Rebuilding the Tower of Babel – A GWAP to Smash the Lingual Divide

Dave Daly[1]

[1] Department of Computer Science and Informatics
University College Dublin
davedaly7@gmail.com

Abstract. The Tower of Babel is a web-based game modeled on the traditional children's game of Chinese whispers with an added multilingual element. The game challenges its players to translate sentences from one language to another. The Tower is a Game With Purpose (GWAP), in the background it is harvesting these translations for consumption elsewhere, as well as intending to use them to improve machine translation systems. Players are motivated to play the game both for entertainment and as a tool to aid their own language acquisition.

Keywords: GWAP, translation, collective intelligence

1 Introduction

1.1 Mechanical Translation

Below is an example of a translation of a sentence from a news article in The Irish Times[1].

1. English: Osama bin Laden was actively engaged in directing his far-flung network from the compound in Pakistan where he was killed, a senior US intelligence official said.

2. Irish: Osama bin Laden a bhí gníomhach i stiúradh a líonra bhfad-flung ón cumaisc sa Phacastáin áit ar tháinig sé, dúirt an oifigeach sinsearach faisnéis US.

3. English: Osama bin Laden who were active in conducting a far-flung network of the compound in Pakistan where he came, said a senior U.S. information.

Number 1 shows the original sentence. Number 2 shows Google Translate's translation of this sentence from English to Irish. Finally Number 3 shows Google

[1] Article accessed at http://www.irishtimes.com/newspaper/breaking/2011/0508/breaking2.html on 8/5/2011

Translate's translation of its own translation back to English[2]. While machine translation techniques have advanced greatly in recent years, the above example illustrates that such an approach to translation still has a long way to go before it reaches a standard that would be accepted of human translators. It was this problem that motivated an attempt to approaching the AI problem of language translation from the angle of Collective Intelligence, and more specifically through the medium of a Game With A Purpose (GWAP).

1.2 The Tower of Babel

The Tower of Babel is a collaborative game that is played through the web. It is modeled on the traditional children's game of Chinese whispers with an added multilingual element. Player's cooperate in chains translating sentences from one language to another, trying to introduce as few errors as possible. In the background the game harnesses the multilingual collective intelligence of its players, by recording and assessing their translations, with the intention of using them as input into the corpora of machine translation systems. Finally, the Tower also acts as a translation service by returning the best translations for particular sentences to those that first input them to the system.

1.3 Outline of the paper

The next section introduces some related work and research in the areas of Machine Translation and Games With A Purpose. Section 3 contains a more in-depth discussion of how the game works. Section 4 outlines the technologies used and some of the more interesting aspects relating to the game's technical design and implementation. Section 5 gives a brief evaluation of some of the site's initial results and proposes some ideas for how to build on the current prototype. Section 6 concludes the paper.

2 Related Work

2.1 Machine Translation

Machine Translation (MT) is a problem that has occupied computer scientists almost as long as the field of Artificial Intelligence itself. The traditional approach to solving the problem was to come up with a complicated set of linguistic rules and apply them as the principle components in creating an algorithm for translation. Following Makoto Nagao's seminal paper in 1984 [1], there was a slow shift away from such a paradigm to the more dominant paradigms of Example Based Machine Translation (EBMT) and Statistical Machine Translation (SMT) with which we are familiar today. The two paradigms differ on some points, but both of these approaches rely on

[2] Translations performed at http://translate.google.com/ on 8/5/2011

the use of a bilingual corpus of translation examples. For modern MT to work, there is a need for a database of both appropriate and high quality examples [2]. It is generally accepted that working with chunks of language at the phrase or sentence level provides the greatest results, and that the linguistic knowledge of the system is enriched as more examples are added to the database [3]. While the present author acknowledges that there is still some debate within the MT research community surrounding these issues, the intention of this section is to provide but a brief overview to motivate the problem at hand. Data sparseness and a lack of high quality translation pairs, particularly with regards to minority languages is a barrier that has been frequently cited to the advancement of machine learning approaches to MT [4]. Such sparseness, and the production of high quality and appropriate language pairs are issues the Tower of Babel seeks to address. It will do this by having its players both identify flaws in a corpus (i.e. flagging poor translations from services such as Google Translate) and provide quality human production of translation pair sentences and phrases to address these same flaws.

2.2 Games With A Purpose

Billions of hours are spent each year by people playing computer games. Despite huge advances over the past 50 years, computers still don't possess some of the basic skills of conceptuality and intelligences that humans take for granted. It was these two thoughts that lead Luis von Ahn of Carnegie Mellon University to found the genre of games with a purpose (GWAP). The basic concept behind a game with a purpose is to harness the collective intelligence of human computation by incentivising them to perform relevant computational tasks through the medium of computer game-play [5]. A range of games[3] developed at Carnegie Mellon, have enabled this collective intelligence approach to be applied to a variety of tasks at which traditional AI has failed, such as adding relevant metadata to images, music and videos, or adding common sense facts to the Open Mind Common Sense Database project [6].

2.3 Language Translation GWAPs

Again von Ahn seems to be leading the charge in this space. He unveiled his idea for a game a called Monolingo at the 2009 Human Computation workshop (HCOMP). The idea for this game was to enable a human powered system for translation, ironically powered by people that speak only one language. The idea was that the computer first uses a dictionary to get multiple translations for each word in the sentence. Using the words available, the human player then selects and forms a sentence that makes the most sense. However, the Monolingo game was never released and the site has since been pulled. It would appear that the project has been superseded by the Duolingo[4] project. At the time of writing this paper, this site has yet to be launched and the project is very much under wraps. At a recent series of Ted Talks, von Ahn spoke about Duolingo [7], it appears that the idea of the game is to be

[3] Visit www.gwap.com to play
[4] Due to be launched at duolingo.com in May 2011

a language learning tool as well as vehicle for translating web content into other languages. Essentially, he hopes to harness some of the translating power of 1.2 billion language learners worldwide, having them simultaneously translate web content while also aiding them in their language acquisition, all through the medium of game play. This project has only come to my attention since Tower of Babel was launched. There are a definite set of parallels to be drawn between the Duolingo project and the Tower. Both applications are aimed at language students, both can be played by beginners and more advanced students alike, and both require knowledge of at least two languages. However, until Duolingo is released, a full comparison between that project and the Tower of Babel cannot be drawn.

3 The Game

3.1 Overview

The Tower of Babel has been modeled on the traditional children's game of Chinese Whispers (known as Telephone to American children). In a game of Chinese Whispers the first player whispers a phrase or sentence to the next player. Each player then successively whispers what he or she heard to the next. The last player announces the statement to the entire group. Errors typically accumulate in the retellings, so the statement announced by the last player differs significantly, and often amusingly, from the one uttered by the first[5]. A commonly cited example of how a sentence might morph, when passed through a group is [8]:

Original: *"Send reinforcements, we're going to advance."*

End of Chain: *"Send three and fourpence, we're going to a dance."*

The Tower of Babel differs from Chinese Whispers in that the Tower is text based, rather than aural/oral, and it is played over the Internet. The added multilingual element means that errors are potentially introduced by players through mistranslation, as opposed to the mishearing of a sentence. Also introduced to the game is a scoring dimension, the purpose is to earn points by translating sentences from one language to another. Players cooperate in a chain, one translating from the source language to the target language and the next back to the source again. The closer the sentence produced at the end of the chain to the original sentence, the more points earned. The reward of points is intended to both motivate players to produce as accurate a translation as possible and to repeatedly play the game so as to earn more points.

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

3.2 The Theme

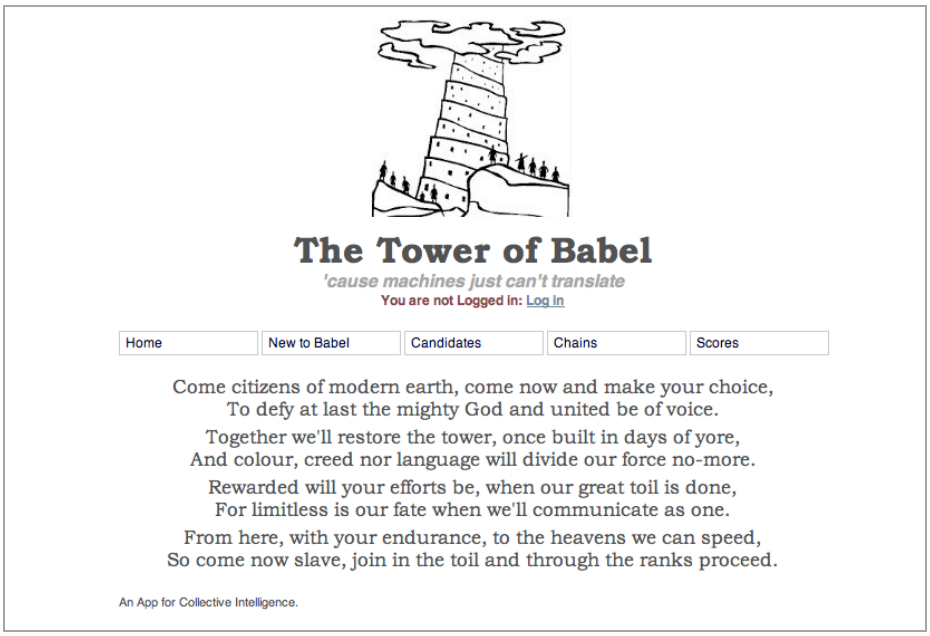

Fig.1 The Home Page

Included in the design of this game is theme centered on the biblical story of the Tower of Babel. The inclusion of role-play in computer games has existed since the production of the first video games in the early nineteen eighties, indeed role playing is so popular in video gaming culture that RPG has proved to be the most enduring genre in video games [9]. The biblical theme was selected as it was considered particularly pertinent to the purpose underlying the game. The basic story of Babel goes that back in the time following the flood of Noah, the whole world had one language. The people of the earth were so united that they decided to build a tower that would reach to heaven. God came to see the tower they were building and perceived their intentions as maligning him. He felt threatened by the powerful force within their unity of purpose. In a bid to retard their technological advancement, God confused their language, causing them to speak different languages so they would not understand each other, and scattered the people of the city all over the face of the earth[6].

As can be seen in figure 1, when players first visit the site they are greeted with verse challenging them to join the effort and unite in the modern era to rebuild the Tower of Babel (metaphor for a powerful machine translation tool). By giving players the sense that they are doing something more than just translating, in that they are contributing toward some greater fictitious purpose (the rebuilding of the Tower of Babel), it is hoped to keep them longer engaged with the game. This theme has

[6] The Bible - Genesis 11:1-9

influenced the look and feel of the site as well as the ranking system adopted by the game (see section 3.6).

3.3 Accumulating Source Sentences

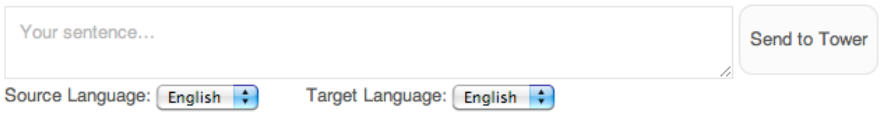

Fig. 2 Adding a Candidate

Sentences that are to be translated in the game are known as Candidates. Players, or guests to the site, can add candidate sentences to the system by going to the add candidate section (see figure 2), where they enter the sentence they wish to see translated, along with the source and target languages. It is intended that these will be sentences that they have tried to translate elsewhere (i.e. On BabelFish[7] or Google Translate[8]), and have been dissatisfied with the results. By adding a candidate, they are challenging the collective intelligence approach to translation to outperform the machine. A registered player to the site can later come and view the top scoring translations for their candidate sentences by visiting the *My Candidates* section of the site.

It is the desire of this author to see the game actually seeded with candidates coming directly from a popular translation service such as Google Translate. The simple addition of a button on the Google Translate page asking the user if they are satisfied with the results of their translation request could be used to log poorly translating sentences. These sentences could then be sent off to the Tower of Babel game as seed candidate sentences.

3.4 Playing the Game

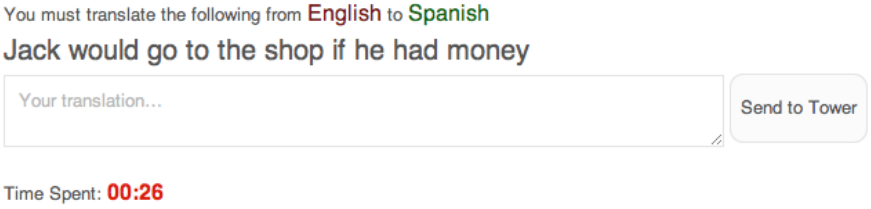

Fig. 3 A Translation Challenge

To play the game, players must *Start a Chain* or *Continue a Chain*. Each event provides a new translation challenge, with the latter building on the former to create a chain. When the player clicks on either start or continue a chain, they are presented with a sentence in a particular language and are challenged to translate it into another

[7] http://babelfish.yahoo.com/
[8] http://translate.google.com/

language (see figure 3). They enter their translation into the textbox and click the *Send to Tower* button when they are satisfied with their translation. Each translation challenge is timed and a timer is displayed under the input box indicating how long the player has spent on this particular challenge. Players are free to navigate away from their current challenge and can return to it at a later time through the *My Active Chains* section of the site. Players can be active in up to five chains at any one time, subsequent requests lead them to being informed that they should complete some of their current challenges before requesting more. A final point to note is that when registering, a player is asked to select a minimum of two languages with which they are familiar. Using this information, suitable translation challenges are served when requested, i.e. players will only be issued challenges requiring languages that concur with their list of familiar languages.

3.5 Scoring Points

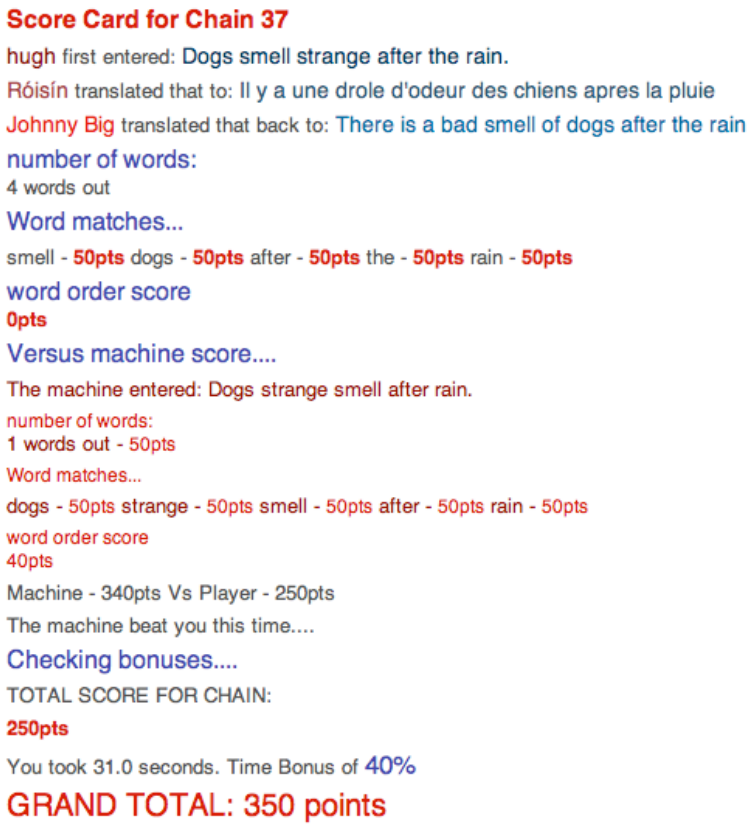

Fig. 4 A Scorecard

Points are rewarded to all players involved in a chain when it is fully completed, i.e. the chain has come full circle and has returned to the source language. Thus scores are

produced for a chain by comparing the original candidate and the sentence produced at the end of the chain. For those completing a chain, they see their score and score card instantaneously (see figure 4). However, for those that began a chain, they will only receive a score when another player completes the chain. These players are informed that they have new scorecards waiting the next time they log in to the site. The basic premise underlying the scoring algorithm is that the more similar the sentence produced at the end of a chain is to the original candidate sentence, the more points the players involved are rewarded. The scorecard outlines each of the steps in creating the score for a chain, they are as follows.

1. *Number of Words* – The number of words in the candidate and the produced sentence are counted. If they are the same number, 100 points is rewarded, lesser points are rewarded for there being 1 or 2 words difference. No points are rewarded if the difference is greater than 2 words.
2. *Word matches* – 50 points is rewarded for each word in the produced sentence that is found in the candidate sentence
3. *Word order* – Further points are rewarded depending on how many of the words are in the correct order
4. *Beat the Machine bonus* – The same translation is performed as a chain using the Google Translate API. This translation is assessed using the same criteria as in the 3 steps outlined above. If the score for the player chain equals or beats the machine score, they are rewarded 100 or 200 points respectively.
5. *Bonuses* – If the players match all of the words in the correct order, they receive a bonus of 100 points. If the produced sentence exactly matches the candidate (i.e. same capitalisation and punctuation as well as words), they receive a further bonus of 200 points.
6. *Time Bonus* – This varies for each player in the chain. A player receives an additional bonus as a percentage of the score already accumulated based on how long they spent on that particular translation challenge. For example, if they had completed their translation in less than 30 seconds, they receive a bonus of 50%. Time bonuses decrease in value to low of 5% for players who spent between 10 and 30 minutes on a translation. Translations that took more than 30 minutes receive no bonus.

3.6 Ranks and the Leader board

As players accumulate more points, they will rise in rank in the game. The ranks are synonymous with class stratification that existed in Babylonian society. A player starts out as a *Slave* and moves through ten levels including ranks such as *Labourer, Merchant, Architect* and *Temple Priest*. The top rank to be achieved is that of *Nimrod*, leader of the Babylonians, which is attained when a player has earned 20,000 points. A top-ten leader board can be viewed by navigating to the *Leaderboard* section of the site (see figure 5). This displays the current top ten scoring players along with their accumulated score, rank and the date they joined. The inclusion of ranks and a leader board have been shown to be an important incentive strategy in other GWAPs [10].

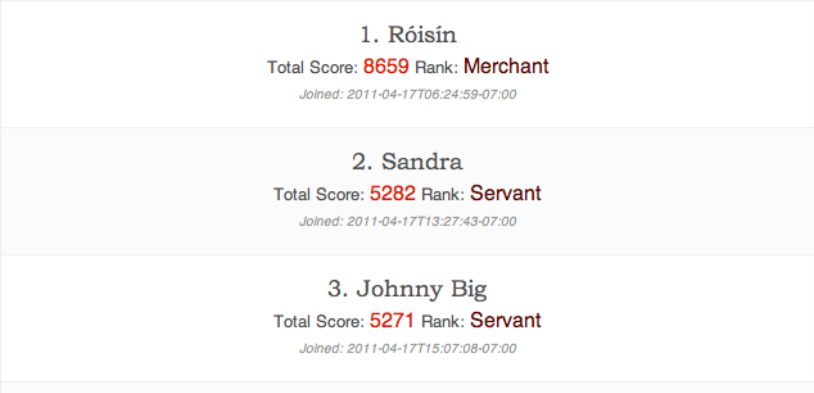

Fig. 5 Top of the Leader board

At least the stigma attached to being labelled a slave is proving enough motivation for players to try a few translations and thereby move off the bottom rung of the Babylonian society ladder. At the top end of the spectrum, the more avid enthusiasts of the game are encouraged to continue playing by competing for the top positions on the leader board. Some healthy competition among friends has also been observed, keen to prove themselves as having the greatest translation prowess among their peers.

3.7 Cheating

The Tower of Babel is a collaborative game: players cooperate in chains to achieve the best possible score. It is hoped that the game will be played honestly in the spirit that was intended, with players translating in their chains from one language to another and using their translated entries as the sole means of communication. However, in designing the game, the fact that players might collude to cheat or may try to use language other than that in which they have been instructed has not been overlooked. In such a scenario, not only would the competitive spirit of the game be in jeopardy, but also the data collected would be compromised. As outlined in the development of other GWAPs [11], it is best not to rely on the honesty of players and as such the inclusion of anti-cheating mechanisms should be considered. The following measures have been included in the game to encourage fairness and prevent cheating:

1. **Serving Translation Challenges.** Players will not be issued translation chains that have been seeded from candidates that they supplied. They will only be served any particular translation challenge on one occasion, and they will not be able to continue chains that they initiated. Translation challenges are served at random from all remaining options once the aforementioned have been deducted from possible challenges available. If there are no remaining options in the system when a challenge is requested (an issue that should only occur when there is a low number of users), they are informed of this and told to make a further request at a later time.

2. **Anonymity.** The identities of players involved in a particular chain, including the player that nominated the candidate, are concealed while a chain is in progress. The names of players involved are only revealed when a chain has been fully completed and scorecards have been issued.
3. **Language checking.** Players are instructed to use the languages specified. To ensure that all submissions are in the correct language, language detection is performed in the background using the Google Language Detect API. Should the system be unable to determine the language, the player is informed of this and asked to make another entry. If the player blatantly attempts to cheat the system by entering the same language as the source sentence in their challenge, they are penalised heavily (deducted 10% of their accumulated points) and their participation in that particular chain is terminated. The accuracy with which Google Language Detect can detect the language of a sentence is far from perfect. Performance particularly suffers in the context of short sentences or in less mainstream languages such as Irish. It does return a confidence level (between 0.0 and 1.0) on each detection it performs. By combining this with a requirement that sentences entered must be longer than 5 words, and excluding the use of minority languages other than Irish, a reasonably robust implementation of these rules has been implemented.
4. **Using Translation aids.** Preventing Players from using translation aids would be an impossibility, and besides it is hoped that players will produce the best possible translations. Thus the use of all available translation aids is encouraged, as clearly stated in *The Rules* section on the *How to Play* page. The obvious choice for a player playing online is to go to the Google Translate site, the service on which this game hopes to improve. The inclusion of a major bonus in the scoring algorithm for improvements upon the Google Translate attempt at the translation is aimed directly at such a player. If they simply use the Google translate offering, they are likely to lose out on this bonus. If they do however, take the Google translate offering and then work out how they can improve on it (perhaps using additional tools, or indeed, their own human understanding of how language works), they give themselves a strong opportunity of earning this bonus. Such behaviour is to be strongly encouraged among players, as it is so directly aligned to the backend purpose of the GWAP.

4 Implementation

4.1 Technologies

Ruby - It was decided to develop the game using the Ruby scripting language. Ruby was selected as it is the language with which the author is most familiar and as it is proving particularly popular in recent years for the development of web applications.

Sinatra – The next step was to select a web framework on which to develop the application. The Rails web framework has almost become synonymous with being the Ruby web framework. However, it was felt that for this particular application, the bulkiness that the Rails generated MVC code entails was considered to be overkill.

The author also had the desire to create all the code in the application himself, thereby increasing the amount he learnt during development. After some research, the Sinatra DSL (domain specific language) was selected. Sinatra is an extremely minimalist approach to Model View Controller (many would question whether it is even an MVC). It has been developed for use with Ruby, and seeks to do no more than simply allow for the coder to tie RESTful commands issued through the browser, to their application code and any views that might be created [12]. To highlight its simplicity included below, in its entirety, is a hello world web app written using Sinatra:

```
require 'sinatra'

get '/hi' do

   "Hello World!"

end
```

Use of Sinatra simply requires the installing of the gem and then requiring it in your application.

Datamapper – Rather than creating a separate database from scratch, an Object Relational Mapper (ORM) was used for data management. This allows for easy persistence of OOP objects created in the application to storage in a backend database. The Datamapper gem was selected as it has been specifically designed for use with Ruby, and its syntax was felt to be more elegant than other available options such as the Active Record gem. During development Datamapper objects were stored in an SQLite database, and in the production environment they are in a PostgreSQL database.

Google Language APIs – There were a number of operations within the application that required the use of language detection and language translation. Fortunately Google make available the use of their language APIs through http JSON requests. These were perfectly suited and seamless to use for these aspects of the application. For the language detection, a class was created that when instantiated makes a call to Google Language detect, passing the relevant text, and then uses the response data in creating the object with all the relevant data to the game exposed through its methods. For the translation tasks, a third party gem was used: ToLang, which is a ruby library that adds language translation methods to strings and arrays. This library uses the Google Translate API to perform its translations.

CSS, HTML and Javascript – The layout of the page was made from scratch using a combination of HTML for static content, embedded ruby for dynamic content, and CSS to achieve the look and feel. Everything right down to the drop down menus were styled using CSS. There was only one exception to this, which was a recursive JavaScript function written to display the timer.

Other Ruby gems – One of the most attractive features about using ruby in application development is the ease with which a wide range of gems can be included,

each offering solutions to particular problems the developer might come up against. Already outlined above is the inclusion of gems such as Sinatra, Datamapper and ToLang. In addition to these, several other gems were utilized, the json gem – to enable JSON passing with the Google API, the rack and rack-flash gems – used to enable flashing notifications to the user, httparty gem – allows for http requests to outside services, rspec gem – used for BDD testing, and the heroku gem – used for deploying the application directly from the project repository on github to the application server at heroku.

Heroku – The application was deployed on Heroku, a server exclusively dedicated to Ruby applications, which is also free for small applications, not costing a cent until there is a need to scale up. This was the obvious choice for the Tower of Babel as Heroku is well regarded in the Ruby development community. Unfortunately just as the game was starting to see some traffic shortly after the launch, an unprecedented event saw the site go down for almost four days[9]. While the pros and cons of cloud computing are not within the scope of this paper, it is felt that this outage of Amazon EC2 and the loss of associated AWS cloud services merits mention, due to the effect it had at a critical juncture in the lifetime of this web application.

4.2 Decisions about Design

A full outline of the design of the application is beyond the scope of this paper[10]. In this section a number of the more interesting aspects encountered during development are discussed.

4.2.1 Connecting players in real time games?

When the GWAP was originally conceived, it was envisaged that games would occur in real time, with players connecting over a game server performing as many translations tasks, concurrently, as they could within the limited time frame of a round. As the game developed, this design was discarded for a number of reasons. Firstly during the research phase, a number of other GWAP sites hosting multiplayer games were visited, and with the exception of von Ahn's offerings[11], all of them were barren of other players, leaving the (presumably sole) user waiting interminably for another player to play against. Secondly, if the player is under time pressure and if they are conscious that another player is waiting for them to enter something, the translations they produce are likely to be of a lower quality than if they know they have some time to work on them. Finally, there are host of technical issues that would need to be addressed to facilitate connecting players in a real time game, the cons of such an implementation were seen to far outweigh the benefits. So, it was decided that an implementation that would allow players pick up on another player's translation at

[9] See http://status.heroku.com/incident/151 for incident report
[10] Fully commented source code available at https://github.com/havingabath/sinatra_gwap
[11] Visit http://www.gwap.com/

an undetermined point in the future was a better scenario to pursue, thereby circumventing all of the aforementioned issues surrounding connecting players in concurrent game-play.

4.2.2 Architecture

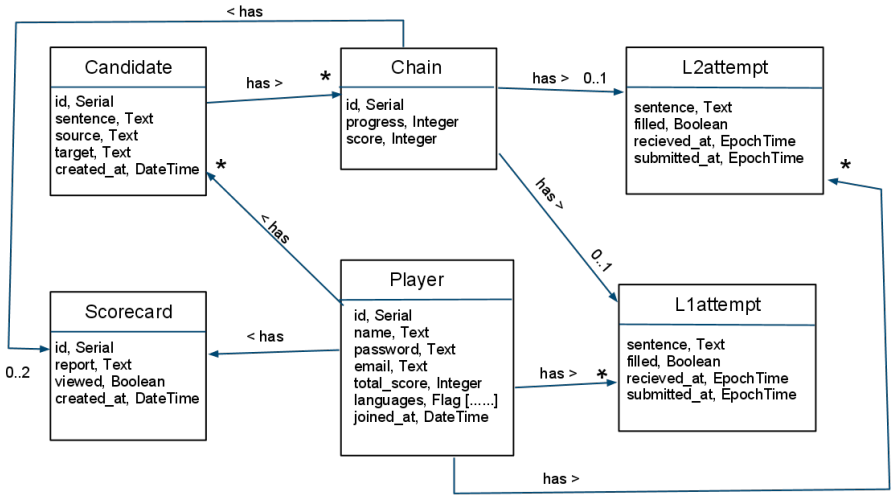

Fig. 6 Principle Classes

Figure 6 shows a diagram representing the principle classes in the Tower of Babel. Each of these includes the Datamapper::Resource as a mix in (a type of inheritance in ruby) and as such, their states can be easily saved to and loaded from the backend database, indeed the types of their data fields are Datamapper types as opposed to Ruby types. More interesting than the objects themselves are the relationships between them. These relationships reveal a lot about how the game has been implemented. We'll start with the player and trace the lifecycle of a translation chain from him or her. A player is created when a player registers to the system, minimal personal data is collected along with a Flag (enum from which more than one item may be selected, in fact in this case a minimum of two must be selected) of language codes. The time they joined is also recorded and their total score is set to zero. Players or guests to the site may create a candidate sentence, for which the text is recorded along with the source and target languages and the time of creation. Players may also initiate chains, by selecting *Start chain* in the game. Any single chain belongs to a candidate, and a candidate can have many chains initiated on it. The progress of a chain, i.e. how many stages have been completed can be tracked through the progress field, which is an integer. The score for a chain is filled when a chain has been completed. The L2attempt (L2 is a common term for second language in the areas of language translation and education) object belongs both to the player who initiated the chain and to the chain itself. This has a field to hold the text the player enters as their

translation, as well as Boolean field that is set to true when the text field has been filled, it also has two time fields, one recording the time the translation challenge was issued and the second recording the time it was submitted. An L1attempt (L1 standing for first language) is issued to a player when they ask to *Continue a chain*. This is much like the L2attempt object, except that the idea is that the previous player's sentence is translated back to the first or source language. Once both the L2 and L1 attempts have been issued and filled for a chain, it is considered to be completed and the score is calculated, with two scorecards being created, one for each player involved in the chain. The scorecard object holds a field containing its text and a Boolean value indicating whether or not it has been viewed by the player or not, this assists in alerting players when they have unviewed scorecards, a common occurrence for the player who completed the L2attempt part of the chain. Figure 7 provides a sequence diagram outlining how players interact with this implementation to produce translations.

Two important classes have been omitted from Figure 6. The first is the chain evaluator class, which has the dual responsibility of performing the scoring algorithm and creating scorecards. An object of this type is instantiated when a chain has been completed, a reference to the chain is passed into the object and the chain evaluator uses this reference to access all the data it needs to calculate the score and create the relevant scorecards. The second class omitted from the diagram is the language detector class, previously discussed in section 4.1.

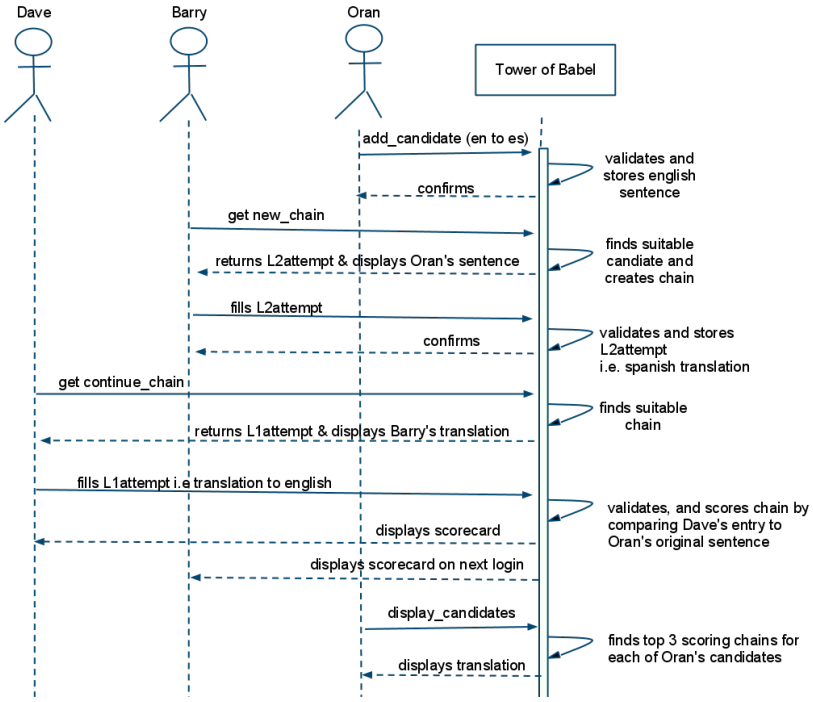

Fig. 7 Sequence diagram for Translation lifecycle

4.2.3 Longer Chains?

Isn't what makes Chinese whispers into a great game, the number of people involved in the chain? The more people there are in a chain, the greater propensity there is for errors to be introduced and the more challenging the task of maintaining the sentence in its original form becomes. It was, and still is, the intention of the author to see the Tower of Babel played in longer chains, for example with a sentence being translated from English to French, then to Irish, then to Spanish, then to German and finally back to English again, involving up to as many as ten players. It was decided however that for this implementation to work, a critical mass of players with broad range of different languages would be required before the game would start functioning properly. Thus the prototype has been implemented to work with chains of just three players. The design, however, is flexible enough for an easy introduction of this functionality in the future. In such a version, rather than a chain just holding a L2 and a L1 attempt, it would hold an array or hash of attempts, with the attempt class having an additional field holding information on its language. Each attempt would correspond to another link or player's input in the translation chain.

5 Evaluation and Future Work

5.1 Translations produced

With the game only in its infancy and despite the setback caused by significant downtime during the first week (see section 4.1), it is already producing translations of high quality. The following pairs of sentence were identified from within the system as being high scoring and either outperforming or equaling Google Translate's conversion of the same sentences.

1. English: The four guests held their glasses to the light and studied their blood-red contents.
   Spanish: *Los cuatro huéspedes pusieron sus vasos en la luz y examinaron el contenido sanguinolento.*

2. English: The primary colours are red, yellow and blue.
   Irish: *Is iad dearg, buí agus gorm na bun-dhathanna.*

3. English: My hovercraft is full of Eels
   French: *Mon aeroglisseur est plein d'anguilles.*

4. French: J'ai des autres chats a fouetter
   English: *I have other fish to fry.*

What is interesting to note in these four examples, is the ability of the human translator to deal with different contexts and translate them appropriately. The first

example appears to come from fiction, containing language that is both idiomatic and contextual. The second is a scientific statement, the third is just plain ridiculous and the fourth is an idiomatic phrase. In each case the human player has displayed an innate intelligence to deal with the translation in an appropriate manner and render something that is grammatically sound, while maintaining the original meaning of the text. The fact that another human was then able to translate these attempts back to the source language in such away that they were an exact or almost exact match of the original, serves to further verify the quality of the translations.

It should be noted however that these translations are the cream of what is being produced, and that for every quality translation produced there are several of lower quality. The scoring algorithm assists somewhat in identifying the better translations, thus helping separate the chaff from the wheat, but it is still far from perfect. Sometimes good translations end up scoring poorly and poor translations end up scoring highly. This is an issue that will have to be addressed when the prototype is overhauled. The scoring algorithm, which in its current state is somewhat rudimentary, could benefit from analysis of what is making it perform well, and could be tweaked to see what helps it improve. The inclusion of techniques from fields such as Natural Language Processing may also serve to enhance it.

Noted in the presentation on the Duolingo project (see section 2.3), was that in their application they intend to use aggregation of all translations collected for a sentence, along with peer review, to produce the best possible overall results. The Tower of Babel might well benefit from the inclusion of such features.

5.2 Reaching a Wider Audience

There are currently thirty-two players registered to the game. Many of these registered in the first few days after the launch, made a few translations and have not returned since. Thankfully, due to a strong personal network of linguists and language learners, those registering tended to have a background or interest in languages (the desired type of user). The game would definitely benefit by being delivered through a medium where it could engage a wider audience and hopefully keep them engaged. The obvious choice would be as a Facebook application, leveraging new users through social connections on a site that has such a huge volume of traffic does not need to be discussed here, but the possibilities are endless, of particular note are the large number of users affiliated to language groups on the site. A more novel, and perhaps more powerful approach might be to implement the game on Twitter. In such a scenario, candidates and attempts (limited to 140 characters) could be pushed between players using tweets who would then give their response in another tweet. Finally, placing a greater emphasis on the game as a fun language-learning tool might also help to attract more users, several language-teaching colleagues have expressed interest in using an adapted version of the game with their students (they would like to control the input of candidates, so that they can be pitched at the appropriate level).

6 Conclusions

> "The process of translating comprises in its essence the whole secret of human understanding of the world and of social communication."
>
> Hans Georg Gadamer [13]

Machine Translation has come along way since the early days of AI, but as this paper has demonstrated, it still has a long way to go before it could be called an exact science. The purpose of the Tower of Babel, using a method inspired by a collective intelligence approach, has been to find a way to harness fresh human understanding of language and communication and use that to perform translation in place of the machine. By identifying poorly performing machine translations and suggesting better translations for them, it is intended that the data captured in the game will serve a dual purpose of both providing translations and teaching machines to translate better. The translation data that has been captured thus far with the first prototype would suggest that there is value in this approach. With further research, looking at ways in which the scoring algorithm could be improved and mediums through which the game could be brought to a wider audience, the Tower of Babel could well contribute to pushing forward the AI problem of 'machine' translation.

References

1. Nagao, M.: A Framework of a Mechanical Translation between Japanese and English by Analogy Principle, in A. Elithorn and R. Banerji (eds), Artificial and Human Intelligence, Amsterdam: North-Holland, pp. 173–180. (1984)
2. Somers, H.: Review Article: Example-based Machine Translation. Machine Translation 14: 113–157. (1999)
3. Way, A.: Panning for EBMT gold, or "Remembering not to forget". Machine Translation 24:177–208. (2010)
4. Zhu, Q., Inkpen, D., Asudeh, A.: Automatic extraction of translations from web-based bilingual materials. Machine Translation 21:139–163. (2007)
5. Von Ahn, L.: Games with a Purpose. Computer 96 – 98. June 2006.
6. Speer, R., Havasi, C., Surana, H., Using Verbosity: Common Sense Data from Games with a Purpose. Conference paper at Twenty-Third International Florida Artificial Intelligence Research Society Conference. (2010)
7. Von Ahn, L.: Duolingo: The Next Chapter in Human Computation. Lecture delivered at TEDxCMU, April 2011. Accessed at http://www.youtube.com/watch?v=cQl6jUjFjp4 on 5/5/2011.
8. Edward, J.: 'Send Three- and Four-pence; We're Going to a Dance': Forward Generating Research. The Australian Educational Researcher 30, 17--32 (2003)
9. Carr, D., Buckingham, D., Burn, A., Schott, G.: Computer games: text, narrative and play. Polity Press. Cambridge, UK. (2006)
10. Von Ahn, L., Liu, R., Blum, M.: Peekaboom: A Game for Locating Objects in Images. CHI 2006, April 22 – 28, 2006, Montreal, Quebec, Canada. (2006)

11. Von Ahn, L.: Human Computation. Thesis, at School of Computer Science, Carnegie Mellon University (2005)
12. Sinatra The Book, http://sinatra-book.gittr.com/
13. Biguenet, J., Schulte, R.: The Craft of Translation. Chicago and London: U Chicago P, (1989)