deletion and substation exist, one of the most common variants is called Levenshtein distance which makes use of insertion, deletion, and substitution operations [47]. Levenshtein Distance is used for the the string matching task in this study and it can be explained in a more infor-

mal way that representing the distance between two strings as the minimum number of string operations which are single-character edits [32]. However, the approach in this study for string matching as explained further only calculates distances between the strings that one of them definitely contains the other one, substitution operation is not applicable. Therefore, the complexity to calculate the distance reduces. The mathematical formula of Levenshtein Distance is given in Equation 3.19 where a and b are two strings. The Levenshtein distance is computed by filling a matrix $lev_{a,b}$, where i and j denote the matrix indices, and the elements are defined recursively. After the matrix has been filled, the Levenshtein distance is the lower right matrix element $lev_{a,b}(N_a, N_b)$ where N_a and N_b are the lengths of the strings a and b. The characteristic function $1_{(a_i \neq b_j)}$ in the Equation 3.19 is equal to 0 when $a_i = b_j$ and equal to 1 otherwise [32].

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & \textit{if } min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{a_i \neq b_j} \end{cases} & otherwise \end{cases} \tag{3.19}$$

Using the Levenshtein Distance, the following approach is applied to validate candidate named entities: Given S as a segment which is a candidate named entity, and E which is the gazetteer entity, among the pairs ensuring S contains E, or E contains S, Levenshtein Distance of the strings are calculated, and the gazetteer entity E resulting in the smallest Levenshtein Distance with the segment S is accepted to be a named entity.

3.5 Generating User Interest Model based on NEs

At this step named entities with their appearance count in a tweet obtained from friends' posts, and friends' relative ranking obtained in data gathering phase is processed as shown in Figure 3.1. Using these data, a user interest model has to be generated in order to have a reference to have an opinion on the candidate tweets whether they are suitable for recommending or not.

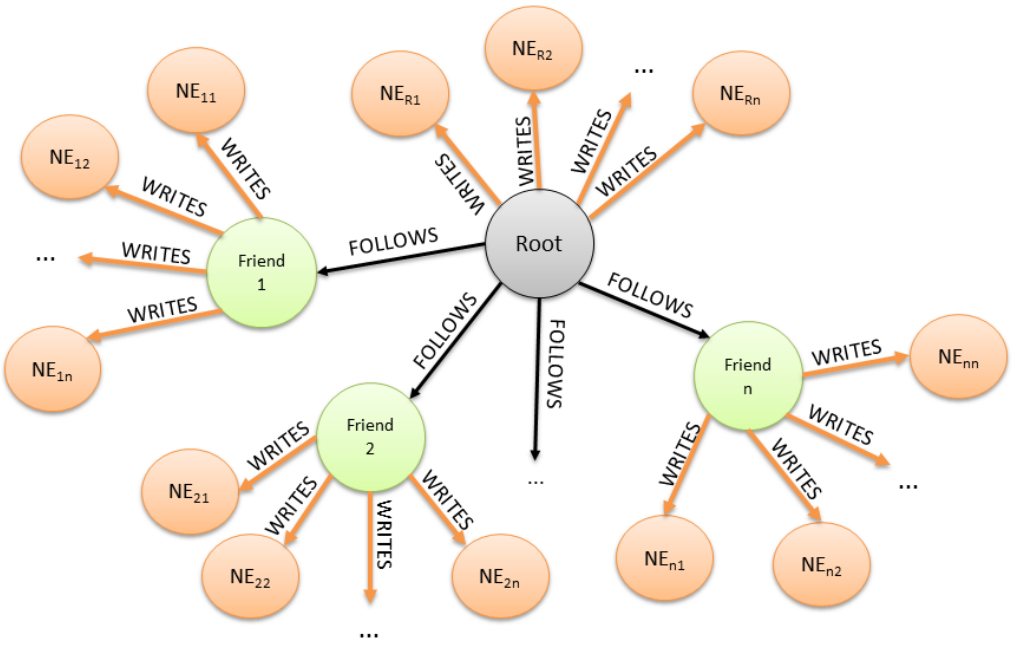

Figure 3.3: Structure of the User Interest Model Graph

User Interest Model is basically a graph based relationship model. Let $G = (V, E)$ be a weighted labelled graph with the node set V and edge set E. Node set V is labelled with the label set L_1 where $L_1 \in \{Root, Friend, NamedEntity\}$ and Edge set E is labelled with the label set L_2 where $L_2 \in \{Follows, Writes\}$. In other words, a user interest model graph has three types of nodes; *Root, Friend, Named Entity*, along with two types of weighted edges; *Writes, and Follows*. Weight of *Writes* edge represents the appearance count of a named entity for a friend's posts where weight of the *Follows* edge represents relative ranking of a friend. Therefore, a twitter profile is represented as *Root* node *Follows* one or many *Friend*s, and a *Friend* node *Writes* one or many *Named Entities*. The structure of the graph is shown in Figure 3.3.

In an ideal world, crawling a user's all posts, all friends and even all friends of friends and so on for user interest model would represent a Twitter profile more realistic, however performance restrictions, and usage habits of Twitter users make this approach inapplicable and misleading for this study. If that were the case, the data set would be huge considering the member number and branching structure in Twitter and it would be impossible to process. In addition, as mentioned in Section 3.1, Twitter users' habits are sometimes misleading. They do not always follow the ac-

counts posting subjects that they are interested. Additionally they sometimes follow a user and do not bother to unfollow even the account is dumped, or posting about irrelevant subjects, due to the large volume of tweets they face every day. Although ranking friends as mentioned in Section 3.1 partly solves this problem, how many of the ranked friends will be included in the model is still an issue. Therefore, for the best user interest model serving our purpose, number of included friends and tweets has to be restricted.

Since user interest model performance can be evaluated based on the classification success, threshold value for tweets to recommend is also important. Therefore, an experiment is conducted on choosing the best values for Number of Friends N_F, Number of Tweets N_T parameters along with Threshold T parameter, and results are given in detail in Section 4.2. The approach of choosing threshold values is also explained in following section.

Ranking quality is another metric for evaluating user interest model performance under different N_F, and N_T values. Therefore, in order to decide N_F, and N_T values, user interest model performances are evaluated in terms of ranking quality and results are given in Section 4.2. Metric for ranking quality, *nDCG*, is explained in following section.

3.6 Tweet Recommendation

In order to recommend tweets, one has to decide whether a tweet is interesting for a user or not, which is a main focus of this study. In this study, defining tweets as interesting or not is achieved by comparing NE representation of the tweet with the generated user interest model. This comparison results in a ranking of candidate tweets. The approach of ranking the candidate tweets in this study results in two kinds of recommendation applications: First application is that candidate tweets are classified and marked as interesting or not, and the interesting ones are defined to recommend. In case of the second application, candidate tweets are shown to the user in the order of interest. The approach for recommendation applications are explained in this section. In addition, the performance of the two recommendation applications

is evaluated, and the results are given in Section 4.3.

First, candidate tweets are processed the same way as tweets used to generate user interest model are processed in Named Entity Recognition phase, and therefore their NE representations are obtained. NE representation of a tweet simply includes the NEs, their appearance count. User interest model is also a NE representation with named entities and their appearance count in terms of each user, but in addition ranking score of the friends of the user is also included. In order to compare the candidate tweet, the user interest model has to be interpreted by including the ranking score factor of the friends. Every friend's named entities and their appearance counts are first multiplied with the friend's ranking, and then summed. Therefore, a set of named entities with their scores based on the user interest model is obtained. The mathematical interpretation to calculate the score of a single named entity is given in Equation 3.20, where SC_{NE} represents the overall score of a named entity, C represents the appearance count of a named entity for a user, n represents the count of friends included in the user interest model, RR represents the relative ranking score of a friend, and U represents the user himself. With the same approach, the final score of all of the named entities appearing in the user interest model is calculated.

$$SC_{NE} = \sum_{i=1}^{n} RR_i \cdot C_i + RR_U \cdot C_U \tag{3.20}$$

After overall score is calculated for all of the named entities in the user interest model, final scores for candidate tweets are calculated in the following approach: Overall score of named entities in NE representation a candidate tweet are multiplied with the appearance count in the NE representation of itself. This operation is done for every named entity in the tweet representation, and then by summing these values, final score of a candidate tweet is obtained. If a named entity in a candidate tweet's NE representation, does not appear in the user interest model, its overall score is accepted as 0 and not taken into consideration assuming the user is not interested in the subject that particular named entity represents. Once final scores for all candidate tweets are calculated, candidate tweets are sorted in a straightforward manner from

the highest to the lowest and therefore tweets are ranked.

$$SC_T = \sum_{i=1}^{m} SC_{NE_i} \cdot C_{NE_i} \tag{3.21}$$

By using Equation 3.21, tweet ranking, which is a must do for this study, is achieved. As mentioned earlier, using this ranking, two recommendation applications are suggested in this study. The recommendation application in which the tweets are shown to the user in the order of interest, is actually achieved by ranking the tweets. However, the ranking quality has to be evaluated and for this reason, an experiment on real Twitter users is conducted to evaluate the performance of the tweet ranking in terms of user interest. Metrics used to evaluate the ranking is explained as following, and details and results of this experiment is given in Section 4.3.

To measure the ranking quality, *DCG* measure is used. *DCG*, *Discounted Cumulative Gain*, is a popular measure and generally used to measure effectiveness of web search engine algorithms or related applications. A search algorithm or a related application returns a result set, and based on a web page's position in the result list, *DCG* measures the gain. Calculation of the gain is achieved via accumulating the gain of each result from the top of the result list to the bottom using a graded relevance scale of web pages in the result set of the search engine [27]. In this study, each individual tweet in the ranked candidate tweet set is treated as a web page, and *DCG* is used to measure the interestingness.

DCG is based on *Cumulative Gain (CG)* whose mathematical formula is given in Equation 3.22. *CG* does not take the position of a result entry into consideration and it is calculated by summing the graded relevance values of all results in the result list, p representing the position of a result entry.

$$CG_p = \sum_{i=1}^{p} rel_i \tag{3.22}$$

DCG, on the other hand, takes the position into consideration, and reduces graded relevance value logarithmically proportional to the position of the result if a highly relevant documents appears lower in the result set, and its mathematical formula is

given in Equation 3.23 [38].

$$DCG_p = rel_1 \sum_{i=2}^{p} \frac{rel_i}{log_2(i)} \tag{3.23}$$

However, comparing the system's performance for different datasets cannot be achieved using *DCG* alone while experimenting since every *PSNL* dataset and user is independent in the experiment in Section 4.3, so the *DCG* value of the tweet ranking task should be normalised using *DCG* value of ideal ordering. Ideal ordering is obtained by sorting tweets in monotonically decreasing order according to the relevance scores given by the user. Normalising with the ideal ordering gives the *nDCG* metric whose formula is given in Equation 3.24 [38].

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{3.24}$$

The other recommendation application classifies tweets as interesting or not for a user. In order for classification, a threshold value has to be determined to take as a reference to classify. Since we are simply dealing with numbers, quartiles and arithmetic mean are chosen as threshold values. Arithmetic mean is the sum of the values divided by the number of items in the sample as given in Equation 3.25 [22]. The quartiles of a sorted set of values in a dataset are the three points that divide the data set into four equal groups, and each group is a quarter of the data.

$$\overline{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} \tag{3.25}$$

The median value which is also the second quartile is the middle value of the data set and the mathematical formula to calculate median index in the dataset is given in Equation 3.26. The middle number between the smallest number and the median of the data set is the first quartile Q_1. Likewise, the middle value between the median and the highest value is the third quartile Q_3. Mathematical formula to calculate indexes of quartiles Q_1 and Q_3 in a dataset is given in Equation 3.27 and 3.28 respectively

[25].

$$Median(Q_2)Index = \frac{1}{2}(n+1) \tag{3.26}$$

$$Q_1Index = \frac{1}{4}(n+1) \tag{3.27}$$

$$Q_3Index = \frac{3}{4}(n+1) \tag{3.28}$$

In order to pick the most suitable threshold for the system, an experiment is conducted whose details and results are given in Section 4.2. In addition, an experiment to evaluate the success rate of classification in terms of tweet recommendation is also conducted and results of this experiment is given in Section 4.3.

CHAPTER 4

EXPERIMENTS

In this chapter, the experimental results of the proposed system are presented. The proposed system is evaluated module by module in an incremental manner. First, Named Entity Recognition module is evaluated in terms of used stickiness function, used corpus and length normalisation inclusion in order to find the best approach to segment a tweet in pursuance of extracting named entities. This module is also compared with different segmentation approaches as baselines and results are given in Section 4.1. Then, as presented in Section 4.2, using the best approach obtained in experiments on Named Entity Recognition Accuracy, an experiment to find best parameters for User Interest Model Generation is conducted. The results of this experiment is given in terms of following parameters: Number of Friends N_F, Number of Tweets N_T, and Threshold T. Last but not least, a set of experiments to evaluate the performance of the Tweet Recommendation phase is performed. Using the best approach to extract named entities, and best parameters to generate the user interest model, classification, and ranking quality accuracy is measured on users having different Twitter usage habits with different type of candidate tweet datasets. This phase's performance is also compared with a baseline method. The results of this experiment are given in Section 4.3.

4.1 Experiments on Named Entity Recognition Accuracy

4.1.1 Approach

Although the major focus of this study is not named entity recognition, since it highly depends on named entities, performance of the named entity recognition phase has to be evaluated.

For this experiment, two types of test data is formed, a small set, and a relatively bigger set. For the small dataset tagged as $NEWS$, newspaper's Twitter accounts' posts are crawled to have a data set on broad subjects. This dataset includes 200 tweets about Soma Mining Disaster and Regional Election of Ankara along with other minor daily news such as a car crash in year 2014. For relatively larger data set tagged as $GEZI$, using Twitter API's search utility, "gezi" keyword is queried, and 1000 tweets are crawled in the purpose of crawling tweets on the subject of Gezi Park Protests which is still popular due to its anniversary although it occurred in year 2013. After eliminating irrelevant, duplicate and non-Turkish tweets, 715 tweets are obtained. Both $NEWS$ and $GEZI$ datasets are human annotated. General information on the datasets are given in Table 4.1 $NEWS$ dataset is used to decide on the NER method, then best method is applied on $GEZI$ dataset to evaluate the system from aspect of data size.

Table 4.1: Test Dataset Statistics of the NER Accuracy Experiments

	Tweet Count	**NE Count**	**Avg. NE per Tweet**	**Avg. NE Length**
$NEWS$	200	804	4.02	2.23
$GEZI$	715	4182	5.85	2.69

In order to prove that the segmentation approach adopted in this study does a real job and generates more accurate named entity candidates, it has to be compared with baseline approaches. Named entity candidates from via baseline approaches are generated and results after validation are obtained.

Named entity recognition method adopted in this study depends on three criteria: Stickiness function, corpus usage, and length normalisation inclusion as mentioned in

Chapter 3. Therefore, the performance of the named entity recognition module has to be evaluated also from these aspects. Segmentation module variations based on these criteria are generated and named entity candidates obtained via these variations are validated. For evaluation, *Precision* and *Recall* metrics are used. Precision is the ratio of correctly found named entities to the found named entities where recall is the ratio of correctly found named entities to real named entities and their formulas are given below. In simple terms, high precision means that substantially more correct named entities than the wrong ones are found, while high recall means that most of the named entities in the test data are found. Precision and recall formulas are given below, where *tp* stands for *true positive* indicating the number of correctly found named entities, *fp* stands for *false positive* indicating the number of wrongly found named entities, and *fn* stands for *false negative* indicating the number of named entities that are wrongly rejected.

$$Precision = \frac{tp}{tp + fp} \tag{4.1}$$

$$Recall = \frac{tp}{tp + fn} \tag{4.2}$$

All results are given in terms of Precision and Recall in Table 4.2 and it can be seen that the best method that gives the best results is achieved when SCP is used as stickiness function on Wikipedia Corpus along with length normalisation. Following subsections discuss the performance of the adopted method via comparison with baseline approaches, comparison of stickiness functions, comparison of corpus usages, and length normalisation inclusion.

4.1.2 Comparison of the Adopted Method with Baselines

A baseline approach has to be defined in order to prove that the segmentation approach adopted in this study makes a difference and generates more accurate named entity candidates. Therefore, two baseline segmentation approaches are determined: Word based segmentation where named entity candidates are simply single words, and segmentation via preprocessing the tweets. Named entity candidates from these

Table 4.2: Performance Analysis of Different Segmentation Methods on Named Entity Recognition in Terms of *Precision* and *Recall*

Segmentation Methods	**Precision**	**Recall**
Word based segmentation	0.557	0.644
Pre-segmentation	0.615	0.306
PMI using Wikipedia Corpus	0.642	0.697
PMI using Wikipedia Corpus and LN	0.699	0.716
PMI using Twitter Corpus	0.588	0.664
PMI using Twitter Corpus and LN	0.608	0.679
Dice using Wikipedia Corpus	0.664	0.701
Dice using Wikipedia Corpus and LN	0.675	0.714
Dice using Twitter Corpus	0.621	0, 637
Dice using Twitter Corpus and LN	0.639	0.647
SCP using Wikipedia Corpus	0, 799	0.821
SCP using Wikipedia Corpus and LN	**0.819**	**0.858**
SCP using Twitter Corpus	0.751	0, 741
SCP using Twitter Corpus and LN	0.789	0.746

baselines are validated and results are obtained.

Word based segmentation basically generates one word length named entity candidates and cannot find word collocations. Since our method is focused on finding the word collocations via their stickiness values along with single word named entities, the difference in the results in terms of precision shows that word collocations can be found via the method adopted in this study. Low precision value of word based segmentation indicates that found named entities are mostly wrong, and this is reasoning from accepting word collocations as more than one named entity where in real there is only one real named entity. This reason also leads to a low recall value, which means most of the named entities in the data set cannot be found correctly.

Pre-segmentation is one of the preprocessing tasks mentioned in Chapter 3. Pre-segmentation which is removal of links, hashtags, mentions, conjunctives, stop words, vocatives, slang words and elimination of punctuation, generates generally long can-

didates containing irrelevant named entities. High precision but low recall value of pre-segmentation compared to the word based segmentation shows that found named entities are mostly true named entities, however, most of the named entities in the data set cannot be found.

Table 4.3: Performance Comparison of Baseline Segmentation Methods with Adopted Method in Terms of *Precision* and *Recall*

	Precision	**Recall**
Word based segmentation	0.557	0.644
Pre-segmentation	0.615	0.306
Our Method (Average)	0.691	0.718
Our Method (Best)	0.819	0.858

As it can be seen from Table 4.3, our method gives better results in terms of both precision and recall even in average of all method variations given in Table 4.2. Our method with best combination which is using SCP stickiness function on Wikipedia Corpus applying length normalisation easily outperforms the baseline approaches. The better results indicate that the method adopted in this study can segment longer segments into correct named entity candidates.

4.1.3 Stickiness Function

To discuss the effect of the stickiness functions, precision and recall values of method variations are averaged in terms of used stickiness functions. As it can be seen from Table 4.4, SCP stickiness function gives better results than Dice and PMI functions on average. Dice stickiness function is better than PMI function at finding correct named entities where PMIs is better at rejecting non named entity candidates. SCP stickiness function outperforms PMI and Dice stickiness functions because they return high values out of proportion for frequent items relative to SCP stickiness function.

Table 4.4: Stickiness Function Comparison in Terms of *Precision* and *Recall*

		Precision	**Recall**
Stickiness Function	**PMI**	0.634	0.689
	Dice	0.649	0.674
	SCP	0.789	0.791

4.1.4 Corpus Usage

To compare the effect of the corpus usage explicitly, precision and recall values of method variations are averaged in terms of used corpus. As it can be seen from Table 4.5, Wikipedia corpus usage gives better results in average. It can also be seen from Table 4.2 that precision and recall values drop suddenly when corpus usage changes while using the same stickiness. In addition, the best method variation that gives the best results uses Wikipedia corpus.

This consequence is reasoning from the characteristics of the Twitter corpus. Twitter corpus is not large as Wikipedia corpus, therefore stickiness values are not as accurate as the ones obtained via Wikipedia corpus. In addition, tweets in Twitter corpus content is collected earlier and it is relatively older than Wikipedia corpus. Therefore recent agenda cannot be captured in Twitter corpus. These two reasons cause performance based on Twitter corpus to go down.

Table 4.5: Corpus Usage Comparison in Terms of *Precision* and *Recall*

		Precision	**Recall**
Corpus	**Wikipedia Corpus**	0.716	0.746
	Twitter Corpus	0.666	0.685

4.1.5 Length Normalisation

To make a clearer analysis on the effect of the length normalisation explicitly, precision and recall values of method variations are averaged in terms of length normalisation inclusion. As it can be seen from Table 4.2, length normalisation increases precision and recall values slightly independent from function selection and corpus

usage. Also in Table 4.6 it can be observed that methods including length normalisation give slightly better results on average.

Without length normalisation, segments with different lengths are treated in the same manner. The focus of the length normalisation task is to favour long named entities. Since it becomes possible to catch long named entities, precision and recall values increase with length normalisation. Considering long named entities are rare in tweets, this increase occurs slightly.

Table 4.6: Effect of Normalisation on Segmentation in Terms of *Precision* and *Recall*

	Precision	**Recall**
Segmentation with LN	0.704	0.726
Segmentation without LN	0.677	0.712

4.1.6 Effect of Dataset Size

To evaluate the system performance independent from the dataset size, a larger dataset *GEZI* is formed. This dataset is nearly 8 times larger than the *NEWS* dataset in tweet size. *GEZI* dataset includes named entities relatively longer than the ones in *NEWS* dataset. Dataset statistics are given in Table 4.1.

Table 4.7: Effect of Dataset Size on Adopted Method in Terms of *Precision* and *Recall*

	$NEWS$		$GEZI$	
	Precision	**Recall**	**Presicion**	**Recall**
Word based segmentation	0.557	0.644	0.537	0.542
Pre-segmentation	0.615	0.306	0.611	0.245
SCP using Wikipedia Corpus and LN	0.819	0.858	0.810	0.826

Since every tweet is handled alone in this NER approach, and there is no training dataset needs as supervised methods do, the size of the dataset does not effect the adopted method performance significantly and adopted method performs on *GEZI* as good as on *NEWS*. Slight differences in adopted method's results are reasoning from the difference in number of named entities and average named entity per tweet. On the

other hand, baseline methods perform worse on *GEZI*, since *GEZI* dataset includes named entities relatively longer than the *NEWS* dataset.

4.2 Experiments on Optimal Parameter Setting for User Interest Model Generation

4.2.1 Approach

In order to generate the most suitable user model for the system, Number of Friends N_F, Number of Tweets N_T parameter values have to be decided. Parameters N_F, N_T values have to be evaluated in terms of both classification and ranking quality. Besides, classification performance cannot be evaluated without determining on a Threshold T value. Since Threshold T parameter's value cannot be decided independently, it is determined in classification based experiment by evaluating in terms of N_F, N_T values.

The approach to find best values for parameters N_F, N_T and T is as follows: The same user data is crawled with many different N_F and N_T values, and for each of them, a User Interest Model is generated. For test data, $GNRL$ dataset, presented in Section 4.3, including tweets from newspaper accounts are crawled in order for test data to be objective for every user profile, and each tweet is marked by the user as interesting or not. Then, tweets are classified for each Threshold T value, compared with the user choices, and match percentages are obtained. This approach for one user is repeated for different users, and the average of the results are taken.

Since every user profile is crawled and User Interest Model is generated many times with different N_F and N_T values, volunteered user for this experiment is a subset of the volunteered users for recommendation experiment presented in Section 4.3. There are 4 volunteer users where half of them are chosen from *Active Users* and the other half are chosen from *Inactive Users* to be objective. For NER task in User Interest Model generation, and test data, SCP measurement on Wikipedia Corpus along with length normalisation is used according to the the results of experiment presented in Section 4.1. For threshold values, mean, first quartile, second quartile, and third

quartile values are chosen to be tested. The results of this experiments are given from the aspect of N_F, N_T and T separately to be clear.

4.2.2 Number of Friends N_F and Number of Tweets N_T

In order to decide on the N_F and N_T that will be used in the resulting user model, the approach explained above is applied. The results for each threshold are averaged to see the relationship between N_F and N_T more explicit and given in Table 4.8.

Table 4.8: Accuracy Rate with Changing N_F and N_T Values in terms of Classification as Percentages

		Number of Tweets N_T					
		1	**5**	**10**	**15**	**20**	**Avg.**
Number of Friends N_F	**5**	13.25	15.50	18.00	16.75	16.75	16.05
	10	13.75	16.50	20.50	20.00	19.50	18.05
	15	38.25	41.50	43.00	43.50	44.00	42.05
	20	63.50	68.00	**71.25**	70.50	69.25	**68.50**
	25	62.00	67.50	68.00	65.50	66.00	65.80
	30	56.50	56.50	58.50	57.50	57.00	57.20
	Avg.	41.20	44.25	**46.54**	45.62	45.41	

As it can be observed from the Table 4.8, as number of tweets increase, correct guess percentage first tends to increase, then starts to decrease. This is due to the fact that increasing number of tweets means including more subjects and apparently begins to disrupt subject of interest of the user. On the other hand, as the number of friends increases, correct guess percentage first increases, then at some point starts to decrease likewise. This is because increasing number of friends results in close relative rankings and therefore most interesting friends begin to lose importance. In addition, more friends brings more subjects, and again this fact results in subject of interest disruption. The best result is taken when number of friends is 20, and number of tweets is 10.

Considering Table 4.9, since ranking values are more uniform and it is highly dependent on user preferences where changing a score of a single tweet can change all of the results, there is no sharp result as in the comparison in terms of classification.

Table 4.9: Accuracy Rate with Changing N_F and N_T Values in terms of Ranking Quality as nDCG Values

		Number of Tweets N_T					
		1	**5**	**10**	**15**	**20**	**Avg.**
Number of Friends N_F	**5**	0.700	0.699	0.729	0.668	0.723	0.704
	10	0.694	0.721	0.720	0.669	0.710	0.703
	15	0.687	0.678	0.723	0.701	0.715	0.701
	20	0.689	0.723	**0.735**	**0.735**	0.730	**0.722**
	25	0.702	0.719	0.720	0.710	0.715	0.713
	30	0.665	0.675	0.680	0.689	0.701	0.682
	Avg.	0.690	0.703	**0.718**	0.695	0.716	

However, one can say that with the same reason as in results in terms of classification, low number of friends and high number of friends give relatively worse results. The best results are taken when number of friends is 20. Although, low number of tweets give relatively worse results, apparently as number of tweets increase, the result does not change significantly. The best results are taken when number of friends is 10, and 15. Also considering results in Table 4.8, for generating user model, number of friends parameter N_F is taken 20, and number of tweets parameter N_T is taken 10.

4.2.3 Threshold T

In order to decide the suitable threshold that will be used to classify tweets after their weight relative to the user model is calculated, the approach explained above is applied. The success rate averages are calculated for changing N_F and N_T as percentages separately. The results are given in Table 4.10. and Table 4.11.

As it can be seen in Table 4.10, correct guess percentages according to changing N_T and T values are compared where average of results for different N_F values are averaged for each combination. Apparently the worst result is when Quartile 1 value is chosen as a threshold. Quartile 2 and Mean values gives comparatively better results than Quartile 1 as threshold, but the best result is when Quartile 3 is chosen as threshold on average. In Table 4.11, results are presented from the aspect of changing N_F and T values. This time, results for different N_T values are averaged for every N_F

Table 4.10: Accuracy Rate With Changing N_F and T Values as Percentages

		Thresholds T			
		Mean	**Quartile 1**	**Quartile 2**	**Quartile 3**
Number of Friends N_F	**5**	22.80	8.40	8.20	24.80
	10	26.00	8.40	10.60	27.20
	15	46.40	25.00	28.20	68.60
	20	75.40	48.40	68.00	82.20
	25	74.60	46.20	66.80	75.60
	30	71.20	22.00	63.00	72.60
Average		**52.73**	**26.40**	**40.80**	**58.50**

and T combination. Results show that although results with the Mean value is close, Quartile 3 as threshold value gives better results on average than the other values. Therefore, in the result of this experiment, Quartile 3 value of the data set containing weight of the candidate tweets relative to the user model is chosen for threshold value among other values.

Table 4.11: Accuracy Rate With Changing N_T and T Values as Percentages

		Thresholds T			
		Mean	**Quartile 1**	**Quartile 2**	**Quartile 3**
Number of Tweets N_T	**1**	51.333	18.333	39.166	56.000
	5	51.666	25.619	40.309	57.714
	10	54.333	29.000	41.666	61.166
	15	53.333	29.333	41.333	58.500
	20	53.000	28.500	41.333	58.833
Average		**52.733**	**26.157**	**40.761**	**58.442**

4.3 Experiments on Tweet Recommendation

In this section, recommendation performance of the system is evaluated from the aspect of two approaches: classification, and ranking. The performance of the recommendation task is measured by comparing the results with real user preferences

in terms of both approaches. In addition, a baseline method for recommendation is defined, and the performance of the baseline method is also presented along with the results of our method in following sections.

4.3.1 Approach

To evaluate the system from recommendation point of view, two types of datasets as candidate tweets for recommendation and two types of user groups to recommend tweets are formed. First dataset of candidate tweets, $GNRL$, is a general dataset containing 100 tweets crawled from newspaper's Twitter accounts. Second dataset, $PSNL$ is a personal dataset containing 100 tweets where tweets are crawled from the user's friends' friends. There are 10 users volunteered for this experiment where half of them are active Twitter users, and the other half are inactive Twitter users. *Active Users* are the users that use Twitter frequently, have retweeting and mentioning habits, and update friends list when necessary where *Inactive Users* do not post, retweet, or mention often, and do not update friends list frequently. Volunteered users are categorised by asking them on their Twitter usage habits.

For each user, by crawling their Twitter information, a user model is created as explained in Chapter 3. In Named Entity Recognition tasks of creating the user model and extracting named entities from candidate tweets, SCP measure on Wikipedia Corpus along with length normalisation is used for stickiness function, which gives the best results according to the Experiment 4.1. Also, user interest model is generated using best N_T and N_F values obtained via Experiment 4.2, therefore 20 friends of the user and 10 tweets of each friend are included in the User Interest Model. After named entities are extracted from candidate tweets, candidate tweets are scored by comparing with User Interest Model as explained in Section 3.6 and then ranked. Meanwhile, each user is asked to classify and score tweets in $GNRL$ and $PSNL$ datasets of candidate tweets. Volunteered users made a two-step evaluation on each tweet for each dataset. They are asked to mark the tweet as interesting or uninteresting, and then if the tweet is interesting, they are forced to score the tweet in the range of $[1-3]$ where 1 is less interesting, 3 is more interesting, and 2 is in somewhere in the middle.

Finally, a baseline method is defined in which user interest model does not contain friend rankings, therefore every named entity is equal weight, in order to see if our user interest modelling really makes a difference in terms of recommendation. Recommendation is performed both for the baseline method, and our method. Then, real user preferences are compared with system's results in terms of classification, and ranking. For system to classify the tweets, T value obtained from Experiment 4.2; in other words, Quartile 3 value of the set of candidate tweet scores are used. To evaluate the accuracy of ranking, *Normalized Discounted Cumulative Gain (nDCG)* measure explained in Section 3.6 is used. In this experiment, for ranking accuracy, *DCG* values for the resulting ranking of our system and the ideal ordering are calculated for the last tweet in ranking in order to compare the ranking as a whole.

4.3.2 Overall Results

In this section, results for both final method and the baseline method is given. The results shown in Table 4.12 shows that baseline method is able to decide whether a tweet is interesting for a user or not with the accuracy of 54,10% in average with classification and 0,624 *nDCG* value in average with ranking, which are lower than the results of our system. The performance of the baseline method in some cases decreases down to 36% correct guess at classification, and 0,322 *nDCG* value at ranking quality. One can easily say that our method whose results are given in Table 4.13 outperforms the baseline method at recommending tweets.

On the other hand, the results shown in Table 4.13 shows that our system is able to decide whether a tweet is interesting for a user or not with the accuracy of 71,05% in average with classification and 0,767 *nDCG* value in average with ranking. Given the suitable user habits and relevant datasets, performance of the system increases up to the 88% correct guess at classification, and 0,958 *nDCG* value at ranking quality. The comparison of two tables show that our user interest modelling approach with ranked friends increases the performance.

Results according to our method are examined in more detail with respect to user types, and datasets in following sections.

Table 4.12: Tweet Recommendation Experiment Results with respect to the Baseline Method

		Classification Acc. (%)		**Ranking Acc. (nDCG)**	
		$GNRL$	$PSNL$	$GNRL$	$PSNL$
Inactive Users	$User_1$	47	49	0.520	0.612
	$User_2$	42	39	0.573	0.654
	$User_3$	36	37	0.433	0.478
	$User_4$	43	36	0.322	0.301
	$User_5$	49	47	0.567	0.514
Average (IU)		**43.40**	**41.60**	**0.483**	**0.512**
Active Users	$User_6$	68	64	0.777	0.909
	$User_7$	66	61	0.699	0.768
	$User_8$	62	56	0.760	0.782
	$User_9$	71	72	0.720	0.815
	$User_{10}$	72	65	0.601	0.677
Average (AU)		**67.80**	**63.60**	**0.711**	**0.790**
Average (Overall)		**54.10**		**0.624**	

4.3.3 Results with respect to Candidate Tweet Datasets

As it can be seen from Table 4.14 where results are averaged with respect to the datasets, the system gives better results on $GNRL$ dataset when classification is used for evaluation. It can be seen more clear from Table 4.13 that regardless from the user type, classification on $GNRL$ dataset is more successful, except for one case, $User_3$. As it may be surprising at first, better results on $GNRL$ for classification is what is supposed to be. Because $GNRL$ dataset includes tweets of broad subjects since it is collected from newspaper accounts' tweets. Therefore, distribution of interesting and uninteresting tweets in the set is more uniform. Given a dataset and a threshold calculated according to the dataset values, classification will inevitably eliminate tweets. For $GNRL$ dataset, uninteresting tweets are eliminated more than $PSNL$ dataset because $PSNL$ dataset is created for each user specifically, therefore interesting tweets for users are high in number than uninteresting ones. Therefore, classification on $PSNL$ dataset causes some of the interesting tweets to be classified as uninteresting

Table 4.13: Tweet Recommendation Experiment Results with Respect to the Proposed Method

		Classification Acc. (%)		**Ranking Acc. (nDCG)**	
		$GNRL$	$PSNL$	$GNRL$	$PSNL$
Inactive Users	$User_1$	69	66	0.723	0.773
	$User_2$	62	58	0.684	0.796
	$User_3$	52	55	0.656	0.616
	$User_4$	67	52	0.590	0.623
	$User_5$	72	69	0.734	0.691
Average (IU)		**64.40**	**60.00**	**0.677**	**0.700**
Active Users	$User_6$	88	86	0.809	0.958
	$User_7$	79	74	0.795	0.888
	$User_8$	74	68	0.812	0.826
	$User_9$	88	85	0.815	0.904
	$User_{10}$	80	77	0.773	0.872
Average (AU)		**81.80**	**78**	**0.801**	**0.890**
Average (Overall)		**71.05**		**0.767**	

although it is interesting for the user, explaining the lower classification accuracy rate.

On the other hand, considering ranking accuracy, $PSNL$ dataset, which is generated for each user specifically by crawling friends' friends' posts and sampling randomly, gives better results. Table 4.13 also shows that for the same user, performance on $PSNL$ dataset is better regardless of user type, ignoring one case, $User_3$. $PSNL$ dataset, user preferences are more close to ideal ordering, because it is formed based on following principle of Twitter. This also proves that following the act itself represents the interest. The performance on the $GNRL$ dataset is relatively low because this dataset includes broad set of subjects since it is formed via crawling newspaper accounts, and user preferences are more different than the ideal ordering generated via scores given to tweets based on User Interest Model, causing *nDCG* value to decrease.

Table 4.14: Tweet Recommendation Experiment Results with Respect to Candidate Tweet Datasets

		Classification Acc. (%)	**Ranking Acc. (*nDCG*)**
Datasets	**GNRL**	73.10	0.739
	PSNL	69.00	0.795
Average		**71.05**	**0.767**

4.3.4 Results with respect to User Types

Results are also examined with respect to the user types and Table 4.15 shows averaged results for *Active Users* and *Inactive Users*. The system gives better results for *Active Users* than *Inactive Users* as expected. As Table 4.13 also shows more detail, performance of the system is higher for *Active Users* than *Inactive Users* regardless of data types. This study is based on the assumption that acts of a Twitter user which are posts, retweets, mentions and followed friends, determines the interest of the user. This experiment shows that this really is the fact because for *Active Users* supplying these information more realistic, results are more close to the real preferences of the users.

Table 4.15: Tweet Recommendation Experiment Results with Respect to User Types

		Classification Acc. (%)	**Ranking Acc. (*nDCG*)**
Users	**Inactive Users**	62.20	0.689
	Active Users	79.90	0.845
Average		**71.05**	**0.767**

CHAPTER 5

CONCLUSION AND FUTURE WORK

This thesis proposes a new approach to the tweet recommendation problem by making use of named entities extracted from tweets. The proposed method is capable of deciding on tweets to be recommended according to the user's interest.

A powerful aspect of NER approach adopted in this study, tweet segmentation, is that it does not require an annotated large volume of training data to extract named entities, therefore a huge overload of annotation is avoided. In addition, this approach is not dependent on the morphology of the Turkish language and eliminates the overload of a detailed morphological analysis.

Based on NER via tweet segmentation, we develop a recommendation system for tweets. The motivation behind the development of this system is to discard the information overload on Twitter that the users are exposed, and to make people reach the information of interest with ease, considering the wide use of Twitter for a source of information. A Twitter user's profile is represented via named entities extracted from his and his friends' posts, assuming the act of following, and the content of the tweets posted shows the subjects of interest. We believe that the proposed method is a good basis for tweet recommendation using named entities in Turkish language.

A potential drawback of the system is that the results are heavily dependent on content and size of the external global context which is used to decide on word collocations. For instance, the Twitter based corpus usage gives worse results than the Wikipedia based corpus, since the size of Twitter based corpus is relatively smaller and its content does not include a wide range subjects as Wikipedia based corpus does. These

properties of the used external global context effects the candidate named entity generation which is an essential task. Therefore, the size and the content of the chosen external global context is significant.

While we are content with the results so far, the experiments and analyses show that there is still much work that can be done. The study may be expanded and improved as follows:

- Extracted named entities are not categorised as names, numeric expressions, and temporal expressions. The study may be expanded by categorising named entities as ENAMEX, NUMEX, and TIMEX.
- While validating candidate named entities, lenition at the end of some words when an affix is added is ignored. The study may be expanded to handle this type of agglutination.
- The categories of article titles of Wikipedia in the constructed knowledge base to validate candidate named entities are not taken into consideration. The study may be expanded by considering a categorical subject of an extracted named entity.
- This study is conducted on Turkish tweets, however we believe that with minor modifications, it may work well with other languages. The study may be expanded by experimenting on languages other than Turkish.

In summary, the presented system has shown that relationship with followed friends, and posted tweets are strong indicators of a user's interest, and can be used to recommend tweets. This study suggests a novel tweet recommendation approach and forms a good basis to create a solid tweet recommendation application.

REFERENCES

[1] Enrique Alfonseca and Suresh Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *In: Proceedings of the 1 st International Conference on General WordNet*, 2002.

[2] Renzo Angles and Claudio Gutierrez. Survey of graph database models. *ACM Comput. Surv.*, 40(1):1:1–1:39, February 2008.

[3] Masayuki Asahara and Yuji Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 8–15, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[4] Ozkan Bayraktar and Tugba Taskaya Temizel. Person Name Extraction From Turkish Financial News Text Using Local Grammar Based Approach. In *23rd International Symposium on Computer and Information Sciences (ISCIS'08)*, Istanbul, 2008.

[5] Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: A high-performance learning name-finder. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, ANLC '97, pages 194–201, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.

[6] Andrew Eliot Borthwick. *A Maximum Entropy Approach to Named Entity Recognition*. PhD thesis, New York, NY, USA, 1999. AAI9945252.

[7] Gökhan Çelikkaya, Dilara Torunoğlu, and Gülşen Eryiğit. 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, AICT2013*, Baku, Azarbeijan, October 2013. IEEE.

[8] Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. Collaborative personalized tweet recommendation. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 661–670, New York, NY, USA, 2012. ACM.

[9] Haoi Leong Chieu and Hwee Tou Ng. Named entity recognition: A maximum entropy approach using global information. In *Proceedings COLING 2002*, 2002.

[10] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110, 1999.

[11] Neo4j Community. Graph concepts, 2014. [Online; accessed 14-May-2014].

[12] Silviu Cucerzan and David Yarowsky. Language independent named entity recognition combining morphological and contextual evidence. pages 90–99, 1999.

[13] Joaquim F. da Silva and Gabriel P. Lopes. A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora. In *Sixth Meeting on Mathematics of Language*, 1999.

[14] Diego Marinho de Oliveira, Alberto H. F. Laender, Adriano Veloso, and Altigran Soares da Silva. Fs-ner: a lightweight filter-stream approach to named entity recognition on twitter data. In Leslie Carr, Alberto H. F. Laender, Bernadette Farias Loscio, Irwin King, Marcus Fontoura, Denny Vrandecic, Lora Aroyo, Jose Palazzo M. de Oliveira, Fernanda Lima, and Erik Wilde, editors, *WWW (Companion Volume)*, pages 597–604. International World Wide Web Conferences Steering Committee / ACM, 2013.

[15] Zemberek Developers. Zemberek project, 2014. [Online; accessed 14-May-2014].

[16] Doug Downey, Matthew Broadhead, and Oren Etzioni. Locating complex named entities in web text. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI'07, pages 2733–2739, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[17] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165(1):91–134, June 2005.

[18] Richard Evans. A framework for named entity recognition in the open domain. In *In Proceedings of the Recent Advances in Natural Language Processing (RANLP)*, pages 137–144, 2003.

[19] Wei Feng and Jianyong Wang. Retweet or not?: Personalized tweet re-ranking. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 577–586, New York, NY, USA, 2013. ACM.

[20] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, 2000. AAI9980887.

[21] A. Flew. *A Dictionary of Philosophy: Revised Second Edition*. St. Martin's Press, 1984.

[22] Paul A. Foerster. *Algebra and Trigonometry: Functions and Applications, Teacher's Edition*. ISBN 0-13-165711-9. Prentice Hall, Upper Saddle River, NJ, 2006.

[23] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[24] Hongzhao Huang, Arkaitz Zubiaga, Heng Ji, Hongbo Deng, Dong Wang, Hieu Khac Le, Tarek F. Abdelzaher, Jiawei Han, Alice Leung, John P. Hancock, and Clare R. Voss. Tweet ranking based on heterogeneous networks. In *COLING*, pages 1239–1256, 2012.

[25] Rob J. Hyndman and Yanan Fan. Sample quantiles in statistical packages. *The American Statistician*, 50:361–365, 1996.

[26] Hideki Isozaki and Hideto Kazawa. Efficient support vector classifiers for named entity recognition. In *In Proceedings of the 19th International Conference on Computational Linguistics (COLING?02*, pages 390–396, 2002.

[27] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[28] Heng Ji and Ralph Grishman. Data Selection in Semi-supervised Learning for Name Tagging. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, pages 48–55, Sydney, Australia, July 2006. Association for Computational Linguistics.

[29] Saul Kripke. *Naming and Necessity*. Harvard University Press, 1980.

[30] Dilek Küçük and Adnan Yazici. Named entity recognition experiments on Turkish texts. In *FQAS*, pages 524–535, 2009.

[31] Dilek Küçük and Adnan Yazici. A hybrid named entity recognizer for Turkish with applications to different text genres. In *ISCIS*, pages 113–116, 2010.

[32] VI Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, 1966.

[33] Chenliang Li, Aixin Sun, Jianshu Weng, and Qi He. Exploiting hybrid contexts for tweet segmentation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 523–532, New York, NY, USA, 2013. ACM.

[34] Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. Twiner: named entity recognition in targeted twitter stream. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 721–730, New York, NY, USA, 2012. ACM.

[35] Wenhui Liao and Sriharsha Veeramachaneni. A simple semi-supervised algorithm for named entity recognition. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, SemiSupLearn '09, pages 58–65, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[36] Feifan Liu, Jun Zhao, Bibo Lv, Bo Xu, and Hau Yu. Product Named Entity Recognition Based on a Hierarchical Hidden Markov Model. In *Proceedings of SIGHAN*, 2005.

[37] Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 359–367, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[38] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[39] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.

[40] Diana Maynard, Valentin Tablan, Cristian Ursu, Hamish Cunningham, and Yorick Wilks. Named entity recognition from diverse text types. In *In Recent Advances in Natural Language Processing 2001 Conference, Tzigov Chark*, 2001.

[41] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[42] Einat Minkov, Richard C. Wang, and William W. Cohen. Extracting personal names from email: Applying named entity recognition to informal text. In *HLT/EMNLP*. The Association for Computational Linguistics, 2005.

[43] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company.

[44] Serap Ozkaya and Banu Diri. Named entity recognition by conditional random fields from turkish informal texts. *IEEE 19th Signal Processing and Communications Applications Conference*, 2011.

[45] Pavel Pecina and Pavel Schlesinger. Combining association measures for collocation extraction. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, pages 651–658, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[46] L. F. Rau. Extracting company names from text. In *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, volume i, pages 29–32. IEEE, February 1991.

[47] Eric Sven Ristad, Peter N. Yianilos, and Senior Member. Learning string edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:522–532, 1998.

[48] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[49] Gökhan Akin Seker and Gülsen Eryigit. Initial explorations on using crfs for turkish named entity recognition. In *COLING*, pages 2459–2474, 2012.

[50] Satoshi Sekine. Nyu: Description of the japanese ne system used for met-2. In *Proc. of the Seventh Message Understanding Conference (MUC-7*, 1998.

[51] Taner Sezer. Ts corpus, the turkish corpus, 2014. [Online; accessed 14-May-2014].

[52] Yusuke Shinyama and Satoshi Sekine. Named entity discovery using comparable news articles. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[53] Serhan Tatar and Ilyas Cicekli. Automatic rule learning exploiting morphological features for named entity recognition in turkish. *J. Inf. Sci.*, 37(2):137–151, April 2011.

[54] David M. J. Tax and Robert P. W. Duin. Support vector data description. *Mach. Learn.*, 54(1):45–66, January 2004.

[55] Gökhan Tür, Dilek Hakkani-Tür, and Kemal Oflazer. A statistical information extraction system for turkish. *Natural Language Engineering*, 9(2):181–210, 2003.

[56] Peter D. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, pages 491–502, London, UK, UK, 2001. Springer-Verlag.

[57] Twitter. About twitter, inc., 2014. [Online; accessed 14-May-2014].

[58] Twitter. Documentation, twitter developers, 2014. [Online; accessed 14-May-2014].

[59] Twitter4j. Twitter4j , a java library for the twitter api, 2014. [Online; accessed 14-May-2014].

[60] Ibrahim Uysal and W. Bruce Croft. User oriented tweet ranking: a filtering approach to microblogs. In Craig Macdonald, Iadh Ounis, and Ian Ruthven, editors, *CIKM*, pages 2261–2264. ACM, 2011.

[61] Johanna Völker. Towards large-scale, open-domain and ontology-based named entity classification. In *In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'05*, pages 166–172. INCOMA Ltd, 2005.

[62] Webknox.com. Named entity definition., 2014. [Online; accessed 14-May-2014].