NAMED ENTITY RECOGNITION IN TURKISH WITH BAYESIAN LEARNING AND HYBRID APPROACHES

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

BY

SERMET REHA YAVUZ

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

DECEMBER 2011

Approval of the thesis:

NAMED ENTITY RECOGNITION IN TURKISH WITH BAYESIAN LEARNING AND HYBRID APPROACHES

Submitted by **SERMET REHA YAVUZ** 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**

Prof. Dr. Adnan Yazıcı ________________
Supervisor, **Computer Engineering Dept. METU**

Dr. Dilek Küçük ________________
Co-Supervisor, **TÜBİTAK UZAY**

Examining Committee Members:

Assoc. Prof. Dr. Pınar Şenkul
Computer Engineering Dept., METU ________________

Prof. Dr. Adnan Yazıcı
Computer Engineering Dept., METU ________________

Dr. Ruken Çakıcı
Computer Engineering Dept., METU ________________

Assist. Prof. Dr. Murat Koyuncu
Information Systems Eng. Dept., Atılım University ________________

Dr. Dilek Küçük
TÜBİTAK UZAY ________________

Date: 15.12.2011

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: SERMET REHA YAVUZ

Signature :

ABSTRACT

NAMED ENTITY RECOGNITION IN TURKISH WITH BAYESIAN LEARNING AND HYBRID APPROACHES

Yavuz, Sermet Reha
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Adnan Yazıcı
Co-Supervisor: Dr. Dilek Küçük

December 2011, 47 pages

Information Extraction (IE) is the process of extracting structured and important pieces of information from a set of unstructured text documents in natural language. The final goal of structured information extraction is to populate a database and reach data effectively. Our study focuses on named entity recognition (NER) which is an important subtask of IE. NER is the task that deals with extraction of named entities like person, location, organization names, temporal expressions (date and time) and numerical expressions (money and percent). NER research on Turkish is known to be rare. There are rule-based, learning based and hybrid systems for NER on Turkish texts. Some of the learning approaches used for NER in Turkish are conditional random fields (CRF), rote learning, rule extraction and generalization.

In this thesis, we propose a learning based named entity recognizer for Turkish texts which employs a modified version of Bayesian learning as the learning scheme. To the best of our knowledge, this is the first learning based system that uses Bayesian approach for NER in Turkish. Several features (like token length, capitalization, lexical meaning, etc.) are used in the system to see the effects of different features on NER process. We also propose hybrid system where the Bayesian learning-based system is utilized along with a rule-based recognition system. There are two different versions of the hybrid system. Output of rule-based recognizer is utilized in different phases in these versions. We observed increase in F-Measure values for both hybrid versions. When partial scoring is active, hybrid system reached 91.44% F-Measure value; where rule-based system result is 87.43% and learning-based system result is 88.41%. The hybrid system can be improved by utilizing rule-based and learning-based components differently in the future. Hybrid system can also be improved by using different learning approaches and combining them with existing hybrid system or forming the hybrid system with a completely new approach.

Keywords: Named Entity Recognition, Machine Learning, Bayesian Learning, Turkish, Information Extraction

ÖZ

BAYES ÖĞRENME VE HİBRİT YAKLAŞIMLAR İLE TÜRKÇE'DE VARLIK İSMİ TANIMA

Yavuz, Sermet Reha
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Adnan Yazıcı
Ortak Tez Yöneticisi : Dr. Dilek Küçük

Aralık 2011, 47 sayfa

Bilgi Çıkarımı (BÇ), doğal dildeki yapısal olmayan metin belgele kümelerinden; yapısal önemli bilgi parçalarını çıkarma işlemidir. Yapısal bilgi çıkarımının nihai amacı bir veritabanını doldurmak ve veriye etkili bir şekilde erişebilmektir. Bizim araştırmamız, BÇ'nin önemli bir alt görevi olan Varlık İsmi Tanıma (VİT) üzerine odaklanmaktadır. VİT görevi; kişi adları, yer adları, organizasyonlar, zamansal ifadeler (tarih ve saat), sayısal ifadeler (para ve yüzde) gibi varlık isimlerininin tanıması ile ilgilenir. Türkçe için VİT araştırmalarının nadir olduğu bilinmektedir. Türkçe için elle oluşturulmuş kural tabanlı, öğrenme tabanlı ve melez VİT çalışmaları bulunmaktadır. Türkçe VİT için kullanılan bazı öğrenme yaklaşımları; şartlı rastgele alanlar (CRF), ezber öğrenme, kural çıkarım ve genellemesi olarak örneklenebilir.

Biz bu tezde, öğrenme yaklaşımı olarak Bayes yaklaşımının değiştirilmiş bir versiyonunu kullanan öğrenme tabanlı bir Türkçe varlık ismi tanıma sistemi öne sürmekteyiz. Bildiğimiz kadarıyla, bu sistem Bayes yöntemini Türkçe varlık ismi tanıma için kullanan ilk sistemdir. Farklı özelliklerin kullanımının VİT işlemine etkisini görmek için sistemde birkaç farklı özellik türü (sözcük uzunluğu, büyük-küçük harf kullanımı, sözlük anlamı gibi) kullanılmıştır. Ayrıca öğrenme tabanlı sistemin, kural tabanlı bir sistemle birlikte kullanımından oluşan hibrit bir sistem de öne sürmekteyiz. Hibrit sistemin iki farklı versiyonu bulunmaktadır. Bu versiyonlarda, kural tabanlı sistemin çıktıları farklı aşamalarda kullanılmıştır. Her iki hibrit sistemin de sonuç performansını artırdığını gözlemledik. Kısmi puanlandırma aktif iken; kural tabanlı sistem %87.43, öğrenme tabanlı sistem de %88.41'lik performans gösterirken hibrit sistem %91.44'lük performansa ulaşmıştır. İleride, kural tabanlı ve öğrenme tabanlı parçalar daha farklı kullanılarak hibrit sistem dah da geliştirilebilir. Ayrıca hibrit sistemi geliştirmek için; farklı öğrenme yöntemleri varolan hibrit sistem ile birleştirilebilir ya da tamamen yeni bir yaklaşımla hibrit sistem oluşturulabilir.

Anahtar Kelimeler: Varlık İsmi Tanıma, Otomatik Öğrenme, Bayes Tabanlı Öğrenme, Türkçe, Bilgi Çıkarımı

To my family

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to my supervisor Prof. Dr. Adnan Yazıcı and my co-supervisor Dr. Dilek Küçük for their encouragement and support throughout this study.

I am deeply grateful to my family for their love and support. Without them, this work could never have been completed.

I would thank the Scientific and Technological Research Council of Turkey (TÜBİTAK) for providing the financial means throughout this study.

Finally, my special thanks go to my friends, Mustafa Aykut Ural, Hasan Gökçınar, Emrah Mert, Halil Uysal, Zafer Gençkaya, Derya Susman, and Osman Nuri Osmanlı for their help, support and cheerful presence through not only the course of this study but also any course of my life. Thanks for giving me a shoulder to lean on, whenever I need.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF ABBREVIATIONS

ACE	Automatic Content Extraction
ASD	After Surrounding Distance
BSD	Before Surrounding Distance
BWI	Boosted Wrapper Induction
CRF	Conditional Random Fields
HMM	Hidden Markov Model
IE	Information Extraction
IR	Information Retrieval
MTC	Metu Turkish Corpus
MUC	Message Understanding Conference
NAF	Nymble Alike Features
NE	Named Entity
NER	Named Entity Recognition
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

1.1 Information Extraction

Need of information has always been a problem throughout the history. Reachable information is growing exponentially day by day with the utilization and popularity of the Internet. Within this mess of data and information, it is getting harder to find valuable or wanted information. Information Extraction (IE) aims to ease accessing to wanted piece of information or data inside of this vast data ocean. IE is the name given to any process or task which selectively combines and structures data which is implied, found or explicitly stated in one or more unstructured texts [27]. The final output of the extraction process varies; in every case, however, it can be transformed so as to populate some type of database and reach the extracted data effectively [27]. Information analysts working long term on specific tasks already carry out information extraction manually with the express goal of database creation as stated in [27].

IE is the area of extracting usable data from unformatted documents or texts. This extraction process usually involves into language processing or linguistic decomposition. Other alternative for IE is the systems that use previous examples of extracted information. That kind of systems requires annotated data from the domain that IE process will be done.

Information extraction systems can be used for extracting valuable information from

- Unformatted texts,
- Web pages,

- Personal documents,
- Processed multimedia.

Main IE tasks are given below [2]:

- Named entity extraction
    - Named Entity Recognition (NER): Recognition of known entity names (for people and organizations), place names, temporal expressions, and certain types of numerical expressions. NER is the main focus of this thesis.
    - Coreference Resolution: Detection of coreference and anaphoric links between text entities. In IE tasks, this is typically restricted in finding links between previously-extracted named entities.
    - Relationship Extraction: Identification of relations between entities, such as: PERSON works for ORGANIZATION (extracted from the sentence "Bill works for IBM.").
- Semi-structured IE which is any IE that tries to restore the information which is lost:
    - Table extraction: Finding and extracting tables from documents.
    - Comments extraction : Extracting comments from actual content of article in order to restore the link between author of each sentence.
- Language and vocabulary analysis
    - Terminology extraction: Finding the relevant terms for a given corpus.

1.2 Contributions and Motivation

Our study is centered on NER task of IE. In this thesis, we propose a learning based NER system for Turkish texts which employs a modified version of Bayesian learning as the learning scheme. Bayesian learning algorithm uses statistical frequency of tokens. In Bayesian algorithm, frequencies of tokens are kept in training phase and this frequency distribution is used on estimation phase to determine Named Entities (NE). The alternative Bayesian approach that we use also exploits statistical frequency of tokens, uses several different features than Bayesian method proposed in [9] and uses an alternative way for combining probabilities created by each feature and token frequencies. To the best of our knowledge, this is the first learning based system that uses Bayesian approach for NER in Turkish.

One of the main advantages of a learning based recognition system is its domain independence. These systems can be used in various domains through training with domain specific examples and annotated data. Changes required for domain adaptability is minimized for a learning based recognition system where there is enough amount of annotated training data for the target domain.

The learning based recognition system will make use of many features like frequency statistics, typography, orthography and formatting as stated in [9]. In addition to the features used in [9], some other features (like lexical resources, case sensitivity etc.) are also used for probability calculations. We want to observe how the performance of whole system changes with combinations of multiple features. Effects of these features to the performance of the system will be reported throughout the thesis and best features or properties for NER in Turkish using Bayesian algorithm will be discussed. Nevertheless, best feature combination may differ for different entity types or according to training or testing data used.

Our secondary target is using this learning based system with previously designed rule-based NER system in [12, 13]. Joint utilization of these systems would create a hybrid system which is expected to work better than both of rule-based system and learning based system. Küçük and Yazıcı’s study presented in [14] is a good example for a hybrid NER system for Turkish. We will design and develop a system which will use the same rule-based component in Küçük and Yazıcı’s study but learning-based component will use a different learning approach and more features. Learning based system in [14] uses “rote learning” as learning algorithm. Our hybrid system will use “Bayesian” algorithm as the learning algorithm.

1.3 Organization of the Thesis

The rest of the thesis is organized as follows: Chapter 2 reviews relevant information about information extraction and named entity recognition. Chapter 3 describes the developed learning based NER system. In Chapter 4, evaluation results of the system are presented. Finally, Chapter 5 includes a conclusion and discussion for future work.

CHAPTER 2

RELATED WORK

In the area of information extraction (IE) there have been considerable researches for many languages. We will only discuss the works on named entity recognition area since focus of this thesis is named entity recognition.

In chapter 2.1 named entity concept and named entity recognition in general will be discussed. In Chapter 2.2, previous works in learning based NER systems will be described. In Chapter 2.3, previous works in Turkish NER will be reported.

2.1 Named Entity Recognition

Named entity recognition (NER), also known as “named entity extraction”, is an important sub-field of IE. Named entities can be used for summarizing or classifying text documents. NER is the task of locating and classifying named entities inside a document. NER is considered as the entrance point for any IE task and prerequisite to other IE tasks like “Relation Extraction” and “Event Extraction” which use named entities.

Named entity can be anything that is valuable for searching, extracting or classifying. But there is a named entity set defined by Message Understanding Conference (MUC) which is used commonly. According to MUC [6, 18], named entities consist of three sub-categories:

- Entity Name Expressions
- Temporal Expressions
- Numerical Expressions

Entity Names are tagged as “ENAMEX” and consist of three types:

- Person
- Location
- Organization

Temporal Expressions are tagged as “TIMEX” and consist of two types:

- Date
- Time

Number Expressions are tagged as “NUMEX” and consist of two types:

- Money
- Percent

Many programs exist for encouraging and supporting IE. MUC is one of the important programs of that kind. As stated in [6], MUC evaluations had been funding the development of metrics and statistical algorithms to support emerging IE technologies. In the mid-nineties MUC evaluations began to provide prepared data and task definitions in addition to providing fully automated scoring software to measure machine and human performance. The tasks grew from just production of a database of events found in newswire articles from one source to the production of multiple databases of increasingly complex information extracted from multiple sources of news in multiple languages. The databases now include named entities, multilingual named entities, attributes of those entities, facts about relationships between entities, and events in which the entities participated [6].

NER is a widely researched area. According to Wikipedia [7], state-of-the-art NER systems for English produce near-human performance. For example, the best system entering MUC-7 scored 93.39% of F-Measure while human annotators scored 97.60% and 96.95% [22, 23]. These algorithms had roughly twice the error rate (6.61%) of human annotators (2.40% and 3.05%).

2.2 Learning Based Named Entity Recognition

Learning based NER systems are studied considerably within scope of IE. Since domains in real world are very interchangeable, learning based systems, which are flexible, are getting more attractive.

Nymble [16] can be considered as one of the most successful studies as an example of learning based NER. In Nymble, an Hidden Markov Model (HMM) is used for NER. Beginning of a sentence is the start point and end of a sentence is the end point of the conceptual model. HMM is used to distinguish named entities and other tokens between start and end points. F-Measure of Nymble is max 93% for English and 90% for Spanish. For English, a training dataset containing 450.000 words is used in this study. For Spanish, the training dataset includes 223.000 words [16].

In [9], many learning based approaches for NER area are introduced. In this thesis; “Rote Learning”, “Bayes”, “Bayes IDF”, “Grammatical Inference”, “SRV” methods or combinations of these methods are used and compared with each other. Bayes IDF method is a very important inspiration point for the current thesis. “Naïve Bayes” and “Bayes IDF” techniques are developed for recognition of Turkish named entities. After some modifications, the resulting “Alternative Bayes IDF” method is used for learning based recognition system.

SRV is a relational learning approach for NER introduced in [9]. In SRV, the learner produces a set of logical rules (or their functional equivalent) on training phase to find and classify named entities. When creating a rule, system starts with an empty logical rule which would find all positive and negative token groups. Step by step most useful logical statement is added to the rule until no negative output is created by learner or no improvement can be done. Resulting logical statement is extracted as one rule of the system. Positive examples that can be found by this rule are removed from training data and extraction of another rule starts. New rule extraction continues until all positive examples in the training data can be extracted by created rules; or the number of rules reaches to a predefined threshold value.

Boosted Wrapper Induction (BWI) is introduced by Kushmerick and Freitag as a supervised learning based NER approach [29]. In BWI, learning a wrapper involves determining the fore (F) and aft (A) detectors and a function H. Fore detectors detects beginning of a named entity, aft detectors detects ending of a named entity and function H gives a likelihood for the length of a named entity. Multiple wrappers are learned in the training phase to determine the named entities with different characteristics. Fore and aft detectors have a parameter called "look-ahead", which determines the number of tokens to be checked when determining the beginning or end of a named entity. BWI also uses wildcards to improve performance of the system.

Support Vector Machine (SVM) is a statistical machine learning approach used for NER in [30]. One of the most successful machine learning methods for IE is SVM. It has achieved state-of-the-art performance on many classification tasks, including named entity recognition (see e.g. [32], [33]). The system produced in [30] uses a variant of the SVM, the SVM with uneven margins [31], which has a better generalization performance than the original SVM on an imbalanced dataset where the positive examples are much less than the negative ones. The original SVM treats positive and negative examples equally such that the margin of the SVM hyper plane to negative training examples is equal to the margin to positive training examples. However, for imbalanced training data where the positive examples are so rare that they are not representative of the genuine distribution of positive examples, a larger positive margin than the negative one would be beneficial for the generalization of the SVM classifier [30].

2.3 Named Entity Recognition in Turkish

NER research in Turkish is known to be uncommon. Recently this area has started to attract interest and research on NER in Turkish has gained acceleration.

NER in Turkish is a little bit different than English and other European-based languages. We can list primary specialties of Turkish for NER as:

- Turkish is an agglutinative language. This feature increases different tokens created using suffixes and decreases the frequency of the tokens
- Suffixes after named entities help recognition since they are considered as separate tokens.

Some studies have been done which use rule-based and learning based approaches for Turkish.

In rule-based recognition, one of the most widely known NER systems for Turkish is the study of Küçük and Yazıcı presented in [11, 12, 13]. They used lexical resources and patterns for extracting named entities. They have also implemented a morphological analyzer for Turkish considering only the noun inflections, so that only those items which both exist in the lexical resources (or conform to the patterns in the pattern bases) and take the appropriate suffixes are extracted from the texts. This system reaches an F-Measure of 78.28% without partial scoring. F-Measure of the system reaches near 88% with partial scoring.

Küçük and Yazıcı also proposed a hybrid NER system in their study presented in [14]. This hybrid system is based on joint utilization of rule-based system and a learning based component which exploits rote learning approach. When capitalization feature is active, this hybrid system reaches an F-Measure of 90.13% on news dataset (a tagged portion of [19]), 92.47% on child stories dataset [20, 21]. These datasets are also used for evaluation of the system proposed in this thesis.

For learning based recognition systems in Turkish, best example is recently published study of Tatar [17]. This study suggests a rule learning system. Each rule consists of three parts in this system;

- PRE-FILLER, which tries to match precedings of Named Entity (NE)
- FILLER, which tries to match target NE
- POST-FILLER, which tries to match the followings of NE

Positive rules and negative rules are created for NER in the study [17]. Results of this system can be considered to be comparable to the studies world-wide. Overall F-Measure of the system is 91.08%.

Another good NER system is the study of Tür et al. presented in [24]. They used n-gram language models embedded in Hidden Markov Models (HMM) as the approach. They used four models in the name tagging task as following:

- Lexical model, which captures the lexical information using only word tokens.
- Contextual model, which captures the contextual information using the surrounding context of the word tokens. For tagging unknown words, this model is claimed to be very helpful.
- Morphological model, which captures the morphological information with respect to the corresponding case and name tag information. Morphological parsing of the words is used in order to build this model.
- Name tag model, which captures the name tag information (person, location, organization, and else) of the word tokens.

This system reaches an F-Measure of 91.56% [24], which is considerably high compared to other studies.

Yeniterzi used a system which is based on CRF (Conditional Random Fields) and exploits morphological features [25]. They state that CRF provides advantages over HMMs and enables the usage of any number of features. In their work, they use two tokenization methods. Initially they start with the sequence of words representation which will be referred as word-level model. They also introduced morpheme-level model in which morphological features are represented as states. Several features are used which were created from deep and shallow analysis of the words. Study of Yeniterzi uses the same dataset with Tür et al. [24]. This system reaches an F-Measure of 88.94% overall.

CRF is also used by Özkaya and Diri for Turkish NER [26]. They exploited CRF for recognition of persons, locations and organizations form informal e-mail documents. Evaluation of this system is based on a relatively small dataset. F-Measure of this system is 91.17% overall for a test set of 637 named entities.

CHAPTER 3

A BAYESIAN LEARNING SYSTEM FOR NAMED ENTITY RECOGNITION IN TURKISH

3.1 Learning Based Recognizer

The learning based recognizer that we propose is a NER system which uses supervised learning techniques. A statistical learning based recognizer generally has three steps for recognition. These steps are:

- Training phase
- Estimation phase
- Output phase

3.1.1 Training Phase:

In this phase, the system is fed with positive and/or negative examples. For example if we want to create a system which finds person names in a document, we should train the system with positive person name examples. The system also needs negative examples to understand the difference of person names from rest of the document. In a document, negative example can be any token which is not a related to a named entity. Tokens of a named entity cannot be negative examples. Also tokens before and after a named entity can be considered as “related to” the named entity.

3.1.2 Estimation Phase:

Estimation phase is where the estimations are performed. The document, in which we are trying to find named entities, is given to the system as input. System creates estimation results or probabilities for some tokens or groups of tokens. These probabilities are calculated using training data and input data together. Calculation technique varies according to the estimation methodology to be used. Output of this phase is probability or scores for groups of tokens to be a named entity.

3.1.3 Output Phase:

Output phase is where results of the system are generated. Output phase takes estimations and their probabilities or scores as input. Applies a threshold for these probabilities or scores, merges or splits some of these groups, handles overlapping groups and generates results. Those results are the named entities found by the system.

Our learning system is based on Bayesian learning techniques. There are advantages and disadvantages of these techniques. Advantages of Bayesian based techniques are:

- Less domain dependence [9]
- Quick (less time consumption) [9]
- Good recognition performance compared to more sophisticated learners [35, 36]
- Provides useful information as a component (usable with other approaches)
- Different features can be used with minimal modification

The disadvantage of Bayesian based technique is that it does not make use of relations between tokens.

Current learning system can be used with Naïve Bayes, Bayes IDF and Alternative Bayes IDF approaches.

In Bayesian learning techniques, probability of a group of tokens to be named entity is derived from the group itself, tokens before the group and tokens after the group. These three parts used for calculation of the probability of the estimation.

3.1.4 Naïve Bayes

Naïve Bayes is the least complex version of Bayes based learning techniques. As stated in the thesis of Freitag [9]:

Bayes' Rule shows us a way to calculate probability of a hypothesis H in response to the evidence contained in some empirically obtained data D:

$$\Pr(H_j D) = \Pr(D_j H) \Pr(H) / \Pr(D) \tag{3.1}$$

In other words, the posterior probability for H being correct is proportional to the product of the prior probability $\Pr(H)$ and the probability of observing the data D, conditioned on H, $\Pr(D_j H)$. In classification, the objective is to choose one of several hypotheses H_i; the data and the denominator $\Pr(D)$ is the same for all H_i and it is disregarded. Bayes' Rule suggests that; the hypothesis that maximizes the product $\Pr(D_j H_i)$ $\Pr(H_i)$ is the best classification to be chosen. For applying Bayes' Rule to estimation problem, two estimates are needed: $\Pr(D_j H_i)$ and $\Pr(H_i)$, which are conditional data likelihood and the prior. In identifying the name of the speaker in a seminar announcement problem, the problem can be modeled as a collection of competing hypotheses, where each hypothesis represents for a group of tokens to be a speaker's name [9].

For a token group, "The token group which starts at token position p and consists k tokens is a person" can be a hypothesis (the hypothesis $H_{p,k}$). For example, $H_{105,4}$ represents a hypothesis that, the token group starting with 105th token and consists of 4 tokens is a person. [9].

In Naïve Bayes, frequency of a token is calculated for $Pr_{before}()$ and $Pr_{after}()$ is calculated as [9]:

$$\frac{Appearance\ inside/before/after\ NE}{Count\ of\ NEs} \tag{3.2}$$

Frequency of a token for $Pr_{in}()$ is calculated as [9]:

$$\frac{Appearance\ inside/before/after\ NE}{Count\ of\ tokens\ inside\ NEs} \tag{3.3}$$

Probability is calculated with multiplication of frequency of each token for before, inside and after named entity [9].

$$Pr() = \prod_{j=-BSD}^{k+ASD} \Pr\left(t_{p+j-1}\right) \tag{3.4}$$

In previous formula, k is the count of tokens inside named entity. “BSD” is the parameter which is count of tokens to be checked before NE. “ASD” is the parameter which is count of tokens to be checked after NE. Resulting probability is calculated by multiplying probability of each token.

3.1.5 Bayes IDF

There are two difference of Bayes IDF from the Naïve Bayes approach. First one is frequency of a token is calculated as [9]:

$$\frac{Appearance\ inside/before/after\ NE}{Appearence\ inside\ all\ document} \tag{3.5}$$

In Bayes IDF, denominator of frequency is token appearance count inside the whole training document.

The second difference is calculation of accumulative probability of each token inside a NE; for each token inside estimated NE, probability is calculated and accumulated. Then this accumulation is divided by average token count of the named entities in the training data [9].

$$Pr_{in}() = \sqrt[k_{avg}]{\prod_{j=1}^{k} \Pr\left(t_{p+j-1}\right)} \tag{3.6}$$

Probability calculation formula of before NE and after NE is the same as Naïve Bayes. Entire probability is calculated as multiplication of $Pr_{before}()$, $Pr_{in}()$ and $Pr_{after}()$ as stated in following formula [9]:

$$Pr() = \prod_{j=-BSD}^{0} \Pr\left(t_{p+j-1}\right) . \sqrt[k_{avg}]{\prod_{j=1}^{k} \Pr\left(t_{p+j-1}\right)} . \prod_{j=k+1}^{k+ASD} \Pr\left(t_{p+j-1}\right) \quad (3.7)$$

3.1.6 Alternative Bayes IDF

We also used a modified version of Bayes technique which gives better results than Naïve Bayes and Bayes IDF. Alternative Bayes IDF uses the same formula with Bayes IDF for calculation of frequency for a token as following:

$$\frac{Appearance\ inside/before/after\ NE}{Appearence\ inside\ all\ document} \quad (3.8)$$

Calculation of probability for a named entity is different than both of Naïve Bayes and Bayes IDF. For calculating probability for a named entity, arithmetical mean of all probabilities is used instead of multiplication of all probabilities. Probabilities created by usage of different features are included when calculating the average probability. Formula for calculating entire probability:

$$Pr() = \frac{\sum_{j=-BSD}^{0} \Pr\left(t_{p+j-1}\right) + \frac{\sum_{j=1}^{k} \Pr\left(t_{p+j-1}\right)}{k_{avg}} + \sum_{j=k+1}^{k+ASD} \Pr\left(t_{p+j-1}\right) + \sum_{i=0}^{FC} \Pr_{\mathrm{FC}}()}{FC+BSD+ASD+1} \quad (3.9)$$

In previous formula FC is the count of used features. Denominator of arithmetic mean is “FC+BSD+ASD+1” statement because arithmetic mean inside NE is already calculated in accordance with Bayes IDF. Count of probabilities come from other statements is equal to sum of count of tokens to be checked before and after NE respectively, count of probabilities comes from features is FC.

Alternative Bayes IDF method has the following benefits besides performance:

- Ease of adding/removing new features

- Coefficient specification possibility for features
- More realistic/understandable probability (not 2^{-35} etc)
- Applicability of identity element

3.1.7 Features:

We used several different features for the system. We intended to observe the effects of different features. These effects provide information for general recognition system and some of those features can be used with other learning approaches. Bayesian approach gives us flexibility to add/remove features from our feature set without too much effort. Also after adding a feature, it can be activated/deactivated easily in the Bayesian approach. We made evaluations with so many different feature sets or coefficients.

Used features can be listed as:

- Before surroundings distance
- After surroundings distance
- Case sensitivity
- Case usage
- Length usage
- Alphanumeric feature usage
- NAF (Nymble alike features)
- Lexical resource usage
- Coefficients

3.1.7.1 Before Surroundings Distance (BSD):

In any Bayesian technique, a number of tokens before and after the named entity are used for training and estimation. We call these tokens “Surroundings”. “Before

Surroundings Distance (BSD)” is the number of tokens to be checked before named entity, to calculate a probability for being a named entity.

When BSD is 3; that means 3 tokens before the named entity will be used for training and 3 tokens before the estimated named entity will be used for calculations of probability, as exemplified below:

Genelkurmay Başkanı Orgeneral **İlker Başbuğ**’un mesajının okunmasından sonra öğrenciler tarafından şiirler okundu.

Consider we are calculating the probability of “İlker Başbuğ” to be named entity. If BSD is 3;

- “Genelkurmay” token will be searched for being the third token before the named entity,
- “Başkanı” token will be searched for being the second token before the named entity,
- “Orgeneral” token will be searched for being the first token before the named entity.

And those probabilities will all affect the probability of “İlker Başbuğ” for being a named entity.

But if BSD is 1, only “Orgeneral” token will be searched for being the first token before the named entity. “Genelkurmay” and “Başkanı” tokens will not be taken into account.

3.1.7.2 After Surroundings Distance (ASD):

After Surroundings Distance (ASD) is very similar to BSD. Only difference is; ASD is the number of tokens to be checked after the named entity, to calculate a probability for being a named entity.

When ASD is 3; that means 3 tokens after the named entity will be used for training and 3 tokens after the estimated named entity will be used for calculations of probability, as illustrated below:

Genelkurmay Başkanı Orgeneral **İlker Başbuğ**'un mesajının okunmasından sonra öğrenciler tarafından şiirler okundu.

Again consider we are calculating the probability of "İlker Başbuğ" to be named entity. If ASD is 3;

- ' (apostrophe) token will be searched for being the first token after the named entity.
- "un" token will be searched for being the second token after the named entity,
- "mesajının" token will be searched for being the third token after the named entity,

And again, all those probabilities will affect the probability of "İlker Başbuğ" for being a named entity.

But if ASD is 1, only ' (apostrophe) token will be searched for being the first token after the named entity. "un" and "mesajının" tokens will not be taken into account.

3.1.7.3 Case Sensitivity:

In any statistical learning approach, the frequency of tokens (for being inside of NE, after NE etc) is kept in training phase. After the training phase, calculations are performed using the frequency of a token that we will use for estimation.

Case sensitivity determines whether the tokens are kept in a case sensitive way or insensitive way. Case sensitivity can affect the results considerably. For example the token "bugün" (means "today") is very useful for finding time expressions or itself can be a time expression. If this token is at the beginning of a sentence its first letter becomes capital and the token becomes "Bugün". It is better to keep the frequencies of that token without case sensitivity since that provides us to give same probability for “bugün” and “Bugün” tokens.

On the other hand; some of the tokens need to be kept case sensitive. For example the token “gül” (means “rose”, “smile” or “laugh”) is not a named entity itself. But current president of Turkey is Abdullah Gül. In news domain, “Gül” is commonly used for referring to the president. So “gül” and “Gül” can be considered as different tokens for finding named entities better.

When case sensitivity is disabled, all tokens in the training corpus are kept in upper case using Turkish locale. In Turkish, upper case of character “i” is “İ” and lower case of character “I” is “ı”. During the estimation phase, frequencies are searched as all characters of the token are turned into upper case, again using Turkish locale.

3.1.7.4 Case Usage:

Case of a token can be a very useful hint to recognize a named entity. Most of the named entities of MUC begin with a capital letter. Only time and date named entities like “bugün” (means “today”) or “gelecek yıl” (means “next year”) do not begin with a capital letter. Other than that, most of the person names, locations, organizations begin with a capital letter. For organizations, all characters can be upper case too, like “TRT” (“Türkiye Radyo ve Televizyon Kurumu” meaning “Turkish Radio and Television Corporation”).

Some tokens (like punctuations, numbers, dates etc) do not have a value in terms of case. Case usage is inapplicable for these tokens, or the tokens which include one or more of these tokens.

When case usage feature is enabled, a token can be mapped one these four values:

- All lower case
- First letter upper case
- All upper case
- Inapplicable

When case usage is enabled; in the training phase, all tokens in the training data is mapped to one of these four values. These values are kept just like the token itself and used in estimation phase.

"Genelkurmay Başkanı Orgeneral **İlker Başbuğ**'un mesajının okunmasından sonra öğrenciler tarafından şiirler okundu."

In the above example, again, consider that, we are calculating the probability of "İlker Başbuğ" to be named entity.

When calculating probability of the inside of the named entity we map "İlker" and "Başbuğ" tokens into case values.

İlker -> First letter upper case

Başbuğ -> First letter upper case

For both of these tokens, training data will be searched for the frequency of "First letter upper case" to be inside of a named entity. This probability will be used along with the frequency of token itself to be inside a named entity.

Same calculations will be done for the tokens before or after named entity.

Applying case usage for before and after the named entity may not provide much help for finding the named entity. Applying this feature for only inside of the named entity is left as a future work. Results of doing this shall be observed and actions shall be taken according to these observations.

3.1.7.5 Length Usage:

Length usage is similar to case usage. Tokens are mapped to length values. Current length values are:

- Zero length
- Singleton

- Doubleton
- Tripleton
- Quadrupleton
- Long

Just like case usage, frequencies of lengths are kept for training data. Those frequencies are used when probability of a token for being inside/before/after a named entity.

3.1.7.6 Alphanumeric Feature Usage:

Alphanumeric feature usage is also similar to case usage. Tokens are mapped to alphanumeric feature values.

Current alphanumeric values are:

- Alpha
- Numeric
- Alphanumeric
- Inapplicable

Just like case usage, frequencies of alphanumeric feature values are kept for training data. Those frequencies are used when probability of a token for being inside/before/after a named entity.

3.1.7.7 NAF (Nymble Alike Features):

Nymble [16] is inspiration point of Nymble alike features (NAF). These features are subset of the features of original Nymble system. It can be considered as superset of some of the previous features.

Current NAF are:

- Two digit number
- Four digit number
- Alphanumeric
- Other number
- All capital
- First capital
- Lower case
- Other

A token can fit into more than one of these patterns. The values are put in order according to their ability for recognition. More salient patterns appear previously than the others. Then the most useful feature of the token will be used.

Just like case usage, frequencies of NAF values are kept for training data. Those frequencies are used when probability of a token for being inside/before/after a named entity.

3.1.7.8 Lexical Resource Usage:

Lexical data which are used in rule based systems can also be used as a feature of the learning system. Frequencies for lexical meaning of a token can be kept on the training phase.

When calculating an estimation for a group of tokens in the estimation phase, these frequencies can be used.

3.1.7.9 Coefficients:

Coefficients can be used for any of the previous features. If we think case usage is more important than the length, than we can give a higher coefficient to the case usage feature.

3.2 Hybrid System

Proposed hybrid system is based on employment of Bayesian learning system together with the rule based system presented in [12, 13, 14]. Three different datasets will be referred in following sections:

- "Training data" is the tagged dataset used for training of the learning system.
- "Test data" is the untagged document, inside of which recognizers will find the NEs.
- "Output data" is the tagged document which is the output of the recognizer. Output data is basically test data including tagged estimations of the recognizer.

Usage of these datasets in the hybrid system is the same as usage of datasets for the learning based recognizer. In hybrid system, the output data of rule based system produced for the test data is used by learning based system to create a hybrid output. We have developed two different versions of hybrid system. These two versions will be referred as "training phase hybrid" and "estimation phase hybrid".

3.2.1 Training Phase Hybrid

In training phase hybrid system, learning based system uses output of rule based system as training data. Learning based system parses the output of the rule based system and retrieves estimated NEs by the rule based system. Then frequencies of tokens inside, before and after these estimations are kept and merged with token frequencies obtained from training data.

Estimation phase and output phase of the hybrid system is the same as learning system. However using token frequencies derived from rule based system, affects probability of tokens to be a NE. Estimation produced by the hybrid system includes some of the NEs which are not produced by the learning based system itself.

3.2.2 Estimation Phase Hybrid

Estimation phase hybrid system uses the output of the rule based system when giving probability for token groups to be NE. The phase that the probability for each possible token group is calculated is the estimation phase. Therefore, the name of this hybrid system is estimation phase hybrid.

In the beginning of the estimation phase, the output of the rule based system is parsed and NEs estimated by rule based system are retrieved. These estimations are kept along with their positions inside the document. When calculating probabilities for each possible token group; the decision of the rule based system is also calculated like a special feature. Rule based system output only affects the probability created by NE itself; it does not affect the probability calculated for before the NE or after the NE. Probability of a token group to be a NE can have three different values as rule based system effect:

- If all of a token group is tagged as NE, which is called exact match, then rule based probability is calculated as 1.0.
- If some part of a token group is tagged as NE, which is called partial match, then rule based probability is calculated as 0.5.
- If any token inside a token group is not tagged as NE, which is called no match, then rule based probability is 0.

When checking if a group is tagged inside the rule based system, also position of the token group is also important. If position of the token does not match, it does not count as exact or partial match.

For example, if the test document contains "Savaş" token as 300th token and 500th token. If "Savaş" token which is the 300th token is tagged by the rule based system and 500th token is not tagged, then the learning based system calculates "rule based effect" of the 300th token as 1.0 but "rule based effect" of the 500th token is calculated as 0.

CHAPTER 4

EVALUATION AND DISCUSSION

Throughout the evaluation we have used different datasets. One of the datasets that we used is a subset of Metu Turkish Corpus (MTC) [19]. MTC consists of 2 million words of Turkish samples. Named entities are not tagged in the documents of MTC. We have used a tagged subset for training and testing purposes which includes approximately 100,000 tokens. This dataset is mostly consisted of articles from news domain. Throughout the chapter, actually this dataset will be referred as MTC. Also Child Stories [20, 21] dataset is used for testing (estimation and evaluation of these tests). These datasets are also used by Küçük and Yazıcı for the rule-based recognition system [11, 12, 13]. Contents of the datasets are given in *Table 4.1*.

Table 4.1 Counts of words and Named Entities inside datasets

Dataset	Words	Total NEs	Person	Location	Organization	Date	Time	Money	Percent
Metu Turkish Corpus	101,700	11,206	3,280	2,470	3,124	1,360	53	510	409
Child Stories	19,000	1,084	836	157	6	55	-	-	-

For statistical learning system, a huge amount of training and testing data is needed. Since there is lack of that kind of annotated data, we used these two datasets throughout our tests.

4.1 Tenfold Cross Validation of MTC

Cross-validation is a well known and well accepted methodology in statistics. Tenfold and threefold cross-validation techniques are also commonly used in the evaluation of learning based systems. During tenfold cross validation, the dataset is partitioned into ten parts. Then each part is tested. On test of each part, system uses other nine parts as training data and creates estimations for the tested part. These estimations are compared with annotated form of the part and results are kept.

Most realistic performance of the system can be observed with tenfold cross evaluation on Metu Turkish Corpus since it is the largest dataset we have. Results for the tenfold evaluation are given in tables 4.2 - 4.9 according to NE type.

During the evaluation of the proposed NER systems; precision, recall and F-Measure values of the system is used. One approach during evaluation is to give credit to only exact matches, that type of NE and each token of NE is found, by the system. Formulas for this evaluation metric are given below.

$$Precision = \frac{Correct}{Correct + Spurious} \tag{4.1}$$

$$Recall = \frac{Correct}{Correct + Missing} \tag{4.2}$$

$$F\text{-}Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.3}$$

Another evaluation approach also gives credit to partial matches, where type of the NE matches but the tokens of NE matches partially as presented in [34]. The precision, recall and F-Measure formulas for this evaluation approach are given below.

$$Precision = \frac{Correct + 0.5 * Partial}{Correct + Spurious + 0.5 * Partial} \tag{4.4}$$

$$Recall = \frac{Correct + 0.5 * Partial}{Correct + Missing + 0.5 * Partial} \tag{4.5}$$

$$F\text{-}Measure = \frac{2 * Precision * Recall}{Precision + Recall} \qquad (4.6)$$

In these formulas, "correct" is the count of the estimations that exactly matches to both type and span of a NE. "Partial" is the count of estimations that matches to a NE partially. When type of estimation matches with type of NE and some tokens of NE do not exist in estimation or some extra tokens exist in the estimation, this means estimation matches to the NE partially. "Spurious" is the count of the estimations which do not match to a NE in the answer set. "Missing" is the count of NEs which are not estimated exactly or partially by the system.

Evaluation results in this chapter employ the second formula set which gives credit for partial matching.

Table 4.2 Tenfold cross evaluation for Person named entities on MTC

Parts	NE Count	Precision	Recall	F-Measure
Part 1	298	90,55	80,92	85,47
Part 2	329	92,2	87,25	89,66
Part 3	283	94,41	94,41	94,41
Part 4	339	97,53	88,18	92,62
Part 5	389	94,52	92,12	93,3
Part 6	351	96,49	85,67	90,76
Part 7	353	94,6	86,44	90,34
Part 8	319	95,23	82,44	88,38
Part 9	335	93,15	82,99	87,78
Part 10	284	94,88	84,12	89,18
Total	**3280**	**94,41**	**86,54**	**90,26**

Table 4.3 Tenfold cross evaluation for Location named entities on MTC

Parts	NE Count	Precision	Recall	F-Measure
Part 1	302	91,42	82,77	86,88
Part 2	240	89,8	86,72	88,24
Part 3	284	97,13	85,25	90,8
Part 4	265	93,59	79,42	85,92
Part 5	257	91,42	80,84	85,81
Part 6	255	90,69	82,29	86,29
Part 7	243	93,58	86,44	89,87
Part 8	192	91,02	81,72	86,12
Part 9	257	90,54	77,37	83,44
Part 10	175	90,54	82,71	86,45
Total	**2470**	**92,10**	**82,54**	**87,03**

Table 4.4 Tenfold cross evaluation for Organization named entities on MTC

Parts	NE Count	Precision	Recall	F-Measure
Part 1	369	88,43	87,37	87,9
Part 2	293	88,02	84	85,96
Part 3	221	84,65	88,47	86,52
Part 4	314	93,06	80	86,04
Part 5	317	83,57	90,12	86,72
Part 6	344	91,18	88,85	90
Part 7	358	91,29	93,83	92,54
Part 8	317	89,66	87,18	88,4
Part 9	308	86,81	87,41	87,11
Part 10	283	89,45	84,46	86,89
Total	**3124**	**88,78**	**87,29**	**87,96**

Table 4.5 Tenfold cross evaluation for Date named entities on MTC

Parts	NE Count	Precision	Recall	F-Measure
Part 1	154	89,22	53,79	67,12
Part 2	143	81,97	80,65	81,3
Part 3	120	95,63	82,94	88,83
Part 4	120	94,15	80,9	87,03
Part 5	117	89,07	78,74	83,59
Part 6	130	88,66	76,11	81,9
Part 7	131	91,35	78,6	84,5
Part 8	132	92,6	78,48	84,95
Part 9	161	92,62	82,48	87,26
Part 10	152	86,96	81,97	84,39
Total	**1360**	**90,07**	**77,19**	**82,83**

Table 4.6 Tenfold cross evaluation for Time named entities on MTC

Parts	NE Count	Precision	Recall	F-Measure
Part 1	4	71,43	71,43	71,43
Part 2	5	46,67	77,78	58,33
Part 3	7	80	66,67	72,73
Part 4	10	100	90	94,74
Part 5	4	71,43	71,43	71,43
Part 6	2	100	50	66,67
Part 7	9	100	75	85,71
Part 8	6	100	63,64	77,78
Part 9	4	75	100	85,71
Part 10	2	100	50	66,67
Total	**53**	**86,13**	**75,17**	**78,63**

Table 4.7 Tenfold cross evaluation for Money named entities on MTC

Parts	NE Count	Precision	Recall	F-Measure
Part 1	22	63,64	87,5	73,68
Part 2	43	86,11	100	92,54
Part 3	45	73,47	94,74	82,76
Part 4	52	61,06	97,18	75
Part 5	56	75	94,29	83,54
Part 6	59	81,61	85,54	83,53
Part 7	58	87,34	92	89,61
Part 8	51	63,1	100	89,79
Part 9	48	77,52	94,52	85,19
Part 10	76	85,37	98,13	91,3
Total	**510**	**76,65**	**94,71**	**85,56**

Table 4.8 Tenfold cross evaluation for Percent named entities on MTC

Parts	NE Count	Precision	Recall	F-Measure
Part 1	19	84	100	91,3
Part 2	63	92,21	97,26	94,67
Part 3	70	95,18	95,18	95,18
Part 4	36	90,91	100	95,24
Part 5	22	83,33	100	90,91
Part 6	27	88,89	100	94,12
Part 7	50	87,1	81,82	84,38
Part 8	38	81,48	100	89,8
Part 9	44	90,16	100	94,83
Part 10	40	77,78	95,45	85,71
Total	**409**	**88,27**	**96,09**	**91,84**

Table 4.9 Tenfold cross evaluation for All named entities on MTC

Parts	NE Count	Precision	Recall	F-Measure
Part 1	1168	89,25	80,26	84,01
Part 2	1116	88,84	86,45	87,57
Part 3	1030	92,25	89,15	90,47
Part 4	1136	93,16	83,91	87,94
Part 5	1162	89,07	87,91	88,28
Part 6	1168	91,87	85,07	88,25
Part 7	1202	92,43	87,78	89,95
Part 8	1055	90,44	84,61	87,60
Part 9	1157	89,98	84,03	86,72
Part 10	1012	90,04	85,08	87,33
Total	**11206**	**90,74**	**85,40**	**87,79**

4.2 Evaluation of Child Stories

“Child Stories” dataset [20, 21] does not have enough tokens or named entities for reaching high evaluation results. Therefore another approach is used for evaluation of this dataset. MTC is used as training dataset. The best results of the system are observed when testing “Child Stories” dataset using MTC as training set. For these tests, Child Stories dataset is split into three parts. Each of these three parts is evaluated where other two parts and MTC are training datasets. During the evaluation, BSD=1, ASD=1, NAF=Active features applied. The results of the evaluation are given in Table 4.10.

The average of all three parts for all named entities is 88.82%. This result is a very promising result for a Bayesian based statistical learner. This result can be increased in the future with addition of new features or co-operation with other learning approaches or directly another learner.

Table 4.10 Best results with using MTC as training data and Child Stories as testing data

	NE Counts					Max F-Measures (%)				
	Total	Person	Location	Organization	Date	Person	Location	Organization	Date	Total
Part1	347	259	60	0	28	99,03	77,86	0	76	93,51
Part2	377	296	65	1	15	97,14	67,72	1,47	80	91,13
Part3	330	281	32	5	12	86,43	54,24	18,18	58,33	81,25
Total	1054	836	157	6	55	94,13	68,85	15,4	73,24	88,82

4.3 Effects of Different Features

The learning based system we have used utilizes many different features. All of these features have positive or negative effects for recognition of named entities. Some of the features can be turned on and off (like case usage or length usage); others are numeric values (like coefficients or after surrounding distance). In order to observe effect of each feature to named entity recognition, we have determined a base feature set. In base feature set, coefficients all are 1, ASD and BSD are 0, and all of other features are off. Tenfold cross validation results are calculated for the base feature set and effect of a feature is determined with turning it on.

Precision, recall and F-Measure values are given for each NE type are given in table 4.11 for base feature set.

Table 4.11 Result of tenfold cross validation of MTC with base feature set

Base Feature Set				
	Precision	Recall	F-Measure	NE Count
Person	74,2	74,63	74,49	3280
Location	84,26	83,85	84,04	2470
Organization	75,05	73,51	74,3	3124
Date	76,34	63,93	69,46	1360
Time	68,89	58,49	63,65	53
Money	44,07	75	55,35	510
Percent	51,11	56,11	53,63	409
Total	74,67	74,32	74,25	11206

Case Usage:

Effect of the case usage feature is given in Table 4.12. This table also includes differences on precision, recall and F-Measure values for each NE type after turning case usage feature on.

It can be observed that case usage increases F-Measures for the person, location and organization NE types. For person NE type, both precision and recall are increased. When case usage is active, some names and surnames which do not exist on training dataset might be recognized. Since most of the person type NEs begin with capital letter, recognizer keeps statistics of tokens begin with capital letter and uses this data for finding NEs which exist or do not exist in training dataset. For the same reason, increase of recognition of location and organization NE types are also expected and observed.

On some NE types like time, money and percent; there is almost no change. Since the NEs are consisted of mostly numbers or lower case tokens, recognition of these NE types don’t change with usage of their cases.

A decrease has been observed for recognition of date NE type. Some date type NEs contains tokens which begin with capital letter. For example “Haziran 2008” is a date. After system is trained with using case data, that might cause some false estimations. Some tokens that begin with capital letter can be tagged as date type NEs even if they are not.

Table 4.12 Results of tenfold cross validation of MTC with case usage feature is on and differences of these results from base feature set

Case Usage						
	Precision	Recall	F-Measure	Precision Dif.	Recall Dif.	F-Measure Dif.
Person	74,83	75,17	75,08	0,63	0,54	0,59
Location	84,25	84,15	84,17	-0,01	0,3	0,13
Organization	76,07	73,66	74,82	1,02	0,15	0,52
Date	75,65	63,86	69,14	-0,69	-0,07	-0,32
Time	68,89	58,49	63,65	0	0	0
Money	44,52	74,02	55,38	0,45	-0,98	0,03
Percent	51,11	56,11	53,63	0	0	0
Total	75,08	74,53	74,56	0,40	0,21	0,31

Case Sensitive:

Effect of the case sensitive mode is given in Table 4.13. This table also includes differences in precision, recall and F-Measure values for each NE type after turning case sensitive mode on.

Table 4.13 Results of tenfold cross validation of MTC with case sensitive mode is on and differences of these results from base feature set

Case Sensitive						
	Precision	Recall	F-Measure	Precision Dif.	Recall Dif.	F-Measure Dif.
Person	76,21	76,02	76,15	1,38	0,85	1,07
Location	84,83	83,87	84,3	0,58	-0,28	0,13
Organization	75,3	76,84	76,05	-0,77	3,18	1,23
Date	74,87	65,63	69,81	-0,78	1,77	0,67
Time	67,39	58,49	63,88	-1,5	0	0,23
Money	44,88	74,8	55,95	0,36	0,78	0,57
Percent	51,11	56,11	53,63	0	0	0
Total	75,31	75,85	75,35	0,64	1,54	1,10

Alphanumeric Feature Usage:

Effect of the alphanumeric feature usage is given in Table 4.14. This table also includes differences in precision, recall and F-Measure values for each NE type after turning alphanumeric feature usage on.

Table 4.14 Results of tenfold cross validation of MTC with alphanumeric feature usage is on and differences of these results from base feature set

Alphanumeric Feature Usage						
	Precision	Recall	F-Measure	Precision Dif.	Recall Dif.	F-Measure Dif.
Person	74,19	74,66	74,5	-0,01	0,03	0,01
Location	84,39	83,81	84,08	0,13	-0,04	0,04
Organization	74,78	73,8	74,3	-0,27	0,29	0
Date	74,93	62,65	68,13	-1,41	-1,28	-1,33
Time	68,89	58,49	63,65	0	0	0
Money	44,19	74,51	55,26	0,12	-0,49	-0,09
Percent	50,11	55,62	52,82	-1	-0,49	-0,81
Total	74,42	74,20	74,06	-0,25	-0,11	-0,18

Length Usage:

Effect of the length usage feature is given in Table 4.15. This table also includes differences in precision, recall and F-Measure values for each NE type after turning length usage on.

Table 4.15 Results of tenfold cross validation of MTC with length usage feature is on and differences of these results from base feature set

Length Usage						
	Precision	Recall	F-Measure	Precision Dif.	Recall Dif.	F-Measure Dif.
Person	73,17	73,55	73,4	-1,03	-1,08	-1,09
Location	84,37	83,79	84,06	0,11	-0,06	0,02
Organization	74,98	73,61	74,3	-0,07	0,1	0
Date	75,9	63,24	68,86	-0,44	-0,69	-0,6
Time	68,89	58,49	63,65	0	0	0
Money	43,84	73,92	54,84	-0,23	-1,08	-0,51
Percent	51,11	56,23	53,68	0	0,12	0,05
Total	74,31	73,89	73,84	-0,36	-0,43	-0,41

NAF Usage:

Effect of the NAF usage is given in Table 4.16. This table also includes differences in precision, recall and F-Measure values for each NE type after turning NAF usage on.

Table 4.16 Results of tenfold cross validation of MTC with NAF usage is on and differences of these results from base feature set

NAF						
	Precision	Recall	F-Measure	Precision Dif.	Recall Dif.	F-Measure Dif.
Person	74,85	75,17	75,09	0,65	0,54	0,6
Location	84,36	84,09	84,19	0,1	0,24	0,15
Organization	76,39	73,34	74,81	1,34	-0,17	0,51
Date	74,33	63,68	68,53	-2,01	-0,25	-0,93
Time	68,89	58,49	63,65	0	0	0
Money	43,14	73,33	54,07	-0,93	-1,67	-1,28
Percent	50,57	54,52	52,49	-0,54	-1,59	-1,14
Total	74,95	74,32	74,39	0,28	0,00	0,14

Lexical Resource Usage:

Effect of the lexical resource usage is given in Table 4.17. This table also includes differences in precision, recall and F-Measure values for each NE type after turning lexical resource usage on.

Table 4.17 Results of tenfold cross validation of MTC with lexical resource usage is on and differences of these results from base feature set

Lexical Resource Usage						
	Precision	Recall	F-Measure	Precision Dif.	Recall Dif.	F-Measure Dif.
Person	68,35	81,6	74,48	-5,85	6,97	-0,01
Location	83,36	83,06	83,15	-0,9	-0,79	-0,89
Organization	69,59	74,71	72,08	-5,46	1,2	-2,22
Date	73,11	65,59	68,99	-3,23	1,66	-0,47
Time	69	65,09	67,04	0,11	6,6	3,39
Money	43,77	54,41	48,51	-0,3	-20,59	-6,84
Percent	48,59	54,89	51,5	-2,52	-1,22	-2,13
Total	70,75	75,77	73,00	-3,93	1,45	-1,25

BSD Feature:

Effect of the BSD is given as a diagram in figure 4.1. This diagram includes differences on F-Measure values for BSD values 1-5. In base feature set, BSD value is zero. In the graphic, x axis shows value of BSD and y axis shows the difference of F-Measure for a specified NE type or overall.

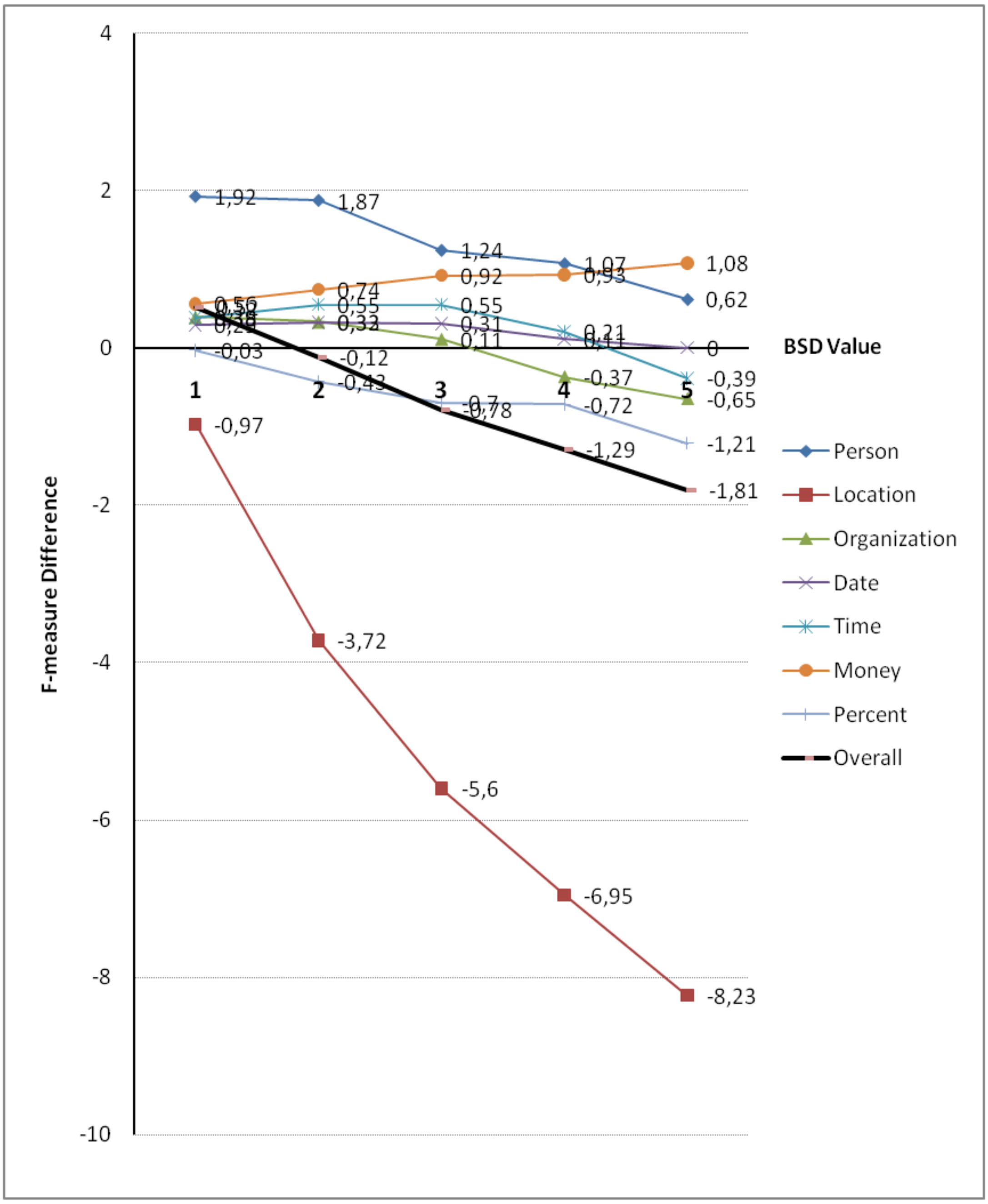

Figure 4.1 BSD value and F-Measure difference diagram

ASD Feature:

Effect of the ASD is given as a diagram in figure 4.2. This diagram includes differences on F-Measure values for ASD values 1-5. In base feature set, ASD value is zero. In the graphic, x axis shows value of ASD and y axis shows the difference of F-Measure for a specified NE type or overall.

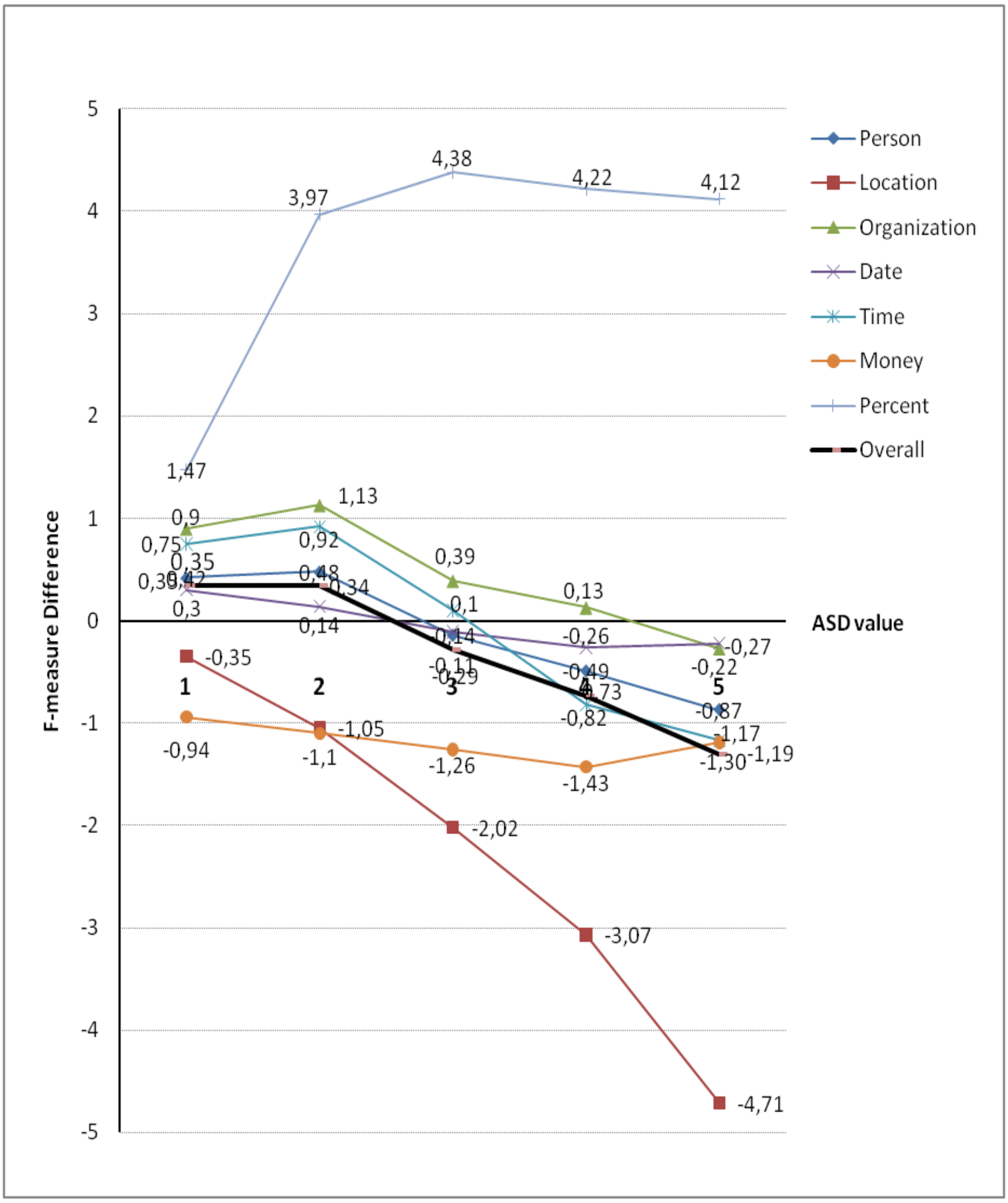

Figure 4.2 ASD value and F-Measure difference diagram

Summary of feature affects are given in Table 4.18. In the table, affect of each feature is given when the feature is turned on. When the F-Measure value increases more than 0.1, for a named entity type, then the feature affect is considered as “positive”. When the F-Measure value decreases more than 0.1, for a named entity type, then the feature affect is considered as “negative”. If the difference on the F-Measure value is between -0.1 and 0.1, then the feature is considered as it has no affect for this named entity type. When a feature has no affect for a named entity type, the affect is shown with “-” mark.

Table 4.18 Summary of affects of each feature for entity types

Summary						
	Case Usage	Case Sensitive	Alphanumeric Feature	Length Usage	NAF	Lexical Resources
Person	Positive	Positive	-	Negative	Positive	-
Location	Positive	Positive	-	-	Positive	Negative
Organization	Positive	Positive	-	-	Positive	Negative
Date	Negative	Positive	Negative	Negative	Negative	Negative
Time	-	Positive	-	-	-	Positive
Money	-	Positive	-	Negative	Negative	Negative
Percent	-	-	Negative	-	Negative	Negative
Total	Positive	Positive	Negative	Negative	Positive	Negative

Combination of Features:

We were expecting for a feature to affect the results similarly when they are combined with other features. Generally this statement is true but sometimes; a feature can affect the results differently, when it is combined with some other features in comparison with the affect of the feature itself. Sometimes a feature, which decreases F-Measure when it is applied itself, can increase F-Measure when it is combined with some other features.

For example, ASD and BSD are decreases F-Measure of location name recognition. But when ASD=1, BSD=1 and NAF combination is used, location name recognition is higher than NAF itself. That means ASD and BSD affected positively to recognition F-Measure when they are combined with NAF.

Since there are lots of features, observing results for combination of each of them takes considerable time and effort. Instead of this we made a heuristic sampling and tried to pick the features that would give best results when they are activated together.

4.4 Evaluation of Hybrid System

For evaluation of hybrid system, a new sample text has been chosen from METU Turkish Corpus, and it is used as test data. This sample is tagged manually to use as answer set. This sample has not been used before for evaluating the Bayesian learning based system or the rule based system. Also this sample is not included in subset of MTC which is used for training.

Table 4.18 includes evaluation results for rule based and hybrid system (rote learning + rule based) studies presented in [12, 13, 14]. Also the evaluation results of the Bayesian learning system, and two different hybrid systems presented in this thesis are included in table 4.19.

Table 4.19 Results for rule based, learning based and hybrid systems

System	Precision	Recall	F-Measure
Rule Based [14]	94,71	81,36	87,53
Bayesian Learning Based	96,16	81,82	88,41
Hybrid (rule based + rote learning) [14]	94,27	81,9	87,65
Hybrid (rule based + Bayesian learning, training phase hybrid)	92,68	87,56	90,05
Hybrid (rule based + Bayesian learning, estimation phase hybrid)	93,38	89,57	91,44

Table 4.19 shows that both hybrid systems presented in this thesis gives higher F-Measure values when compared to both of rule based system and Bayesian learning system.

Precision values of the rule based system and learning system are higher than both of the hybrid systems but recall values of hybrid systems are higher than both of rule based system and learning system. Both rule based and Bayesian components employed in hybrid system affect the named entities found by hybrid system. Hybrid system produces more estimations than estimations produced by rule based and Bayesian system alone. Recall values of hybrid systems increase significantly since the output of the system includes more correct answers and total count of NEs does not change with estimation count. However precision of hybrid systems decreases a little since spurious estimation count also increases.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This thesis has focused on named entity recognition in Turkish using Bayesian learning approach and different features. Also this thesis explains the joint utilization of the learning based system with a rule-based system to create two hybrid systems.

Learning based approach is best suited for using the recognition system for interchangeable domains. It is aimed to find named entities from different domains after training the system on datasets from these domains. Because of the annotated data limitations, we could not test the performance of the system with a large training data (datasets including 100K or more named entities). Yet we tested system with the dataset we have and got good results for a Bayesian based recognizer. Using different features helped to improve performance of the system.

Another aim of this thesis is to observe effects of different features to a learning based recognition system. As a result we have found out that each type of named entity requires different feature sets for the peak F-Measure rates. Since the system is developed to extract one type of named entity, the best recognizer instance is used for extracting each type of named entity. Then these results are merged with a merging component.

With collaboration of rule-based recognition system and learning based recognition system, an increase of 3-4% on F-Measures is observed. Hybrid system can be improved for better performance. Currently; training phase hybrid system and estimation phase hybrid system are developed and evaluated.

The organization of hybrid system can be improved by using multiple instances of learning based system as a rule inside rule-based system. Since the threshold of learning based recognizer can be tuned, a high precision recognizer, a high recall instance, a best F-Measure instance can be used collaboratively. To improve the hybrid system, rule based recognizer can also be used as a feature of learning based recognizer.

The resulting learning based system has near 87.79% F-Measure value for news domain and 88% F-Measure value for child stories domain when partial extraction scoring included. These results are not the best results for Turkish NER, but they are promising for additional improvements of the system, considering the fact that the training datasets are not very large.

REFERENCES

[1] Automatic Content Extraction (ACE) Program.
http://www.itl.nist.gov/iad/mig/tests/ace/, accessed 04 September 2011.

[2] Information Extraction – Wikipedia.
http://en.wikipedia.org/wiki/Information_extraction, accessed 04 September 2011.

[3] Cross Validation – Wikipedia.
http://en.wikipedia.org/wiki/Cross-validation_(statistics), accessed 04 September 2011.

[4] Java SE Technologies.
http://www.oracle.com/technetwork/java/javase/overview/index.html, accessed 04 September 2011.

[5] Eclipse Project. http://www.eclipse.org/, accessed 04 September 2011.

[6] Message Understanding Conference (MUC).
http://www-nlpir.nist.gov/related_projects/muc/, accessed 04 September 2011.

[7] Named Entity Recognition (NER).
http://en.wikipedia.org/wiki/Named_entity_recognition, accessed 04 September 2011.

[8] Dayne Freitag. Information extraction from HTML: Application of a general learning approach. In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI), pages 517–523, 1998.

[9] Dayne Freitag. Machine Learning for Information Extraction in Informal Domains. Mach. Learn., 39(2-3):169–202, 2000.

[10] Dilek Küçük and Adnan Yazıcı. Identification of coreferential chains in video texts for semantic annotation of news videos. In Proceedings of the International Symposium on Computer and Information Sciences (ISCIS), pages 1–6, 2008.

[11] Dilek Küçük and Adnan Yazıcı. Employing named entities for semantic retrieval of news videos in Turkish. In Proceedings of the International Symposium on Computer and Information Sciences (ISCIS), pages 153–158, 2009.

[12] Dilek Küçük and Adnan Yazıcı. Named entity recognition experiments on Turkish texts. In Proceedings of the International Conference on Flexible Query Answering Systems (FQAS), pages 524–535, 2009.

[13] Dilek Küçük and Adnan Yazıcı. Rule-based named entity recognition from Turkish texts. In Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications (INISTA), pages 456–460, 2009.

[14] Küçük, D. and Yazıcı, A. A Hybrid Named Entity Recognizer for Turkish. Expert Systems with Applications. Volume 39, Issue 3, pp. 2733-2742, February 2012.

[15] Ahmet Hamdi Tanpınar. Beş Şehir. Dergah Publications, 2007.

[16] D. M. Bikel, R. Schwartz, and R. Weischedel. 1997. Nymble: A High Performance Learning Name-finder. In Proceedings of the 5th Conference on Applied Natural Language Processing ANLP-97), Washington. D.C. pp 194-201.

[17] Serhan Tatar. Automatic rule learning exploiting morphological features for named entity recognition in Turkish. Journal of Information Science Volume 37 Issue 2, pages 137-151, April 2011

[18] MUC. Named Entity Task Definition. July 1995.

[19] Bilge Say, Deniz Zeyrek, Kemal Oflazer, and Umut Ozge. Development of a corpus and a treebank for present-day written Turkish. In Proceedings of the 11th International Conference of Turkish Linguistics (ICTL), 2002.

[20] Rıfat Ilgaz. Bacaksız Kamyon Sürücüsü. Çınar Publications, 2003.

[21] Rıfat Ilgaz. Bacaksız Tatil Köyünde. Çınar Publications, 2003.

[22] Elaine Marsh, Dennis Perzanowski. MUC-7 Evaluation of IE Technology: Overview of Results. 29 April 1998

[23] Message Understanding Conference Proceedings MUC-7 Table of Contents. http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_toc.html, accessed 04 September 2011.

[24] Gökhan Tür, Dilek Hakkani-Tür, and Kemal Oflazer. A Statistical Information Extraction System for Turkish; Natural Language Engineering, 9:181–210, June 2003.

[25] Reyyan Yeniterzi. Exploiting Morphology in Turkish Named Entity Recognition System. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics – Student Session (ACL 2011), Portland, Oregon, USA, 2011.

[26] S. Ozkaya, B Diri. Named Entity Recognition by Conditional Random Fields from Turkish informal texts. 2011 IEEE 19th Conference on Signal Processing and Communications Applications (SIU), 20-22 April 2011.

[27] Jim Cowie and Yorick Wilks. Information Extraction. Communications of the ACM Volume 39 Issue 1, Jan. 1996 New York, NY, USA

[28] Ralph Grishman. Information Extraction: Techniques and Challenges. SCIE '97: International Summer School on Information Extraction (1997), pp. 10-27.

[29] Dayne Freitag and Nicholas Kushmerick. Boosted Wrapper Induction. In Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pages 577–583. AAAI Press, 2000.

[30] Yaoyong Li, Kalina Bontcheva and Hamish Cunningham. SVM Based Learning System for Information Extraction. In Proceedings of Sheffield Machine Learning Workshop, Lecture Notes in Computer Science, Springer Verlag, 2005.

[31] Y Li, J Shawe-Taylor. The SVM with uneven margins and Chinese document categorization. In Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation (PACLIC17), pages 216–227, Singapore, Oct. 2003.

[32] H Isozaki, H Kazawa. Efficient Support Vector Classifiers for Named Entity Recognition. In Proceedings of the 19th International Conference on Computational Linguistics (COLING'02), pages 390–396, Taipei, Taiwan, 2002

[33] J Mayfield, P. McNamee, C Piatko. Named entity recognition using hundreds of thousands of features. In Walter Daelemans and Miles Osborne, editors, Proceedings of CoNLL-2003, pages 184–187. Edmonton, Canada, 2003.

[34] D. Maynard, V. Tablan, C. Ursu, H. Cunningham and Y. Wilks, Named entity recognition from diverse text types. In Proceedings of the Conference on Recent Advances in Natural Language Processing(RANLP) , 2001.

[35] R. C. Holte. Very simple classification rules perform well on most commonly used datasets. In Machine Learning, 11:63–91, 1993.

[36] P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classifier. In Proceedings of the Thirteenth International Conference on Machine Learning (ICML-96), pages 105–112, July 1996.