Towards a Gamified System to Improve Translation for Online Meetings

Laura Guillot[1,2], Quentin Bragard[1], Ross Smith[3], Dan Bean[3] and Anthony Ventresque[1]
[1]Lero, School of Computer Science, University College Dublin, Ireland
[2]École Centrale de Nantes, France
[3]Microsoft Corporation, Skype Division, Seattle, USA
laura.guillot@eleves.ec-nantes.fr, quentin.bragard@ucdconnect.ie, rosss@microsoft.com, danbean@microsoft.com, anthony.ventresque@ucd.ie

Abstract

Translation of online meetings (e.g., Skype conversations) is a useful feature that can help users to understand each other. However translations can sometimes be inaccurate or they can miss the context of the discussion. This is for instance the case in corporate environments where some words are used with special meanings that can be obscure to other people. This paper presents the prototype of a gamified application that aims at improving translations of and for online meetings. In our system, users play to earn points and rewards – and they try to propose and vote for the most accurate translations *in context*. Our system uses various techniques to split conversations in various semantically coherent segments and label them with relevant keyphrases. This is how we extract a description of the context of a sentence and we use this context to: (i) weight users' expertise and their translation (e.g., an AI specialist is more likely than a lay person to give a correct translation for a sentence about deep learning) (ii) map the various translations of words and phrases and their context, so that we can use them during online meetings.

In: F. Hopfgartner, G. Kazai, U. Kruschwitz, and M. Meder (eds.): Proceedings of the GamifIR 2016 Workshop, Pisa, Italy, 21-July-2016, published at http://ceur-ws.org

1 Introduction

Machine Translation has been a rich field of research for decades and many directions and models have been proposed [Koe09]. Commercial systems, such as, Microsoft Translator[1], Google Translate[2] or Systran[3], are commonplace nowadays and have proven to be useful to users on the Internet and in real life. However, despite their obvious successes, academia and industry are still facing hard challenges and many translations proposed 'in the wild' lack quality. By 'in the wild' we mean translation of short and, at times, low linguistic quality texts, as we see them for instance on online social network applications. This is especially a challenge for Machine Translation as speech (e.g., during online meetings) and short written posts and comments (e.g., on online social applications) are usual ways of communicating nowadays. These ways of communicating, not only they are incomplete and noisy, but they are also contextual in nature: they are often linked to a particular community (e.g., teenagers, employees of an Enterprise, members of a professional field) and evolve quickly. For instance some phrases are subversive and become quickly popular among a group of peers while the rest of the population does not know their meaning or how to use them. The expressions "deadly" (as in, "it was deadly", which means "it was great") that you will hear in Dublin or "c'est solide" ("it's fair") used by French teenagers are hardly found on online resources and Machine Translation systems are unlikely to handle them correctly.

In this paper we propose a gamified application that aims at *collecting translations from and for online meetings*. First, our system encourages users of online

[1] https://www.microsoft.com/en-us/translator

[2] https://translate.google.com/

[3] http://www.systransoft.com/

meeting systems (our system is based on Skype) to *submit elements of their discussions and the translations that go with them.* These elements are then *segmented into contextually homogeneous partitions* and we apply a *topic labelling* mechanism to detect the relevant keyphrases to describe them. Users can then play with our system and try to find the *best* translations given the *context* of the sentences. In the back-end, our system selects the best translations depending on both the crowds' preference and the expertise of the players (using a mapping between context and expertise). Our system can then be used during online meetings, where the context is monitored to find the most accurate translation.

We perform evaluations for the topic detection and usability (i.e., how easy and satisfying is the interface) elements of our system. We show that our system finds the right description of the context 65.8% of the time – and that our users find the application simple and pleasing (usability score of 82%).

Using the crowd (and discriminating workers using their competence) to obtain quality translations is not a novel idea as such (e.g., see the work done by Chris Callison-Burch and his team [ZCB11]). However, using a game and the notion of context (to qualify sentences and players' expertise) to increase the quality of the translations is new as far as we know.

The rest of this paper is organised as follows: Section 2 describes the first, important, element of our system: the segmentation into semantically homogeneous contexts and their descriptions using topics and keywords; Section 3 describes our prototype: the architecture of the game, the game design and the motivations to play the game; finally Section 4 concludes our paper and discusses some of the future directions we plan to follow.

2 Contextual Translation

One of the main ideas behind our work in this paper is that translation of online meeting, i.e., speech and potentially short sentences, needs to be correlated to the context of the discussion. This context, in the noisy, limited and community-oriented environment we described in the previous section, is what allows to get more accurate translations. Especially as our gamified system records the context associated to a sentence and uses it to: (i) help the translators/players to find the best translation by giving them the context of the discussions; and (ii) eventually translate more accurately online meetings.

2.1 Topic Detection

Topic detection consists in discovering the important keywords in a document or a part of a document. In this paper, we combine different techniques to generate a cloud of topic labels. First, we apply a text segmentation [SCS04] method on the meeting transcript to split it into one-subject sections. Then, we retrieve a distribution of keywords for each section using a Latent Dirichlet Allocation [MBYNIJ03] (LDA). Eventually, we put a label [Sch09] on the list of keywords using the Wikipedia category network, finding the category the describes the best the keyphrases contained in the topic.

2.1.1 Text Tiling

For the first part of our topic detection algorithm, we use a Text Tiling [BIR] method which, given a document, returns a list of segments where each segment corresponds to a single subtopic. This method is based on the computation of lexical coherence between two adjacent blocks of text to determine where there is a topic change.

We start by pre-processing the document using stop-word removal and tokenisation. This leaves us with a list of tokens (t_i) for the document d: d = {t_1, t_2, ..., t_n}. Then, we define a sequence of blocks of tokens, each of the same size (K):

$$b_i = \{t_j, j \in [i, i+K]\} \tag{1}$$

The block $\mathrm{b_i}$ is the one which begins with the token $\mathrm{t_i}$. For empirical reasons, we have chosen K=20.

For each pair of successive blocks in the document, $\mathrm{b_i}$ and $\mathrm{b_{i+K+1}}$, our method computes the cohesion score of the associated gap $\mathrm{g_i}$ between the two blocks using the vocabulary introduction metric:

$$Score(g_i) = \frac{New(i) + New(i+K+1)}{2K} \tag{2}$$

where New(i) is the number of new terms introduced in the block $\mathrm{b_i}$ that were not in the document before $\mathrm{b_i}$.

Our solution uses these scores to detect where the similarity between two blocks is minimum, using the following depth score metric:

$$DepthScore(g_i) = Score(g_l) + Score(g_r) - 2.Score(g_i) \tag{3}$$

This metric compares the novelty (in term of common tokens) between $\mathrm{b_i}$ and two others blocks $\mathrm{b_l}$ and $\mathrm{b_r}$ which are the two blocks with a smaller score than $\mathrm{b_i}$ on the left and on the right of $\mathrm{b_i}$. This gives an indication of the depth of the gap $\mathrm{g_i}$: the higher the score the more dissimilar are the two blocks before and after the gap.

The issue now is that our metric gives us a large number of local maxima. Thus, we use a smoothing

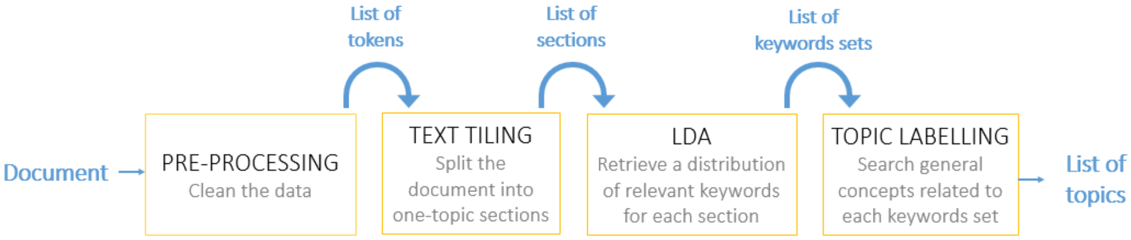

List of tokens
List of sections
List of keywords sets
Document
PRE-PROCESSING
Clean the data
TEXT TILING
Split the document into one-topic sections
LDA
Retrieve a distribution of relevant keywords for each section
TOPIC LABELLING
Search general concepts related to each keywords set
List of topics

Figure 1: Workflow of our topic detection algorithm

process here to highlight the relevant maxima by averaging them with a fixed window:

$$DepthScore(g_i) = \frac{1}{2a} \sum_{j=i-a}^{i+a} DepthScore(g_j) \qquad (4)$$

a is a parameter that we set at 5 after an empirical evaluation and given the size of our documents.

every time we find a depth score higher than:

$$\bar{s} + \frac{\bar{s} - \max_i(DepthScore(g_i))}{2} \qquad (5)$$

where $\bar{s}$ is the average of the scores, we split the document after the block and expect the section before to be about a different topic than the section after. Figure 2 gives a visual representation of our Text Tiling algorithm applied to a concatenation of 3 Wikipedia articles. This validation gives us an interesting output as our algorithm splits the text into 4 segments (see Figure 2) – while the text is the concatenation of 3 documents. However, we noticed that one of the document is actually quite heterogeneous semantically and it seems to have 2 different topics.

2.1.2 Latent Dirichlet Allocation (LDA)

Once the different contextual sections are identified (we assume that they have only one topic), we apply a Latent Dirichlet Allocation [MBYNIJ03] (LDA) to each of them. This algorithm gives a discrete distribution of words with their probabilities for each topic. The difference between our scenario and the standard use of LDA is that we potentially apply it to short segments (not a lengthy corpus) and we configure it to retrieve only one topic for each contextual section.

Furthermore, we have chosen to set at 5 the number of words per topic given by the algorithm. This number is enough to enable a correct topic labelling during the next step.

We have used a Java package for the LDA and DMM topic models called jLDADMM[4] because it provides alternatives for topic modelling on short documents such as tweets or in our case segments of a conversation.

2.1.3 Topic Labelling

At the end of the LDA step, we end up with a list of keyword sets (one set per contextual section). The objective is now to obtain one or two labels for each topic. For example, for the following distribution of words: *java - classes - objects - algorithm - instantiate*, we would like to have something like: *Object programming - Computer science.*

The main idea of our labelling approach is to use the Wikipedia categories as labels. We believe it is relevant technique for topic labelling as Wikipedia categories carry a good semantic content [Sch09, NS08]. To map words and categories, we have used the articles of Wikipedia and more specifically their titles. For each article, we have taken the title, removed the stop words and returned a list of relevant words. Then, we have mapped each of these words to the categories present in the article. At the end, we have obtained a matrix which gives the probability of a word to be related to a category. To process this matrix, we have parsed approximately 15% of the whole Wikipedia XML dump[5], removing the disambiguation pages, articles describing categories and redirection pages. We have also deleted categories which covered too many semantically unrelated articles like *American films* or *1900 deaths* as well as unrepresentative words, i.e., words related to too many categories. At the end, we have indexed around 42,000 categories and 64,000 words. Given our matrix, we can retrieve the categories corresponding to each topic ranked by their combined score:

$$Score(C) = \sum_{w \in T} W(w \in T).P(w \in C) \qquad (6)$$

[4] http://jldadmm.sourceforge.net/
[5] https://dumps.wikimedia.org/

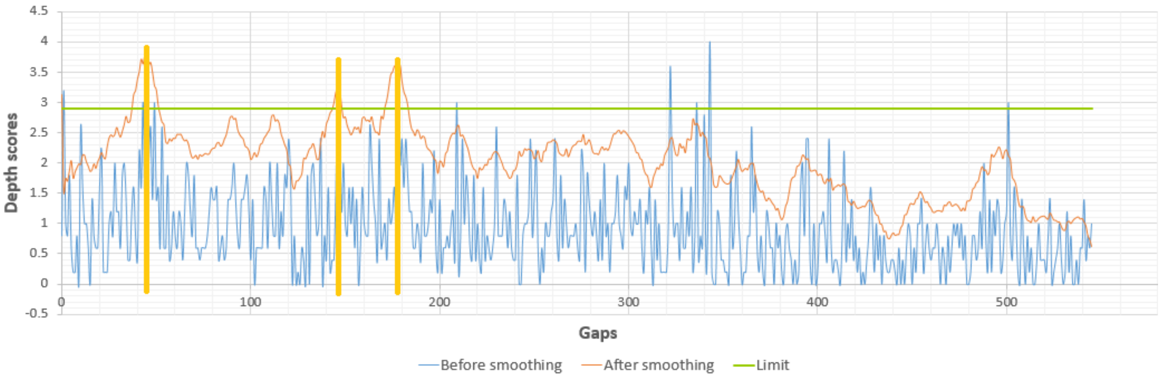


Figure 2: Gap score results of the analysis of the concatenation of 3 articles from Wikipedia (*Signal processing*, *Dublin* and *Anarchism*). The x-axis gives the gap index and the y-axis gives the depth score of each gap. Yellow straight lines show the final segmentation. Notice that while we had 3 articles from Wikipedia, the system split them in 4 segments. It actually makes sense as one of the articles had clearly two separate contents (article *Anarchism* in Wikipedia)

.

Where:

- C is a category
- T is a topic given by LDA (i.e., a list of words)
- w is a word of T
- W(w ∈ T) is the weight of the word w in the topic T
- P(w ∈ C) is the probability for w to be related to C, found in the Wikipedia matrix

2.2 Evaluation

To validate our approach, we have used a benchmark (Wiki20[6] [Med09]) which consists of 20 computer science articles annotated by 15 teams of graduate students. The objective is to measure the similarity between the keyphrases assigned by the humans and our system.

To compute similarity between two topic labels, we have used the DISCO API which is based on a precomputed database of word similarities, called a word space. In this space, each word is associated to a distribution of semantically similar words. Thus, to compare two words, we finally compare their associated words vector by using a statistical analysis done on very large text collections. With this tool, for each label assigned by humans, we search the most similar category retrieved by the algorithm and we compute the average of the similarity scores of these selected categories. That gives us a similarity value for each document of the dataset. Figure 3 shows how close our own labels are from the ones picked by the human assessors, for group of teams of assessors and each document of the benchmark. The global average over all the documents is 65,8%. As a quick discussion of these results, we should say that this evaluation is not perfect as: (i) it is performed on a homogeneous corpus (all documents are from the field of Computer Science); (ii) the DISCO similarity API is not comprehensive and lacks of technical vocabulary. This study probably needs to be replicated and extended to make sure our labelling system is accurate and relevant.

3 Description of our Prototype

Our system is based on a gamified collection of user feedback: users submit sentences and translations, and they vote for the most accurate ones. In general gamification has proven to be effective at collecting users feedback [HKS14, Smi12] – but games need to be well designed to give users incentives to participate [RD00]. Our game is composed of three micro-tasks and provides game mechanisms that we believe would make the users keen to participate. More specifically, our use case is the following:

- Users of Machine Translation systems for online meetings submit some of the translations they are offered (they earn points for doing that)
- Players improve the translations and vote or those they consider the best, and earn points

Our prototype is a web application that uses Microsoft Skype API[7] to interact with the online meeting application. At the moment our prototype ingests all the data collected in a database that stores user profiles

[6] https://github.com/zelandiya/keyword-extraction-datasets

[7] https://www.skype.com/en/developer/

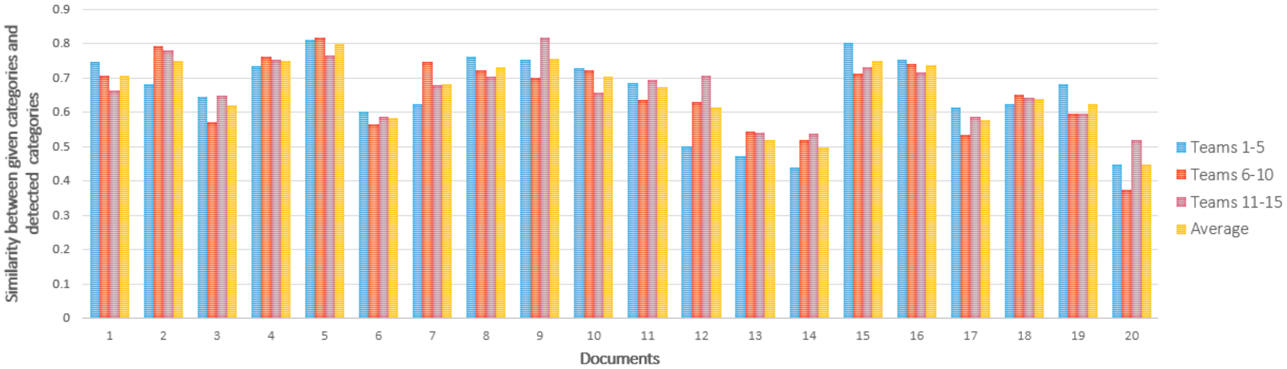


Figure 3: Similarity between the labels selected by our system and the tags chosen by human assessors (grouped in 15 teams). Notice that on average we have 65% of agreement with the assessors (not seen on the figure).

(points, comments, votes), sentences, translations and contexts.

The current section goes through the different elements of our prototype: the system architecture, the game design, the gameplay and the motivation.

3.1 Architecture of the Game

Our game is composed of two parts. The first part is the submission of translations during or just after online meetings and the second part is the evaluation of the submissions.

Figure 4 shows the interface of the submission part of our prototype: some people are having a discussion on an online meeting application (Skype in our prototype, seen on the right of Figure 4) and their speech is being recorded (using some speech-to-text component – see on the left of Figure 4).

Every sentence is translated and the users can submit the sentences and/or correct them. The context of the discussion is monitored and segmented if required (see Section 2) The context is eventually used to label the sentences - and keep track of their context.

After being uploaded, the sentences are submitted to the vote of the community. The initial sentence, the associated translations and a cloud of keywords which describes the context are given with every sentence. Users have to choose one of the proposed translations or if none of these translations suit them, they can add their own. An example is given on the Figure 5. This consensus decision making leaves the system with all the options (translations) which can be handy in case of evolution or if a statistical model is applied.

Our game has two modes: a classical mode and a challenge mode. During the classical mode players evaluate 10 translations at a time whereas in the challenge mode they evaluate translations for as long as they find the 'correct' answer (i.e., in agreement with what the community thinks) – the objective being to score the highest mark.

3.2 Game Design

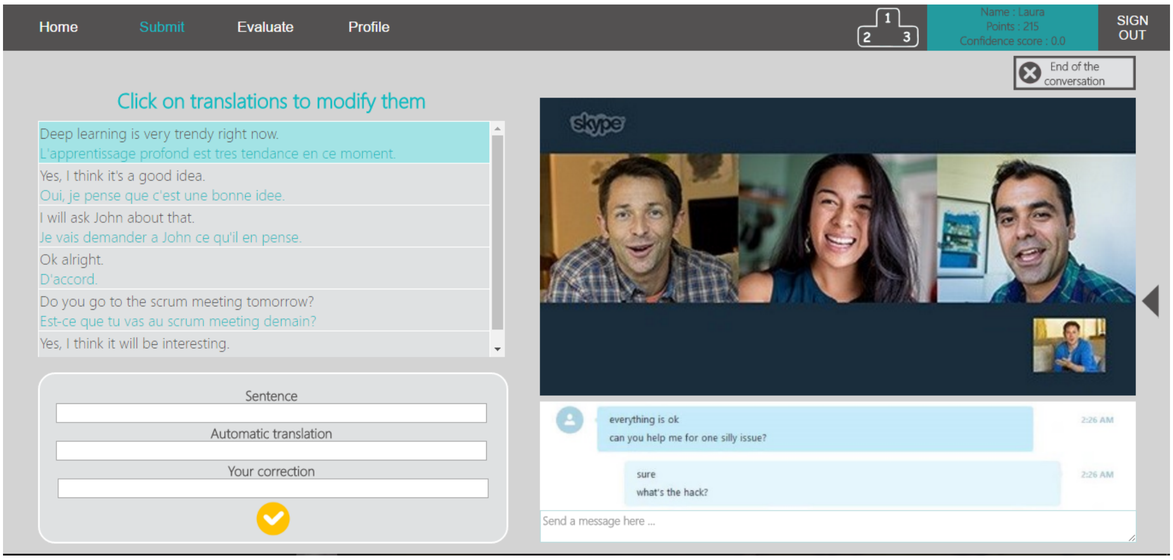
The game is designed to follow the look and feel of the Skype online meeting application. The objective is to have a simple and attractive interface motivating people to participate.

To test the usability of our system, i.e., its effectiveness, efficiency and satisfaction, we have used the System Usability Scale [B$^+$96] (SUS). This method gives a score between 0 and 100 which indicates how much an application is pleasant to use. The SUS framework has set the average at 68 according to a research[8]. It is based on a questionnaire consisting of the 10 following questions:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Users give a score from 1 (strongly disagree) to 5 (strongly agree) to every question. We have conducted the survey on 7 bilingual students on the evaluation part of the game. We have obtained a usability of

[8] http://satoriinteractive.com/system-usability-scale-a-quick-usability-scoring-solution



Name : Laura
Points : 215
Confidence score : 0.0

Click on translations to modify them

Deep learning is very trendy right now.
L'apprentissage profond est tres tendance en ce moment.

Yes, I think it's a good idea.
Oui, je pense que c'est une bonne idee.

I will ask John about that.
Je vais demander a John ce qu'il en pense.

Ok alright.
D'accord.

Do you go to the scrum meeting tomorrow?
Est-ce que tu vas au scrum meeting demain?

Yes, I think it will be interesting.

everything is ok
can you help me for one silly issue? 2:26 AM

sure
what's the hack? 2:26 AM

Figure 4: Interface for submitting and updating translations

82%, which is a very good score. The worst score was given to the first question. We assume that the reason for this is that some users did not feel immersed in the gamified universe as they were just asked to try once the game with a fictive account.

3.3 Game Mechanisms

To motivate the users, all the tasks in our prototype are linked to a *Points-Badges-Leader board* system. Players earn points for each submission or evaluation, and these points enable the players to go through different progress status (e.g., beginner, expert). Our prototype is also composed of a leader board that summarises users' points and status. Other elements of importance in gamified applications that we have implemented are missions (e.g., "evaluate 30 translations today") and trophies that players win as they progress in the leader board and by leading it.

The points, trophies and status are displayed on the user profile so that users can see their progress in the game. You can see an example of a profile on Figure 6. All these extrinsic motivators are used to increase the participation and give the user a visual representation of an accomplished task.

In addition to the points, each player has a confidence score which tells how relevant their participation is. It is computed as follows:

$$CS(p) = \left(1 + \frac{N_{v,a}(p)}{N_v(p)} + \frac{N_{s,a}(p)}{N_s(p)}\right) * level(p) \qquad (7)$$

where:

- p: player
- level(p): player p's level (given by the points p earned)
- $\mathrm{N_v}(p)$: number of votes made by the player p
- $\mathrm{N_{v,a}}(p)$: number of votes made by p that are approved (see below)
- $\mathrm{N_s}(p)$: number of submissions made by the player p
- $\mathrm{N_{s,a}}(p)$: number of submissions made by p that are approved

The confidence score is used to weight the votes of each player p by their declared expertise and perceived expertise (from the crowd) through how many of their votes were 'correct'.

The relevance of a translation is computed by:

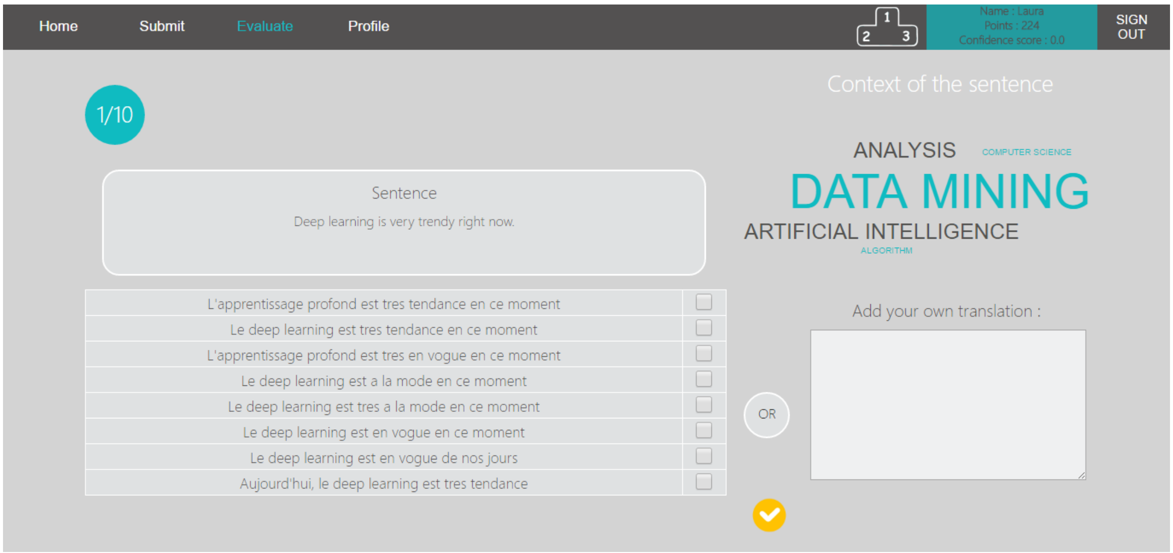
$$Relevance(t_i^s) = \frac{\sum_{p \in V_{for}(t_i^s)} v(p, t_i^s) - \sum_{p \in V_{against}(t_i^s)} v(p, t_i^s)}{\sum_{p \in V(t_i^s)} v(p, t_i^s)} \qquad (8)$$

With

$$v(p, t_i^s) = CS(p).k(p, s) \qquad (9)$$

Where:

- t_i^s : translation i of the sentence s
- $\mathrm{V_{for}}(t_i^s)$:set of the players who voted for t_i^s
- $\mathrm{V_{against}}(t_i^s)$: set of the players who voted for the other translations
- V(s) : set of the player who have evaluated s
- k(p,s) : is a factor equal to 1 if the player is familiar with the context of the sentence and 0.5 otherwise – players have the possibility to add topics in which they have some knowledge to their profile.



Name : Laura
Points : 224
Confidence score : 0.0

Context of the sentence

ANALYSIS COMPUTER SCIENCE DATA MINING ARTIFICIAL INTELLIGENCE ALGORITHM

Sentence

Deep learning is very trendy right now.

- L'apprentissage profond est tres tendance en ce moment □
- Le deep learning est tres tendance en ce moment □
- L'apprentissage profond est tres en vogue en ce moment □
- Le deep learning est a la mode en ce moment □
- Le deep learning est tres a la mode en ce moment □
- Le deep learning est en vogue en ce moment □
- Le deep learning est en vogue de nos jours □
- Aujourd'hui, le deep learning est tres tendance □

Add your own translation :

Figure 5: Interface for evaluating translations

> These topics are used to determine if the player is familiar with the context of a sentence or not. This factor enables to weight the votes with the user's expertise.

The confidence score is updated retroactively each time a sentence is approved. This system of confidence score is useful to tell the difference between relevant participation and participation only motivated by earning points. In order to motivate people to increase their confidence scores, we have also added a leader board based on them.

3.4 Intrinsic Motivation

In addition to these rewarding systems (a.k.a., extrinsic motivation systems), it is important for us to entice people to play by intrinsic motivation [RD00]. Intrinsic motivation in gamification optimises human motivation to use the system and consequently brings better data quality. The objective is that players play for inherent satisfaction and not simply for additional rewards. In our case, we have chosen to focus on the feedback given to the player. For instance, we enable players to see the votes in favour of their own submissions. Thus, they have a feedback on their participation and can improve their skills.

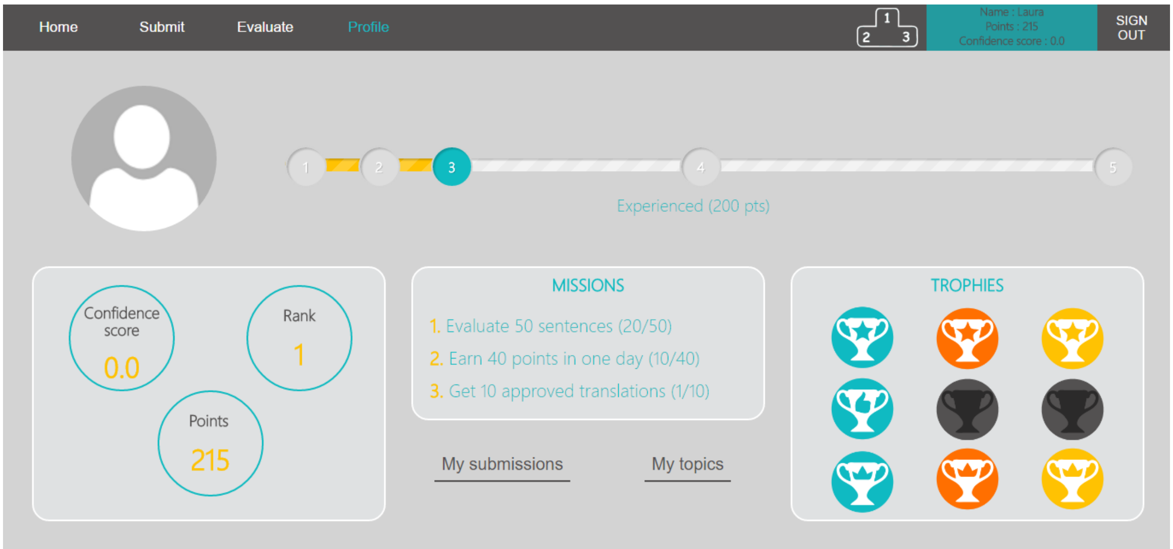
Moreover, we also display some statistics about the game, such as, the number of validated sentences or the number of submitted sentences, and we provide a visualisation of the approved corrections per language. These pieces of information show the players that they participate for a real purpose and they are members of a real community. This may particularly be important in a corporate context: staff members of a company could work together to improve the translations within their company – it is a possible extension of our system, which could be deployed/used internally in an Enterprise or group. In that particular case, we can probably also count organizational citizenship behavior (i.e., the willingness to perform tasks that help the organisation without explicit rewards) [SON83] to increase intrinsic motivation.

4 Conclusion and Future Work

In this paper we have presented our prototype of a gamified translation improvement system for online meetings. Our system collects translations during online meetings and asks the crowd to improve them in context. Players earn points when they submit the translations and when they vote for them. The vote of the players, weighted by their expertise in specific contexts, helps our online meeting translation system to be more accurate – again, in context.

Using a known topic labelling benchmark, we have validated that our topic detection and labelling component works well - we got 65% agreement with human assessors. We have also conducted a system usability scale survey and the respondents acknowledged that our system is easy to use and brings satisfaction to players - score of 82%.

As future work, we would like to (i) test our topic detection and topic labelling algorithms on more exhaustive benchmarks; (ii) improve the topic labelling algorithms using more structural and semantic information (links between categories, text of the articles, hierarchy of categories); (iii) use our system to compare different machine translation systems or different



Name : Laura
Points : 215
Confidence score : 0.0

Experienced (200 pts)

Confidence score 0.0

Rank 1

Points 215

MISSIONS

1. Evaluate 50 sentences (20/50)
2. Earn 40 points in one day (10/40)
3. Get 10 approved translations (1/10)

TROPHIES

Figure 6: User profile: note the various elements: score, progress made, missions and trophies.

parameters/versions of machine translation systems – this would particularly be interesting for machine learning-based systems; (iv) evaluate our own system borrowing the ideas we can find in the crowdsourcing domain (for instance what Chris Callison-Burch and his team do in [ZCB11]).

Acknowledgement

This work was supported by Science Foundation Ireland grant 13/RC/2094 to Lero - the Irish Software Research Centre.

References

[B^{+}96] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 1996.

[BIR] Satanjeev Banerjee and Alexander I. Rudnicky. A text tiling based approach to topic boundary detection in meeting.

[HKS14] Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work?–a literature review of empirical studies on gamification. In *HICSS*, 2014.

[Koe09] Philipp Koehn. *Statistical machine translation.* Cambridge University Press, 2009.

[MBYNIJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of machine learning research*, 2003.

[Med09] Olena Medelyan. *Human-competitive automatic topic indexing.* PhD thesis, The University of Waikato, 2009.

[NS08] Vivi Nastase and Michael Strube. Decoding wikipedia categories for knowledge acquisition. In *AAAI*, 2008.

[RD00] Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1), 2000.

[Sch09] Peter Schönhofen. Identifying document topics using the wikipedia category network. *WI/AS*, 2009.

[SCS04] Nicola Stokes, Joe Carthy, and Alan F Smeaton. Select: a lexical cohesion based news story segmentation system. *AI Communications*, 17(1), 2004.

[Smi12] Ross Smith. The future of work is play. In *International Games Innovation Conference*, 2012.

[SON83] CA Smith, Dennis W Organ, and Janet P Near. Organizational citizenship behavior: Its nature and antecedents. *Journal of Applied Psychology*, 68(4), 1983.

[ZCB11] Omar F. Zaidan and Chris Callison-Burch. Crowdsourcing translation: Professional quality from non-professionals. In *ACL*, pages 1220–1229, 2011.