



HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

SCHOOL OF COMPUTER SCIENCE

T-622-UROP UNDERGRADUATE RESEARCH OPPORTUNITY

Orðspor: Labeling named entities with computer games

Students:

Ólafur Páll GEIRSSON

Guðmundur HARÐARSON

Guðmundur Þórður GUÐMUNDSSON

Supervisor:

Dr. Hrafn LOFTSSON

Reykjavik May 26, 2014

Abstract

Most current natural language processing models use large labeled datasets to achieve good performance, but large accurately labeled datasets for different languages and domains are hard to come by. The standard method to produce labeled data has been through manual labor which is expensive and tedious. This thesis presents an experiment to use computer games and crowdsourcing as a cheap and fun alternative to produce labeled data for the named entity recognition task. We introduce *Orðspor*, a website with three computer games designed with the objective to collect named entity tags for the Icelandic language text while providing players with an enjoyable time. The games are similar in many ways; they share a common API, they receive input from the same source and their output is merged to produce a single dataset. By using a shared API, we were able to rapidly develop the three games and experiment with different game designs. We envision that many more games could be developed with our API and attract a broad range of players. Moreover, such games could with proper marketing effort be embedded into people's daily routines in various ways producing a wealth of datasets for use in a variety of natural language processing tasks.

Contents

1	Introduction	4
2	Named entity recognition	5
2.1	Definition	5
2.2	Taxonomy	6
2.3	Implementation	6
2.4	Datasets	8
2.5	Application	9
3	Games with a purpose	9
3.1	Motivation	10
3.2	Game templates	10
3.2.1	Output-agreement	10
3.2.2	Inversion-problem	11
3.2.3	Input-agreement	11
3.2.4	Single player	11
3.3	Design guidelines	11
3.3.1	Player enjoyment	12
3.3.2	Output accuracy	12
3.4	Evaluation	13
3.5	Crowdsourcing	13
4	Orðspor	14
4.1	API	14
4.2	Front page	15
4.3	Orð-árás	16
4.3.1	Design	16
4.3.2	General gameplay	16
4.4	Orð-skot	17
4.4.1	Design	17
4.4.2	General gameplay	20
4.5	Orð-val	21
4.5.1	Design	21
4.5.2	General gameplay	21
4.6	Badges	23
4.7	Scoreboard	23
4.8	Output accuracy	26

5	System components	28
5.1	Server-side	28
5.1.1	Play Framework	28
5.1.2	CouchDB	29
5.2	Client-side	30
5.2.1	Orð-árás	30
5.2.2	Orð-skot	31
5.2.3	Orð-val	33
5.3	Version control	33
5.4	Continuous Integration	33
5.5	Team Communication	34
6	Experimental results	35
6.1	Dataset	35
6.2	Marketing	35
6.3	Empirical results	35
6.4	User experience	38
7	Learning outcomes	39
7.1	Design and develop a web application	39
7.2	Design and develop web-based games	40
7.2.1	Orð-árás	40
7.2.2	Orð-skot	40
7.2.3	Orð-val	41
7.3	Independent learning and research	41
8	Future work	41
8.1	Games	41
8.2	System and API	42
9	Conclusions	43
A	API	47
B	User Roles and Product backlog	53

1 Introduction

Text analytics systems empower us to make full use of textual information available on the Web and elsewhere. Government officials, politicians, marketing agents, stock brokers and regular people browsing the Web are examples of people who may benefit from automatic text analytics from the news, media and online discussions.

Named entity recognition (NER) is an important subtask in text analytics. The standard technique used to develop state-of-the-art NER systems is to employ supervised learning methods [Finkel et al., 2005, Nadeau and Sekine, 2007, Ratinov and Roth, 2009]. One benefit of supervised learning methods is that they can quite easily be applied to multiple languages, including less commonly spoken languages such as Icelandic. The well known downside to supervised learning methods however, is that they require large amounts of training data.

The most common method for obtaining precise named entity labels has been to manually label text, which is laborious and costly. The *ESP game* [von Ahn and Dabbish, 2004] introduced a novel approach to solve a somewhat similar problem, namely produce meaningful descriptions for all images on the Web. In the game, the incentive for users to take part was only the pleasure of playing. Through the lifetime of the ESP game, a total of 1,271,451 labels were generated for 293,760 images which then were released for use to further improve image recognition algorithms. This success inspired a new genre of games commonly referred to as “*games with a purpose*” (GWAP), a phrase popularized by Luis von Ahn in [von Ahn, 2009]. Various attempts (e.g., [von Ahn et al., 2006a, von Ahn et al., 2006c, Scharl et al., 2012, Siorpaes and Hepp, 2008]) have been made to repeat the idea on different kinds of labeling tasks with varying results.

Considering the success of the ESP game, one may wonder, what would happen if we made the labeling named entities more enjoyable? Would perhaps people be willing to label named entities for free if we turned the experience into a game? In this thesis, we have made an attempt to make labeling named entities in text fun.

We present *Orðspor*¹, a website with three games that are enjoyable to play for the user while they produce meaningful named entity labels. The website has been open to the public for one week at the time of this writing and our experimental results so far show that over 55.000 words have been annotated in over 500 user sessions by 84 players. With this rate, we could annotate the gold-standard tagged Icelandic corpus which contains 1 million words in around 20 weeks. We consider the main contributions of this thesis to be:

¹<http://ordspor.org>

- How we designed a generic application programming interface (API) for use in almost any kind of named entity labeling game, explained in Section 4.1.
- How we rapidly developed three web games with the use of the previously mentioned API, explained in sections 4.3, 4.4 and 4.5.

Furthermore, we consider the main learning outcomes to develop a client- and server-side web service, explained in sections 5.1, 5.2, 5.3 and 5.4 and work together as a distributed team, explained in Section 5.5.

2 Named entity recognition

Who? What? Where? When? These interrogative questions may be the first thing that come to our minds when we read the news, study a book or take in new information. The answers to our questions are commonly names of people, locations or specific dates. Named entity recognition is the field within natural language processing (NLP) that deals with automatically recognizing these important pieces of information in text.

2.1 Definition

Named entity recognition (NER) is the process of detecting words in a text that constitute proper nouns and classifying them into a defined category [Jurafsky et al., 2000]. A proper noun refers to a *unique* entity, such as “Iceland” or “John”. The category which the proper nouns are classified into are common nouns which refer to a category of entities, such as “location” or “person”. Take the sentence

John spent his summer in Iceland working at Reykjavik University

which contains three named entities, namely John, Iceland and Reykjavik University. To perform NER here we would label those three named entities into their respective category and mark all other words as O, denoting “other”, resulting in

John/PERSON spent/O his/O summer/O in/O Iceland/LOCATION
working/O at/O Reykjavik/ORGANISATION University/ORGANISATION.

Observe that although Reykjavik University is a multi-word unit and refers to the same entity and is therefore counted as one entity.

2.2 Taxonomy

NER systems are often developed for a particular domain and, thus, the taxonomy used in each system varies greatly. Some specialized systems may be only interested in commercial products, work of art or biological entities. In finances, the definition of a named entity may be extended to include entities which are not even proper nouns such as temporal and numerical expressions, counts and prices. Examples of entity types which may be present in generic NER systems include people, organizations, locations, geo-political, facilities and vehicles.

The language independent named entity recognition shared task at CoNLL² 2003 used four categories [Tjong Kim Sang and De Meulder, 2003], namely: persons, organizations, location and miscellaneous. A detailed breakdown of the four categories can be seen in Table 1. Observe how diverse the categories are, it should not come as a surprise that a general agreement has not been (and may never be) made in the literature on what categories to include or exclude in NER system.

2.3 Implementation

There have traditionally been two different approaches to developing NER systems, linguistic based and data-driven. The linguistic based on one hand relies on handwritten rules such as regular expressions. The data-driven, or machine learning, approach on other hand relies on having large sets of annotated training data in order to perform statistical inference when classifying unseen words. The focus in NER research over the recent years has largely focused on the latter approach [Nadeau and Sekine, 2007]. Traditionally, the task of predicting a class type given a word is viewed as a sequential prediction problem. Some successful methods include hidden Markov models [Rabiner, 1989], conditional random fields [Lafferty et al., 2001] and sequential application of Perceptron or Window [Collins, 2002].

Recent studies stress the importance of using non-local information in training NER classifiers. Mannig and Finkel pioneered [Finkel et al., 2005] this approach in 2005 with the Stanford NER by using Gibbs sampling, as they put it in [Finkel et al., 2005, p. 1]: “Most current statistical NLP models use only local features . . . but this makes them unable to fully account for the long distance structure that is prevalent in language use”. When Ratinov and Roth [Ratinov and Roth, 2009] presented their LBJ-NER in 2005, they similarly considered non-local dependencies to be a key design decision in developing their NER system. As mentioned in [Finkel et al., 2005] and [Ratinov and Roth, 2009], the Stanford NER and LBJ-NER are to this date some of the best performing NER systems for the English lan-

²Conference on Computational Natural Language Learning

Persons	first, middle and last names of people, animals and fictional; characters; aliases
Organizations	words of which one part is a location, organisation; miscellaneous, or person; adjectives and other words derived from a word; which is location, organisation, miscellaneous, or; person; religions; political ideologies; nationalities; languages; programs; events (conferences, festivals, sports competitions; forums, parties, concerts); wars; sports related names (league tables, leagues, cups); titles (books, songs, films, stories, albums, musicals; TV programs); slogans; eras in time; types (not brands) of objects (car types, planes; motorbikes)
Locations	roads (streets, motorways); trajectories; regions (villages, towns, cities, provinces, countries, continents; dioceses, parishes); structures (bridges, ports, dams); natural locations (mountains, mountain ranges, woods; rivers, wells, fields, valleys, gardens; nature reserves, allotments, beaches; national parks); public places (squares, opera houses, museums, schools; markets, airports, stations, swimming pools; hospitals, sports facilities, youth centers; parks, town halls, theaters, cinemas, galleries; camping grounds, NASA launch pads, club; houses, universities, libraries, churches; medical centers, parking lots, playgrounds; cemeteries); commercial places (chemists, pubs, restaurants, depots; hostels, hotels, industrial parks; nightclubs, music venues); assorted buildings (houses, monasteries, creches, mills; army barracks, castles, retirement; homes, towers, halls, rooms, vicarages; courtyards); abstract “places” (e.g. the free world)
Miscellaneous	companies (press agencies, studios, banks, stock; markets, manufacturers, cooperatives); subdivisions of companies (newsrooms); brands; political movements (political parties, terrorist); organisations; government bodies (ministries, councils, courts, political unions; of countries (e.g. the U.N.)); publications (magazines, newspapers, journals); musical companies (bands, choirs, opera companies, orchestras); public organisations (schools, universities, charities); other collections of people (sports clubs, sports); teams, associations, theaters companies; religious orders, youth organisations

Table 1: The named entity taxonomy used at ConLL-2003, borrowed from <http://www.cnts.ua.ac.be/conll2003/ner/annotation.txt>

guage. One challenge with using non-local features however is that they requires large datasets to train on.

2.4 Datasets

In order to evaluate a NER system one must have available accurately labeled datasets to compare the produced output with the expected output. Given the popularity of supervised methods, one must also have available datasets, besides the datasets used for evaluation, to train ones classifier. This imposes some challenges if there exist none or only small tagged datasets for a given language or a particular domain.

A comparison of the sizes of some corpora tagged with named entities for different languages can be seen in Table 2. By looking at the figures we see a striking pattern.

For more commonly spoken languages, reasonably large datasets with named entity labels already exist. At the English Named Entity Task at the Message Understanding Conference in 1997 (MUC7), the winning team manually annotated a large collection of New York Times news wire articles [Miller et al., 1998]. Unfortunately, this corpus was never made publicly available. At the CoNLL02 in 2002, datasets for the Spanish and Dutch languages were released at the NER shared task [Tjong Kim Sang, 2002]. The Spanish text is a collection of news wire articles published by the Spanish EFE News Agency and the Dutch text comes from the Belgian newspaper “De Morgen”. The following year in 2003, two additional annotated datasets for the English and German languages were released at the CoNLL03 NER shared task [Tjong Kim Sang and De Meulder, 2003]. The English text is a collections of Reuters news wire articles from 2002 and the German text is taken from the ECI Multilingual Text Corpus extracted from the German newspaper Frankfurter Rundschau.

For less commonly spoken languages such as the Nordic languages, reasonably large datasets do not appear to exist. The paper “Named Entity Recognition for the Mainland Scandinavian Languages” presents experimental results from the comparison of six different NER systems (both linguistic and data-drive) for Swedish, Norwegian and Danish [Johannessen et al., 2005]. Two different methods were used to evaluate the results: one small-scale, manual comparative method and one larger-scale automatic method. The only attempt, we know of, to develop an Icelandic NER system relied on hand-written rules because of the fact that no available corpus exists with tagged named entity labels [Tryggvason,].

Corpus	Tokens	Named entities
MUC7 English	790,000	65,500
CoNLL 2002 Spanish	380,923	58,292
CoNLL 2002 Dutch	333,582	52,666
CoNLL 2003 English	301,418	35,089
CoNLL 2003 German	310,318	20,357
Scandinavian 2003 small-scale	1,800	115–146 ³
Scandinavian 2003 large-scale	40,000	1,775

Table 2: NER labeled corpora and their respective sizes.

2.5 Application

Before we continue it is worth mentioning that NER systems themselves are generally not performed as standalone task but rather as a subtask for other common NLP tasks, the most common one being information extraction.

Information extraction, also known as text analytics, is the process of turning semantic information from unstructured text into structured data. Unstructured text refers to any written human language text that does not follow a structured pattern. Some of the most common mediums for sharing human knowledge are through unstructured text such as books, articles, laws, news stories, emails, letters and so on. Although unstructured text has been useful in this sense for knowledge exchange between a writer and his readers, it gives little opportunity for one to automatically reason about the contents of large collections of text. If we encode the information embedded in unstructured text into relational database tables ones capability to perform this kind of reasoning and further decision making is greatly increased.

3 Games with a purpose

Loosing track of time while playing an enjoyable computer game is fun! What if you contribute to a valuable work while playing, would that increase your satisfaction? Well, the work is not necessarily valuable for the player. Other parties such as the game owner and possibly other users benefit form the data or knowledge collected from players. The outweighing motivation for the player is entertainment. Few players would bother playing a boring game, at least not repeatedly.

3.1 Motivation

Games with a purpose (GWAPs) are designed to be enjoyable while ensuring that the data they collect is free from error [von Ahn and Dabbish, 2008]. The “purpose” in the phrase “game with a purpose” refers to the data or knowledge gathered from players playing that particular game.

The GWAP concept is strongly related to another concept *gamification*. A definition of gamification is proposed by Deterding et. al [Deterding et al., 2011, p. 2]: “Gamification is the use of game design elements in non-game contexts”.

What is the rationale of encouraging people to play computer games? Luis von Ahn identifies three factors as a support for the GWAP approach: (i) the number of people using the Internet increases year by year; (ii) there are tasks that humans find easy but are hard to solve with computers; (iii) people spend enormous amount of time playing computer games [von Ahn and Dabbish, 2008].

Let’s elaborate on the last factor to give some feeling for the size of the computer games market. To simplify this thought process just focus on one single game, namely the world famous cross platform game Angry Birds. At the Web 2.0 Summit 2011, Andrew Stalbow, at that time Rovio Entertainment’s General Manager of North America, stated that Angry Birds was played roughly 300 million minutes per day [O’Reilly, 2011, 2m42s]. He furthermore declared that the company had not spent one dollar in marketing for the game. The game is simply played because players enjoy it.

3.2 Game templates

Dabbish and von Ahn describe three GWAP templates: output-agreement games, inversion-problem games, input-agreement games [von Ahn and Dabbish, 2008]. All three games are two player games where the goal of the gameplay is to form a some sort of agreement between those two players.

3.2.1 Output-agreement

The game is set up so that two players are chosen randomly from the pool of players. Both players are given the same input. Their task is to guess the same output based on the input. The ESP game is an example of an output-agreement game, where two players try to guess on the same label for a given picture.[von Ahn and Dabbish, 2008].

3.2.2 Inversion-problem

The game shows an input to one of the players (player A). Player B, the second player receives a number of outputs created by player A. The outputs are supposed to give player B a clue of the input given to player A. Player B's goal is to guess what input was given to player A.

- *Peekaboom*: A game that help computers to located objects within an image. The result is us used to improve the accuracy of computer vision algorithms[von Ahn et al., 2006c].
- *Phetch*: A game that collects explanatory descriptions of images on the web [von Ahn et al., 2006a].
- *Verbosity*: In this game the purpose is to collect common-sense facts in order to make computer programs more intelligent [von Ahn et al., 2006b].

3.2.3 Input-agreement

The game assigns inputs to two randomly chosen players. The input given is either the same input or different. The players generate outputs to each other as a clue of the their inputs. The goal for the two players is to judge from the outputs they receive, if their inputs are different or the same.

3.2.4 Single player

Not all GWAPs fall into the above mentioned templates. There are also single player GWAPs that have become successful. One of those games is Phrase Detectives, developed to collect data about anaphora and themed around a detective story [Poesio et al., 2012]. The agreement mechanism in Phrase Detective is different from the two player templates, where two players are needed to form an agreement. The game accepts the input from a single player without further check.

3.3 Design guidelines

In “Designing Games with a Purpose” [von Ahn and Dabbish, 2008], von Ahn and Dabbish discuss design guidelines for GWAP. The guidelines introduce some factors that are deemed important for a game to be successful. Firstly, factors that are important for *player enjoyment* and, secondly, factors that are of importance for *output accuracy*.

3.3.1 Player enjoyment

The factors related to player enjoyment:

- *Timed response*. Players only have limited time to solve the puzzles given to them.
- *Score keeping* gives a feedback of players performance in the game.
- *Player skill levels* can be used to increase difficulty when player's skill increases. This a tool to motivate players to strive for higher rank within the game.
- *High-score lists* are important to see how player ranks relatively to other players.
- *Randomness* involves random pairing of players and random inputs for varying difficulty.

3.3.2 Output accuracy

Dabbish and von Ahn state that the game templates must have some sort of mechanism to ensure accuracy of output and prevent cheating on players behalf. These factors include:

- *Random matching*. Players are paired randomly to prevent agreements before the game is played.
- *Player testing*. Output known by the game is tested on players. If output from player repeatedly differs from correct output, that player is deemed untrustworthy and the output is discarded.
- *Repetition*. The same game is repeated certain number of times. The final output from that game is not considered correct unless at least 50% of games give the same result.
- *Taboo results*. This handles the situation when many different outputs are regarded as correct answer to the same input. Here an example of correct output is displayed on screen while the game is played, but players are not allowed to choose that particular output as an answer. The example output is the so called "taboo" or "off-limits" output.

3.4 Evaluation

It is important to have some measure on a GWAP performance. A useful measurement is defined by von Ahn and Dabbish as the *expected contribution* from a game. The exact definitions are borrowed from [von Ahn and Dabbish, 2008]:

Throughput Average number of problem instances solved per human hour. This average is taken over all game sessions through a reasonable lengthy period of time and over all players of the game.

ALP Overall amount of time the game is played by each player averaged across all people who have played it.

Expected contribution Average number of problem instances a single human player can expect to solve by playing particular game.

Observe that *expected contribution* is *throughput* multiplied with *ALP*.

GWAP with higher expected contribution is favored over a game with lower expected contribution. It is judged to be more successful.

3.5 Crowdsourcing

The utilization of GWAP is frequently related to large time consuming problems that needs large number of players to get the work done. This approach to problem solving can be identified as “crowdsourcing”. Many varying definitions of crowdsourcing can be found in the literature and there seems to be a lack of general acceptance of a single definition. An excellent attempt to define crowdsourcing was made by Estellé-Arolas, et. al [Estellés-Arolas and González-Ladrón-de Guevara, 2012, p. 2]:

“Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their

advantage what the user has brought to the venture, whose form will depend on the type of activity undertaken.”

The following facts from the above definition of crowdsourcing are directly related to Orđspor:

- A user / player is needed to perform the task of NER for us.
- His / her motivation is the satisfaction of playing the games.

4 Orđspor

In this section, we present our work. Orđspor is a website with three games that are enjoyable to play for the user while they collect meaningful named entity tags. The games are similar in many ways; they share a common API, they receive input from the same source and their output is merged to produce a single dataset.

4.1 API

The core of Orđspor is its application programming interface (API). The API provides methods for the games to retrieve text for annotation and submit the annotation produced by the game players. Using a shared API allowed for rapid development of our games.

The main parts of the Orđspor API are shown in Listing 1. A handful of concepts behind the API architecture are borrowed from Fielding’s RESTful principles [Fielding, 2000], most notably: (i) the HTTP request semantics are signified by the request method name, (ii) data storage concerns are separated from the user interface concerns, (iii) communication between the client and server is stateless since the client cannot take advantage of any stored context on the server and (iv) the url scheme takes potential future modifications to the API into account by including a version prefix for every url (as seen with `v1` in the listing below).

Listing 1: The Orđspor API, the first column denotes the HTTP request method and the second column denotes the methods relative path

GET	/v1/next-sentence
POST	/v1/annotate
GET	/v1/total-words-annotated

Sending an HTTP GET request to `/v1/next-sentence` will return a sentence ready to be annotated. The server will respond with one of two types of sentences, either a “new sentence”, i.e., a sentence no one has annotated before, or an “old sentence”, i.e., a sentence which other players have annotated before. New sentences are chosen in the order by how many proper nouns they contain, part-of-speech tags are provided by IceNLP [Loftsson and Rögnvaldsson, 2007]. New sentences are handed out only to experienced players with probability $\frac{1}{3}$. The criterion for players to be considered as experienced was chosen quite liberally, in Orðspor we considered players who have submitted more than 10 annotations to be experienced. An old sentence is returned for inexperienced players in all cases and two out of three times for experienced players. The server keeps track of when sentences are handed out to avoid re-sending the same sentence twice within a short interval time period. An example of a response from `/v1/next-sentence` can be found in the Appendix as Listing 2.

To submit an annotation, an HTTP POST request is sent to `/v1/annotate` with an annotation object attached as data. The annotation object is validated by the server before it is stored in the persistence layer. After the sentence has been safely stored in the database, the server will prepare a badge (see Section 4.6) and send a confirmation response to the client. An example of an annotation object can be found in the Appendix as Listing 3.

Observe that the API is not strictly limited to the named entity recognition task. It could be possible to extend it with new methods for posting annotation tasks of almost of any type. Examples of such tasks could include part-of-speech tagging, parsing, sentiment analysis and word sense disambiguation.

4.2 Front page

First impressions are important and the front page of Orðspor is constructed with this thought in mind. On the front page the users are greeted by a tag line followed by a success meter that shows how many words have been tagged so far, see Figure 1. This meter acts as an encouragement for the users to take part by playing the games and a visible difference on the meters values can be recognized after each session of playing one of the games.

Providing an easy access to information regarding the project are essential. To motivate curious users, information extraction was given as an example of an application using named entity recognition, see Figure 2. Furthermore, the front page provides information on how it can be beneficial to collect labels for the named entities and what is meant by *named entities*. Other more technical information regarding the project are provided in the footer of the front page.

Getting users to play one or even all of the three games is the main objective by



Figure 1: Tag line and success meter welcomes the user on the front page.

the front page. By providing three different pathways to each game the front page makes it easy for the user to access each game, see Figure 3.

4.3 Orð-árás

Many veteran gamers have played the classic video game Space Invaders released 1978. The idea behind Orð-árás (e. Word Invaders) is derived from that game. Instead of shooting down bad aliens the goal in Orð-árás is to shoot down words, particularly named entities. Before the named entity is shot down the player needs to select the correct category for it. In that sense it is more complex than Space Invaders where the player shoots down all aliens with the same gun. In contrast the named entities and other words are friendly and do not fire back missiles at the player.

4.3.1 Design

One of the design criteria for Orð-árás was to create a game that was primarily aimed at touch screen devices. This constraint made us somewhat drift away from original idea of 2D simple shooting game played on desktop, where the natural gameplay is to use both hands. That does not work well on a touch screen devices, especially not on a phone.

4.3.2 General gameplay

The gameplay for Orð-árás consists of three screens. The intro screen, the main play screen and the quit screen, as shown in Figure 4. The category for named entities is selected by clicking the mouse on correct category or by tapping on it on

1. Hvers vegna viljum við kenna tölvum íslensku?

Til þess að þær geti hjálpað okkur að nota þær upplýsingar sem við skrifum á íslensku. Til dæmis er hægt að láta þær setja upplýsingar í töflu í stað þess að gera það handvirk.

Upplýsingar í samfelldum texta

Arnar Már til liðs við WOW air

Vísaskipti | mbl | 7.9 | 11.07

Arnar Már Arnþórsson hefur verið ráðinn framkvæmdastjóri sölu- og markaðssviðs WOW air. Arnar Már kemur til WOW air frá Bláa Lóninu hf. þar sem hann hefur gengt starfi sölu- og markaðsstjóra frá árinu 2009 til 2012. Frá árinu 2009 til 2012 starfaði Arnar sem markaðsstjóri Icelandair í Norður Ameríku og sem sölustjóri sama svæðis frá 2008 til 2009.

Rakel fréttastjóri RÚV

Innlent | mbl | 16.4 | 14.54 | Upptækt 19.11

Rakel Þorbergsdóttir hefur verið ráðin fréttastjóri Ríkisútvarpsins. Þröstur Helgason hefur verið ráðinn dagskrárstjóri Rásar 1, Frank Þórir Hall dagskrárstjóri Rásar 2 og Skarphéðinn Guðmundsson dagskrárstjóri Sjóvarps. Andrea Röbertsdóttir verður mannauðsstjóri RÚV. Ný framkvæmdastjóri RÚV var lýmt á starfsmannafundi í dag. Fjórir konur voru ráðnar framkvæmdastjórnar og fjórir karlar.

Árdis ráðin framkvæmdastjóri Skema

Vísaskipti | mbl | 16.4 | 11.09

Árdis Ármannsdóttir hefur verið ráðin til að gegna stöðu framkvæmdastjóra Skema. Hún tekur við af Þórunni Jónsdóttur sem mun flytjast til starfstöðuva móðurfyrirtækis Skema, Rekode Education í Washington fylki í Bandaríkjunum.

Sömu upplýsingar í töflu

Ráðningar í fréttum

Fyrirtæki	Ráðin staða	Hver var ráðin(n)	Dags.
WOW Air	Framkvæmdastjóri sölu- og markaðssviðs	Arnar Már Arnþórsson	16.5.2014
Skema	Framkvæmdastjóri	Árdis Ármannsdóttir	16.4.2014
Ríkisútvarpið	Fréttastjóri	Rakel Þorbergsdóttir	14.4.2014
Rás 1	Dagskrárstjóri	Þröstur Helgason	14.4.2014
Rás 2	Dagskrárstjóri	Frank Þórir Hall	14.4.2014
Sjóvarpið	Dagskrárstjóri	Skarphéðinn Guðmundsson	14.4.2014
RÚV	Mannauðsstjóri	Andrea Röbertsdóttir	14.4.2014

Figure 2: Text on why we should teach computers Icelandic is a part of the information provided on the front page.

a touch screen device. The challenge is to select the correct category (by clicking / tapping on a specific named entity) before it lands on the ground and disappears. The category “Annað” (e. other) is for all words that are not named entities. If a word lands on the ground it is automatically annotated as *Other*. After each round played the game gives feedback on the player’s performance compared to other players and score is registered on the score table.

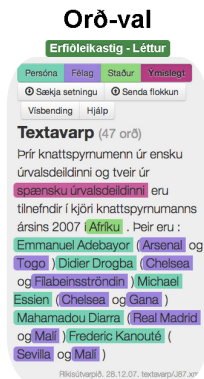
4.4 Orð-skot

“All hand hoay! Load them guns and fire!” Games where the players use some kind of a projectile in their advancement have been popular through the years. Similarly to Orð-árás, Orð-skot’s core idea builds on the design of games where the gameplay is dependent on the players loading a gun and shooting an item onto or into a predefined area.

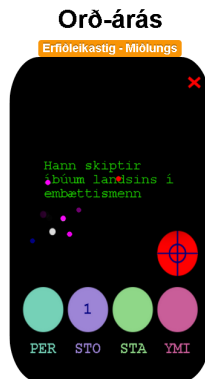
4.4.1 Design

Understanding and pin pointing what makes a game popular is a tough challenge; replicating that design paradigm to a new game, even more so. Even the big companies in the game industry were puzzled by the success of Flappy Bird, a game that went against all of the preferred design paradigms identified in most popular games [Pratt, 2014]. The demanding and tough to beat gameplay of Flappy Bird is what inspired the game design for Orð-skot. By making the game fast paced

(a) The navigation bar at the top of the page, provides links to each game and other pages.



Lærðu á flokkana og sjáðu hversu sammála þú ert öðrum spílurum



Skjóttu orðin áður en þau ráðast á þig



Allir upp á dekk! Hlaðið byssurnar og skjótið!

(b) Difficulty labels, screenshots and tag lines help the user to decide what game to play.

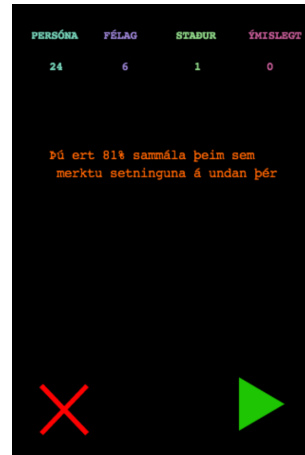


(c) At the bottom of the page colour coded buttons lead to each game.

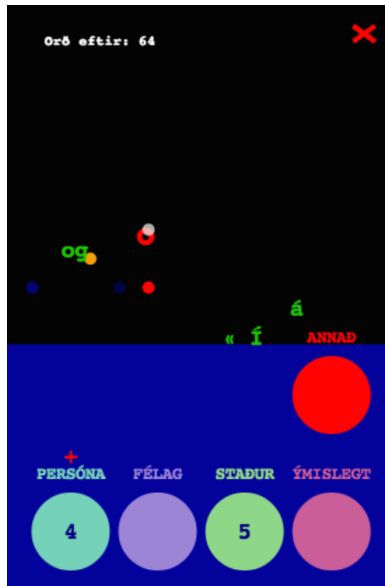
Figure 3: An overview of the three different pathways to each game.



(a) Start screen: Game instructions. Push green button to play.



(b) End screen: Displays players performance. Push green button for another round or push "X" to quit.



(c) Play screen: Player selects category with click / tap and shoots down word with click / tap.

Figure 4: Game screens.

and demanding the players may struggle becoming successful at playing the game (i.e. tag named entities quickly and accurately). The reward from playing the game becomes that much higher.

In Orð-skot, text tagging accuracy is possibly sacrificed for the gameplay. For novice players the game is hard and they might therefore miss named entities in the text. In order to become good at the game, the players are forced to read the text as fast as they can and in most cases embrace the in-game hot-key system. Our hope is that this kind of game design makes the game more enjoyable to play and the players might be tempted to keep going, resulting in the game receiving a fair amount of play-time. The more the game is played by each player, the better he or she becomes in tagging words, resulting in a more accurate named entity tagging.

4.4.2 General gameplay

The starting screen of Orð-skot provides the user with a tutorial on how to play the game, see Figure 5a. The tutorial informs the users of the objective of the game, that they need to identify words in the text as named entities and select them. Once the players have read through the tutorial they can start the game by pressing the button labelled with “*Spila*”.

When the game starts the players are provided with a text that scrolls up from the screen.

By selecting a word, that word is transferred as a cannon ball into the cannon, see Figure 5b. The players can not have more than three charges of cannon balls in the cannon at each time, providing the player with interesting strategic gameplay. Once a cannon ball has been loaded to the cannon, the players tag the word by shooting the cannon ball into an appropriate category. It is possible to shoot from the cannon by either by clicking (or tapping) the category on the screen or by using the in-game hot-keys. Once a word is added to a category, other words with the same lemma are automatically added to the same category. Lemmatisation is provided by the IceNLP toolkit [Loftsson and Rögnvaldsson, 2007].

Orð-skot provides methods for undoing actions. If the players recognize that they made a mistake, either by loading the cannon with a word that is not a named entity or by shooting the cannon ball into the wrong category, they can remove the cannon ball from the cannon or category, with a simple click (or tapping). Removing the cannon ball from the cannon returns it to the unlabeled text. Removing a cannon ball from a category loads it back into the cannon.

The round ends when all of the text has scrolled up from the screen. The players can then review their work and remove words that are not named entities or remove named entities that are in the wrong category, see Figure 5c. The players are not able to move words from one category to another in this phase.

Once the players have reviewed their work they can either continue playing or quit the game. If the players decide to continue playing they receive a feedback, see Figure 5d. The feedback is provided in the form of a badge, see Section 4.6.

4.5 Orð-val

The standard way to annotate text is by brute labor. For this reason, we wanted to create an interface that perhaps does not share the same game-like features as Orð-árás and Orð-skot but offers an efficient way for labeling named entities. In the worst case we thought, we could at least create a practical tool.

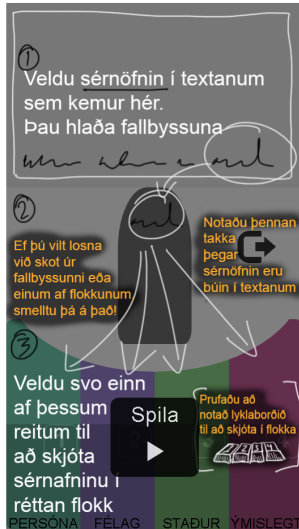
4.5.1 Design

The key design decision in Orð-val was to make it easy to use, efficient at annotating named entities and most importantly be able to produce accurate output. In order to make Orð-val easy to use, the player (or annotator) ought to have access to full descriptions of the categories listed in Table 1, previously mentioned in Section 2.2. Furthermore, the player should always have access to a help screen to become familiar with Orð-val’s user interface. To maximize the annotation efficiency, a minimum number of clicks or keyboard presses ought to be necessary to submit a complete annotation. To ensure tagging accuracy, we considered that the player should be able to edit any word in his submission throughout the whole period of each iteration in the game. Finally, the practical mission of Orð-val ought not to come at the cost of making the game unnecessarily boring.

4.5.2 General gameplay

Figure 6 shows an overview of Orð-val’s user interface. Let’s look at how the above mentioned design decisions are reflected in Orð-val’s user interface. When opening the game, the user is prompted with a short tutorial, presented in Figure 6a. When the user hovers with a mouse over a button, Orð-val reveals a small tooltip reminding the button’s respective hotkey. This emphasis on keyboard short-cuts is shown in Figure 6d.

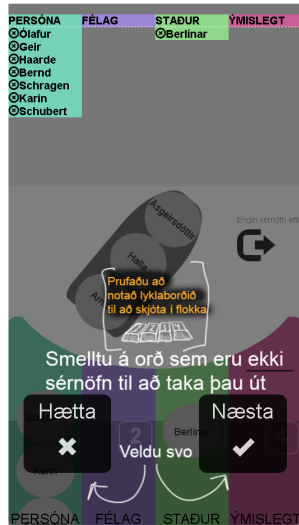
The user has the option to highlight nouns in the game, an example of this feature shown in Figure 6b. Conceivably, by highlighting nouns, the player can more easily locate named entities in the text. Part of speech tags for detecting nouns are provided by IceNLP [Loftsson and Rögnvaldsson, 2007]. Below the main interface of Orð-val, the user has access to thorough examples, displayed in Figure 6c, of what type of named entities fall into each of the four categories. In contrast with Orð-árás and Orð-skot, a timed response was not introduced into the game. We



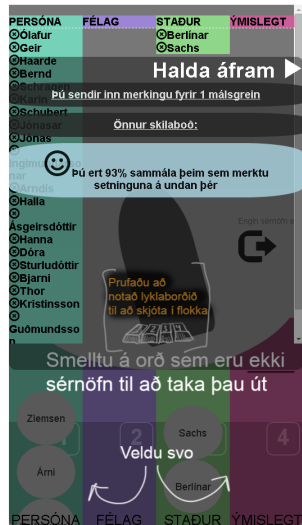
(a) The tutorial provides the user with all the information he or she needs in order to play the game successfully.



(b) The game components consist of the scrolling text on the text, cannon in the middle and the categories in the bottom.



(c) The player is able to review his or hers work, removing words that are not named entities or are wrongly categorized NE.



(d) A feedback on how the players annotations on the given text compare to the annotations of other players

Figure 5: An overview of Orð-skot's user interface

considered that the lack of timed response provided the player sufficient time to carefully consider the accuracy of his annotation before submitting to the server.

4.6 Badges

When a player finishes a round in any of the three games, he or she receives a reward in the form of a badge. The badge provides information on how their annotation compares to other players contributions. We made use of two types of badges:

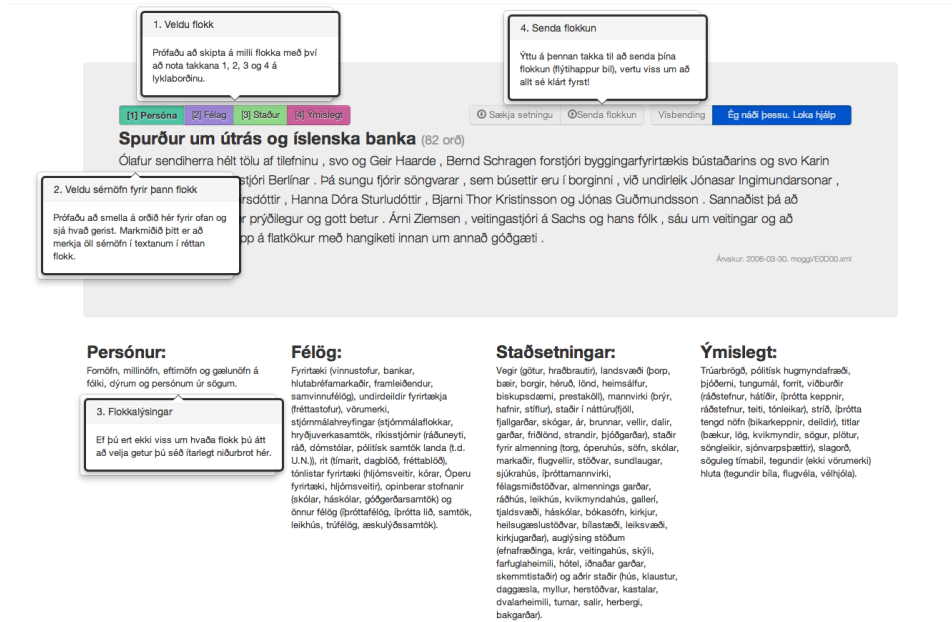
- *First annotation badge*: As explained in Section 4.1, once a player has annotated enough of text, he or she will possibly be provided with a sentence that has not been annotated before. For such a text it is impossible to provide any feedback on the players annotation, so the player is rewarded with a First annotation badge instead. The icon for the First annotation badge is displayed in Figure 7.
- *Followup badge*: If a player annotates a text that has already been annotated by other players he or she is rewarded with a Followup badge. The Followup badge contains an agreement score that indicates how much the player agrees with the rest of players on this particular sentence. The formula to calculate the agreement score is explained in Section 4.8. Figure 8 shows the six variations of the Followup badge, the icons vary depending on agreement score value.

4.7 Scoreboard

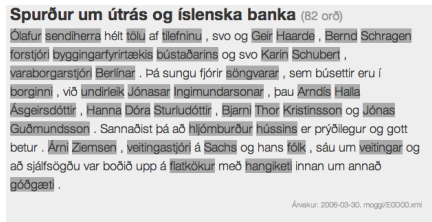
To allow players to keep track of their progress in the games, we designed a scoreboard. A scoreboard provides logged-in players with a way to observe their collection of badges and to measure their performance compared to others. The scoreboard page on the Orðspor website was divided into three parts: a public scoreboard, some annotation statistics and a private scoreboard.

Figure 9a shows the public scoreboard. As mentioned in Section 3.3.1, our hope is that this scoreboard would create a positive atmosphere of competition between players and acts as an encouragement for them to improve their gameplay and play more frequently.

Figure 9b show the annotation statistics table. Here one can see the distribution of labels between the four named entity categories. We believe that this kind of detailed information would be of interest to the most motivated players.



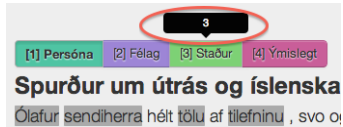
(a) A four step tutorial to begin using Orð-val. The tutorial is closed by pressing the wide blue button.



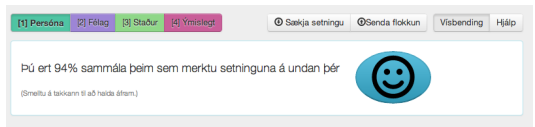
(b) Users have the option of highlighting nouns in the text



(c) Below the main text, user have access to concrete examples for named entity in each category



(d) Users are encouraged to use keyboard shortcuts through black tooltips (highlighted in red circle)



(e) After submitting an annotation, the user receives feedback on his annotation before continuing with the next sentence

Figure 6: An overview of Orð-val's user interface



Figure 7: The First annotation badge. Players who are first to annotate a text get this badge

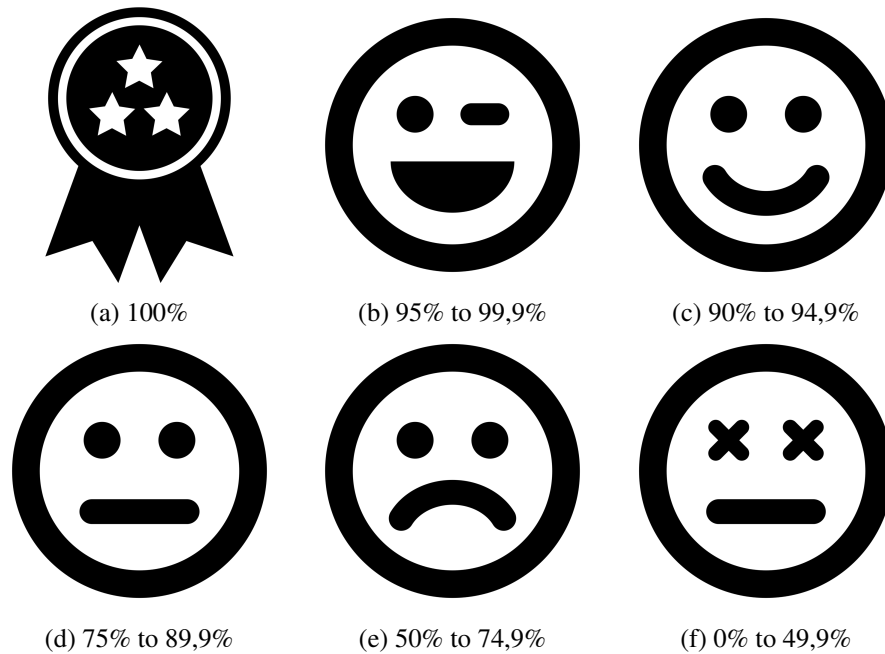
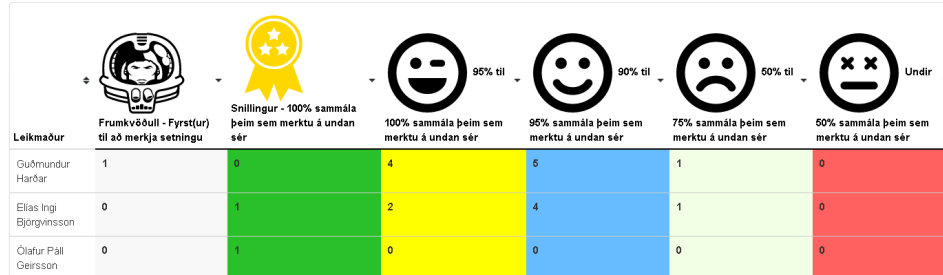


Figure 8: The Followup badges. The percentage of how much the player agrees with previous annotations is indicated below each icon.

Stigatafla allra

Hver hefur besta orðspóni?



(a) Users can track the badge collection of all players who signed up for an account, including their own collection.

Dreifing milli flokka

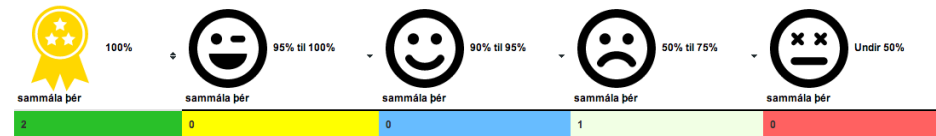
Hér sérðu dreifinguna milli flokkanna

Persóna	Félag	Staður	Ýmislegt	Allt annað (ekki sérnöfn)
6892	970	1492	1493	43042

(b) Users can see how many named entities have been tagged in each category.

Stigataflan þín

Þessi tafla sýnir hversu sammála aðrir eru þér í þeim setningum sem þú merktir fyrst



(c) Players who have been the first to annotate text are provided with the information on how other players annotate the same text.

Figure 9: An overview of Orðspor's scoreboard page.

Figure 9c shows the private scoreboard that is only visible to logged-in users who have received a First annotation badge. It seemed unfair towards the motivated users who receive a First annotation badge to not being able to follow how much other players agree with his first annotation. In the private scoreboard, a user can see how other players agree with the user's first annotation.

4.8 Output accuracy

It can be challenging to make sure that the produced output from the games is accurate. As explained in Section 3.3.2, we as game developers have a couple of ways to ensure that our players make an honest attempt to produce the most accurate annotations. The method we considered to be the best for our needs was repetition.

Repetition is the method where we make a handful of players annotate the same text and consider the sentence to be annotated when at least 50% of the players agree on the same result. In Orđspor, we call a produced label for a given token a “vote”. When we have two players who voted the same on the same input we consider the sentence to be correctly annotated. Although the final result with this method may be inaccurate at times, we consider that repetition offers a good balance between the expected contribution (taken from Section 3.4) and accuracy of the output. Furthermore, repetition has several advantages to the other methods explained in Section 3.3.2. First of all, repetition is simple to implement. We let the players annotate as usual and when a consensus has been made on a sentence, the API will stop returning that sentence on requests to `/v1/next-sentence`. Secondly, repetition works well with single player games. When a player visits our website he or she can start playing immediately, we do not have to let the player wait until we reach an even number of players online. Finally, since we let multiple people annotate the same sentence, we can provide them with feedback on how they they are doing in the game through agreement scores.

The agreement score gives our players an indication of how much they agree with other players on their annotation. The score is shown to the players in the form of a badge after they have successfully submitted an annotated sentence to the server. The formula to calculate the agreement score for a player a on a sentence S with n tokens is as follows: let S_i be the i -th token of S for some $0 < i \leq n$, P be the set of all players who have submitted an annotation to sentence S , C be the set of categories $\{PER, ORG, LOC, MISC, O\}$ (O denotes “other than a named entity”), p_{S_i} be the category in C which player $p \in P$ voted for token S_i and let

$$S_{i,c}^p = \begin{cases} 3 & \text{if player } p \text{ voted that } S_i \text{ belongs to category } c \in C \setminus \{O\} \\ 1 & \text{if player } p \text{ voted that } S_i \text{ belongs to category } O \\ 0 & \text{otherwise.} \end{cases}$$

Observe that we give a three times higher weight to votes on named entity categories than O . We can now define

$$\text{agreements} = \sum_{i=1}^n \sum_{p \in P \setminus \{a\}} S_{i,a,S_i}^p$$

and

$$\text{disagreements} = \sum_{i=1}^n \sum_{p \in P \setminus \{a\}} \sum_{c \in C \setminus \{a,S_i\}} S_{i,c}^p.$$

Finally, we calculate the agreement score

$$\text{agreement score} = \frac{\text{agreements}}{\text{agreements} + \text{disagreements}}.$$

5 System components

The different pieces that make up a software system, both during development and production, are many and diverse. A collection of server-side and client-side components were needed to develop Orðspor. During development a number of tools and methods were applied, for the system to successfully come together. In this chapter we go through the key components and software development process of Orðspor.

5.1 Server-side

The Orðspor API (see Section 4.1), an integral part of the Orðspor project, is run server-side. The server-side is composed of two major components: the Play Framework web server and *CouchDB* database server.

5.1.1 Play Framework

The Orðspor web application was built using the *Play Framework*⁴. The Play Framework is an open source web application framework, which makes it easy to build web applications both with Java and Scala. It is built on Akka, a JVM-based toolkit that implements the so called actor model. “The actor model is a message-passing paradigm that resolves some of the major challenges of writing concurrent, scalable code for today’s distributed systems.” [Haines, 2013] Play minimizes its resources consumption and therefore allows for the development of highly-scalable applications with a simple and smart development support. For the Orðspor project, the Play Framework offered all that the project and the team needed.

The Play Framework is equipped with a number of built-in tools to handle tasks and simplify development. The build system is based on sbt⁵, which is a minimally non-intrusive build tool for Java and Scala projects. Just a few of the built-in tasks are: (i) compile and build, (ii) a testing framework and (iii) packaging for distribution. All of these proved a simple setup with the Continuous Integration server (see below in Section 5.4) and general development of the application.

⁴<http://www.playframework.com/>

⁵<http://www.scala-sbt.org/>

We used the *Play Authenticate*⁶ plug-in, as our log-in system and to keep track of our user base. The plug-in allowed for rapid development and out of the box functionality for common OAuth2 providers such as Facebook and Google as well as user-name/password account services, such as multiple accounts merging and password reset.

5.1.2 CouchDB

To manage our persistence layer we used CouchDB, a document-oriented non-relational database management system (DBMS). A non-relational DBMS we considered would offer us flexibility in how we would structure our data while developing Orðspor. However, the decision to use such an unconventional DBMS added some additional challenges to the development of Orðspor.

The CouchDB philosophy is *relax*. One is constantly reminded of this philosophy is repeatedly through the *CouchDB* name, logo, documentation⁷ and Stack Overflow⁸ discussion forums. Even when you start CouchDB, you see:

Apache CouchDB has started. Time to relax.

This great emphasis on relaxation comes from the idea that developer productivity is increased when tools are easier to use. At the beginning of our project, we were unsure of what this relaxed philosophy fully entailed. Once we started to understand CouchDB's core concepts however, we saw it became a powerful tool for making technical decisions. When we were unsure of which way we should solve a particular problem, we embraced the relaxed philosophy and chose the most “relaxed” solution.

Documents are CouchDB's central data structure. CouchDB documents are represented as *JSON*⁹ strings consisting of attribute-value pairs making real-world objects easily represented in the database. Being schema-free, the documents offered us freedom to quickly start small and then iterate and improve our model as the Orðspor project grew bigger.

We used the *Lightcouch*¹⁰ java library to integrate CouchDB with our Play web server. Despite being a relatively new library, at release version 0.0.6 when we first used it, the Lightcouch library gave us access to the majority of CouchDB features needed to develop Orðspor. However, we did struggle to find a simple way to organize our model with Lightcouch.

⁶<http://joscha.github.io/play-authenticate/>

⁷<http://docs.couchdb.org>

⁸<http://stackoverflow.com/questions/tagged/couchdb>

⁹A lightweight data-interchange format widely used in the modern web, see <http://json.org/>

¹⁰<http://www.lightcouch.org/>

We wrote a class `CouchModel` to abstract how model objects are stored in CouchDB. `CouchModel` employs the layer supertype pattern [Fowler, 2002, p. 475] by acting as the supertype for all types in the model layer. Amongst the services that `CouchModel` provides include:

- provide a shared `org.lightcouch.CouchDbClient` singleton object to handle connections to the CouchDB server
- provide a shared `com.google.gson.Gson` singleton object to allow for custom object JSON serialization and de-serialization and
- add an automatically generated GUID¹¹ `_id`, revision `_rev` and class type `type` fields for communicating with CouchDB documents.

After we implemented `CouchModel`, it became easy to define almost any type of data access object (even with complex inheritance rules) and out-of-the-box access to `save`, `update`, `find` and `remove` methods that talked to the database.

CouchDB offers a powerful way to create views on top of your database with *map* and *reduce* functions. A map function produces a map result which is effectively a list of key-value pairs. Optionally, one can use a reduce function to produce a single scalar value from a map result. The Orðspor API makes extensive use of CouchDB views, in particular to (i) order sentences by complicated key values, (ii) calculate agreement scores and (iii) aggregate statistics for the scoreboard page. Getting map/reduce functions to produce the desired result did require some experimentation within Futon, a web interface distributed with CouchDB. To synchronize our map/reduce functions to the CouchDB server we used CouchApp¹², a python command line utility for managing CouchDB design documents.

5.2 Client-side

Modern browsers coupled with *HTML5*, *CSS3* and *JavaScript* have become powerful tools and their usages has extended greatly from showing simple static websites. By developing the Orðspor project as a web application, the dependency and restriction from operating systems were removed and public access greatly improved.

5.2.1 Orð-árás

The game is implemented in pure JavaScript and does not rely on external libraries. It follows a known design pattern for web based mini games that utilizes the canvas element in HTML5.

¹¹Global unique identifier

¹²<https://github.com/couchapp/couchapp>

A simplified description of the running cycle of the game is described in Algorithm 1:

Algorithm 1: Game loop for Orð-árás

```
initialize new game;
draw start screen;
while true do
    update state for game objects;
    draw state on canvas;
    if game over then
        draw result on end screen;
        quit game or play again;
```

5.2.2 Orð-skot

Utilizing the powers of the CSS3 and JavaScript were the main focus point in the development of Orð-skot. Orð-skot makes the most of CSS3 performance tricks by forcing the graphics card, instead of the central processing unit, to handle calculations for animations. Doing this, results in a faster rendering time on each animation frame and therefore a positive experience for the user both visually and performance wise.

By making use of object creation patterns¹³ for the JavaScript code, it becomes more efficient and maintainable. The JavaScript code for Orð-skot uses constructor functions and adds functions to the prototype chain of an object. A proposed class diagram for Orð-skot can be viewed in Figure 10.

In order to hide and show components of the website during the game play and to make the user experience more enjoyable, the *HideShow* constructor functions was made. Both Orð-skot and Orð-árás utilize the prototype functions of *HideShow*, mainly so that an active game is the main focal point on the screens of mobile devices.

¹³A clear and concise explanation on the subject may be found here: <http://www.codeproject.com/Articles/687093/Understanding-JavaScript-Object-Creation-Patterns>

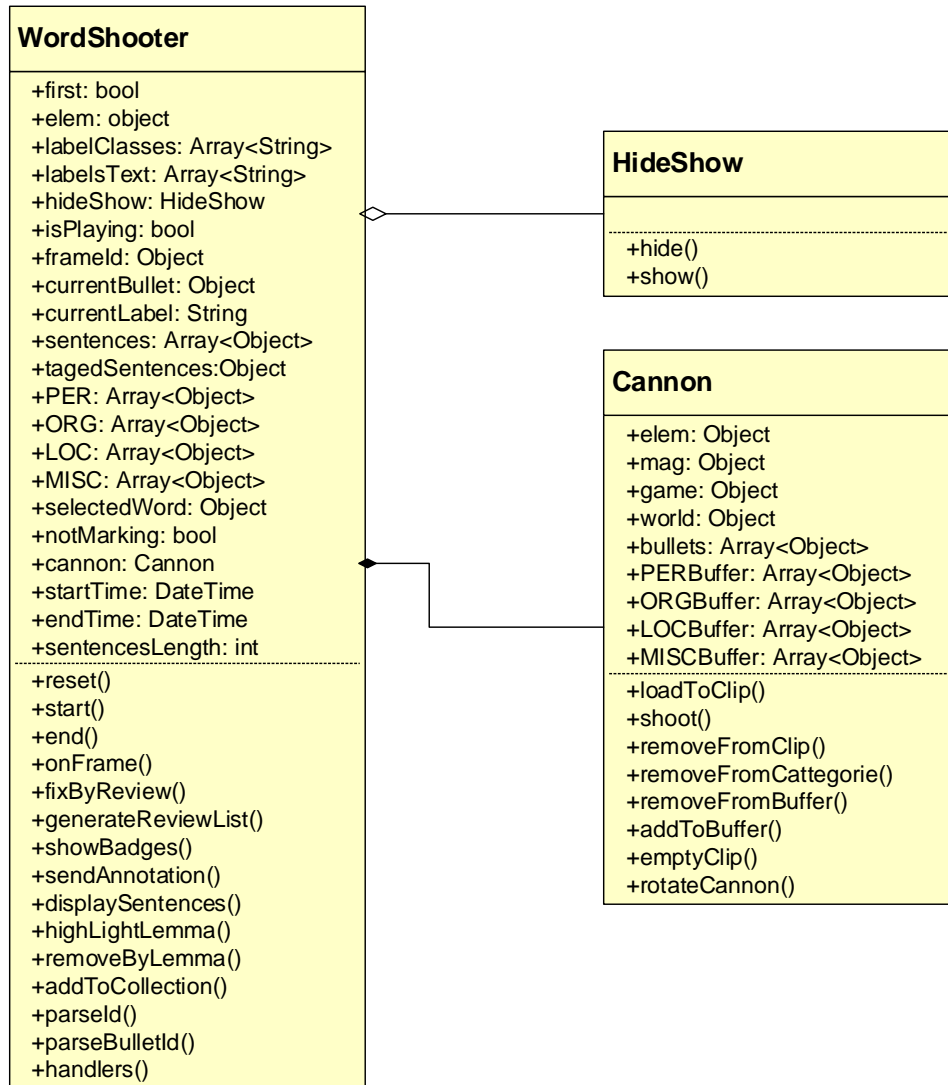


Figure 10: The *WordShooter* constructor functions and it's prototype functions handles the main mechanics of the game while *Cannon* handle the mechanics for the in-game cannon and it's functionality. The *start()*, *end()* and *reset()* functions are the building blocks of the game and the *onFrame()* function makes sure that all animation frames are updated. All of the *Cannon* prototype functions along with the *displaySentences()*, *addToCollection()* and *handlers()* functions handle all of the main game functionality. Other functions provide support regarding visual feedback and/or end-game word review.

5.2.3 Orð-val

Orð-val is a relatively simple web app. The app is written in *Coffeescript*¹⁴ and makes extensive use of *jQuery*¹⁵. Coffeescript is a small language that compiles to JavaScript. The language offers a variety of syntactic sugar features on top of JavaScript. Besides the syntactic sugar features however, Coffeescript is merely plain old JavaScript. The Play Framework offers a convenient integration with Coffeescript through automatic compilation shortening each iteration of writing code and testing it on the browser. The majority of behavior in Orð-val is implemented with jQuery through click and keypress events. jQuery is a JavaScript library that makes it simple to write cross-browser compatible web apps.

5.3 Version control

In every software project, some kind of version control system is essential in order to simplify the file management. For the Orðspor project we used *GitHub* as the version control system. While developing Orðspor, the team followed a centralized workflow. We used a single repository with a single master branch that all team members worked off from on their local development machines. Although this kind of workflow is not sufficient for some teams, it proved more than sufficient for our needs.

5.4 Continuous Integration

In order to keep the code production ready from the start, a Continuous Integration server was needed. Continuous Integration is a methodology where the software is built in a consistent and known environment to avoid dependencies on local developers machines. A number of Continuous Integration tools are available for different platforms and programming languages. *Jenkins*¹⁶, formerly known as Hudson, is an open-source Continuous Integration tool that can be customized for each project through a variety of plug-ins.

Jenkins was a perfect fit for Orðspor. A server running Jenkins was set up and modified to build and unit test the project on each push on the GitHub repository. If the project managed to build and all unit tests were successful, Jenkins made a distribution package that was used to deploy the website manually. The build and test status of the project was indicated on the projects GitHub repository website, so developers would be informed on the build status of the project. The server

¹⁴<http://coffeescript.org/>

¹⁵<http://jquery.com/>

¹⁶<http://jenkins-ci.org/>

administrator was as well informed via E-mail if a build failed due to compile errors or failing tests.

Without a physical server machine, a web application can not serve any clients. During the fall semester of 2013, *Greencloud*¹⁷ provided each student in the Software Engineering (T-303-HUGB) course with a free virtual server machine running *Ubuntu Server 12.10*. Two instances of such machines run the Orðspor system, one runs both the Jenkins Continuous Integration server as well as the web application server while the other one runs the CouchDB database.

5.5 Team Communication

In software development, it is essential that all members share a common vision. One of the members in our team, Guðmundur Harðarson, lives in the eastern part of Iceland and could therefore not be physically present at our team meetings. We were determined that our team's communication should not be restricted by the distance between us.

In order to share and convey ideas between team members, daily meetings were held. The meetings happened on *Skype* and informal meeting reports were noted in *Google Drive* documents. This way, all members of the team could express their thoughts and ideas.

During the starting stages of the project, we identified user roles. The complete list of user roles is shown in the Appendix as Table 13. Following this, we constructed a requirements list that served as a product backlog as is shown in Appendix as Table 14. In the product backlog, we identified main components of the core functionalities of the system through *User Stories* and the Acceptance tests elaborate on each User Story. While writing the User Stories, we used a template provided by Mike Cohn¹⁸. The user stories and the product backlog proved to be useful for our team to prioritize on feature implementations.

During the recent years the need for software and tools that support global communications has grown rapidly. The Orðspor team, as noted above, used *Skype* for daily meetings and *Google Drive* for simple text documents. Additionally to the above mentioned tools for communication, we used a Facebook group as a messages board, *Google Drive* to collaborate on spreadsheets and *Trello*¹⁹ as an agile task board. *Trello* is a simple project management tool that allowed us to manage our work online as if we were using a physical whiteboard and a stack of

¹⁷<https://www.greencloud.com/>

¹⁸<http://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template/>

¹⁹<http://trello.com/>

task cards by our desk. This functionality provided by Trello proved useful for the team, both to keep track of the project and to see who was working on what task.

6 Experimental results

What is the correct utility measure of the GWAP exercise? One primary measure of success is the amount of correct data the players will contribute. A measure of GWAP quality is *expected contribution* [von Ahn and Dabbish, 2008], defined as the number of problems solved per human hour spent in the game.

6.1 Dataset

The dataset we used for our experiment is *MÍM* (abbreviation for “Mörkuð íslensk málheild”, e. Tagged Icelandic corpus) [Helgadóttir et al., 2012]. We selected 3596 documents containing roughly 1.4 million tokens. The source for our annotations is news-wire articles from “Morgunblaðið”, the most visited news media website in Iceland. The content includes categories such as: General news, sports, politics and death announcements. We had two important reasons for selecting a text with diverse content. Firstly to get a broad coverage of named entities and secondly to show diversified text to the players.

6.2 Marketing

Our marketing strategy was modest and did not reach beyond our social network. We relied on word of mouth and Facebook shares to get the website out to public. A handful of people re-shared our Facebook posts about the website and shared a link to our website. Keeping this in mind lets look at the results.

6.3 Empirical results

We present data from two sources, the output from games and traffic analytics in the Orðspor website.

A comparison of the output from the games is shown in Table 3. Orð-val has the most number of annotated sentences while Orð-áras and Orð-skot have a similar amount of output.

Data collected by the use of *Google Analytics*²⁰, which monitored the websites traffic, can be seen in Tables 4 to 12 and in Figure 11. The data presented is from the period from 12th to 17th of May 2014. Here are explanations, borrowed from Google Analytics, on the terms presented:

²⁰<http://www.google.com/analytics/>

# Annotated	Games			Total
	Orð-val	Orð-áras	Orð-skot	
Words	40.257	8.734	9.056	58.047
Named entities	8.449	1.104	2.271	11.824
Sentences	252	59	53	364

Table 3: Output from Orð-val, Orð-áras and Orð-skot

- *Sessions*: Total number of Sessions within the date range. A session is the period time a user is actively engaged with your website, app, etc. All usage data (Screen Views, Events, Ecommerce, etc.) is associated with a session.
- *New Sessions*: An estimate of the percentage of first time visits.
- *Session Duration*: The length of a session. After a session starts, it can last as long as activity is tracked. By default, a web session is closed after 30 minutes of inactivity. For mobile apps, a session times out after 30 seconds of inactivity or with the app running in the background. Any activity that happens after a session is closed is tracked in a new session.
- *Pageviews*: The total number of pages viewed. Repeated views of a single page are counted.
- *Unique Pageviews*: The number of visits during which the specified page was viewed at least once. A unique pageview is counted for each page URL + page Title combination.
- *Time on Page*: The total amount of time spent on a specified page.

Date	Ratio
12.5.2014	76.29%
13.5.2014	70.99%
14.5.2014	45.76%
15.5.2014	31.58%
16.5.2014	0%

Table 4: New Sessions by Date. An estimated percentage of how many of the sessions each day were new visitors.

Date	Duration
12.5.2014	16:11:57
13.5.2014	09:30:39
14.5.2014	05:31:23
15.5.2014	04:42:55
16.5.2014	00:21:32

Table 5: Total session Duration by Date.

Date	Sessions
12.5.2014	232
13.5.2014	162
14.5.2014	59
15.5.2014	38
16.5.2014	16
Total	507

Table 6: Sessions by Date

Browser	Sessions
Chrome	382
Safari	34
Firefox	33
Safari (in app)	21
Android Browser	20
Internet Explorer	16
Total	506

Table 7: Sessions by Browser

Page	Unique Views
/ord-val	195
/ord-aras	152
/ord-skot	86
Total	433

Table 9: Unique Pageviews by Page

Date	Pageviews
12.5.2014	900
13.5.2014	464
14.5.2014	173
15.5.2014	141
16.5.2014	30
17.5.2014	4
Total	1712

Table 11: Pageviews by Date

Device	Sessions
Desktop	382
Mobile	111
Tablet	18
Total	511

Table 8: Sessions by Device Category

Page	Time
/ord-val	05:04:34
/ord-aras	04:48:42
/ord-skot	01:19:55
Total	11:13:11

Table 10: Time on Page by Page

Country/Territory	Sessions
Iceland	469
United States	14
Denmark	6
Finland	6
Netherlands	4
United Kingdom	2
(not set)	2
Austria	1
Brazil	1
Switzerland	1
Sweden	1
Norway	1
Spain	1
Japan	1
Svalbard & Jan Mayen	1
Total	511

Table 12: Sessions by Country/Territory

Tables 4 to 6 serve as testimony that some form of repeating visits from some users occurred. One could conclude that the games appealed enough for people to return back to the website and continue playing the games.

Tables 7 and 8 provide us the a fun and educational information on through

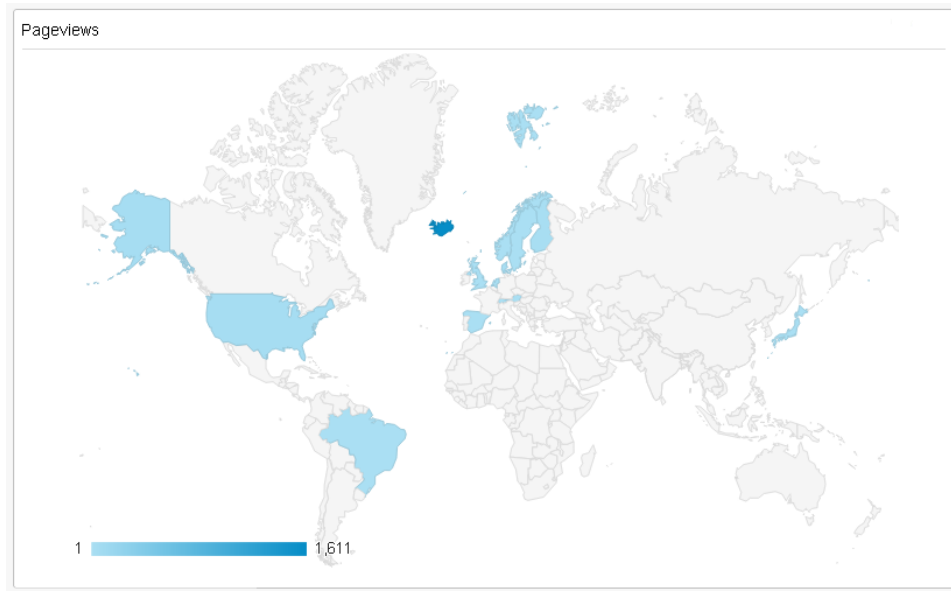


Figure 11: Color scale of pageviews per Country/Territory

which devices and browsers users accessed our website. Of the 511 sessions conducted to the site, an estimated 25% was through touch screen devices. And the “market value” of Chrome is around 75% of our user network.

As seen in Tables 9 and 10 the total play time of the games combined sums up to just over 11 hours in 433 unique pageviews, just in five days. We are happy that our games managed to mobilize that amount of play time.

Table 11 shows that the total number of pageviews to be 1712, which is the number of all pages that are in the website that were viewed. Repeated viewing of each page by each user is also counted to this number.

According to Figure 11 and Table 12, one could extrapolate that our network of friends and family extends throughout the world. We assume most of the visits to the website were by Icelanders and people that know the language well enough to annotate named entities correctly, given that all of the websites instructions is in Icelandic. As expected the majority of visits were conducted from Iceland.

6.4 User experience

Several informal user tests were conducted to evaluate the user experience of the website and the three games. The time allocated to each test was 15–30 minutes. The observer asked the user to open the website and play the games. While the user

played the games, the observer observed, mostly silently. The observer answered questions from the user if something was unclear regarding the functionality of the games or the representation of the scores or other website material.

The main remarks noted from the user tests where:

- Users were not sure what to do in the games.
- Sessions were too long.
- Gameplay is overly simple.
- The games could be fun for children improving their skills in Icelandic.
- More instant feedback on players performance.
- Get rid of punctuations and commas, especially in Orð-árás.

7 Learning outcomes

Developing a web application can be a big task to undertake, even more so if the development of three games is added to it. Through the development of Orðspor the team members expanded their knowledge in various fields.

7.1 Design and develop a web application

From the start, the team believed that to build the application with a framework would be the best solution. The Play Framework was chosen as the building block for the Orðspor project. Some of main learning lessons taken from using the Play Framework include how to (i) work with multiple Java libraries at the same time, (ii) employ unit testing to ensure software quality, (iii) implement non-blocking asynchronous function calls in Java and (iv) apply enterprise software design pattern in a non-trivial setting. Moreover, some additional lessons were taken from the integration between the Play web server and CouchDB.

Our decision to use CouchDB for our persistence layer was largely taken out of curiosity. Non-relational, or NoSQL, databases appear to be a popular choice for developers today and we were excited to become familiar with this technology. Our decision led us on a path to learn new and unfamiliar topics such as map/reduce functions, B+ trees and database replication. We could have taken a safe approach and use a Postgre relational database instead, but we are confident that we would then not have learned as much from the whole process.

Even though some team members had made used of a Continuous Integration tool before in a previous project, that particular tool was not applicable for use with

the Play Framework and the set-up of the Orðspor project. The Jenkins CI proved a good fit for the Orðspor project. Going through the set up and customization of a Jenkins Continuous Integration server, through a variety of plug-ins, to fit our needs proved educational. Especially since a problem with a virtual server instance, resulted in a complete re-set up of the whole system. Customizing the server that ran Jenkins and Jenkins itself to prepare a distribution package of the project, on each successful build was also an enlightening process. The set-up of *WebHooks* from the GitHub repository to Jenkins, so that Jenkins would build and test the project on each push to the repository, provided useful information. So did the creation of a bash script to manually launch the web application using the distribution package from Jenkins, through the Ubuntu Server terminal.

7.2 Design and develop web-based games

The group had limited prior experience in game design and development so the making of the games proved both challenging and educational. Considering that the three games were developed separately by each of the three team members, we will share individually what we consider to be the main learning outcomes from developing our games.

7.2.1 Orð-árás

In retrospect, my life could have been made easier by using a well established game-engine for the development of Orð-árás. Making such a decision could have resulted in more compelling gameplay and design. Nonetheless, spending time learning how to make basic game components from scratch in JavaScript proved to be a fun and educational experience, both to gain knowledge of the building blocks of computer games and also a deeper understanding of JavaScript.

— Guðmundur Þ Guðmundsson

7.2.2 Orð-skot

Developing Orð-skot gave me an opportunity to recap ideas and methods on CSS3 and JavaScript that I learned in the Web-Programming II (T-427-WEPO) course. In particular, the tips and tricks taught on how to structure CSS documents, for responsiveness, and on CSS animations proved to be useful. Furthermore, my deepened understanding of JavaScript and its inner workings, initially touched on in the Web-Programming II course, proved to be helpful for my team members.

— Guðmundur Harðarson

7.2.3 Orð-val

This was my first experience in designing and implementing a user-interface from scratch. Having now gone through the experience, I've made the observation that it seems suboptimal to manually manage the app state while applying changes to the browser's DOM structure. This observation has given me interest to try out tools in the likes of (i) *AngularJS*²¹ that provides automatic data-binding between the model and the view components and (ii) *React*²² that rerenders the whole DOM on every state change.

— Ólafur Páll Geirsson

7.3 Independent learning and research

We observed that progress in the project generally happened slower than what we had anticipated. Hopefully in the future, we will not fall into same trap and underestimate the effort needed to do complete something that we have never done before. We went through the steps of designing the project from scratch, to implementing the project, releasing the website and write about the experience. We believe that this hands on experience will prove valuable in our future studies and work.

8 Future work

While developing Orðspor, we implemented most of the requirements that we set out with in the beginning. A lot of new ideas arose through the development and sadly some of them did not make it into the final product. In this chapter we go over some requirements that were not implemented as well as improvement suggestions.

8.1 Games

It is a challenging task to create an interesting game, especially when the main objective of the game is to get valuable data from players. The following items are taken from feedback we received from our players:

- Make the sessions shorter and give more frequent feedback to players on how well or badly they are doing.
- Sweep away or remove words that are not named entities in Orð-árás.

²¹<https://angularjs.org/>

²²<http://facebook.github.io/react/>

- Get rid of punctuations and commas in Orð-árás.
- Make it possible to fire many words to the same category with one shot from the cannon.
- Provide players with the information on how their annotations on given text were different from the annotations of other players and introduce a voting system on the annotations.

The following items are our own suggestions for improvements:

- Create a browser extension that enables users to annotate text while they are surfing and reading material on the web.
- Implement pause functionality for Orð-árás and Orð-skot, possibly showing static tutorials or help while paused.
- Implement a multiplayer version of some of the games, Orð-val could for instance become an output agreement game, where the two player can only advance to a new round once they agree on the annotations.
- Improve the responsive design for Orð-val and Orð-skot. Orð-skot's game play is very dependent on the in-game hot-key system and Orð-val's tool box is static to the top of the page.
- Introduce an interactive tutorial to teach players how to play the games with a predefined dataset.

8.2 System and API

The foundation for Orðspor is it's API and web application. Some requirements for the core system did not make it into the final product, such as a private interface for a researcher for various customizations and controls. Making the API, and the whole system, ready for a general release as an Open Source Named Entity tagging tool template, will also require extended work.

- Implement a back-end for researchers with taxonomy controls and text upload and annotated data downloads.
- Let the system specifically handle mischievous annotators.
- Mechanism for ranking annotators, that assigns more weight on annotations and votes from skilled players.

- Release the API and other code as a open source template for a Named Entity tagging tool.
- Make use of news RSS feeds, to provide new text for annotation.

9 Conclusions

The aim of this thesis was to explore the use of computer games and crowdsourcing as a cheap and fun method to collect labeled data for the named entity recognition (NER) task. We defined the NER task, the different approaches to developing NER systems and the challenges that constrain such development. We touched on the different taxonomies for NER systems and how the language-independent named entity recognition shared task at CoNLL used four categories (i.e., person, organization, location and miscellaneous) that served as the taxonomy for our project. We have also described the concept of a “game with a purpose” (GWAP) and how it can be applied to collect large amounts of meaningful data. Although GWAPs have been successfully applied in various domains they have, to our best knowledge, not been applied to the NER task.

Our contribution is Orðspor, a website with three computer games, Orð-val, Orð-árás and Orð-skot, to collect named entity tags. The games are similar in many ways; they share a common API, they receive input from the same source and their output is merged to produce a single dataset. During the first week while the website was live, over 80 player annotated over 55.000 word in the period of almost 11 hours. We described the system components and infrastructure that run Orðspor, both in development and production. We furthermore detail the methodologies we employ from software engineering, such as continuous integration, RESTful API design and patterns of enterprise application architecture.

Moreover, we stated what we learned from this project, what we thought went well and recognized opportunities for improvement, both in this project and for future projects. We see room for improvements of the Orðspor API and the games, identified both by ourselves and our players, and outline which extensions to the system could be worth exploring.

We consider that the Orðspor website and its components can serve as a valuable asset to produce labeled data sets for the named entity recognition task as well as other natural language processing tasks. By developing a generic API, we were able to rapidly develop three games and experiment with different game designs. We envision that many more games could be developed with our API and attract a broad range of players. Such games could with proper marketing effort be embedded into people’s daily routines in various ways producing a wealth of datasets for use in a variety of natural language processing tasks.

References

- [Collins, 2002] Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, page 1–8. Association for Computational Linguistics.
- [Deterding et al., 2011] Deterding, S., Dixon, D., Khaled, R., and Nacke, L. (2011). From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, page 9–15. ACM.
- [Estellés-Arolas and González-Ladrón-de Guevara, 2012] Estellés-Arolas, E. and González-Ladrón-de Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Journal of Information science*, 38(2):189–200.
- [Fielding, 2000] Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California.
- [Finkel et al., 2005] Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, page 363–370.
- [Fowler, 2002] Fowler, M. (2002). *Patterns of enterprise application architecture*. Addison-Wesley Longman Publishing Co., Inc.
- [Haines, 2013] Haines, S. (2013). Open source Java projects: Akka. *JavaWorld*. Retrieved May 13, 2014, from <http://www.javaworld.com/article/2078775/scripting-jvm-languages/open-source-java-projects-akka.html>.
- [Helgadóttir et al., 2012] Helgadóttir, S., Svavarsdóttir, A., Rögnvaldsson, E., Bjarnadóttir, K., and Loftsson, H. (2012). The Tagged Icelandic Corpus (MÍM). *Language Technology for Normalisation of Less-Resourced Languages*, page 67.
- [Johannessen et al., 2005] Johannessen, J. B., Hagen, K., Haaland, \., Jónsdóttir, A. B., Nøklestad, A., Kokkinakis, D., Meurer, P., Bick, E., and Haltrup, D. (2005). Named entity recognition for the mainland Scandinavian languages. *Literary and Linguistic Computing*, 20(1):91–102.

- [Jurafsky et al., 2000] Jurafsky, D., Martin, J. H., Kehler, A., Vander Linden, K., and Ward, N. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, volume 2. MIT Press.
- [Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- [Loftsson and Rögnvaldsson, 2007] Loftsson, H. and Rögnvaldsson, E. (2007). IceNLP: A Natural Language Processing Toolkit for Icelandic. In *Proceedings of interspeech*.
- [Miller et al., 1998] Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schwartz, R., Stone, R., and Weischedel, R. (1998). The Annotation Group. Algorithms that learn to extract information BBN: description of the sift system as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, volume 17.
- [Nadeau and Sekine, 2007] Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26.
- [O'Reilly, 2011] O'Reilly (2011). Web 2.0 Summit 2011: Andrew Stalbow, "High Order Bit". Retrieved May 15, 2014, from <http://youtu.be/8Wnrc6HKLO8?t=2m40s>.
- [Poesio et al., 2012] Poesio, M., Chamberlain, J., Kruschwitz, U., Robaldo, L., and Ducceschi, L. (2012). The Phrase Detective multilingual Corpus, release 0.1. In *Collaborative Resource Development and Delivery Workshop Programme*, page 34.
- [Pratt, 2014] Pratt, C. (2014). Flappy Bird is proof that no one knows what the audience wants. *Polygon*. Retrieved May 12, 2014, from <http://www.polygon.com/2014/2/5/5382450/flappy-bird-is-proof-that-no-one-knows-what-the-audience-is-looking>.
- [Rabiner, 1989] Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.
- [Ratinov and Roth, 2009] Ratinov, L. and Roth, D. (2009). Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL '09*, page 147–155, Stroudsburg, PA, USA. Association for Computational Linguistics.

- [Scharl et al., 2012] Scharl, A., Sabou, M., Gindl, S., Rafelsberger, W., and Weichselbraun, A. (2012). Leveraging the Wisdom of the Crowds for the Acquisition of Multilingual Language Resources.
- [Siorpaes and Hepp, 2008] Siorpaes, K. and Hepp, M. (2008). Games with a Purpose for the Semantic Web. *Intelligent Systems, IEEE*, 23(3):50–60.
- [Tjong Kim Sang, 2002] Tjong Kim Sang, E. F. (2002). Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL-2002*, page 155–158. Taipei, Taiwan.
- [Tjong Kim Sang and De Meulder, 2003] Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, page 142–147. Association for Computational Linguistics.
- [Tryggvason,] Tryggvason, A. Named Entity Recognition for Icelandic Research Report. Student’s Bachelor’s thesis. Retrieved May 26, 2014, from http://www.ru.is/~hrafn/students/BScThesis_NamedEntityRecognitionforIcelandic.pdf.
- [von Ahn, 2009] von Ahn, L. (2009). Human computation. In *Design Automation Conference, 2009. DAC’09. 46th ACM/IEEE*, page 418–419.
- [von Ahn and Dabbish, 2004] von Ahn, L. and Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, page 319–326.
- [von Ahn and Dabbish, 2008] von Ahn, L. and Dabbish, L. (2008). Designing Games with a Purpose. *Commun. ACM*, 51(8):58–67.
- [von Ahn et al., 2006a] von Ahn, L., Ginosar, S., Kedia, M., Liu, R., and Blum, M. (2006a). Improving accessibility of the web with a computer game. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, page 79–82.
- [von Ahn et al., 2006b] von Ahn, L., Kedia, M., and Blum, M. (2006b). Verbosity: a game for collecting common-sense facts. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, page 75–78.
- [von Ahn et al., 2006c] von Ahn, L., Liu, R., and Blum, M. (2006c). Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, page 55–64.

A API

Listing 2: Example response from sending an HTTP GET request to next-sentence

```
{
  "_id": "20e6d5ca-64a3-4314-836c-e6a1e48dd5da",
  "_rev": "12-25b8ccbec5d716ed85f68936ea6e7ed7",
  "file": {
    "corpus": "MÍM",
    "name": "moggi/E0D40.xml",
    "title": "Kristjana Ragnheiður Ágústsdóttir",
    "subTitle": "",
    "publisher": "Árvakur",
    "author": "",
    "publishDate": "2007-01-02",
    "_id": "da7b7f61-be68-4cd1-bb06-83443aa58518",
    "type": "TEIFile"
  },
  "index": 2,
  "nrOfNamedEntities": 61,
  "length": 229,
  "lastHandout": 1399850741959,
  "annotationsSubmitted": 3,
  "tokens": [
    {
      "pos": "nven-s",
      "lemma": "kristjana",
      "text": "Kristjana",
      "type": "TaggedWord"
    },
    {
      "pos": "sfm3en",
      "lemma": "gifta",
      "text": "giftist",
      "type": "TaggedWord"
    },
    {
      "pos": "ta",
      "lemma": "26.",
      "text": "26.",
    }
  ]
}
```



```

    "type": "TaggedWord"
  },
  {
    "pos": "nkeo",
    "lemma": "febrúar",
    "text": "febrúar",
    "type": "TaggedWord"
  },
  {
    "pos": "ta",
    "lemma": "1956",
    "text": "1956",
    "type": "TaggedWord"
  },
  {
    "pos": "nkeþ-s",
    "lemma": "magnús",
    "text": "Magnúsi",
    "type": "TaggedWord"
  },
  {
    "pos": "nkeo-s",
    "lemma": "skógur",
    "text": "Skóg",
    "type": "TaggedWord"
  },
  {
    "pos": "nkeþ-s",
    "lemma": "rögnvaldssynur",
    "text": "Rögnvaldssyni",
    "type": "TaggedWord"
  },
  {
    "text": ",",
    "type": "Punctuation"
  },
  {
    "pos": "nkee",
    "lemma": "vegaverkstjóri",
    "text": "vegaverkstjóra",

```

```

        "type": "TaggedWord"
    },
    {
        "pos": "ap",
        "lemma": "í",
        "text": "í",
        "type": "TaggedWord"
    },
    {
        "pos": "nkeþ-s",
        "lemma": "búðardall",
        "text": "Búðardal",
        "type": "TaggedWord"
    },
    { /* Rest of sentence cut out */ }
],
"type": "Sentence",
"firstAnnotatorId": "3c0c9f54-d6a7-1004-8d2e-
d4653cb1fc7e"
}

```

Listing 3: Example annotation object sent to /v1/annotate via an HTTP POST request

```

{
  "_id": "178413de3fce45318d64ec5495297c0a",
  "_rev": "1-fb375e7388120f058a84093e2a580d07",
  "s": {
    "file": {
      "corpus": "MÍM",
      "name": "moggi/E0D40.xml",
      "title": "Kristjana Ragnheiður Ágústsdóttir",
      "subTitle": "",
      "publisher": "Árvakur",
      "author": "",
      "publishDate": "2007-01-02",
      "_id": "da7b7f61-be68-4cd1-bb06-83443
aa58518",
      "type": "TEIFile"
    }
  }
}

```

```

},
"index": 2,
"nrOfNamedEntities": 61,
"length": 229,
"lastHandout": 1399849937893,
"annotationsSubmitted": 2,
"firstAnnotatorId": "3c0c9f54-d6a7-1004-8d2e-
  d4653cb1fc7e",
"tokens": [
  {
    "token": {
      "pos": "nven-s",
      "lemma": "kristjana",
      "text": "Kristjana",
      "type": "TaggedWord"
    },
    "text": "PER",
    "type": "TokenLabel"
  },
  {
    "token": {
      "pos": "sfm3en",
      "lemma": "gifta",
      "text": "giftist",
      "type": "TaggedWord"
    },
    "text": "O",
    "type": "TokenLabel"
  },
  {
    "token": {
      "pos": "ta",
      "lemma": "26.",
      "text": "26.",
      "type": "TaggedWord"
    },
    "text": "O",
    "type": "TokenLabel"
  },
  {

```

```

    "token": {
      "pos": "nkeo",
      "lemma": "febrúar",
      "text": "febrúar",
      "type": "TaggedWord"
    },
    "text": "0",
    "type": "TokenLabel"
  },
  {
    "token": {
      "pos": "ta",
      "lemma": "1956",
      "text": "1956",
      "type": "TaggedWord"
    },
    "text": "0",
    "type": "TokenLabel"
  },
  {
    "token": {
      "pos": "nkeþ-s",
      "lemma": "magnús",
      "text": "Magnúsi",
      "type": "TaggedWord"
    },
    "text": "PER",
    "type": "TokenLabel"
  },
  {
    "token": {
      "pos": "nkeo-s",
      "lemma": "skógur",
      "text": "Skóg",
      "type": "TaggedWord"
    },
    "text": "PER",
    "type": "TokenLabel"
  },
  {

```

```

    "token": {
      "pos": "nkep-s",
      "lemma": "rögnvaldssynur",
      "text": "Rögnvaldssyni",
      "type": "TaggedWord"
    },
    "text": "PER",
    "type": "TokenLabel"
  },
  {
    "token": {
      "text": ",",
      "type": "Punctuation"
    },
    "text": "O",
    "type": "TokenLabel"
  },
  {
    "token": {
      "pos": "nkee",
      "lemma": "vegaverkstjóri",
      "text": "vegaverkstjóra",
      "type": "TaggedWord"
    },
    "text": "O",
    "type": "TokenLabel"
  },
  {
    "token": {
      "pos": "ap",
      "lemma": "í",
      "text": "í",
      "type": "TaggedWord"
    },
    "text": "O",
    "type": "TokenLabel"
  },
  {
    "token": {
      "pos": "nkep-s",

```

```

        "lemma": "búðardall",
        "text": "Búðardal",
        "type": "TaggedWord"
    },
    "text": "LOC",
    "type": "TokenLabel"
},
{ /* Rest of sentence cut out */ }
],
"_id": "20e6d5ca-64a3-4314-836c-e6a1e48dd5da",
"_rev": "7-060c67ce00b4bf2fa6a5c64840cee8ac",
"type": "Sentence"
},
"agreements": 0,
"disagreements": 0,
"nrOfNamedEntities": 59,
"msPerWord": 325.73799126637556,
"annotator": {
    "username": "Anonymous",
    "id": "58861c74-d697-1004-847e-b0f1c3aeb1da"
},
"origin": "paper",
"date": 1399850178371,
"type": "NamedEntityAnnotation"
}

```

B User Roles and Product backlog

Researcher	A person that uses the website as a tool to get a text annotated according to his or hers rules. This person also has a great interest in the data generated by annotators through the website.
Annotator	A person that uses the website to play a game that revolves around annotating texts.

Table 13: Identified User Roles for the application

Priority	User Story	Acceptance Test	Finished
----------	------------	-----------------	----------

1	As an annotator, I want to be able to request a text for annotation from the server through a command line interface and send a response back to the server with my annotations so that I can start annotating text without using a browser.	Verify that the server can respond to a request for text to annotate and with enough information for the user to send a response back to the server complete annotations of the text	X
2	As an annotator, I want to be able to categorise named entities with less than 3-4 clicks from a given text through a web interface (mobile and desktop) so that they are annotated within that text.	Verify that it's possible to categorise named entity's from a text	X
3	As an annotator, I want to be able to select the named entity that I want to annotate so that I have control over which named entity in the text gets annotated by me.	Verify that a the named entities of the text are selectable for annotation.	X
4	As an annotator, I want to get a text when I begin annotation so that I can annotate that text.	Verify that a text block is provided to the annotator for annotation	X
5	As an annotator, I want to be able to get a new text to annotate once I finish annotating the given text so that I can continue playing this awesome game.	Verify that a new text is provided once a annotator finishes annotating given text	X
6	As an annotator, I want to be assured that my annotations are recorded by the server and no data is lost so that I don't have to worry that my work will be lost.	Verify that all annotations are recorded to the server instantly	X
7	As a researcher, I want to be able to request annotated (processed from annotations according to some simple heuristic) data from the server in xml format so that I can use that data in what ever purpose I choose.	Verify that the annotated data is exported to chosen data format	
8	As an annotator, I want to see visual difference in annotated text and unannotated text so that I get a visual conformation on my annotations.	Verify that a different colour code is applied to annotated text	X

9	As an annotator I want to be able to go to the website and with two clicks or less, start a new game instance so that I can start having fun annotating named entities.	Verify that a "start game" button shows up on the initial web page and that it starts a new game instance.	X
10	As an annotator, I want to be able to start playing the game without having an account so that I can annotate data without registering to the site.	Verify that it's possible to start playing the game without having an account	X
11	As a annotator, I want be able to quit a game at any given time so that I don't feel restricted to the game.	Verify that the user can quit, either by exiting the site or by using a quit button in the game, a game without having a bad impact on the website or the annotated data	X
12	As a annotator,I want to be able to take a tutorial on how to play the gameso thatI can feel safe that I'm playing it correctly.	Verify that the annotator can go throguh or watch a tutorial on how to annotate	X
13	As an annotator, I want to be able to log in to an account so that I can keep track of my annotations	Verify that the annotator can log onto the site and view his user specific data related to his / hers annotations	X
14	As an annotator, I want information about my annotations to be stored on my account so that I can view my progress.	Verify that data about annotations for each user are stored on his / hers account	X
15	As an annotator, I want to be able to see how many named entities I have annotated on the website so that I can tell my friends how awesome I am.	Verify that the annotator can view how many named entities he/she has annotated	X
16	As an annotator, I want my log in credentials to be secure so that no one can hack my account.	Verify that log in credentials are salted, encrypted and stored securely	X
17	As a researcher, I want my log in credentials to be secure so that no one can hack my account.	Verify that log in credentials are salted, encrypted and stored securely	X
18	As a logged in annotator, I want to be able to log out of my account so that I don't stay logged in for longer than I want to.	Verify that a annotator can log out of his/hers account	X

19	As an annotator, I want to be able to change my password and other information on my account so that I can change them for whatever reason.	Verify that it's possible for a annotator to change his settings and information on the site	X
20	As a annotator, I want my user name to be unique so that I am the only annotator with that username.	Verify that no two annotator can have the same username	X
21	As a researcher,I want to be sure that no mischievous annotator is having a negative effect on the obtained label integrity so that I am safe to use the output from the game for my research.	Verify that the game offers some mechanism to detect mischievous labellers.	
22	As a researcher, I want to be able to view statistics about the amount of annotated text so that I can get up to speed on that.	Verify that it's possible to generate reports on how much text has been annotated.	X
23	As an annotator,I want to hear a satisfying sound when I annotateso that I feel good while annotating.	Verify that some entertaining sound is played during annotation	X
24	As an annotator, I want to see a annotation high score board so that I can aim for the top and stay there.	Verify that a high score board is displayed on the main page	X
25	As a annotator, I want to be able to delete my account to the site so that no information about my account will be stored after that.	Verify that all annotator specific data is deleted (username, password, etc.)	
26	As an annotator, I want all text input boxes to be safely constructed so that I don't need to worry about hackers attacking the site.	Verify that all input text is parsed	X
27	As a researcher, I want to be able to customize the classification taxonomy so that I can use different types of taxonomies	Verify that it's possible to define and edit categories for annotation and the text can be marked according to the selected taxonomy.	

28	As a researcher, I want to be able to modify how many different annotators are needed to annotate a certain text so that I can have a certain confidence in the annotation correctness.	Verify that it's possible to modify how many different annotators will be asked to annotated the same text	
29	As an annotator, I want to be able to start playing without logging in and when I log in into my account my progress will be recorded to my account so that I don't have to worry about losing any data because I started playing and forgot to log in.	Verify that some mechanism records the progress of a annotator that is not logged in and that progress will be added to the account of a annotator, when he /she logs in	X
30	As an annotator, I want to be able to view an about page so that I know more about the purpose of the website.	Verify that there is a about page on the website and there are information regarding the sites purpose.	X
31	As a researcher, I want to be able to log into a administration page for the website so that I have access to some mechanisms of the website.	Verify that there is a administration part of the website that needs a log-in to access	
32	As a researcher, I want to be able to upload a text in a specific format through a web interface so that I can have that text annotated by users.	Verify that it's possible for the researcher to select and upload text to the site	
33	As a an annotator,I want to be able to see how accurately a computer can annotate the textso that I can follow up on the progress made with my annotations.	Verify that the website can show output from an automatic classification of named entities	
34	As an annotator,I want to be able to challenge a friend of mineso that I can play with my friends which I find more enjoyable than playing alone	Verify that an annotator can challenge a friend to a game	

Table 14: Product backlog. Most of the requirements for annotators were implemented but not all for researchers. An extensive user interface and control center for researchers proved not to be an important functionality for the application, as most of the desired functions for that were a accessible through the user interface of Couch db.

