A TV CONTENT AUGMENTATION SYSTEM EXPLOITING RULE BASED NAMED ENTITY RECOGNITION METHOD

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

BY

YUNUS EMRE IŞIKLAR

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

SEPTEMBER 2014

Approval of the thesis:

A TV CONTENT AUGMENTATION SYSTEM EXPLOITING RULE BASED NAMED ENTITY RECOGNITION METHOD

submitted by **YUNUS EMRE IŞIKLAR** 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. Nihan Kesim Çiçekli _____
Supervisor, **Computer Engineering Dept, METU**

Examining Committee Members:

Prof. Dr. Ali Doğru _____
Computer Engineering Dept., METU

Prof. Dr. Nihan Kesim Çiçekli _____
Computer Engineering Dept., METU

Prof. Dr. Ferda Nur Alpaslan _____
Computer Engineering Dept., METU

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

Msc. Deniz Kaya _____
Arçelik A.Ş.

Date: 05.09.2014

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: YUNUS EMRE IŞIKLAR

Signature:

ABSTRACT

A TV CONTENT AUGMENTATION SYSTEM EXPLOITING RULE BASED NAMED ENTITY RECOGNITION METHOD

Işıklar, Yunus Emre

M.S., Department of Computer Engineering

Supervisor: Prof. Dr. Nihan Kesim Çiçekli

September 2014, 76 pages

In this thesis, a TV content augmentation system taking the advantage of named entity recognition methods is proposed. The system aims to automatically enhance TV program contents by retrieving context related data and presenting them to the viewers without any necessity of another device. In addition to conceptual description of the system, a prototype implementation is developed and demonstrated with predefined TV programs. The implementation utilizes Electronic Program Guide (EPG) data of programs crawled from web resources in order to extract named entities such as person names, locations, organizations, etc. For this purpose, a rule based Named Entity Recognition (NER) algorithm is developed for Turkish texts. Detailed information about the extracted entities is retrieved from Wikipedia after semantic disambiguation and its summarized form is presented to the users. A set of experiments have been conducted on two different data sets in order to evaluate the performance of the rule based NER algorithm and the behavior of the TV content augmentation system. The experimental results show that the content augmentation with NER methods is quite successful in TV domain especially for channels broadcasting news and series.

Keywords: Content Augmentation, Connected TV, EPG (Electronic Program Guide), Named Entity Recognition (NER), Semantic Disambiguation

ÖZ

KURAL TABANLI VARLIK İSMİ TANIMA YÖNTEMİ KULLANARAK TELEVİZYON İÇERİĞİNİ ZENGİNLEŞTİRME SİSTEMİ

Işıklar, Yunus Emre

Yüksek Lisans, Bilgisayar Mühendisliği

Tez Yöneticisi: Prof. Dr. Nihan Kesim Çiçekli

Eylül 2014, 76 sayfa

Bu tezde, kural tabanlı varlık ismi tanıma yönteminden yararlanan bir televizyon içeriği zenginleştirme sistemi önerilir. Sistem televizyon programlarının içeriklerini, bağlamla ilgili verileri alarak, otomatik olarak artırmaya ve bunları kullanıcılara başka bir cihaz ihtiyacı olmaksızın sunmaya çalışır. Sistemin kavramsal tanımının yanı sıra, bir prototip uygulaması geliştirilmiş ve önceden tanımlanmış televizyon kanalları ile birlikte gösterilmiştir. Uygulama kişi, yer, organizasyon gibi varlık isimlerini çıkarmak için web kaynaklarından taranmış elektronik program rehberi bilgisinden faydalanır. Bu amaçla, Türkçe için kural tabanlı bir varlık ismi tanıma algoritması geliştirilmiştir. Semantik anlam ayrımından sonra, çıkarılan varlık isimlerinin detaylı bilgisi Vikipedi'den bulup getirilir ve özet hali kullanıcıya gösterilir. Kural tabanlı varlık ismi tanıma algoritmasının performansını ve televizyon içeriği zenginleştirme sisteminin davranışını değerlendirmek için iki farklı veri seti üzerinde bir dizi deney yürütülmüştür. Deney sonuçları, varlık ismi tanıma yöntemleriyle beraber içerik zenginleştirme sisteminin özellikle haber ve dizi yayınlayan kanallarda oldukça başarılı olduğunu göstermektedir.

Anahtar Kelimeler: İçerik Zenginleştirme, İnternet Bağlantılı Televizyonlar, EPG (Elektronik Program Rehberi), Varlık İsmi Tanıma (NER), Semantik Anlam Ayrımı

To my precious family

ACKNOWLEDGEMENTS

I would like to announce my deep gratitude to my supervisor Prof. Dr. Nihan K. Çiçekli for her valuable supervision, advice, useful critics and discussions throughout this study. It was really a pleasure working with such a friendly, thoughtful, intellectual and motivating supervisor.

I wish to express my endless thanks to every member of my family, especially to my mother Ayşe Işıklar and to my father Cumali Işıklar, for their unconditional love. Without their encouragements and advices, that journey would be endless for me.

I would like to thank the Scientific and Technological Research Council of Turkey (TÜBİTAK) for providing the financial means throughout this study with the project number of 112E111.

I am thankful to Burak Demirtaş, head of the METU-TECHNOPOLIS R&D Office of Arçelik, for his contributions on the topic, his shares and help for some parts of this study.

I am deeply grateful to Emrah Şamdan, Arda Taşçı, İrem Gökçe Aydın, İlker Argın and Ali Karakaya for their friendships, encouragement and support.

Finally, I wish to express my warmest thanks to Özgül Sarılı for her endless love, patience, care, motivation and most importantly morale support at every stage of this study. Without her love and friendship, this work would last longer and harder for me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVATIONS

ACE	(Automatic Content Extraction)
API	(Application Programming Interface)
CR	(Capitalization Rule)
EPG	(Electronic Program Guide)
IG	(Inflectional Group)
LG	(Local Grammar)
MUC	(Message Understanding Conference)
NER	(Named Entity Recognition)
NERC	(Named Entity Recognition and Classification)
PC	(Personal Computer)
RV	(Reporting Verbs)
TV	(Television)

CHAPTER 1

INTRODUCTION

1.1. Motivation

Television has provided rich content, delivering news, information and entertainment for many years. User viewing habits have changed in parallel with functionalities since the invention of the television. It is claimed that reasons of watching TV vary with respect to time and context [1]. Authors describe three levels of TV viewing:

- At low level of viewer engagement, users watch TV with minimum effort and the primary purpose of users is to relax.
- At medium level of viewer engagement, users watch programs which they are interested in periodically.
- At high level of viewer engagement, programs of specific interest like documentaries are primary concern of viewers.

Additionally, there are two fundamental cognitive modes, namely experiential mode and reflective mode, requiring different technological support [2]. The experiential mode is similar to low-level and medium level of viewer engagement; on the other hand reflective mode implies high level of viewer engagement. Several kinds of internal and external factors may lead users to alternate between these two modes [3].

Although viewers usually watch TV in the experiential mode and more relaxed way; viewer's mode can be more active and reflective thanks to the televisions with

connectivity features. In the last decade, with the plethora of the Internet, network connectivity has been integrated into new TV designs. In this manner, connected TVs, Smart TVs and Internet Protocol televisions have come into our lives and *"The TV is dead"* motto of just a few years ago has been completely changed [4].

While connected and smart TVs can access the web data, they do not combine TV and web experience effectively in order to provide more reflective cognitive mode. Therefore, users tend to search and learn about content while watching TV by using other devices such as smart phones or PCs. To illustrate, according to a Yahoo funded survey, 86% of the respondents in the US access mobile internet while watching TV [5]. The survey indicates that TV viewers usually exploit their mobile devices to access web and this adversely affects television experience. One of the reasons is that they want to learn more detailed and relevant information about the program or movie which is being watched. However, it may not be easy to reach required information and related content due to excessive results. Users also miss the content while the searching process. Therefore, viewers need TV contents with more additional information and new interfaces to learn details about the content without using any other devices. By constructing new functionalities and augmenting TV features such as providing additional information, viewer's mode can be more active and reflective. An important point to highlight is that TV augmented functionalities should not be complicated and our concern still must be user experience quality.

Our main motivation in this study is to design a TV content augmentation system retrieving additional information and presenting them to the viewers by preserving user experience quality for Turkish language.

1.2. Problem Definition and Our Approach

Enhancement of the content meaning and intelligibility by providing additional information is known to be content augmentation. We design a TV content augmentation system by information extraction methods. In order to construct such a system, the following issues should be dealt with: extraction of the content meaning,

retrieving additional related information and presenting additional data without annoying users and delay.

We first gather descriptions of TV programs in order to interpret the meaning. Electronic programming guide (EPG) data which is the description of the corresponding program, is utilized for this purpose. Since named entities are important words in program descriptions, we try to extract them in order to comprehend the content of the TV program. The most challenging part of this study is keyword extraction i.e. named entity recognition (NER). We have implemented a rule based NER method for Turkish texts to overcome this issue by utilizing language morphological structure and lexical resources. We define entity recognition rules by exploiting Turkish local grammar features for person, location and organization names.

Retrieving relevant data about the content is performed by collecting detailed information of extracted keywords from Wikipedia web site. While retrieving the description of a keyword we also deal with ambiguous named entities. We take the advantage of context, which the keyword appears, and exploit vector based similarity method [6] to determine the actual meaning of the named entity.

During the presentation of the relevant information to the users, we pay attention to the requirements of preserving the quality of TV viewing experience. We define the desired properties of TV content augmentation system as flexibility, sufficiency, simplicity and timeliness. Augmented content that will be presented to the viewers, is organized with respect to the desired properties. In order to fulfill simplicity and sufficiency requirements, we summarize additional information by constructing an information box including the most important data pieces.

While designing the whole TV system architecture, we also put emphasis on timeliness requirement. In order to provide the users with the additional information quickly whenever they request it, we collect and prepare additional relevant information as a

summary before the broadcasting of TV programs. We exploit a Turkish newspaper web site *Radikal*[1] to gather periodic EPG data of TV programs.

Towards our goal, we also evaluate EPG data of programs with respect to the number of named entities and keywords which can be augmented with additional data. We calculate the average number of named entities and augmented named entities for each program and extract top 5 channels according to their augmentation rate. By observing the results of the experiments, we try to understand the behavior of such a content augmentation system for TV.

1.3. Contributions

The main contributions of this thesis can be summarized as follows:

- We propose and implement a new design of TV content augmentation system for Turkish TV programs that can be adapted to Arçelik smart TVs. The implemented components can readily be used for different purposes such as text augmentation and can be extended to support other domains such as web streaming as well.
- We have implemented a named entity recognition algorithm for Turkish texts by exploiting the lexical and contextual features of the language. We have tried to keep the number of lexical resources to minimum and made our rule definitions comprehensive to be applicable on EPG data.
- By gathering EPG data of Turkish TV programs from web sources, we have constructed a beneficial data set of periodic program descriptions for NER research on this domain.
- While retrieving additional data about a named entity, we have implemented a semantic disambiguation module by exploiting Wikipedia as a lexical resource for Turkish named entities.

[1] http://www.radikal.com.tr/tvrehberi/

- In order to evaluate the behavior of the proposed system we have processed 2700 TV program descriptions and obtained significant results which may be good indicators of requirements for TV content augmentation system in real life.

1.4. Organization of Thesis

The thesis consists of 6 chapters including this first chapter.

Chapter 2 presents the related work in the literature on TV content augmentation systems, named entity recognition on Turkish and semantic disambiguation.

Chapter 3 explains the TV content augmentation system that we propose. Requirements for an ideal system are described and components of the system are explained in two parts, the server and the client side.

Chapter 4 introduces the prototype implementation of TV content augmentation system. Exploited tools and design issues of the system are given. In addition, components of the system are elaborated. The most important component in the system, which is NER implementation on Turkish texts, and details of rule based algorithm are described.

Chapter 5 presents experimental results of our rule based NER implementation, discussion of these results and error analysis of the algorithm. In addition, evaluation of TV content augmentation system is given with meaningful results about TV channels and programs.

Finally, in chapter 6 we conclude our thesis with a very brief summary of the study, discussions and possible future work that is feasible towards extending this work.

CHAPTER 2

LITERATURE SURVEY

There are four major steps in designing a TV content augmentation system: 1) gathering data about content from the Internet; 2) extraction of the keywords in the collected information; 3) relevant information retrieval via extracted words and semantic disambiguation; 4) presentation of the additional content on TV. The research issues in implementing such a system are concerned mainly with the actual design of such a system in general, named entity recognition methods in particular and disambiguation methods. In this chapter, we first present the existing TV content augmentation systems and their functionalities. We then give general information about named entity recognition and present different named entity recognition techniques especially for Turkish, which are used in information retrieval part of the content augmentation system. Lastly, we explain semantic disambiguation techniques.

2.1. Similar Systems on TV Domain

There are many TV content augmentation methods in the literature and prominent ones are explained below.

The work inspiring our study is InfoSip [7] which is a movie information retrieval application that analyzes the movie content and gives audiences information (overlaid onscreen) such as who the actor is, which song is played, etc. It can be said this work is one of the initial studies about content augmentation. To get such additional information, InfoSip uses image and audio processing methods to determine the people

on the screen. After the relevant information is extracted from web sources, it is presented together with the video. Geographic location through the zip code, if available, is also used to extract relevant information. This system provides enrichment and personalization on content, however the information it gives is limited and additional information is not detailed. Furthermore, televisions do not have powerful processors and sufficient resources; therefore implementing image processing techniques on TV is not a realistic solution. Even if image processing operation is performed on the server side, transaction of the images between server and client to extract meaningful information may hamper efforts to run system in real time.

Nadamoto and Tanaka, in their work [8], state that the fusion of TV and web content will shape next generation systems and there are two technical issues to implement such systems: how to extract related web pages and how to present them. They focused on the latter problem since Ma et al. (Ma, Nadamoto, & Tanaka, 2004) have researched the first problem. They have developed a 'TV-style presentation' prototype system which is capable of transforming the text and image based web content extracted from related web pages into audio-visual TV and fusing it with normal broadcasted TV program contents. Content augmentation is performed with the created web pages and all information is presented on TV. However, in this system TV viewing experience may become intrusive since all information is presented on TV like a web page. According to the users in the experiment [9], when the presented information is long and covers all screen, it is difficult to understand the related topic.

Newstream [10] provides additional information about visited web pages and videos which is being watched without depending on any platform. Users may view extra information immediately or later by using storage utility of Newstream. Related additional data created by using social networks may also be pushed to other devices such as TV, PC and mobile devices depending on the viewer's demands. The prominent property of this system is focusing on cross media content which distinguishes this study from other experiences. TV, PC and mobile devices remain in contact with each other and it may send additional information to the most feasible device for which the context is more meaningful. Communications between devices

are performed simultaneously and mobile devices can be used as a remote control to the TV and PC. Despite the multi-device and cross-medium approach, Newstream has limited resources for extra content. Since only social networks, which have limited viewer, are exploited to extract related content for the system, additional information may not be accurate and does not fulfill the requirements of the users. Extracting relevant knowledge may also be challenging because of redundant content in social media. Besides; even if relevant content is extracted, it includes general issues about context rather than particular information need.

Chattopadhyay and Pal indicated in their work [11] that TV services, such as customers can get additional related information from the Internet, are in demand according to the recent survey about user desires. They have developed a system [11] to recognize breaking news story and present detailed information about the breaking news on TV before broadcast content is made available. At the identification section, image processing techniques are used to extract candidate text regions from the video by using the localization approach. Keyword selection on the located text regions is performed by a heuristic approach which is based on some observations. The extracted keywords from each news event are investigated by Google search engine, and the relevant News and RSS feeds are fetched and presented on top of TV video. Although there is content enrichment on TV, it is implemented for only news story and does not support other video categories. Since brief word groups on video screen are used to extract keywords, it is impossible to gather adequate information about the topic from the internet.

ContextController [12] is a platform that augments the current audiovisual broadcast with related contextual information by using automatically collated videos. In this study, the authors focus on news related broadcast like Chattopadhyay's system and state that the viewer's understanding of the original content will be improved by providing the ability to discover new contextual information. The project also aims to enable users to provide summarized information which is externally sourced and improve the quality of TV viewing experience by using a second screen.

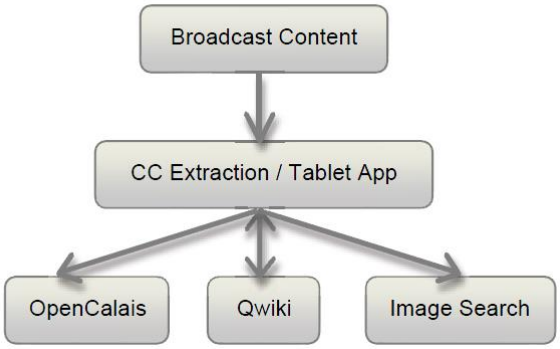

Figure 2-1 : ContextController System Overview

Augmentation is performed with the aid of available tools which are Closed Caption Extractor Tool, Open Calais, QWiki and Google Image Search API. Figure 2-1 shows the ContextController system overview. Once keywords and named entities are recognized by Open Calais, representative image for each keyword is identified with the aid of Google API. Finally, Qwiki API generates the summary video of a topic and it is transmitted to the second screen application (Figure 2-2). Presenting summarized relevant content as a video on second screen different from TV is a considerable functionality among similar systems. However, in such a system, viewers have to manage second device while watching TV and may miss the original content on TV screen.

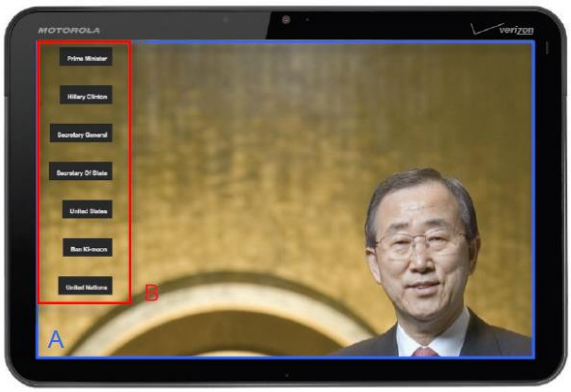

Figure 2-2 : ContextController Second Screen, Keyword: Ban Ki Moon

One of the most comprehensive researches about TV content augmentation is Prata and Chambel's work [3]. They have developed a new crossmedia approach to the content enrichment problem and taken the advantage of this while designing the whole system and TV application. Cognitive and affective aspects have been taken into account during the design of interaction with different media. According to Robinson et al. [13] the best approach is to investigate device characteristics and particular situations associated to its use instead of classification of devices. Thereby, from reasons and ways of watching TV to levels of users' attention during the viewing experience are examined in this study. The eiTV application (Figure 2-3), which is capable of creating additional information about video being watched, accessing and sharing that content via TV, PC and mobile devices, is implemented according to the addressed cognitive and affective aspects. Pre-established important keywords which are underlined are presented to the users with closed captions and when a keyword is chosen, a web page is generated on the server side. The user can access the additional information via PC or mobile device later. The drawback of the application is that keywords should be prepared for video content and it focuses on scientific words rather than other named entities. If keywords and relevant data is not prepared before corresponding TV program is broadcasted, retrieving additional data from the Internet may delay the content augmentation operation.

Figure 2-3: eiTV Application

2.2. Named Entity Recognition

In this section, general information about named entity recognition and classification (NERC) systems are presented. Different techniques and feature space for NERC are explained. The details of evaluation techniques and metrics are discussed.

2.2.1. NER Definition and Application Areas

Named entity recognition is a text processing task to identify information units like people, locations and organizations as well as percentage and numeric expressions including date, money and time. Recognizing these entities in natural language documents and unstructured texts is called "Named Entity Recognition and Classification (NERC)" which is one of the main information extraction (IE) tasks [14]. Even though entity types are not limited; ENAMEX (names), TIMEX (temporal expressions) and NUMEX (numerical expressions) are defined as three basic types of named entities by MUC [15] and CoNLL [16] conferences. Entity types extracted from documents in several domains such as financial texts [17], business information texts [18] and biomedical texts [19] can be used for automatic summarization, semantic multimedia annotation and other various purposes.

2.2.2. NER Techniques

Techniques of NER vary according to purposes and textual domains [14]. While most of the early studies exploit rule based techniques, the recent ones use learning techniques which are supervised, semi-supervised and unsupervised learning. Supervised learning, the most used technique in this field, is carried out by reading a large annotated corpora and extracting automatically rules based on discriminative features of entities [14]. The other technique, semi-supervised learning exploits relatively small annotated corpus and tries to identify contextual clues corresponding to annotated entities in order to extract named entity rules [14]. The last learning technique which does not use annotated corpora and utilizes dictionaries, lexical

patterns and statistics computed on a large annotated data is called unsupervised learning.

2.2.3. Feature Space for NERC

Characteristics of words or documents are described as a feature for NERC systems. According to Nadeau and Sekine, the most used features are word-level, list-lookup and document and corpus features. Word-level features refer to the word characteristics including case, punctuation, numerical value and special characters. Figure 2-4 shows the features of word-level with subcategories. One of the primary features in NERC is list-lookup feature which is often used with *gazetteer*, *dictionary* or *lexicon* terms [14]. These lists, which are used to ease recognition and classification tasks, can be categorized such as cities, persons, well known organizations etc. On the other hand, document and corpus features are described with respect to document and content characteristics. For example, meta-information of document such as e-mail header or word occurrences are good indicators for recognition and classification tasks.

Features	Examples
Case	- Starts with a capital letter - Word is all uppercased - The word is mixed case (e.g., ProSys, eBay)
Punctuation	- Ends with period, has internal period (e.g., St., I.B.M.) - Internal apostrophe, hyphen or ampersand (e.g., O'Connor)
Digit	- Digit pattern (*see section 3.1.1*) - Cardinal and Ordinal - Roman number - Word with digits (e.g., W3C, 3M)
Character	- Possessive mark, first person pronoun - Greek letters
Morphology	- Prefix, suffix, singular version, stem - Common ending (*see section 3.1.2*)
Part-of-speech	- proper name, verb, noun, foreign word
Function	- Alpha, non-alpha, n-gram (*see section 3.1.3*) - lowercase, uppercase version - pattern, summarized pattern (*see section 3.1.4*) - token length, phrase length

Figure 2-4: Word-level features [14]

2.2.4. Evaluation of NERC

According Nadeau et al., there are three main evaluation methods, namely MUC, exact-match and ACE[2].

MUC Evaluations: The performance of a NER system is measured with respect to its ability to find the correct type and ability to find exact text in Message Understanding Conference (MUC) events [20]. If the type of entity is correctly recognized regardless of boundaries as long as overlap, it is counted as a correct type. If entity boundaries are correctly recognized regardless of the type, it is counted as a correct text. The number of actual entities (ACTUAL), the number of entities retrieved by the system (RETRIVED) and the number of actual entities correctly recognized by the system

[2] https://www.ldc.upenn.edu/collaborations/past-projects/ace

(CORRECT) are calculated for both type and text. The final score f-measure is calculated by taking harmonic mean of precision and recall:

$$precision = \frac{correct}{retrieved}$$

$$recall = \frac{correct}{actual}$$

$$f - measure = \frac{2 * precision * recall}{precision + recall}$$

Exact-Match Evaluations: The difference between MUC and this evaluation technique is the matching criteria of entities. In this evaluation method, f-score is measured as MUC technique; however a named entity is counted as a correct entity only if both its boundaries and type exactly matched.

ACE Evaluation: The evaluation procedure of ACE is more complex compared to other techniques. In this method, the system is scored by also considering detailed evaluation issues such as partial matches, wrong type or even subtypes of entity. However, comparing ACE scores may be problematic when detailed parameters are not fixed [14].

2.3. NER on Turkish Texts

Since the most important part of our TV content augmentation system is NER and the system is designed for Turkish, we have researched NER studies in Turkish particularly.

NER is a long-studied topic, indeed the initial studies in this area were started in 1990s for English. Thereafter, several programs such as Message Understanding Conference (MUC) series have put emphasis on IE research especially NER for English, Arabic, Chinese and Japanese languages. In 1995, with this high interest of the research community, the success rates attained high level of annotation performance above 90

% that is close to human performance on English news text [21]. Nonetheless, information extraction and especially NER on Turkish texts is a research area that is not studied adequately. Limited number of researches on Turkish texts is addressed below:

The first remarkable NERC work that is tested on Turkish documents with relatively good results is the work by Cucerzan and Yarowsky [22] which implements a language independent system by combining morphological and contextual evidence. The aim of their study is to develop a language independent system for both entity recognition and classification without using any information about the text language and long word lists like persons and organizations. Since creating manually annotated corpora and multilingual word lists is a costly operation, it is not practicable to implement learning techniques such as supervised learning method for this system. The algorithm developed by Cucerzan et al. is based on morphological and contextual patterns of each different language. For instance, according to them suffixes such as "-escu", "-wski", "-ovic" and "-son" are the reliable indicators of a last name in Romanian, Polish, Serbo-Croatian and English respectively. Similarly, contextual patterns like "Mr." and "in" are perfect clues to determine entity types. By using these features, Cucerzan et al. construct a tree with patterns extracted from small lists at bootstrapping part. After generating distribution tree, which is called 'trie' in that work, named entities in the text are identified and classified by using this tree structure. Experiments of this algorithm are carried out for different languages and F-measure in Turkish texts is evaluated as 53.04%. Even though this is not a satisfactory result, it can be counted as reasonable for a multi-language NERC algorithm.

Tür et al. have also focused on morphological structure of the Turkish words and developed a statistical information extraction system for Turkish [23]. The primary concern of this work is the difference between Turkish morphology and other languages such as English. Since Turkish language has very productive agglutinative, the number of possible word forms (surface forms) of a root, which is used to build statistical models for IR tasks, is usually more than other languages like English (Figure 2-5).

Language	Vocabulary Size
English	97,734
Turkish	474,957
Turkish (only roots)	94,235

Figure 2-5: Comparison of the number of unique word forms in English and Turkish, in large text corpora [23]**.**

As seen in Figure 2-5, the vocabulary size of Turkish surface forms brings data sparseness in the training data. Therefore, the authors exploit the morphological analyses of the words. Inflectional groups (IGs) are described by decomposing and analyzing words morphologically. To illustrate, the word *sağlamlaştırdığımızdaki* represented as "sağlam+laş+tır+dı+ğı+mız+da+ki" has the following six IGs with respect to morphological decomposition.

1. Sağlam+Adj
2. Verb+Become
3. Verb+Caus+Pos
4. Adj+PastPart+P1sg
5. Noun+Zero+A3sg+Pnon+Loc
6. Adj

Although Tür et al. utilize inflectional groups for sentence segmentation, topic segmentation and name tagging; from the perspective of our thesis, we are concerned with the name tagging process. Name tagging is implemented by using lexical, contextual and morphological tagging models. In our work, we use lexical and contextual model to recognize named entities similarly, and the details of how they are exploited are explained in Chapter 4. However, using the combination of lexical, contextual and morphological models separates this study from our work. The system was trained by using 492,821 words of newspaper articles and results are presented in Figure 2-6. The figure shows that morphological model increases the performance of NER system slightly; therefore, we also use morphological model in our work.

Model	Text (%)	Type (%)	F-Measure (%)
Lexical	80.87	91.15	86.01
Morphological	36.52	79.73	58.12
Lexical+Contextual	86.00	91.72	88.86
Lexical+Contextual+Morphological	87.12	92.20	89.66
Lexical+Contextual+Tag	89.54	92.13	90.84
Lexical+Contextual+Morphological+Tag	**90.40**	**92.73**	**91.56**

Figure 2-6: Accuracy of the name tagging task using lexical, contextual, morphological and tag models.

The exemplary studies having probabilistic (the supervised machine learning) and symbolic approach (the rule based) require tagged corpora, dictionaries and list of entities like person names. On the other hand, these requirements can be avoided by using local grammar (LG) approach [24]. The purpose of LG approach is to understand the behavior of words in a specific context and infer the patterns in their usage [24]. According to Traboulsi, reporting verbs (RV) such as *said*, *told* and *added* are good indicators of person names in financial news texts [25]; therefore, Bayraktar et al. exploit this feature to recognize person names. The work consists of two important parts; the first is extracting most significant RVs in Turkish texts and the second is the evaluation of LG approach for Turkish texts. The authors processed Turkish texts, EC2000[3] and METU TC [26], containing totally 9.154.458 words by using Nooj software[4] and text analysis tools, and they determined top 5 most frequent RVs.

[3] http://www.aa.com.tr/
[4] http://www.nooj4nlp.net/pages/nooj.html

Table 2-1: Top 5 most frequent RVs found in the EC2000 (The numbers are normalized).

RVs	FEC2000	METU TC
Belirtmek	2.261,44	1.181,53
Demek	2.103,35	4.876,45
Söylemek	1.756,68	2.252,95
Kaydetmek	1.647,81	488,07
Bildirmek	1.500,74	163,04

After extracting RVs, Bayraktar et al. examined different financial texts and generate four patterns (Figure 2-7) used for extracting person names in financial news texts.

- **Pattern 1;**
  [Title]? + [W*]? + [PN] + [W*]? + [, | (<E>ise) | (<E>de) | (<E>da)] + [W*] + [RVF1] + [.]
- **Pattern 2;**
  [RVF2] + [Title]? + [PN] + [,|(<E>ise)|(<E>de)|(<E>da)] + [W*]
- **Pattern 3;**
  [Title] + [PN] + [, | (<E>ise) | (<E>de) | (<E>da)] + [W*] + [RVF3]
- **Pattern 4;**
  [W*]? + [Title]? + [PN] + [, | (<E>ise) | (<E>de) | (<E>da)] + [W*]? + [,] + [RVF4] + [:]

Figure 2-7: Extracted Patterns according to RVs

The experiment is carried out by using Turkish news having 41.322 tokens and compared with Capitalization Rule (CR) which tries to find all capitalized words [24]. The experimental results of this work are presented at Table 2-2. It can be said that LG approach is successful to extract person names in a special context; nonetheless, the authors did not employ this approach for other entity types and different text types.

Table 2-2: Experimental results for the LG approach with C.R.

	Recall	Precision	F-measure
LG Approach with CR	86,91	78,13	81,97
Only CR	99,37	6,66	12,48

Küçük and Yazıcı developed NER system by utilizing rule based approach on Turkish texts [21]. The information sources they used to recognize named entities are lexical and pattern based resources (Figure 2-8). While determining candidate entities, they also exploit a morphological analyzer for noun inflections, in case named entities are inflected. We have also dealt with inflected named entities similarly in our NER system.

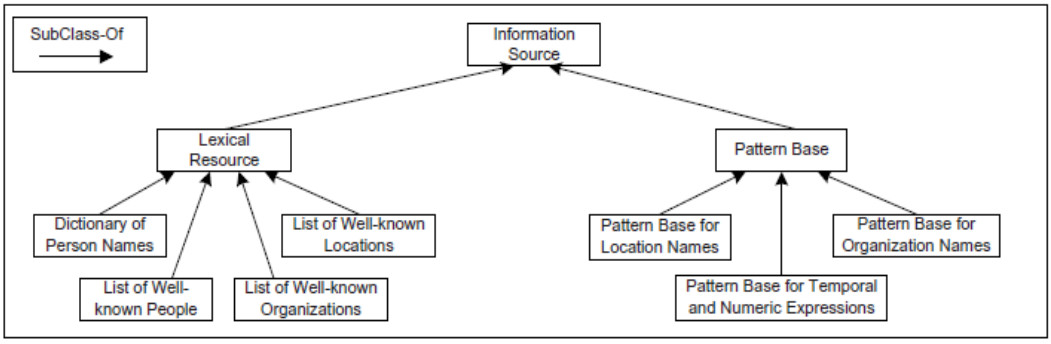

Figure 2-8: The Taxonomy of Information Sources [21]**.**

The lexical resource is comprised of person names, well-known locations, well-known people and well-known organizations lists as it is seen at Figure 2-8. Besides these lists, there are pattern based sources designed for location, organization and temporal/numerical expressions. The authors described patterns by analyzing large Turkish corpora and created rules with respect to extracted patterns. For instance, 'X Üniversitesi' (*'X University'*) is a pattern for tagging an organization name [21]. Similarly, location name patterns exploit almost permanent names such as street, road, bridge, river, mountain etc. as well as numerical and temporal expression patterns. Entity tagging is performed by first annotating the matching phrases with the help of

lexical resources, then employing rule patterns on the resulting text to recognize previously un-annotated entities.

Since, porting rule based systems to other domains decrease the performance, they enhanced their rule based recognizer and implemented a hybrid recognizer [27] that learns from available annotated data through the agency of rote learning [28]. Rote learner calculates confidence value of each entity with respect to the input genre. For instance, suppose that entity *Pınar,* which can be a person, organization or a common noun, occurs in an annotated text ten times. Assume that it is tagged as a person name in six of these cases and tagged as an organization in three of cases. Thus, confidence values of word are 0.6 and 0.3 for persons and organizations respectively. Rote learning component recognizes *Pınar* as a person name rather than organization name, since its organization confidence value is under the threshold value 0.5. The authors emphasized that the employed rote learning procedure improves the performance of rule based system and tested the system four different data sets. The results which are presented in Figure 2-9 were found satisfactory.

Data set	Precision (%)	Recall (%)	F-measure (%)
News text data set	93.38	87.14	90.13
Financial news text data set	83.56	71.94	76.80
Child stories data set	89.72	95.45	92.47
Historical text data set	82.15	79.53	80.66

Figure 2-9: Ten-fold cross validation results of the hybrid named entity recognizer (capitalization feature is turned on) [27]

Another study on Turkish NER literature is given in [29] where the aim is to implement an automatic rule learning system exploiting morphological features for NER in Turkish. Çiçekli et al. also utilize supervised learning strategy to learn rules automatically from the annotated data and employ rule filtering and rule refinement to increase the accuracy performance. While learning rules , which is the crux of the work, the concept of specific generalization of strings as described in [30] is exploited to generalize the rules. The authors described eight features for each token; token,

morphological tag, low-level gazetteer set, high-level gazetteer set, case tag, length class, token length and type class. Generalized rules are created by processing differences and similarities of these features [29]. After that, with the help of the generated rules NER is carried out. Experimental results of the work [29], presented in Figure 2-10, are better than most of the previous studies and almost same with [23].

NE category	Precision (%)	Recall (%)	F-score (%)
Person	96.69	92.08	94.33
Location	90.20	89.86	90.03
Organization	87.36	88.01	87.68
Date	97.69	95.34	96.50
Time	93.12	91.00	92.05
Overall	91.74	90.43	91.08

Figure 2-10: Quantitative performance results of the system

2.4. Disambiguation Techniques

Word sense disambiguation (WSD) is a problem of identifying which meaning of a word is used in a sentence, when the word has multiple meanings [31]. For example, the meaning of word *Batman*, which is a movie series and also a city in turkey, may differ with respect to the context it is used.

Research about WSD systems has achieved sufficiently high levels of accuracy on a variety of word types especially for English language. Many techniques have been used including lexical based, supervised machine learning and unsupervised learning methods in order to carry out the disambiguation process. Although supervised learning methods have been the most successful among these techniques, dictionary and knowledge (Lexical) methods are also used if there are no annotated data.

Bunescu and Paşca in their study [31], perform named entity disambiguation by using encyclopedic knowledge. They used Wikipedia as a lexical resource and created a dictionary from ambiguity pages. To illustrate, for a named entity *John Williams*, three

definitions corresponding to Wikipedia titles *John Williams (Wrestler), John Williams (Composer)* and *John Williams (VC)* are added to dictionary. In order to determine the real sense of a named entity in a query, they measured the similarity between query and corresponding article content in dictionary. Cosine similarity, commonly used when comparing two documents or texts, is also used in this study while determining the real meaning of the named entity. They represent each entity definition or query as a vector space model, where each component corresponds to a term in the vocabulary. Each term in vector model refer the term weight which is the standard tf-idf score [6]. Considering each entity description or query as a document, tf-idf scores and weights of the terms are calculated in the following way:

$Term\ frequency\ (tf) =\ the\ frequency\ of\ a\ term\ in\ a\ document$

$Inverse\ term\ frequency =\ measure\ of\ how\ much\ information\ the\ word\ provides$

$N = total\ number\ of\ document$

$df = the\ number\ of\ times\ that\ term\ t\ occurs\ in\ document\ d$

$w = the\ weight\ of\ term\ that\ is\ wanted\ to\ be\ calculated$

$idf = \log\frac{N}{df}$

$w = tf * idf$

Since we also exploit Wikipedia articles for information retrieval and disambiguation, we utilize Bunescu and Paşca's method with some modifications in our system. Similar to their method, we use cosine similarity which is a measure of similarity between two vectors of an inner product space [6].

CHAPTER 3

TV CONTENT AUGMENTATION SYSTEM

This chapter presents the conceptual description of a TV content augmentation system using information extraction and retrieval techniques. Before elaborating the proposed system, we first list the ideal system requirements for a TV content augmentation system. After that, the system architecture and the components of the system are explained in detail.

3.1. Ideal System Requirements

There are four requirements for an ideal TV content augmentation system considering user habits and the nature of TV viewing:

Flexibility: A perpetual augmentation system running on TV is not a preferable practice for users. As it is mentioned in section 1.1, viewers can change their modes from relaxing mode to reflective mode or vice versa. Therefore the users should be given the opportunity to alternate their modes. Furthermore, it should be taken into consideration that viewers usually use TV as an entertainment and relaxing device.

Sufficiency: In the television context, augmentation should be performed by providing additional related information about the content being viewed. When users are in reflective mode, they may want to learn additional information about the program content. Supplementary information should be adequate and related to the content. The viewer should understand the topic by observing the relevant data without looking elsewhere.

Simplicity: Presenting augmented data is the critical part of such an ideal system. Since intrusive data on TV screen may annoy viewers, relevant data should be summarized and demonstrated properly. Moreover, the designed graphical interface on TV must be user-friendly.

Timeliness: All the requirements described so far should be met in real time during TV watching activity. It means that when a viewer wants to learn something about the content, it should be provided at that exact moment.

We have designed a TV content augmentation system by considering these requirements. The details of the proposed system architecture are explained next section.

3.2. Proposed System Architecture

The proposed system generates and presents relevant data about current content on TV so that viewers are aware of the content in detail and specific information about the program content. The major contribution of the proposed system is using textual description of a TV broadcast program and providing additional data in real time. General structure of this system is shown in Figure 3-1.

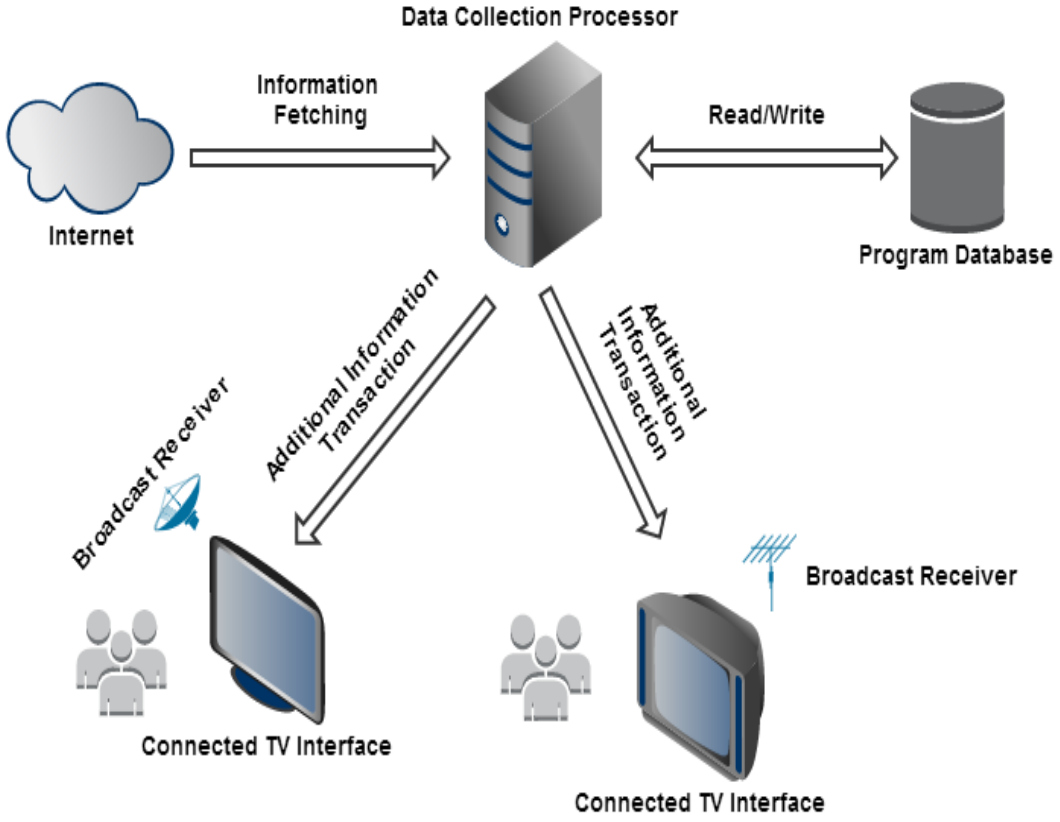

Figure 3-1: General Structure of the Proposed System

The proposed system consists of two main parts which are the server side and the client side. Server side is the part that data gathering, database and web service operations are performed. Server communications and mostly presentation/user interface operations are carried out at the client side. Overall architecture of the system is shown in Figure 3-2. These two main components with their sub-components are illustrated in Figure 3-3.

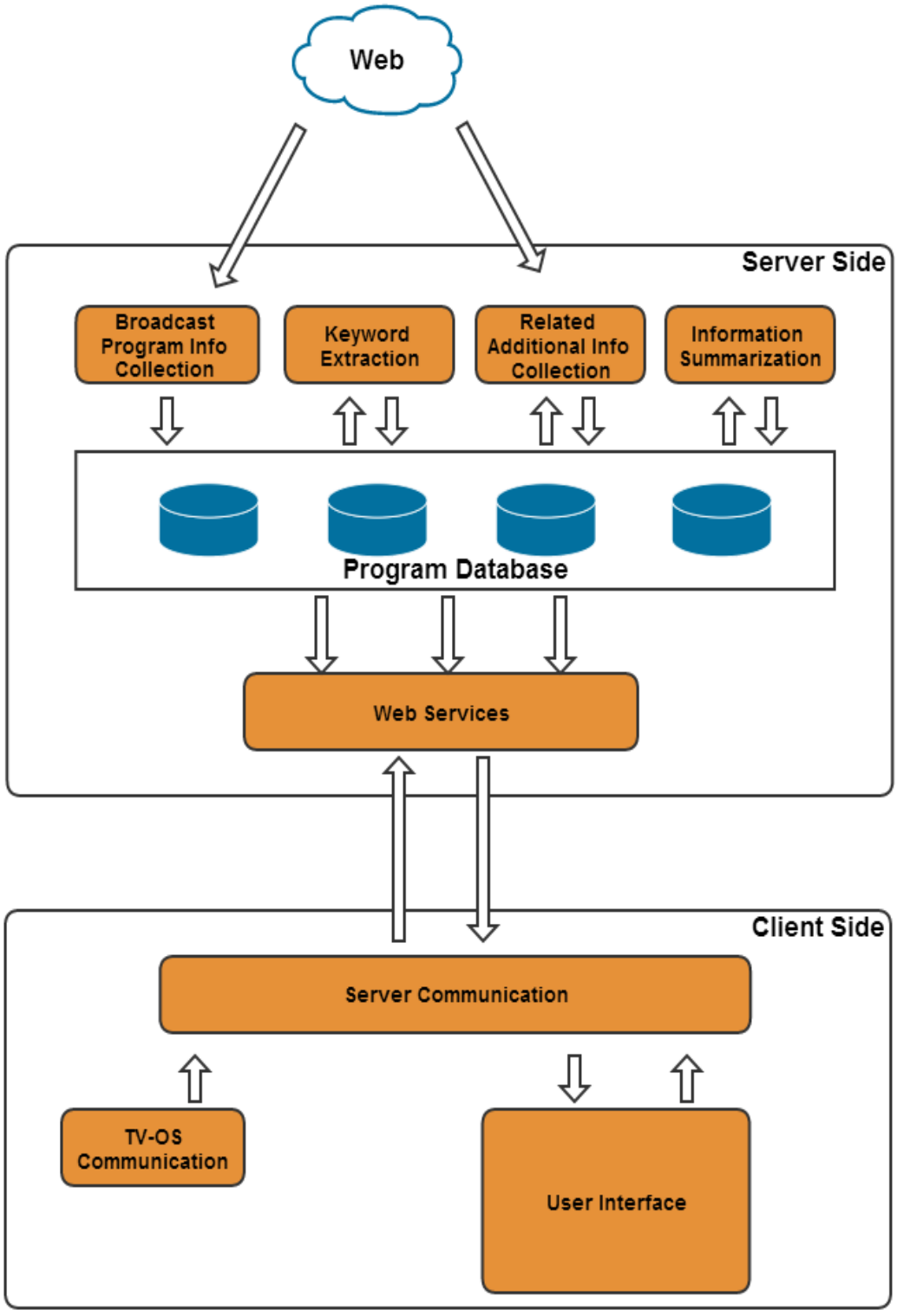

Figure 3-2: Architecture of the Proposed System

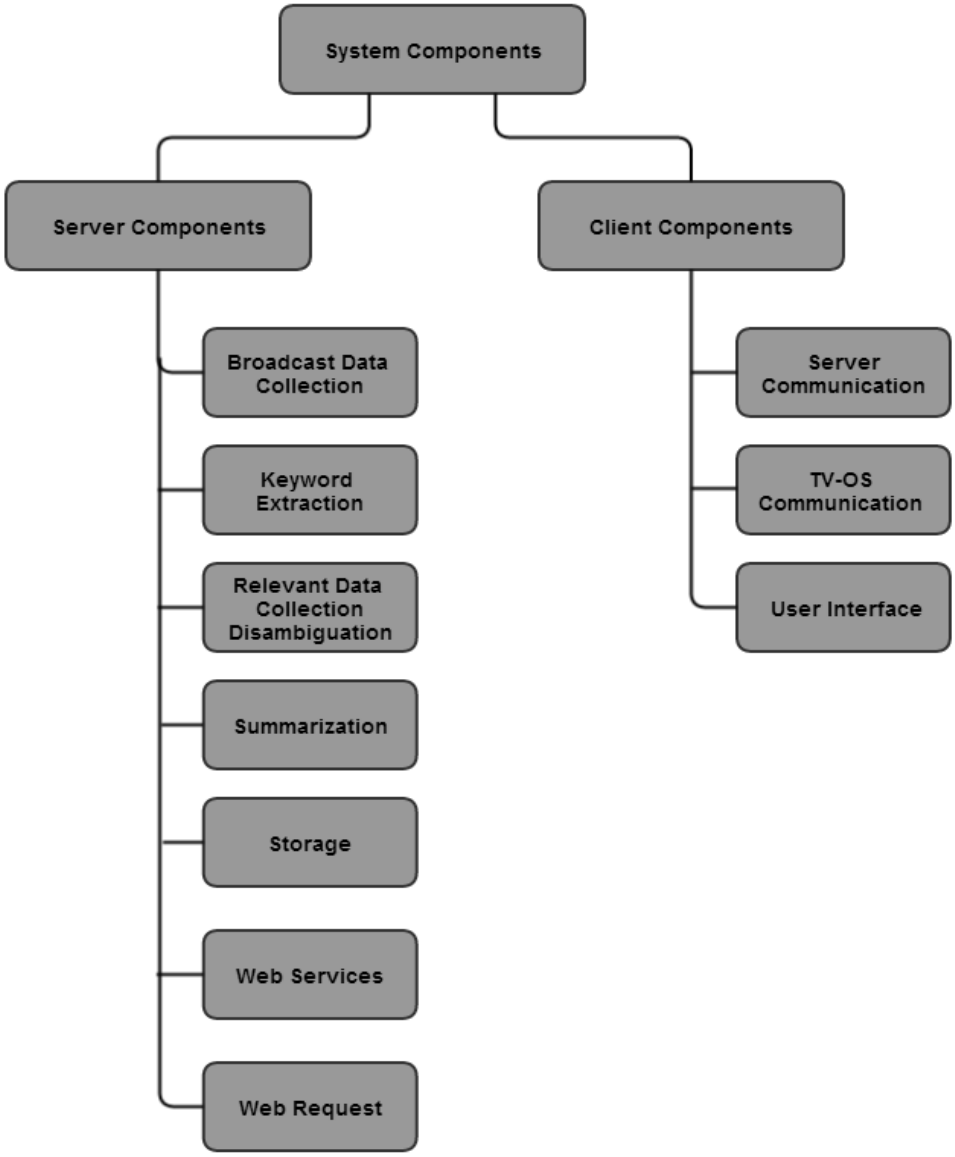

Figure 3-3: Components of the Proposed System

3.2.1. Server Side

There are seven modules of the server component: broadcast data collection, keyword extraction, relevant data collection and disambiguation, summarization, storage, web services and web request module.

3.2.1.1. Broadcast Data Collection Module

The schedule of TV programs are described and published as EPG by broadcasters twenty four hours in advance. Some websites and media companies put these documents on their web page. Gathering and managing these documents are performed by broadcast data collection module. Firstly, web sites or company web services if exist are defined for this module. Then, it runs on server by utilizing predefined web sites every midnight and gathers the next day's broadcast information. Database objects which will also be used in the other phases are created in this module by setting their program description field. This module completes its job by saving program objects into the database.

3.2.1.2. Keyword Extraction Module

Keyword extraction module implements specific named entity recognition (NER) algorithms to determine named entities. To increase NER performance, it utilizes some databases such as DBPedia [32] and tools like DBPedia Spotlight [33]. This module processes each program description previously saved into database. With the help of NER algorithm, named entities are determined and their types are specified. Types and sub-types of entities are significant for the relevant data collection and disambiguation module. As a last operation in this module, named entities with their types extracted from each description are set into program objects as keywords of that program.

3.2.1.3. Relevant Data Collection & Disambiguation Module

This module consists of two important sub-modules; relevant data collection and disambiguation.

Relevant data collection part searches extracted keywords on the Internet similar to broadcast data collection module. This module reads every keyword in a program object and requests their detailed information from predefined web sites. To decrease

the number of redundant requests; if the keyword's detailed information already exists in the database, it is obtained from there. Raw data of a keyword with detailed description and related terms are saved into the database.

Sometimes a word has more than one meaning such as *Batman*. *Batman* is a movie series and also a city in Turkey. Disambiguation sub-module ensures that any ambiguity is removed before saving definition of the keyword. This module first determines whether the keyword is ambiguous or not by exploiting responses of the web sites, database records and entity types. If there is an ambiguity, relevant data collection module fetches all possible descriptions of the keyword. Program description is compared with each of the possible descriptions and the most similar one with respect to context is selected as the real description of the keyword by the disambiguation module. This module prevents presenting the irrelevant data to the viewers as related information.

3.2.1.4. Summarization Module

Summarization module is responsible for summarizing the raw data of each keyword and constructing a template including the most significant features of the keyword. This module identifies the features that can be highlighted according to the entity types. For instance, birth date is a meaningful property for a person name, population property is significant for a location/city. The created templates can be presented as an info box like Wikipedia infobox[5] as seen in Figure 3-4. Furthermore, the module extracts a short definition for summary information from raw data. All these created summaries are saved into the database for future usage.

[5] http://en.wikipedia.org/wiki/Infobox

Figure 3-4: METU Infobox retrieved from Wikipedia

3.2.1.5. Storage Module

Storage module provides the capability to store downloaded program information, extracted keywords and related data acquired by the collection and extraction modules. The communication between the components of the system is carried out by the storage module. The collection and extraction modules access the database for both reading and updating, on the other hand web services module access the database to read the enhanced data and transport it to the client side.

3.2.1.6. Web Services Module

Web services module provides services and functions for the communication between the server side and the client side. The client application invokes methods defined in

this module and gets augmented program information features which are description, keywords, definition of keywords and summarized data.

3.2.1.7. Web Request Module

Web request module makes sure that the augmentation system fetches and downloads the requested information by utilizing some web site urls. The collection modules invoke this module, which is a layer between the server application and the Internet world, to perform a downloading operation.

3.2.2. Client Side

The client side is actually an application which runs on the connected TV. There are three modules of the client component: server communication, TV-native OS communication and user interface module.

3.2.2.1. Server Communication Module

The server communication module provides the capability to call methods defined in server side as a web service. These methods are mentioned at section 3.2.1.6.

3.2.2.2. TV-Native OS Communication Module

This module provides the communication between the client application and TV operating system. The client application needs to get some information from TV. In order to request program related data by exploiting the server communication module, TV time and information of the program which is currently playing on TV, are needed. This module conveys this acquired data from TV-OS to the server communication module.

3.2.2.3. User Interface Module

The user interface module is the most important part of the client side. TV viewers communicate with the augmentation system via this module and initiate their requests by using action items on the interface. The detailed information, relevant data and keyword descriptions can be seen through this user interface. The user interface also gives users the capability to choose the application mode including *overlay* and *minimized*. In the *Overlay* mode, the interface covers a small part of the screen and the user can watch the program at the same time. If users do not want to see the interface on the screen continuously, they may switch to *minimized* mode in which the interface appears only when a notification occurs.

CHAPTER 4

PROTOTYPE IMPLEMENTATION

Following the conceptual description of the proposed system, the prototype of the TV content augmentation system is implemented using rule based named entity recognition techniques and information retrieval methods. We describe the details of the prototype system in this chapter.

4.1. Description of the Prototype Implementation

In order to show the feasibility and applicability of the proposed system, a working system is constructed according to its conceptual design except the TV interface part. The main differences of the implementation from the proposed system are user interface and server-client communication. Server side and client side are integrated in the prototype system and it is implemented as a java application running on PC. The prototype mainly consists of TV program data collection module, Turkish NER module for keyword extraction, information retrieval module including semantic disambiguation and summarization, and a java SWT desktop application. The main issues in the implementation have been developing a NER algorithm for Turkish texts and constructing a TV content augmentation application for Turkish TV programs by using the implemented NER algorithm.

In the prototype, TV program guide information is retrieved from Radikal TV Guide web page[6] which publishes program schedule of TV channels daily. Program title, actors, director, writer, short and long description are extracted for each program.

In our implementation, keyword extraction module utilizes two instruments for Turkish named entity recognition; DBpedia Spotlight [33] and our Turkish NER algorithm. The extracted keywords and features of the program such as actor and director are passed to the information retrieval module to fetch detailed description and related data about the program. Information retrieval module offers three functionalities; fetching detailed description, disambiguation and summarization of the relevant data. Descriptions of keywords are retrieved from Wikipedia web site [34]. Disambiguation is performed by using tf-idf and cosine similarity techniques [6].

The prototype system runs data collection, keyword extraction and relevant data retrieval operations every night and database is prepared for the user requests next day. A java SWT desktop application is implemented for user interface operations. The user can request detailed information about a TV program and view augmented data via this application.

Since our main focus is to develop a Turkish NER algorithm, we have implemented a NER application for Turkish texts: Augmentext. This application exploits the system components described above and offers a user interface to enter texts and determine the entities in the given text. Augmentext retrieves the detailed description of the extracted keywords from Wikipedia and presents them to the user.

4.2. Exploited Tools

The software tools and libraries while implementing the prototype system are listed in this section.

[6] http://www.radikal.com.tr/tvrehberi/

4.2.1. DBPedia Spotlight

DBPedia is a project which aims to extract structured content from the Wikipedia and make this information accessible on the web [32]. It is published in 2007 by the people at Free University of Berlin and University of Leipzig in collaboration with OpenLink Software. Afterwards, the researchers at the Free University of Berlin constructed a tool DBPedia Spotlight [35] that annotates mentions of DBPedia resources in text. In other words, if a DBPedia resource, which is a page title extracted from Wikipedia, is mentioned in a text; DBpedia Spotlight can recognize it as a named entity. The main facility of Spotlight is entity extraction including entity detection and disambiguation. This tool can be used for named entity recognition tasks and customized for many use cases. There are different ways for exploiting DBPedia Spotlight. Users can utilize its methods by using web services, ubuntu/debian package, jar and war files. It can also be built with maven and run by using scala plugin. Spotlight also provides a large amount of data set for the purpose of entity recognition. Lucene and Statistical are two facilities for using these data sets. Moreover, Spotlight can also be used with different datasets and other texts in custom domains.

Debian package installation, is used as a web service by using local host address with the specified port number. Spotting service of this tool is utilized for recognizing named entities. While using web service, it is made a request by appending text and spotter implementation technique to the end point url as a parameter. An example call and output is shown as:

Example call:

http:// localhost:2222/rest/spot/?text=Berlin bir başkenttir&spotter=Default

Example output in json format:

{"annotation":{"@text":"Berlin bir
başkenttir","surfaceForm":{"@name":"Berlin","@offset":"0"}}}

4.2.2. Zemberek

Zemberek is an open-source natural language processing framework which is developed for Turkish language. The framework can perform basic NLP operations including spell checking, morphological parsing, stemming, word construction and suggestion, converting words from ASCII characters and extracting syllables [36]. Since it is a java based framework, it can be simply exploited as a library without dependency of any platform. A class called Zemberek in the library is used for high level operations such as morphological parsing. For instance, in order to obtain word forms of a Turkish word, the following sample code is enough:

```
Kelime[]wordForms =
zemberek.kelimeCozumle(WORD,COZUMLEME_SEVIYESI_CONSTANT);
```

Zemberek is also used in real world applications such as OpenOffice.org and Turkish Linux Distribution Pardus.

4.2.3. MediaWiki

MediaWiki is a web service API providing high-level access to Wikipedia pages and features [37]. In order to get and post data by making HTTP requests to the web service, client services must log in to Wiki. The endpoint url of API for English Wikipedia is given as:

http://en.wikipedia.org/w/api.php

Extra parameters can be added to this url and the content of response can be designated depending on users' needs. For instance, when the format parameter is given as json, the response is obtained in json format. A simple example for getting main page of Wikipedia in json format is given as:

http://en.wikipedia.org/w/api.php?format=json&action=query&titles=Main%20Page&prop=revisions&rvprop=content

4.3. Database Design Issues

MySQL relational database management system is used during the development of our prototype implementation. In our database schema, there are two main tables which are *Programs* and *Keywords*. There is also *Actors* table which is used to keep actors of a TV program.

The structures of these tables and junction table keeping the relationships between two tables are described in this section.

4.3.1. Program Table

Program table is used to store TV program information and manage program related operations. Programs are created by TV program collection module and updated by keyword extraction and information retrieval module. The attributes of this table are also exploited during the presentation of the system. The structure of the table is shown at Table 4-1.

Table 4-1: Structure of Program Table

Attributes	Summary
Id	Id attribute is used to ensure uniqueness of program object.
timeOfProgram	The attribute is used to keep time of the TV program.
Genre	Type of a program such as *adventure* and *comedy* is stored by using this attribute.
channelName	The name of the channel which program is broadcasted.
shortDescription	Summary of program description is kept by this attribute.
longDescription	Detailed description of TV program.

4.3.2. Keyword Table

Keyword table is used to store extracted named entities with their features and manage keyword related operations. Keywords are created by the keyword extraction module and updated by the information retrieval module. This table is utilized by our

Augmentext application besides prototype implementation. The structure of the table is presented at Table 4-2.

Table 4-2: Structure of Keyword Table

Attributes	Summary
content	Content of the keyword with affix and punctuation.
clearedContent	Content of the keyword without affix and punctuation.
type	This attribute keeps type of the keyword such as *location* or *person*.
subType	subtype represents the specific type of the keyword such as *city*.
indexInSentence	Starting index of the keyword content in the program description (long).
detailDescription	Summary description of the keyword retrieved from Wikipedia page.
infobox	The information box including most important features of the keyword.

4.3.3. Junction Tables' Structures

There are two important junction tables in our prototype implementation, which are program_actor and program_keyword tables. These tables are used for joining two different tables to create many-to-many relations. Structures of these tables are shown at Table 4-3. Actor table is used only if actors of a program are defined in the program description.

Table 4-3: Structure of Junction Tables

Table Name	Fields in Table	Summary
program_actor	program_id, actor_name	This table is used for joining program and actor tables. A program may have many actors; similarly, an actor may appear in more than one program.
program_keyword	program_id, keyword_cleared_content	This table is used for joining program and keyword tables. The characteristic of this table is same as program_actor table.

4.4. Components of the Prototype

The prototype system has four implementation components in accordance with the proposed system architecture. These implementation components are Radikal TV guide collection, Turkish NER for keyword extraction, Wikipedia Information retrieval and desktop applications.

4.4.1. Radikal TV Guide Collection

In our implementation, EPG text of TV program is used as a resource for augmenting the program content. In order to get proper EPG information we have used Radikal website, which is a news site publishing daily news in Turkish[7]. It periodically publishes three days program schedule of thirty nine channels in TV guide page of the web site. An example of channel program is illustrated in Figure 4-1.

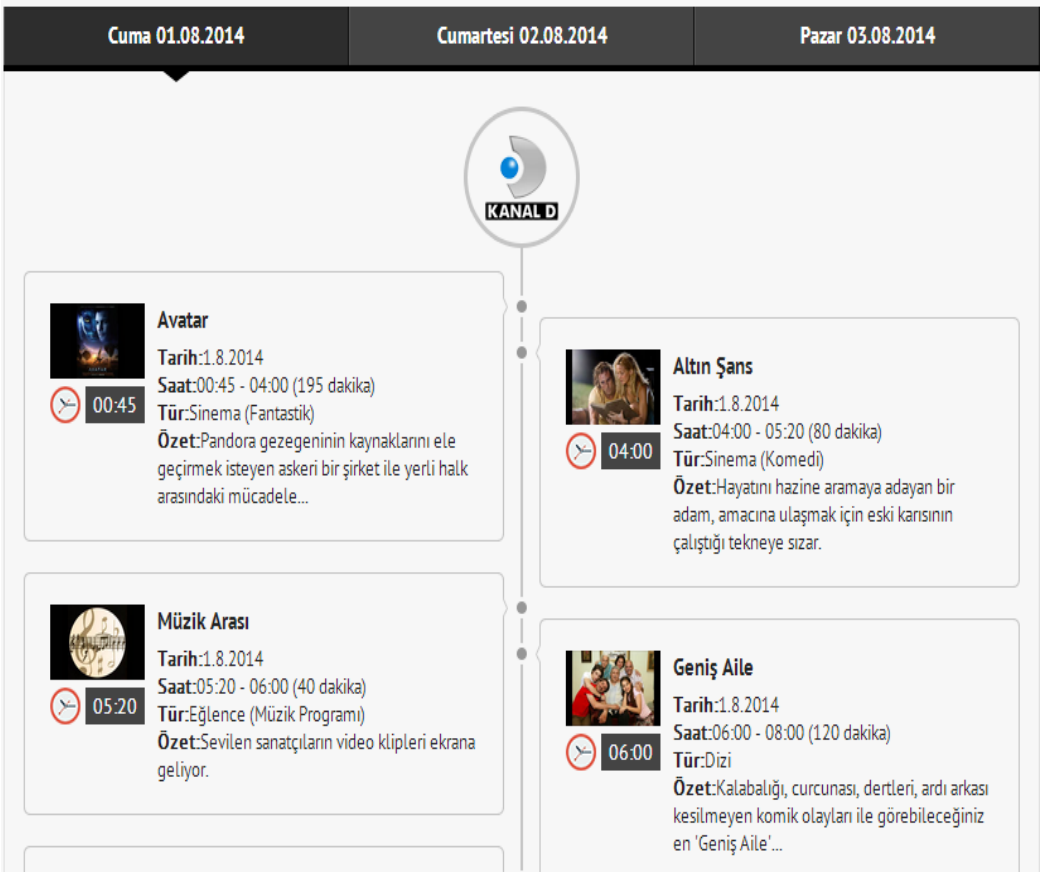

Figure 4-1: The program schedule of channel KanalD

The details of each program are given in the corresponding page and presented as illustrated in the Figure 4-2. As it is seen, a program is presented with some features

[7] http://www.radikal.com.tr/tvrehberi/

like the channel name, day and time of the program, type, director, cast information, summary and long description.

We extracted channel names manually and saved into a configuration file. By using these channel names, we defined the url of each channel to get the program schedule. For example, in order to fetch KanalD program schedule as an html file, following url is used:

http://www.radikal.com.tr/tvrehberi/kanald/

We have exploited a java html parser library, jsoup [38], to parse the retrieved html page and to extract the necessary information. Textual data in html page is retrieved by using DOM traversals and CSS selectors. After that, we process textual data considering some characteristic features of texts to get the desired result. For instance, each actor name is extracted by splitting *cast* text based on commas.

In this way, each program of a channel is retrieved with its features. Program objects are created with the extracted features and saved into the database. Yet, some attributes of a program object are not created in this part.

Figure 4-2: Detailed description of a TV program

4.4.2. Turkish NER

Program descriptions in Turkish are processed to recognize named entities. This operation is the challenging and most important part of our prototype system. DBPedia Spotlight tool processes the descriptions and surface forms in the result are extracted as a name entity before our NER implementation runs.

Since our prototype system works with Turkish TV programs and text documents, we have implemented our NER algorithm for Turkish language. Before determining the NER method we would use, we analyzed the texts in EPG data of TV programs and defined types of entities that we would recognize. After this analysis, we set our goal to recognize person names, organizations, locations including cities and countries, and abbreviations. Due to the limited number of entity types to be recognized, we have chosen rule based approach among other NER methods reviewed in chapter 2. Moreover, a rule based system does not need an annotated corpus for training, which requires considerable time and effort.

Our approach is similar to Küçük's method [21] which uses both lexical and pattern base resources with the help of a morphological analyzer. Similarly, we have defined lexical and pattern base resources in order to tag named entities. Lexical resources consist of entity type dictionaries: person names, cities and countries. Unlike other rule based systems, the number of dictionaries in our system is few. Thus, we have tried to recognize named entities by using pattern base resources. Pattern base resources include organization and location patterns which are determined by analyzing Turkish texts. The classification of these resources is presented in Figure 4-3.

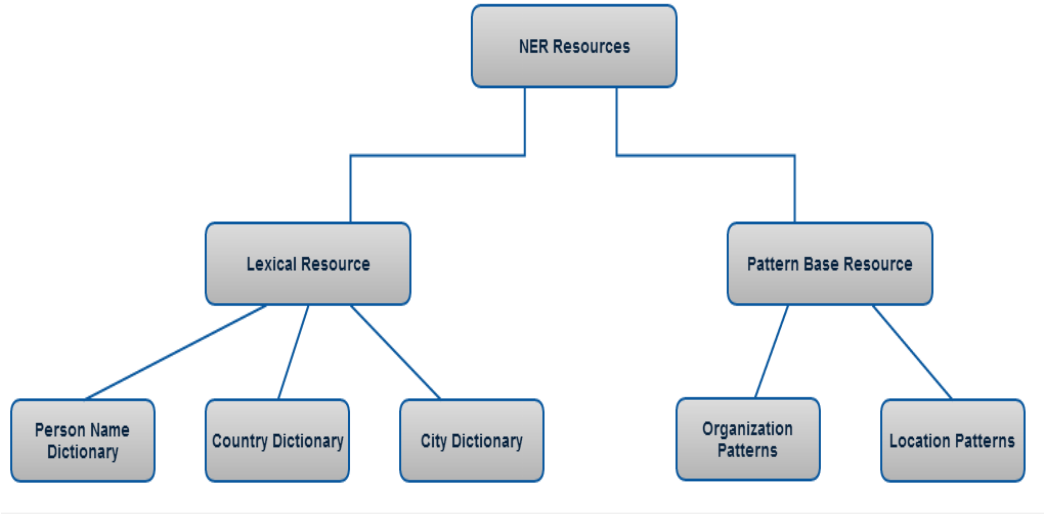

Figure 4-3: Classification of Resources in Our NER Approach

1. *Person Name Dictionary:* We have gathered about 10000 person names in Turkish from different sources including a blog page listing Turkish person names [39].
2. *Country Dictionary:* In this resource, Turkish names of all countries are included such as *Çek Cumhuriyeti* (*'Czech Republic'*) or *Faroe Adaları* (*'Faroe Islands'*).
3. *City Dictionary:* This resource encompasses the names of cities and towns in Turkey such as *Adıyaman* or *Gölbaşı*.
4. *Organization Patterns:* This resource consists of several patterns which are used for the extraction of organization names such as university, sport

organization or a foundation. There are more than ten patterns in this resource and the following patterns are examples where X represents the entities before the specified word in each pattern:

X Üniversitesi/ Derneği/ Hastanesi/ Takımı/ Ligi/…
X University/ Association/ Hospital/ Team/ League/…

5. *Location Patterns:* Similar to organization patterns, this resource consists of several patterns which are used for the extraction of location names such as road, street or geographic place name. Example patterns in this resource are listed as follows:

   X Meydanı/ Caddesi/ Köyü/ Yolu/…
   X Square/ Street/ Town/ Road/…

We have developed our algorithm by considering the resources and determined its steps as follows:

1. Determine the boundary of sentences in a given text by using sentence boundary detection method of Zemberek [36].
2. Extract each word as a token from sentences and put it to the word list of the corresponding sentence.
3. Run abbreviation, location, organization and person name entity finder on word lists of each sentence respectively.
4. If a word or word group is tagged as an entity, add it into the annotated word list.

An entity finder basically takes a word list as input and gives named entities as output. There are four types of entity finders:

Abbreviation Entity Finder: This entity finder simply processes each word by checking whether all letters of them are capitalized or not, after affixes are discarded. If all letters of a word are capitalized, then this word is tagged as an abbreviation. If there are dots between letters such as *S.W.A.T*, this is also considered as an abbreviation.

Location Entity Finder: Location named entity finder first takes entities from location dictionaries and search them on the given sentence without addressing each word separately. If a matching case occurs, then the matched word or word group is tagged as a location entity. For instance, if a given sentence includes *Faroe Adaları* word group, this entity finder sets type of it as location and sub-type as country. After using the lexical resource, finder exploits location pattern base resource. In order to use patterns, noun inflections of the word are determined by using the morphological parser of Zemberek. Noun inflections of each word are searched on the patterns list whether there is a match or not. If there is a match, it is tagged as a possible location and previous words are analyzed iteratively until finding a word that does not begin with a capital letter. To illustrate, in a sentence such as "Akşam saatlerinde İstiklal Caddesi'ni dolduran göstericiler sloganlar atarak ilerledi", *İstiklal Caddesi* is tagged as a location. The flow diagram of this approach is presented in Figure 4-4.

Organization Entity Finder: This entity finder performs the same operations as the location entity finder with the only difference that the organization entity finder does not use any lexical resource to find entities. It tries to recognize named entities by using only pattern base resources which is exploited as location entity finder do. The flow diagram of this approach is also presented in Figure 4-4.

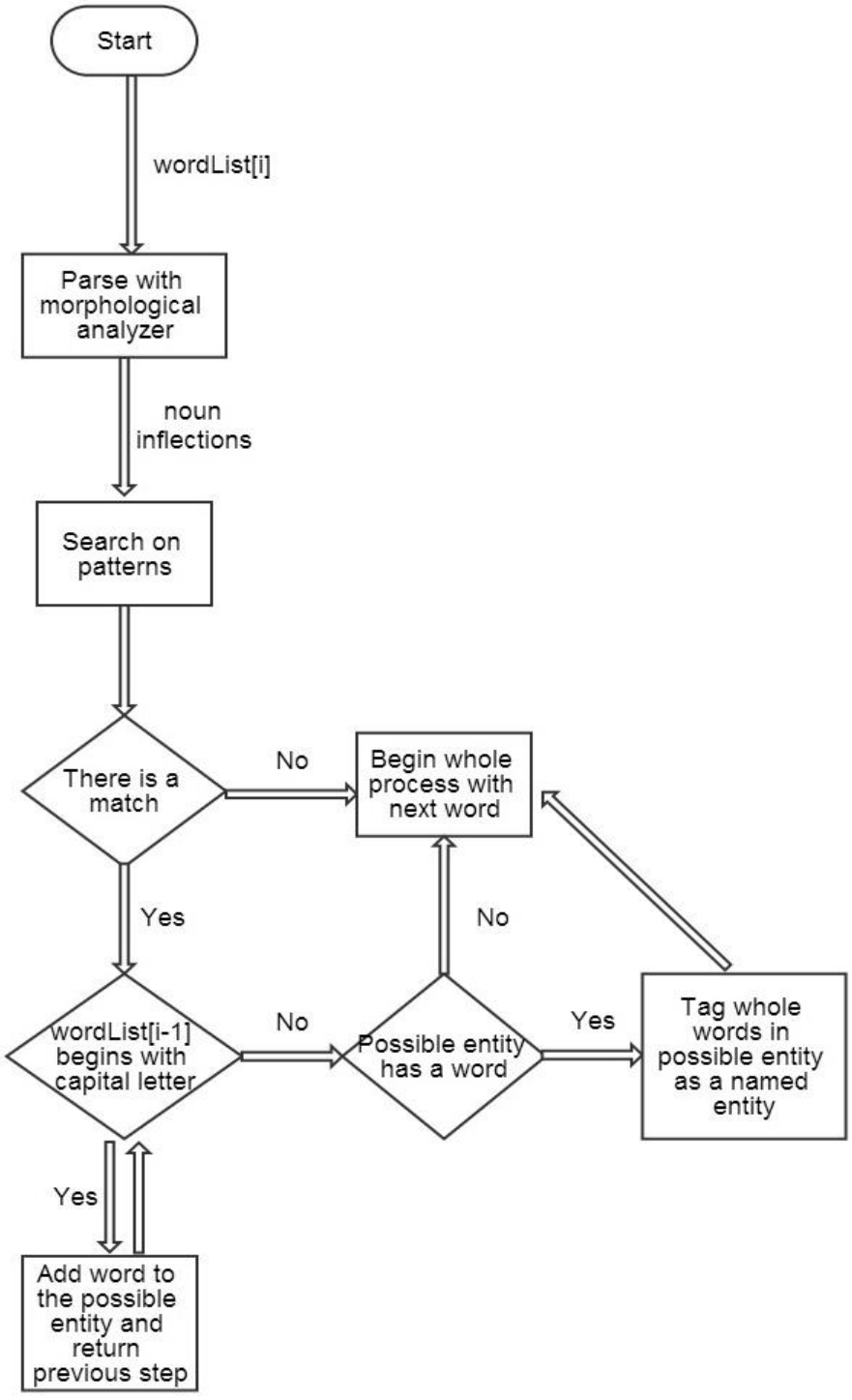

Figure 4-4: Flow Diagram of Location-Organization Tagging Operation using Pattern Base Resources

Person Name Entity Finder: Person names are recognized by utilizing only lexical resource which is a Turkish person name dictionary. The entity finder searches words

on person name list after discarding affixes. If there is a match, it is tagged as person name and next words are analyzed iteratively until finding a word that does not begin with a capital letter to recognize middle name and surname. For instance, in a sentence such as "Akşam saatlerinde bir açıklama yapan Ersin Mustafa Özbükey, çalışmaların tamamlandığını söyledi", *Ersin Mustafa Özbükey* is tagged as a person name. Note that, if a word is already tagged as an abbreviation, organization or location; it is ignored by this entity finder. The flow diagram of the whole process is presented in Figure 4-5.

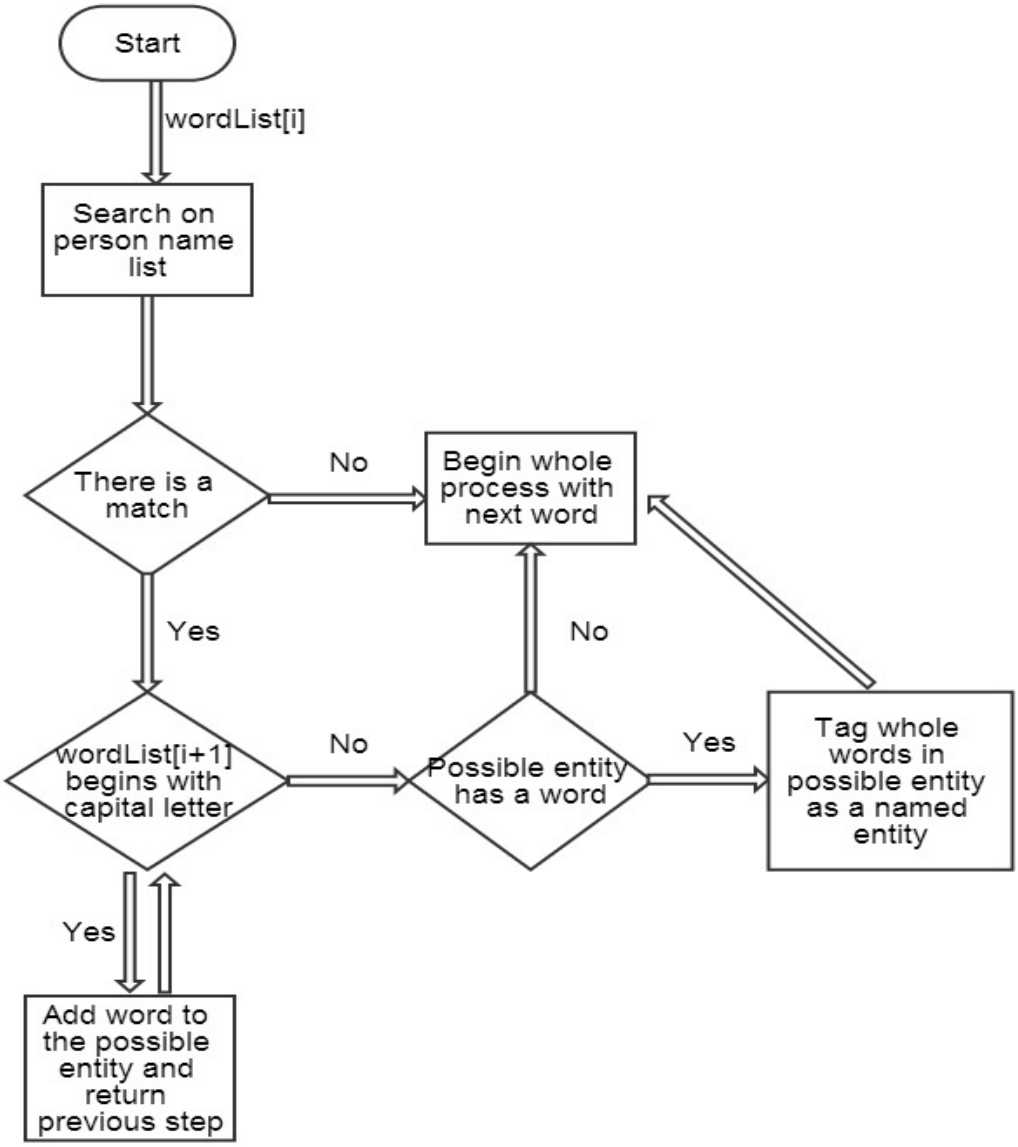

Figure 4-5: Flow Diagram of Person Name Tagging Operation Using Lexical Resource

There are some issues of our heuristic algorithm that should be highlighted. One of them is that there can be more than one entity for the same index. For example, in a sentence such as “Dün akşam Soma Kömür İşletmeleri’nde gerçekleşen kazada 302 madencimiz hayatını kaybetti.”, *Soma* and *Soma Kömür İşletmeleri* are two different named entities with the same starting index 10, where the first one is city and the latter is an organization. We added these two entities into the annotated word list as different entities. Since they have different meaning for a user, during the information retrieval part we retrieved detailed information for both. The other issue is the capitalization clue that we have utilized to extract the named entities. While searching the dictionaries, in order to increase the performance of the algorithm, we search words only if they begin with a capital letter. On the other hand, we do not use capitalization clue for pattern base resources.

4.4.3. Wikipedia Information Retrieval

The user may request some more information about an extracted named entity. Our content augmentation system provides additional information for the named entity by searching it as a keyword in Wikipedia and bringing a summarized form of the information found. The desired content augmentation is performed by three sub-tasks: Wikipedia information gathering, disambiguation and summarization process.

4.4.3.1. Wikipedia Information Gathering

Fetching data from Wikipedia page with the help of MediaWiki API service [37] is a very straightforward operation. We have called web service methods of MediaWiki API as described in section 4.2.3 by giving extracted keyword as a parameter. The result of request is obtained as a json format and may vary depending on the keyword information on the Wikipedia:

- If Wikipedia has no page for given keyword, it returns a null json object. When the result is null, we simply do nothing and cannot save detailed information of the given keyword.
- If there are different pages in Wikipedia for the same keyword, it returns all page titles with disambiguation information without their details. For instance, for *Batman* keyword, result includes titles *Batman (City)* and *Batman (Movie Series).* In this case, we carry out our disambiguation process.
- In Wikipedia, sometimes a title is referred by different words especially for abbreviations. In such cases, Wikipedia directs the request to the actual wiki page. For example; when *AFAD* keyword is requested from Wikipedia, it returns actual page title *Afet ve Acil Durum Yönetim Başkanlığı.* If the result is a redirected page, then we re-request from Wiki with the actual page title parameter.
- If a given keyword has no ambiguity or redirected page and there is exactly one page for the keyword in Wikipedia; it returns all textual data of keyword with the information box. In this case, we parse json result and extract wiki page text with sub-titles and the information box. After that, we perform our summarization process in order to create structured summary information about the given keyword.

4.4.3.2. Disambiguation of the Retrieved Data

In order to determine the actual meaning of the given keyword, we extract all possible titles from json result. We re-request and obtain all possible descriptions of the keyword. For disambiguation operation, we have exploited Bunescu and Paşca's approach. By processing all descriptions, including the text from which the keyword is extracted, we eliminate punctuations and stop words in order to calculate tf-idf scores accurately.

After calculating the tf-idf scores and weights, we construct a vector that includes weights of all terms for each document. In order to find the most similar document, we

use cosine similarity method which is a measure of similarity between two vectors of an inner product space [6]. The vector for the document from which the keyword is extracted is compared to other documents and the one with the highest similarity score is chosen as the actual meaning of the given keyword.

4.4.3.3. Summarization of the Retrieved Data

Retrieved Wikipedia data is too long and complicated to read while watching TV. Thus, we summarize Wikipedia texts so that users understand the related additional data at a glance. Usually, Wikipedia short descriptions give an idea of the topic and most of the time the first sentence is enough for defining the keyword. For the summarization, we take first one or two sentences of the short description of the keyword. If the first sentence is too long, then we only take the first sentence of the keyword description.

Moreover, infobox of the retrieved Wikipedia page may be exhaustive for learning important features of the keyword. Therefore, we reduce the number of infobox features with respect to Wiki type of the keyword such as person or city. We have determined important features for each entity type and taken only these features for infobox presentation. For example, for a person entity type; birthdate, birthplace and profession are common. Additional features are added to our infobox presentation by considering the sub-type of the keyword such as football player or politician. Since, there are more types and sub-types in Wikipedia pages than our entities; we put more emphasis on wiki types while summarizing infobox features.

4.4.4. Desktop Applications

We have implemented two desktop applications exploiting prototype system utilities; information collection, keyword extraction and information retrieval. We have used java standard widget toolkit (SWT) library while implementing these applications named TV scenario application and augmentext.

4.4.4.1. TV Scenario Application

We have tried to implement this desktop application according to conceptual description of the proposed system. However, it was not possible to carry out a complete TV augmentation application on PC. Therefore, we have made some changes on the user interface of the proposed system. Since users cannot open or switch channels on a desktop application as in TV, we provide the users with the channel and program list, gathered from Radikal TV guide. The user interface of TV scenario application has four main parts as seen at figure 4-6; channel buttons on the most left area, program combobox with representative image of selected program in the middle, keyword/actor selection list at the bottom of program image, and infobox/additional information of selected keyword/actor on the most right area. During the initialization of the application, all channels, programs, keywords and additional data retrieved from Wikipedia are prepared for user requests. Items of the user interface are described as follows:

Channel Buttons: Seven channels; ATV, Star TV, Show TV, CNN Turk, TRT1, NTV, KanalD, are listed in this area with their logos, and it can be selected by the users.

Program Combobox and Representative Image: When the user selects a channel, corresponding programs are listed in this combobox. Representative image is shown according to the selected program.

Actor/Keyword Selection List: Actors and keywords about a TV program are listed in this area.

Infobox and Additional Information Area: The field that infobox and additional summarized information is presented.

The usage of application is very straightforward. When the user selects a channel, the corresponding program list is presented to the users. If the user selects a program, actors and keywords about the selected program are shown in the actors and keywords list. The selection of each keyword or actor triggers loading of infobox and relevant additional information area.

Figure 4-6: A Screenshot of TV Augmentation Application

4.4.4.2. Augmentext Application

Augmentext is implemented for enriching any textual data rather than EPG texts by recognizing named entities in the given text and gathering related information about them. The user interface of the application consists of three main parts; text field, keywords list and detailed information area as shown at figure 4-7. The user can enter any text in the text field and enhance this textual data by clicking *Augment the World* button.

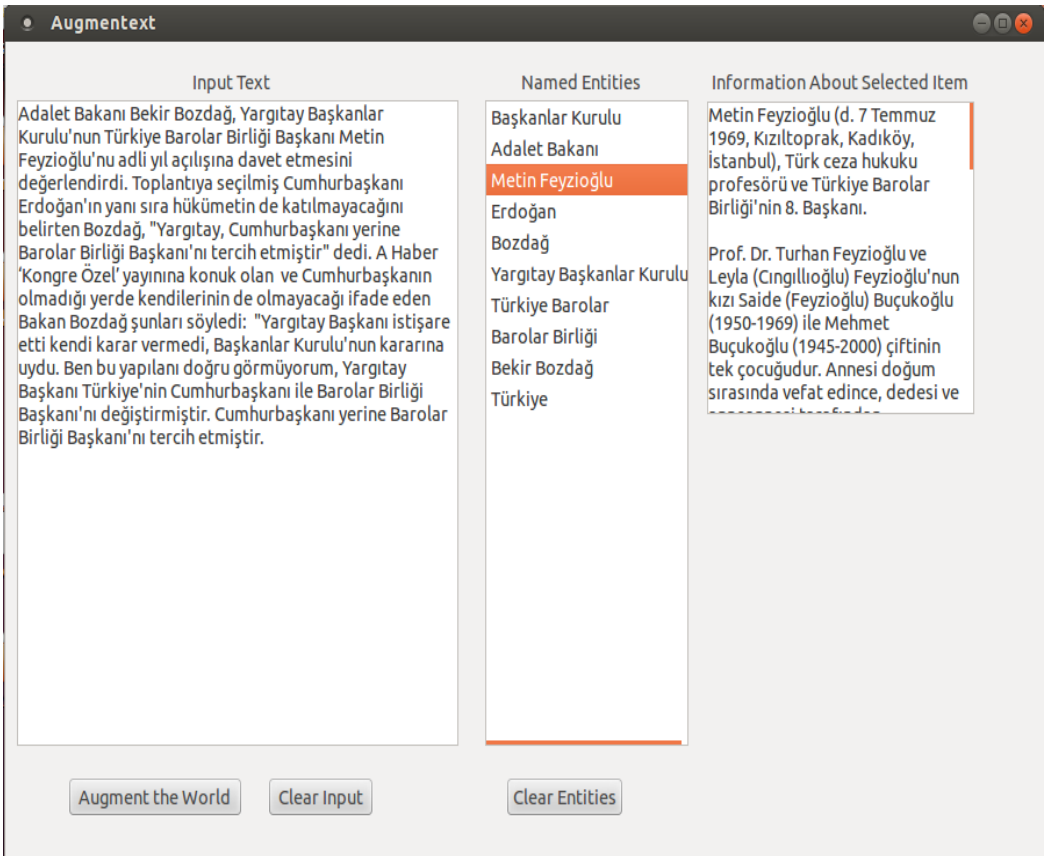

Figure 4-7: The Screenshot of Augmentext Application

CHAPTER 5

EXPERIMENTAL RESULTS

We have conducted a set of experiments in order to evaluate the performance of our NER algorithm and the behavior of the TV content augmentation system. We carried out the experiments with two different data sets. In this section, these experiments are explained and the results are discussed by comparing several other studies.

5.1. Experimental Evaluation of NER

5.1.1. Data and Methodology

As we stated, NER research on Turkish texts is known to be rare and therefore the most important drawback in NER studies for Turkish is the scarcity of publicly available annotated corpora. Thanks to the Çiçekli et al. and their study [29], we conducted our experiments on an annotated Turkish corpus. They generated the corpus by manually tagging 355 news articles on terrorism from both online and print news sources in Turkish. Since our aim is to extract person, location and organization names, we did not consider numerical and temporal expressions during the evaluation of the algorithm. Except the numerical and temporal named entities, there are 5672 NEs in this corpus; 1335 person names, 2355 location names, 1218 organization names. In order to evaluate the system performance, we measured precision, recall and F-score in two different ways according to Message Understanding Conference (MUC) evaluation metrics. The difference between measurements is the method of

determining the truth value of the matching. Variables used in formulas are given as follows:

Correct: Number of items correctly recognized by the system with correct boundaries and type.

Spurious: Number of items erroneously extracted by the system, although they are not in the actual annotated data set.

Missing: Number of items which are not annotated by the system, although they are in the annotated data set.

Partial: Number of items recognized by the system with correct type and overlapping boundaries.

In the first measurement, we did not consider partial matching and we calculated scores as follows:

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

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

$$f - measure = \frac{2 * precision * recall}{precision + recall}$$

For the second measurement, we considered partial matching and calculated scores as in the study [40]:

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

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

In order to make fair comparisons, we used results calculated by utilizing both evaluation metrics, while comparing our algorithm performance with the other systems.

5.1.2. Results and Discussions

We have provided the results using two different criteria which are explained in the previous section. The quantitative results for each entity type and overall scores, which are calculated by considering exact matching in the data set are shown at Table 5-1.

Table 5-1: Evaluation Results of the System Considering Exact Matches

NE Category	*Precision (%)*	*Recall (%)*	*F-score (%)*
Person	*82.19*	*86.33*	*84.21*
Location	*74.62*	*86.50*	*80.12*
Organization	*76.38*	*76.00*	*76.19*
Overall	*76.76*	*83.12*	*79.81*

We are pleased to see that f-score for person and location entities are higher than 80%. The results show that our system is more successful in finding person names than location and organization names. High recall and relatively low precision values for person and location entities indicate that our system tags some entities as a location or person although they are actually not. For example, our system extracts ‘Nisan’ as a person entity from the sentence: “1 Nisan şakaları bu sene de güldürmedi.”. Such cases decrease the performance of our algorithm for location and person entities. On the other hand, f-score is 76.19% for organization type entities, which is a more challenging type, indicating that our system can still be improved to recognize more organization entities. The overall performance f-score = 79.81% can be considered as a successful result among similar rule based systems.

When we took into consideration partial entity matches like in the study [40] we have obtained better performance results as presented in Table 5-2.

Table 5-2: Evaluation Results of the System Considering Overlapping Boundaries

NE Category	*Precision (%)*	*Recall (%)*	*F-score (%)*
Person	*82.43*	*86.52*	*84.42*
Location	*77.54*	*90.12*	*83.35*
Organization	*86.20*	*85.75*	*85.97*
Overall	*81.57*	*88.52*	*84.90*

As it is seen in the table, the performance of person name recognition is almost not changed when we consider overlapping boundaries. This indicates that our system usually finds person names with their correct boundaries. On the other hand, location and organization performance increased by 3% and 9% respectively and the raise of organization performance is remarkable. Also, as expected, considering overlapping boundaries with exact matches enhanced our overall performance to f-score = 84.90% which is better than 79.81%.

We exploit these results while comparing our NER algorithm with the other systems, since most of the other studies use these two evaluation methods. In some studies, it is not indicated whether exact or partial matches are used for evaluation. For such cases, we assume that they use exact matches and compare our exact criteria results with them. Before discussing results with other studies, it should be highlighted that comparisons were not made on the same datasets since they are not publicly available except Çiçekli and Tatar's test corpus.

In the first NER study for Turkish language [22], the authors reported F = 53.04% by conducting their experiments on a relatively small corpus. Since they implemented a language independent system without using gazetteers or dictionaries, our algorithm produced better results with exact matching criteria. Moreover, the advantage of our algorithm is to utilize Turkish language specific patterns to extract named entities. In the study [24], which exploits specific grammar features in Turkish, it is reported that F = 81.97% for person names. They extracted reporting verbs by analyzing Turkish texts and focused on these verbs to recognize person names. Our algorithm's performance for person names with exact matching is F = 84.21 which is better than Bayraktar and Temizel's algorithm performance. We can infer that exploiting person name dictionary with patterns produce better results than using reporting verbs.

Küçük et al. developed a rule based-system [21] which achieved F = 78.7% for extraction named entities from Turkish news articles. In our opinion; the data set they used and ours are similar, since both of them are collected from Turkish news articles. Therefore, comparing this system with ours is a good indicator for understanding the performance of our system. Our rule based system with exact matching criteria produces F = 79.81% which is slightly better than Küçük's system. Although they used more lexical resources and dictionaries such as well-known organizations or locations, our system is more successful. This indicates that our rules are more powerful to extract named entities without using additional dictionaries.

The hybrid recognizer system [27] which is developed by improving the rule based system mentioned above produce better results than its predecessor. The authors added rote learning [41] approach to the rule based system and achieved F = 90.13% calculated by considering partial matches. It should be highlighted that this score includes date and numerical expression matches which have high score than other types. We measured our performance with partial matches as F = 84.90% which is not bad. However, we should note that adding a learning technique to a rule based system improves the performance remarkably, in general. Since we did not exploit a learning technique in our algorithm, the difference between the performance results is not surprising.

Another study [29] which used supervised learning method carried out by Çiçekli and Tatar reported F = 91.08% for exact matched named entities in Turkish. Since our system is conducted on the same data set, performance results are more comparable. Note that, their results include date and numerical expressions which have 96.50% and 92.05% f-score. When we consider only person, location and organization named entities, their f-score is 90.10% which is clearly better than ours. Similarly, by exploiting statistical learning method, Tür et al. reached F = 91.56% which is one of the best scores for person, location and organization names among Turkish studies which we have investigated so far [23]. Since learning techniques in NER systems are superior to rule based methods, the systems exploiting learning techniques with lexical and morphological features produce better results than our rule based system. As a future work we might consider enhancing our rule-based system with learning methods.

5.1.3. Error Analysis

When we analyze the cases that hurt the performance of our system, we observe that the errors more frequently occur in the following cases:

- While carrying out the recognition of organization and location names, our system performs erroneous extractions such as *Benim üniversitem* as an organization name and *Onun mahallesi* as a location name in the beginning of the sentence. The reason is that we tag an entity if an organization or location pattern matches and the previous word begins with a capital letter as in this case. Similarly, some words in the beginning of the sentence may be tagged as a person name by our system if that word appears in the person name dictionary. For instance, in the sentence "Hayati önem taşıyan bu sistemin aksaması birçok probleme yol açar", *Hayati* is recognized as a person name.
- Our system suffers from the erroneous extractions in case of consecutive named entities such as *Murat Yıldız Ağrı Devlet Hastanesinde* although *Murat Yıldız* and *Ağrı Devlet Hastanesinde* are two different entities.

- The system also extracts wrong matches in case of nested entities especially organizations and locations such as *Emniyet Müdürlüğü İstihbarat Şubesi*. For such cases, the system extracts two distinct organization name entities *Emniyet Müdürlüğü* and *İstihbarat Şubesi* although both of them constitute a single organization name.
- During person name extraction, our system erroneously recognize some date expressions such as month name, which is also a person name like *Nisan* or *Ekim*, as a person named entity.
- If a named entity includes numerical expressions such as *100. Yıl Üniversitesi*, our system recognize it *Yıl Üniversitesi* instead of *100. Yıl Üniversitesi.*
- When named entities are not typed by considering capitalization rules, the system does not recognize them even if they exist in our dictionary. For instance *sezen aksu* is not tagged as a person name since both words begin with lower case.

5.2. Evaluation of the TV Content Augmentation System

The evaluation of our TV content augmentation system is the challenging part of our experiments. Indeed, the most suitable method to evaluate the performance of the system is conducting a user satisfaction survey. However, we could not implement the augmentation system on TV as an application due to the time constraints and lack of environment for TV application development. Therefore, we did not have the opportunity to conduct an experiment by using TV application with real viewers.

Instead of conducting a survey with real users, we analyzed the system by considering augmentation performance with NER operation on EPG data. In other words, after extracting named entities from TV program guide data, we calculated the number of named entities which have a response on Wikipedia. Moreover, we estimated average number of NER and NER with detailed information in program descriptions. Determining TV channels which have more satisfactory program descriptions is another operation during the analysis of EPG data. In our opinion, these inferences are

valuable for understanding the performance and behavior of a TV content augmentation system.

5.2.1. Data and Methodology

In order to carry out the experiment, we utilized three-month period EPG data gathered from Radikal TV guide web site as explained in section 4.4.1. Gathered data includes 28 channels and about 45000 programs. Since processing all of the program descriptions is too costly, we have chosen a time interval *prime time*, which is mentioned at dayparting article on Wikipedia[8], to extract program descriptions. *Prime time* programs are broadcast after 8:30pm when most of the viewers are in front of the television. There are about 6500 programs in prime time and we utilized 1700 of them which include 69133 words.

We have considered the performance scores with overlapping boundaries presented at Table 5-3, while commenting on NER results of program descriptions. Since the main goal of TV content augmentation system is to provide the most relevant information about the content, we think that it is enough to determine entities in EPG data with overlapping boundaries for retrieving a detailed description. For instance, when 'Gezi Parkı' entity is extracted from sentence 'Gezi Parkı olaylarında 10 insanımız gözünü kaybetti.' rather than actual entity 'Gezi Parkı olayları'; relevant description about 'Gezi Parkı' gives adequate information about the content to the users.

[8] http://en.wikipedia.org/wiki/Dayparting

Table 5-3: Overall Evaluation Results of the System with Overlapping Boundaries

Data Set	*Precision (%)*	*Recall (%)*	*F-score (%)*
News Text	*81.57*	*88.52*	*84.90*

We have also determined a measurement technique for evaluating the performance of augmentation in our system as follows:

$$Augmented\ Named\ Entity = The\ entity\ has\ a\ Wikipedia\ page$$

$$Augmentation\ Rate = \frac{Number\ Of\ Augmented\ Named\ Entity}{Number\ of\ Entities\ Recognized\ by\ the\ system}$$

5.2.2. Results and Discussions

Table 5-4 shows the statistical results of the experiments on EPG data set performed with the help of our NER and information retrieval modules. Our system recognize 5102 named entities; 3373 person names, 1320 location names and 409 organization names among 69133 words. EPG data set that we processed is not annotated; therefore we could not know how many of the recognized entities are correct. However, by considering our NER performance measured in section 5.1 we can assume that 85% of the recognized named entities are correct.

Table 5-4: Statistics on EPG Data Set

	Number of Person Names	*Number of Location Names*	*Number of Organization Names*	*Total Number of Named Entities*
Recognized by the System	*3373*	*1320*	*409*	*5102*
Augmented by Wikipedia Information	*1166*	*1108*	*253*	*2527*

When we request detailed information about these entities from Wikipedia, the number of named entities for which we can fetch detailed information is 2527: 1166 person names, 1108 location names and 253 organization names. Table 5-5 presents the performance of the augmentation operation. It is clear that the augmentation performance of the system is almost 50% which means half of the recognized named entities does not have information page on Wikipedia.

One reason may be the failure of our NER algorithm. That means the system may not recognize named entities in program descriptions properly. However, when we consider our NER system performance results in section 5.1, most of the named entities are recognized by the system with the high f-score 85%. Moreover, when we investigated some program descriptions including person named entity, which has the worst augmentation performance among other types, there were no Wikipedia page of these person names although they are actually correctly named entities. For example; *Tolga, Nihan and Bülent* are common person named entities and our system tagged them as a person name, however common person names usually do not exist in Wikipedia with detailed information. Therefore, augmentation performance of the

system considerably dropped due to the non-existence of detailed information on Wikipedia.

A notable augmentation performance result occurs for location named entities in our experiment. 83.93% of the location named entities is augmented with detailed information. This indicates that detailed information of location names is more easily reachable on Wikipedia. However, it is not so successful for organization named entities for which the rate of augmentation is 61.85%. We believe that unavailability of some Turkish named entities in Wikipedia yields bad augmentation rates in the experiments.

Table 5-5: Augmentation Performance of the System

	Person Names (%)	*Location Names (%)*	*Organization Names (%)*	*All Named Entities (%)*
Augmentation Rate	*34.56*	*83.93*	*61.85*	*49.52*

Table 5-6 shows the average number of named entities and augmented named entities for each program. It is seen that each program description has 3 named entities and 1.48 augmented named entities on the average. The lowest average number of the augmented entities is organization. Also average numbers of augmented person and location entities are almost the same. These average numbers indicate that if we focus on all channels and programs instead of specific ones, augmentation operation may not be successful as desired. Therefore we also analyzed each channel according to the augmentation performance and the number of named entities.

Table 5-6: Average numbers of EPG Data Analyze

Average per Program	*Person Named Entities*	*Location Named Entities*	*Organization Named Entities*	*All Named Entities*
Recognized by the System	*1.98*	*0.77*	*0.24*	*3.00*
Augmented by Wikipedia Information	0.68	0.65	0.14	1.48

We believe that programs including sufficient number of named entities and having high augmentation rate can increase the augmentation performance in a content augmentation system. Therefore we extracted top 5 channels according to the number of named entities after processing 100 programs for each channel. The results are presented at Table 5-7 with their augmentation rate. It is seen that channels usually broadcasting series such as *Muhteşem Yüzyıl* or *Med Cezir* have higher number of named entities among other channels. Furthermore, these channels are the ones with higher rating in Turkey. Therefore the channels presented in Table 5-7 can be suitable for a TV augmentation system even if their augmentation rates are relatively low.

Table 5-7: Top 5 Channels According to Number of Named Entities with Their Augmentation Rate

Channel Name	*Number of Named Entities*	*Augmentation Rate (%)*
Fox TV	*481*	*34.71*
Star TV	*452*	*33.84*
Kanal D	*392*	*33.67*
Show TV	*387*	*33.35*
Kanal 7	370	32.97

We also extracted top 5 channels according to their augmentation rate which can be seen at Table 5-8. It is interesting that channels having best augmentation rates are the news channels. Most of the news program descriptions contain fewer words among other channels; nevertheless, they are more likely to contain named entities which can be augmented. Therefore, the other suitable channel category for TV augmentation system may be the news channels.

We have observed that news channels and channels usually broadcasting series contain more satisfactory content for augmentation purposes. We exploited this inference while constructing the demo of the TV content augmentation system on PC as explained in section 4.4.4.

Table 5-8: Top 5 Channels According to Augmentation Rate with Number of Named Entities

Channel Name	*Augmentation Rate (%)*	*Number of Named Entities*
A Haber	*94.02*	*67*
Euro Sport	*80.70*	*57*
TRT3 (TRT Spor)	*78.37*	*74*
CNN Turk	*70.81*	*185*
NTV	69.59	148

CHAPTER 6

CONCLUSION & FUTURE WORK

Smart and connected TVs provide web accessing features to the users; however they do not combine TV and web experiences effectively. TV viewers need systems and applications enabling them to get additional relevant information automatically.

In this thesis, we have proposed a TV content augmentation system exploiting named entity recognition methods for Turkish language. We have implemented a prototype by considering user desires for such an augmentation system. The system has four components; EPG data collection, Turkish NER, Wikipedia information retrieval and desktop applications. Firstly, EPG data collection module gathers TV program descriptions by utilizing Radikal TV guide web page. Next, Turkish NER component determines the named entities in program descriptions and extract them as keywords. Detailed information of keywords are retrieved from Wikipedia by considering the disambiguation issue and summarized before presenting to the viewers. The last component, a desktop application demonstrates TV content augmentation system on PC.

In our study, NER implementation on Turkish is the most important part of the system. We developed a rule based NER algorithm by using lexical and contextual features of Turkish language. We created person and location name dictionaries besides pattern base resources which is used for recognizing organization and location named entities. For the evaluation of the performance of our NER algorithm, we conducted an experiment on annotated Turkish corpus. Our NER system produced better results than

similar systems using rule based and local grammar approach, with an overall f-score = 79.81% and person f-score = 84.21%.

We have also evaluated TV content augmentation system by processing 2700 TV program descriptions. We extracted keywords by using NER algorithm and tried to fetch detailed information for them from Wikipedia. We measured the number of named entities in program descriptions on the average and augmentation rate of named entities with respect to TV channels. According to the experimental results, implementing such a TV content augmentation system for news channels and channels usually broadcasting series is more appropriate. Also we observed that our content augmentation system can easily be integrated to smart TVs of Turkish companies such as Arçelik.

Possible future work that is feasible towards extending this work include the following:

- Since learning techniques in NER systems are superior to rule based methods, our NER system can be implemented by exploiting learning techniques in order to increase the recognition performance.
- We utilize only Radikal TV guide in order to gather EPG data of TV programs. However the number of EPG resources can be increased to obtain more satisfactory program descriptions with respect to named entities.
- While augmenting the content, we rely on Wikipedia; nevertheless, utilizing different web sources similar to Wikipedia may result in an improved augmentation rate.
- The implementation of TV content augmentation system can be integrated with Arçelik smart TVs as an application instead of PC prototype. Thus, we may have the opportunity to conduct an experiment by using TV application with real viewers in order to evaluate the behavior of the system.

REFERENCES

[1] A. Taylor and R. Harper, “Analysis of Routine TV Watching Hbits and Their İmplications for Electronic Program Guide Design,” pp. 1–12, 2002.

[2] D. Norman, “Things that make us smart. 1993,” *Addison-Wesley*, pp. 43–76, 1993.

[3] A. Prata and T. Chambel, “Going Beyond iTV : Designing Flexible Video-Based Crossmedia Interactive Services as Informal Learning Contexts,” *EuroITV’11 - Ubiquitous TV Conf. Proc.*, pp. 65–74, 2011.

[4] M. Montpetit, P. Cesar, and M. Matijasevic, “Surveying the social, smart and converged tv landscape: Where is television research headed?,” *arXiv Prepr. arXiv*, pp. 1–18, 2012.

[5] Yahoo, “Mobile Shopping Framework: The role of mobile devices in the shopping process,” 2011. [Online]. Available: http://news.yahoo.com/blogs/yahoo-advertising-solutions/mobile-shopping-framework-role-mobile-devices-shopping-process-232156844.html?soc_src=copy. [Accessed: 17-Aug-2014].

[6] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*, vol. 9. ACM press, 1999, p. 513.

[7] N. Dimitrova, J. Zimmerman, A. Janevski, L. Agnihotri, N. Haas, and R. Bolle, “Content Augmentation Aspects of Personalized Entertainment Experience,” *Proc. Third Work. Pers. Futur. TV*, 2003.

[8] A. Nadamoto, “Complementing Your TV-Viewing by Web Content Automatically-Transformed into TV-program-type Content,” pp. 41–50, 2005.

[9] Q. Ma, A. Nadamoto, and K. Tanaka, “Complementary information retrieval for cross-media news content,” *Inf. Syst.*, vol. 31, pp. 659–678, 2006.

[10] R. Martin and H. Holtzman, “Newstream: A Multi-Device, Cross-Medium, and Socially Aware Approach to News Content,” pp. 83–90, 2010.

[11] T. Chattopadhyay, A. Pal, and U. Garain, “Mash up of breaking news and contextual web information: A novel service for connected television,” *Proc. - Int. Conf. Comput. Commun. Networks, ICCCN*, 2010.

[12] R. Hemsley, A. Ducao, E. Toledano, and H. Holtzman, “ContextController : Augmenting broadcast TV with real- time contextual information,” pp. 833–836, 2013.

[13] S. Robertson, C. Wharton, C. Ashworth, M. Franzke, and C. S. Group, “Dual Device User Interface Design: PDAs and Interactive Television,” in *CHI ’96 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1996, pp. 79–86.

[14] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” no. 1991, pp. 1–20, 2006.

[15] E. Marsh, “TIPSTER information extraction evaluation: the MUC-7 workshop,” in *Proceedings of the TIPSTER Text Program Phase III*, 1999, pp. 233–234.

[16] E. F. Tjong Kim Sang, “Memory-based named entity recognition,” *proceeding 6th Conf. Nat. Lang. Learn. COLING02*, vol. 20, pp. 1–4, 2002.

[17] J. L. Seng and J. T. Lai, “An intelligent information segmentation approach to extract financial data for business valuation,” *Expert Syst. Appl.*, vol. 37, pp. 6515–6530, 2010.

[18] N. H. Sung and Y. S. Chang, “Business information extraction from semi-structured webpages,” *Expert Syst. Appl.*, vol. 26, pp. 575–582, 2004.

[19] T. H. Tsai, W. C. Chou, S. H. Wu, T. Y. Sung, J. Hsiang, and W. L. Hsu, “Integrating linguistic knowledge into a conditional random fieldframework to identify biomedical named entities,” *Expert Systems with Applications*, vol. 30. pp. 117–128, 2006.

[20] R. Grishman and B. Sundheim, “Design of the MUC-6 evaluation,” in *Proceedings of a workshop on held at Vienna, Virginia: May 6-8, 1996*, 1996, pp. 413–422.

[21] K. Dilek, “Named Entity Recognition Experiments on Turkish Texts,” pp. 524–535, 2009.

[22] S. Cucerzan and D. Yarowsky, “Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence,” pp. 90–99, 1997.

[23] G. Tür, D. Hakkani-Tür, and K. Oflazer, “A statistical information extraction system for Turkish,” *Nat. Lang. Eng.*, vol. 9, no. 2, pp. 181–210, Jun. 2003.

[24] O. Bayraktar and T. T. Temizel, “Person name extraction from Turkish financial news text using local grammar-based approach,” *2008 23rd Int. Symp. Comput. Inf. Sci.*, pp. 1–4, Oct. 2008.

[25] H. Traboulsi, “Arabic named entity extraction: A local grammar-based approach,” in *Proceedings of the International Multiconference on Computer Science and Information Technology, IMCSIT ’09*, 2009, vol. 4, pp. 139–143.

[26] B. Say, K. Oflazer, U. Özge, and N. B. Atalay, “METU Tukish Corpus,” 2007. [Online]. Available: http://ii.metu.edu.tr/corpus.

[27] D. Küçük and A. Yazıcı, “A hybrid named entity recognizer for Turkish,” *Expert Syst. Appl.*, vol. 39, no. 3, pp. 2733–2742, Feb. 2012.

[28] D. Freitag, “Machine learning for information extraction in informal domains,” *Mach. Learn.*, vol. 39, pp. 169–202, 2000.

[29] S. Tatar and I. Cicekli, “Automatic rule learning exploiting morphological features for named entity recognition in Turkish,” *J. Inf. Sci.*, vol. 37, no. 2, pp. 137–151, Feb. 2011.

[30] I. Cicekli and N. K. Cicekli, “Generalizing predicates with string arguments,” *Appl. Intell.*, vol. 25, no. 1, pp. 23–36, Aug. 2006.

[31] R. Bunescu and M. Pas, “Using Encyclopedic Knowledge for Named Entity Disambiguation,” no. April, pp. 9–16, 2006.

[32] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, “DBpedia - A crystallization point for the Web of Data,” *J. Web Semant.*, vol. 7, pp. 154–165, 2009.

[33] P. N. Mendes, M. Jakob, A. García-silva, and C. Bizer, “DBpedia Spotlight : Shedding Light on the Web of Documents,” *Proc. 7th Int. Conf. Semant. Syst. (I-Semantics).*, vol. 95, no. 2, pp. 1–8, 2011.

[34] Jimmy Wales, “Wikipedia.” [Online]. Available: http://www.wikipedia.org/. [Accessed: 20-Jul-2014].

[35] P. N. Mendes, M. Jakob, A. García-silva, and C. Bizer, “DBpedia Spotlight : Shedding Light on the Web of Documents,” in *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics).*, 2011, vol. 95, pp. 1–8.

[36] A. Af, “Zemberek , an open source NLP framework for Turkic Languages.”

[37] “MediaWiki,” 2014. [Online]. Available: https://www.mediawiki.org/wiki/MediaWiki. [Accessed: 31-Jul-2014].

[38] “Java HTML Parser,” 2013. [Online]. Available: http://jsoup.org/. [Accessed: 01-Aug-2014].

[39] “Türkçe Adlar,” 2012. [Online]. Available: http://turkceisimlistesi.blogspot.com.tr/. [Accessed: 02-Aug-2014].

[40] D. Maynard, V. Tablan, C. Ursu, H. Cunningham, Y. Wilks, R. Court, and P. St, “Named Entity Recognition from Diverse Text Types,” 1998.

[41] D. Freitag, “Machine learning for information extraction in informal domains,” *Mach. Learn.*, vol. 39, pp. 169–202, 2000.