Rosetta Code: Improv in Any Language

Piotr Mirowski[1], Kory Mathewson[1], Boyd Branch[1,2]
Thomas Winters[1,3], Ben Verhoeven[1,4]
[1]Improbotics (https://improbotics.org)
[2]University of Kent, United Kingdom
[3]KU Leuven, Dept. of Computer Science; Leuven.AI, Belgium
[4]ERLNMYR, Belgium

Abstract

Rosetta Code provides improv theatre performers with artificial intelligence (AI)-based technology to perform shows understandable across many different languages. We combine speech recognition, improv chatbots and language translation tools to enable improvisers to communicate with each other while being understood—or comically misunderstood—by multilingual audiences. We describe the technology underlying Rosetta Code, detailing the speech recognition, machine translation, text generation and text-to-speech subsystems. We then describe scene structures that feature the system in performances in multilingual shows (9 languages). We provide evaluative feedback from performers, audiences, and critics. From this feedback, we draw analogies between surrealism, absurdism, and multilingual AI improv. Rosetta Code creates a new form of language-based absurdist improv. The performance remains ephemeral and performers of different languages can express themselves and their culture while accommodating the linguistic diversity of audiences.

Figure 1: Example of a performed *Telephone Game*. Performers are aligned and one whispers to their partner on the right a phrase in a foreign language (here, in Swedish), which is then repeated to the following performer, until the last utterance is voiced into automated speech recognition and translation to show how information is lost.

Introduction

Theatre is one of the most important tools we have for sharing experiences and building cross-cultural understanding. Moreover, theatre performers and audiences who speak different languages are more connected than ever, thanks to increasing ease of communication, dissemination of culture, translation, travel, and improvements in remote performance capabilities. In particular, improvised theatre (improv) is well positioned to connect culture given its universality, accessibility, and low barriers to entry: improvisation techniques can be readily understood and internalized, and in a short manner of time, individuals from diverse cultures empathize with each other while performing scenes together, with deep characters, relationships, settings, motivations, and subtext. Improv serves as a microcosm of cultural communication; it is "the theatre of the people" in moment (Boal 2006). Improv is therefore an ideal test-bed to explore broad cultural and communication questions (Mathewson 2019).

Improv is also a paradoxical cultural artifact. On one hand, improv is ubiquitous and conveys universal messages about the human condition and the vagaries of life. On the other hand, as a highly linguistic art form, improv is nearly impossible to understand if you do not know the language in which it is performed. Given that improv is based on the connection between the audience and the performers, watching improv in a foreign language severely limits this link. This contrasts with scripted theatre, which has been salvaged from monolingual oblivion: Sophocles, Shakespeare, and Sartre continue to be translated into many different languages, reinterpreted, and enjoyed by audiences around the world. Improv has not had such an opportunity, and performance groups are bound to remain local or switch to English as a *lingua franca* when performing internationally.

The art of improvisation is derived from the connections between performative layers, both between the performers, and between the performers and the audience. Improv embraces the audience to create collaboratively together. In this way improvisation is a democratic narrative, and the potential impacts of improvised theatre between performers and audiences of different cultures and languages are significant. Most international improvisational collaboration is English based, but many regional festivals take place in the languages of the host region. These performances exclude audiences without knowledge of the performance language, and limits the contributions of improvisors who do not speak the language. Without translation, improvisation misses out on important voices due to language limitations.

How can we create conditions so that improvisors from different cultures can improvise together in their own language? How can audiences understand performers using different languages? How might we grow our cultural communication and empathy while only being able to speak one language? **Rosetta Code** answers these questions, and gives theatrical improv a suite of software, scenes, and show structures from which to advance and expand.

The **Methods** section describes the technical details of the system, the challenges associated with improv in any language, and how we used our system in the context of theatrical improvisation. In section **Rosetta Code on Stage** we provide results of using the system in three shows using nine languages. We also present evaluative feedback from performers, audiences, and critics. In section **Related Work, Historical Context and Discussion** we situate Rosetta Code at the intersection of improvisational theatre and language, present an exploration of the cultural importance of multi-lingual artistic performance, and provide several directions for future work.

Methods

Artificial intelligence-based improvisation is an art form, where a robot and/or AI is used on stage as an improv stage partner (Bruce et al. 2000; Mathewson and Mirowski 2017a; 2017b; Mirowski and Mathewson 2019; Jacob et al. 2019; Winters and Mathewson 2019; Liu et al. 2019; Mathewson 2019). That robot relies on a generative language model to produce lines or actions in response to context, and can in itself be seen as a computationally creative system. A variation of that format, *Improbotics*, designed in 2016, consists in letting human actors enunciate the lines: the chatbot effectively whispers lines into the ears of human improvisers, who are only allowed to repeat exactly those lines, but are otherwise free to express themselves with a full vocabulary of physicality and emotions (Mathewson and Mirowski 2018). We have adopted this configuration for the *Rosetta Code* show.

The core idea of **Rosetta Code** is to build on existing, state-of-the-art language technology (Section: *Technology Overview*) to enable a palette of improvisational games (Section: *Improv Games*). Rosetta Code thereby allows improvisors speaking different languages to perform multilingual improv theatre together on stage.

Technology Overview

The technical setup used in this project consists of several elements that can be seen as independent building blocks, each corresponding to a piece of equipment or to an Application Programming Interface (API).

- Speech recognition (e.g. Google Speech-to-text API[1] or Web Speech API[2]), which works in multiple languages[3], running in a browser application. In order to successfully capture the improviser's voice while occluding ambient noise and other performers' voices, we rely on handheld dynamic vocal microphones (with an on-off button that can be triggered by the user), connected to the computer via an analog-to-digital audio interface.
- An instantaneous translation system, e.g. Google Translate API[4], is used as communication channel to convert recognised speech from one language into another.
- A surtitle visualization interface (Figure 2) that enables the audience to follow the conversation using instantaneous translation, and allows improvisors to modify translation language settings.
- Text-to-speech synthesis API to automatically voice translations.
- In-ear headphone interfaces (or *earpieces*) enable individual performers to listen to audio translation while still being able to follow other conversations. Our setup to transmit sound from the computer to the improviser relies on FM radio transmitters that can multicast information to multiple FM radio receivers worn by several improvisers.

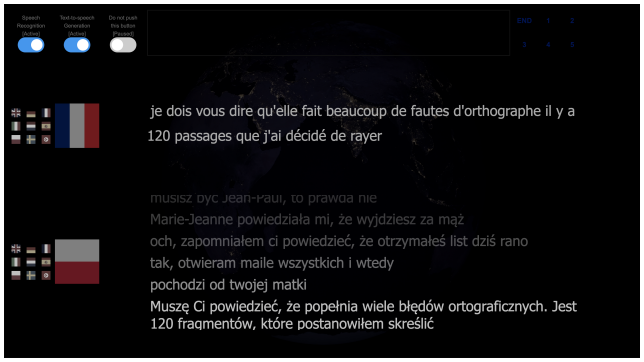

Figure 2: Illustration of the visual interface used in *Rosetta Code* as seen by the audience. The top part displays the choice of the language used for speech recognition and the latest recognised sentence (here French). The bottom part shows the target language for machine translation (here Polish), as well as the last few translation results. The buttons and input box (top) enable overriding speech recognition and activating / deactivating text-to-speech.

Improvisational Chatbot System To respond meaningfully to human improvisor input utterances, the AI improv system works by using a statistical language model to generate sentences in continuation of some context presented as text. Previous versions of AI improvisation were built upon the neural network sequence-to-sequence architecture (Sutskever, Vinyals, and Le 2014) trained on a *pseudo-translation task* from the context into the generated output (Vinyals and Le 2015). For Rosetta Code, we rely on the GPT-2 neural network transformer architecture (Radford et al. 2019), trained on a large corpus of web pages, which we fine-tuned on the OpenSubtitles corpus[5] of film subtitles (Tiedemann 2009).

[1] https://cloud.google.com/speech-to-text
[2] https://google.com/intl/en/chrome/demos/speech.html
[3] https://cloud.google.com/speech-to-text/docs/languages
[4] https://cloud.google.com/translate/docs
[5] https://opensubtitles.org/

Figure 3: Example of performed translation scene. The performer downstage speaks into the microphone in Dutch. Translation into English appears on screen and is fed via earpiece into the ear of the performer upstage.

It is straightforward to integrate any existing chatbot into the *Rosetta Code* system, by replacing the machine translation component by that chatbot component. The chatbot acts like a sort of translation from one language into that same language, one sentence later. The virtual AI improv chatbot controlling a human performer can be seen as a difficult stage partner whose language often veers on the absurd and forces improvisers to resort to nonverbal communication (Mathewson and Mirowski 2018; Mathewson 2019).

Improv Games

There are several improv games that can be played by multilingual improvisers using the aforementioned technology. These games can be subdivided into *translation-free games* (that do not require the translation service), *translation-based games*, and *vocalisation-based games*.

In the following descriptions, we refer to the primary language spoken by the audience as the *majority* language and to the other, "foreign", languages spoken by the improvisers as *minority* languages. We also make several assumptions about the improvisers, the audience, and the languages they speak. First, we assume that some improvisers speak only one (majority) language, while others master or have working knowledge of multiple (majority and minority) languages. Second, we also account for some members of the audience being fluent in multiple spoken languages. These configurations enable different combinations of information asymmetry during the show.

Translation-free Games There are several different improv games we propose for playing improv with improvisers speaking different languages. For this, we build upon *gibberish* improv games, in which one or more of the performers speak in a non-existing language (Johnstone 1979). However, in our games, each performer is allowed to express themselves by formulating real language. This setup is not unlike the vision of absurdist playwrights such as Samuel Beckett or Eugene Ionesco. They see language as being purely aesthetic (in our case, each utterance is fully formulated) and devoid of semantic significance (in our case, most performers and audience cannot understand the minority languages): *"the Theatre of the Absurd shows the world as an incomprehensible place"* (Esslin 1960).

Specifically, we propose to adapt existing improv exercises to the following games:

- **Stranger in a Strange Land**: a stranger does not understand the language of the others, and the others do not understand the language of the stranger either. This game can be played when several minority language performers (e.g., Chinese speakers in an European country) improvise with a majority language performer and for a majority language audience (e.g., English), thereby reversing the usual majority-minority status relationships faced by minority language speakers.
- **Languages of Love**: two performers are on a blind date or in a long term relationship, but clearly speak a different language. This is a setup that invites the performers to seek equal status.
- **Tower of Babel**: Every performer speaks a different, unique, minority language, and does not understand the language of the others.

These games were designed to investigate how meaning and understanding can emerge without words, and force the performers to explore alternative means of communication, via body language, signalling, and bold assumptions. This relies on the assumption that they share some cultural and social references. These games can be adapted to reinstate partial information flow and understanding from some performers, by making one of the performers speak in a majority language. For instance, a variation of *Languages of Love* or *Tower of Babel* pairs majority language performers with minority language performers, and all performers assume they fully understand every one else, creating opportunity for comedic confusion if this is not the case.

Translation-based Games We propose the following translation-based games that we have devised, and which rely on speech recognition, live translation, and a combination of earpieces for performers and surtitles for the audience.

- **Lost in Translation**: this game is the translation-based equivalent of *Stranger in a Strange Land*: a minority language improviser speaks in their own language, with live translation. While the audience can read the surtitles, the remaining majority language improvisers cannot. The status and comedy of this game stem from allowing minority language performers to be understood by the audience while majority language performers struggle to make educated guesses about the meaning of the scene (Fig. 3).
- **Foreign Film**: every performer speaks the same minority language, and the majority language audience sees the subtitled translation of the scene, thus being able to connect with the story told by minority language players.
- **Babelfish**: combining ideas from both above games, we allow one or several minority language improvisers to

speak in that language while live translation is simultaneously shown to the audience (via surtitles) and sent to majority language improvisers (via earpieces). Everyone can understand everyone else - albeit with time delays combined with speech recognition and machine translation errors.

Vocalisation-based Translation Games Finally, and rather than focusing on multilingual understanding, we go back to linguistic experiments – dear to the Surrealists – done purely on the sound of words, and we adapted two existing translation-based games inspired by Raymond Queneau's "Poor lay Zanglay" – a seemingly gibberish text that makes sense in French when read aloud by an English speaker, taken from *Exercices de Style* (Queneau 1947).

- **Telephone Game**: in this simplest of translation-based games, performer A whispers a phrase in their minority language (that only A can speak) to performer B on their right. Performer B then tries to repeat, as well as possible, what they heard to C, who whispers it in turn to D, and so on. At the end of the game, both A and the last performer speak the phrase out loud to the audience and into a translation system, and the original utterance is compared with its repeated distortion.
- **Diplomat**: in this variation of *Telephone Game*, a majority language performer needs to deliver a full speech in a minority language that they do not know. They receive that speech via an earpiece and what they say is translated into majority language via live translation.

While such a list is far from being exhaustive, we believe it exhibits a wide variety in the amount of information that can be transmitted between the performers and from performers to the audience, The varying ratios of information asymmetry and misunderstanding thus create multiple opportunities for comedy.

Structure of a Multilingual Improv Performance

Using the technology and interface described above, along with the various games we have devised, we came up with the following script for a language technology-enabled scene-based improvised comedy show:

1. Part 1: Human Miscommunication:
   (a) *Telephone Game*,
   (b) *Languages of Love*,
   (c) *Tower of Babel*, followed by a replay of that same scene in the majority language, and
   (d) *Diplomat* or a scene with two improvisers trying to perform in a relatively well-known minority language (e.g., French in the UK).
2. Part 2: Machine Translation:
   (a) *Lost in Translation*,
   (b) *Foreign Film*, and
   (c) several language combinations in *Babelfish*.
3. Revelation: the AI behind machine translation takes over.
4. Part 3: Artificial Intelligence-based Improvisation.

As the last two items indicate, we have created a narrative arc in the structure of the show: namely, we add a revelation where the AI tools used for speech recognition, machine translation and text-to-speech end up getting a life of their own and taking over the show.

This revelation serves two purposes. The first one is to remind the audience that today's multilingual communication is enabled by machines, specifically by pattern recognition-based AI algorithms, that present their own limitations and sources of errors: the machine take-over in the show serves as metaphor for the rise of imperfect AI in mediating human communication, and the increased risk for miscommunication and misunderstanding due to algorithmic errors. The second one is, as we describe in the following section, a return to the absurdist roots of our language-based games; by delegating some of the language generation in improvised scenes to a machine, we force the performers to find and create meaning outside of the realms of verbal communication.

Experiments with Automatic Translation

In parallel with the development of Rosetta Code, we piloted basic interaction-based experiments with automatic translation, the most significant being the *automatic translation toward meaninglessness (ATTM)*. The ATTM system allows for automatic homophonic translation, also known as allographic translation or transphonation. The origins of this linguistic genre can be traced back to at least 1450, with the English-Latin transphonation "Mare eate ootys", now more colloquially known as "Mairzy Doats" (Kington-Oliphant 1886; Opie 1952). ATTM is a digital interface between a human and a web-enabled computer serving a webpage. The automatic process works as follows:

1. ATTM takes as input any single line of text from a human interacting with it. For instance, the human might say: "the sun sets behind the mountain, as the snow relents." ATTM's speech recognition system[6] would attempt to convert the captured audio inputs to text. The recognition is not perfect and the minor errors are where the beauty of ATTM stems from.
2. ATTM then uses text-to-speech to synthesize the text to audio which is played over the computer's speakers.
3. While ATTM is synthesizing the new line, it is simultaneously listening with the microphone to the synthesized sound. That is, it is listening to itself speak and attempting to recognize its own words.
4. ATTM loops forever.

ATTM can speak to itself in this endless, yet continuously degrading, loop. There is a minor modification which introduces the difficulties of multiple language understanding. Rather than setting the system to recognize the spoken text as an English sentence, it instead recognizes the spoken audio as if it were a French speaker saying a French sentence. Obviously, the English sentence doesn't sound like a French sentence, but the system does its best to recognize the words

[6] https://cloud.google.com/speech-to-text

spoken and parse it into a French sentence. Then, the French sentence can be translated to an English sentence, and the process can repeat. The system delights in that it progresses toward a more "French" sounding English phrase.

This small experiment parallels the surrealist linguistic work of Douglas Barbour in the 1980s with "homolinguistic translation" (Barbour and Scobie 1981). In that work, the author played with the sound of English phrases by turning them into something that the same sounds but wildly different meanings. For example, "The Pirates of Penzance" becomes "The Pirates of Pen's Chance".

We can very easily enable Automatic Translation Toward Meaninglessness within our *Rosetta Code* framework, simply by listening to text-to-speech synthesis using a loudspeaker instead of in-ear headphones. If both the microphone and the loudspeaker are on, the system will continuously feed on its own outputs. This situation can also be avoided by programmatically deactivating speech recognition during text-to-speech synthesis, or by using a microphone with an on-off switch that is triggered only when a person talks.

Methods for Multilingual AI Improv

We further combined the idea of multilingual improv with real-time translation, as we do in *Rosetta Code*, to implement a non-English version of AI improv comedy such as *Improbotics* and to enable an existing AI improv chatbot to work in multiple languages. Specifically, we have made two multi-lingual version of AI improv shows, one performed in Sweden (in Swedish and in English) and one in Belgium (in Flemish and in English).

Out of several options to perform the local language Improbotics show, the casts considered: a) fine-tuning the models into another language, b) completely retraining the model from scratch, or c) keeping the chatbot as is, but adding translation from and to a different language. This last option can either: 1) translate to the target language when pronounced in the earpiece and keep the interaction in English in the interface, or 2) translate from and to the language of the chatbot and thus allow the target language in the interface.

The first option, fine-tuning the GPT-2 model into another language, e.g. using Dutch corpora, assumes that the transformer model learned transferable language structures internally, giving an advantage when fine-tuning given a different language. While there are successful multilingual transformers, e.g. multilingual BERT (Devlin et al. 2018; Pires, Schlinger, and Garrette 2019), these were trained from scratch on multiple languages. A multilingual GPT-2 did not exists at the time of the production, so fine-tuning would have needed to use the English GPT-2. While using English tokenisation has performed decently for Dutch transformer models in classification tasks (Delobelle, Winters, and Berendt 2020), GPT-2's English tokenizer, vocabulary and pre-training would have likely limited the linguistic correctness of generated Dutch sentences. The second option (retraining a GPT-2 model from scratch into the target language and then fine-tuning) was too costly for the production of the show (Synced 2019). We therefore chose the third option, i.e. to add a translation service, and thus treat the improvising AI like a *Rosetta Code* actor. The interface translates all human input from Dutch to English, and translates all the AI responses from English back into Dutch. Since the inputs are typed manually, and since Dutch-English translation is of high quality, the overall performance of the AI improv does not suffer. Using the translation service also opens up interesting routes for the future of Rosetta code. For instance, we are developing structures where the chatbots whisper generated responses to different performers in languages that they may or may not speak.

Rosetta Code on Stage

Rosetta Code was designed in 2018 and performed as a full show twice in November 2019 at the Voila! Europe Theatre Festival, "multilingual festival, often programming performances that use 2-3 languages in the same show" (Deyzac and Tasker 2020) at the Rich Mix theatre.[7] The framing of the show was detailed in the shows' description in the festival program, and on-stage at the beginning of the performance, making clear to the audience that the computer assists the creativity of the improvisers using automatic translation (Colton, Charnley, and Pease 2011). Covering 9 languages (Arabic, Dutch, English, French, German, Italian, Polish, Norwegian, and Swedish), it was presented as a multi-lingual Turing Test (Turing 1950) where the challenge for the audience was "to decipher who is human and who is a robot"[8].

The Rosetta Code system was situated in the theatre setting. The show itself contextualized the technical details of the show. The show received positive response from performers, audience members, and critics. For instance, one quote from a performing musician reinforces the novelty and innovation of the show:

> *If you get a chance to see these guys, and it's cross country, then see them. Polyglots improvising cross languages with other polyglots/native speakers. Improvisers improvising with AI. They will blow your mind. I have been lucky to improv with them the last few nights.*[7]

The festival organisers observed the comedic potential of translation and miscommunication:

> *Rosetta Code is a fascinating experiment in how we create meaning. There were times when the audience had more understanding than the performers, as we could see the projected translations as well as gestures and body language. This imbalance of understanding can be a rich source of comedy.*

The show was previewed and reviewed by multiple theatre critics and experts in innovative technical improvisation.[7]

Select quotes from reviews of the show include reference how the technology initially is intimidating but ultimately augments the humans. They also acknowledge that the system is not perfect, but rather, that there is beauty in the mistakes that are made.

[7] http://j.mp/rosetta-code-supplementary

[8] https://richmix.org.uk/events/rosetta-code/

. . . what started as a potentially daunting evening . . . , ended up being one of light entertainment, sprinkled with some good humour and some sophisticated technology used to a less sophisticated, but definitely charming, purpose.

Observers also noted that the AI tools that underlay the show build iteratively over the course of the performance.

The AI is first featured in the background of things — providing translations or coming up with full dialogue sentences delivered to the actors via a headset, which they then act out on stage — and later at the forefront, with the above mentioned, absolutely non-scary robot taking center stage, teamed up with a human actor for some one–on–one improvisation.

An edited video, based on the one-hour performance of Rosetta Code, has been uploaded to YouTube[9].

Finally, the Rosetta Code technology has also been deployed in two local language AI Improv shows in Sweden and in Flanders[10].

Historical Context, Related Work, Discussion

Rosetta Code is a first attempt at building augmentative tools for performing translation-based theatrical improvisation in any language. There are many challenges (some unforeseen) to building and deploying such a system. In this section we discuss challenges associated with improv in any language, explore the cultural importance of multi-lingual artistic performance, and provide several directions for future work.

Historical Context

Communication with and the understanding of other humans are fundamental properties of the shared human condition. The challenge of global communication is captured in the story of the Tower of Babel, an origin story that provides an explanation as to why the world's people speak different languages. This story is echoed in indigenous legends around the world. The notion is that if all of the people of the world spoke the same language, their combined capacities and abilities would know no bound.

The Rosetta Stone, after which *Rosetta Code* is named, is a historical artifact discovered in 1799 (Girard 1799). On it are three versions of a decree issued in Egypt in 196 BC. The texts are in Ancient Egyptian (hieroglyphic and Demotic) and Ancient Greek. With only minor differences between the versions in different languages, the Rosetta Stone became key to deciphering Egyptian hieroglyphs and opened a window into the study and understanding of ancient Egyptian history (Champollion 1828).

Nowadays, we cannot expect that others have the same language or lived experience as we do. Some say that the inherent problems of translation in cultural and political contexts have long plagued diplomacy.

. . . it is argued that the translation strategies adopted by the media contribute to actively shape international relations, and that translation activities deserve to be attentively taken into consideration by policy and opinion makers, as well as by the general public (Zanettin 2016).

Rosetta Code provides a platform to explore the power of translation in meaning making between collaborating humans. We acknowledge that there is much to be understood about others through interactions that are non-language based. Dialogue between individuals, and performers on stage, can happen through many means, including but not limited to vocalized language. They might use body language, size, shape, and speed to construe meaning to each other. We note that these channels of communication are complimentary and can be used in concert to build shared understanding between people.

Improv, Thinking, and Language

Verbal and Non-verbal Improvisation Improvisational theatre is theatre that is created and performed at the same time. It can be seen as a constrained human interaction game and has been qualified as "real-time dynamical problem solving" (Magerko et al. 2009; Johnson-Laird 2002) in the settings of both jazz music and theatre. Improv requires performers to exhibit, among others, acute listening to both verbal and non-verbal suggestions coming from the other improvisers, short- and long-term memory of narrative and character elements, and practised storytelling skills (Johnstone 1979). One could categorise the behaviour of improvisers using the Dual Process psychological theory (Wason and Evans 1974). That theory distinguishes "system 1" cognitive processing corresponding to fast, intuitive, instinctive, and emotional reactions—which can be honed using theatrical actor training practice (Benedetti 1999) to be able to react truthfully in the moment (Meisner and Longwell 2012)—and "system 2" reasoning which is slower, more deliberative, logical and, in the case of improv, often more verbal (Evans 1984; Kahneman 2003).

The Languages of Improv Improv is performed in many different languages around the world.[11] Many times, improvisors share the stage with others who do not speak the same first language. In these situations, the performers must find a level of scenic understanding using more than linguistics and meta-pragmatic dialogue (Sawyer and Sawyer 2003). They must respond to experience with intuition (Spolin 1963), and make inferences about meaning. Sometimes the performers resort to using gibberish, finding that it might be easier to use a language that none of them, nor the audience understand (Johnstone 2014).

Misunderstanding in Improv In improvisation, there is an inherent tension between understanding and misunderstanding. This is based on the nature of the artform. It is an artform where choices are simultaneously made and understood in the same moment. The audience is also experiencing these choices in the same moment that the collaborating improvisors on stage hear them. Moment-to-moment

[9] https://www.youtube.com/watch?v=P6vpL1CqeCs

[10] http://www.erlnmyr.be/2020/02/28/improbotics-in-de-media/

[11] http://bit.ly/improv-worldwide

understanding is critical to the progression of an improvised performance.

Improvisation for Language Learning Improvisation has been used as a form of language practice and learning. Theatrical improvisation promotes a special kind of verbal flow that may be particularly well suited to language learning (Egbert 2003). By engaging in role-playing activities, improvised with a variety of scene partners, language learners can engage in situated experiential learning (Butt 1998).

Augmentative AI Language Technology

Modern speech recognition, machine translation and dialogue systems, three popular fields of research in artificial intelligence (AI), share the same underlying mechanism: statistical language models trained on large corpora of text (Brants et al. 2007). Modern language models are implemented as neural networks and they estimate the likelihood of the next word or character token given some context about previous tokens (Bengio et al. 2003). Relatively similar neural networks models can be used for translation (i.e. generating a sentence in language B that corresponds to a context in language A), speech recognition (i.e. generating a sentence of symbols that corresponds to a sequences of acoustic phonemes), and in chatbots (i.e. generating sentences of dialogue likely to follow a given conversation context). Some examples of such modern text generation models include *GPT-2* (Radford et al. 2019), *Turing-NLP* (Rosset 2020) and *Meena* (Adiwardana et al. 2020). These models cannot ground language understanding in the human sense. They manage to recognise speech, translate sentences with near human-level accuracy, generate plausible responses, and solve language comprehension tasks all owing to the large amounts of training data and a well-defined training objective.

These limitations can be linked to classic theatre theory and actor preparation. Complaining about old-fashioned actor training, the seminal Russian theatre director and theorist Stanislavski explained that "the mistake most actors make is that they think about the result instead of the action that must prepare it" (Stanislavski 1936). That is, most actors understand *what* to say next, but not *why* they might say such a thing, or *how* it might be organized in a deliberate and impactful way (actor training has, since then, changed). This mirrors the capacity of the computation language models, which know the likelihood of what might come next, but not why or how it should follow. Neural models have a grammar, as Chomsky (2002) would describe it: "a device of some sort for producing the sentences of the language under analysis" but not a rationale as to why such a device is meaningful in context. Such models generate sentences which are grammatically correct but semantically nonsensical (Chomsky 2002).

These language models serve as building blocks for augmentative theatre technology, allowing performers to bring AI onto the stage. We demonstrated how to augment English-centric dialogue systems and extend the applicability of such technology to non-English improv, by combining chatbots with translation for multilingual AI improv shows.

Conclusion

In this paper we have presented *Rosetta Code*, a technology-based show structure that allows for improvised theatre in any language. We described the technical details underlying the system, and the theatrical context within which the system is situated. *Rosetta Code* is a show about language understanding via communication, that uses and celebrates the tools developed specifically to enable communication, namely speech recognition, machine translation and text-to-speech. We provide results and analysis of the first 3 shows covering 9 languages, as well as evaluative feedback from performers, producers, audiences and critics. Future work will involve systematic quantitative evaluation, enabled both by online performances and by sharing the tools with more improv troupes. Our framework allows for performers and audiences around the world to enjoy improvised theatre in any language, and we hope that these tools and techniques will empower and augment the art form. We will also explore incorporating language technologies with computationally creative systems for video and music generation.

Acknowledgements

The authors would like to thank Sharlit Deyzac and Amy Clare Tasker from Voila Europe Festival of European Theatre, as well as Mats Eldøen, Alice Major, Adam Meggido, Julian Faid for helpful discussions. Thomas Winters is a fellow of the Research Foundation-Flanders (FWO-Vlaanderen). The performers and creative cast in the November 2019 performances of Rosetta Code included: Boyd Branch, Harry Turnbull, Jenny Elfving, Kieran Stallard, Kory Mathewson, Marouen Mraihi, Paul Little, Piotr Mirowski, Rhiannon Jenkins, Roel Fox and Sarah Davies.

References

Adiwardana, D.; Luong, M.-T.; So, D. R.; Hall, J.; Fiedel, N.; Thoppilan, R.; Yang, Z.; Kulshreshtha, A.; Nemade, G.; Lu, Y.; et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Barbour, D., and Scobie, S. 1981. *The Pirates of Pen's Chance: Homolinguistic Translations*. Northwestern University Press.

Benedetti, J. 1999. *Stanislavski: his life and art*. Methuen Drama.

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Boal, A. 2006. *The aesthetics of the oppressed*. Routledge.

Brants, T.; Popat, A. C.; Xu, P.; Och, F. J.; and Dean, J. 2007. Large language models in machine translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Bruce, A.; Knight, J.; Listopad, S.; Magerko, B.; and Nourbakhsh, I. R. 2000. Robot improv: Using drama to create believable agents. In *IEEE International Conference on Robotics and Automation*, volume 4, 4002–4008. IEEE.

Butt, R. 1998. Notes from the classroom: Improvisation and language acquisition: Actor training as a tool to enhance fluency. *College ESL* 8(2):58–63.

Champollion, J.-F. 1828. *Précis du système hiéroglyphique des anciens Égyptiens, ou Recherches sur les élémens premiers de cette écriture sacrée, sur leurs diverses combinaisons, et sur les rapports de ce système avec les autres méthodes graphiques égyptiennes avec un volume de planches*. Imprimerie royale.

Chomsky, N. 2002. *Syntactic structures*. Walter de Gruyter.

Colton, S.; Charnley, J. W.; and Pease, A. 2011. Computational creativity theory: The face and idea descriptive models. In *ICCC*, 90–95.

Delobelle, P.; Winters, T.; and Berendt, B. 2020. Robbert: a dutch roberta-based language model. *arXiv preprint arXiv:2001.06286*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Deyzac, S., and Tasker, A. C. 2020. Private communication.

Egbert, J. 2003. A study of flow theory in the foreign language classroom. *The Modern Language Journal* 87(4):499–518.

Esslin, M. 1960. The theatre of the absurd. *The Tulane Drama Review* 3–15.

Evans, J. S. B. 1984. Heuristic and analytic processes in reasoning. *British Journal of Psychology* 75(4):451–468.

Girard. 1799. Rosette, le 2 fructidor an 7. *Courrier de l'Égypte* (37).

Jacob, M.; Chawla, P.; Douglas, L.; He, Z.; Lee, J.; Sawant, T.; and Magerko, B. 2019. Affordance-based generation of pretend object interaction variants for human-computer improvisational theater. In *ICCC*, 140–147.

Johnson-Laird, P. N. 2002. How jazz musicians improvise. *Music Perception* 19(3):415–442.

Johnstone, K. 1979. *Impro: Improvisation and the Theatre*. Faber and Faber Ltd.

Johnstone, K. 2014. *Impro for storytellers*. Routledge.

Kahneman, D. 2003. A perspective on judgment and choice: mapping bounded rationality. *American psychologist* 58(9):697.

Kington-Oliphant, T. L. 1886. *The New English*, volume 1. Macmillan and Company.

Liu, R.; Christopher, C.; Martens, C.; Ontañón, S.; Mirowski, P.; Mathewson, K. W.; Winters, T.; and Farrugia, S. 2019. Playable experiences at the 15th aaai conference on artificial intelligence and interactive digital entertainment (aiide'19). In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, 234–238.

Magerko, B.; Manzoul, W.; Riedl, M.; Baumer, A.; Fuller, D.; Luther, K.; and Pearce, C. 2009. An empirical study of cognition and theatrical improvisation. In *Proceedings of the seventh ACM conference on Creativity and cognition*, 117–126.

Mathewson, K. W., and Mirowski, P. 2017a. Improvised theatre alongside artificial intelligences. In *AAAI AIIDE*.

Mathewson, K. W., and Mirowski, P. 2017b. Improvised comedy as a turing test. In *NeurIPS Workshop: Machine Learning for Creativity and Design*.

Mathewson, K. W., and Mirowski, P. 2018. Improbotics: Exploring the imitation game using machine intelligence in improvised theatre. In *AAAI AIIDE*.

Mathewson, K. W. 2019. *Humour-in-the-loop: Improvised Theatre with Interactive Machine Learning Systems*. Ph.D. Dissertation.

Meisner, S., and Longwell, D. 2012. *Sanford Meisner on acting*. Vintage.

Mirowski, P., and Mathewson, K. W. 2019. Human improvised theatre augmented with artificial intelligence. In *Proc 2019 Creativity and Cognition*, 527–530.

Opie, I. A. 1952. *The Oxford dictionary of nursery rhymes*. Clarendon Press.

Pires, T.; Schlinger, E.; and Garrette, D. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Queneau, R. 1947. *Exercices de style*. Gallimard.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1(8):9.

Rosset, C. 2020. Turing-nlg: A 17-billion-parameter language model by microsoft.

Sawyer, R. K., and Sawyer, R. K. 2003. *Improvised dialogues: Emergence and creativity in conversation*. Greenwood Publishing Group.

Spolin, V. 1963. Improvisation for the theatre 3rd ed.

Stanislavski, C. 1936. *An actor prepares*. Routledge.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Synced. 2019. The Staggering Cost of Training SOTA AI Models. https://syncedreview.com/2019/06/27/the-staggering-cost-of-training-sota-ai-models/.

Tiedemann, J. 2009. News from opus-a collection of multilingual parallel corpora with tools and interfaces. In *Recent Adv. in Natural Language Processing*, volume 5, 237–248.

Turing, A. M. 1950. Computing machinery and intelligence. *Mind* 59(236):433–460.

Vinyals, O., and Le, Q. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Wason, P. C., and Evans, J. S. B. 1974. Dual processes in reasoning? *Cognition* 3(2):141–154.

Winters, T., and Mathewson, K. W. 2019. Automatically generating engaging presentation slide decks. In *International Conference on Computational Intelligence in Music, Sound, Art and Design*, 127–141. Springer.

Zanettin, F. 2016. 'the deadliest error': translation, international relations and the news media. *The Translator* 22(3):303–318.