Morphological Annotation of a Corpus with a Collaborative Multiplayer Game

by

Onur Güngör

B.S., Computer Engineering, Boğaziçi University, 2006

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2009

Morphological Annotation of a Corpus with a Collaborative Multiplayer Game

APPROVED BY:

Assoc. Prof. Dr. Tunga Güngör (Thesis Supervisor)	....
Prof. Dr. Tevfik Akgün	....
Dr. Suzan Üsküdarlı	....

DATE OF APPROVAL: 30.07.2009

ACKNOWLEDGEMENTS

Type your acknowledgements here.

ABSTRACT

Morphological Annotation of a Corpus with a Collaborative Multiplayer Game

In most of the natural language processing tasks, state of the art systems usually rely on machine learning methods for building their mathematical models. Given that the majority of these systems employ supervised learning strategies, a corpus that is annotated for the problem area is essential. The current method for annotating a corpus is to hire several experts and make them annotate the corpus manually or - in its best practice- by using a helper software. However, this method is costly and time-consuming if not error free. Our work proposes a method that aims to solve these problems at once. By employing a multiplayer collaborative game that is playable by ordinary people on the Internet, we think that it is possible to direct the covert labour force so that people can contribute just by playing a fun game. Through a game site which incorporates some functionality inherited from social networking sites, people are motivated to contribute to the annotation process by answering some questions about the underlying morphological features of a target word. The results reported in the thesis are compiled from the first eleven days of the experiment which is planned to continue until an indeterminate date. It is reported that the 63.5% of the actual question types are successful based on two phases. The current 74 question types cover 58.3% of the corpus completely while increasing this number to only 100 types increases the coverage rate to 70.7%. Due to the time constraints and the relatively low traffic to the site, we were not able to annotate the corpus completely, but we can nevertheless estimate a hypothetical rate of successful morphological disambiguation as 37.0% of the whole corpus which is calculated to be completed in two and a half months if the game were to be hosted on a major web site. This is indeed a relatively short duration for a bootstrapping of this size when compared with the current methods.

ÖZET

Çok Oyunculu ve Yardımlaşmacı bir Oyun Aracılığıyla Türkçe bir Derlemin Biçimbilimsel İşaretlenmesi

Doğal dil işleme görevlerini gerçekleştirmek için geliştirilmiş en gelişkin sistemler çoğunlukla matematiksel modellerini kurarken makine öğrenmesi yöntemleri kullanırlar. Çoğunun öğreticiyle öğrenme yolunu seçtikleri göz önüne alınırsa, çözüm gerektiren doğal dil işleme sorununa uygun olarak işaretlenmiş bir derlemin zorunluluğu ortaya çıkar. İşaretlenmede kullanılan güncel yöntem, ilgili konuda uzmanlaşmış kişilerin elle veya yardımcı bir yazılım kullanarak işaretlemeyi gerçekleştirmesidir. Lâkin, bu yöntem yer yer hatalar içerebilmesinin yanında, masraflıdır ve uzun zaman gerektirir. Bizim yöntemimiz bu sorunların hepsini bir anda çözmeyi hedefler. Herhangi bir internet kullanıcısının oynayabileceği yardımlaşmacı bir oyun aracılığıyla, yalnızca eğlence amaçlı bir oyunu oynatmak marifetiyle açığa çıkmamış işgücünün derlem işaretlenmesi yönünde değerlendirilebileceğini düşünüyoruz. İnsanlar, sosyal ağ sitelerinden devşirilmiş bazı özellikleri de taşıyan bir sitede karşılarına çıkan belirli bir sözcük hakkındaki sorulara cevap vererek işaretleme sürecine katkıda bulunmaya teşvik ediliyor. Tezde verilen sonuçlar gerçekleştirilen deneyin ilk on bir gününden oluşturulmuştur. Deney belirsiz bir tarihe kadar devam etmek üzere hala çalışmaktadır. Sonuçlara göre, halihazırdaki 74 soru çeşidinin iki fazdan oluşan değerlendirmesine göre %63.5'lük bir başarı oranı yakalanmıştır. Bahsi geçen soru çeşitleri derlemin %58.3'ünün biçimbilimsel çözümlemesini yapabilmektedir. Soru çeşidi sayısını 100'e çıkarmak, bu oranı %70.7'e çıkaracaktır. Zaman kısıtı ve ziyaretçi azlığından dolayı bahsedilen düzeyde bir işaretleme yapılamamasına rağmen, ulaşılacak başarı oranı üzerine bir tahmin yapmak söz konusu gerekirse %37.0 oranı elde edilecektir. Bu işlemin, büyük bir ulusal gazetenin web sayfasında gerçekleştirildiği takdirde, iki buçuk ay içinde tamamlanacağı düşünülmektedir. Bu süre, bu çaptaki bir işaretleme işi için göreli olarak kısa bir süredir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

1.1. The Problem

In most of the natural language processing tasks, state of the art systems usually rely on machine learning methods for building their mathematical models. Given that the majority of these systems employ supervised learning strategies, a corpus that is annotated for the problem area is essential.

But having a relevantly annotated corpus is not enough on its own. The corpus must have a number of crucial features. First, it must include a set of carefully selected examples so that the method can train the model without bias. For the training to be successful, the corpus must include a specific number of examples which is mainly determined by the characteristics of the training method itself. Additionally, while being able to train, the corpus must not introduce bias to the trained model. Second, the corpus must be free of errors. While some methods may be resistant to several kinds of errors in the corpus, in most cases it prevents the method from training the model to its maximum extent.

When we recognize the crucial position of an error-free corpus having a vast number of examples in solving natural language processing tasks, the problem of building a corpus with these properties gains importance. The most prominent method of building corpora today is to divide the work among some number of experts and wait for them to finish their own work. However, it can be argued that this method is flawed in a number of points.

First of all, the method dictates that the people who work on the work units must be experts in their field. In other case, they must be educated to handle the task. In either case, we need to spend a substantial amount of the research fund to hire experts or -if they are not available- spend time (which is another important resource) to find one or to spend both time and money if we had to educate people to be able to work on the task. Even if we were successful in finding and hiring experts to work on building the corpus, there are other things that hinder the process. For example, the annotation patterns of two experts -even if they are highly experienced in the area- may be very different and thus the resulting annotation may contain inconsistencies. This is believed to be common for especially small and spontaneous annotation projects where experts do not work in pairs and do not later correct inconsistencies with their pairs.

As a result of these problems, the process of building a corpus using the current methods is slow and expensive, if not low quality. This in turn affects the development rate of natural language processing research as well as the scope of it. This thesis recognizes this problem as an important hindrance to the further development of natural language processing research and proposes a new method for building corpora. But before continuing with the proposal of the thesis in the next section, we would like to describe the natural language processing task which we chose for applying the new method.

We chose Turkish morphological disambiguation as the target domain because of three reasons. First, this problem is at the core of other Turkish natural language processing tasks, i.e. parsing, speech recognition and sense tagging to name a few. Second, there are several other research paths going on around the natural language processing group at the artificial intelligence laboratory in the department. And as the last reason, we had a corpus tagged for this task which enables us to test our results. In fact, the corpus mentioned Yüret is one of the very few annotated corpora in Turkish. More details on the morphological disambiguation problem itself will be given in Section 1.4.

1.2. Proposal

What we propose for a new solution to the problem of annotating a corpus is basically to build a system for harvesting the free time of Internet users towards the completion of this job. Before detailing further, we want to elaborate on whether this system really solves the problem or not. To be assured of this, we have to look at a few facts and make some design decisions.

First of all, we want the annotation process to be fast, or at least faster than today's methods. When we look at the current state of Internet usage in Turkey, we see that there are 26 million Internet users Afra (2008). And if we speculate on the amount of time spent online by an average Internet user, we think that it is reasonable to assume that an average user spends an amount of time comparable to time spent watching TV. Having made this assumption, we can conclude that if we are successful in making people devote even only a very small proportion of their total time spent online to our method instead of other time spending activities, we would be employing a very huge labor force for our needs.

The idea of people working for us instead of visiting other sites seems fine, but we have to find a way of making a project for relatively serious purposes achieve that result. The answer is games. If we could be able to transform the process of morphological tagging which requires the operator to have some knowledge of word morphology to an online game which can be played by people with no education in the field, we would be enjoying that huge labor force for building an annotated corpora. However, transforming the problem into a game does not automatically solve all problems. There are several points to consider to make the game playable and indeed played by ordinary people. Most importantly, in addition to having lower barriers for people who are not experts, the game must be fun and be able to motivate people by other means such as competition or being helpful to science.

This schema has additionally the positive effect of avoiding the errors introduced by experts in the previous methods. This is because in our method, the annotation process is achieved by having a statistically significant number of people to annotate the same example.

Finally, the proposal for our thesis can be described completely. We aim to build a system which incorporates a collaborative game which is played by one or two users at a time. Two modes of game play will be available, one is the single player game in which the users can answer quiz-like questions as long as they like; the second will be played by two users simultaneously while one user tries to explain a concealed word to the other and meanwhile answer some questions that are valuable for our annotation needs. The game will be open to anyone and hosted on a web server publicly accessible. We want to evaluate certain aspects of the system. Most importantly, we aim to make an assessment of our abstraction of morphological constructs solidified as questions in our game. Second, but of nearly equal importance, we want to extract some game design tips by observing the interaction of the users with our game setup.

1.3. Related Work

In this section, we give a survey of game design aspects with a focus on human computing.

We started our survey by examining the references of the seminal work of Luis von Ahn. We continued with going through the citations of those papers and began browsing the publications of related conferences. This process essentially helped us to draw some guidelines for game design. Second line of work was to inspect the related ACL conferences. However, rather than finding relative work that describes a method of abstracting the morphological data, we found several papers that report on some experiences of primitive cooperative interfaces for corpus annotation. An additional contribution of these work was to enable us to see that a general annotation scheme would be very helpful if we keep in mind that the adaptation of the idea to other linguistic task data collection problems would be highly rewarding. Finally, to be able to generalize the idea of turning particular problems into games, we surveyed some books and several resources on the Internet. Additional resources (such as Internet sites or informal communication) that were accessed during the process will be mentioned in relevant sections of the document.

We gathered many design ideas from the work of Luis von Ahn scattered over several years from 2004 to 2007. The first game he designed was called ESP Game von

Ahn and Dabbish (2004) in which players are matched up with each other randomly and expected to win points by matching their inputs when viewed the same image simultaneously. Given that no other means of communication is possible, the most obvious thing to input is the most distinctive figure in the image. It posed a nice challenge, this caused people to have a lot of fun and some of them eventually grew an addiction which lead to a very effective and fast way of labeling images on the web. This is the seminal work which introduced the idea of turning particular problems into games that people enjoy by harvesting the "wasted human-cycles"[1] .

Later games by Luis von Ahn further extend the idea to various areas. Peekaboom von Ahn et al. (2006c) utilizes the idea to mark the portions of the images that depict target labels. Phetch von Ahn et al. (2006a) collects text descriptions of images by making one player to describe the image and a group of players to simultaneously make guesses from the set of images they are confronted by a search engine result. Verbosity von Ahn et al. (2006b) collects facts about objects again by exploiting the collaborative game play method explained before. In Verbosity, one player must make the other player to guess the secret word that is exposed to her. To give clues to the other user, she could only use some predefined sentence templates like "it contains __". When blanks are filled with appropriate data, this input conveys very much information about the object in question. Last game that Ahn designed is Tagatune Law et al.. It aims to transform the work of tagging music clips into a game. It works much like ESP Game. But it seems like it could not be that successful mainly because it is difficult to agree on a common word to describe the clip and listening to a sound could take a bit and become boring.

In Chklovski (2005), a method for collecting alternative forms of phrases, namely paraphrases is discussed. For achieving their goal, they develop a web site that makes people cooperate. The most important component of the system is their partial hinting system. By default, they already have 2-3 paraphrases. But they want to increase this number. So this requirement is realized with the partial hinting component of the game. At the start of the game, no hint is given and users are expected to enter novel

[1] A term coined by Luis von Ahn to refer to the term "cpu-cycles"

paraphrases of their own. If they are able to guess the already known paraphrases, this contributes to the confidence of that paraphrase. Otherwise, the contribution is stored as a new paraphrase to be guessed by other contributors. This much like resembles social bookmarking sites. After guessing a paraphrase, if it is unsuccessful, the partial hinting mechanism reveals 33% of the already obtained paraphrases like “this ... help”.

A follow-up paper Chklovski and Gil (2005) of the previous work draws five design decisions. First, it is important to fine tune templates which will collect semantic information (and will abstract morphological data in our case). Besides fine tuning, it is necessary to provide guidance to users. It is also advisable to break the annotation process into several steps to be able to distribute the work among users. This way multiple users could validate the annotations. Also it would be good to have a way to automatically repair the contributions at least to some extent.

Casey et al. (2007) transfers the idea of collaborative tagging to physical locations. Inspired by ESP Game, the players in this game try to guess what the other players could have guessed about the current area. A much emphasized point which we must take into is that whatever complexity the rules of the game have, users must only be faced with little or no rules. The rules of the game should be learned from the points earned during game play.

In Su et al. (2007), the question of whether some problems idiosyncratically are much suitable for collaborative solving is asked while the main research is on another topic which is an evaluation of a pre-qualification mechanism for increasing the accuracy of contributors who work on a pay-per-answer basis.

Ames and Naaman (2007) investigates the motivation behind the increased use of tags in web services that are noted with their social aspects. They publish the results of a research project which employed ZoneTag (a cellphone program which uploads photos to Flickr) with 500 people whom uploaded about 45000 photos in 6 months. They draw some conclusions after making semi-structured interviews with a group of selected participants. We think that some of them are applicable to general user interface design, so it is appropriate to include here concisely: make annotation easy, do not always force them to annotate, relevant annotation suggestions facilitate users to annotate later, enable users to modify and add annotation later.

In Richardson and Domingos (2003), it is suggested to have a rewarding mechanism which is not only based on instant rewards after successful annotation but rather enable users to earn points later when some other player makes the same annotation.

In Choi et al. (2007), authors have observed the results of changing the levels of task and reward interdependency in online games. High task interdependency requires each member's effort to be in accord to accomplish the task, while in low task interdependency each member of the team is independent and it may be enough to act on her own. High reward interdependency distributes the reward equally among the team members. Low reward interdependency distributes the reward according to player's individual performance. After setting up an experiment and collecting the results, it is concluded that low task interdependence is more fun only when its is supported with low reward interdependency. But the perceived performance and fun increase when both task and reward interdependencies are high. We can also learn our lesson from this experiment and design the rewarding mechanism such that collaboration favors them the most. Actually, at first thought we can say that this outcome resembles some game schemas from game theory in which cooperation results in the best outcome for all players.

A semi-collaborative approach to corpus annotation is described in Bontcheva. But the system simply acts as a data repository that can be accessed simultaneously or not over the web (Ma et al. (2002) is also similar in this way). So the ability or maybe the chance to work online in a collaborative manner is not fully exploited. However, a well thought mechanism is implemented: the contributors are presented with a readily annotated text which is output by a program which accomplishes the task that the collected corpora will help developing programs for. We think this can be further extended to incorporate active learning in the system.

A work by Gülşen Eryiğit Eryiğit (2007) describes a standalone (non-web) program which can be used as a tool for dedicated contributors. Relying on specially trained people to annotate the corpus is destined to be slow and costly, despite the increase in speed by using this tool.

In Stührenberg et al. (2007), several users can annotate the corpus, and later one "consensus user" selects the best annotation. Thus, we think the cooperation aspect of the project is weak by design. Additionally, contribution requires specialized knowledge in the area and no ordinary user can help readily. But the technology used in implementing the web site[2] is one of the aspects of the work that must be adopted.

Finally, I would like to note our observation of the current state of game sites in Turkey. As we will be targeting a Turkish speaking audience with online game playing habits, we wanted to evaluate the current level of activity in such sites. What we expected was to measure the relative size of this society. While the list does not contain all gaming sites on the Internet, we visited at least ten sites with the total user base in the scale of hundreds of thousands. Also recall that at the time of writing, there are nearly three million homes connected to the Internet through ADSL service, a fact that strengthens the success probability of the deployment of the thesis idea (unfortunately, a reliable reference could not be found other than news sources).

1.4. Morphological Disambiguation

The minimal unit that contains meaning in word is the morpheme. Morphemes are used to build words by combining in several different ways. They are divided into two big classes: stems and affixes. The stem is the morpheme that represents the basic meaning of the word. For example, in the word "insanlar" (humans), "insan" (human) is the stem and the "-lar" morpheme (corresponds to the plural suffix in English) is an affix which follows the stem. The affixes are also composed of several classes: prefixes, suffixes, infixes and circumfixes. While suffixes are added after the stem (like in the previous example), the prefixes precede the stem. On the other hand, infixes are

[2] http://coli.lili.uni-bielefeld.de/serengeti

inserted inside the stem and circumfixes both precede and follow the stem.

Table 1.1: All Possible Parses of the word “kalemlerini”

1)	kalem[Noun]+[A3sg]+lArH[P3pl]+NH[Acc]
2)	kalem[Noun]+lAr[A3pl]+Hn[P2sg]+NH[Acc]
3)	kalem[Noun]+lAr[A3pl]+SH[P3pl]+NH[Acc]
4)	kalem[Noun]+lAr[A3pl]+SH[P3sg]+NH[Acc]

The parse of a word is compiled of these morphemes and some features that does not have a surface form. There can be several parses of the word due to the ambiguity in the stem and the morphemes (see Table 1.1). Note that in the Table, the second person single possesive marker, the third person plural possesive marker and the third person single possesive markers are both alternatives to each other. This is because their resulting lexemes found in the surface form of the word become the same when combined with the next suffix. These alternative parses can be obtained by using a morphological parser. The morphological disambiguation problem is to select the correct morphological parse of a word in a given context among all of these parses of a word.

As our focus in the thesis is to build an unambigously annotated corpus for morphological disambiguation of Turkish, we would like to list some of the current approaches to the problem. A trigram-based statistical model is presented in Tur and Oflazer (2000). In Yuret and Türe (2006), a decision list induction algorithm is introduced for performing morphological disambiguation. There are also several constraint-based methods for disambiguation Oflazer and Tur (1996), Oflazer et al. (1997). Another method employs a perceptron algorithm for morphological disambiguation Sak et al. (2007). We use the tool produced by this study as a morphological parser for our various needs. These ranged from preparing the corpus to the online question generation. Given that the most of these methods use supervised learning algorithms, it can be said that a corpus that is fresh and error-free would help these methods and the future development.

1.5. Outline

In the next chapter, you will find a brief description of the problem of morphological disambiguation which focuses on the Turkish case. Readers accustomed with the problem should skip the chapter. After that chapter comes Chapter 2 which elaborates on the very finest details of the game and the overall system that encapsulates the game. In Chapter 3, we describe the experiment's setup and the results obtained after the experiment. In Chapter 4, we evaluate the results obtained mainly on the basis of the assessment of questions. This chapter also includes a collection of game design tips compiled from the successes and failures of the current game design. In the last Chapter 5, we draw conclusions and talk about some further research topics to pursued.

2. The Game

As stated in the proposal, (Section 1.2), our method for building an annotated corpus heavily relies on a successful design and implementation of an online game which must possess several properties that are essential for both playability and robustness of annotation.

In this chapter, we will first talk about these properties detailing how they are reflected in the design of the game. Then in the following three sections, we will describe the game site and the two modes of game play present on the game site. Section 2.4 mainly talks about the abstraction of morphological constructs. This abstraction is used for generating questions for the game, both in single and two player modes. The last section of this chapter details the architectural decisions and gives information about some implementation details of the game that we think are important for further development on the subject.

We continue with elaborating on the crucial properties which the game must possess. First of all, the game must be playable by ordinary people who are not necessarily educated in the field. This means that we have to find a way to break up the disambiguation process into pieces to be able to tailor the process for non-experts. To do that, first we have to define the morphological disambiguation process with more detail.

In literature, we separate two concepts: morphological parsers and morphological disambiguators. Basically, to analyze a word's morphology is to enumerate all the possible parses of that word. What disambiguators do is to select the parse with the highest probability to be that word's correct parse. The actual method that is employed by these disambiguators may change, it varies from rule-based systems to systems equipped with perceptrons.

At this point, we assume that humans are equipped with a covert ability to sense the correct parse of the word. This ability is learned in the childhood but there is no known way of consistently describing this ability so that it can be programmed to be executed on computers. Thus it seems reasonable to generate all possibilities with a morphological parser and then somehow make the user select the correct parse. One huge problem here is that these parses cannot be directly understood by a person without knowledge on the subject. Given the facts that humans covertly "know" to separate the good parses form the bad parses and that the raw parses are not sufficiently clear, we find it useful to form questions acting as an abstraction layer between the user and the raw parses. Thus, we propose to discard bad parses from the set of parses by asking questions of two types; yes/no questions and multi-option questions. These questions must be prepared so that they are automatically generated for any word in the corpus and be clearly understood by the users. By asking this question to a statistically sufficient number of users, we became assured whether the parses that are to be discarded will be discarded or not.

Possibly there will be other questions, because one question will discard only a portion of the set of all possible parses. However, after aggregating the users' answers for this questions, we will have discarded all the bad parses. This means that we have finished disambiguation and left with the correct parse.

In conclusion, our game is capable of generating questions for the words in the corpus automatically. These questions are asked in several stages of both the single and two player game. After aggregating sufficient number of answers, the correct parse of the corpus word is detected. The process of question generation will be detailed in Section 2.4. Section 1.4 may serve as a quick refreshment of knowledge of morphological disambiguation.

An additional aspect of the game is that it must be publicly accessible by our target population. To provide this, we chose to host the game on a web site which is accessible at any time of the day and without device restriction. One can access the site by just having the standard equipment which is used to browse the web, namely web browsers. Moreover, we allow people to access our game without formal introduction or qualification tests. This is unlike the previous corpus annotation efforts in which nearly all of them require their contributors to be known and recognized by the people responsible with the process. If we recall that they also usually require the contributors to come to a special office where the work is done, the advantage of our approach is recognized better. In summary, we host the game on a publicly accessible site and allow anyone to join and start the annotation. This in turn makes the potential level of participation (thus work accomplished) much higher than the previous annotation methods. If we take into account that the Internet is maybe the most frequently utilized time killing activity, we can assume that this potential to grow even more.

Motivation of the users is another issue which is very closely related with the game design and the site that it is contained. We have three basic notions for building and nourishing motivation.

The first is fun. If the game is fun enough, people will begin to grow an addiction to the game instead of other time spending activities which sometimes can be boring in themselves. To provide the fun element to the game, we introduce a special stage in the game. This stage contains similar elements from Taboo and a famous game in which you try to explain some film title to the audience without speaking. As you might recall, in Taboo, similar to the game about explaining film titles, you are trying to convey a specific concept to the audience without using some words which are prohibited from using -even parts of it. This stage of the game, we call it as the taboo stage for simplicity, is activated only when playing the two player game. One of the users are chosen as the teller and the other as the guesser. The objective of the teller is to give clues about some specific word to the guesser to accomplish her own objective which is to guess the word as fast as possible. The word that is to be conveyed is actually a word in its sentence context. The sentence is shown to both players. But, obviously, the word in question is concealed from the guesser. The two players enjoy a sense of cooperation while the teller gives clues and the guesses tries word after word. At the same time, they are challenged with a time limit that keeps them alive and attached to the game. Additional points that add to the fun element will be discussed when describing the single and two player games.

The other aspect of the game which is thought to increase motivation is competition. Naturally, people tend to compete with other people when challenged with a fairly hard problem. The key point here is to design the game so that it is neither too hard nor too easy. We employed several methods for building motivation. The run against the time limit in stage 2 is itself a competitive factor. In that stage, players compete against the time cooperating with the other player. This forms the basic motivation for the game. Another method is to build motivation by introducing competition based on group membership. This idea is based on the fact that it is known that people form around groups to enjoy group membership advantages. These advantages can vary from just declaring that someone is a member of a prestigious group to gaining benefits for themselves by using the connections among the group. The site which the game is embedded provides users a way to create and join groups as they wish. People can create groups to represent their school, their football team or a way of thinking. People can also do this for completely arbitrary groups. When a group is created, anyone who wants to join is allowed, and as a result the points that are earned by that user are added to the total points of the group. Competition among the groups are thus constituted. We expect to see the total motivation to build up as a result of this competition.

Another dimenson of the competition factor in the game is to focus on individual representation. As it can be guessed, besides group membership, people pay attention to keep their online presences in a state which is desirable by other people. And to do that, people may want to devote a lot of time to earn high points in a game if the result is to be presented to a lot of audience as a highly skilled person. Thus, in order to exploit this behaviour, we present the highest scoring ten users on the home page of the game site. We assume that people will be motivated to get into that list. In addition to showing off high scores, we could introduce the usual methods that are employed in the social networking sites that are proven to increase participation. Among them the most important functionalities are being able to upload a profile picture, present a short description of herself, add friends indicating that they are interested in each other, messaging between users and a forum where users could discuss about the game including strategies or similar topics.

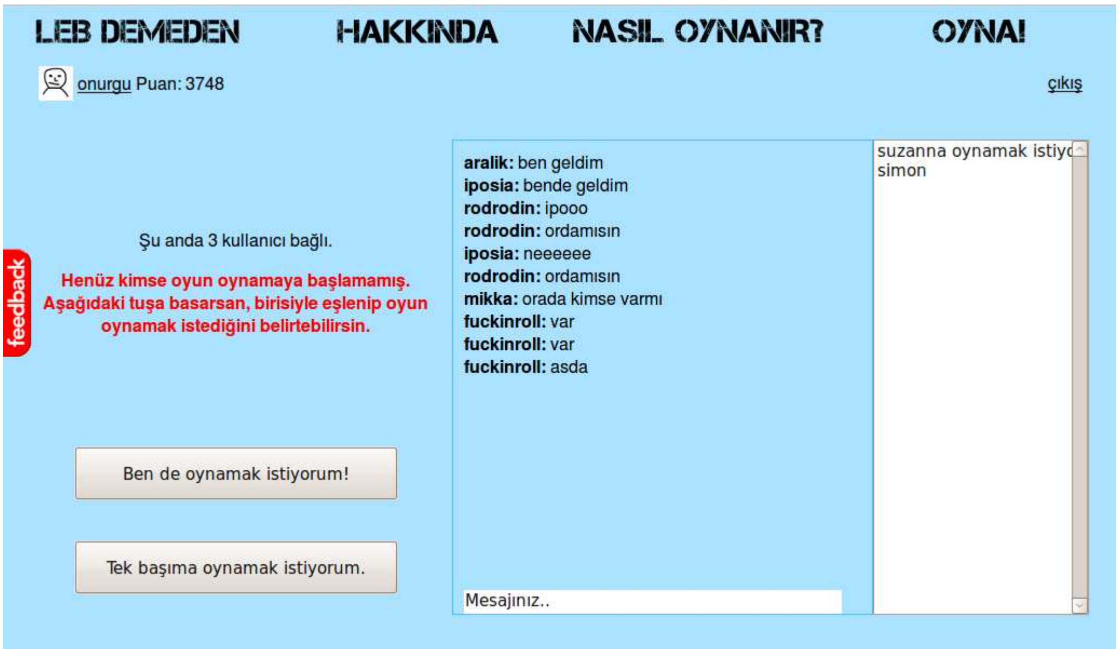
Having fun and competing against other players and groups may be motivating, but we must not forget the motivating force of being helpful to science from the warmth of their house. A lot of people today are excited about the current development pace of science and technology. This makes it reasonable for them to spare a little time of their own to have a little contribution of their own to this pace. But what is crucial here is to establish a clear and interactable interface in which they can also satisfy their recreational needs, such as group membership and identity constuction. To address this issue, we clearly state that the game that is meant to be played is in fact a collaborative effort for contributing to science. We think that this is also a big motivation for some of our target population, if not most.

In the next three sections, we will turn these concepts and design decisions into concrete examples from the game site and the two modes of game play.

2.1. Game Site

The game site basically consists of four main pages. The most obvious one is the home page. This page includes an emphasized link to the game page and the lists of best performing groups and users. These lists are updated in real time. As explained in the previous section, they are meant to build motivation for the users. Maybe the most important page of the site is the page that hosts the game. In this page, which we call the game lounge, the players are welcomed and presented with a list of other online players (Figure 2.1). Here they can communicate by using the chat widget on the page. This is especially important because it is used for acknowledgment to join a game with other people. Alternatively, people may choose to play on their own. In fact, the single player game is played only when there is no available player on the game site to match up.

There are two other pages with secondary importance. The first one gives infor-



Şu anda 3 kullanıcı bağlı.

Henüz kimse oyun oynamaya başlamamış. Aşağıdaki tuşa basarsan, birisiyle eşlenip oyun oynamak istediğini belirtebilirsin.

aralik: ben geldim
iposia: bende geldim
rodrodin: ipooo
rodrodin: ordamısın
iposia: neeeeee
rodrodin: ordamısın
mikka: orada kimse varmı
fuckinroll: var
fuckinroll: var
fuckinroll: asda

suzanna oynamak istiy
simon

Figure 2.1. Game Lounge

mation about the thesis, our motivation and contact details. The second of this group of pages includes information about how the game is played. All of these four pages include links to each other to faciliate navigation throughout the site. Other than these pages, there are automatically generated pages for each user and group. Profile pages and group information pages are of this kind.

The anticipated usage pattern of a new visitor to the site starts with registering a user on the site. After getting a handle and logging in, the user can visit the game page and start playing. Alternatively, the user can visit other three pages mentioned before at any time she wishes.

One aspect of the site along with the game itself is designed so that it is aesthetically acceptable. Though the overall aim of the thesis will benefit from the good looking of the site, we lack the required traits for designing a site that matches the level that it would be eye catching enough. This becomes inevitable when we think of the time constraints. In any way, the game manages to sustain a level of aesthetics that is sufficient enough for our primary purposes.

Another issue for increasing participation in the site that was mentioned before is to include some methods that are utilized in social networking sites. Some of these include being able to upload a profile photo, adding people indicating that they are interested in each other, having forums where people can discuss strategies and problems about the game itself, or use merely for socializing. Among these functionality, we could only include group membership due to time constraints which was chosen because it was the feature that we think that we would get most use and easier to implement than the others.

The actual game site is hosted on a server belonging to the artificial intelligence lab in our department's network. However, for the game to be attractive for our target audience, we had to choose an easy-to-remember domain name. After a quick survey, we decided on lebdemedenleblebi.com. In Turkish, there is an idiom like "Leb demeden leblebiyi anlamak". It is used for people who understand a concept or a fact quickly or even without explaining them. This name was chosen because it successfully gives the message that the game is about guessing something as quick as possible or maybe without waiting for clues. The actual game site can be visited on the URL: http://lebdemedenleblebi.com[3] .

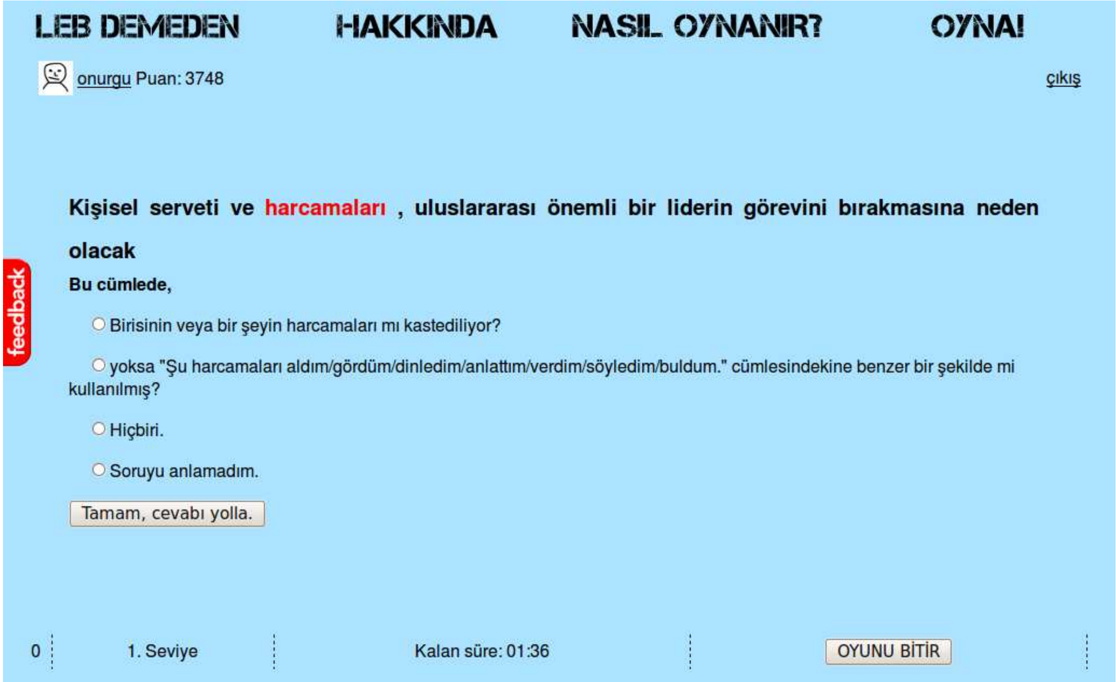
Another module on the site is a button which the users can give feedback on the site and the game itself, this functionality is provided by a third party service provider.

2.2. Single Player Game

In single player game mode, the player is first shown a sentence from the corpus. One of the words in the sentence is marked with a distinctive color, namely red. The player is asked a question that is designed to detect a morphological feature of the indicated word. The answer of the player is stored and the game advances. The finest details of the process of question generation will be handled in Section 2.4.

The next stage is actually the same as the previous stage but this time another word from another sentence is selected and displayed along with its context. A new

[3] Last accessed at Jul 2009.



Kişisel serveti ve harcamaları , uluslararası önemli bir liderin görevini bırakmasına neden olacak

Bu cümlede,

○ Birisinin veya bir şeyin harcamaları mı kastediliyor?

○ yoksa "Şu harcamaları aldım/gördüm/dinledim/anlattım/verdim/söyledim/buldum." cümlesindekine benzer bir şekilde mi kullanılmış?

○ Hiçbiri.

○ Soruyu anlamadım.

Tamam, cevabı yolla.

Figure 2.2. Single Game. The first level of a single game.

question for the word is generated and posed to the player. Like in the previous stage, the answer is stored and this continues in a cycle. The game must be ended by the player itself. An example of a single game can be found in Figure 2.2.

The target words are selected according to the experiment plan which is prepared before the experiment began. The plan for selecting the words is prepared while making sure that every type of question gets a statistically significant number of answers. The method employed in preparing the experiment plan and the motivations behind this choice will be detailed in Section 3.1.1.

To make the player answer in a reasonable time, there is a time limit on the first stage which was set to two minutes during the experiment. A discussion on this time limit and a number of better alternatives will be given in Chapter 4.

2.3. Two Player Game

Before starting a two player game, the system matches two users who indicate that they are willing to join a two player game session. After a pair is matched up,

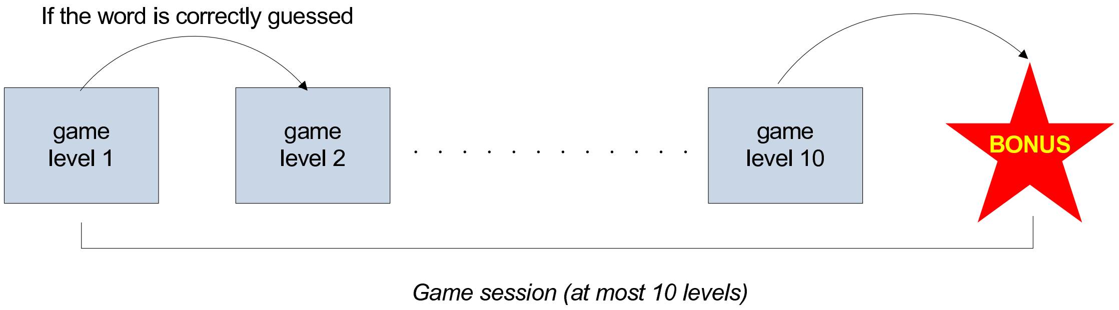

If the word is correctly guessed
game level 1
game level 2
game level 10
BONUS
Game session (at most 10 levels)

Figure 2.3. Game Session and Games.

they are registered for the same game session.

The game session consists of games that are played consequently (Figure 2.3). The rules of winning a game session is that you have to win all the ten games in a row. If you are not able to win a game in the process, you are not allowed to go to the next game and as a result the game session ends.

We call one of the players as “the teller”, the other as “the guesser” throughout a game.

A game of two player mode consists of three stages:

1. the question is asked to the teller
2. the taboo stage
3. the question is asked to the guesser

In the first stage, the teller is first shown a sentence with one of the words marked. Then the player is asked a question that is generated automatically to test the existence of a morphological feature in the indicated word. This question is typically a yes/no question or a multi-option one. This is basically the same with the single game which is given an example in Figure 2.2. The answer submitted by the player is stored and the player is awarded 50 points. Then, the game advances to the next stage.

Meanwhile, the guesser waits for the teller to answer the question while the game displays the same sentence but the target word concealed. This is to warm up the guesser to the second stage and help her to build up some excitement instead of waiting tediously.

In the second stage which we call the taboo stage, the same sentence and the indicated word is shown to the teller. But the guesser still doesn't see the concealed word. The objective of this stage is to operate collaboratively to guess the word as quick as possible.

Through an interface which they can communicate simultaneously, the teller tries to give as many clues as possible while the guesser acts upon these clues to guess the target word.

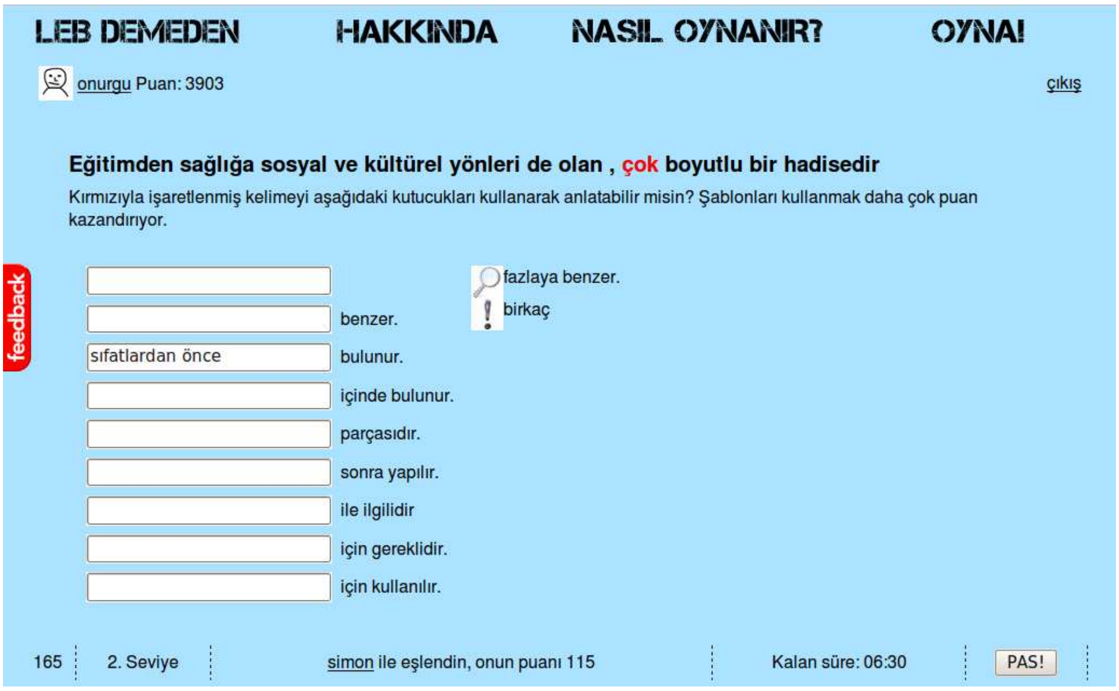
The interface for the teller is different form the interface of the guesser as you can imagine. While the guesser can only utilize a single text box to submit her guesses, the teller's interface contains much more text boxes. Compare Figure 2.4 with Figure 2.5. There are a total of nine boxes which the teller can fill with clues. However, each of these boxes differ in the meaning they convey when used. The first box is for clues that are input in free form. While it would be sufficient for the communication between the users, we design the remaining boxes so that each of them reflects another semantic relation between the clue input and the target word itself. We call them clue templates.

The motivation behind these additional text boxes is to gather more fine-grained information about the target word. In fact, we see this is a side effect of the proposed game. A game feature which we add to make the game fun turns out to be helpful for another purpose in the end. This extra information about the word itself possibly can be used for sense tagging. The actual decriptive text on these clue templates and the meanings associated can be found in Table 2.1.

The first text box and the clue templates also differ in the points the teller gains when submitting using them. The points you get is higher if you use the text boxes which correspond to semantic relations. The actual numbers are 5 to 50 points which

Table 2.1. Clue Templates

Clue Template	Semantic Relation	Description	Example
______ benzer.	Similarity	Defines a similarity between two objects.	vapur “arabaya” benzer.
______ bulunur.	LocationOf	Location information.	araba “yolda” bulunur.
______ içinde bulunur.	a special case of LocationOf	Inside another object.	kalem “kalemliğin” içinde bulunur.
______ parçasıdır.	PartOf	Being a part of another object.	tekerlek “arabanın” parçasıdır.
______ sonra yapılır.	LastSubeventOf	Done after another process.	düğün “nişandan” sonra yapılır.
______ ile ilgilidir.	Related	Is related with the object or the concept.	zeka “dil” ile ilgilidir.
______ için gereklidir.	EventRequiresObject	This event requires another event to function.	ayağa kalkmak “yürümek” için gereklidir.
______ için kullanılır.	UsedFor	This object is used for doing something.	balta “odun kesmek” için kullanılır.



onurgu Puan: 3903

Eğitimden sağlığa sosyal ve kültürel yönleri de olan , çok boyutlu bir hadisedir

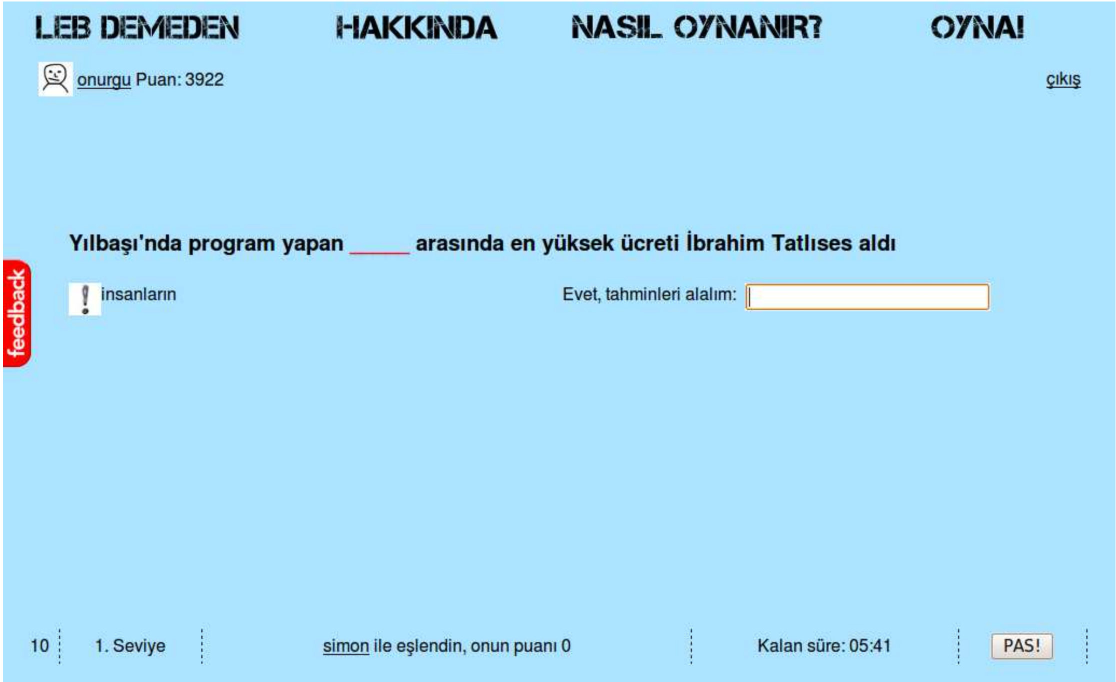
Kırmızıyla işaretlenmiş kelimeyi aşağıdaki kutucukları kullanarak anlatabilir misin? Şablonları kullanmak daha çok puan kazandırıyor.

fazlaya benzer.
birkaç

benzer.
sıfatlardan önce bulunur.
içinde bulunur.
parçasıdır.
sonra yapılır.
ile ilgilidir
için gereklidir.
için kullanılır.

165 | 2. Seviye | simon ile eşlendin, onun puanı 115 | Kalan süre: 06:30

Figure 2.4. Stage Two. The Teller's interface. Note that there is the messaging widget that is used for communication.



onurgu Puan: 3922

Yılbaşı'nda program yapan _____ arasında en yüksek ücreti İbrahim Tatlıses aldı

insanların

Evet, tahminleri alalım:

10 | 1. Seviye | simon ile eşlendin, onun puanı 0 | Kalan süre: 05:41

Figure 2.5. Stage Two. The Guesser's interface.

indicates a factor of ten between the two numbers.

We had to implement a filter to prevent cheating using these boxes. If we recall the experience obtained from previous work, the participants in these kind of games that offer you fame and some kind of identity representation medium often try to cheat to get those awards more easily (see von Ahn and Dabbish (2004)). Thus, when creating these kind of public games, you always have to keep in mind that your game design must not allow cheating. So, we chose to limit the text that can be input in the text boxes of the teller. The filtering mechanism works like this: First it is checked whether the clue text as a whole can be found in the text of target word, if it is found, the clue is discarded. If it is not, it is checked whether the text of target word can be found in the clue text, if it is found, the clue is discarded, otherwise the clue is accepted. When the clue is discarded, it is not shown to the other user not even partly.

While the interfaces for the teller and the guesser differ generally, there is indeed a widget which is common to both of them. This widget displays the conversation between the teller and the guesser in a sequential manner. As a new guess or clue is submitted, the widget is updated.

We chose a time limit of ten minutes for this stage. This limit is intended to encourage participation in fear of not being able to complete the stage. Unfortunately, as it turned out, the time limit for this stage is set too high. This conclusion is based on the fact that -as we will see in Section 3.2.1- the average duration for stage two is about 70 seconds. Moreover, in most of the games, the duration is between 0 and 60 seconds. This means that the current level of the time limit didn't pose a challenge for the players, so this probably worked in the wrong direction: decreasing the motivation of the players.

As you might expect, this stage continues until either the time limit expires or the pair succeeds in guessing the word correctly. Regardless of the situation, we advance to the next stage. However, if they couldn't guess the target word, the whole game session finishes after the next stage.

We need to summarize the awarding mechanism for this stage. Each guess from the guesser receives 10 points. Each free text clue is awarded by giving out 5 points. However, if the clue is submitted using the clue templates, the teller earns 50 points. When the pair successfully guesses the target word, they receive 500 points.

In the third and the last stage of this game, the guesser is exposed the same question as the teller in the first stage. None of the settings differ from the first stage. Basically, the stage is designed to guarantee obtaining answers from different people for each question.

After the stage three is finished, the game session goes on with another game if the target word is guessed successfully in stage two. If the number of consequent games that were successful reaches ten, we say that the game session finishes successfully and the pair is taken back to the game lounge with a greeting note. As a result of this row of winning games, they are both awarded 5000 points. On the other hand, in case the stage two was unsuccessful, the game session is finished and they receive no points.

2.4. Questions

Before delving into the details of question generation, we must first explain the purpose of asking questions. The answers submitted by the players are utilized for disambiguating words, so it is clear that they lie in the core of our method.

As you will recall from Section 1.4, the result of a morphological analyzer is the set of all parses of that word. As seen in Table 2.2, most of the parses share a lot of morphological tags. However, this view does not directly present an understanding. We think that if the common parts are simplified by unification, it could be easier to analyze the set of parses. So the idea is to form a tree with an artificial root and parse tags as the nodes while the nodes are connected if they are consecutive in the parse. An additional rule which also performs the unification is that the children of a node must be unique. We call this tree as the parse tree of the word. The tree representation of the parse set in Table 2.2 is in Figure 2.6. To build this tree, we first create an artificial

Table 2.2. All Possible Parses of the Word “kalemi”

	Parse
1	kale[Noun]+[A3sg]+Hm[P1sg]+NH[Acc]
2	kalem[Noun]+[A3sg]+SH[P3sg]+[Nom]
3	kalem[Noun]+[A3sg]+[Pnon]+YH[Acc]

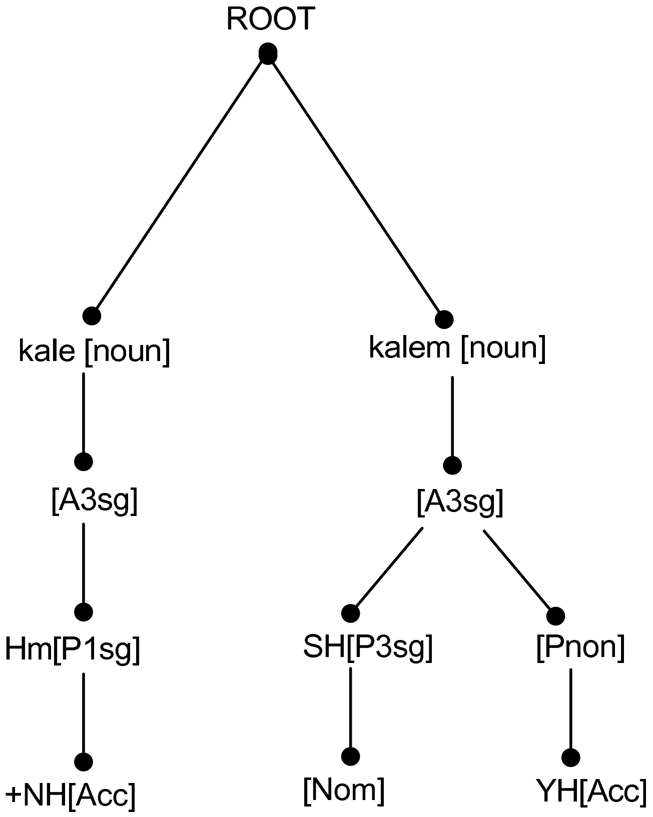
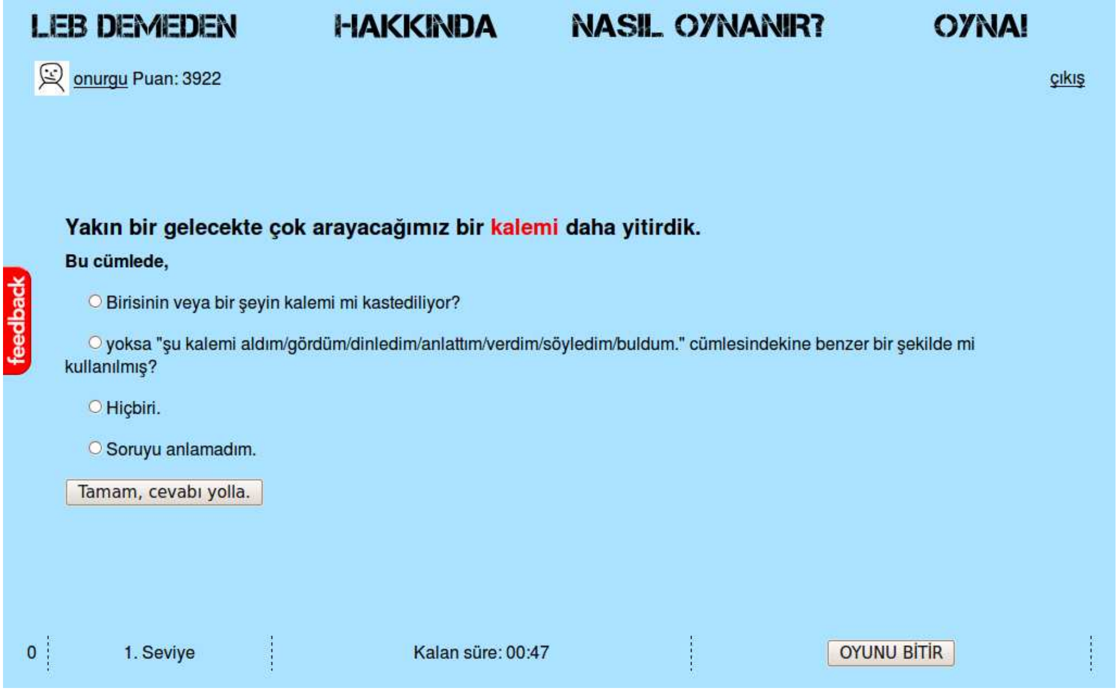


Figure 2.6. Parse Tree Example.

root. Then, we analyze the first parse tags of each parse. Discarding the duplicates, we are left with the children of the artificial root. Obviously, we attach these children to the root and continue with one of these children. Then the next parse tags of each parse that this node corresponds are collected. Like the previous, we discard the duplicates and attach the remaining. So this process continues in a recursive fashion. By design, each leaf in this tree corresponds to a different parse in the set. This means that if we can determine the path from the root to the leaf that corresponds to the correct parse, we would have solved the disambiguation problem.

As you might have understood already, the questions are for selecting the correct way in each junction along the path to the correct leaf. The information we had given up to now already describes a part of the question generation process. To generate a question, we first start with enumerating the set of all morphological parses of the word by using a morphological analyzer. We then transform it into a tree. After this transformation, we have to detect the junction points. This detection is done by the observation rules which will be described in detail in Section 2.4.1.

The detection of junction points results in abstract objects called observations. These observations are then matched with question rules. Each matched question rule is applied to the word to generate the unique questions which are tailored solely for determining the correct way to choose in the junction that is represented by the observation. A more detailed information about the question rules can be found in Section 2.4.1.



Yakın bir gelecekte çok arayacağımız bir kalemi daha yitirdik.

Bu cümlede,

○ Birisinin veya bir şeyin kalemi mi kastediliyor?

○ yoksa "şu kalemi aldım/gördüm/dinledim/anlattım/verdim/söyledim/buldum." cümlesindekine benzer bir şekilde mi kullanılmış?

○ Hiçbiri.

○ Soruyu anlamadım.

Figure 2.7. Question Example. The figure depicts a finalized question.

After this stage, the questions are said to be finalized and ready to take part in a game in the game site. There are two types of questions: yes/no questions and multi-option questions. Both of the types contains two standard options which are called ‘None’ and ‘I did not understand the question’. An example finalized question can be seen in Figure 2.7. As we have told elsewhere, the questions are asked to about 40 people. Each person submits only one answer. These answers are aggregated and the

Table 2.3. An example of Options and Resolutions

Option	Resolution Tag
1	+SH[P3sg]
2	+YH[Acc]

option with the most submissions is selected to be the final answer. We call this as the agreement answer. After determining the aggreement answer, we are ready to evaluate the question to assess whether the question is successful in choosing the correct way in the related junction. We do this by checking whether the correct parse reported in the corpus contains the resolution parse tag that is attached to each option. For this check to function, each option of each question is manually attached a parse tag which is used to select the way in the junction. For example, Table 2.3 summarizes the resolution tags for a question that is matched with the observation -coinciding with the example given in the previous figures in the section. If the outcome of the experiment is the first option, then the parse which ends with '+YH[Acc]' is selected, or in the other case the correct parse is the one that contains '+SH[P3sg]'.

We told that the standard method for choosing the final answer of a question is to select the one with the most number of submissions. However, we see that this method can be tweaked up a little bit to receive a performance increase. The modification is to discard the 'None' and 'I did not understand the question' answers if they are the most submitted ones. Inspection tells us that an increase around 10% can be achieved using this method. Further details can be found in Section 3.2.

We must note that the success of the agreement answer in selecting the right way in a junction is not sufficient for a complete disambiguation of the word in general. This is because there are usually more than one question that needs to be asked for a word. Further discussion about the percentage of the covered words in the corpus by the actual observation rules can be found in Section 4.1.

2.4.1. Observation and Question Rules

First we must explain that all types of rules defined in the system (including Question rules) are entered into the sytem using a set of specific files. This method was employed to bring flexibility in expanding the actual set of observations later. Also we thought that the simple domain specific language which is used in these files could be improved to be able to define more complex expressions for extracting observations.

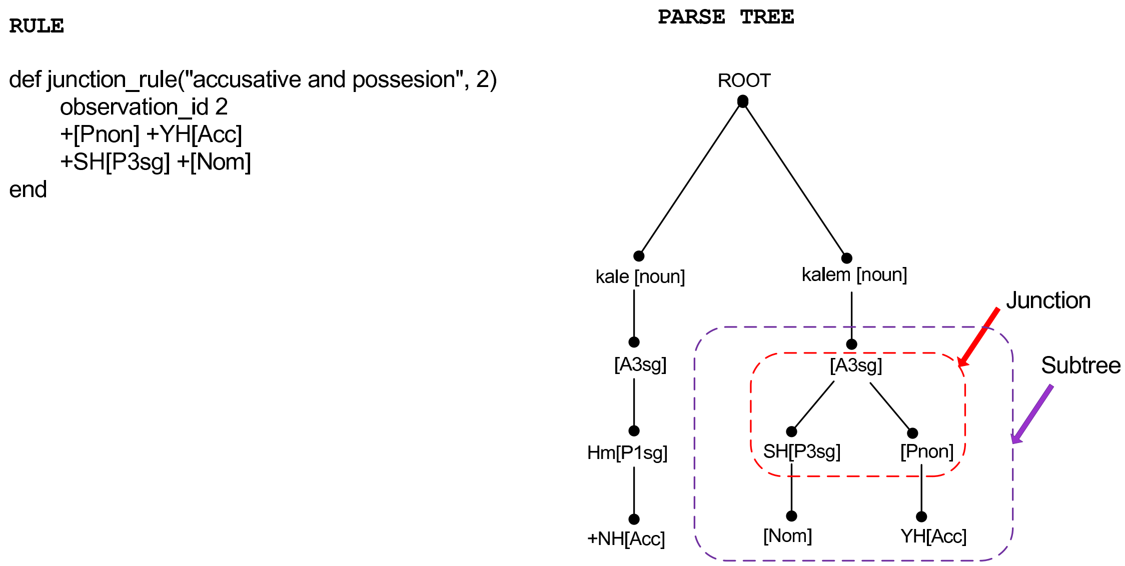
These files are structured to include a declaration at the beginning which defines the class of the rule. For example, to define a Junction rule (which is an Observation rule also), you must begin the file with a line reading:

```
def junction_rule('a short description', rule_id)
```

The declaration is similar for Question rules. The other details will explained throughout the section.

We will now look into the function of Observation rules and give some details about their definitions through rule files. There are two types of observations defined in the system. We call the first of these types as Junction rules. They are especially designed to detect a junction with two alternative ways to go. In Figure 2.8, you can find a Junction rule sample along with a junction which is detectable with this rule. The rule file for Junction rules start with the declaration as all other rule types. The next line is a declaration of the observation. Recall that observations are abstract objects that are created by observation rules to be later matched with Question rules. The next two lines defines a mini subtree. This tree has two children in the first level and the first level children all has only one child. The subtree corresponding to the mini subtree definition on the left can be found in Figure 2.8. Basically, what the Junction rule does is to check whether this subtree can be found in the parse tree of the word.

The second type of Observation rules is called Pattern rules. They are the same



```
RULE

def junction_rule("accusative and possesion", 2)
      observation_id 2
      +[Pnon] +YH[Acc]
      +SH[P3sg] +[Nom]
end
```

Figure 2.8. Junction Rule and the Corresponding Subtree.

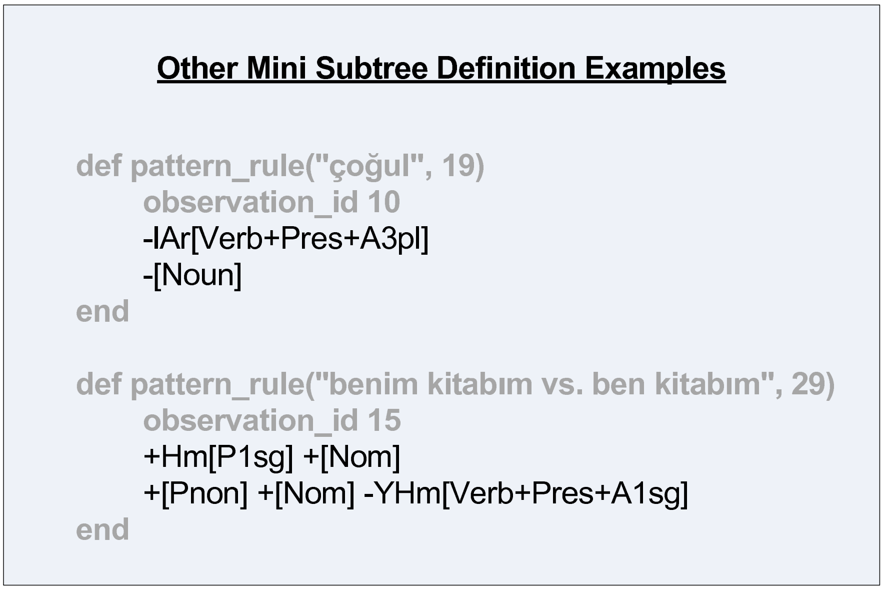
with Junction rules in all aspects except that they are more free in the definition of mini subtrees. Unlike in Junction rules, they are not constrained to having only two first level children. Also, they are not restricted in the depth of the tree. You can find some examples in Figure 2.9.

2.5. Architecture

2.5.1. Main Modules

The game logic is both implemented on the client as well as on the server side. As one can see in Figure 2.10, the client communicates with the server using only one point. In fact, the client is in constant touch with the logic on the server through periodic requests for messages which dictate the next action to perform. The server is then able to manage all the users through these messages.

Each request for the next action is directed to a module named Server Control. This module uses Game Control module to start a game session and the related games. After a game is created however, the Game Control module is bypassed and Server Control module directly accesses the relevant games to initiate a level transition or even stop them. The Game Control module is bypassed here because in any way,



Other Mini Subtree Definition Examples

```
def pattern_rule("çoğul", 19)
        observation_id 10
        -lAr[Verb+Pres+A3pl]
        -[Noun]
end

def pattern_rule("benim kitabım vs. ben kitabım", 29)
        observation_id 15
        +Hm[P1sg] +[Nom]
        +[Pnon] +[Nom] -YHm[Verb+Pres+A1sg]
end
```

Figure 2.9. Mini Subtree Definitions. Two example definitions that demonstrate subtrees with varying properties.

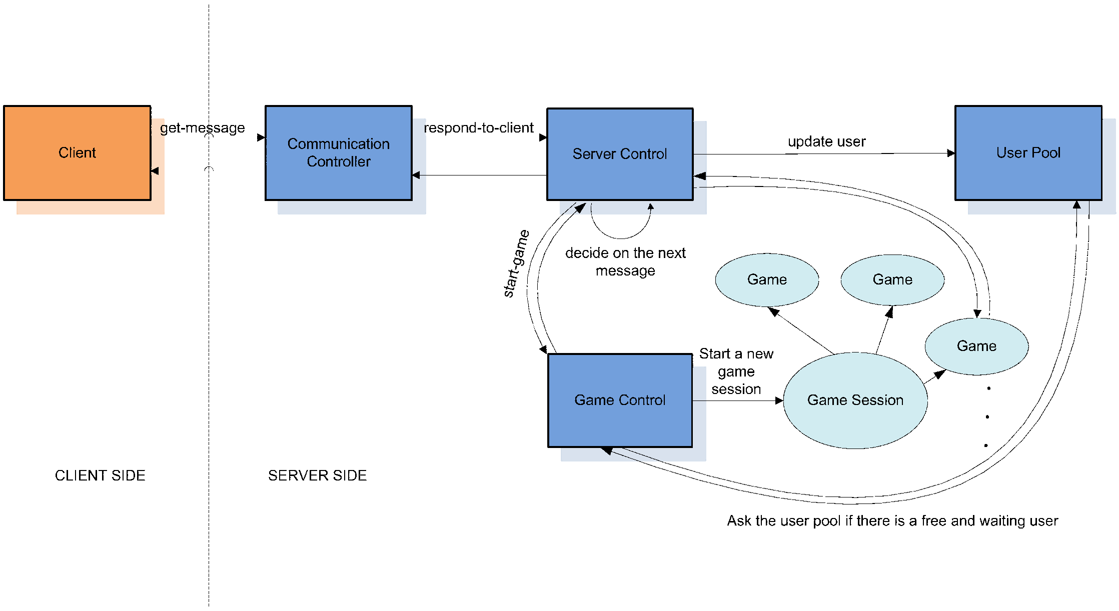


Figure 2.10. Main Modules

the logic to perform these operations is implemented in the Game class already. The intention of the player to join a game or stop one is transmitted to the Server Control through a query parameter and in turn the Server Control executes the logic for the request.

There is another module implemented to track the users. This module which is called User Pool basically keeps a list of online users and provides a function to match a user who wants to join a game with another user with the same intentions. Both Game Control and Server Control modules make use of this module.

We would like to briefly talk about the technologies we chose for the development of the game. In the client side, we employ an environment Google (2008) which translates Java code to Javascript code which can be run on most of the popular web browsers. The advantages of this mechanism is to be able to write code in an established programming language which can enjoy many auxilary tools to help the development process, such as code editors, debuggers, etc. Another positive aspect is that the translated JavaScript code is created so that it is portable to many browsers. Some widgets that can be used in GUI is also supplied by the environment. In the server-side, we employ a very popular web programming framework which was mainly valuable for us because of its model-view-controller architecture, vast number of members of the user community, the tools and methods utilized for developing and deploying a web site and finally its open nature to modification.

2.5.2. Data Structure

In this section, we will talk about the structure of the data related with the game as depicted in Figure 2.11. As you can guess, at the heart of the data structure are the words. We have an additional data model ‘Spellings’ for each spelling of a word. We also have a data model named ‘Correct Parses’ which is used to store the correct parses of each word occurence. ‘Game Template’ data model is the bridge between the static part of the data structure with the dynamic parts. It is used to connect a sentence, a word and a question together in a game. These are all handled by the

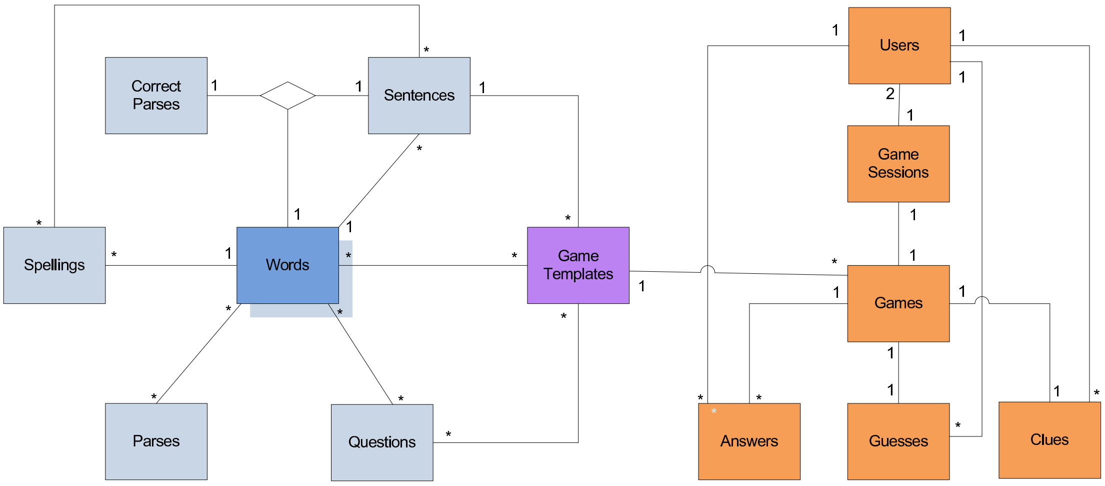


Figure 2.11. Data Structure

interrelations between ‘Words’, ‘Sentences’, ‘Questions’ and ‘Games’ data models. As you know, games are associated with game sessions. Each game is also associated with the answers submitted, guesses made and clues given during the game. So they are also related through the data models with the obvious names.

3. Experiment

In this chapter, we will be first describing the experiment setup in Section 3.1. Additionally, we will clearifying the evaluation metrics of our method. In Section 3.2, we will give the results of our experiment.

3.1. Setup

The experiment had been done through a game site which is accessible publicly on the web[4] . The details about the game site can be found in Section 2.1. The two game modes, single player and two player modes (see Section 2.2 and 2.3), were opened to public on 29th of June 2009. From then on, the site is continuing its operation.

Our contributors, namely the players, do not need to do anything further than filling out a very basic form to setup an account to access the site. This was necessary because for the game to function as desired, we had to have a way of separating the visitors. After the registration, we do not require them to do anything special. They should be using our site as they would be visiting any other site.

However, we expect them to visit the game lounge and join a game or start a new single game by themselves. In each game they play, they contribute to the experiment by answering the quiz-like questions which we discussed in Section 2.4.

As we have told earlier, the game site is online since the launch date. By design, answers are automatically aggregated enabling us to further evaluate our method. But we had to stop somewhere to begin the evaluation process. Thus, we chose to use the data collected until 9th of July 2009, approximately 6000 answers.

Before listing the results in several dimensions, we think that it is necessary to talk about the actual evaluation metrics that we will employ to assess our method.

[4] http://lebdemedenleblebi.com

3.1.1. Evaluation Metrics

As discussed in Section 2.4, for one to disambiguate a word completely, a relatively high number of questions must be answered. After speculating on the expected number of visitors, we calculated that it could be infeasible to evaluate our method on the basis of complete disambiguation.

The calculation showed us that indeed it was possible to disambiguate a number of words completely, but this would severe the playability of the game. The reason for that damage was that because in that case we should be exposing a single player with the same word and the same sentence in a row several times. This would obviously be annoying, if not boring. So fearing that this would even decrease the degree of participation, we chose to evaluate our method by assessing the individual question qualities themselves.

Thus, we introduced an experiment plan. By sticking to that plan, we would be collecting about 30 answers for each question type without damaging the playability of the game itself. Please note that there are currently a total of 74 different type of questions and with them we are able to completely disambiguate 58% of the total word occurences in the corpus. Further discussion about the current coverage rates and additional hypothetical rates can be found in Section 4.1.

3.2. Results

As told in Section 3.1.1, we evaluated our method on the quality of the automatically generated questions. For this, we created two different sets of questions. We will refer to them as Phase 1 and Phase 2. They both contain 74 questions that correspond to different observations. This is equal to the number of observations which can be detected in the game currently. So in one phase, we cover all of the question types that can be generated by the game. We call a question successful when the answer aggreement of that question resolves into a parse tag which can be found in the correct parse of the relevant word occurence in the corpus.

Our aim is to analyze the data from an eleven day period beginning with the launch (between 29^{th} of June and 9^{th} of July) and to report the success rate over all question types. We assume that the rate of success calculated here will be replicated throughout the portions of the corpus where the current question rules cover. Our second aim is to perform a qualitative analysis of the unsuccessful questions. By doing that, we hope to achieve some important conclusions about the preparation method of the questions. You can find the discussion in Section 4.2.

For calculating the success rate of the question types, we prepared the following two tables: Table 3.1 and 3.2. They each contain the evaluation results of a particular phase. The column named 'Success' indicates whether this question was successful or not. If that column reads 'NAN', then this means that the agreement answer did not resolve into a parse tag, so the evaluation mechanism could not check whether it is found in the correct parse of the word. However, this is effectively a failure case. The next two columns named '#NP' and '#NPS' indicate the number of all possible parses of that word and the number of parses that are left after discarding the others respectively. Therefore, we can calculate the increase in the base probability of disambiguation. The last two columns show the base probability before the question is asked and the increase in this probability in percents respectively.

From these tables, we calculate the rate of successful questions in the first phase as 79.7%. This figure is realized as 71.6% in the second phase. However, we want to report that a little modification to the definition of a successful question would increase this values to 87.8% and 79.7%. This modification would be to discard the answers of type 'None' or 'I did not understand the question' if they are the highest ones. We observed that this modification increases the rates but in any way we did not change the evaluation method so that to allow an elaboration.

When we look at the combined results of these two phases, we see that the percentage of question types that are successful in both of these phases is 63.5%. We will be talking about these figures and their meaning in evaluating the performance of our method over the whole corpus in 4.1.

Table 3.1: Results of the first phase - #P: Number of Possible Parses of that word, #PS: Number of Parsers Selected after evaluating this question, BP: Base Probability, BPΔ: Base Probability Increase

Template No	Word	Condition	Success	#P	#PS	BP	BPΔ
280	listenin	1	OK	4	2	0.25	100%
55	başkanı	2	OK	6	4	0.17	50%
75	hareketini	3	OK	2	1	0.5	100%
40	yönetenler	4	OK	4	2	0.25	100%
1	savunma	5	OK	4	3	0.25	33%
131	servetlerine	7	OK	16	4	0.06	300%
86	isimleri	8	FAIL	7		0.14	0%
50	sırp	9	OK	3	1	0.33	200%
7	askerlikten	11	FAIL	8		0.13	0%
139	umduğu	12	OK	4	1	0.25	300%
344	esastır	13	NAN	8		0.13	0%
36	gelecek	14	OK	14	1	0.07	1300%
297	ediyorum	15	NAN	5		0.2	0%
144	ordumuz	16	OK	6	4	0.17	50%
24	komutanların	17	OK	8	4	0.13	100%
77	kimileri	18	OK	9	2	0.11	350%
414	verilmelidir	19	OK	12	7	0.08	71%
45	halkın	20	OK	4	2	0.25	100%
367	işlemektedir	21	NAN	11		0.09	0%
1961	gördük	22	OK	4	1	0.25	300%
1907	okumamışlardır	23	OK	10	8	0.1	25%
475	sanayileşmeden	24	OK	5	2	0.2	150%
1861	görebileceğiniz	25	FAIL	9		0.11	0%
598	deniz	26	FAIL	6		0.17	0%
1068	yitirdikçe	27	OK	2	1	0.5	100%

Table 3.1: Results of the first phase (continued)

Template No	Word	Condition	Success	#P	#PS	BP	BPΔ
67	açıklamada	28	FAIL	7		0.14	0%
16	alınan	29	NAN	8		0.13	0%
247	katılacağı	30	OK	14	7	0.07	100%
2352	değerlendirilmesi	31	OK	6	2	0.17	200%
2081	yaratıyor	32	OK	10	5	0.1	100%
266	kaldırıldıktan	33	NAN	4		0.25	0%
62	bir	34	OK	8	1	0.13	700%
90	bu	35	OK	9	4	0.11	125%
71	de	36	OK	4	1	0.25	300%
187	için	37	OK	12	3	0.08	300%
118	ile	38	OK	7	1	0.14	600%
200	çok	39	OK	9	4	0.11	125%
889	o	40	OK	13	9	0.08	44%
538	vardır	41	OK	9	5	0.11	80%
645	gibi	42	OK	6	3	0.17	100%
1057	ona	43	OK	15	13	0.07	15%
310	en	44	OK	3	1	0.33	200%
1115	sonra	45	OK	6	3	0.17	100%
981	ama	46	OK	3	1	0.33	200%
976	kadar	47	OK	6	1	0.17	500%
1020	değil	48	OK	3	2	0.33	50%
933	son	49	OK	6	4	0.17	50%
207	bulunacak	50	NAN	16		0.06	0%
108	ise	51	OK	3	1	0.33	200%
1512	o	52	OK	13	4	0.08	225%
273	yıl	53	OK	3	2	0.33	50%
1230	yeni	54	OK	8	4	0.13	100%
113	iki	55	OK	6	4	0.17	50%

Table 3.1: Results of the first phase (continued)

Template No	Word	Condition	Success	#P	#PS	BP	BPΔ
323	türkiye	56	OK	3	1	0.33	200%
359	insan	57	OK	6	4	0.17	50%
2027	ben	58	OK	6	2	0.17	200%
234	bakanları	59	OK	14	7	0.07	100%
899	ancak	60	OK	2	1	0.5	100%
1453	belirtildi	61	OK	2	1	0.5	100%
1516	diye	62	OK	5	3	0.2	67%
102	arasında	63	NAN	6		0.17	0%
255	ilk	64	FAIL	7		0.14	0%
402	önce	65	FAIL	8		0.13	0%
299	önünde	66	OK	8	4	0.13	100%
656	iyi	67	OK	7	4	0.14	75%
465	yüzde	68	OK	10	4	0.1	150%
155	göre	69	OK	5	3	0.2	67%
443	özel	70	OK	5	4	0.2	25%
372	etmiştir	71	OK	14	7	0.07	100%
378	dışında	72	OK	8	4	0.13	100%
2232	yok	73	OK	10	2	0.1	400%
558	yer	74	OK	8	2	0.13	300%
220	karşı	75	OK	10	3	0.1	233%
921	ya	76	NAN	5		0.2	0%
	# of Correct Disambiguations		59				
	# of Incorrect Disambiguations		7				
	# of Questions with an indeterminant answer		8				

Table 3.2: Results of the second phase - #P: Number of Possible Parses of that word, #PS: Number of Parsers Selected after evaluating this question, BP: Base Probability, BPΔ: Base Probability Increase

Template No	Word	Condition	Success	#P	#PS	BP	BPΔ
392	ülkenin	1	OK	8	2	0.13	300%
46	fikri	2	FAIL	8		0.13	0%
158	eşini	3	OK	4	3	0.25	33%
41	yönetilenler	4	OK	4	2	0.25	100%
63	açıklama	5	OK	4	3	0.25	33%
184	çocuklarından	7	FAIL	12		0.08	0%
78	kimileri	8	NAN	9		0.11	0%
37	nihayet	9	NAN	4		0.25	0%
2	bakanlığı	11	FAIL	12		0.08	0%
52	umduğu	12	NAN	4		0.25	0%
146	tarafsızdır	13	NAN	10		0.1	0%
167	olacak	14	OK	14	5	0.07	180%
491	çağırıyorum	15	NAN	5		0.2	0%
57	ordumuz	16	OK	6	4	0.17	50%
100	ünlüler	17	OK	17	10	0.06	70%
85	isimleri	18	OK	7	2	0.14	250%
513	edilmeli	19	OK	9	5	0.11	80%
8	için	20	NAN	12		0.08	0%
401	yapmaktadır	21	OK	11	7	0.09	57%
2099	geçirdik	22	OK	4	1	0.25	300%
1917	haftalardır	23	OK	4	1	0.25	300%
1852	konmadan	24	OK	5	1	0.2	400%
2791	uygulayacağınız	25	FAIL	9		0.11	0%
1181	gazeteniz	26	OK	3	2	0.33	50%
4794	gittikçe	27	OK	3	1	0.33	200%

Table 3.2: Results of the second phase (continued)

Template No	Word	Condition	Success	#P	#PS	BP	BPΔ
476	sanayileşmeden	28	FAIL	5		0.2	0%
497	alınmak	29	NAN	4		0.25	0%
238	katılacağı	30	OK	14	7	0.07	100%
2590	değerlendirmemiz	31	OK	9	3	0.11	200%
2288	yarattık	32	OK	8	4	0.13	100%
865	kaldırılabilir	33	NAN	10		0.1	0%
69	bir	34	OK	8	1	0.13	700%
26	bu	35	OK	9	4	0.11	125%
76	de	36	OK	4	1	0.25	300%
195	için	37	OK	12	3	0.08	300%
230	ile	38	FAIL	7		0.14	0%
385	çok	39	OK	9	1	0.11	800%
982	o	40	OK	13	9	0.08	44%
876	var	41	OK	7	4	0.14	75%
1135	gibi	42	OK	6	3	0.17	100%
890	o	43	OK	13	9	0.08	44%
322	en	44	OK	3	1	0.33	200%
267	sonra	45	OK	6	3	0.17	100%
1527	ama	46	OK	3	1	0.33	200%
996	kadar	47	OK	6	1	0.17	500%
1028	değil	48	OK	3	2	0.33	50%
1008	son	49	NAN	6		0.17	0%
270	bulunan	50	OK	8	4	0.13	100%
112	ise	51	OK	3	1	0.33	200%
1627	o	52	OK	13	4	0.08	225%
97	yılın	53	OK	6	4	0.17	50%
1359	yeni	54	OK	8	4	0.13	100%
911	iki	55	OK	6	4	0.17	50%

Table 3.2: Results of the second phase (continued)

Template No	Word	Condition	Success	#P	#PS	BP	BPΔ
368	türkiye	56	OK	3	1	0.33	200%
975	insanlar	57	OK	7	3	0.14	133%
3259	beni	58	OK	5	1	0.2	400%
1445	bakanlar	59	OK	7	3	0.14	133%
969	ancak	60	FAIL	2		0.5	0%
1572	belirtti	61	OK	2	1	0.5	100%
1841	diye	62	NAN	5		0.2	0%
306	arasındaki	63	OK	3	1	0.33	200%
924	ilk	64	OK	7	4	0.14	75%
430	önce	65	OK	8	3	0.13	167%
403	önce	66	NAN	8		0.13	0%
785	iyi	67	OK	7	4	0.14	75%
483	yüzde	68	OK	10	4	0.1	150%
17	göre	69	OK	5	3	0.2	67%
527	özelleştirmedeki	70	NAN	2		0.5	0%
136	ettiler	71	OK	2	1	0.5	100%
330	dışında	72	NAN	8		0.13	0%
2235	yok	73	OK	10	2	0.1	400%
1162	yer	74	OK	8	2	0.13	300%
960	karşı	75	OK	10	3	0.1	233%
1636	ya	76	NAN	5		0.2	0%
	# of Correct Disambiguations		53				
	# of Incorrect Disambiguations		7				
	# of Questions with an indeterminant answer		14				

3.2.1. Statistics about the Game Play

In Table 3.3, we give statistics about the usage rates of all the units in our method. Before looking at the results, we would like to report that there are about 400 registered users at the end of the experiment period. As you will recall, the user contributes by submitting answers, clues and guesses. Therefore, we think it is useful to report the total and the average quantities that are important when evaluating the overall participation rate of the system. In the first row, we see that the average number of answers over all users (U) is a bit smaller than the value over only the users who submitted at least one answer (CU). This means that some users never submitted answers. This is normal because in all systems that does not require some kind of agreement or responsibility, there will be always people who just register to just check the system out. However, when we compare this difference with the amount of difference in guesses and clues, we see that the latter difference is very much higher in both of them. This is due to the relatively low number of two player games played. Recall that guesses and clues are submitted only in two player games. This can be cross-checked from Table 3.4.

Table 3.3: General Game Statistics. U: User, CU: Contributed User, ADCT: Average Number of different clue types by a user who contributed a clue

Per	U	CU	Game	Game Session	Day	ADCT	Total
Avg. # of Answers	15.33	17.13	1.19	9.68	570.18	NAN	6272
Avg. # of Guesses	3.73	20.87	0.29	2.35	138.55	NAN	1524
Avg. # of Clues	2.74	25.5	0.21	1.73	102	3.31	1122
Avg. # of Games	12.93	NAN	NAN	8.16	480.73	NAN	5288

Table 3.4: Game Mode Breakdown

	Per User	Total
Single	11.7	4784
Two players	1.23	504

One interesting thing to note that although a small percentage of users contributed by submitting clues, they did so by using the clue templates although with somewhat low frequency. This can be seen in the ADCT column of Table 3.3 along with a usage frequency of clue templates in Table 3.5. We can see that the most used template is the free text input template. The next one -though a lot smaller- is the clue template 0 which is described in Table 2.1. The remaining entries all have a value about 20. We think that this means that the remaining clue templates did not receive much attention.

Table 3.5: Clue Type Breakdown. This table lists the total number of clues with a specific clue type.

Type	Frequency
99	920
0	94
1	23
2	20
3	9
4	10
5	16
6	12
7	18

When we look at Figure 3.1, we see that most users spent under a minute per game. However, there is also a considerable number of users who spent more than one minute per game. These longer times probably originate from two player games. Thus, given that the number of two players are low, we see the curve concentrated under one minute. This conjecture strengthens when the Table 3.6 reads around 30 seconds of average time for answering a question.

Table 3.6: Time Durations (in seconds). AGD: Average Game Duration. AGSD: Average Game Session Duration. ATU: Average Amount of Time a user spent on a game. ATFA: Average Time Required for Answering a question.

	AGD	AGSD	ATU	ATFA
Duration	36.7	372.8	39.74	36.17

The distribution observed in Figure 3.1 can be also seen in Figure 3.2. However, in this figure there is a more fractured plot. Most of the users clearly spent a total time of under 550 seconds. This coincides with the average number of games per user and the average time duration of a game figures from Table 3.3 and 3.6. The actual values are 12.93 and 36.7, when we multiply them we get 474.5 which is around 550 as expected.

We plotted the graph in Figure 3.3 to observe the distribution of time spent on stage two. Like the previous two figures, the curve is exponentially decreasing. However, this is a characteristics of contribution based systems. We notice that there are only a few users behind 250 seconds. This proves that our earlier prediction that the time limit for stage two is set to an amount higher than the ideal one. Having said that, we are not surprised by the low number two player games anymore. We read from the graph that the ideal limit should be below 50, for example 30 seconds. This way, the stage two would be a real challenge for most of the users. So they would be more motivated to play two player games.

Referring to Table 3.6, the average time required to answer a question is 36.17. However, this is averaged over all questions. We think that it is helpful to calculate this average over all question types. We did this calculation to see that the question type which required the most time took 87 seconds to solve on average. The minimum of this figure is 19.17. Further investigating, we marked the question types which belong to

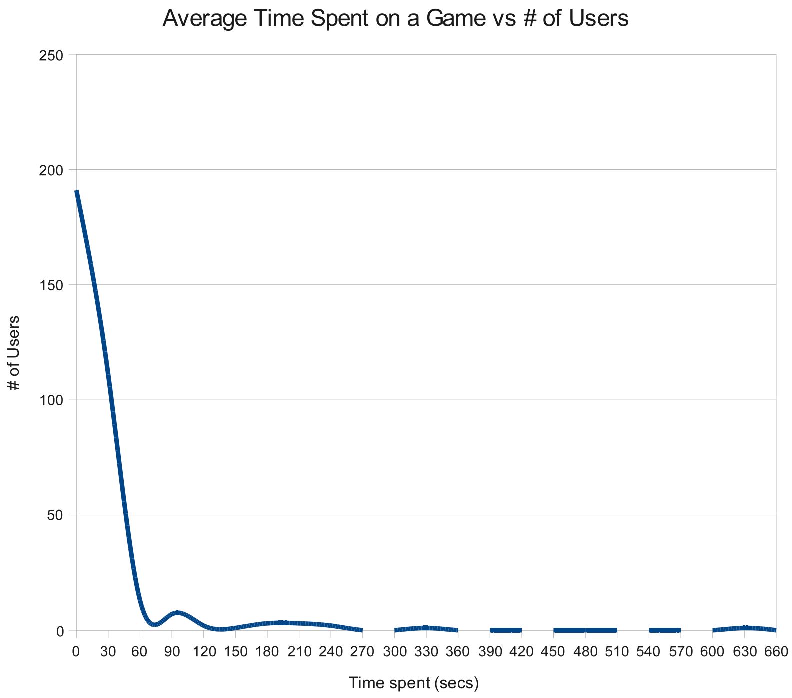


Figure 3.1. Average Time Spent on a Game by a User

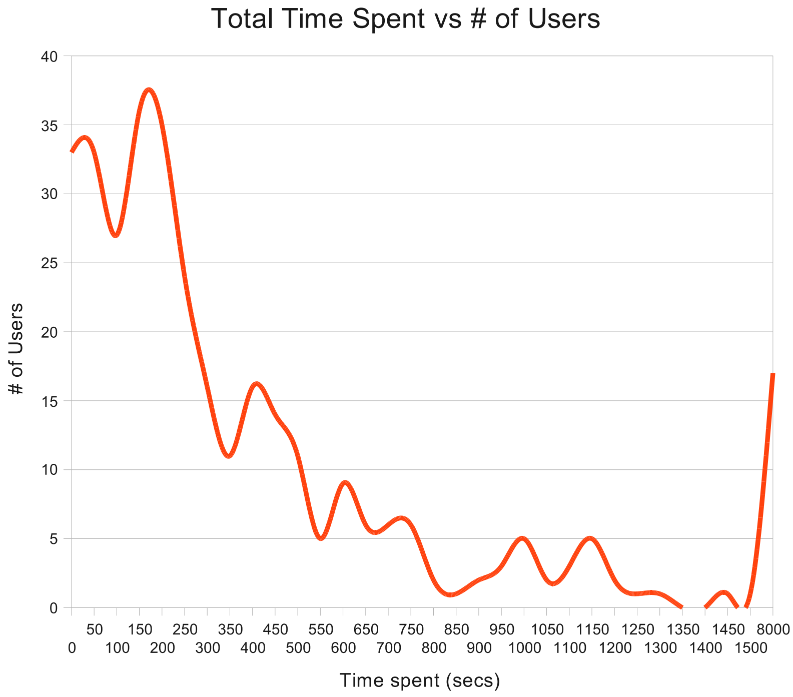


Figure 3.2. Total Time Spent vs. Users

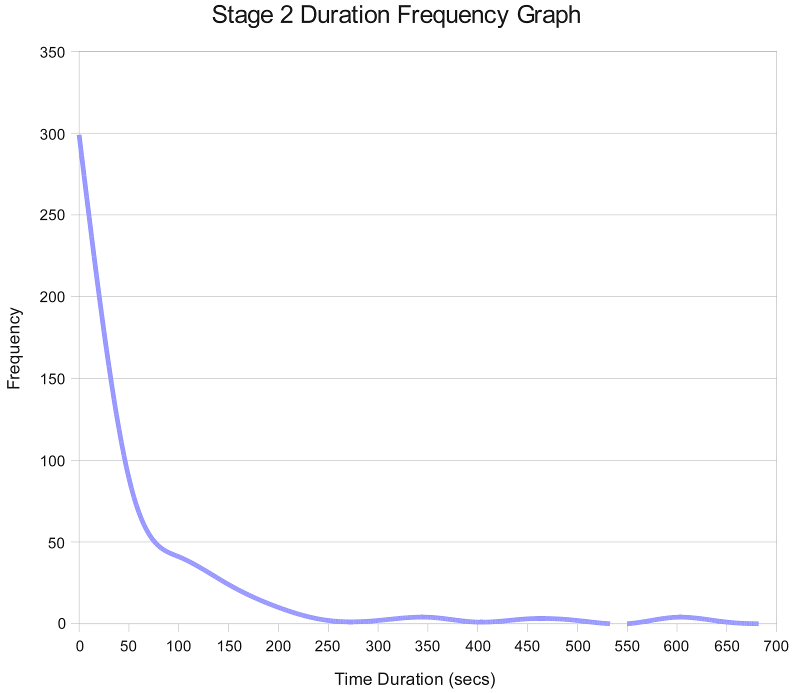


Figure 3.3. Stage Two Duration Frequency Graph: An average stage two instance took 71.2 seconds

unsuccessful questions and produced the graph in Figure 3.4. This is indeed a spectrum of all question types, from the least time required to the most time required. The yellow stripes indicate the question types which correspond to unsuccessful questions. After briefly analyzing that, we see a slight correlation between the time required to solve and the question performance. Clearly, the unsuccessful questions tend to take more time to solve than the successful questions.

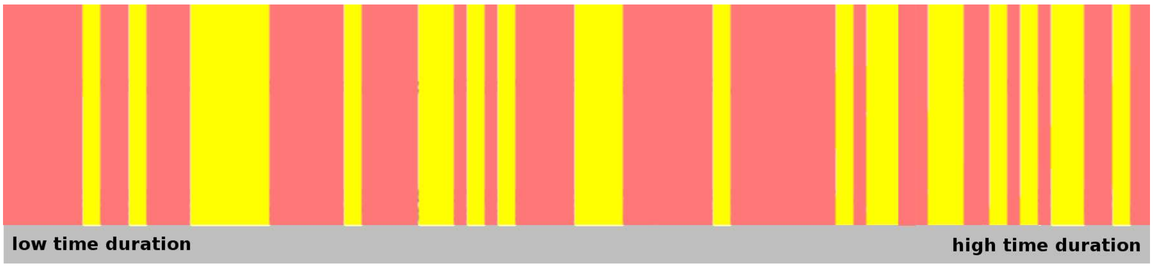


Figure 3.4. Question Type Spectrum

In this graph, question types are ordered by the corresponding time required to answer on average. The yellow stripes indicate the question types which failed in at least one phase.

4. Discussion

4.1. Morphological Disambiguation with Our Method

In this section, we will be elaborating on the success rate we report in Section 3.2. Recall that the overall percentage of successful questions is 63.5%. However, this value is calculated over only two phases. This is a hinderance to the evaluation of our method but nevertheless we think that we can at least estimate an overall performance of our method. Also, these success rates do not indicate the success rate of an actual morphological disambiguation. Thus, we will be looking at the reasons behind this drawback and calculate the requirements to get over it. Additionally, we will be speculating about an estimated success rate of morphological disambiguation that our method will achieve over the whole corpus if we were to have sufficent data.

We begin with reporting that even though that the combined success rate is 63.5%, the figures for the first and second phases are 79.7% and 71.6% respectively, indicating a potential of higher rates if we could be able to test our method further with additional data. In fact, the figures go higher to 87.8% and 79.7% with a slight modification to the evaluation mechanism. In the case of modified mechanism, we would be discarding the 'None' and 'I did not understand the question' answers if they are the most frequent among the answers submitted. The rationale here is that even if the majority got confused about the question, the other people who submitted a valid answer may have had no problems. Thus, looking at the distribution after discarding these two types of answers may be a good idea. We see that indeed it is when we observe that the increase in the success rate is obvious.

We report a success rate over all possible question types by analyzing two instances of each of them. However, we can only report an estimation of the success rate over the whole corpus. To calculate that estimate, we have to first calculate the amount of corpus we cover with the current possible questions.

To calculate the amount of corpus we cover, we first have to define what is a covered word and then report the percentage of covered words over all the words in the corpus. The definition of a covered word requires the reader to recall the question generation process. If we refer to Section 2.4, we will remember that the question generation process works as a succession of events. First, we analyze the word to enumerate all of the possible parses of it. Then these parses are combined to build a tree out of them. Next, this tree is analyzed to find the junction points and the patterns. The junction points and patterns are actually sets of tokens, so the term token sets. After detecting the features, we build the questions for the word by using these features. Basically, the feature id is looked up in the question types table and the corresponding question template is selected. The question generated is served to the players to get answers. After collecting a sufficient number of answers, it is possible to analyze the answers and determine the aggreement answer. The aggreement answer is further used to decide on the set of parses to discard. Thus a number of parses are discarded for that word. But this may not result in a direct answer always. We may have to generate several questions and process them like the previous one to leave only one parse as the final parse. However, in cases where we do not have a feature which corresponds to the token set observed in that junction point or pattern, we may not be able to generate all the questions required to leave a final parse. We call such words as uncovered. Thus, a covered is a word in the corpus where all of the questions required for completely disambiguating the alternative morphological parses of that word.

So based on this definition, we prepared the table in Figure 4.1. Each section of the table contains the number of uncovered word occurrences and their percentage over the whole corpus for the indicated number of token sets. The first one which is titled ‘Actual - 69’ reports the coverage rate with the current token sets recognized by the system. The others represent the hypothetical systems with the indicated number of token sets. They help to speculate on the cost of reaching a desired coverage rate over the corpus. The cost is basically the time required to produce the features that react to the remaining token sets. By the way, as we mention in Section 2.4, we compiled the current set of features after analyzing all the tokens that are observed throughout the corpus. Thus, the actual set of token sets which are recognized by the system consists of token sets which seem reasonable to include to cover the most number of words without spending an enormous time. This decision was crucial to be able to advance in the course of the thesis process. Nevertheless, the remaining token sets can be included in the system after a similar work.

After analyzing the first section of the table, we see that 58.3% of the words in the corpus are covered by the actual system. We also include the words with only one parse in this value to get a real impression of the capability of our method. This figure means that if we had been able to collect sufficient data, we would be able to resolve all the required questions for 58.3% of the corpus. Thus, after determining the agreement answers, we would be able to come up a success rate of complete morphological disambiguation. However, we can estimate that the percentage of successful disambiguations will be at least 63.5% of the part that is completely covered by the actual system. This corresponds to a rate of 37.0% for successful complete morphological disambiguation over the whole corpus.

This could seem very poor especially when coupled with the current rate of collected answers. In fact, if the system were to operate with the rate of answers in the first week, it would take 46 years to achieve the success rate reported above. However, the situation starkly changes when we calculate the same figure if we were to host the game on a site which attracts a lot of visitors who in general has a lot of spare time. For example, if we take the site of a major nation-wide newspaper to be visited by about 50000 visitors daily, the figure goes down to only 0.2 years, in other words two and a half months. Thus, in its very basic sense, we can see our method as a fair bootstrapping method even with the current set of token sets.

We continue by noting that the previous reported success rate would increase to 44.8% and 54.6% if we only added 40 and 400 token sets into the current set of token sets respectively. This indicates that with a relatively small amount of work we achieve a fair amount of increase in the success rate. As expected, further improvement is observed if we continue adding token sets which can be observed in Figure 4.1.

TOKEN SETS																				
Actual – 69			100			500			1000			2000			5000			ALL TOKEN SETS		
#UTS	#UUW	#UWO	#UTS	#UUW	#UWO	#UTS	#UUW	#UWO	#UTS	#UUW	#UWO	#UTS	#UUW	#UWO	#UTS	#UUW	#UWO	#UTS	#UUW	#UWO
0	20799	158020	0	30952	262199	0	37988	389695	0	42870	439855	0	48849	476288	0	57092	502326	0	60949	507254
1	13185	113117	1	19912	176713	1	16545	87842	1	13145	50370	1	8864	23158	1	2787	3589			
2	4055	43551	2	792	3746	2	56	341	2	50	60	2	5	6	2	1	1			
3	1598	7263	3	234	519	3	6	6	3	4	4									
4	1515	8029	4	177	242	4	3	3	4	3	3									
5	821	3193	5	67	75	5	1	1	5	1	1									
6	368	1526	6	17	27															
7	277	721	7	17	24															
8	64	178	8	4	4															
9	179	375	9	6	6															
10	99	261	12	1	1															
Other.	205	517	15	1	1															
NAN	17784	500770	NAN	8769	393964	NAN	6350	359633	NAN	4876	347228	NAN	3231	338069	NAN	1069	331605	NAN	38430	330267
UW1P	38430	330267	UW1P	38430	330267	UW1P	38430	330267	UW1P	38430	330267	UW1P	38430	330267	UW1P	38430	330267	UW1P	38430	330267
T#UUW	40150		T#UUW	29997		T#UUW	22961		T#UUW	18079		T#UUW	12100		T#UUW	3857		T#UUW	0	
T#CUW	20799		T#CUW	30952		T#CUW	37988		T#CUW	42870		T#CUW	48849		T#CUW	57092		T#CUW	60949	
T#UW	99379		T#UW	99379		T#UW	99379		T#UW	99379		T#UW	99379		T#UW	99379		T#UW	99379	
%UWC/WO1	34.1%		%UWC/WO1	50.8%		%UWC/WO1	62.3%		%UWC/WO1	70.3%		%UWC/WO1	80.1%		%UWC/WO1	93.7%		%UWC/WO1	100.0%	
%UWC/W1	59.6%		%UWC/W1	69.8%		%UWC/W1	76.9%		%UWC/W1	81.8%		%UWC/W1	87.8%		%UWC/W1	96.1%		%UWC/W1	100.0%	
T#WO	837521		T#WO	837521		T#WO	837521		T#WO	837521		T#WO	837521		T#WO	837521		T#WO	837521	
%CWO/WO1	31.2%		%CWO/WO1	51.7%		%CWO/WO1	76.8%		%CWO/WO1	86.7%		%CWO/WO1	93.9%		%CWO/WO1	99.0%		%CWO/WO1	100.0%	
%CWO/W1	58.3%		%CWO/W1	70.7%		%CWO/W1	86.0%		%CWO/W1	92.0%		%CWO/W1	96.3%		%CWO/W1	99.4%		%CWO/W1	100.0%	
%PCWO/WO1	66.4%		%PCWO/WO1	87.4%		%PCWO/WO1	94.2%		%PCWO/WO1	96.7%		%PCWO/WO1	98.5%		%PCWO/WO1	99.7%		%PCWO/WO1	100.0%	

#UTS: # of uncovered token sets. #UUW: # of Uncovered Unique Words. #UWO: # of Uncovered Word Occurences. UW1P: Unique words with only one parse. T#UUW: Total # of Uncovered Unique Words. T#CUW: Total # of Completely Covered Unique Words. T#UW: Total # of Unique Words. %UWC/WO1: % of Unique Word Coverage (without words with single parses). %UWC/W1: % of Unique Word Coverage (including words with single parses). T#WO: Total number of word occurences in the corpus. %CWO/WO1: % of Completely Covered Word Occurences (without words with single parses). %CWO/W1: % of Completely Covered Word Occurences (including words with single parses). %PCWO/WO1: % of Partially Covered Word Occurences (without words with single parses).

Figure 4.1. Corpus Coverage. This figure presents the actual and the hypothetical level of corpus coverage in differing numbers of token sets

4.2. Assessment of Questions

In this section, we will be briefly elaborating on the unsuccessful questions in both phase one and phase two.

After examining the sentence and the question that correspond to the observation 28, we see that the players are directed in the false direction by including an example which is very much the same with the one in the sentence (Figure 4.2 - Question 1). The player is asked to compare one usage with another, but it is not clear which example represents the definition in the related option.

In the question that is triggered with the observation 64 (Question 2 in Figure 4.2), we see that a significant number of people selected the right answer (first option), but more people chose the option which clearly asks whether the "ilk" was used to indicate whether an action is taken for the first time (third option) which is indeed the exact situation in the sentence. However, the third option questions whether the word is an adverb or not, so it is falsely tagged as an adverb instead of adjective.

In Question 3 in Figure 4.2, it is asked whether the target word carries the second person plural possesion marker. Almost all players answered 'No' which is the correct one. But the question is faulty in assuming that the other alternative is the only possible outcome left. However, the target word is indeed a root word in its own sense. So the question fails.

In the Question 4, the question is not asked clearly. When we look at the most frequent answer which is 'No', it makes sense. But in fact the question was trying to ask whether the word is marked by a second person plural possession marker or not. So the question falsely expected people to answer 'Yes' to this question. This assumption is done because while designing the question, it was seen that the most frequent context was that. However, this makes the question irrelevant in this sentence.

Although the Question 5 is clear for this sentence, the aggreement answer is

QUESTION 1

WORD	açıklamada
SENTENCE	Açıklamada , Cumhurbaşkanına " destek " işareti veren bir ifade yer almadı
QUESTION	"açıklamada" kelimesi,
Option 1	Bir yerde oluş halini mi anlatıyor?
Option 2	Yoksa bir şey yapmakta olmayı mı anlatıyor? Örn. "Açıklamada yer alan ifadeler" ile "Yapmada olduğunuz şeyler" ifadelerini karşılaştırın.

Answer Statistics	Option	Frequency
	2	24
	1	16
	98	16
	99	9

QUESTION 2

WORD	ilk
SENTENCE	Cumhuriyet tarihinde ilk kez TSK'ya ait arazi ve sosyal tesisler satışa çıkarılıyor
QUESTION	Bu cümledeki "ilk" kelimesi,
Option 1	"Kapıdan çıkan ilk kişi o kız oldu." cümlesindeki gibi sıfat olarak mı kullanılmıştır?
Option 2	Yoksa "Bu kurum hep ilklere imza atmasıyla ünlüdür." cümlesindeki gibi isim olarak mı kullanılmıştır?
Option 3	veya "Bunu ilk yapıyorum." cümlesindeki gibi eylemin ilk olarak yapıldığını mı gösteriyor?

Answer Statistics	Option	Frequency
	3	18
	1	17
	98	3
	2	2

QUESTION 3

WORD	deniz
SENTENCE	Türkiye , kara , deniz ve hava taşımacılığından kazandığı dövizi artırmalıdır
QUESTION	"deniz" kelimesi 'kalem' kelimesinin "sizin kaleminiz" tamlamasındaki kullanımına mı benziyor?
Option 1	Evet
Option 2	Hayır

Answer Statistics	Option	Frequency
	2	45
	1	2
	99	1

QUESTION 4

WORD	görebileceğiniz
SENTENCE	Tek tek kağıda dökün ve evinizde ya da işyerinizde her gün görebileceğiniz bir yere asın
QUESTION	"görebileceğiniz" kelimesi 'kalem' kelimesinin "sizin kaleminiz" tamlamasındaki kullanımına mı benziyor?
Option 1	Evet
Option 2	Hayır

Answer Statistics	Option	Frequency
	2	33
	1	7
	99	2

QUESTION 5

WORD	ancak
SENTENCE	Ancak Soğuk Savaş'ın sona ermesiyle , değişme başladı
QUESTION	"ancak" kelimesi,
Option 1	"Öyle diyorsun ancak peki ya buna ne demeli?" cümlesindeki gibi iki cümleyi bağlamak için mi kullanılmıştır?
Option 2	Yoksa "Şimdi çıkarsan ancak yetişirsin." cümlesindeki gibi "ucu ucuna, yeteri kadar" anlamında mı kullanılmıştır?

Answer Statistics	Option	Frequency
	2	14
	98	12
	1	10
	99	3

Figure 4.2. Question Assesment. Option code 98 represents the ‘None’ answers while the code 99 represents ‘I did not understand the question’ answers.

wrong. However, we see that the figures for all options other than the ‘I did not understand the question’ is very close to each other.

4.3. Game Design Remarks

As we noted in Section 4.1, the time required to completely disambiguate the covered parts of the corpus is virtually forever if the number of visitors to our site did not increase. Unfortunately, this situation got worser after two weeks. In fact, currently we only get five visitors a day in a week[5] . This shows that we failed to create addicted users. We made a few observations regarding this problem throughout the experiment and during the preparation of this document to report the results. We would like to list them for a discussion of the problem.

First of all, the sentences that contain the words that we ask questions about are too old for them to be interesting today. They are compiled from online newspapers which date back to 1997. Additionally, the topics are too diverse to build some motivation to learn about by the reader. Our opinion on this subject is also backed up by user feedback. A considerable amount of feedback was complaining about the ‘strange’ nature of the sentences.

Second, the time limit for the second stage is a much longer than the optimum. As discussed in Section 3.2.1, while the average time spent on stage two was about 70 seconds, the time limit of this section is ten minutes. After analyzing Figure 3.3, it is clear that this limit must be as low as 30 seconds for the two player game to be challenging enough. The same observation goes for the time limit allowed for answering a question which was two minutes. While we see that the average time for answering a question took only 36.17 seconds from Table 3.6. Clearly, like the time limit in stage two, this does not pose a challenge for the players. Thus, one of the reasons for the low participation rate is that we leave the players unchallenged in the game.

Another feedback we got from our visitors is that the questions are too simple

[5] This was checked in 22 July 2009

or just boring. Other than the erroneous questions, several of the questions ask for very simple aspects of the word. For example, one question asks whether the word is a proper noun or not. Another question literally asks the player to indicate whether the word in question belongs to herself. This is confusing if not boring. In fact, We were expecting this kind of complaints because the questions must ask the players about the most intricate details of word morphology. So, naturally, the simple questions are found to be boring and the intricate ones are confusing. Therefore, we suspect that this is another reason for the low participation rates.

As we have seen in Chapter 2, the players receive points for their achievements during the game. But there is a subtle problem here. We had to always give a fixed amount of points to each contribution. This is because we also did not know the correct parse of the word that the question is about. However, this is due to the nature of the game. Afterall, the game is meant to be played for disambiguating a word without any tagging beforehand. So, lack of appropriate awarding mechanism leaves players unattached to the game thus resulting in low participation. Fortunately, we have a solution for this problem. After the game continues to run for a while, we will be able to determine the agreement answers for some of the words. When one of these words are the target of a question, we will be able to award the contribution from the player in a more suitable way. But for this study, we were not able to utilize this mechanism.

Without a team composed of several people with some specific professions, such as graphical designers and public relations managers, a game site like this is destined to have some deficiencies. One of them is the lack of an established aesthetics for the site. The current design of the site and the game is acceptable for our study which is a prototype for demonstrating the capabilities of our method. But, to be successful in our final motivation of being able to disambiguate any amount of unrestricted Turkish free text, we need to tie the visitors to the site and make them play as much as time allows them. To do that, we need a design that compels to our target audience which is virtually anyone on the Internet today. So the current design is not capable of that and needs to be replaced with a more professional look. The other deficiency is that the site lacks the standard functionalities that are found in every social networking site on the Internet. Some of them include profile pages, forums, private messaging, friendship networks and groups. These are all proven to increase the ties between the users and the site. However, we could not implement most of these in our game site due to the lack of time.

Among the list of functionalities above, we were only able to add groups functionality to the game site. Even this has received attention from the users. This is supported by the facts that 22 groups are created during the first eleven days, 46 people joined these groups and some groups have as many as 9 people.

Another deficiency was that there were not enough people to play a two player game even at the start of the experiment. This caused a major drawback for both the players and the experiment. When the players could not find somebody to play, they did not come back to check the site. So the number of visitors to the site decreased day by day. We think that there is a threshold for an online game to be successful. If that threshold can be reached, we observe that even more people start to visit the site. However, this was not the case for our game site. To challenge the aforementioned threshold, we made the announcement through several channels: two blogs[6] , two social networking sites[7] and some number of mailing lists.

[6] `http://ileriseviye.org/blog/?p=2351` and `http://www.fazlamesai.net/?a=article&sid=5314`

[7] `http://friendfeed.com/onurgu` and `http://facebook.com`

5. Conclusions

In this work, a game for morphological annotation of a Turkish corpus is developed. The game is meant to be played by two players simultaneously over the Internet. Alternatively, there is a single player mode. Basically, the annotation is done by collecting answers to questions that are automatically created based on a number of templates prepared manually. In fact, the questions are abstractions of the morphological features of a word in its context in the corpus. The two player game mode consists of three stages. In one of the stages, one of the players has to describe the target word to the other player trying to collaboratively guess the word as fast as possible. The guesses and clues submitted in this stage are not directly related with our motivation. Nevertheless, they are valuable given that the assumption is that the words that are submitted by the users are semantically related with the target word. The answers to the questions posed in the other stages are then analyzed statistically and an aggreement answer is determined. An aggregation of these aggreement answers result in a complete morphological disambiguation.

The game is hosted on a publicly accessible web site. The experiment was started on 29^{th} of June 2009. The results reported in the thesis are compiled from the data obtained until 9^{th} of July 2009. The evaluation was done by assesing the performance of all question types over two instances. This was required because we did not have the time and the traffic rates that would allow us to annotate the corpus completely. The reported success rate over the two phases is 63.5%. Given that the actual question types cover 58.3% of the corpus completely, the estimated rate for a complete disambiguation of the corpus is 37.0%. This may sound as a minor achievement. However, it is also shown that if the game were to be hosted on a major nation-wide newspaper, the task would be completed in about two and a half months. Additionally, the corpus coverage can be maximized by adding the remaining possible question types. In this case, the time required to complete the annotation of the whole corpus would be about eight months only. Besides, the time duration loses importance when we realize that our method allows a continuous annotation of any free text. In summary, the results show that the morphological annotation of a Turkish corpus with a multiplayer collaborative game is promising.

One final remark about the thesis is that after the small publicity created during the advertisement of the game site, it was observed that the idea of utilizing “wasted human cycles” through systems over the Internet was recognized at large. This further supports that the human computation is a promising area to explore.

5.1. Future Work

In this section, we list some possible future work that can be pursued on the subject. First of all, the awarding system was static in the sense that in all of the stages of the game the points were fixed. They did not reflect the performance of the players. Although this is because of the nature of the game, incorporating an awarding system that can measure the performance of the players and award accordingly can be more facilitating. In fact, we could implement this system after we collected sufficient number of answers for each question in the first and the second phases.

In the current state of the game, all of the games have random difficulty arising from the randomly selected questions. Nevertheless, we can not measure the difficulty of a question currently. A method for measuring the difficulty of a question or at least categorizing them by hand would enable us to modify the game so that the levels become harder and harder, so the game is more challenging.

As we have seen in Section 3.2, we only cover 58.3 per cent of the whole corpus. This rate could be increased by adding new observation rules by hand or some semi-automatic way. Noting that a majority of the disambiguation can be done by differentiating between the stem nouns, the method of showing the definitions of the alternative nouns to the player and asking which meaning is seen in the sentence can be employed.

Another feature to implement may be to rate users according to their ability in answering the questions correctly. By this way, we could direct the more difficult questions to these users to have a higher percentage of correct disambiguations.

REFERENCES

Sina Afra. Turkish internet sector overview, 2008. URL http://www.slideshare.net/webrazzi/turkey-internet-sector-2008. accessed at 16 Jul 2009.

Morgan Ames and Mor Naaman. Why we tag: motivations for annotation in mobile and online media. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 971–980, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-593-9. doi: http://doi.acm.org/10.1145/1240624.1240772.

Cunningham Tablan Bontcheva. Language engineering tools for collaborative corpus annotation. URL citeseer.ist.psu.edu/734322.html.

Sean Casey, Ben Kirman, and Duncan Rowland. The gopher game: a social, mobile, locative game with user generated content and peer review. In *ACE '07: Proceedings of the international conference on Advances in computer entertainment technology*, pages 9–16, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-640-0. doi: http://doi.acm.org/10.1145/1255047.1255050.

Timothy Chklovski. Collecting paraphrase corpora from volunteer contributors. In *K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture*, pages 115–120, New York, NY, USA, 2005. ACM. ISBN 1-59593-163-5. doi: http://doi.acm.org/10.1145/1088622.1088644.

Timothy Chklovski and Yolanda Gil. Improving the design of intelligent acquisition interfaces for collecting world knowledge from web contributors. In *K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture*, pages 35–42, New York, NY, USA, 2005. ACM. ISBN 1-59593-163-5. doi: http://doi.acm.org/10.1145/1088622.1088630.

Boreum Choi, Inseong Lee, Dongseong Choi, and Jinwoo Kim. Collaborate and share: An experimental study of the effects of task and reward interdependencies in online games. *CyberPsychology & Behavior*, 10(4):591–595, 2007. doi: 10.1089/cpb.2007.9985. URL http://www.liebertonline.com/doi/abs/10.1089/cpb.2007.9985.

Gülşen Eryiğit. ITU treebank annotation tool. In *Proceedings of the Linguistic Annotation Workshop*, pages 117–120, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W07/W07-1519.

Google. Google web toolkit, 2008. URL http://code.google.com/webtoolkit/.

E. L. M. Law, L. von Ahn, R. B. Dannenberg, and M. Crawford. Tagatune: A game for music and sound annotation.

Xiaoyi Ma, Haejoong Lee, Steven Bird, and Kazuaki Maeda. Models and tools for collaborative annotation, 2002. URL http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0204004.

Kemal Oflazer and Gokhan Tur. Combining hand-crafted rules and unsupervised learning in constraint-based morphological disambiguation. In Eric Brill and Kenneth Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 69–81. Association for Computational Linguistics, Somerset, New Jersey, 1996. URL citeseer.ist.psu.edu/article/oflazer96combining.html.

Kemal Oflazer, Gökhan Tür, and Gskhan Tfir. Morphological disambiguation by voting constraints. pages 222–229, 1997.

Matthew Richardson and Pedro Domingos. Building large knowledge bases by mass collaboration. In *K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture*, pages 129–137, New York, NY, USA, 2003. ACM. ISBN 1-58113-583-1. doi: http://doi.acm.org/10.1145/945645.945665.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. Morphological disambiguation of Turkish text with perceptron algorithm. In *CICLing 2007*, volume LNCS 4394, pages 107–118, 2007. URL http://www.cmpe.boun.edu.tr/~hasim/papers/CICLing07.pdf.

Maik Stührenberg, Daniela Goecke, Nils Diewald, Alexander Mehler, and Irene Cramer. Web-based annotation of anaphoric relations and lexical chains. In *Proceedings of the Linguistic Annotation Workshop*, pages 140–147, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W07/W07-1523.

Qi Su, Dmitry Pavlov, Jyh-Herng Chow, and Wendell C. Baker. Internet-scale collection of human-reviewed data. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 231–240, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: http://doi.acm.org/10.1145/1242572.1242604.

Hakkani D. Tur and K. Oflazer. Statistical morphological disambiguation for agglutinative languages, 2000. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.1781`.

Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, New York, NY, USA, 2004. ACM. ISBN 1-58113-702-8. doi: http://doi.acm.org/10.1145/985692.985733.

Luis von Ahn, Shiry Ginosar, Mihir Kedia, Ruoran Liu, and Manuel Blum. Improving accessibility of the web with a computer game. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 79–82, New York, NY, USA, 2006a. ACM. ISBN 1-59593-372-7. doi: http://doi.acm.org/10.1145/1124772.1124785.

Luis von Ahn, Mihir Kedia, and Manuel Blum. Verbosity: a game for collecting common-sense facts. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 75–78, New York, NY, USA, 2006b. ACM. ISBN 1-59593-372-7. doi: http://doi.acm.org/10.1145/1124772.1124784.

Luis von Ahn, Ruoran Liu, and Manuel Blum. Peekaboom: a game for locating objects in images. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64, New York, NY, USA, 2006c. ACM. ISBN 1-59593-372-7. doi: http://doi.acm.org/10.1145/1124772.1124782.

Deniz Yuret and Ferhan Türe. Learning morphological disambiguation rules for turkish. In *HLT-NAACL 06*, June 2006. URL `/pub/hlt-naacl-06,/pub/hlt-naacl-06/morph-disamb.pdf,/pub/hlt-naacl-06/hlt06.ppt`.

Deniz Yüret. Morphologically tagged corpus. URL `http://www.denizyuret.com/turkish`. accessed at 16 Jul 2009.