Surprise Languages: Rapid-Response Cross-Language IR

Douglas W. Oard
Marine Carpuat
Petra Galuščáková
Joseph Barrow
Suraj Nair, Xing Niu
Han-Chin Shing
Weijia Xu, Elena Zotkina
University of Maryland, USA

Kathleen McKeown
Smaranda Muresan
Efsun Selin Kayi
Ramy Eskander
Chris Kedzie
Yan Virin
Columbia University, USA

Dragomir Radev*
Rui Zhang*
Mark Gales†
Anton Ragni†
Kenneth Heafield‡
*Yale University, USA
†Cambridge University, UK
‡University of Edinburgh, UK

ABSTRACT

Sixteen years ago, the first "surprise language exercise" was conducted, in Cebuano. The evaluation goal of a surprise language exercise is to learn how well systems for a new language can be quickly built. This paper briefly reviews the history of surprise language exercises. Some details from the most recent surprise language exercise, in Lithuanian, are included to help to illustrate how the state of the art has advanced over this period.

CCS CONCEPTS

• **Information systems → Retrieval effectiveness**.

KEYWORDS

Fast, retrieval, evaluation

ACM Reference Format:
Douglas W. Oard, Marine Carpuat, Petra Galuščáková, Joseph Barrow, Suraj Nair, Xing Niu, Han-Chin Shing, Weijia Xu, Elena Zotkina, Kathleen McKeown, Smaranda Muresan, Efsun Selin Kayi, Ramy Eskander, Chris Kedzie, Yan Virin, Dragomir Radev, Rui Zhang, Mark Gales, Anton Ragni, and Kenneth Heafield. 2019. Surprise Languages: Rapid-Response Cross-Language IR. In *Proceedings of EVIA '19*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The surprise language exercises were born at 4:17 A.M. on March 5, 2003, when the US Defense Advanced Research Projects Agency (DARPA) designated Cebuano, the second most widely spoken indigenous language in the Philippines, as the focus of the first surprise language exercise [8, 9]. Teams were given 10 days to assemble language resources and to create whatever human language technology they could in that time. Surprise language exercises have since grown up, and in March of 2019 they turned sixteen. As if to celebrate, the Intelligence Advanced Research Projects Activity (IARPA), at 4:14 PM on April 1, 2019 started the cycle again, this time with Lithuanian. In this short paper we review the early days of surprise language exercises, contrasting that early activity with what is possible today.

EVIA '19, 0.00
2019. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 GENESIS

Our story begins with the failure of machine translation. After substantial investment, the U.S. National Research Council's Automatic Language Processing Advisory Committee (ALPAC) concluded in 1966 that current and foreseeable machine translation technology would not be able to approach human performance at translation tasks any time soon [3]. Given the limited number of languages that were of strategic interest at the time, they concluded that it would make more sense to invest in language education than to invest in the immediate development of operational language technologies. They did, however, recommend continuing research investments. Nonetheless, many language technology researchers mark the ALPAC report as the start of the "Natural Language Processing (NLP) winter" in which large-scale investments in language technology essentially dried up.

A bit over twenty years later, things began to change. The first major advance was the development of Statistical Machine Translation (SMT) at IBM, for which the first published paper dates to 1988 [2]. The early work on SMT drew inspiration from the success of Hidden Markov Models in speech recognition, which dates to at least 1980 [4]. Of course, what has been called the "statistical turn" in NLP [6] undoubtedly also owes much to the contemporaneous development of statistical techniques for information retrieval.

Early work on SMT was, perhaps unsurprisingly, directed towards what ultimately came to be called high-resource languages (such as French), since the early goal was to find out well this new approach could do. That was soon to change, however. The key impetus was the dissolution of the Soviet Union that effectively ended the Cold War on December 26, 1991. This was quickly followed by an explosion of what came to be called globalization, with the European Union's Treaty of Maastricht in 1993, the public introduction of the World Wide Web that same year, and China's accession to the World Trade Organization in 2001 as some major milestones. This push toward globalization generated substantial commercial interest in machine translation. The decade of the 1990's was also one of great ferment in what came to be called Cross-Language Information Retrieval, first at the Text Retrieval Conference (TREC), and towards the end of the decade at the NACSIS Test Collection for Information Retrieval (NTCIR) and the Cross-Language Evaluation Forum (CLEF) evaluations in Japan and Europe, respectively.

The terrorist attacks of September 11, 2001 in the United States brought the fourth piece of the puzzle into focus, making it clear that the ALPAC report's conclusion that language learning was a

scaleable solution for "strategic languages" was simply no longer true. As the ALPAC report had anticipated, investments in basic technology had by then started to pay off. Large investments were being made in advancing the state of the art in SMT, most notably in the DARPA program on Translingual Information Detection, Extraction and Summarization (TIDES). This investment was at the time largely focused on improving translation quality for high-resource languages, but statistical methods also had obvious potential for rapid development. This was the case because, unlike earlier knowledge-intensive methods, much of the work to be done by statistical methods could be done by the machine. It was therefore in this context that the DARPA TIDES program organized two surprise language evaluations in 2003.

3 TIDES SURPRISE LANGUAGE EXERCISES

The plan for the TIDES surprise language exercises involved two phases. In the first phase, a ten-day "dry run" data collection exercise was planned, just to be sure that the language data that was needed for such an exercise could indeed be rapidly collected. In the 29-day second phase, conducted on a different language, the goal would be to both collect the language data that was needed and to build systems for that language. In recognition of the broad range of systems that were of interest to TIDES participants, no restrictions were placed on what systems were to be built. The basic idea was that research teams would simply try out what they were already working on using some new language.

No plan survives its first contact with reality, and so it was with the data collection dry run, for which Cebuano was the language chosen by DARPA. Data was indeed collected, including a million words of parallel text and several types of translation lexicons, but the real surprise in the dry run was the number of systems that could be constructed for a new language in so short a period. Ignoring the intent of the dry run to simply focus on collecting langauge data, participating teams also built Cebuano systems for entity tagging, part of speech tagging, noun phrase chunking, time expression detection, stemming, morphological analysis, machine translation (five systems), CLIR (three systems), and summarization (two systems) [7]. The first publication on this work was submitted 60 hours into the dry run, written so rapidly that apparently nobody (including the reviewers!) detected the spelling error in the title [8].

With this experience behind them, the fifteen participating teams in the actual surprise language exercise thought they were ready. Not so. One June 2 of the same year, DARPA selected Hindi as the surprise language. In this case, the biggest surprise was that 10 days later there were no systems at all that were capable of operating on general Hindi text. The problem was not the technology but rather the character encoding. At the time, proprietary fonts, each with different digital encodings, were the norm for Hindi. As a result, systems trained on text from one source simply would not work on text from another. The challenge was compounded by the fact that characters in the Devanagari script used to write Hindi were typically represented as a sequence of encodings for parts of the character, and these encodings needed to be reverse engineered for each source. Two weeks into the surprise language exercise this problem was overcome by development of character set normalization, and after that an even broader range of language technologies were developed. Among the notable systems built during this time were creation – from scratch – of a precursor to what people would recognize today as Mechanical Turk to obtain translations [12] and creation of an interactive system for answering factual questions posed in English that did so with reference only to Hindi documents [10].

At least three important things were learned from this surprise language exercise. First, with few exceptions, the systems that were built were systems of the same type that people were already working on; only the language was different. Second, rapidly built systems were, unsurprisingly, not as good a those that had been the focus of longer development cycles. And third, after the surprise language exercise ended, people generally preferred to get back to their work on making systems better, rather than making them more rapidly. This third factor led to what might be termed the "surprise language winter," as the next surprise language exercise wasn't conducted until 2018. When it was, the world was quite a different place. In the mean time, two other programs, DARPA's LORELEI program and IARPA's Babel program, had focused on building component technologies for low resource languages. The table was thus set for a surprise language renaissance, this time with complete application-scale systems as the focus.

4 MATERIAL

One thing that has remained constant is that you seemingly can't run a research program until you have a good acronym. For the next surprise language it was not DARPA, but rather IARPA, that would run the program, which is called Machine Translation for English Retrieval of Information in Any Language (MATERIAL). Unlike TIDES, which had focused on component technologies, the focus of MATERIAL is on end-to-end systems.

One notable characteristic of the work in the TIDES surprise language exercise is that it was focused entirely on text. MATERIAL, by contrast, includes both text and speech. End-to-end systems in MATERIAL ingest text and speech in some language other than English, they accept queries in English, and they produce a set of English text summaries for (hopefully!) relevant text and speech sources.

A second important contrast is that teams in MATERIAL start with a "language pack" that includes the minimal language data needed to train speech recognition and machine translation systems and to evaluate information retrieval performance. At the time of the TIDES surprise language exercises there had been some debate about whether systems should be built on the fly, as in the TIDES surprise language, or in advance. There were principled arguments on both sides, with advocates of pre-building noting that maintaining the infrastructure for rapid response is both expensive and limiting, while advocates for building on the fly noted that advances in the state of the art could require redoing all of the work. Language packs essentially represent a middle ground in which the language data is assembled in advance and then whatever technology exists at the time can be used to build or re-build the systems when they are needed.

The language packs in MATERIAL have three parts: a build pack containing speech recognition and machine translation training

Pack	Documents	Queries	Relevant
BUILD	T:13,177 / S:303		
ANALYSIS	T:614 / S:215	300	476
DEV	T:433 / S:238	300	505

Table 1: Lithuanian language pack statistics (T=Text, S=Speech). Relevant is the total number of relevant documents, across all queries.

data, a development pack to support parameter selection for end-to-end systems, and an analysis pack that can support both component and end-to-end evaluation. Statistics of these collections are shown in Table 1.

The machine translation training data in the Lithuanian build pack contains 835,516 English words and 609,621 Lithuanian words in 42,635 translation-equivalent sentences. The speech recognition portion of the build pack consists of conversations recorded over cellular phone networks. Each conversation involves two participants discussing everyday subjects. The audio is sampled at 8 kHz which is a common sampling rate for Conversational Telephone Speech (CTS) data sets. The duration of audio excluding silence, unintelligible, mispronounced and fragment words is estimated to be 50 and 10 hours for the training and tuning portions of the speech recognition build pack, respectively. The audio in the analysis and the development testing (DEV) packs on which we evaluate initial systems consists largely of news and topical broadcasts, along with smaller amounts of CTS data that is similar to that used for speech recognition system development. The much higher sampling rates of 44.1 and 48 kHz used for news and topical broadcast data adds additional complexity to the existing mismatch between training and testing domains. The duration of speech-only audio in the CTS, news and topical broadcast parts of the ANALYSIS and DEV packs is estimated to be 1, 3 and 6 hours, respectively, in each pack.

5 LITHUANIAN AS A SURPRISE LANGUAGE

The design of the MATERIAL program includes three cycles, each of which included a development period using one or more training languages, followed by a surprise language exercise. The surprise language exercises will be progressively shortened, with the third surprise language exercise running for just a few weeks (seven, in current plans). In reality, however, the release of any new language can be treated as a surprise language. On April 1, 2019, IARPA released the two "practice languages" for the second cycle's development period, Bulgarian and Lithuanian. We elected to perform a surprise language exercise on Lithuanian in order to self-assess our ability to rapidly build end-to-end systems for a new language.

The US National Institute of Standards and Technology (NIST) distributed the build packs for Bulgarian and Lithuanian at 2:43 PM[1] on Monday April 1. The languages were not identified by IARPA until 4:14 PM, by which time language identification had correctly identified the languages and we had selected Lithuanian.[2]

[1] All times are Eastern Daylight Time.

[2] Our team includes a Bulgarian speaker, and we prefer to assess our capabilities on languages for which we have no language expertise.

By 5:41 PM on Tuesday April 2, we had built a dictionary-based query translation CLIR system using a quite small dictionary extracted from Wiktionary [1]. Only the build pack was available at that point, so we initially evaluated that system on a test collection that we built from the machine translation training data in the build pack, achieving a Mean Average Precision (MAP) of 0.04. Lithuanian "documents" were created for this test collection by grouping sentences which had been selected from the same document, information which was available in the metadata of the build pack. Queries were built from the associated English documents as follows. First, English stopwords were removed, then we found all unique {1,2,3,4}-grams of English words. 100 queries of each sequence length were sampled from this set of n-grams, totaling 400 queries. Our sampling strategy was designed to generate a diverse range of high, medium, and low frequency word n-grams, as n-gram frequency is related to the number of relevance judgments for that query. Because we constructed this test collection from parallel text, positive relevance judgments were simply the Lithuanian documents whose associated English document contained the sampled n-gram. We ended up with 5,978 positive relevance judgments for the 400 queries.

By 8:30 PM on Wednesday April 3, we had built seven CLIR systems using Probabilistic Structured Queries (PSQ) [11]. The best of these achieved a MAP of 0.08 on the MT Training test collection by using translation probabilities inferred from several Lithuanian-English bilingual dictionaries (assigning higher probabilities to translations that appeared in more dictionaries).

By 6:45 PM on Thursday April 4 we had improved our MAP to 0.41 on the MT Training test collection by using PSQ with SMT translation probabilities. These SMT translation probabilities were trained using Giza++ from the same MT training data as we were using for evaluation, so at this point our evaluation results became sanity checks rather than fair evaluations.

By 8:40 PM on Friday April 5, we had further improved our MAP to 0.43 on the MT training test collection by using one-best neural machine translation built using the Marian toolkit [5] to translate the documents into English (again, noting that this is a test-on-training condition).

During the first week we also created a Lithuanian Automatic Speech Recognition (ASR) system, initially at 5:22 PM on April 2 (55% Word Error Rate (WER)). An improved ASR system was produced at 5:00 AM on April 3 (41% WER). Our third ASR system was produced during the next week at 4:07 PM on April 9 (36% WER).

Over the weekend we did use some machine time to perform machine translation, but we suspended development work.

By 10:48 PM on Monday April 8, the fifth full day of our surprise language exercise, we had achieved a MAP of 0.82 on the MT Training test collection by performing post-retrieval system combination on the results from four CLIR systems. Because some of these systems were trained on the same MT training data, these are again unfair evaluation results that were useful principally as sanity checks. The development and analysis packs were released by NIST on Monday April 8 at 5:13 PM.

At 2:50 PM on Tuesday April 9, our first CLIR results on the development pack became available (AQWV=0.21 on DEV for a single PSQ system, indexing only text documents). Actual Query

Figure 1: An example summary for a conjunctive query for "food shortage" AND symptom (as in the symptom of an illness). Expressed in the MATERIAL query language, the query is "food shortage",symptom+[syn:a sign of illness].

```
CLOSE MATCH (food shortage):
...can cause stress, some food products, air change, lack of
food, misunderstandings, as well as many other factors.
Klasterinis headache pain It is quite rare, strong headache,
which is more widespread between men than women. Klasterinis
headache may arise one time during the day...

MOST RELEVANT SENTENCES (symptom):
Antrines or symptoms of headache
Headache often causes stress and anxiety, but rarely is serious
symptoms of disease.
Headache - this is the most widespread symptom, whose reason can
be very different.
Pain headache - it pain headaches and cocoa area, which means
health disorders or symptoms of...
```

Weighted Value (AQWV) is the set-based evaluation measure used in the MATERIAL program, which is computed as a weighed linear combination of misses and false alarms.[3] Based on past experience with other languages, we used a fixed cutoff at 2 documents to create sets from ranked lists for this evaluation.

By 4:43 PM on Wednesday April 10 our best AQWV on DEV was 0.40, obtained using PSQ with SMT translation probabilities that were trained on both the build pack and the larger ParaCrawl Lithuanian-English parallel text,[4] again at a fixed cutoff at 2 documents. Both text and speech were indexed for this and subsequent experiments.

By 4:02 PM on Thursday April 11 our best MQWV on DEV was 0.54, where the Maximum Query Weighted Value (MQWV) is an oracle result for the highest obtainable AQWV, based on a post-hoc threshold sweep to learn the optimal rank cutoff. This system used post-retrieval system combination on the results of four CLIR systems. Using the same system combination and the same system parameters, an AQWV of 0.47 was achieved on the ANALYSIS collection.

At 10:39 PM on Friday April 12, automatic generation of summaries for each document returned by Thursday's CLIR system for every one of the 300 queries in the ANALYSIS set was completed. Subsequent manual evaluation of the summaries over the following two weeks, using crowd workers to judge summary relevance to each query, resulted in an AQWV of 0.19 for an end-to-end interactive system in which documents marked by crowd workers as non-relevant were dropped. A sample summary is shown in Figure 1. For comparison, the same summarization and evaluation approach yielded an AQWV of 0.34 when using reference human transcriptions (for speech) and reference human translations (for text and speech) to construct the summaries.

Not counting the weekend, 9 days, 6 hours and 57 minutes had elapsed since NIST had first distributed the Lithuanian build pack.

[3] AQWV is a value in the range [-40,1]. Higher values are better, and an AQWV of 0 can be achieved by returning nothing. AQWV is defined as $1 - (p_{miss} + 40 * p_{falsealarm})$. See https://www.nist.gov/iarpa-material-machine-translation-english-retrieval-information-any-language-program for details.

[4] https://paracrawl.eu/

6 LESSONS LEARNED

What can we learn from these three stories of Cebuano, Hindi and Lithuanian surprise languages exercises? We see three broad messages:

- It helps to know what you are doing. The Cebuano and Hindi surprise language exercises were exploratory, and they did help us to learn what was easy and what was hard. But mostly we learned that what was easy was what we were already doing. So the key is to already be doing what you want done. Getting that to work in a new language takes a little work, but it need not take a lot of time.
- Language packs help. They help in three ways. First, they provide a point at which issues such as character encoding can be sorted out in advance, thus allowing system development to go more smoothly. Second, even though we can and should seek our additional resources at the point when the need for a language becomes clear, language packs provide a solid base on which to build and resources that can be used almost immediately to produce initial systems. Third, the evaluation resources in the language pack make it possible to rapidly iterate towards effective systems.
- Earlier access to a test collection containing relevance judgments could have further accelerated our progress on Lithuanian. As it was, by the third day of the evaluation we were generating systems based on MT training data, and we had the ability to index speech. Neither of those could be fairly evaluated in an end-to-end CLIR task for another four days (again, not counting the weekend), so over that period our measurement of progress was limited to sanity checks for newly implemented systems, and those sanity checks were not able to support reliable system comparison.

One need not be interested in surprise to be interested in the MATERIAL surprise languages. Many people, working alone or in small groups, are looking for ways of building systems for specific languages. It seems reasonable to expect that much of the same technology that allows us to build systems quickly will also help us to build systems cheaply. Something like language packs could allow the essential first steps of resource development to be crowdsourced, and experience with getting the most out of small training and test collections will undoubtedly be useful in resource limited settings. Moreover, the repeated examples that MATERIAL will generate over time will, we hope, serve as an inspiration for those who wish to further extend the reach of CLIR more broadly across the human family.

ACKNOWLEDGMENTS

The authors are grateful to Salim Roukos, Carl Rubino, Audrey Tong, Charles Wayne and Ilya Zavorin for contributing to the development of our thinking about surprise language exercises. This research has been supported in part by an Amazon Web Services Machine Learning Research Award and by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the

REFERENCES

[1] Judit Acs. 2014. Pivot-based multilingual dictionary building using Wiktionary. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* (26-31). Reykjavik, Iceland.

[2] P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, and P. Roossin. 1988. A Statistical Approach to Language Translation. In *Proceedings of the 12th Conference on Computational Linguistics - Volume 1*. 71–76. https://doi.org/10.3115/991635.991651

[3] National Research Council. 1966. *Languages and Machines: Computers in Translation and Linguistics*. National Academy of Sciences. http://www.mt-archive.info/ALPAC-1966.pdf

[4] B. H. Juang and Lawrence A. Rabiner. 2005. Automatic Speech Recognition: A Brief History of the Technology. In *Encyclopedia of Language and Linguistics, Second Edition*. Elsevier. https://www.ece.ucsb.edu/Faculty/Rabiner/ece259/

[5] Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-Effective High-Quality Neural Machine Translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. Association for Computational Linguistics, 129–135.

[6] Judith K. Klavans and Philip Resnik. 1996. *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*. MIT Press.

[7] Doug Oard. 2003. Surprise: Its Cebuano. In *Team Tides Newsletter*. http://language.cnri.reston.va.us/TeamTIDES/TeamTIDESapr2003.pdf

[8] Douglas W. Oard. 2003. Desparately Seeking Cebuano. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

[9] Douglas W. Oard. 2003. The surprise language exercises. *ACM Trans. Asian Lang. Inf. Process.* 2, 2 (2003), 79–84. https://doi.org/10.1145/974740.974741

[10] Satoshi Sekine. 2003. Rapid Development of Cross-Lingual Question Anwering Using Information Extraction. In *Team Tides Newsletter*. http://language.cnri.reston.va.us/TeamTIDES/tt02e3-final.pdf

[11] Jianqiang Wang and Douglas W. Oard. 2012. Matching meaning for cross-language information retrieval. *Inf. Process. Manage.* 48, 4 (2012), 631–653. https://doi.org/10.1016/j.ipm.2011.09.003

[12] David Yarowsky. 2003. Scalable Elicitation of Training Data for Machine Translation. In *Team Tides Newsletter*. http://language.cnri.reston.va.us/TeamTIDES/tt02e3-final.pdf